MastersThesis: On the Design of Effective Modular Reconfigurable Grippers: an Iterative Approach

Filippo Sanfilippo, University of Siena, Italy

Università degli Studi di Siena

Facoltà di Ingegneria

Corso di Laurea Specialistica in Ingegneria Informatica

On the Design of Effective Modular Reconfigurable Grippers: an Iterative Approach

Relatore

Chiar.mo Prof. Domenico Prattichizzo

Correlatori

Eng. Gionata Salvietti

Prof. Houxiang Zhang

Prof. Dr. Jianwei Zhang

Tesi di laurea di

Filippo Sanfilippo

A. A. 2010/2011

29 Aprile 2011

I dedicate this Thesis to my dear family and to all my friends.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Grasping is a crucial topic in robotics research. Robots capable of performing grasping tasks are required in industrial application as well as in daily life tasks. Nevertheless, most of the grippers available on the market are usually difficult to adapt to different grasping operations or to the grasping of objects with dissimilar size. Some of these problems are approached using robotic hands, which are however, still very expensive and difficult to control. These are the motivation that lead us to investigate on alternative solutions for grasping. In our opinion, modular robotics is one of the most promising solutions.

As the name suggests, modular robots are composed of multiple building blocks, called modules. They are linked together allowing the transfer of mechanical forces and moments, electrical power and what is needed for the communication throughout the robot. Usually, a modular robot consists of some primary structure with the possibility to add specialized units like grippers, feet, wheels etc.. Studies and realizations of this devices are quite recent, CEBOT [1], was developed on 1988 and later on, thanks to miniaturization of motors and of sensors, several models have been built like, for example, Polipod [2](1993), ATRON [3](2003), M-TRAN [4] (2002) shown in Fig.1.1 and GZ-I [5] System (2008).

Figure 1.1: M-TRAN III: an example of modular robot

Lots of studies on replicating, existing in nature, gaits and on self reconfigurable systems have been made using this kind of robots, especially in a snake-like configuration. Snake-like grasping capabilities have been investigated by Salvietti et al. in [6]. In this work, a task priority based approach is presented in order to manage grasping and locomotion functions starting from the hyper redundant manipulator theory. An example of grasping mode with locomotion capability is shown in Fig.1.2.

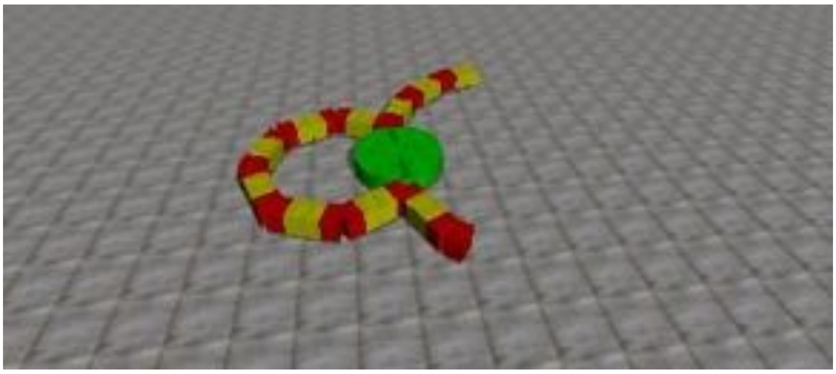


Figure 1.2: A grasping mode with locomotion capability

To the best of our knowledge, few works have been done on the possibility of developing a modular gripper.

In this Thesis, we focus on the advantages obtained by a modular robotic grasping design. In fact, a modular construction permits to overcome some of the problems mentioned above by introducing a possible task-oriented, object-oriented or hand-oriented design of the gripper. The modular approach allow also us to find a trade off between a simple gripper and a more complex human like manipulator. This idea is shown in Fig.1.3.

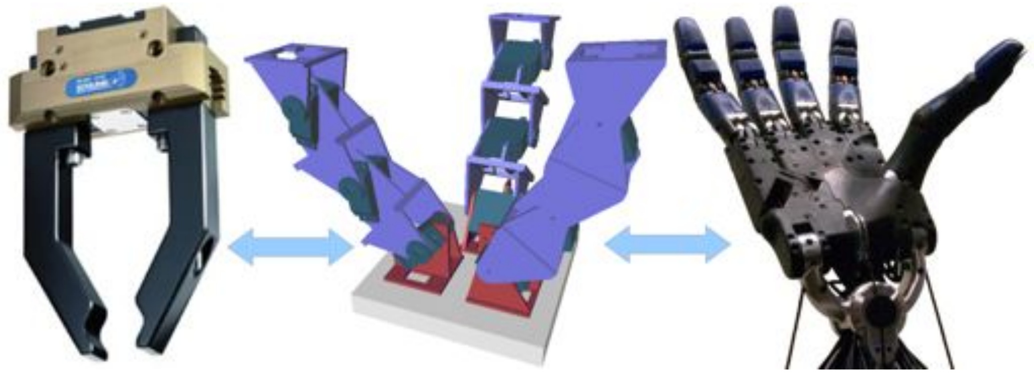


Figure 1.3: The modular grasping idea: thanks to its flexibility our device can reproduce both a simple gripper and a more sophisticated kinematics like an anthropomorphic robotic hand.

We investigate the possibility of developing a modular robotic gripper that allows for easy adaptation to different requirements and situations. In other words, we define the guidelines for creating a device capable of adapting its structure and functionality to the characteristics of an object or a set of objects to be grasped. An algorithm capable of determining efficient modular gripper configurations to get a stable grasp of given objects was developed. It consists of an iterative procedure that consents to design the structure of the gripper. Starting from a simple configuration, the target is to reach a prefixed grasp quality using the minimum number of modules. During the design procedure, the grasp capability of the device is improved adding the necessary modules and fingers and adjusting its base configuration. The possibility of dealing with several objects, set of objects and the easy and fast reconfiguration of the gripper structure was considered.

Furthermore, we built two of the grippers that we obtained using the design algorithm. For our preliminary tests, a simple 1-DOF module was chosen, inspired by the one described in [7]. This low-cost and easy to control module was adopted as the base element for the design of our modular gripper.

Finally, we explored different control procedures for the gripper. Despite the simplicity of a modular manipulator model, with the increase in the number of its fingers and modules, it becomes rival to the human hand in complexity. A possible solution to this problem could be to draw inspiration from the human hand not only with regard to size and configuration, but rather as regards to the control. Recent advances in neuroscience research have shown that control of the human hand during grasping is dominated by movement in a configuration space of highly reduced dimensionality. Such configuration space is known as postural synergies and its main components are referred as eigengrasps. In order to explore this approach, we decided to use the two dominant human eigengrasps in order to control one of the gripper obtained using our design algorithm. For this manipulator we defined the eigengrasps drawing inspiration from the human hand and we also performed some simple grasps.

This work is organized as follows. In Chapter 2 a brief description of modular robots with greater emphasis on the previous research on the modular gripper design problem, is given. In Chapter 3 an introduction to the grasping problem is given, describing in particular fundamental models of grasp analysis. Moreover the concept of safe grasping is investigated and a possible controller inspired by human method of grasping is discussed [8].In Chapter

4 the innovation brought out by main contribution of this work is presented. Our object-oriented algorithm is introduced. It allows to determine effective manipulator configurations to obtain stable grasps of given objects. In Chapter 5 we expose simulations and results obtained using our iterative algorithm. In Chapter 6 we investigate the possibility of using the two dominant human eigengrasps in order to control the modular manipulator. In Chapter 7, conclusion and future work are outlined.

As last, in Appendix, source codes and a paper submitted to the IEEE/RSJ International Conference on Intelligent Robots and Systems ("IROS 2011") are shown.

Chapter 2

Modular Robots and Grippers

The field of modular self-reconfigurable robotic systems addresses the design, fabrication, motion planning, and control of autonomous kinematic machines with variable morphology. Over the last two decades, the field of modular robotics has advanced from proof-of-concept systems to elaborate physical implementations and simulations. However, only few works have been done on the possibility of developing a modular gripper.

The goal of this chapter is to outline some of this progress with greater emphasis on those relating to the modular gripper design.

2.1 Definition

A modular robotic system consists of several modular links and modular joints with standardized connecting interfaces. It can be easily assembled and disassembled, so it can accomplish a larger number of classes of tasks through the reconfiguration of a small inventory of modules [9].

Modular robots are usually composed [10] of multiple building blocks of a relatively small repertoire, with uniform docking interfaces that allow for the transfer of mechanical forces and moments, electrical power and communication throughout the robot. The modular building blocks usually consist of some primary structural actuated units, and potentially additional specialized units such as grippers, feet, wheels, cameras, payload and energy storage and generation.

2.2 Motivation and inspiration

There are many advantages and motivations in designing and developing modular robotic systems.

- Functional advantage: Self reconfiguring robotic systems are potentially more robust and more adaptive than conventional systems. The reconfiguration ability allows a robot or a group of robots to disassemble and reassemble machines to form new morphologies that are better suitable for new tasks, such as changing from a legged robot to a snake robot and then to a rolling robot. Since robot parts are interchangeable (within a robot and between different robots), machines can also replace faulty parts autonomously, leading to self-repair.

- Economic advantage: Self reconfiguring robotic systems can potentially lower overall robot cost by making a range of complex machines out of a single (or relatively few) types of mass-produced modules.

  The quest for self-reconfiguring robotic structures is to some extent inspired by envisioned applications such as long-term space missions, that require long-term self-sustaining robotic ecology that can handle unforeseen situations and may require self repair. A second source of inspiration are biological systems that are self-constructed out of a relatively small repertoire of lower-level building blocks (cells or amino acids, depending on scale of interest). This architecture underlies the ability to physically adapt, grow, heal, and even self replicate capabilities that would be desirable in many engineered systems.

2.3 Previous Research on Modular Robots

2.3.1 Modular Robots

Modularity in robotics is a quite new concept. To the best of our knowledge, few works have been done on the possibility of developing a modular gripper. Some initial studies are related to the self-reconfigurable robots literature [11]: a cellular robot capable of adapting its shape and functions to changing environments and demands by rearranging its mutual mechanical connection. In [12], the authors proposed an algorithm for grasping objects with a self-reconfigurable system. Although the idea that a modular gripper can handle objects of unknown shape and size was pointed out, the work reported preliminary results still far away from their real implementation.

In [13] Yu and Nagpal presented a generalized distributed consensus framework for self-adaptation tasks in modular robotics. They demonstrated that a variety of modular robotic systems and tasks can be formulated within such a framework. An adaptive column that can respond to external forces, a modular tetrahedral robot that can move towards a light source, and a modular gripper that can wrap around fragile objects have been presented. The decentralized control used is based on the sharing of information about pressure given by sensors included in each module. This solution is not applicable when, for instance, a fingertip manipulation is required. Furthermore investigation on grasp stability is missed by the authors.

As well as the self-reconfigurable framework, the creation of modular grippers has been used in rapid prototyping and part holding problems. In [14] the design and techniques for part insertion into a non-assembly, multi-articulated, dexterous finger prototype built with stereo-lithography are presented. This solution makes possible a rapid development of robotic systems that have all the necessary components inserted, with no assembly required, and ready to work when the manufacturing process is complete. However, the actuation and control of such a device is still an open problem. Alternatively, Brown and Brost presented in [15], a modular vise which is a parallel-jaw vise, with a regular grid of precisely positioned holes on each jaw. Parts are held by placing pins in the holes so that when the vise is closed, the parts are reliably located and completely constrained. Even though the modular vise concept can be adapted to the design of modular parallel-jaw grippers for robots, no application to dexterous manipulation can be applied.

A general classification [16] of the different configurations of modular robots is essential for the study of their properties, but is difficult to realize because of the exponential growth of robot type and configuration of the modules.

Firstly we group the modular robots according to their locomotion capability. The first group features active modules which comprise several identical modules that can move autonomously. Each one is an entire robot system that can perform distributed activities. Meanwhile, all of them can interconnect by specially designed docking joints which enable the adjacent modules to adopt optimized configurations to negotiate difficult terrain or split into several small units to perform tasks simultaneously.

A sub-classification of this kind of modular robots according to their kinematics modes includes wheeled and chain-track vehicles; those are relatively portable thank to their high adaptability to unstructured environments. The application of powered tracks to field robots enhances their configurations and improves their adaptability to environments.

Figure 2.1: The serpentine robot Souryu-II

A serpentine robot (Fig.2.1) from Takayama and Hirose consists of three segments; each one is driven by a pair of tracks, but all tracks are powered simultaneously by a single motor located in the center segment. The special ability of adapting to irregular terrain is passive and provided by springs. The OmniTread [17](Fig.2.2) serpentine robot for industrial inspection and surveillance was developed by Grzegorz Granosik. Optimal active joints actuated by pneumatic cylinder make the robot strong and flexible. Even if this prototype is quite adaptable, it is obstructed by a tube for pressurized air and cables for control signals connected to the supporting system.

Figure 2.2: The serpentine robot OmniTread

Another robot KOHGA [18] (Fig.2.3) has recently been developed for rescue purposes by I.R.S.I. (International Rescue System Institute) in Japan; it consists of serially interconnected individual units with two tracks except the first and the last modules.

Figure 2.3: The robot KOHGA

Another group of reconfigurable modular robots features passive modules. The robotic system comprises several identical modules without any locomotion capability. It can only move after the modules have been assembled. New configurations can be achieved very fast and easily using this kind of mechanical modules. The robot used in this work belongs to this group. Some researchers established a classification of this kind of modular robots: lattice and chain robot. The former type arranges modules to form a grid, just like atoms forming complex 3D molecules or solids. One of the promise of this kind of robots is the possibility of building solid objects, such as cups of chairs, and then rearranging the atoms to form other solids. The latter structures are composed of chains of modules.

Figure 2.4: An example of chain-robot

Chain-robots (Fig.2.4) are suitable for locomotion and manipulation since the modular chains are like legs or arms. It is possible to classified 1D, 2D and 3D chain robots, according to their topology. 1D-chain robots are like snakes, worms, legs, arms or cords. They can change their body to adopt different shapes; moreover they are suitable for passing through tubes, grasping object and moving in rough terrain. However, the pitch connecting robots can only move forward and backward. Their movements can be generated by means of waves that travel through the body of the robot from the tail to the head. M-TRAN and Yamour are similar prototypes that can only connect in a pitch-pitch way.

Another part of chain-module robot features yaw-connections. All the joints rotate around the yaw axes. In order to get propelled, these robot creep along a given curve path, but the body should slip in the tangential direction without any sliding in the direction normal to the body axis. A lot of research has been done on this kind of robots. Yaw-connecting robots were first studied by Hirose [19] who developed the Active Cord Mechanism (ACM). Other groups have developed yaw-connecting robot; we cite the SES-2 [20], WormBot [21] and the swimming Amphibot II [22] (Fig.2.5).

