



NATIONAL TECHNICAL UNIVERSITY OF ATHENS  
SCHOOL OF ELECTRICAL & COMPUTER ENGINEERING

SECTOR OF SIGNALS, CONTROL & ROBOTICS

## Object-Frame Impedance Control for Dexterous Robotic Grasping

DIPLOMA THESIS

by

**Konstantinos P. Vasilios**

**Supervisor:** Konstantinos S. Tzafestas  
Assistant Professor, NTUA

Athens, January 2013



## Abstract

This diploma thesis deals with the subject of dexterous robotic manipulation with anthropomorphic robotic hand (comprising opposing kinematic chains). The goal of any dexterous robotic manipulation task is to achieve a desired pose (position / orientation) for the manipulated object by means of the internal coordinated motion of the robotic fingers within the workspace of the hand, along with achieving absolute control of the internal grasping forces.

This diploma thesis starts by presenting a literature survey covering the major issues in the field of dexterous robotic manipulation, outlining the state-of-the-art with particular focus on the control design for robot grasping. This work is based on a custom-adapted dynamic simulation platform, using features from the Open Dynamics Engine (ODE) open source API, in particular: physics-based simulation, multi-body dynamics and collision detection and handling. These features are integrated within a Simulink environment using a custom-modified MEX C++ Function Block. Within this simulation platform, a robot hand has been modeled and dynamically simulated based on the kinematic and dynamic characteristics of the DLR Hand II.

From a theoretical point of view, in this work we apply an integrated passivity-based impedance control scheme to achieve stable robot grasping by properly defining the static and dynamic properties along with the internal forces on the manipulated object. In addition, we deal with the redundant degrees of freedom of the hand, by exploiting the null-space of the fingers task-space.

Analyzing the performance of this robot-grasping control scheme leads to the conclusion that it presents certain drawbacks, namely: steady-state positioning errors, inconsistencies regarding the definition of a decoupled stiffness matrix, as well as increased risk of potential contact slippage for specific object geometries. In this diploma thesis, we propose an extension of this passivity-based object-level impedance control scheme, aiming to control more efficiently the internal grasping forces, using information based on the local object surface geometry properties. The goal is to reduce steady-state errors, as well as to mitigate the effect of coupling between independent degrees of freedom, in order to produce a more consistent object-level grasping stiffness matrix and to achieve better control of the contact forces inside the friction cones constraints, thus reducing slippage possibility. Finally, this impedance control scheme is further extended by introducing an active gravity-compensation term, aiming to minimize any static errors that may be due to the effect of the manipulated object's weight. Simulation results, obtained by conducting extensive trials within the dynamic simulation platform described above, demonstrate the efficacy of the proposed robot grasping control scheme and the improved object-level impedance characteristics achieved.

## Key Words

Dexterous Robotic Grasping, passivity-based dynamic control, Grasp Force Optimization, robot impedance control, Dynamic Modeling and Simulation, internal force optimization, gravity compensation

# Acknowledgements

First of all, I would like to thank my supervising professor, Prof. Konstantinos Tzafestas, for the trust he placed in me regarding this challenging yet fascinating topic in robotics. His support and guidance, often under difficult circumstances and time constraints, were of crucial importance for the successful completion of this work.

I would also like to particularly thank my parents, Panos and Zoi, whose moral and practical support throughout my years of study enabled me to remain fully focused on the learning process.

# Contents

<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Robotics Science . . . . .	1
1.2 Dexterous Robotic Manipulation . . . . .	7
1.3 Document Organization . . . . .	10
<b>2 Dexterous Robotic Grasping Control: A Survey</b>	<b>12</b>
2.1 Essential Classical Definitions . . . . .	13
2.2 System Description . . . . .	17
2.2.1 Modeling . . . . .	17
2.2.2 Sensory Synthesis . . . . .	18
2.3 Robotic Grasping Control Methodologies . . . . .	20
2.3.1 Force Control . . . . .	20
2.3.2 Object-Level Stiffness Control . . . . .	22
2.3.3 Object-Level Stiffness Control with Redundant Kinematic Configurations . . . . .	23
2.3.4 Admittance Control . . . . .	24
2.3.5 Hybrid Dynamic Control of Robotic Grasping System . . . . .	25
2.3.6 Grasping Force Optimization (GFO) . . . . .	26
2.3.7 Robotic Hand Preshaping . . . . .	27
2.3.8 Other Techniques . . . . .	28
2.3.9 Stability Issues . . . . .	28
<b>3 Modeling &amp; Control of Robotic Grasping: Theoretical Background</b>	<b>31</b>
3.1 Modeling of the Robotic Hand . . . . .	31
3.1.1 Kinematic Model of the Robotic Hand . . . . .	31
3.1.2 Dynamic Model of the Robotic Hand . . . . .	34
3.1.3 Basic Properties of the Robotic Hand Dynamic Model . . . . .	35
3.1.4 Passivity-Based Robotic Control . . . . .	36
3.2 Static Grasp Analysis . . . . .	37

3.2.1	Static Equilibrium & Grasp Force Transformation Matrix . . . . .	37
3.3	Passivity-Based Object-Level Impedance Control . . . . .	39
3.3.1	Virtual Object Frame . . . . .	39
3.3.2	Object-Level Control Law . . . . .	41
3.3.3	Rotational Stiffness . . . . .	43
3.3.4	Translational Stiffness . . . . .	44
3.3.5	Connection Stiffness . . . . .	44
3.3.6	Control of Redundant Degrees of Freedom . . . . .	45
3.3.7	Damping-Term Design . . . . .	47
3.4	Extensions and Modifications of the Object-Level Impedance Controller	49
3.4.1	Internal-Force Control Based on Surface Characteristics (IPC–IF)	49
3.4.2	Gravity Compensation . . . . .	51
<b>4</b>	<b>Implementation &amp; Simulation Results</b>	<b>53</b>
4.1	Construction of the simulation environment . . . . .	53
4.1.1	DLR Hand 2 . . . . .	54
4.1.2	Open Dynamics Engine (ODE) . . . . .	55
4.1.3	Open Dynamics Engine S-Function Block – Robotic hand simulation . . . . .	56
4.1.4	Overall implementation structure . . . . .	57
4.2	IPC controller simulation results . . . . .	59
4.2.1	Step response – translational and rotational motion . . . . .	59
4.2.2	Stiffness measurement . . . . .	65
4.3	Simulation results of the IPC–IF robotic grasp controller with internal force compensation . . . . .	69
4.3.1	Step response – translational and rotational motion . . . . .	69
4.3.2	Stiffness measurement . . . . .	74
4.4	Comparison of controllers IPC and IPC–IF . . . . .	77
4.5	Gravitational compensation . . . . .	78
<b>5</b>	<b>Conclusions &amp; Future Work</b>	<b>80</b>
5.1	Summary . . . . .	80
5.2	Extensions – Future Directions . . . . .	82
<b>Bibliography</b>		<b>83</b>

# List of Figures

1.1	Intelligent Robotic Systems: Starting from left to right and row by row, AlphaDog (DARPA), Atlas (DARPA), Twendy-One (WASEDA University Sugano Laboratory TWENDY team), Seagle (FESTO), Asimo (Honda), Opportunity (NASA), PR2 (Willow Garage), Robonaut (NASA), Industrial Arm (KUKA), UAV Drone, DaVinci, Google Autonomous Car	5
1.2	Order of magnitude for the number of inputs-outputs in the human system . . . . .	7
1.3	Dexterous Anthropomorphic Robotic Hands: Starting from left to right and row by row: DLR Hand (German Aerospace Center), DLR Hand 2 (German Aerospace Center), Twendy-one Hand (WASEDA University Sugano Laboratory TWENDY team), Robonaut Hand Schematic (NASA), Shadow Robot Hand, FESTO ExoHand . . . . .	9
2.1	Closure types for planar motion. (a) Passive Form Closure. (b) Passive Force Closure. (c) Active Closure. (d) Hybrid Active/Passive Closure [4]	16
2.2	Hybrid System HDS [16] . . . . .	18
2.3	Bank of Impedance Controllers [16] . . . . .	26
3.1	Denavit–Hartenbeg Parameters [31] . . . . .	32
3.2	Kinematic structure of an anthropomorphic 4-DOF finger . . . . .	33
3.3	Schematic diagram of an object grasped by an $N$ -finger hand with point contacts and friction [16] . . . . .	38
3.4	Virtual frame construction [27] . . . . .	40
3.5	Intrinsically Passive Control scheme [29] . . . . .	42
3.6	The fingertips $x = [x_1, x_2, x_3, x_4]^T$ hold a spherical/cylindrical object using the virtual springs that define the control, as described in (3.48) . . . . .	42
3.7	Large tangential force with respect to the object surface at fingertips $x_2$ and $x_3$ due to the spherically symmetric connection stiffnesses $K_{Conn2}$ and $K_{Conn4}$ . . . . .	49
3.8	Intrinsically Passive Control with internal-force control by applying normal pressure on the surfaces defined by the contacts (IPC–IF). . . . .	50
3.9	Internal-force control with normal pressure on the object surfaces in combination with gravity-compensation terms. . . . .	52