Figure 2.5: The swimming robot Amphibot-II

2.3.2 TAMS Group and SIRSLab: Works on Modular Robots

In 2004, our international group, composed by TAMS (Technical Aspects of Multimodal Systems, University of Hamburg) and SIRSLab (Siena Robotics and Systems Lab), began to work on low-cost passive modular robots. The Y1 modular robot with one DOF was designed as the first prototype. Based on the Y1, the following project of 2006 was aimed at developing a real low-cost, robust, fast-prototyping modular robot with an on board controller, sensors and a friendly, easy-to-use programming environment for testing and evaluating inspired technology [23].

We then focused our efforts on using modular robots to perform object manipulation. We started by considering a simple hyper-redundant manipulator. Consequently, we investigated the possibility of developing a novel modular grasping approach that combines manipulation capability and locomotion mobility to implement possible tasks. We proposed a snake-like configuration and we introduced a task priority approach to manage both grasping and locomotion capabilities [24]. Even if we outlined several robot configurations that lead toward stable grasping, the characteristics of snake-like robots are more suitable for Search and Rescue missions than for the manipulation of objects in industrial scenarios.

2.3.3 Modular Robot Design

There are many studies on kinematics, dynamic, and control design of modular robots.

Paredis and Khosla [25, 26] described a general flow chart of a selection program for modular robots, which covers three phases: kinematics, dynamics, and sensor-based control. Furthermore, in their scheme, kinematics design played a dominant role in the sense that the other two phases contribute to modification of the result derived from the kinematics design. Such a design procedure is called a sequential design procedure. Matsumaru [27] presented another kind of sequential design procedure for his reconfigurable modular manipulator system. Cohen [28] and Hooper and Tesar [29] also used this method for synthesizing a modular robotic configuration. Fryer, McKee, and Schenker [30] proposed that an object-oriented concept could provide a useful tool for verifying the configuration of modular robotics systems, and they modeled robot resources and considered how the models could be adapted through the introduction of semantic annotations to accommodate the configuration process. All these works try to achieve the system flexibility by adjusting parameters in one module or selecting an alternative module for a different parameter.

Chen and Burdick [31] presented an approach to task optimal modular robot assembly configuration. Philosophically, their work was built upon the concept of traditional mechanism type synthesis, where the joints actually make no physical sense, so their work try to achieve the system flexibility by varying different configurations where the selection of types of modules and their connections is of a concern.

Fukuda and Kawauchi [32] proposed a synthesis approach for their CEBOT systems. Their approach assumed that each module had the same length and they considered only the reachability of working points with a robot hand as a performance index. Chen and Burdick [33] Han et al., [34] and Chocron and Bidaud [35] employed genetic algorithms (GA) for task-oriented design of modular robots, but considered only kinematics synthesis.

Bi and Zhang presented an optimization design methodology called concurrent optimization design method (CODM) [9]. A modular robot is taken as a case study. The CODM differs from the precedent methods for modular robot configuration design in the sense that traditional type synthesis and dimensional synthesis can be trated once. This mathematically implies that (i) variables are defined for both types and dimensions, and (ii) all the variables are defined in one optimization problem formulation. A genetic algorithm is used to solve for this complex optimization model which contains both discrete and continuous variables.

In short, previous works in this research field implied that synthesis for robots with nonmodular architecture does not differ from synthesis for robots with modular architecture. The approach proposed in our work goes away from this thought. It is an iterative procedure that is based on the flexibility offered by modular robotics.

Our algorithm involves all the three levels of modular robot architecture:

- Module-level;
- Assembly-level;
- Configuration-level.

The algorithm is outlined in detail in Chapter 4.

Chapter 3

Grasping

This chapter introduces fundamental models of grasp analysis. The overall model is a coupling of models that define contact behavior with widely used models of rigid body kinematics and dynamics. The contact model essentially boils down to the selection of components of contact forces and moments that are transmitted through each contact.

3.1 Model and Definition

A mathematical model of grasping [36] must be capable of predicting the behavior of the hand and object under various loading conditions that may arise during grasping. Generally, the most desirable behavior is grasp maintenance in the face of unknown disturbing forces and moments applied to the object. Typically these disturbances arise from inertia force which become appreciable during high-speed manipulation or applied forces such as those due to gravity. Grasp maintenance means that the contact forces applied by the hand are such that they prevent contact separation and unwanted contact sliding. The special class of grasps that can be maintained for every possible disturbing load is known as closure grasps. Fig.3.1 shows the Salisbury Hand [37], executing a closure grasp of an object by wrapping its fingers around it and pressing it against its palm.

Figure 3.1: The Salisbury hand grasping an object

Fig.3.2 illustrates some of the main quantities that will be used to model grasping systems. Assume that the links of the hand and of the object are rigid and that there is a unique, well-defined tangent plane at each contact point. Let $\{\mathbf{N}\}$ represent a conveniently chosen inertial frame fixed in the workspace. The frame $\{\mathbf{B}\}$ is fixed to the object with its origin defined relative to $\{\mathbf{N}\}$ by the vector $\mathbf{p} \in \Re^3$ where $\Re^3$ denotes three-dimensional Euclidean space. A convenient choice for $\mathbf{p}$ is the center of mass of the object. The position of the contact point i in $\{\mathbf{N}\}$ is defined by the vector $c_i \in \Re^3$. At contact point i, we define a frame $\{\mathbf{C}\}_i$, with axes $\{\hat{\mathbf{n}}_i, \hat{\mathbf{t}}_i, \hat{\mathbf{o}}_i\}$. The unit vector $\hat{\mathbf{n}}_i$ contains c_i is normal to the contact tangent plane and is directed toward the object. The other two unit vectors are orthogonal and lie in the tangent plane of the contact.

Let the joints be numbered from 1 to n_q. Denote by $\mathbf{q} = [q_1 ... q_{n_q}]^T \in \Re^{n_q}$ the vector of joint displacement, where the superscript ^{T} indicates matrix transposition. Also, let $\boldsymbol{\tau} = [\tau_1 ... \tau_{n_q}]^T \in \Re^{n_q}$ represent joint loads (forces in prismatic joints and torque in revolute joints). These loads can result from actuator actions, other applied forces, and inertia forces. They could also arise from contacts between the object and hand. However, it will be convenient to separate joint loads into two components: those arising form contacts and those arising from all other sources.

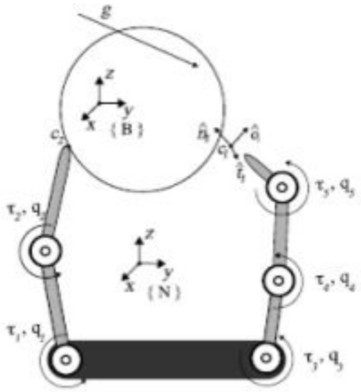


Figure 3.2: Main quantities for grasp analysis

Throughout this chapter, non-contact loads will be denoted by τ.

Let $\mathbf{u} \in \Re^{n_u}$ denote the vector describing the position and orientation of $\{\mathbf{B}\}$ relative to $\{\mathbf{N}\}$. For planar systems $n_u = 3$. For spatial systems, n_u is three plus the number of parameter used to represent orientation, typically three for Euler angle and four for quaternion. Denote by $\boldsymbol{\nu} = [\mathbf{v}^T \boldsymbol{\omega}^T]^T \in \Re^{n_\nu}$ the twist of object described in $\{\mathbf{N}\}$. It is composed of the translational velocity $\boldsymbol{\nu} \in \Re^3$ of the point $\mathbf{p}$ and the angular velocity $\boldsymbol{\omega} \in \Re^3$ of the object, both expressed in $\{\mathbf{N}\}$. A twist of a rigid body can be referred to any convenient frame fixed to the body. The components of the referred twist represent the velocity of the origin of the new frame and the angular velocity of the body, both expressed in the new frame. Note that for planar systems, $\boldsymbol{\nu} \in \Re^2$ and $\omega \in \Re$, and so $n_u = 3$.

Another important point in that $\dot{\mathbf{u}} \neq \boldsymbol{\nu}$. Instead, these variables are related by the matrix $\mathbf{V}$ as:

$$\dot{\mathbf{u}} = \mathbf{V}\boldsymbol{\nu}, \tag{3.1}$$

where the matrix $\mathbf{V} \in \Re^{n_u \mathrm{X} n_v}$ is not generally square but nonetheless satisfies $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ [38], $\mathbf{I}$ is the identity matrix, and the dot over the u implies differentiation with respect to time. Note that, for planar systems, $\mathbf{V} = \mathbf{I} \in \Re^{3\times3}$.

Let $\mathbf{f} \in \Re^3$ be the force applied to the object at the point $\boldsymbol{p}$ and let $\mathbf{m} \in \Re^3$ be the applied moment. These are combined into the object load, or wrench, vector denoted by $\mathbf{g} = [\mathbf{f}^T \mathbf{m}^T]^T \in \Re^{n_\nu}$, where $\boldsymbol{f}$and $\boldsymbol{m}$ are expressed in $\{\mathbf{N}\}$. Like twists, wrenches can be referred to any convenient frame fixed to the body. One can think of this as translating the line of application of the force until it contains the origin of the new frame, then adjusting the moment component of the wrench to offset the moment induced by moving the line of the force. Last, the force and adjusted moment are expressed in the new frame. As done with the joint loads, the object wrench will be partitioned into two main parts: contact and non-contact wrenches. Throughout this chapter, $\boldsymbol{g}$ will denote the non-contact wrenches on the object.

3.2 Velocity Kinematics

The material in this chapter is valid for a wide range of robot hands and other grasping mechanisms like our modular robot. The hand is assumed to be composed of a palm that serves as the common base for any number of fingers, each with any number of joints. Any number of contacts may occur between any link and the object.

3.2.1 Grasp Matrix and Hand Jacobian

Two matrices are of the utmost importance in grasp analysis: the *grasp matrix* $\boldsymbol{G}$ and the *hand Jacobian* $\boldsymbol{J}$. These matrices define the relevant velocity kinematics and force transmission properties of the contacts. The following derivations of $\boldsymbol{G}$ and $\boldsymbol{J}$ will be done under the assumption that the system is three-dimensional.

Each contact should be considered as two coincident points; one on the hand and one on the object. The Hand Jacobian maps the joint velocities to the twists of the hand to the contact frames, while the transpose of the Grasp Matrix refers the object twist to the contact frames. Finger joint motions induce a rigid-body motion in each link of the hand. Thus these matrices can be derived from the transforms that change the reference frame of a twist.

To derive the grasp matrix, let $\boldsymbol{\omega}_{obj}^{N}$ denote the angular velocity of the object expressed in $\{\mathbf{N}\}$ and let $\boldsymbol{\nu}_{i,obj}^{N}$, also expressed in $\{\mathbf{N}\}$, denote the velocity of the point on the object coincident with the origin of $\{\mathbf{C}_i\}$. These velocities can be obtained from the object twist referred to $\{\mathbf{N}\}$ as:

$$\begin{pmatrix} \boldsymbol{\nu}_{i,obj}^N \\ \boldsymbol{\omega}_{obj}^N \end{pmatrix} = \boldsymbol{P}_i^T \nu \tag{3.2}$$

where

$$\boldsymbol{P}_i = \begin{pmatrix} \boldsymbol{I}_{3\times3} & 0 \\ \boldsymbol{S}(\boldsymbol{c}_i - \boldsymbol{p}) & \boldsymbol{I}_{3\times3} \end{pmatrix} \tag{3.3}$$

$\boldsymbol{I}_{3\times3} \in \Re^{3\times3}$ is the identity matrix, and $\boldsymbol{S}(\boldsymbol{c}_i - \boldsymbol{p})$ is the cross-product matrix, that is, given a three vector $\boldsymbol{r} = [r_x r_y r_z]^T$, $\boldsymbol{S}(\boldsymbol{r})$is defined as:

$$\boldsymbol{S}(\boldsymbol{r}) = \begin{pmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{pmatrix}$$

The object twist referred to $\{\boldsymbol{C}\}_i$ is simply the vector on the left-hand side of (3.2) expressed in $\{\boldsymbol{C}\}_i$. Let $\boldsymbol{R}_i = [\hat{n}_i \hat{t}_i \hat{o}_i] \in \Re^{3\times3}$ represent the orientation of the i-th contact frame $\{\boldsymbol{C}\}_i$ with respect to the inertial frame. Then the object twist referred to $\{\boldsymbol{C}\}_i$ is given as:

$$\boldsymbol{\nu}_{i,obj} = \bar{\boldsymbol{R}}_i^T \begin{pmatrix} \boldsymbol{\nu}_{i,obj}^N \\ \boldsymbol{\omega}_{obj}^N \end{pmatrix} \tag{3.4}$$

where $\bar{\boldsymbol{R}}_i$ =Blockdiag$(\boldsymbol{R}_i, \boldsymbol{R}_i) = \begin{pmatrix} \boldsymbol{R}_i & 0 \\ 0 & \boldsymbol{R}_i \end{pmatrix} \in \Re^{6\times6}$

Substituting $\boldsymbol{P}_i^T \nu$ from (3.2) yields the partial grasp matrix $\tilde{\boldsymbol{G}}_i^T \in \Re^{6\times6}$, which maps the object twist from $\{\boldsymbol{N}\}$ to $\{\boldsymbol{C}\}_i$:

$$\boldsymbol{\nu}_{i,obj} = \tilde{\boldsymbol{G}}_i^T \nu \tag{3.5}$$

where

$$\tilde{\boldsymbol{G}}_i^T = \bar{\boldsymbol{R}}_i^T \boldsymbol{P}_i^T \tag{3.6}$$

The hand Jacobian can be derived similarly. Let $\boldsymbol{\omega}^N_{i,hnd}$ be the angular velocity of the link of the hand touching the object at contact i, expressed in $\{\boldsymbol{N}\}$ and define $\boldsymbol{\nu}^N_{i,hnd}$ as the translational velocity of contact i on the hand, expressed in $\{\boldsymbol{N}\}$. These velocities are related to the joint velocities through the matrix $\boldsymbol{Z}_i$ whose columns are the Plücker coordinates of the axes of the joints [39]. We have:

$$\begin{pmatrix} \boldsymbol{\nu}^N_{i,hnd} \\ \boldsymbol{\omega}^N_{i,hnd} \end{pmatrix} = \boldsymbol{Z}_i \dot{\boldsymbol{q}} \tag{3.7}$$

where $\boldsymbol{Z}_i \in \Re^{6\times n_q}$ is defined as:

$$\boldsymbol{Z}_i = \begin{pmatrix} \boldsymbol{d}_{i,1} & ... & \boldsymbol{d}_{i,n_q} \\ \boldsymbol{l}_{i,1} & ... & \boldsymbol{l}_{i,n_q} \end{pmatrix} \tag{3.8}$$

with the vectors $\boldsymbol{l}_{i,j}, \boldsymbol{d}_{i,j} \in \Re^3$ defined as:

$$\boldsymbol{d}_{i,j} = \begin{cases} \boldsymbol{0}_{3x1} \text{ if contact } i \text{ does not affect joint } j, \\ \hat{\boldsymbol{z}}_i \text{ if joint } j \text{ is prismatic}, \\ \boldsymbol{S}(\boldsymbol{c}_i - \boldsymbol{\zeta}_j)^T \hat{\boldsymbol{z}}_j \text{ if } j \text{ is revolute} \end{cases}$$

$$\boldsymbol{l}_{i,j} = \begin{cases} \boldsymbol{0}_{3x1} \text{ if contact } i \text{ does not affect joint } j, \\ \boldsymbol{0}_{3x1} \text{ if joint } j \text{ is prismatic}, \\ \hat{\boldsymbol{z}}_j \text{ if } j \text{ is revolute} \end{cases}$$

where $\boldsymbol{\zeta}_j$ is the origin of the coordinate frame associated with the j-th joint and the $\hat{\boldsymbol{z}}_j$ is the unit vector in the direction of the z-axis in the same frame, as shown in Fig.3.3.