4.1	Top-level Simulink implementation view including the system model, the controller, and the auxiliary tools. . . . .	58
4.2	Overall view of the Simulink controller structure. . . . .	58
4.3	View of the Simulink internal stiffness terms (rotational and translational). . . . .	59
4.4	General view of the initial grasp configuration with a spherical object under manipulation. . . . .	60
4.5	Top view – final pose after rotational displacement about the $z$ axis (rotation 0.6 rad). . . . .	60
4.6	Top view – displacement along the $x_o$ axis (translation 40 cm). . . . .	61
4.7	Impedance torque $m_O$ during rotation about the $z$ axis (rotation 0.6 rad). . . . .	61
4.8	Impedance force $f_O$ during rotation about the $z$ axis (rotation 0.6 rad). . . . .	61
4.9	Fingertip-to-virtual-frame connection force $f_{Conn}$ during rotation about the $z$ axis (rotation 0.6 rad). . . . .	62
4.10	Applied joint torques of fingers 2,3,4,1–thumb, during rotational motion about the $z$ axis (rotation 0.6 rad). . . . .	62
4.11	Object position $x_O$ during displacement along the $x_o$ axis (translation 40 cm). . . . .	63
4.12	Impedance torque $m_O$ during displacement along the $x_o$ axis (translation 40 cm). . . . .	63
4.13	Impedance force $f_O$ during displacement along the $x_o$ axis (translation 40 cm). . . . .	63
4.14	Fingertip-to-virtual-frame connection force $f_{Conn}$ during displacement along the $x_o$ axis (translation 40 cm). . . . .	64
4.15	Applied joint torques of fingers 2,3,4,1–thumb, during translational motion $x$ , 0.40m, as produced by the control law. . . . .	64
4.16	Virtual frame position $x_o$ under applied forces along the $x$ , $y$ , $z$ axes, respectively. . . . .	66
4.17	Virtual frame quaternion vector $e_b$ under applied forces along the $x$ , $y$ , $z$ axes, respectively. . . . .	66
4.18	Object position $x_o$ after applying forces along the $x$ , $y$ , $z$ axes of frame $H_o$ , respectively. . . . .	67
4.19	Virtual frame position $x_o$ under applied torques along the $x$ , $y$ , $z$ axes, respectively. . . . .	67
4.20	Virtual frame quaternion vector $e_b$ under applied torques along the $x$ , $y$ , $z$ axes, respectively. . . . .	68
4.21	Object quaternion after applying torques along the $x$ , $y$ , $z$ axes of frame $H_o$ , respectively. . . . .	68
4.22	Impedance torque $m_O$ during rotation about the $z$ axis (rotation 0.6 rad), with the IPC–IF controller. . . . .	69
4.23	Impedance force $f_O$ during rotation about the $z$ axis (rotation 0.6 rad), with the IPC–IF controller. . . . .	70

4.24 Magnitude of internal forces $f_{sn}$ during rotation about the $z$ axis (rotation 0.6 rad), with the IPC–IF controller. . . . .	70
4.25 Applied joint torques of fingers 2,3,4,1–thumb, during rotational motion about the $z$ axis (rotation 0.6 rad), with the IPC–IF controller. . . . .	71
4.26 Object position $x_O$ during displacement along the $x_o$ axis (translation 40 cm), with the IPC–IF controller. . . . .	71
4.27 Object impedance force $f_O$ along the $x_o$ axis (translation 40 cm), with the IPC–IF controller. . . . .	72
4.28 Object impedance torque $m_O$ along the $x_o$ axis (translation 40 cm), with the IPC–IF controller. . . . .	72
4.29 Magnitude of internal forces $f_{sn}$ during displacement along the $x_o$ axis (translation 40 cm), with the IPC–IF controller. . . . .	72
4.30 Applied joint torques of fingers 2,3,4,1–thumb, during displacement along the $x_o$ axis (translation 40 cm), with the IPC–IF controller. . . . .	73
4.31 Virtual frame position $x_o$ under applied forces along the $x$ , $y$ , $z$ axes, respectively, with the IPC–IF controller. . . . .	74
4.32 Virtual frame quaternion vector $e_b$ under applied forces along the $x$ , $y$ , $z$ axes, respectively, with the IPC–IF controller. . . . .	75
4.33 Object position $x_o$ after applying forces along the $x$ , $y$ , $z$ axes of frame $H_o$ , respectively, with the IPC–IF controller. . . . .	75
4.34 Virtual frame position $x_o$ under applied torques along the $x$ , $y$ , $z$ axes, respectively, with the IPC–IF controller. . . . .	76
4.35 Virtual frame quaternion vector $e_b$ under applied torques along the $x$ , $y$ , $z$ axes, respectively, with the IPC–IF controller. . . . .	76
4.36 Object quaternion after applying torques along the $x$ , $y$ , $z$ axes of frame $H_o$ , respectively, with the IPC–IF controller. . . . .	77
4.37 Tilt – steady-state position error in the presence of gravity. Introduction of the gravitational compensation term with recovery of position–orientation. . . . .	78
4.38 Object position $xO$ during application of the gravity term at 0.3sec and the compensating term at 1.3sec. . . . .	78
4.39 Translational impedance force $f_O$ during application of the gravity term at 0.3sec and the compensating term at 1.3sec. . . . .	78
4.40 Rotational impedance $m_O$ during application of the gravity term at 0.3sec and the compensating term at 1.3sec. . . . .	79
4.41 Quaternion vector $e_b$ during application of the gravity term at 0.3sec and the compensating term at 1.3sec. . . . .	79

# List of Tables

3.1	Denavit–Hartenberg parameter table for an anthropomorphic 4-DOF finger	32
4.1	Table of basic geometric–inertial characteristics of the DLR Hand 2 . . .	55
4.2	ODE parameter values . . . . .	56
4.3	Table of basic geometric–inertial characteristics of the simulated robotic hand . . . . .	57
4.4	Mean square errors ( $\sum_{i=1}^N e^2/N$ ) with respect to the apparent translational stiffness of the robotic grasp. . . . .	77
4.5	Mean square errors ( $\sum_{i=1}^N e^2/N$ ) with respect to the apparent rotational stiffness of the robotic grasp. . . . .	77

# Chapter 1

## Introduction

In this introductory chapter, we first attempt to formulate a definition of the science of robotics and robotic systems. We then provide some background elements, proceeding with a basic analysis of the current state and developmental goals of robotics. A natural connection in the study of almost any robotic system—especially those aimed at autonomous, adaptive-intelligent action—are biological systems, and particularly humans themselves. Immediately after, we focus our attention on the branch of robotics that deals with dexterous manipulation, which is also the subject of this diploma thesis, mentioning some basic characteristics of the efforts made so far to construct such devices. Finally, we present the structure of the thesis on a chapter-by-chapter basis, along with their corresponding summaries.

### 1.1 Robotics Science

An elegant, general, and abstract definition of robotics science is the following.

“Robotics is defined as the science that studies the intelligent connection between perception and action [1]”

The word robot originates from the Czech word *roboťa*, which is used to describe boring or laborious, possibly compulsory, work.

Various definitions have been proposed for robotic systems. We present some of the most characteristic ones.

“A robot is defined as a mechanical or virtual agent, usually a complex electromechanical device, which is driven by programmable elements”

*Wikipedia*

“A machine capable of carrying out a complex series of actions automatically, programmable by computational elements”

“A machine resembling a human being, capable of replicating certain specific human movements and abilities, performing these automatically”

“Refers to a person who behaves mechanically in a non-emotional manner”

*Google Dictionary*

“A machine, sometimes resembling a human being, capable of performing a range of often complex human tasks based on commands or prior programming”

“A machine or device that acts automatically or through remote control”

*American Heritage Dictionary*

“An automated machine programmed for the execution of specific mechanical tasks automatically or under the guidance of a person”

*Collins English Dictionary*

In the case of a virtual agent, the most commonly used term is that of a *bot*.

We attempt an abstract synthesis of the above definitions with the aim of a systemic approach.

The science of robotics deals with the study of systems which, by combining sensory elements, computational capability—intelligence, and actuators—kinematic configurations, are able to acquire information from the environment, process it, and ultimately act upon it in a manner that serves the purpose for which they were designed.

The field of robotics science encompasses an extremely broad spectrum of individual scientific fields, with the most characteristic being electronics, computer science, cognitive science (mainly the artificial intelligence component), mechatronics, nanotechnology, biomechanics, and others. The successful integration of these for the synthesis of a successful robotic application constitutes a particular challenge.

The breadth of the application field, the multiple scientific disciplines, and perhaps the youth of robotics science are likely some of the reasons that justify the absence of a strict, universally accepted definition.

To date, the most extensive and successful robotics applications are limited to industrial production lines, performing repetitive tasks in strictly structured and predictable environments. It is even considered that the relevant technologies of industrial robots have reached a mature state [2]. In most of these applications, position control techniques are used exclusively, thus giving a clear indication of industry’s tendency to prefer traditional, classical control techniques.

However, the challenge for robotics has been and remains successful penetration into the real world with systems capable of intelligent autonomous action where the

environment is unstructured and the occurrence of unpredictable events prevails. Such an evolution would also have very significant benefits in the industrial sector where robotic systems operate, as it is estimated that the cost of configuring the workspace of a robotic system is four times greater than that of installing the robot itself.