Both vectors are expressed in $\{\boldsymbol{N}\}$. These frames may be assigned by any convenient method, for example, the Denavit-Hartenberg method. The $\hat{\boldsymbol{z}}_j$-axis is the rotational axis for revolute joints and the direction of translational for prismatic joints.

The final step in referring the twist to the contact frames is to change the frame of expression of $\boldsymbol{\nu}^N_{i,hnd}$ and $\boldsymbol{\omega}^N_{i,hnd}$ to $\{\boldsymbol{C}\}_i$:

$$\boldsymbol{\nu}_{i,hnd} = \bar{\boldsymbol{R}}_i \begin{pmatrix} \boldsymbol{\nu}^N_{i,hnd} \\ \boldsymbol{\omega}^N_{i,hnd} \end{pmatrix} \tag{3.9}$$

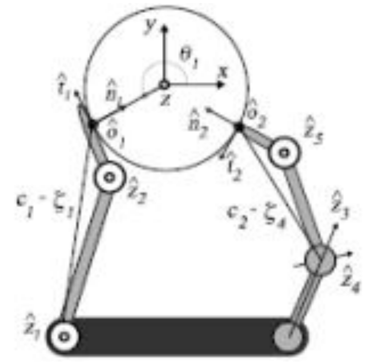


Figure 3.3: A sphere grasped by a two-fingers hand with 5 revolute joints

Combining (3.9) and (3.7) yields the partial hand Jacobian $\tilde{\boldsymbol{J}}_i \in \Re^{6\times n_q}$, which relates the joint velocities to the contact twists on the hand:

$$\boldsymbol{\nu}_{i,hnd} = \tilde{\boldsymbol{J}}_i \dot{\boldsymbol{q}} \tag{3.10}$$

where

$$\tilde{\boldsymbol{J}}_i = \bar{\boldsymbol{R}}_i \boldsymbol{Z}_i \tag{3.11}$$

To compact notation, stack all the twists of the hand and object into the vectors $\boldsymbol{\nu}_{c,hnd} \in \Re^{6n_c}$ and $\boldsymbol{\nu}_{c,obj} \in \Re^{6n_c}$ as follows:

$$\boldsymbol{\nu}_{c,\xi} = (\boldsymbol{\nu}_{1,\xi}^T ... \boldsymbol{\nu}_{n_c,\xi}^T), \xi = \{obj, hnd\}$$

Now the *complete*[1] *grasp matrix* $\tilde{\boldsymbol{G}} \in \Re^{6\times 6n_c}$ and the *complete hand Jacobian* $\tilde{\boldsymbol{J}} \in \Re^{6n_c\times 6n_q}$ relate the various velocity quantities as

$$\boldsymbol{\nu}_{i,obj} = \tilde{\boldsymbol{G}}^T \nu \tag{3.12}$$

$$\boldsymbol{\nu}_{i,hnd} = \tilde{\boldsymbol{J}} \dot{\boldsymbol{q}} \tag{3.13}$$

[1]The term complete is used to emphasize that all the $6n_c$ twist component at the contacts are included in the mapping.

where

$$\tilde{\boldsymbol{G}}^T = \begin{pmatrix} \tilde{\boldsymbol{G}}_1^T \\ \vdots \\ \tilde{\boldsymbol{G}}_{n_c}^T \end{pmatrix}, \tilde{\boldsymbol{J}} = \begin{pmatrix} \tilde{\boldsymbol{J}}_1 \\ \vdots \\ \tilde{\boldsymbol{J}}_{n_c} \end{pmatrix} \tag{3.14}$$

3.2.2 Contact Modeling

The three models of greatest interest in grasp analysis are known as *point-contact-without-friction*, *hard-finger* and *soft-finger* [37]. These models select components of the contact twists to transmit between the hand and the object. This is done by equating a subset of the components of the hand and object twist at each contact. The corresponding components of the contact force and moment are also equated, but without regard for the constraints imposed by a friction model.

The point-contact-without-friction (PwoF) model is used when the contact patch is very small and the surfaces of the hand and object are slippery. With this model, only the normal component of the translational velocity of the contact point on the hand (i.e., the first component of $\boldsymbol{\nu}_{i,hnd}$) is transmitted to the object. The two components of tangential velocity and the three components of angular velocity are not transmitted. Analogously, the normal component of the contact force is transmitted, but the frictional forces and moments are assumed to be negligible.

A hard finger (HF) model is used when there is significant contact friction, but the contact patch is small, so that no appreciable friction moment exists. When this model is applied to a contact, all three translational velocity components of the contact point on the hand (i.e., the first three components of $\boldsymbol{\nu}_{i,hnd}$) and all three components of the contact force are transmitted through the contact. None of the angular velocity components or moment components are transmitted.

The soft finger (SF) model is used in situations in which the surface friction and the contact patch are large enough to generate significant friction forces and a friction moment about the contact normal. At a contact where this model is enforced, the three translational velocity components of the contact on the hand and the angular velocity component about the contact normal are transmitted (i.e., the first four components of $\boldsymbol{\nu}_{i,hnd}$). Similarly, all three components of contact force and the normal component of the contact moment are transmitted.

To develop the PwoF, HF, and SF models, define the relative twist at contact i as:

$$(\tilde{\boldsymbol{J}}_i \quad \tilde{\boldsymbol{G}}_i^T)\begin{pmatrix}\dot{\boldsymbol{q}}\\ \boldsymbol{\nu}\end{pmatrix} = \boldsymbol{\nu}_{i,hnd} - \boldsymbol{\nu}_{i,obj}.$$

A particular contact model is defined through the matrix $\boldsymbol{H}_i \in \Re^{l_i \times 6}$, which select l_i components of the relative contact twist and sets them to zero:

$$\boldsymbol{H}_i(\boldsymbol{\nu}_{i,hnd} - \boldsymbol{\nu}_{i,obj}) = 0.$$

These components are referred to as transmitted degrees of freedom (DOF). Define $\boldsymbol{H}_i$ as:

$$\boldsymbol{H}_i = \begin{bmatrix}\boldsymbol{H}_{iF} & 0\\ 0 & \boldsymbol{H}_{iM}\end{bmatrix} \tag{3.15}$$

where $\boldsymbol{H}_{iF}$ and $\boldsymbol{H}_{iM}$ are the translational and rotational component selection matrices, respectively. Table 3.1 gives the definition of the selection matrices for

Model	l_i	$\boldsymbol{H}_{i,F}$	$\boldsymbol{H}_{i,M}$
PwoF	1	(1 0 0)	vacuous
HF	3	$I_{3\times3}$	vacuous
SF	4	$I_{3\times3}$	(1 0 0)

Table 3.1: Selection matrices for three contact models

the three contact models, where *vacuous* means that the corresponding block row matrix in (3.15) is void (i.e. it has zero rows and columns). Notice that, for the SF model, $\boldsymbol{H}_{i,M}$ selects rotation about the contact normal.

After choosing a transmission model for each contact, the contact constraint equations for all n_c contacts can be written in compact form as:

$$\boldsymbol{H}(\boldsymbol{\nu}_{i,hnd} - \boldsymbol{\nu}_{i,obj}) = 0, \tag{3.16}$$

where

$$\boldsymbol{H} = \text{Blockdiagonal}(\boldsymbol{H}_i, ..., \boldsymbol{H}_{n_c}) \in \Re^{l\times 6n_c},$$

and the number of twist components l transmitted through the contacts is given by $l = \sum_{i=1}^{n_c} l_i$.

Finally, by substituting (3.12) and (3.13) into (3.16) one obtains:

$$(\tilde{\boldsymbol{J}}_i \quad \tilde{\boldsymbol{G}}_i^T) \begin{pmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{\nu} \end{pmatrix} = 0, \tag{3.17}$$

where the grasp matrix and the hand Jacobian are

$$\begin{aligned} \boldsymbol{G}^T &= \boldsymbol{H}\tilde{\boldsymbol{G}}^T \in \Re^{l\times 6}, \\ \boldsymbol{J} &= \boldsymbol{H}\tilde{\boldsymbol{J}} \in \Re^{l\times n_q} \end{aligned} \tag{3.18}$$

It is worth noting that (3.17) can be written in the following form:

$$\boldsymbol{J}\dot{\boldsymbol{q}} = \boldsymbol{\nu}_{cc,hnd} = \boldsymbol{\nu}_{cc,obj} = \mathbf{G}^T\boldsymbol{\nu}, \tag{3.19}$$

where $\boldsymbol{\nu}_{cc,hnd}$ and $\boldsymbol{\nu}_{cc,obj}$ contain only the components of the twists that are transmitted by the contacts. Note that this equation implies that grasp *maintenance* is defined as the situation in which all these equations are maintained over time. Thus, when a contact is friction-less, contact maintenance implies continued contact, but sliding is allowed. However, when a contact is of the type HF, contact maintenance implies sticking contact, since sliding would violate the HF model. Similarly, for a SF contact, there may be no sliding or relative rotation about the contact normal.

3.3 Dynamics and Equilibrium

Dynamic equations of the system can be written as:

$$\begin{aligned} \boldsymbol{M}_{hnd}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{b}_{hnd}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{J}^T\boldsymbol{\lambda} &= \boldsymbol{\tau}_{app} \\ \boldsymbol{M}_{obj}(\boldsymbol{u})\dot{\boldsymbol{\nu}} + \boldsymbol{b}_{obj}(\boldsymbol{u}, \dot{\boldsymbol{\nu}}) - \boldsymbol{G}\boldsymbol{\lambda} &= \boldsymbol{g}_{app} \\ &\text{subject to constraint (3.17)} \end{aligned} \tag{3.20}$$

where $\boldsymbol{M}_{hnd}(\cdot)$ and $\boldsymbol{M}_{obj}(\cdot)$ are symmetric positive definite inertia matrices and $\boldsymbol{b}_{hnd}(\cdot,\cdot)$ and $\boldsymbol{b}_{obj}(\cdot,\cdot)$ are the velocity product terms, $\boldsymbol{g}_{app}$ is the force and moment applied to the object by gravity and the other external source, $\boldsymbol{\tau}_{app}$ is the vector of external loads and actuator actions, and the vector $\boldsymbol{G\lambda}$ is the total wrench applied to the object by the hand. The vector λ contains the contact force and moment components transmitted through the contacts and expressed in the contact frames. Specifically $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1^T \cdots \boldsymbol{\lambda}_{nc}^T]^T$, where $\boldsymbol{\lambda}_i = \boldsymbol{H}_i[f_{in} f_{it} f_{io} m_{in} m_{it} m_{io}]^T$. The subscript indicate the normal (n) and the two tangent (t,o) components of the contact force $\boldsymbol{f}$ and the moment $\boldsymbol{m}$. For an SF, HF, or PwoF contact, $\boldsymbol{\lambda}_i$ is defined as in Table 3.2. Finally, it is worth noting that $\boldsymbol{G}_i\boldsymbol{\lambda}_i = \tilde{\boldsymbol{G}}_i\boldsymbol{H}_i\boldsymbol{\lambda}_i$ is the wrench applied through contact i, where $\tilde{\boldsymbol{G}}_i$ and $\boldsymbol{H}_i$ are defined in (3.6) and (3.15). The vector $\boldsymbol{\lambda}_i$ is known as the wrench intensity vector for contact i.

Model	$\boldsymbol{\lambda}_i$
PwoF	(f_{in})
HF	$(f_{in} f_{it} f_{io})^T$
SF	$(f_{in} f_{it} f_{io} m_{in})^T$

Table 3.2: Wrench intensity vector, transmitted through contact i

Equation (3.20) represent the dynamics of the hand and object without regard for the kinematics constraints imposed by the contact models. Enforcing them, the dynamic model of the system can be written as:

$$\begin{pmatrix} \boldsymbol{J}^T \\ -\boldsymbol{G} \end{pmatrix} \boldsymbol{\lambda} = \begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{g} \end{pmatrix} \tag{3.21}$$

subject to $\boldsymbol{J}\dot{\boldsymbol{q}} = \boldsymbol{G}^T\boldsymbol{\nu} = \boldsymbol{\nu}_{cc}$, where

$$\begin{aligned} \boldsymbol{\tau} &= \boldsymbol{\tau}_{app} - \boldsymbol{M}_{hnd}(\boldsymbol{q})\ddot{\boldsymbol{q}} - \boldsymbol{b}_{hnd}(\boldsymbol{q},\dot{\boldsymbol{q}}) \\ \boldsymbol{g} &= \boldsymbol{g}_{app} - \boldsymbol{M}_{obj}(\boldsymbol{u})\dot{\boldsymbol{\nu}} + \boldsymbol{b}_{obj}(\boldsymbol{u},\dot{\boldsymbol{\nu}}) \end{aligned} \tag{3.22}$$

One should notice that the dynamic equations are closely related to the kinematics model in (3.17). Specifically, just as $\boldsymbol{J}$ and $\boldsymbol{G}^T$ transmit only selected components of twists, $\boldsymbol{J}^T$ and $\boldsymbol{G}$ in (3.20) serve to transmit only the corresponding components of the contact wrenches.

When the inertia terms are negligible, as occurs during slow motion, the system is said to be quasistatic. In this case, (3.22) becomes:

$$\begin{aligned} \boldsymbol{\tau} &= \boldsymbol{\tau}_{app} \\ \boldsymbol{g} &= \boldsymbol{g}_{app} \end{aligned} \tag{3.23}$$

and does not depend on joint and object velocities. Consequently when the grasp is in a static equilibrium or moves quasistatically, one can solve the first equation and constraint in (3.21) independently to compute $\boldsymbol{\lambda}$,$\dot{\boldsymbol{q}}$ and $\boldsymbol{\nu}$. It is worth noting that such a force(velocity) decoupled solution is not appreciable, since the first equation in (3.21) depends on the third one through (3.22).