Such an evolution presupposes enhanced perception capabilities, cognitive capabilities, as well as kinematic and dynamic actuation capabilities in the surrounding space. Many efforts have been made to develop such systems by governmental and private entities worldwide, mainly from advanced countries; however, in no case do they constitute part of daily life for societies to the degree that would be expected. A very characteristic example is that of the nuclear accident at Fukushima, Japan, where although a particularly advanced country in the field of robotics was expected to have robotic systems for disaster response, in the end it was workers who were exposed to this dangerous environment to control the crisis. This lag raises serious questions about which paths and directions robotics should follow in the coming years.

Of considerable interest on these topics is the new DARPA Robotics Challenge (Defense Advanced Research Projects Agency of the United States), which poses as a challenge the construction of humanoid robotic systems capable of action in rescue scenarios in dangerous, degraded human-made environments. If we consider the success of DARPA's previous corresponding challenge for the construction of self-driving vehicles (in the state of Nevada, the first license was already issued for Google's self-driving vehicle), then developments for autonomous robotic systems are expected to be rapid.

In recent years, certain trends are being formed that reinforce the tendency toward the development of advanced robotics systems capable of autonomous action in the real world. Smartphones, with the tremendous development they have shown in recent years, have dramatically reduced the cost of the individual microsystems they incorporate, providing cheap and reliable solutions in integrated circuits for the provision of abundant computational power as well as large volumes of data from a wide range of sensors such as gyroscopes, accelerometers, proximity sensors, touch sensors, visible electromagnetic spectrum sensors, Earth's magnetic field detection sensors, etc., thus providing capabilities on a new scale for the development of robotic systems[3].

At the same time, cognitive science has begun to show significant tangible progress in recent years, with particular development in the area of artificial intelligence through the development of impressive applications (see IBM Blue Gene, Watson, Google Car-Stanford SUV, MS Kinect, etc.). Despite this impressive development in the field of artificial intelligence, we must note that these systems are still far from being characterized as truly intelligent based on corresponding biological models. Today it is commonly accepted in the scientific community that the artificial intelligence component essentially constitutes the bottleneck in the development of integrated autonomous robotic systems [4].

Significant progress can also be found in the field of materials with the construction of efficient joints, actuators, as well as capable soft-flexible elements (soft robotics).

Combining these elements, many were those who expected an explosion of innovative activity in this field during the current decade, similar to that of information technology in the 1990s, investing in related start-up companies with the goal of widely expanding the market for robotic systems, as a result of which over 80 related companies are currently operating in Silicon Valley with forecasts for even greater growth in the near future [5, 6].

In Figure 1.1, some of the most impressive examples of recent robotic systems are presented indicatively, which also outline the trends that are very likely to drive future developments.

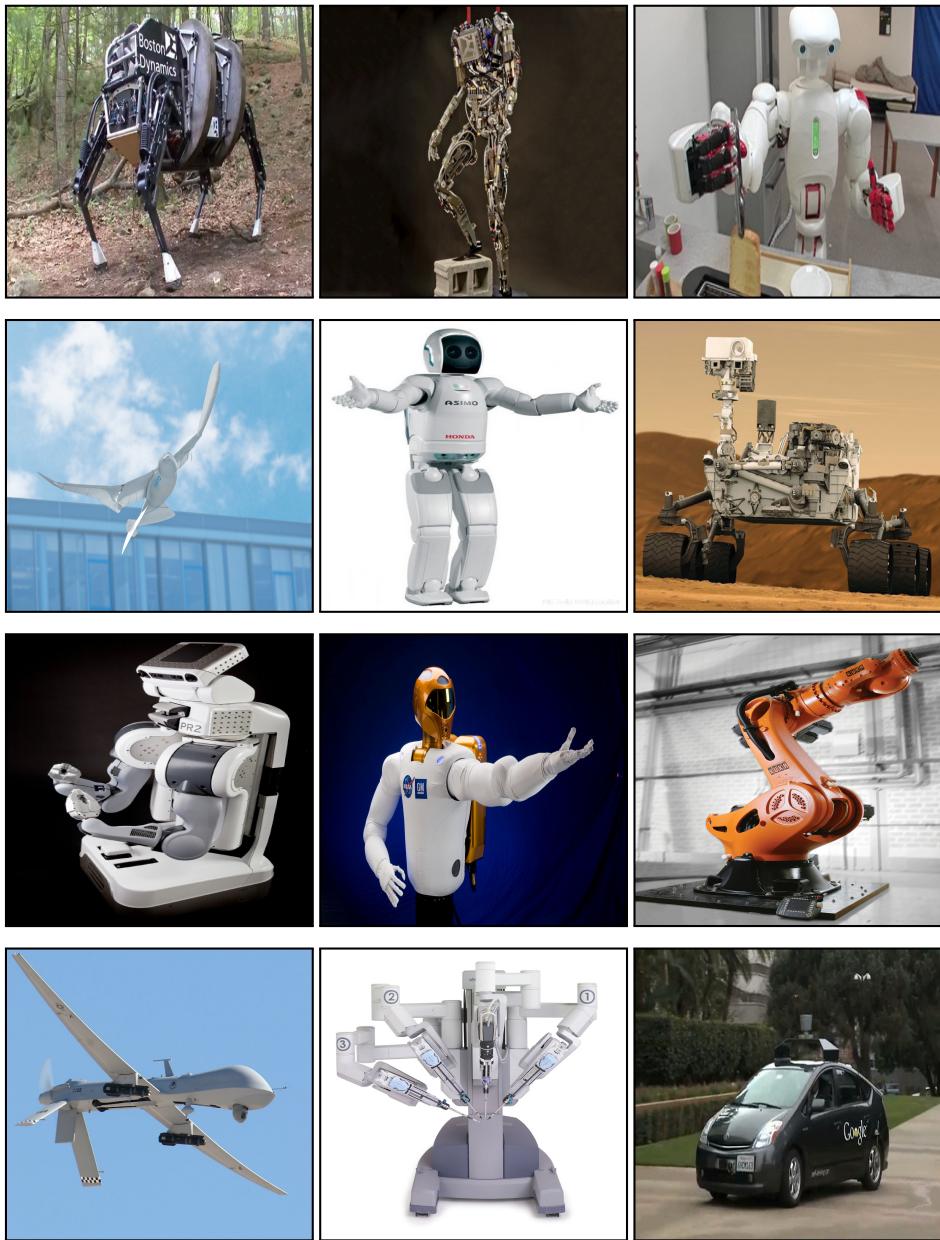


Figure 1.1: Intelligent Robotic Systems: Starting from left to right and row by row, AlphaDog (DARPA), Atlas (DARPA), Twendy-One (WASEDA University Sugano Laboratory TWENDY team), Seagle (FESTO), Asimo (Honda), Opportunity (NASA), PR2 (Willow Garage), Robonaut (NASA), Industrial Arm (KUKA), UAV Drone, DaVinci, Google Autonomous Car

## **Contribution of Robotic Systems**

The general sectors of activity for robotic systems are mainly:

- Industrial applications, mainly in production processes in heavy industry.
- Aerospace
- Medicine, with particular emphasis on robotic surgery.
- Prosthetics, with the construction of artificial limbs as well as patient rehabilitation procedures.
- Elderly care.
- Action in dangerous-degraded environments.
- Automation of daily tasks in workplaces and residences.
- Unmanned warfare.
- Entertainment and amusement applications.

## **Biological Model**

Biological systems constitute the most successful examples of such systems, from the humblest microorganisms to advanced mammals. The fractal structure-morphology and the resulting dynamics of these systems, given a process of biological evolution spanning hundreds of millions of years that leads to optimization of energy management through metabolic processes, correspondingly gives us a picture of the complexity of the unstructured natural environment as well as the challenges of robotics science in constructing intelligent autonomous or semi-autonomous systems. A basic model and source of inspiration in this whole endeavor is, naturally, the human being itself.

Making a simplifying systemic approach of input-output relationship in the “human system,” the following interesting elements emerge that reveal, to only a small degree, its complexity.

Approximately, for human sensory nerve endings, we have a total of 300,000,000 sensory inputs—nerve endings, of which: 120,000,000 Rods and 6,000,000 Cones in the retina of each eye. 40,000,000 nerve endings for smell. 3,500,000 nerve endings for touch. 15,000-20,000 Auditory nerve receptors in each ear, and 10,000 taste receptors.

The output of the system is essentially expressed through the musculoskeletal system, which acts upon the environment. 270 bones and 650 muscle fibers (up to 850 depending on how the count is made) undertake to carry out the particularly complex game of Newtonian dynamics within the framework of interaction with the environment.

We can say overall for the system that it receives 300,000,000 inputs and has only 800 outputs, with the dominant sensory set being that of vision [7].

Similarly, the problem of designing an appropriate intelligent robotic controller based on existing technology is summarized by the expression “pixels to torques.”