Remark *Equation (3.21) highlights an important alternative view of the Grasp Matrix and the Hand Jacobian.* $\mathbf{G}$ *can be thought of as a mapping from the transmitted contact forces and moments to the set wrenches that the hand can apply to the object, while* $\mathbf{J}^T$ *can be thought of as a mapping from the transmitted contact forces and moments to the vector of joint loads. Notice that these interpretations hold for both dynamic and quasistatic conditions.*

3.4 Restraint Analysis

The most fundamental requirements in grasping and dexterous manipulation are the abilities to hold an object in equilibrium and control the position and orientation of the grasped object relative to the palm of the hand. The two most useful characterizations of grasp restraint are *force closure* and *form closure*. These names were in use over one hundred twenty five years ago in the field of machine design to distinguish between joints that required an external force to maintain contact, and those that did not. For example, some water wheels had a cylindrical axle that was laid in a horizontal semi-cylindrical groove split on either side of the wheel. During operation, the weight of the wheel acted to "close" the groove-axle contacts, hence the term *force closure*. By contrast, if the grooves were replaced by cylindrical holes just long enough to accept the axle, then the contacts would be closed by the geometry (even if the direction of the gravitational force were reversed), hence the term *form closure*. When applied to grasping, form and force closure have the following interpretations. Assume that a hand grasping an object has its joint angles locked and its palm fixed in space. Then the grasp has *form closure*, or the object is *form closed*, if it is impossible to move the object, even infinitesimally. Under the same conditions, the grasp has *force closure*, or the object is *force closed*, if for any non-contact wrench experienced by the object, contact wrench intensities exist that satisfy equations (3.20) and are consistent with the constraints imposed by the friction models applicable at the contact points. Notice that all form closure grasps are also force closure grasps. When under form closure, the object cannot move at all, regardless of the non-contact wrench. Therefore, the hand maintains the object in equilibrium for any external wrench, which is the force closure requirement.

Roughly speaking, form closure occurs when the palm and fingers wrap around the object forming a cage with no "wiggle" room such as the grasp shown in Fig.3.4. This kind of grasp is also called a power grasp or enveloping grasp. However, force

Figure 3.4: The palm, fingers, wrist, and watch band combine to create a very secure form closure grasp

closure is possible with fewer contacts, as shown in Fig.3.5, but in this case force closure requires the ability to control internal forces. It is also possible for a grasp

Figure 3.5: This grasp has a force closure grasp appropriate for dexterous manipulation

to have partial form closure, indicating that only a subset of the possible degrees of freedom are restrained by form closure. An example of such a grasp is shown in Fig.3.6. In this grasp, fingertip placement between the ridges around the periphery of the gasoline cap provide form closure against relative rotation about the axis of the helix of the threads and also against translation perpendicular to that axis, but the other three degrees of freedom are restrained through force closure. Strictly speaking, given a grasp of a real object by a human hand it is impossible to prevent relative motion of the object with respect to the palm due to the compliance of the hand and object. Preventing all motion is possible only if the contacting bodies are rigid, as is assumed in most mathematical models employed in grasp analysis.

3.5 Safe Grasping: a Possible Approach

3.5.1 Optimal Actuator Efforts and Contact Forces

In order to grasp a given object, the fingers have to slowly close around it until they cannot close further. Each link has to stop only when it reaches its joint angle

Figure 3.6: In the grasp depicted, contact with the ridges on the gasoline cap create partial form closure in the direction of cap rotation (when screwing it in) and also in the directions of translation perpendicular to the axis of rotation

limit or when one of the successive segments in its chain collides with the object. The idea is shown in Fig.3.7.

Anyway, using this approach, the contact points are only touched by the fingers. No force or only little force due to offset error is applied through these contact points. Hence, it is necessary to determine the optimal actuator efforts and the corresponding contact forces.

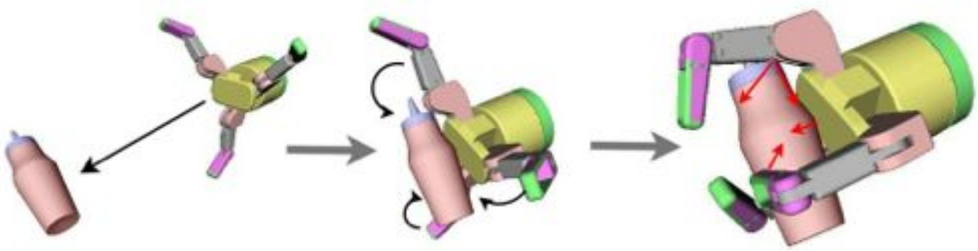


Figure 3.7: A possible approach in order to grasp an object

3.6 Safe Grasping

3.6.1 Form Closure and Force Closure Methods

Holding and manipulating delicate objects such as a glass of water, introduces the concept of safe grasping.

Definition 1 Safe grasp. *In order to perform a safe grasp, a robot must grasp the object in a stable way to prevent slipping by applying the minimum necessary grasping force to avoid object breaking or deformation.*

It is a common practice among researchers to simplify dynamic of system by assumptions such as permanent contact or incipient slip condition between object and gripper. Although such assumptions might be good in general grasping problem, where applying large contact forces is permissible, they are not valid in the case of safe grasping in which the contact forces should be minimal.

Safe grasping can be accomplished by two different ways, the first - called form closure - is based on handling object with frictionless contacts, while, the second - called force closure - is based on holding the object by applying proper amount of friction force to it. Although, the first method of grasping seems to be more convenient for safe grasping purpose, it needs very good knowledge of geometry of the object. Moreover, it needs a dexterous hand which has enough fingers to hold the object in a stable way. Such a hand may not be able to perform a good form closure grasping as the object shape, in real situation, may be somewhat different from what is expected to be. This means that, perhaps, form closure is not a suitable method for grasping unknown objects or objects in unknown circumstances. Also, form closure is not a good approach for grasping tiny delicate objects which need pinching.

On the other hand, force closure, which could be regarded as the solution for grasping of unknown objects, implies the need for a precise force control to accomplish safe grasping. In addition safe grasping by force closure needs detection of object slipping and use of this information in control of the fingers in order to hold the object with minimum necessary force - like what the human hand does.

Adigh and Ahmadi proposed a new controller inspired by human method of grasping [8]. The proposed controller, which only needs measurement of the contact forces, is employed in conjunction with jacobian transpose method for grasping by multi-link fingers.

3.7 A Controller Inspired by Human Method of Grasping

3.7.1 Problem Formulation

A gripper, with two multi-link fingers, as shown in Fig. 3.8 is assumed to grasp and move an object along a vertical line. Considering the free-body diagram of

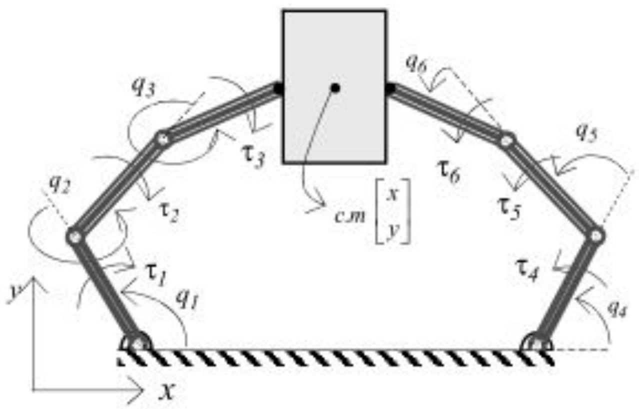


Figure 3.8: Schematic diagram of an object grasped by a two-finger gripper.

object, shown in Fig. 3.9, and taking into account that the motion is rectilinear in z direction, one might simply derive its equation of motion as follow:

$$f_R + f_L - mg = m\ddot{z}_o \tag{3.24}$$

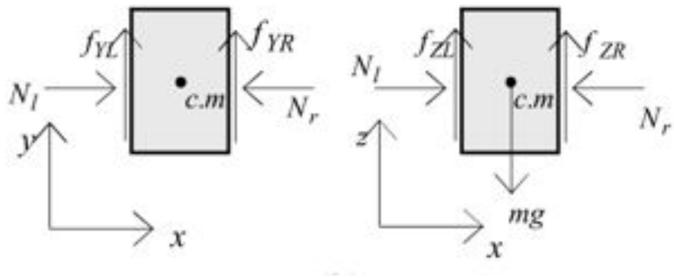


Figure 3.9: Free-body diagram

in which mass and acceleration of the object in z direction, gravity acceleration, and friction forces on left and right sides are respectively shown by m, $\ddot{z}_O$, g, f_L and f_R.

To compute the friction forces we need to consider four different situations that may occur in reality; i.e. incipient slip, slip, non-slip and free falling modes. The equations of motion for each situation are as follows:

1) Incipient slip mode; in this mode the object is kept with minimum normal force, and it is about to slip.

$$\begin{aligned}
&\ddot{z}_o = (f_R + f_L)/m - g \\
&N = k_e x \\
&f_i = -\mu_i N sgn(\ddot{z}_o - \ddot{z}_g), \qquad i = L, R
\end{aligned} \tag{3.25}$$

Where μ_i, k_e, x and $\ddot{z}_g$ denote, respectively, the coefficient of friction between gripper and object, the equivalent stiffness of the finger including rubber skin, the amount of movement of left and right fingers from their neutral position (the position in which the object is grasped with zero normal force and the acceleration of the gripper).

2) Slip mode;

$$\begin{aligned}
&\ddot{z}_o = (f_R + f_L)/m - g \\
&N = k_e x \\
&f_i = -\mu_i N sgn(\dot{z}_o - \dot{z}_g), \qquad i = L, R
\end{aligned} \tag{3.26}$$

3) Non-slip mode;

$$\begin{aligned} \ddot{z}_{obj} &= \ddot{z}_g \\ f_R + f_L &= m(\ddot{z}_g + g) \\ N &= k_e x \end{aligned} \tag{3.27}$$

4) Free falling mode; in this mode the gripper is opened and the object starts to fall.

$$\begin{aligned} \ddot{z}_o &= -g \\ N &= 0 \\ f_R + f_L &= 0 \end{aligned} \tag{3.28}$$

3.7.2 A Human-like Control Law

The controller proposed by Sadigh and Ahmadi feedbacks the normal and tangential force and based on it decide whether to close the gripper jaws or not. It is necessary to use a control law of the following form for $x'_i \in \mathbb{R}^1$ which shows the closing speed of i-th gripper jaw:

$$\ddot{x}_i = \frac{\nu - k_N \left(N_i - \dfrac{k_\mu |f_i|}{\mu_i} \right)}{2}, \qquad i = L, R \tag{3.29}$$

in which k_N, k_μ and ν are some constants to be chosen to meet desired performance. On the other hands μ_i, N_i, and f_i are, respectively, friction coefficient and the measured values of normal and friction forces of the i-th finger. It also worth mentioning that $\ddot{x}_i$ is considered positive when fingers are moving toward each other. The first term in this controller gives a constant intention of closing the gripper with a speed of ν; while the second term gives an intention of opening the gripper with a speed proportional to $N_i - \dfrac{K_\mu |f_i|}{\mu_i}$ which is an indication of excessive applied normal force.

The action of this control is such that at the beginning of grasping procedure, when normal and friction forces are zero, it closes the fingers with speed of ν. As the normal force increases and exceeds the necessary amount of normal force the second term starts to reduce the closing speed and even may reverse the direction of $\dot{x}$ and opens the gripper. This procedure ends when these two intentions get balanced.

3.7.3 Friction Coefficient Estimator

As the discussed controller is a function of friction coefficient, it is quite predictable that its performance greatly depends on good knowledge of it, which may not always be available forehand. Investigating sever effect of uncertainties in μ , Sadigh and Ahmadi [40] proposed an estimator which takes advantage of coulomb friction properties. Recalling that f/N always gives an upper limit for μ and once the system starts to slip it gives the actual value of μ , and the fact that during slip mode the ratio of f and N remains constant irrespective of the amount of N, they proposed the following algorithm to estimate μ.

- Choose the first estimate of μ to the best of our knowledge. It might even be a constant value depending on normal operational condition of the system.
- for $N \neq 0$ calculate $\hat{\mu}_{(j)} = f_{(j)}/N_{(j)}$ at each step of measurement. Where $f_{(j)}$ and $N_{(j)}$ depict the measured values of friction and normal forces at j-th step of measurement. If $N_{(j)} = 0$ we may just skip estimation at this step of measurement.
- For $\hat{\mu}_{(j)} \geq \mu$ put $\mu = \hat{\mu}_{(j)}$. This means that the friction has a capacity more than what expected before.
- For $\hat{\mu}_{(j)} \leq \mu$ two cases might happen:
  - Case1, in this case we get $\hat{\mu}_{(j)} \neq \hat{\mu}_{(j-1)}$ which means the system is none slipping so part of the friction capacity is used and f/N is greater than μ . In this case $\hat{\mu}_{(j)}$ can not improve the estimated value of μ . So, we may leave μ as it is.
  - Case2, the system is slipping or it is in incipient mode. In this case the system experiences the situation of $\hat{\mu}_{(j)} = \hat{\mu}_{(j-1)}$ where $N_{(j)} \neq N_{(j-1)}$.

It means that in this case even with change of normal force from step $j-1$ to step j the value of $f_{(j)}/N_{(j)}$ remains constant. In this case we may put $\mu = \hat{\mu}_i$. This way, we may judge about slipping merely based on knowledge of contact forces.

This estimator saves the best estimated value of friction coefficient in μ which is used in controller.

Chapter 4

Modular Gripper Design Algorithm: an Object-oriented Approach

This chapter deal with the innovation and the advantages brought by the possibility to adjust the modular robot structure depending on the object to grasp size and shape. We present a possible object-oriented algorithm for designing effective modular manipulator configurations in order to grasp a given object or a set of objects.

4.1 Limitation of Robotic Hands

Human hands have great potential not only for grasping objects of various shapes and dimensions, but also for manipulating them in a dexterous manner. In literature, "dexterity" means the capability of changing the position and orientation of the manipulated object from a given reference configuration to a different one, arbitrarily chosen within the hand workspace [41], while "grasping" indicates an action of a hand on an object consisting in preventing its motions relative to the hand, possibly in the face of disturbance forces acting on the object itself. The task of grasping is, in some sense, converse to that of manipulation, and it can be expected that in the design of a robotic hand, or more in general a robotic gripper, trade-offs between dexterity and grasping robustness have to be sought. It is obvious that this kind of dexterity cannot be achieved by a very simple gripper capable of open/close motion only. A multi-fingered robot gripper can therefore provide a great opportunity for achieving such a dexterous manipulation in a robotic system [42].