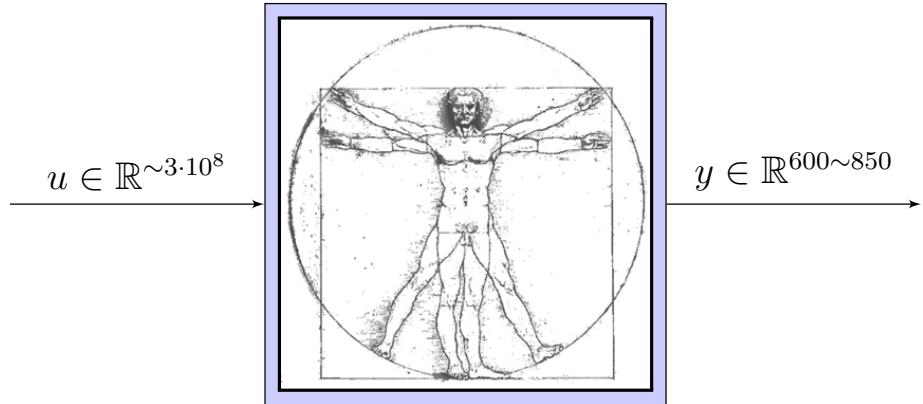


Figure 1.2: Order of magnitude for the number of inputs-outputs in the human system

## 1.2 Dexterous Robotic Manipulation

The construction of robotic hands has been one of the most important areas of research since the beginning of robotics science. This is logical, as the manipulation of environmental elements through direct exchange of forces constitutes one of the fundamentally sought-after robotic functions, being also one of the most basic, inseparable prerequisites for the action of robotic systems—primarily humanoids—in unstructured environments.

In this effort, it is impossible to overlook the corresponding biological model, which is none other than the human hand itself. The effectiveness of the biological model becomes immediately apparent through the obvious observation that the absolute totality of human-generated activity is the result of the action of hand and mind. From an anthropological perspective, it is proven that mechanical dexterity itself is one of the basic causes that triggered the development of the human mind [8]. The dexterity of the human hand is still today quite ahead of any corresponding mechanical construction and will probably maintain this primacy for much longer. While in individual capabilities and technical characteristics, such as speed and durability, some robotic hands appear to be superior, it is the breadth of capabilities of the human hand to successfully address an impressively large range of applications that essentially makes it a design standard. The answer, however, to whether the designer should pursue either anthropomorphism or some optimal design with respect to specific parameters depends on the respective application and its requirements [8].

A particularly characteristic example of this is the grasping technology of the robotic system on the back of NASA’s Space Shuttle, where although a classical scheme of

opposing kinematic chains had initially been proposed, following anthropomorphic models, a solution was ultimately preferred that grasps objects in space through an opening-closing diaphragm [9]. Another interesting case of an alternative manipulation proposal uses a spherical elastic rubbery bag filled with organic material. This bag, once it comes into contact with an object, compliantly takes the shape of the object, and then through suction, the bag is compressed, resulting in it ultimately embracing the object to an absolute degree, ensuring a very good mechanical connection [10].

A basic element of anthropomorphism in design that is being applied more and more in dexterous robotic hands is the use of soft materials at the fingertips, with the goal of enhancing gentle compliant behavior (compliant behavior) of high bandwidth, which makes the device more capable of robust grasping and manipulation [11] compared to absolutely rigid elements. A hindrance in this field is the lack of strict mathematical formalism for describing the dynamic behavior of flexible robotic systems [12], which is also the main reason why any control techniques in this field constitute extensions of classical ones for rigid robotic hands [13].

A basic element that plays a decisive role in design is whether the robotic hand is intended to be adapted to an already existing robotic kinematic arm configuration, in place of the end effector, or whether it constitutes part of a fully customized mechanical design of hand-forearm. In the first case, the entire mechanical configuration (actuators, motion transmission elements) is placed either inside the hand or in a special enclosure placed externally above the palm, a space that does not constitute the workspace of the system. In the case of a fully customized design, clearly greater freedom is given regarding the mechanical configuration. In high-performance systems, the anthropomorphic model is preferred with placement of actuators in the link between the wrist and elbow and transmission of motion through tendons. This configuration places the largest part of the mechanical elements—and thus the largest part of the mass of the configuration—close to the forearm and main body, with positive effects on the overall dynamic characteristics of the whole system as well as of the hand itself, ultimately having lighter links-joints.

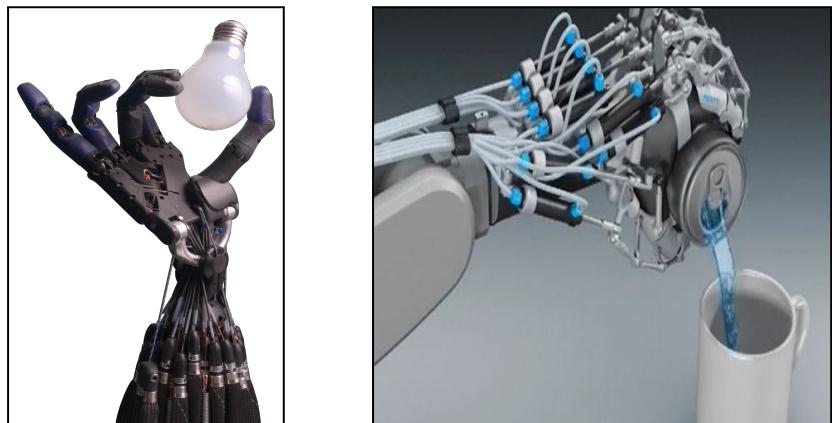
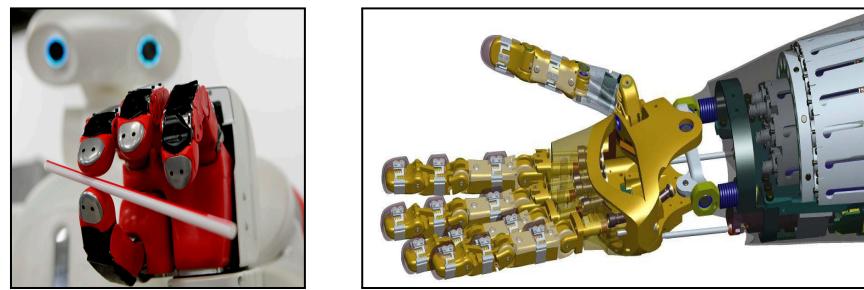


Figure 1.3: Dexterous Anthropomorphic Robotic Hands: Starting from left to right and row by row: DLR Hand (German Aerospace Center), DLR Hand 2 (German Aerospace Center), Twendy-one Hand (WASEDA University Sugano Laboratory TWENDY team), Robonaut Hand Schematic (NASA), Shadow Robot Hand, FESTO ExoHand

## 1.3 Document Organization

- In *Chapter 2*, a literature review of control techniques for dexterous robotic grasping is presented. Initially, the overall manipulation problem is decomposed into its constituent sub-problems. Basic definitions are then given, introducing a first stage of formalism as developed for the topic of robotic grasping in the relevant literature. Immediately after, the methods of describing the robotic grasping system through mathematical modeling and its corresponding techniques follow, as well as the description of the sensory synthesis applied in such configurations. Once the description of modeling and sensory elements has been given, some dominant control technique schemes for robotic grasping systems are presented.
- In *Chapter 3*, we analyze the theoretical background upon which the implementation within the framework of this work is based. A hierarchically structured approach is attempted, which starts from the analytical description of the kinematic-dynamic model of the robotic finger and, by extension, the hand, according to the kinematic-dynamic models of the DLR Hand 2 robotic hand. Some fundamental properties of the dynamic model based on passive systems theory are analyzed, upon which the control law is based. The static analysis of the grasp and the definition of the grasp matrix are structured, as well as the corresponding least squares problem solution. The definition of the virtual frame of the robotic hand is introduced, and based on this, the corresponding potential functions are structured, which define the stiffness at the object level in all 6 Cartesian degrees of motion as well as in the internal force space. The dynamic technique for handling the redundant degrees of freedom of the robot is then presented, as well as the design of the damping term. The basic problems—disadvantages of control in the internal force space through the definition of stiffness with respect to the virtual frame are identified, and a technique based on the geometric characteristics of the object is proposed. Finally, a suitable extension of the linear and rotational stiffness terms is proposed and analyzed, with the aim of gravity compensation.
- In *Chapter 4*, the methodology applied for the implementation, in a structured simulation platform, of the control techniques studied within the framework of this work is described in detail, and the results of the conducted tests are presented. Some basic elements are given for the individual structural components of the implementation, which are: the model-prototype robotic hand DLR Hand 2, the software—ODE dynamics simulation API which is integrated into a Simulink environment through a MEX C++ S-Function Block. The simulation results follow, which include the step response of the system for rotational and translational motion as well as the measurement of stiffness through the application of forces-torques on the manipulated object. These measurements are initially performed for the simple Intrinsic Passive Control (IPC) algorithm

and then repeated for the internal force control scheme based on object geometry (IPC—IF), in order to draw appropriate conclusions about their comparative performance. The results demonstrate the effectiveness of the proposed technique.

- *Chapter 5* constitutes the epilogue of the work, where a summary is made, general conclusions are drawn, and an exploration of possible future extensions is conducted.

# Chapter 2

## Dexterous Robotic Grasping Control: A Survey

In this chapter, we attempt a general literature review, mainly on the control techniques for dexterous robotic grasping, as well as on certain specific topics.

### Problem Decomposition

In order to analyze the complex subject of dexterous manipulation, it can be decomposed and examined separately based on the following semi-autonomous successive tasks.

1. Approach of the target object-surface, a phase during which there is no constraint on motion and appropriate trajectory planning for approaching the contact points is performed.
2. Contact establishment and thus a specific constraint on motion through force exchange. In this phase, we desire control with respect to the interaction forces-torques as well as with respect to the position of the end effector.
3. Manipulation process. The manipulation process is achieved by simultaneously controlling the interaction forces as well as the corresponding position of the contact points in space.

These individual problems are interconnected, as for example, the selection of contact points greatly affects the subsequent manipulation capability. Nevertheless, they can also be examined independently by developing control techniques for each of these sub-problems separately.

In this diploma thesis, we deal mainly with the last part, that of dexterous manipulation of the object once successful approach of the contact points has been achieved.

For the development of the analysis as well as for any other description, we first provide some fundamental classical definitions regarding the characteristics and properties of the grasping system.

## 2.1 Essential Classical Definitions

### Dexterity

Refers to the ability to change the position and orientation of the manipulated object from an initial reference configuration in space to another arbitrarily defined one within the workspace of the fingers [13, 8].

It constitutes a fairly broad concept that concerns the simultaneous capability as well as stability in performing movements of the manipulated object by the palm and fingers.

### Grasp Robustness

The ability to maintain the object stable regardless of disturbances of any form (such as unexpected forces, incorrect estimations of object characteristics) while at the same time the internal grasping forces (internal/grip forces) are constrained so as not to cause damage to the overall system.

### Human Operability

The capability for easy and safe interaction in an environment with human physical presence.

In practical applications, these basic criteria may be difficult to coexist, requiring the designer to make compromise decisions [8].

### Grasp Closure – Contact Modeling

A grasp is called closed if and only if it is in a state of equilibrium for any arbitrary vector of generalized external forces acting on the object [14].

The balancing forces that resist the movement of the object are produced through direct contact of the robotic device with the object. The criticality of selecting appropriate contact points as well as their dynamics is decisive for the behavior of the entire system. Contact modeling is extremely important in the analysis of the robotic grasp-object system. Generally, in the literature, simplifying assumptions are usually made in contact modeling, considering them as point contacts with friction based on the Coulomb model. A basic classical categorization regarding contact models is the following.

- **Point Contact with or without friction – *Hard Finger*.** Exerts force in the direction of the object (locally perpendicular to the surface of the object at the contact points) and in the case of friction, tangentially as well.
- **Soft contact – *Soft Finger*.** Can additionally exert contact torque about the normal axis at the contact point.

Very important properties that are taken into account in the contact model are the visco-elastic behavior (visco-elastic behaviour, rigid, isotropically elastic), the rolling and sliding conditions, i.e., the static and dynamic terms of the model, as well as whether the bodies in contact exhibit rolling (*rolling contact*) or sliding (*sliding contact*). On these topics, the classical works of Salisbury and Mason, professor researchers at MIT, are relevant, who with their now classic works in the 1980s laid the foundations for the analysis and design of dexterous robotic grasps. Salisbury first showed that the theoretically minimum number of degrees of freedom necessary to achieve dexterity in a hand with rigid fingers, without rolling and sliding phenomena, is 9, with each finger having at least 3 DOFs. Correspondingly, he also proceeded with the design-construction of a robotic hand with these characteristics.

For increasing flexibility in manipulation operation, several researchers introduced redundant degrees of freedom in their designs, with the most classic addition being that of one additional but coupled degree of freedom per finger at the distal interphalangeal joint. This approach mimics the human model and essentially the degrees of freedom remain 3.

A basic prerequisite for the penetration of dexterous manipulation systems into the real world is the reduction of complexity at every level of implementation [8]. The particularly complex nature of the individual mechanical parts comes into direct conflict with design criteria that impose a high degree of reliability, low cost, and low weight. The mechanical complexity of robotic hand construction is characteristically reflected in the number of actuators used, which starts from 9 and can reach 32 or more.

On the topic of optimized design, it is very important to note that the required number of degrees of freedom to achieve dexterity is absolutely connected to the initial assumptions we make about the contact model. For example, under the assumption that the contacts are “soft-finger,” the minimum degrees of freedom for each finger to achieve dexterity result in 4 [8].

Thus, simpler configurations can be defined from the mechanical level to the control level that can be equally dexterous in manipulation by exploiting alternative techniques such as *Regrasping & Finger Gaiting* or techniques that exploit rolling and sliding phenomena of contacts, *Rolling & Sliding* [8]. For these techniques, a major obstacle is the difficulty of establishing closed mathematical descriptions for these phenomena to a degree that would make the application of a suitable control technique feasible. Particularly for the phenomenon of rolling between a soft micro-finger and a rigid object, the Lagrangian dynamic model of the system has been defined only for two

fingers with 2 DOFs in 2 dimensions [12]. We will refer to these techniques in more detail in the next section.

Based on this contact modeling as analyzed above, the corresponding types of grasp closure that can arise are defined.

### **Form Closure**

Refers to the ability of the grasp to prevent object movements based on constraints created by unilateral, frictionless contacts [8, 14, 13].

This problem, directly related to the design of mechanical devices for immobilizing objects in space for assembly and manufacturing processes, has been studied since the 19th century with the first significant theoretical results from the father of kinematics of mechanical systems, as Franz Reuleaux is characteristically called. Theoretical study shows that at least 4 frictionless contacts are needed to immobilize an object in the plane and 7 for 3D space. The form closure problem is also presented in reverse form, that of analysis, where given the grasp, it is examined whether there are available degrees of freedom for the object, and if so, in which direction. An extension of the classical definition of form closure, under the term “immobilization problem,” takes into account 2nd order phenomena that develop due to the relative morphology of the surface of the two bodies, ultimately providing greater accuracy in the analysis [8].

### **Force Closure**

Force closure determines the ability of the grasp to resist any external forces and usually refers to point contacts with friction. In the latter case, the grasp can withstand any force or applied torque given the existence of a sufficiently large normal force at the contact point [8, 14, 13].

Also, grasps that are force-closed, depending on whether the constraint points of object motion are active elements (i.e., whether they have the capability of controlled motion or not), can be characterized as *Active (Active Force Closure)* or *Passive (Passive Form Closure)* respectively. In case there are active and passive elements (such as a constraint from a kinetically inert element of the environment) acting in different directions, then the force-closed grasp is characterized as *hybrid (Hybrid Force Closure)* [4]. By definition, form closure arises only under passive constraining elements.

### **Static Grasp Equilibrium**

A grasp can be in equilibrium when the Convex Hull composed of the vectors of generalized forces exerted by the fingers on the object includes the zero point of the vector basis.

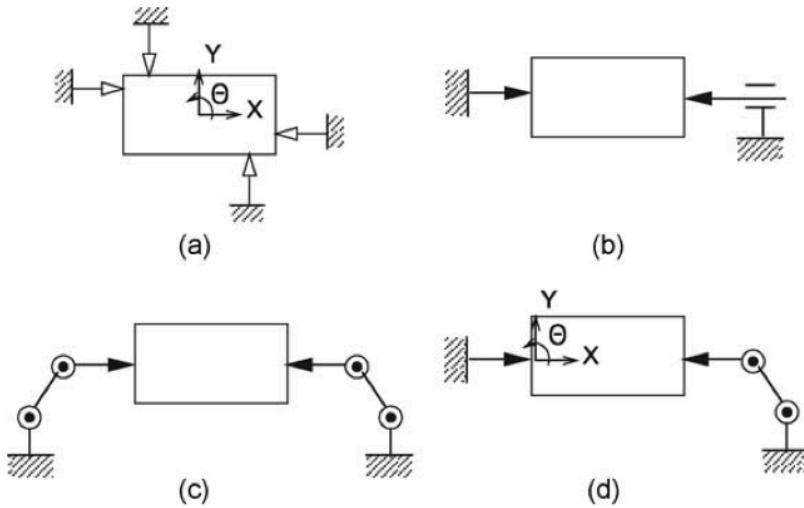


Figure 2.1: Closure types for planar motion. (a) Passive Form Closure. (b) Passive Force Closure. (c) Active Closure. (d) Hybrid Active/Passive Closure [4]

### Fingertip Grasp

The grasp is able to resist any arbitrary external force with the only contact points being those between fingertips and object.

### Power Grasp (or enveloping grasp)

Refers to the type of grasp that is used for stabilizing-fixing objects using many contact points-surfaces to maximize capability for large loads and their stable attachment.

### Force Distribution Problem

A very fundamental problem in dexterous robotic manipulation is the selection of appropriate grasping forces so as to avoid, or minimize, the risk of object slippage. The grasping forces, or otherwise called internal forces, lie in the null space of the grasp matrix. The contact forces that are not directly internal affect the equilibrium of the object and are referred to as manipulation forces. The problem of selecting joint torques so as to create appropriate manipulation forces for the task, while at the same time having internal forces that guarantee avoidance of loss of support through satisfaction of friction model conditions, is referred to as the force distribution problem. This is a common problem in other areas of robotics science as well, such as robotic walking, cooperative or constrained manipulation. An important property on which the nonlinear constraint optimization problem is based, and on which the force distribution problem is founded, is convexity. Given the satisfaction of this property, efficient finding of solutions to this complex problem becomes possible. For the same problem, numerical solutions of iterative form have also been proposed through the integration of smooth ordinary differential equations (ODE). Important for the for-

mulation of the optimization problem is the finding that the nonlinear constraints for friction can be formulated as appropriately positive definite matrices. This formulation of constraints in matrix form also led to an extension, transforming the problem into a standard Linear Matrix Inequality (LMI) problem where ready mature solutions are implemented and available in widely available software.