When designing a grasping device, depending on the intended use, there are many issues that should be addressed: anthropomorphic or non-anthropomorphic aspect, number and kinematic configuration of the fingers, built-in or remote actuation, transmission system (in the case of remote actuation), sensor assignment, integration with a carrying device (robot arm), and control. Anthropomorphic design permits an easier mapping of human skills allowing to plan and to program actions by "demonstrating" the desired human behaviors in manipulation and grasping[43]. In such systems, easily available sensorized gloves, or in some cases mechanical masters, are used to provide measurements of the master's hand movements. Moreover, some results obtained in neuroscience can be also used to improve control algorithms and to reduce the dimension of the controllable DOFs [44]. However, this approach has also some disadvantages especially when the environment is not totally available for design decisions and the controlling is done by computer programs. It is worth noting that human-like grippers present complex kinematic structure, a high number of actuators and a high sophistication of sensing systems. Cost-effectiveness and reliability are at a premium in factory applications of robot hands, and make the simplest grippers an optimal solution for most trivial grasping tasks. Manufacturing large enough batches of products justifies the development of specialized grippers for the task even if the need for adaptability in part-handling devices becomes more and more important nowadays.

4.2 Why Modular

We focus here on the advantage obtained by a modular robotic grasping design. A modular construction allows to overcome some of the problems mentioned above by introducing a task-oriented or object-oriented approach to the design of the grippers. In other words, we define the guidelines for creating a device capable of adapting its structure and functionality to the characteristics of an object or a set of objects to be grasped.

In doing this we want to respect the principle of minimalism: choose the simplest mechanical structure, the minimum number of actuators, the simplest set of sensors, etc., that will do the job, or class of jobs. Complexity reduction is especially important in terms of the hardware components system, as they often make for most of the cost, weight, and failures of robots [45]. Using a dedicated gripper for an object or a set of objects would enhance the simplicity of the task planning and control. Moreover, the production costs would be considerably cut by building a specialized device capable of grasping objects by using only the number of modules and fingers required. The costs of maintenance and repair would also be kept low. Besides, the weight of the manipulator would be minimized to bare necessities. This would be very useful in space applications where it is really important to reduce the weight of the device to be sent into space. Finally, the modular structure can lead toward the miniaturization of the gripper since the reduction of its size only depends on the module characteristics, while the structure can be kept. This property of scalability can also be useful for dealing with objects of unknown size. In fact, the dimension of the device can potentially grow or decrease without modification on the control law or in the determination of the configuration.

4.3 Modular Gripper Design Algorithm

We defined an iterative procedure to design the structure of the gripper. The idea is to simulate the grasping of an object starting from a simple configuration and add modules until a certain value of grasp quality for the given object is achieved. The possibility of dealing with several objects and the easy and fast reconfiguration of the gripper structure was considered. The main idea of the algorithm is shown in Fig.4.1.

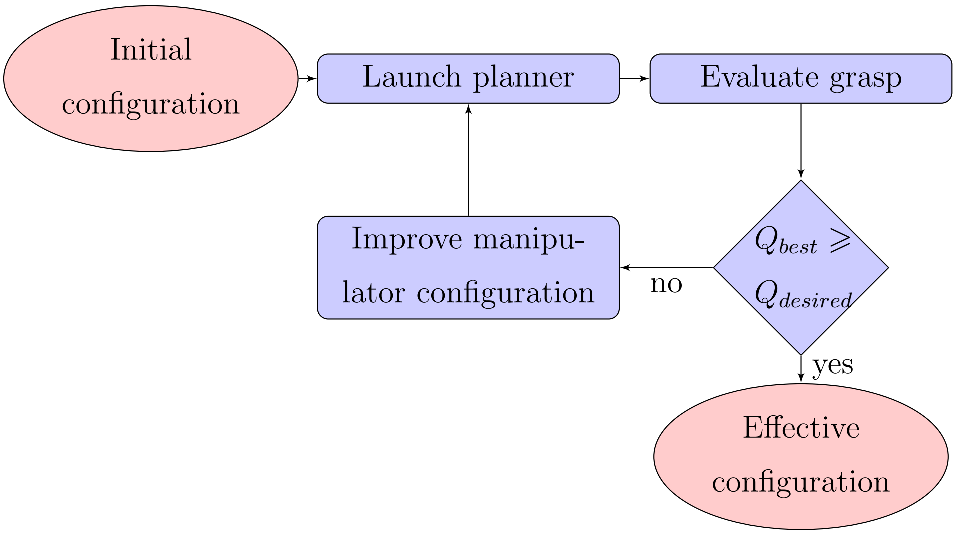


Figure 4.1: The basic idea of the algorithm

The modular gripper consists of one or more chains of modules fixed on a base. Referring to a robotic hand, each chain can be thought as a finger and the base as a palm. We assume that all the modules have the same size and shape.

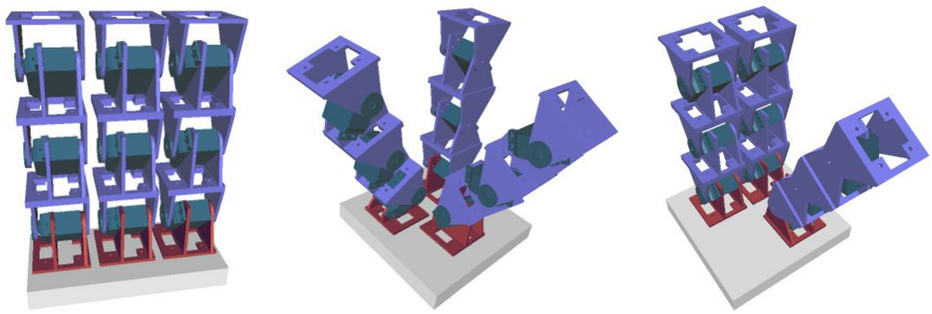
Firstly, the main variables of the algorithm are introduced. Let m be the modules total number of the gripper. Let M be the maximum number of modules per each finger. M is chosen as a trade-off between an upper bound function of the maximum motor torque expressed by the modules and a lower bound defined as

$$M_{min} = \left\lceil \frac{R}{L} \right\rceil, \tag{4.1}$$

where R represents the radius of the minimum sphere that envelops the object to grasp and L is the length of one module. M_{min} takes into account the dimension of the object to grasp. Obviously, the upper bound is the most important limitation.

Concerning the base of the gripper, it is our idea that it should be composed of modules itself. However, in this preliminary study we considered a prototype base plate with given positions where the modules can be fixed. We defined three kinds of *base dispositions* as shown in Fig.4.2 that allow to perform almost all the possible pre-shapes:

- *no finger opposition base*: the fingers are placed in a way that finger opposition is not possible;
- *circular base*: the fingers are placed in a circular configuration;
- *i-opposable-thumbs base*: one or more fingers are set to be opposable to the others.



(a) (b) (c)

Figure 4.2: Possible base dispositions for $f = 3$: *no finger opposition* (a), *circular* (b) and 1-*opposable-thumbs* (c).

We have also considered the possibility to change the distance between the slots where the fingers are placed on the base. This is the first *heuristic* of the proposed algorithm since these three kinds of bases do not cover all possible grippers configurations but, are able to describe the most significant grasp models according to the taxonomy presented in [46].

The basic idea of the algorithm is to design a modular device through an iterative and incremental procedure. The target is to reach a prefixed grasp quality $Q_{desired}$ using the minimum number of modules. In literature, there are different methods for assessing the grasp quality. Among the possible solutions, we used the quality criteria introduced by Ferrari and Canny in [47]. However, all the other solutions could be implemented and used in our algorithm. In particular we used as measure the radius of the largest inscribed sphere centered at the origin contained in the so called *Grasp Wrench Space*. The GWS is the set of all wrenches that can be resisted by a grasp if unit contact forces are applied at the contact points. This measure represents also the magnitude of the largest worst-case disturbance wrench that can be resisted by a grasp with a unit strength grip. It is usually called *Q1*.

At every iteration a new gripper configuration is generated. A grasp planner is used to determine the best grasp achievable with the current configuration. If the corresponding grasp quality is less then $Q_{desired}$ and all the possible module permutations for the fingers as well as all the base configurations have been tried, the current gripper structure changes adding one more module.

In the following the steps of the algorithm, whose flow chart is reported in Fig. 4.3, are described.

Table 4.1: Relevant parameters and variables

m	Current number of modules
f	Current number of fingers
M	Maximum number of modules per finger
f_{min}	Minimum number of fingers
$Q_{desired}$	Predefined desired grasp quality

Initialize algorithm

In this phase the value of M, $Q_{desired}$ and f_{min} are assigned. f_{min} represents the minimum number of finger that the gripper must have. Moreover the features of the modules and of the object to grasp are defined. When the algorithm starts the gripper configuration consists of only one finger and one module and $f_{min} = 1$.

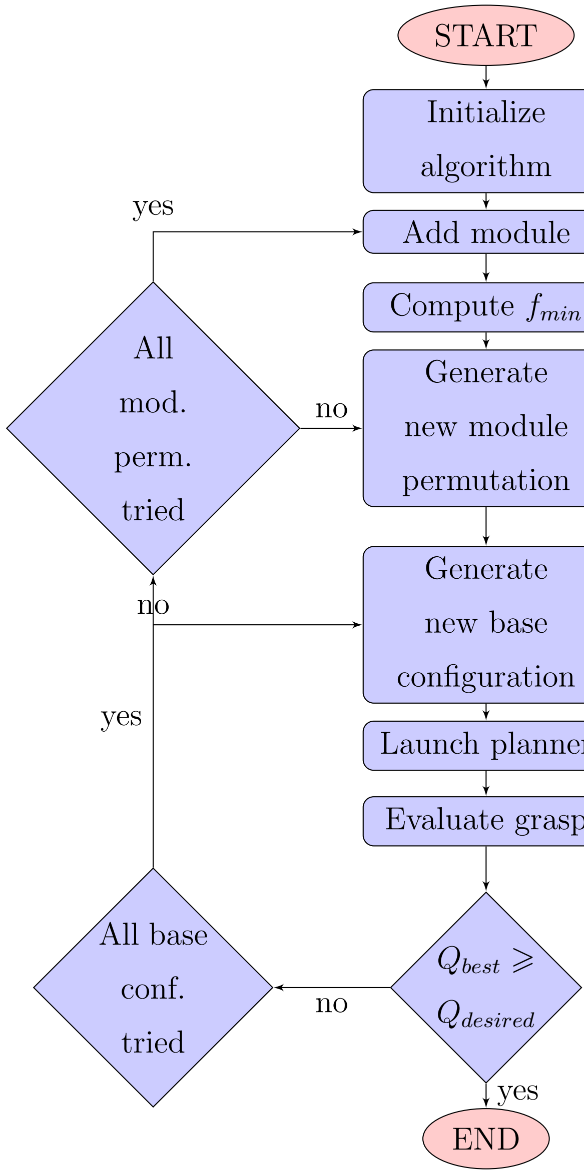


Figure 4.3: Algorithm for minimal configuration.

Compute f_{min}

When a module is added, the following condition is evaluated:

$$\frac{m}{f_{min}} \leqslant M. \tag{4.2}$$

If the inequality is not satisfied, f_{min} is incremented by one to avoid the insertion of more than M modules into a finger. Let suppose that we have $m = 4$ and $M = 3$. f_{min} is set to 2. So that, it is not possible to have a configuration with only one finger with four module and the limit M is respected.

Generate new module permutation

This step is necessary to guarantee that given a number of modules, all the possible configuration of fingers are tried. All the module permutations for the fingers can be obtained as:

$$\sum_{i=1}^{F} x_i = m, \tag{4.3}$$

where $x_i \in \mathbb{N}$ and $f_{min} \leq F \leq m$. Hence x_i represents the number of modules for the i-th finger.

Generate new base configuration

This step consents to test all the possible base dispositions and the distances between the slots once a permutation is generated. Firstly, it is necessary to select one of the defined *base dispositions*. Then the distance between the fingers has to be selected. The number d of different distances to be considered in the algorithm is related with the size of the object to grasp and it is defined as:

$$d = \left\lceil \frac{R}{s} \right\rceil, \tag{4.4}$$

where s represents the distance between two adjacent slots and R is again the radius of the minimum sphere that envelops the object to grasp. The *first distance* can be obtained by fixing the fingers in adjacent slots, the *second distance* can be obtained by simply fixing the fingers at each two slots and so on. An example of second distance is showed in Fig. 4.4.

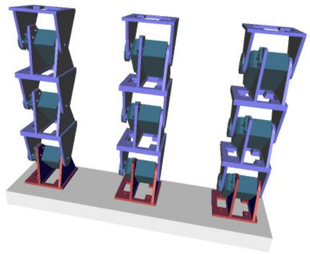


Figure 4.4: Second finger distance for a *no finger opposition* base disposition.

Launch planner

Once the configurations of fingers and base are defined and a new gripper configuration is generated, a grasp planner is used in order to determine the grasps achievable with the current configuration. In general, the grasp planning problem can be solved in either forward or backward direction. We used a forward solution that is implemented in the grasp planning simulator "Openrave" [48] in which a grasp is simulated by giving an initial pose and initial joint angles (pre-shape) to the end-effector. After that the end effector moves along the normal to the palm plane until it hits the target object. Then the fingers of the end-effector slowly close around the object until they cannot close further. The contacts between the end-effector and the object are extracted, and grasp quality index is calculated.

Note that we used the grasp planner implemented in Openrave but this does not affect the generality of our approach since any other planner, like for instance the one implemented in *GraspIt!* [49], can be used without affecting the effectiveness of the proposed iterative algorithm.

Evaluate Grasp

As we said before, in order to assess the grasp quality, we used the quality criteria introduced by Ferrari and Canny in [47]. However, all the other solutions could be implemented and used in our algorithm.

In the following, we briefly report the main aspects of this criteria. A grasp is said *force closed* if given any external wrench applied to the object there exists a set of contact forces that can counterbalance the external wrench. The set of all wrenches that can be resisted by a grasp if unit contact forces are applied at the contact points is called the *Grasp Wrench Space* (GWS). In order to find the total grasp wrench space we need a finite basis set of vectors, so it is necessary to approximate this cone with a proper pyramid of k sides. Therefore each contact force can be expressed as the convex sum of forces f_j on the boundary of the friction pyramid:

$$f = \sum_{j=1}^{k} \alpha_j f_j. \tag{4.5}$$

In [47] two methods of finding the unit grasp wrench space are described. One limits the maximum magnitude of the contact normal forces to 1, and the other limits their sum magnitude to 1. We have implemented the second option due to its ease of computation. Under this constraint, the set of wrenches that can be exerted on the object is:

$$GWS = ConvexHull\left(\bigcup_{i=1}^{n} \{w_{i,1}, ..., w_{i,k}\}\right), \tag{4.6}$$

or equivalently

$$GWS = \sum_{i=1}^{n} \sum_{j=1}^{k} \alpha_{i,j} \cdot w_{i,j}, \tag{4.7}$$

or equivalently $GWS = \sum_{i=1}^{n} \sum_{j=1}^{k} \alpha_{i,j} \cdot w_{i,j}$, with $\sum_{i=1}^{n} \sum_{j=1}^{k} \alpha_{i,j} \leq 1$ and $\alpha_{i,j} \geq 0$. If the convex hull contains the center of mass, i.e. the origin of the wrench space, then the grasp is force closed. The definition of the standard grasp quality measures follows from this approximation of the GWS. In particular we used *Q1*, one of the most used grasp measures that is the radius of the largest inscribed sphere centered at the origin contained in the GWS. It represents the magnitude of the largest worst-case disturbance wrench that can be resisted by a grasp with a unit strength grip. If $Q1 < 0$ the convex hull does not contain the origin and the grasp is non force closed.

Stop condition

The algorithm stops when the desired value for the grasp quality *Q1* is reached by the best grasp returned by the planner. The corresponding configuration is selected as the minimal configuration. This is the second *heuristics* of our algorithm.

It should be mentioned that the proposed algorithm works for a given object to grasp and this may appear a limitation of the approach. However, during the use of the algorithm, we realized that even if the modular gripper is designed for a given object it is able to work with many other objects with comparable size. To generalize this idea to a set of objects, we propose to consider as target object for the modular gripper design the object obtained considering the bounding box of all the elements in the set as described in [50].

4.4 Pseudocode of the Algorithm

In the following, we propose the pseudo-code of the proposed algorithm.

```
target = object to grasp; define: $Q_{desire}$, module, $M$, $s$
$m = f_{min} = 1$
distances.generate($s$)
permutations.generate($m$, $f_{min}$)
while true do
    if $m/f_{min} \geq M$ then
        $f_{min}++$
        permutations.generate($m$, $f_{min}$)
    end if
    while permutations.hasNext() do
        current permutation = permutations.next()
        dispositions.generate(current permutation)
        while dispositions.hasNext() do
            current disposition = dispositions.next()
            while distance.hasNext() do
                current distance = distances.next()
                current configuration = configuration.generate(current permutation, current disposition, current distance)
                $Q_{best}$ = planner.launch(current configuration, target)
                if $Q_{best} \geqslant Q_{desire}$ then
                    return current configuration
                end if
            end while
            distances.reset()
        end while
    end while
    m++
end while
```

Algorithm 1: Algorithm pseudo-code.

Chapter 5

Simulations and Experimental Results

In this chapter we expose some results obtained using our iterative algorithm. We used a software simulator in order to test our procedure. Finally, two possible implementations of our modular gripper are reported.

5.1 Effective Configurations for Everyday Objects

To test our algorithm we performed some simulations. All simulations were carried out using Openrave. As base module we used the $Y1$ model [23]. A single body module is 80 mm long, 50mm wide and 50 mm high. We considered $M = 3$ as the maximum number of module per finger and two different distances between the slots ($d = 2$).

The algorithm was tested for finding effective configurations of the modular gripper for several everyday objects.

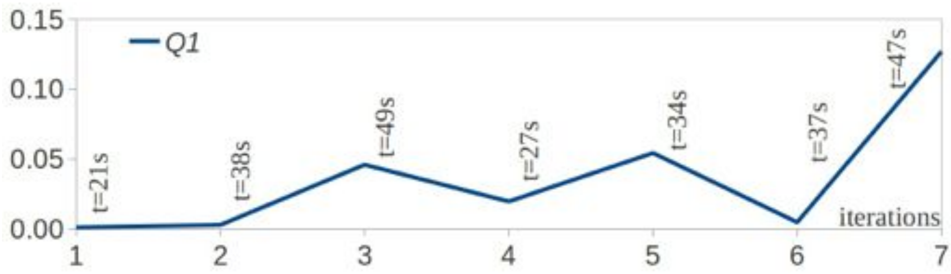


Figure 5.1: Steps of the algorithm for finding the minimum configuration in order to grasp a ketchup bottle.

5.1.1 A Simple Demo: Grasping a Ketchup Bottle

For the sake of simplicity, only the example of a ketchup bottle is described here in detail. According to [51] the threshold was set at $Q_{desired} = 0.1$ since this or a greater measure of quality corresponds to grasps that a human would consider "stable".

We observed that one finger with three modules is enough to reach the desired grasp quality. The steps of the algorithm are shown in Fig. 5.1. The reported value of *Q1* refers to the best obtained grasp for each gripper configuration tested. In Table 5.1 experiment details are reported. The minimum configuration obtained is shown in Fig. 5.3-a.

Looking for the Best Achievable Configuration

For the same object the number of modules (m) was continually increased even after reaching $Q_{desired}$ and it was verified that after a certain number of modules m_{opt} the value of the grasp quality remained almost stable around a threshold as is shown in Fig. 5.2 . In this case m_{opt} was 9. The obtained device is represented in Fig. 5.3-b. For this reason, in general, we can consider the configuration corresponding to m_{opt} modules as the best achievable.

Table 5.1: Steps of the algorithm for the ketchup bottle minimum configuration

iter.	configuration	*Q1*	time
I	$m = 1,\ f = 1$	0.0011	21s
II	$m = 2,\ f = 1$	0.0028	38s
III	$m = 2,\ f = 2,$ lin. disp., 1^{st} scal.	0.0459	49s
IV	$m = 2,\ f = 2,$ lin. disp., 2^{st} scal.	0.0196	27s
V	$m = 2,\ f = 2,$ circ. disp., 1^{st} scal.	0.0543	34s
VI	$m = 2,\ f = 2,$ circ. disp., 2^{st} scal.	0.0047	37s
VII	$m = 3,\ f = 1$	0.1270	47s

5.1.2 Grasping Other Objects or Sets of Objects

Some other simulations were performed in order to obtain effective configurations for grasping other objects or sets of objects. A phone, a book, a flask, a cup, a glass and an aircraft model were tested. The resulting configurations are shown in Fig. 5.4. It is worth noting that the complexity of the gripper increases together with the complexity of the shape of the object. For the aircraft model we also tested the possibility to change the direction of the modules rotation axis. What was obtained is quite different from classical grippers and is reported in Fig. 5.4-f. Table 5.2 shows in detail the obtained gripper configurations and the correspondent grasp qualities.

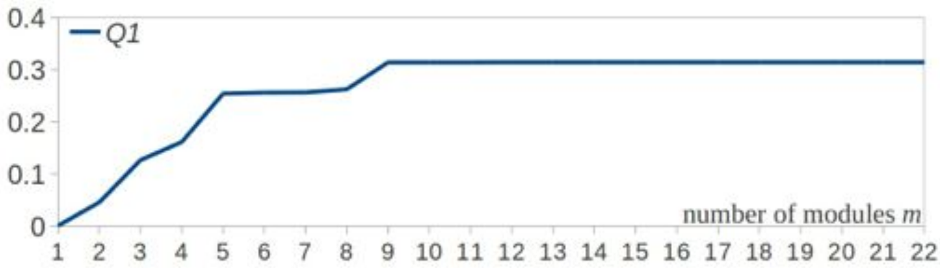


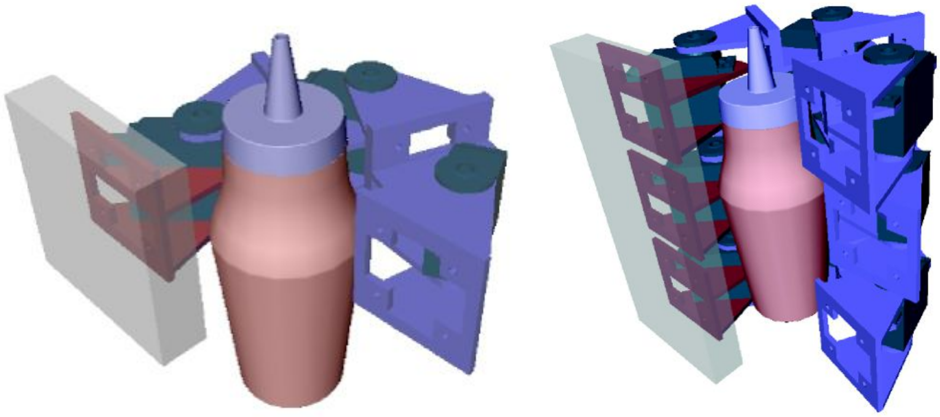
Figure 5.2: Steps of the algorithm for finding the best configuration in order to grasp a ketchup bottle. The grasp values refer to the best grasp quality obtainable at each step with m modules

Table 5.2: Effective confgurations for some everyday objects

object	m	f	base disposition	finger distance	module permutation	*Q1*
phone	3	1	linear	1^{st}	3	0.13
book	5	2	circular	2^{nd}	2,3	0.13
flask	9	3	circular	1^{st}	3,3,3	0.14
cup	4	2	circular	2^{st}	2,2	0.11
glass	5	2	linear	1^{st}	2,3	0.12
aircraft	14	3	circular	1^{st}	5,5,4	0.53

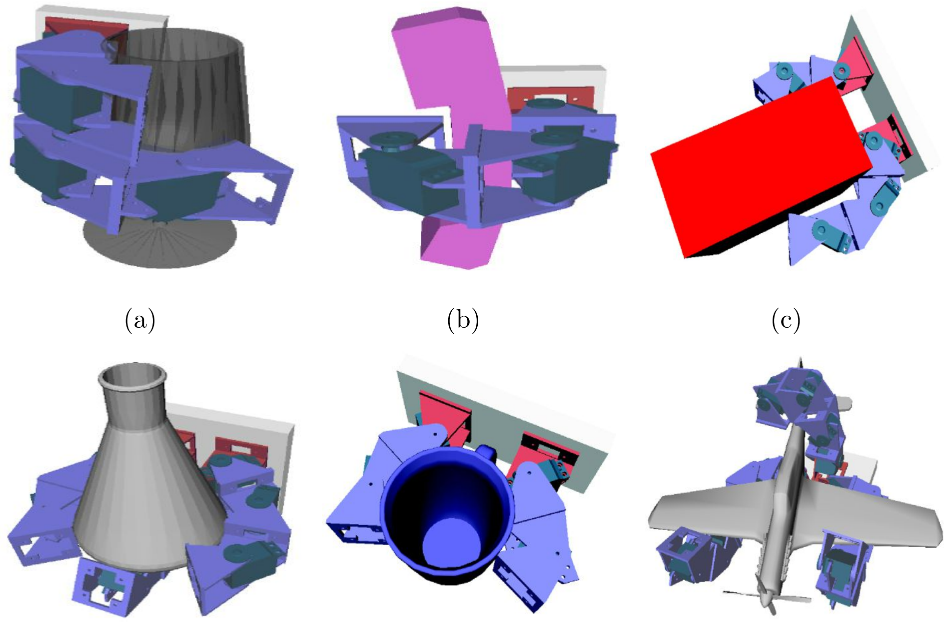
5.2 Possible Implementations

Finally, two possible implementations of our modular gripper are reported. In Fig. 5.5 the obtained gripper for cubic and cylindrical object are shown. In this case we used a different module size for the design. The used modules dimensions are 42 mm long, 33 mm wide and 16 mm high.



(a) (b)

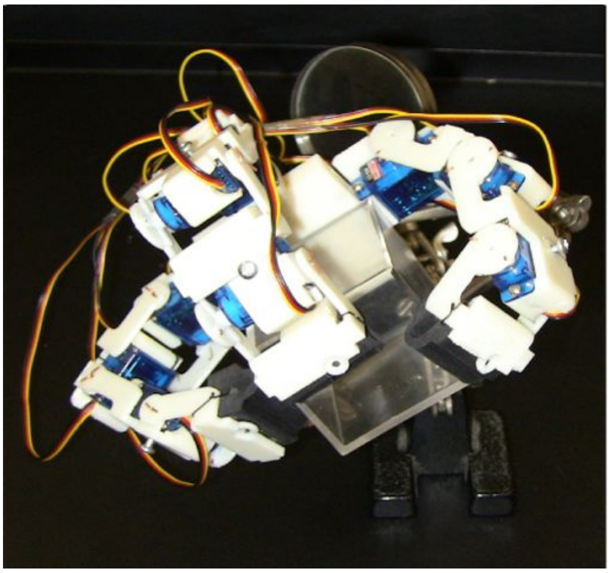
Figure 5.3: Minimum (a) and best (b) gripper configurations for grasping a bottle of ketchup.



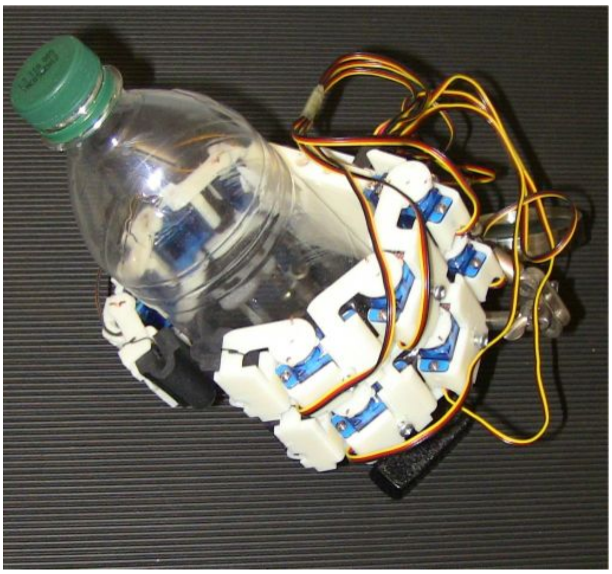
(a) (b) (c)

(d) (e) (f)

Figure 5.4: Minimum manipulator configurations for respectively grasping a glass (a), a phone (b), a book (c), a flask (d), a cup (e) and an aircraft model (f).



(a)



(b)

Figure 5.5: Gripper configurations for cubic (a) and cylindrical (b) objects.

Chapter 6

Imitating the Human Postural Synergies

In this chapter the possibility of using human postural synergies in order to control modular manipulators is considered.

6.1 Problem: High-dimensional Configuration Space

One of the hardest problems in robotic grasping is the creation of control algorithms for new hand designs that are beginning to rival the human hand in complexity. This also happens using the approach proposed by Adigh and Ahmadi [8] and discussed in Chapter 4. A possible solution to this problem could be to draw inspiration from the human hand not only with regard to size and configuration, but rather as regards to the control [52]. Recent studies in neuroscience research have shown that the grasping human hand movements can be described by trajectories in a configuration space of much smaller dimension than the kinematic count would suggest. Such configuration space is known as postural synergies and its main components are referred as eigengrasps.

6.2 Eigengrasps

Any hand posture is fully specified by its joint values, and can therefore be thought of as a point in a high-dimensional joint space. If d is the number of degrees of freedom (DOF) of the hand, than a posture p can be defined as

$$p = [\theta_1 \theta_2 ... \theta_d] \in \mathcal{R}^d \quad (6.1)$$

where θ_i is the value of i-th degree of freedom.