## 2.2 System Description

### 2.2.1 Modeling

The robotic manipulation system constitutes a complex nonlinear dynamics system. Additionally, its subsystems, links - joints, may be coupled. This inevitably leads to particularly advanced modeling and consequently control techniques. Also, during contact with the environment, a kinematic constraint is introduced into the system through a process of mechanical deformations. This deformation depends on the hardness - stiffness of the object as well as on the hardness - stiffness and shape of the end effector of each robotic configuration. Given this interaction, a reaction force is expected to be created at the end effector which is channeled to each link of the robotic configuration.

#### Hybrid Modeling of Grasping Dynamic System

Dexterous control and manipulation of objects by a multi-finger robotic grasp combines characteristics from two types of interacting dynamic systems. On one hand, we have the complex multibody dynamics, which is modeled by a set of nonlinear differential equations, subject to holonomic kinematic and dynamic constraints. This part falls under the theory of Continuous Variable Dynamic Systems – CVDS.

On the other hand, we have a set of discrete phenomena describing the state of contacts (*discrete grasp states of fingers*) which are described by the theory of discrete event dynamic systems (DEDS).

For the modeling of systems that combine discrete and continuous dynamics, the theory of *Hybrid Dynamical Systems* [15] can be used, introducing the corresponding modeling-analysis as well as the related control techniques.

For robotic systems, an additional formalism can be introduced in their description in order to specialize the general theories and descriptions of hybrid dynamical systems theory for *Mechatronic Multicontact Systems* [16].

A central element in the description of hybrid dynamical systems is the *Hybrid State Model–HSM*, in which the evolution of the system over time is given at any time either by the continuous-time differential equation, in case the appropriate selection function gives a non-zero value, or by the discrete state function for a zero value of the selection function. This approach of description - analysis - control through the theory of hybrid dynamical systems constitutes the highest level of formalism that one

can achieve for the holistic description of the robotic grasping system [16].

The Hybrid description of the dynamic system, although complete, introduces a significant degree of complexity and difficulty in analysis and development of control techniques. Thus, usually most works dealing with dexterous manipulation focus on the continuous dynamics part that describes either the robotic system in free space or the grasp-object system under stable and closed grasp.

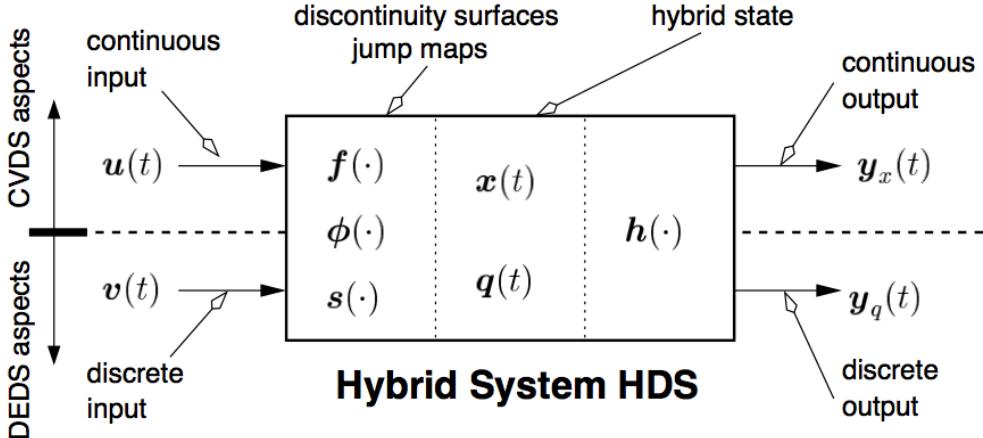


Figure 2.2: Hybrid System HDS [16]

## 2.2.2 Sensory Synthesis

For the realization of any intelligent action, through some control scheme, information about the state of the environment is necessary.

For robotic manipulation systems, two are the dominant sensory pathways from the environment and arise from vision systems and force/tactile sensors.

### Vision Systems

One of the first control methods used in robotic manipulators is visual information feedback. This type of control has proven to be an effective way for accurate guidance in free space, always within the robotic workspace, without prior accurate modeling of the system. Basic practical uses of such systems are found in trajectory planning tasks and in determining the geometry of unknown objects. Two configurations are more common: placement of the optical element at a fixed point in space and placement on the end effector.

Practically, obtaining sufficient depth information with a single measurement is difficult. Thus, for 3D measurement, techniques such as multiple view synthesis and stereovision are used. Generally, approaches for vision-based control can be divided into two categories [13].

- Position-based techniques, where a set of images are initialized together with a known camera model for extracting position/orientation information in 3D

space. The variables under control are the Cartesian position and orientation of the object. In the case where the camera is at a fixed point and the object's position/orientation is under control, the variables describing position and orientation are reconstructed from the available images. Consequently, object detection can be accomplished by calculating the error in 3D space, and the object's position can be extracted using image information and a calibrated camera model.

- Image-based techniques. In this approach, the variables under control are defined directly as features in image space and thus full 3D scene reconstruction is not necessary. Object detection with this particular technique is accomplished by calculating the error in image space and applying control that guarantees this error will asymptotically decrease to zero. For a fixed camera, the image Jacobian matrix can be calculated using a camera model. Due to distortions introduced in the image, feature identification is not accurate. Even worse results are introduced in the camera-on-end-effector configuration.

Finally, both techniques, given their inaccuracy in determining position and orientation, are generally judged unsuitable for use in achieving and maintaining contact with the object surface [13].

However, we should not categorically reject techniques that aim to achieve stable grasp for an unknown object based solely on visual information. For example, in [17], reconstruction of the unknown object model is performed through a specially adapted laser 3D scanner on the robotic arm's wrist with simultaneous adoption of optimal grasp selection technique based on closure criteria as well as motion within the friction cone, automating the object grasping process. In the case of two soft fingers, a quality criterion for grasping the object based on visual information consists of a term that minimizes the distance of the fingertips parallel with minimizing the distance of the geometric center of the object from the axis connecting the fingertips, and a second term that aims at applying force from the fingers as perpendicular as possible to its surface [4].

The vision system generally, however, seems to find complementary application in the first stage, that of approaching the object and perhaps early grasp formation [13]. Once the robotic end effector reaches an appropriate distance, the process of achieving and optimizing stable contact is performed using information provided in real-time from tactile and force sensors. Control strategies that are adopted and make use of force and tactile sensors usually aim at minimizing internal forces (grasping forces) or optimizing the position/orientation of the object and ultimately achieving dexterous manipulation. The goal of the entire system is the autonomy of the manipulation system and its control only at the high level of direct object control.

### **Generalized Force Sensors – Tactile**

Force sensors that are commercially available are usually installed in the corre-

sponding robotic wrist, or in the tendons of the robotic hand. They usually measure the forces and torques that develop in the robotic hand during interaction with the environment. The largest part of such sensor configurations consists of transducers that detect some geometric change - deformation of some appropriately designed - placed element, as a function of some applied force-torque.

Tactile sensors are usually placed on the surface intended for direct contact, mainly the fingertips but also other internal points of the fingers and palm. The measurement concerns the applied pressure observed during interaction. This is performed through an electronic configuration that includes a planar symmetric array of smaller pressure detection sensor elements that collectively give us a mapping of the applied pressures. The most advanced of these sensors are able to give us a complete picture of the 6D force vector of the contact.

It is particularly important to mention at this point that traditional force sensors yield noisy signals, difficult to process [18]. This fact also serves as a trigger for the implementation of alternative dexterous manipulation techniques that do not make exclusive or any use of force sensors at contact points.

Force-tactile sensor technology, however, seems to be evolving, with research efforts providing new solutions, offering the capability for accurate knowledge of the dynamic characteristics of the contact in real time, with the direct result of applying direct force control. An interesting approach is the development of high-precision optical tactile sensors capable of measuring normal as well as simultaneously tangential forces, which are able under an appropriate control scheme to acquire during the first approach of the object information about its stiffness, adapting the dynamic manipulation control parameters subsequently, offering robust dexterous manipulation for a wide range of objects of various mechanical properties [19].

Another promising tactile sensor technology is able to acquire information about the texture of objects, in a manner similar to humans, by rubbing the sensor adapted to each fingertip on the surface of the object [20].

Finally, we should note that the technique of appropriate integration of force-tactile sensors in robotic hands with soft surfaces constitutes a particular challenge given that sensory information acquisition should not directly affect the manipulation process [11].

## 2.3 Robotic Grasping Control Methodologies

### 2.3.1 Force Control

Two basic control categories that arise for a robotic kinematic chain are “direct force control” and “indirect force control” which achieves force control through kinematic control. The basic practical implementations of these two categories are achieved with

hybrid position/force control and impedance control respectively.

### **Hybrid Position/Force Control**

Hybrid position/force control attempts to decouple the directions in which force and position control are performed. The direction in which there is no constraint is handled with position control and the direction in which there is constraint-contact with force control. Thus, in the final control design, there are two parallel control loops. Practically, the switching between these two loops may not happen fast enough to handle environmental changes [4].