Most human grasping postures derive from a relatively small set of discrete pre-grasp shapes. These hypothesis are supported by the work of Santello et al. [53]. In his work, a certain number of subjects were asked to shape the right hand as if to grasp and use a large number of familiar objects. Static hand posture was measured by recording the angular position of 15 joint angles of the fingers and of the thumb. Although subjects adopted distinct hand shapes for the various objects, the authors noticed that the joint angles of the digits did not vary independently. Principal components analysis showed that the first two components could account for more than 80% of the variance, implying a substantial reduction from the 15 degrees of freedom that were recorded.

In other words, the first two components form a low-dimensionality basis for grasp postures, and can be linearly combined to closely approximate most common grasping positions. Allen et al. referred to the Principal Components of the postures as eigengrasps [52].

Each eigengrasp e_i is a d-dimensional vector and can also be thought of as direction of motion in joint space. Motion along one eigengrasp direction will usually imply motion along all (or most) degrees of freedom of the hand.

$$e_i = [e_{i,1} e_{i,2} ... e_{i,d}] \quad (6.2)$$

By choosing a basis comprising b eigengrasps, a hand posture placed in the subspace defined by this basis can be expressed as a function of the amplitudes a_i along each eigengrasp direction:

$$p = \sum_{i=1}^{b} a_i e_i \tag{6.3}$$

and is therefore completely defined by the amplitudes vector $a = [a_1...a_b] \in \mathcal{R}^b$.

6.3 Grasp Planning Using Eigengrasps

The grasp planning task can be thought as an optimization problem in a high-dimensional space that describes both hand posture (intrinsic DOF's) and position (extrinsic DOF's). Let us consider the goal of minimizing an energy function of the form:

$$E = f(p, w) \tag{6.4}$$

If d is the number of intrinsic hand DOF's then $p \in \mathcal{R}^d$ represents the hand posture and $w \in \mathcal{R}^6$ contains the position and orientation of the wrist.

Intuitively, this energy function has to be related to the quality of the grasp. However, the traditional formulations pose a number of problems. First, it can be very difficult, or even impossible, to compute an analytical gradient. Second, such functions are highly non-linear, as small changes in both finger posture and wrist position can drastically alter the quality of the resulting grasp. Finally, the legal parameter space is complex, having to satisfy multiple constraints: prevent inter-penetration with the object to be grasped as well as potential obstacles, and maintain joint values within their acceptable ranges.

For all these reasons, Ciocarlie in [52] proposed to perform the optimization in eigengrasp space, as opposed to DOF space. The energy function takes the form

$$E = f(a, w) \tag{6.5}$$

where $a \in \mathcal{R}^2$ is the vector of eigengrasp amplitudes. This effectively reduces the parameter space to 8 dimensions (2 eigengrasp amplitudes plus 6 extrinsic DOFs) from as high as 26 dimensions in the case of the human hand.

One possible quality metric that can be adapted and used as energy function is the one described by Ferrari and Canny [54].

6.4 Application for Robotic Hand Models

Although the work of Santello et al. is centered on the study of the human hand, the eigengrasps approach is extremely useful for robotic hands as well. Ciocarlie in [52] applied the eigengrasp concept to a total of 5 hand models: a simple gripper, the Barrett hand, the DLR hand [55], the Robonaut hand [56] and finally a human hand model.

The eigengrasp concept allows to design flexible control algorithms that operate identically across different hand models. The key to this approach is that the eigengrasps encapsulate the kinematic characteristics of each hand design. This enables control algorithms that operate on eigengrasp amplitudes to ignore low-level operations and concentrate on the high-level task.

6.5 Application for Modular Robotic Gripper

Despite the simplicity of our modular manipulator model, with the increase in the number of its fingers and modules, it also becomes rival to the human hand in complexity. The added degrees of freedom make modular robots more versatile in their potential capabilities, but also incur a performance tradeoff and increased mechanical and computational complexities. A possible solution to this problem is to use the human eigengrasps in order to control our modular manipulator. In this way we can operate identically across all the possible modular grippers.

The concept of synergies is closely related to the human hand, but if we analyze the problem as a simplification of the grasp search space we can extend this concept also to the modular robotic hand.

However, modular hands can be very different from the human one. The number of fingers, the finger size and shape, the number of phalanges per finger, the manipulator base configuration are all parameters that may vary. Intuitively, the quality and the quantity of achievable synergies depend by the hand configuration.

6.5.1 A Possible Gripper Model

In order to explore this approach, we decided to use the two dominant human eigengrasps in order to control one of the gripper obtained using our design algorithm.

While we found our choices to produce good results, the optimal choice of eigengrasps for non-human hands, as well as the choice of which eigengrasps to use for a particular task, are open questions and interesting directions for future research.

Referring to the nomenclature introduced in the previous chapter, the modular gripper that we considered for performing this experiment is characterized by the following features:

- two fingers;
- three modules for each finger;
- six intrinsic DOFs;
- *1-opposable-thumbs base*.

Once again we used the $Y1$ model [23] as base module.

Since the number of intrinsic DOFs is six then $p \in \mathcal{R}^6$ represents the gripper posture. We also have to consider the position and orientation of the base $w \in \mathcal{R}^6$. So, the parameter space has 12 dimensions. Using the two dominant eigengrasps we can effectively reduce the parameter space to 8 dimensions (2 eigengrasp amplitudes plus 6 extrinsic DOFs).

We decided to use *GraspIt!* [49] simulator in order to develop this model. *GraspIt!* already includes eigengrasp information for many dexterous hands. For the human hand, they provide eigengrasp directions matching those discovered through user studies by Santello et al. [53]. Moreover, in *GraspIt!* three more dexterous hands have eigengrasp information pre-defined: the Robonaut hand, the DLR hand and the Barrett hand. In addition, eigengrasp informations can be defined for any manipulator model.

For our gripper model, we defined two different eigengrasps:

- proximal joints flexion shown in Fig. 6.1;
- distal joints flexion shown in Fig. 6.2.

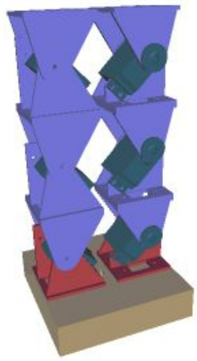


Figure 6.1: Proximal joints flexion eigengrasp

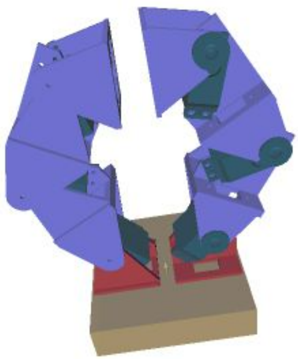


Figure 6.2: Distal joints flexion eigengrasp

We also performed some grasps of a glass using the defined eigengrasps as shown in the following Figures. The advantage of using eigengrasps for controlling the gripper is evident. The *GraspIt!* EigenGrasp Interface menu shows the controls for two dominant eigengrasps.

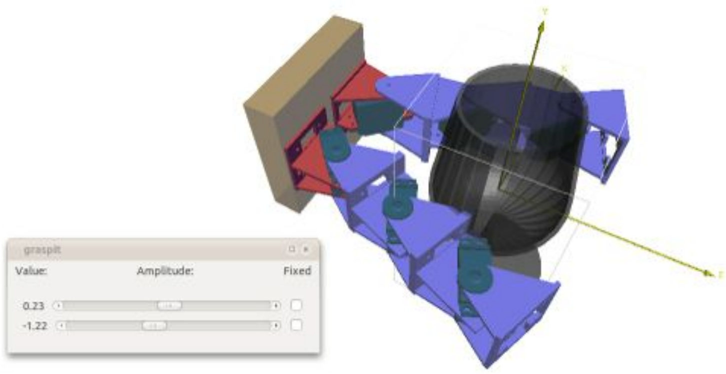


Figure 6.3: Grasp of a glass performed using the proximal joints flexion eigengrasp

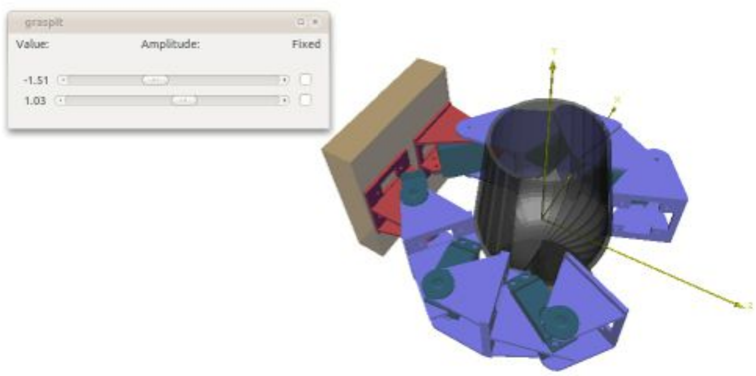


Figure 6.4: Grasp of a glass performed using the distal joints flexion eigengrasp

Chapter 7

Conclusion and Future work

In this work we studied the possibility of developing modular robotic grippers able to adapt to different requirements and situations. We presented an algorithm for finding effective configurations of a modular gripper. We observed that by increasing the number of fingers and modules, it is possible to increase the grasping quality. However, after a certain threshold, we noticed that the grasp quality cannot be further enhanced. Moreover, we realized the modular gripper for two of the obtained configurations with cubic and cylindrical objects.

As last, we investigated the possibility of using the two dominant human eigengrasps in order to control the modular manipulator. We implemented a simple gripper model that can be controlled using this approach. While we found our choices to produce good results, the optimal choice of eigengrasps for non-human hands, as well as the choice of which eigengrasps to use for a particular task, are open questions and interesting directions for future research.

Problems

The proposed algorithm is quite complex from a computational point of view. The most time consuming part is the grasp planning phase. The time required to complete this stage depends on the kind of planner used and on the number of modules involved.

Adapting the device base to the task to be performed could reduce the number of possible configurations to test and it could also allows for a better adaptation of the gripper to the manipulation tasks.

Finally, we want to remark that in this work a given base plate was used as the palm for the gripper. In order to increase the flexibility of the iterative design approach, we have planned to substitute the base plate with modules as well.

7.1 Future works

7.1.1 Task-oriented Grasping

Commonly used criteria for grasp evaluation take into account only forces. These criteria guarantee resistance of the grasp against any external force, which might be applied to the grasped object but it often leads to large forces and torques at the grasping points. Even worse, it may generate useless grasping postures. Some grasps which might be form- or force-closure grasps may not be suitable with respect to a certain task, being executed after the object has been grasped. For example, the grasp of a cup from the top might be very stable and give a good performance in case of force optimization but it is unusable if something is to be filled in the cup or poured out. To be more precise, most of the currently proposed methods do not use any knowledge about the object functionality. They are not task-oriented. To avoid this problem we should also take into account the re-usability of a grasped object.

As future work, we have planned to modify the iterative algorithm to take into account task-oriented quality measures like the one presented in [57]. For this reason we should modify our algorithm to make it task-oriented. Thanks to its modularity, we can maintain the same structure. Only the grasp evaluation phase has to be modified in order to use a task-oriented metric.

7.1.2 Hand-oriented Grasping

Another possible future work could be to modify the proposed algorithm in order to make it hand-oriented. At each iteration, once the current manipulator configuration is generated, the achievable eigengrasps have to be defined. A method for obtaining the optimal choice and mapping of human synergies for non-human hands is necessary. We also need to use an eigengrasps planner. In this way the algorithm could return not only an effective configuration that allows to grasp the given object or a set of objects but also the achievable postural synergies in order to control the gripper.

Appendix A

Source code

In Chapter 6 we presented a simple gripper model that can be controlled using the two dominant eigengrasps. We report here the source code to be used with *GraspIt!*.

A.1 ModularGripper.xml

In *GraspIt!* [49], a robot is made up of multiple links, connected into kinematic chains. In general, in GraspIt!, a robot configuration file contains the following data:

- the palm, this is simply a pointer to the body file that contains the palm;
- degrees of freedom;
- kinematic chains.

We defined all these fields. Below the gripper configuration file.

```
<?xml version="1.0" ?>
<robot type="ModularGripper">
    <palm>palm.xml</palm>
```

```
<dof type="r">
    <defaultVelocity>1.0</defaultVelocity>
    <maxEffort>2.5e+9</maxEffort>
    <Kp>1.0e+11</Kp>
    <Kd>1.0e+7</Kd>
    <draggerScale>20</draggerScale>
    <breakAwayTorque>0.5</breakAwayTorque>
</dof>
<dof type="r">
    <defaultVelocity>1.0</defaultVelocity>
    <maxEffort>2.5e+9</maxEffort>
    <Kp>1.0e+11</Kp>
    <Kd>1.0e+7</Kd>
    <draggerScale>20</draggerScale>
    <breakAwayTorque>0.5</breakAwayTorque>
</dof>
<dof type="r">
    <defaultVelocity>1.0</defaultVelocity>
    <maxEffort>2.5e+9</maxEffort>
    <Kp>1.0e+11</Kp>
    <Kd>1.0e+7</Kd>
    <draggerScale>20</draggerScale>
    <breakAwayTorque>0.5</breakAwayTorque>
</dof>
<dof type="r">
    <defaultVelocity>1.0</defaultVelocity>
    <maxEffort>2.5e+9</maxEffort>
    <Kp>1.0e+11</Kp>
    <Kd>1.0e+7</Kd>
    <draggerScale>20</draggerScale>
    <breakAwayTorque>0.5</breakAwayTorque>
```

```
</dof>
<dof type="r">
    <defaultVelocity>1.0</defaultVelocity>
    <maxEffort>2.5e+9</maxEffort>
    <Kp>1.0e+11</Kp>
    <Kd>1.0e+7</Kd>
    <draggerScale>20</draggerScale>
    <breakAwayTorque>0.5</breakAwayTorque>
</dof>
<dof type="r">
    <defaultVelocity>1.0</defaultVelocity>
    <maxEffort>2.5e+9</maxEffort>
    <Kp>1.0e+11</Kp>
    <Kd>1.0e+7</Kd>
    <draggerScale>20</draggerScale>
    <breakAwayTorque>0.5</breakAwayTorque>
</dof>

<chain>
    <transform>
        <translation>32 −10 50</translation>
        <rotation>90 x</rotation>
    </transform>
    <joint type="Revolute">
        <theta>d0+90</theta>
        <d>0</d>
        <a>82</a>
        <alpha>0</alpha>
        <minValue>−90</minValue>
        <maxValue>90</maxValue>
        <viscousFriction>5.0e+7</viscousFriction>
```

```
    </joint>
    <joint type="Revolute">
        <theta>d1</theta>
        <d>0</d>
        <a>72</a>
        <alpha>0</alpha>
        <minValue>0</minValue>
        <maxValue>68</maxValue>
        <viscousFriction>5.0e+7</viscousFriction>
    </joint>
    <joint type="Revolute">
        <theta>d2</theta>
        <d>0</d>
        <a>50</a>
        <alpha>0</alpha>
        <minValue>0</minValue>
        <maxValue>90</maxValue>
        <viscousFriction>5.0e+7</viscousFriction>
    </joint>
        <link dynamicJointType="Revolute">link2.xml</link>
        <link dynamicJointType="Revolute">link2.xml</link>
        <link dynamicJointType="Revolute">link3.xml</link>
</chain>

<chain>
    <transform>
        <translation>-32 10 50</translation>
            <rotation>90 x</rotation>
            <rotation>180 y</rotation>
    </transform>
    <joint type="Revolute">
```

```
            <theta>d3+90</theta>
            <d>0</d>
            <a>82</a>
            <alpha>0</alpha>
            <minValue>-90</minValue>
            <maxValue>90</maxValue>
            <viscousFriction>5.0e+7</viscousFriction>
        </joint>
        <joint type="Revolute">
            <theta>d4</theta>
            <d>0</d>
            <a>68</a>
            <alpha>0</alpha>
            <minValue>0</minValue>
            <maxValue>70</maxValue>
            <viscousFriction>5.0e+7</viscousFriction>
        </joint>
        <joint type="Revolute">
            <theta>d5</theta>
            <d>0</d>
            <a>50</a>
            <alpha>0</alpha>
            <minValue>0</minValue>
            <maxValue>90</maxValue>
            <viscousFriction>5.0e+7</viscousFriction>
        </joint>
            <link dynamicJointType="Revolute">link2m.xml</link>
            <link dynamicJointType="Revolute">link2m.xml</link>
            <link dynamicJointType="Revolute">link3m.xml</link>
    </chain>
```

```
    <approachDirection>
        <referenceLocation>0 0 0</referenceLocation>
        <direction>0 0 1</direction>
    </approachDirection>
    <eigenGrasps>eigen/barrett_eigen.egr</eigenGrasps>
</robot>
```

Regarding the model of the module we used the same link geometry files availables in the Modular Robots plug-in for Openrave colled OpenMR [58].

A.2 ModularGripper.egr

Using *GraspIt!*, eigengrasp information can be defined using a text file which is loaded together with the hand configuration file. Usually, eigengrasp information files are placed together with the rest of the information that defines a robot, such as the configuration file or link geometry files. The eigengrasp information file has the following characteristics:

- the first line contains the keyword *DIMENSIONS* followed by the number of DOF's of the hand;
- each eigengrasp begins with the keyword *EG*. On the next line, a single value containing the eigenvalue associated with this particular eigengrasp. Finally, on the next line, the d-dimensional vector that defines the eigengrasp;
- an arbitrary number of eigengrasps can be defined (we defined a 2-dimensional subspace with two eigengrasps);
- the origin of the subspace is defined exactly like an eigengrasp, but it is preceded by the keyword *ORIGIN*;
- the normalization information is optional. If desired, it can be defined like an eigengrasp, preceded by the keyword *NORM*. If this information is not present in the file, no normalization is used.

Here we show the source code.

```
DIMENSIONS 6

EG
0.51
1.0 0.0 0.0 1.0 0.0 0.0
EG
0.25
1.0 1.0 1.0 1.0 1.0 1.0

ORIGIN
0.0000
0.0 1.0 0.0 0.0 1.0 0.0
```

A.3 Submission of a paper to "IROS 2011"

About the work outlined in this Thesis, we also submitted a paper to the IEEE/RSJ International Conference on Intelligent Robots and Systems ("IROS 2011"). We attach the paper at the end of this Thesis.

Acknowledgments

Thanks to Professor Domenico Prattichizzo. He supported me from the beginning. He is a great leader and he gave me something to believe in, not something to do. I thank him for making me understand the true meaning of research.

I am very grateful to Eng. Gionata Salvietti for his support and his helpfullness. He is a strong researcher and it was a pleasure for me to work with him.

I thank all my family especially Mom and Dad. Your irreplaceable heartiness, love and understanding are the sources of my strength. Please, accept my heartfelt thanks for your cordiality, trust and whole-hearted support. Thank you for your care, support and incomparable love. I am who I am, because of you. Thank you for that. Thanks also to my sister Elisa and my brother Riccardo for having always been close to me.

A big thanks to my second big family in Siena, the University Chapel of St. Vigilio. Thank you all for the endless love that is still surrounding me. A special thanks goes to Don Roberto for being always supportive and understanding.

I really want to say thank you to Dr. Houxiang Zhang for his priceless help during all my experience in Hamburg. He is also a really good friend and encouraged me a lot. He also taught me to face the criticism of people who do not like my work.

Thanks to Professor Jianwei Zhang who has allowed me to spend six unforgettable months in Hamburg.

Thanks to the TAMS Group for the kindness and for make me felt home, especially Lu who has been always a precious help inside and outside the University. She is a very friendly and active person, I cannot imagine the TAMS Group without

her.

Thanks to the friends I have known in Hamburg and to all my other friends. Friendship is the most valuable thing in the world. So accept my grateful appreciation for being that valuable for me, dear friends.

Bibliography

[1] T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss. Self Organizing Robots Based on Cell Structures-CKBOT. In *Intelligent Robots, 1988., IEEE International Workshop on*, pages 145–150. IEEE, 1988.

[2] M. Yim, Y. Zhang, and D. Duff. Modular robots. *Spectrum, IEEE*, 39(2):30–34, 2002.

[3] M.W. Jorgensen, E.H. Ostergaard, and H.H. Lund. Modular ATRON: Modules for a self-reconfigurable robot. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 2, pages 2068–2073. IEEE, 2004.

[4] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji. M-TRAN: Self-reconfigurable modular robotic system. *Mechatronics, IEEE/ASME Transactions on*, 7(4):431–441, 2002.

[5] H. Zhang, J. Gonzalez-Gomez, Z. Me, S. Cheng, and J. Zhang. Development of a low-cost flexible modular robot GZ-I. In *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, pages 223–228. IEEE, 2008.

[6] G. Salvietti, HX Zhang, J. Gonzalez-Gomez, D. Prattichizzo, and JW Zhang. Task priority grasping and locomotion control of modular robot. In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pages 1069–1074. IEEE, 2009.

[7] H. Zhang, J. Gonzalez-Gomez, Z. Me, S. Cheng, and J. Zhang. Development of a low-cost flexible modular robot GZ-I. In *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, pages 223–228. IEEE, 2008.

[8] M.J. Sadigh and H. Ahmadi. Safe grasping with multi-link fingers based on force sensing. In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pages 1796–1802. IEEE, 2009.

[9] ZM Bi and WJ Zhang. Concurrent optimal design of modular robotic configuration. *Journal of Robotic systems*, 18(2):77–87, 2001.

[10] Wikipedia. Self-reconfiguring modular robot, 2011. [Online; accessed 3-April-2011].

[11] S. Murata and H. Kurokawa. Self-Reconfigurable Robots. *IEEE Robotics and Automation Magazine*, pages 71–78, March 2007.

[12] H. Bojinov, A. Casal, and T. Hogg. Emergent structures in modular self-reconfigurable robots. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 2, pages 1734 –1741 vol.2, 2000.

[13] Chih-Han Yu and Radhika Nagpal. Self-adapting modular robotics: A generalized distributed consensus framework. *2009 IEEE International Conference on Robotics and Automation*, 67(1):1881–1888, May 2009.

[14] K. J. De Laurentis. Rapid fabrication of a non-assembly robotic hand with embedded components. *Assembly Automation*, 24(4):394–405, 2004.

[15] R.G. Brown and R.C. Brost. A 3-D Modular Gripper Design Tool. *IEEE Transactions on Robotics and Automation*, 15(1):174–186, 1999.

[16] S. Murata and H. Kurokawa. Self-reconfigurable robots. *Robotics & Automation Magazine, IEEE*, 14(1):71–78, 2007.

[17] G. Granosik, M.G. Hansen, and J. Borenstein. The OmniTread serpentine robot for industrial inspection and surveillance. *Industrial Robot: An International Journal*, 32(2):139–148, 2005.

[18] T. Kamegawa, T. Yamasaki, H. Igarashi, and F. Matsuno. Development of the snake-like rescue robot. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 5081–5086. IEEE, 2004.

[19] S. Hirose and M. Mori. Biologically inspired snake-like robots. In *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on*, pages 1–7. IEEE.

[20] J. Ute and K. Ono. Fast and efficient locomotion of a snake robot based on self-excitation principle. In *Advanced Motion Control, 2002. 7th International Workshop on*, pages 532–539. IEEE, 2002.

[21] J. Conradt and P. Varshavskaya. Distributed central pattern generator control for a serpentine robot. In *International Conference on Artificial Neural Networks (ICANN 2003)*. Citeseer, 2003.

[22] A. Crespi, A. Badertscher, A. Guignard, and A.J. Ijspeert. Swimming and crawling with an amphibious snake robot. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3024–3028. IEEE, 2005.

[23] J. Gonzalez-Gomez, H. Zhang, E. Boemo, and J. Zhang. Locomotion capabilities of a modular robot with eight pitch-yaw-connecting modules. In *Proceeding of CLAWAR*, pages 12–14. Citeseer, 2006.

[24] G. Salvietti, H.X. Zhang, J. Gonzalez-Gomez, D. Prattichizzo, and J.W. Zhang. Task Priority Grasping and Locomotion Control of Modular Robot. In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pages 1069–1074, 2010.

[25] C.J.J. Paredis and P.K. Khosla. An approach for mapping kinematic task specifications into a manipulator design. In *Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 556–561. IEEE, 1991.

[26] C.J.J. Paredis and P.K. Khosla. Kinematic design of serial link manipulators from task specifications. *The International Journal of Robotics Research*, 12(3):274, 1993.

[27] T. Matsumaru. Design and control of the modular robot system: TOMMS. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 2, pages 2125–2131. IEEE, 1995.

[28] R. Cohen, MG Lipton, MQ Dai, and B. Benhabib. Conceptual design of a modular robot. *Journal of Mechanical Design*, 114:117, 1992.

[29] R. Hooper and D. Tesar. Computer-aided configuration of modular robotic systems. *Computing & Control Engineering Journal*, 5(3):137–142, 1994.

[30] J.A. Fryer, G.T. McKee, and P.S. Schenker. Configuring robots from modules: an object oriented approach. In *Advanced Robotics, 1997. ICAR'97. Proceedings., 8th International Conference on*, pages 907–912. IEEE, 1997.

[31] I.M. Chen and J.W. Burdick. Enumerating the non-isomorphic assembly configurations of modular robotic systems. *The International Journal of Robotics Research*, 17(7):702, 1998.

[32] T. Fukuda and Y. Kawauchi. Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 662–667. IEEE, 1990.

[33] I.M. Chen and J.W. Burdick. Determining task optimal modular robot assembly configurations. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 1, pages 132–137. IEEE, 1995.

[34] WK Chung, J. Han, Y. Youm, and SH Kim. Task based design of modular robot manipulator using efficient genetic algorithm. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 1, pages 507–512. IEEE, 1997.

[35] O. Chocron and P. Bidaud. Genetic design of 3D modular manipulators. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 1, pages 223–228. IEEE, 1997.

[36] D. Prattichizzo and JC Trinkle. Grasping. *Chapter 28th, Springer Handbook of Robotics, Springer*, 2008.

[37] J.K. Salisbury Jr. *Kinematic and force analysis of articulated hands.* John Wiley & Sons, Inc., 1985.

[38] T.R. Kane, P.W. Likins, and D.A. Levinson. Spacecraft dynamics. *New York, McGraw-Hill Book Co, 1983, 445 p.*, 1, 1983.

[39] MS Ohwovoriole and B. Roth. An extension of screw theory. *Journal of mechanical design*, 103:725–735, 1981.

[40] M.J. Sadigh and H. Ahmadi. Robust control algorithm for safe grasping based on force sensing. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 1279–1284. IEEE, 2009.

[41] A. Bicchi. Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity. *Robotics and Automation, IEEE Transactions on*, 16(6):652–662, 2000.

[42] C. Melchiorri and M. Kaneto. Robot hands. In B. Siciliano and O. Kathib, editors, *Handbook on Robotics*, pages 671–700. Springer, 2008.

[43] T. Baier-Lowenstein and J. Zhang. Learning to Grasp Everyday Objects Using Reinforcement-Learning with Automatic Value Cut-Off. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1551–1556. IEEE, 2007.

[44] D. Prattichizzo, M. Malvezzi, and A. Bicchi. On motion and force controllability of grasping hands with postural synergies. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[45] A. Bicchi. Hands for Dexterous Manipulation and Robust Grasping: A Difficult Road Toward Simplicity. *IEEE Transactions on Robotics*, 16(6):652–662, 2000.

[46] M.R. Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *Robotics and Automation, IEEE Transactions on*, 5(3):269–279, 1989.

[47] C. Ferrari and J. Canny. Planning optimal grasps. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 2290–2295. IEEE, 2002.

[48] R. Diankov and J. Kuffner. Openrave: A planning architecture for autonomous robotics. *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34*, 2008.

[49] A.T. Miller and P.K. Allen. Graspit! a versatile simulator for robotic grasping. *Robotics & Automation Magazine, IEEE*, 11(4):110–122, 2004.

[50] G. Barequet and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '99, pages 82–91, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics.

[51] C. Goldfeder, P.K. Allen, C. Lackner, and R. Pelossof. Grasp planning via decomposition trees. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 10–14. Citeseer, 2007.

[52] M. Ciocarlie, C. Goldfeder, and P. Allen. Dimensionality reduction for hand-independent dexterous robotic grasping. In *Intelligent Robots and Systems,*

2007. IROS 2007. IEEE/RSJ International Conference on, pages 3270–3275. IEEE, 2007.

[53] M. Santello, M. Flanders, and J.F. Soechting. Postural hand synergies for tool use. *The Journal of Neuroscience*, 18(23):10105, 1998.

[54] C. Ferrari and J. Canny. Planning optimal grasps. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 2290–2295. IEEE, 1992.

[55] CS Lovchik and M.A. Diftler. The robonaut hand: A dexterous robot hand for space. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 907–912. IEEE, 1999.

[56] J. Butterfass, G. Hirzinger, S. Knoch, and H. Liu. DLR's multisensory articulated hand. I. Hard-and software architecture. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 3, pages 2081–2086. IEEE, 1998.

[57] J. Aleotti and S. Caselli. Grasp programming by demonstration: A task-based quality measure. In *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*, pages 383–388. IEEE, 2008.

[58] iearobotics. Openmr: Modular robots plug-in for openrave, 2011. [Online; accessed 3-April-2011].