### **Impedance Control**

In contrast to hybrid position/force control, impedance control combines position/force control. This approach aims at smoothing the stiffness of the robotic manipulator by defining the desired impedance at the end effector. From a different perspective, this method aims at controlling position and force at the same time by expressing the desired task as the achievement of appropriately defined desired impedance. The final complex mechanical admittance of the environment, as well as the final position and applied force, will be a function of the robotic impedance. Impedance control is considered the most suitable solution for handling interactions in unstructured environments [13]. Problems are identified due to modeling errors or due to unmodeled dynamics where the controller causes unwarranted action.

The impedance control scheme has two expressions.

- Impedance Control. The robotic configuration reacts to deviation from the given trajectory by generating forces.
- Admittance Control. The robotic configuration reacts to external forces by deviation from the desired trajectory, maintaining interaction forces at desired values.

Special cases of Impedance and Admittance control are the stiffness control and compliance control techniques, respectively, where we are interested only in the static relationship between the position - orientation of the end effector and desired motion and the contact force - torque which are taken into account. If the relationship between the contact force-torque and the linear and angular velocities of the end effector are the quantities of interest, the control scheme is called damping control.

### **Hybrid Impedance Control**

Hybrid impedance control combines hybrid control and impedance control, with an inner inverse dynamics control loop and an outer loop aimed at achieving appropriate desired characteristics, such as setpoint tracking, disturbance rejection, as well as robustness issues. Depending on what needs to be controlled each time, this scheme is renamed more specifically, hybrid impedance/position control or hybrid

impedance/force control. For example, impedance control with respect to force can be used for the generation of those internal forces that will guarantee that contact with the object will not be lost, while impedance control with respect to position can place the fingers and thus objects at the desired point.

At the controller level itself, various classical as well as modern controllers have been proposed from the robotics science literature for controlling manipulator motion. These vary from classical PID to nonlinear modern approaches such as “variable structure,” adaptive, and robust. Recently, the trend in the robotics community is shifting to Artificial Intelligence topics, such as expert systems, fuzzy logic, neural networks. Industry, however, seems to remain loyal to analytical solutions, although simple learning algorithms always constitute a possible solution for industrial applications, especially at the task level.

### 2.3.2 Object-Level Stiffness Control

A dexterous hand has the capability to hold any arbitrary object as well as to react to arbitrary movements and forces that may act on the object. These movements and forces, being coupled and based on an appropriate control scheme, can determine the generalized stiffness of a manipulated object as well as its corresponding compliance center.

Of interest is the application of this object stiffness control scheme so that the reaction force during interaction guides some assembly process as part of a broader task of the robotic manipulator. Practically, a robotic hand can be used in the production process at assembly points where they can provide active motion capabilities of mechanical compliance, which is traditionally done by an inherently passive manipulator or by a passive remote center compliance element. A robotic hand, due to its particularly low inertial characteristics, can potentially offer higher bandwidth compared to a typical serial link manipulator [21].

For the successful assembly process, the resultant force on the object must be properly coupled with the object’s motion. This can be done in various ways, one of which is the introduction of an appropriate control scheme for determining the mechanical impedance of objects. This technique does not aim at dexterity as we defined it in terms of the ability to orient the object in space.

Several approaches define control only with respect to the static term of the object impedance model, that of stiffness (object stiffness control). Control moves around the static equilibrium point, generating force on the object through the fingertips proportionally to the displacement from the equilibrium point, simultaneously with internal force control logic (Grasp Force Optimization) to avoid loss of support.

A technique for controlling internal forces that acts within the framework of object position stiffness control considers these forces parallel to the respective axes connecting the contact points of the fingers and incorporates them into the generalized force vector of the object. Subsequently, the grasp matrix is extended with respect to the rows for

mapping from fingertip force space to internal force space on the object. Object stiffness control through determination of appropriate joint torques is done by inverting the now square grasp matrix and then using the Jacobian matrix. This approach, based on an experimental tendon-driven motion transmission configuration, has already yielded satisfactory results [21].

### 2.3.3 Object-Level Stiffness Control with Redundant Kinematic Configurations

In the case of object stiffness control using fingers with redundant degrees of freedom, the investigation of appropriate techniques for satisfying desired individual criteria is required. An interesting object stiffness control technique decomposes the problem by initially determining a static model with which the stiffness of the object space is connected to the individual stiffness parameters of the fingertips. Given the desired object stiffness, the problem is reduced to determining the stiffness parameters of the fingertips. Toward this direction, the initial model can be brought to linear form by assuming that the stiffness matrix has exclusively non-zero diagonal elements and forming with these a vector of stiffness coefficients.

The calculation of fingertip stiffness elements is done based on an optimization algorithm where minimization of the norm of the difference of stiffness elements with the minimum value they can have is required, satisfying, as an equality constraint, the linear equation of the system. This algorithm is called the Fingertip Stiffness Synthesis algorithm (FSS) and consists of two stages.

#### FSS Algorithm

- Initially ensures the appropriateness of the selection of object stiffness parameters so that stability results for the system.
- Calculates the elements of the stiffness matrix based on the classical solution for least squares problems with equality constraint.

Subsequently, the problem is reduced to determining joint stiffness based on the calculated stiffness matrix in the fingertip space. For this purpose, a control technique called Orthogonal Stiffness Decomposition Control (OSDC) has been developed.

#### OSDC Algorithm

Based on the static equation that connects joint torques with the force applied by the fingertips, through the Jacobian kinematic matrix, the relationship between joint stiffness and fingertip stiffness is derived. Given that the fingers have redundant degrees of freedom, a technique of defining the null space of the mapping through an appropriate similarity transformation is followed.

Finally, control at the joint level includes the stiffness term with gain calculated based on OSDC, a dynamic damping term with gain determined by the inverse matrix of the stiffness matrix term mapping to the non-null space, and a dynamic gravity compensation term.

The entire control scheme that combines the individual techniques FSS and OSDC, called Decentralized Object-Level Stiffness Control (DOSC), is effective in reliably determining stiffness at the object level with robotic fingers of redundant degrees of freedom [22].

Another technique for determining joint stiffness for a given diagonal fingertip stiffness matrix in redundant kinematic configurations is based on the augmented spaces technique [23]. Essentially, a subtask is defined in the null space of the Jacobian matrix of the kinematic configuration, decoupled from the main task. In this way, the stiffness of the main task, which is the stiffness in the fingertip space, can be mapped separately from the stiffness of the subtask, thus giving a full-rank joint stiffness matrix.

### 2.3.4 Admittance Control

As we have already mentioned, robotic admittance control aims to produce appropriate displacements of the robotic configuration, thus reacting to externally applied forces while maintaining interaction forces at desired values. For the robotic grasping system, this dynamic compliance behavior is desirable to be adopted in relation to the manipulated object.

In the case where admittance control is limited only to determining the static parameters of the dynamic model of the closed system, then the control scheme is called compliance control. This type of static analysis is generally useful for general conclusion extraction about the grasping system, particularly in linear approximation around an equilibrium point or during small motions where inertial terms are small.

A methodology for achieving compliance control at the object level consists of appropriately determining the stiffness matrices, first with respect to the fingertips - contact points, and then with respect to the finger joints, as in object stiffness control as described in previous sections. In [18], a control scheme is proposed for achieving compliance at the object level with a methodology for determining stiffness matrices so as to simultaneously achieve decoupling between fingers as well as between the joints of each finger. This decoupling facilitates hand control. More specifically, the implementation includes two basic stages implemented with two different algorithms respectively.

- **Resolved Interfinger Decoupling Solver (RIFDS)**

Initially, it is checked whether the desired object stiffness matrix is feasible by the respective robotic manipulation configuration based on a heuristic model implemented with predefined matrices that take into account the contact model as well as the kinematic configuration based on available degrees of freedom. Subsequently, once the Grasping Matrix is calculated for the current time step, the

equations connecting the stiffness matrices of the object and fingertips, which arise from the static force-velocity equation of the grasp, are appropriately transformed into a linear relationship, with the calculation of the fingertip stiffness matrix now having been converted into a linear programming problem.

- **Resolved Interjoint Decoupling Solver (RIJDS)**

At this stage, the goal is to determine the stiffness parameters for each joint of the fingers given that the stiffness matrix with respect to the fingertip has already been determined with the RIFDS algorithm. In a similar manner, the mathematical relationship connecting the two stiffness matrices based on the differential Jacobian matrix is transformed into linear form, ensuring decoupled behavior between joints. The final solution is obtained by simple inversion of the linear mapping matrix of this relationship.

### 2.3.5 Hybrid Dynamic Control of Robotic Grasping System

As we mentioned in the chapter dealing with system modeling 2.2.1, the highest level of formalism we can achieve for the complete description of the robotic grasp - object system is that of hybrid dynamical systems that combine discrete and continuous nature dynamics. Schleg, Buss & Schmidt in their work [16] attempt, within the framework of this holistic view of the problem, to introduce a complex control scheme with which they can endow the robotic grasp with almost all the characteristics that can characterize it as dexterous. More specifically, their design allows the robotic grasp robust object grasping, free placement in position and orientation, change of grasp configuration with regrasping, handling of disturbances and modeling inaccuracies.

This complex control consists of three basic parts,

- Hybrid Reference Generator (HRG). The desired trajectories are generated using hybrid reference automata. The hybrid trajectory combines the desired contact state characterized by the number of fingers in contact, the type of contact, the desired position of the fingertips as well as their corresponding velocities.
- Hybrid Controller (HC). This inner control loop drives the system toward the desired reference trajectory. In its discrete part, it generates an appropriate switch vector based on the discrete grasp state vector, aiming at switching between impedance controllers. In its continuous part, it undertakes the generation of the error signal that subsequently feeds the selected impedance controller.
- Grasp Force Optimization (GFO) & Impedance Control. The GFO technique used is analyzed in the next section [24]. The impedance controller has a series of individual impedance controllers from which the appropriate one is selected at each moment by the hybrid controller based on the vector describing the grasp

state at each time instant (see figure 2.3), thus ensuring the desired dynamic response of the system for each given moment.

Knowledge about the force applied to the object, necessary for the GFO control stage, is provided based on measurements from 6D generalized force sensors on the fingertips and mapping of these to the space of forces applied to the object through the grasp transformation matrix.

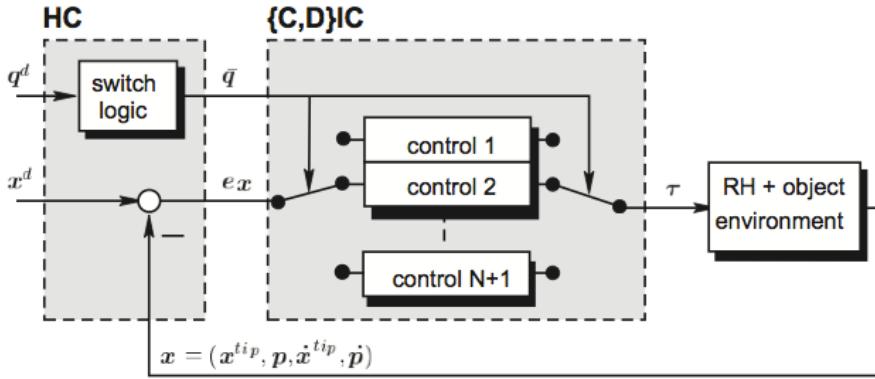


Figure 2.3: Bank of Impedance Controllers [16]

### 2.3.6 Grasping Force Optimization (GFO)

The application of grasping forces by the robotic manipulation configuration on the object in an optimal manner constitutes one of the basic prerequisites for achieving robust dexterous manipulation. Classical approaches to this particular problem operate under the assumptions of point contact with friction and the interaction of compact objects. Under this simplifying prism for analysis, the two most basic points that optimal grasping force control must satisfy are always moving within the friction cone, ensuring adequate support of the object, and simultaneously minimizing internal forces under the logic of minimum effort as well as protecting the overall system from damage. It is immediately obvious that these goals are diametrically opposed and require a compromise.

Regarding the formalism of describing the grasping force optimization problem, the formulation of the Coulomb friction model terms for each contact in the form of a positive definite matrix has contributed greatly. Based on this, a corresponding cost function can be formed that describes the optimization criteria mentioned above. Such a cost function could be, for example, the sum of elements of the weighted friction term matrix plus the sum of the weighted inverse matrix of friction terms. Thus, by selecting appropriate weighting matrices and minimizing the cost function, it is possible to satisfy the criteria we described. This weighting can even change dynamically for application in regrasping technique where smooth transition from the contact stage to the non-contact stage is required for each finger and vice versa, in combination

with a set of additional linear constraints with respect to the friction term matrix for improving the effectiveness of the technique [24].

Regarding the computational aspect of this method, there are many approaches based on classical optimization technique theory which, however, are not judged suitable for real-time application [24]. In the case where real-time response is required, algorithms of iterative numerical solution techniques are preferable, which indeed may reduce accuracy requirements in favor of system speed, i.e., to ensure immediate response. Buss and Schleg [24] propose an algorithm which, initially, given the external force with respect to the object, increases the internal forces linearly, so that the total force from each finger is within the friction cone, and subsequently the adjustment of internal forces is done by optimizing the cost function described above with an iterative numerical method.

### 2.3.7 Robotic Hand Preshaping

Essential for the manipulation of an object is the stage during which the robotic configuration approaches the object, forming an appropriate shape for establishing a closed grasp. This approach, although intuitively simple, presupposes a coordination of complex subtasks.

- Recognition of the object, its position, shape, and size.
- Generation of appropriate control signals for the kinematic configuration for tracking object motion.
- Design of appropriate grasp based on the shape and surface morphology of the object, its application, and finally manipulation of the object.

An interesting approach [25] sets up an appropriate virtual impedance scheme with two components. One component is defined as the virtual stiffness for each fingertip with respect to the object position at each time instant, thus ensuring tracking of object motion. The second virtual stiffness component ensures that the fingertips will symmetrically cover the space “embracing” the object symmetrically using appropriate functions that repel all fingers from each other, thus ensuring closure. These two virtual impedances are superimposed on the final control signal and mapped to joint space according to the classical static model with the inverse differential Jacobian matrix.

### 2.3.8 Other Techniques

#### Machine Learning Techniques

Machine learning techniques are constantly gaining ground in robotic systems, either complementary to classical control techniques or as standalone intelligent control systems.

In [12], a control technique is presented for a 2-finger 2 DOFs grasping system with soft fingertips with rolling capability on the object surface. The dynamics of this system, also referred to as “Pinching Motion,” is described with a Lagrangian dynamic model only for motion in a 2-dimensional plane. Control is based on the system being passive (Passivity Based Control) and includes three additive individual terms. The first term performs stable dynamic grasping by controlling internal forces, with the second undertaking object rotation, while the third term, a learning term, is iteratively updated using a dynamic regression integration differential term based on the dynamic characteristics of the Lagrangian robotic hand model. The goal of control is to perform stable periodic motion with gradual convergence to the desired trajectory by zeroing the position error with respect to the object through the dynamic learning term. This technique generalizes to 3 DOF fingers with successful results [12].

#### Control with Soft Contact Points

As we have already mentioned, most analytical studies assume rigid fingers and rigid environment objects. This creates instability and adaptability problems in different situations in practical applications. For this reason, there is particular development in the field of soft robotic systems (Soft Robotics).

An interesting technique for manipulating objects with 2 soft fingertips that have adapted force sensors uses a classical PID controller at the joints where the desired position acts as a regulator of the applied force [4]. More specifically, first a filter is defined that denoises the signal from the force sensors. Then the grasping force is defined as the smaller value of the forces acting on the object along the axis between the two contact points. The error between the grasping force and the desired one is defined as proportional to the rate of change of the fingertip distance from the center. Based on inverse kinematics, the desired joint position is finally defined. An extension made to this particular scheme is the introduction of a compliance technique for large forces. In this case, when an externally applied force on the object greater than a threshold is detected, then the center between the two contact points is moved toward the direction of the externally applied force in order to reduce it. This technique can be used for executing tasks where external forces drive the task [4].

### 2.3.9 Stability Issues

One of the most important properties of the grasp is its stability. In the literature, the term stability is used based on at least two meanings [8] and is determined

mainly at the object level [23]. One has to do with Lyapunov stability, and indicates (asymptotic) stability if the system dynamics are such that when the object is displaced from the reference position/orientation, it stays close (and eventually returns to it). A second definition, according to Lagrange stability, states that a conservative system is considered stable if it is at a strict local minimum of its potential energy. The second definition is the dominant one in the analysis and study of stability of robotic grasping systems. In stability analysis, the compliant dynamics of the system must also be taken into account, such as any elasticity phenomena in the fingers.

Lagrange stability analysis presents some practical problems. In mechanics, the position that an equilibrium point is unstable if it is not a minimum point of potential is not proven for systems with more than 2 degrees of freedom. In a real application, non-conservative forces can also act, created by imperfections in the mechanical structure of elements, or/and by the control law, rendering Lagrange stability analysis invalid. Lyapunov stability analysis as well as other structural properties (controllability, observability, stabilization) in the general grasping system are analyzed with respect to their linear approximation where the system is practically elastic around the equilibrium point.

In an elastic system, stability practically translates as the property according to which any displacement from the equilibrium point will create a force that will tend to return the system to the equilibrium state (*stiffness-effect*). Energetically, this is interpreted in terms of the work that the system must produce for returning to the equilibrium point. For the analysis of this problem, the relationship between stiffness matrices between object, fingertips, and joints is usually first structured, which arises from the static analysis of the system at the object level through the grasp matrix and the hand Jacobian matrix. From this relationship, the grasp stability conditions are extracted, usually based on some simplifying assumptions to facilitate analysis, such as omitting gravitational effects or other unmodeled effects and second-order terms. In this simplifying assumption, setting the object stiffness matrix positive definite ensures stability for the elastic system. Usually, stiffness matrices are chosen diagonal, ensuring ease of analysis and decoupled behavior with respect to Cartesian directions.

Obviously, appropriate compensation of various effects on the system is deemed necessary for achieving stability. In this direction, in [22], an algorithm for appropriate selection of object stiffness matrix through an iterative heuristic method is proposed that guarantees stability for the elastic system, taking into account each time the gravitational effect as well as the effect of internal forces. In [23], considering that a direct force control law (Closed Loop) acts on the fingertips of the grasping system, the grasp stability conditions for the linearized elastic system with gravitational effect are structured, under a simple model of point contacts with friction, extending the corresponding conditions for the open-loop system. Another approach proposes the introduction of a disturbance observer in the control scheme either of the servomechanism or in force or compliance control, for compensating uncertainty, the gravitational term, elasticity, and link friction. The effectiveness and performance of

the disturbance observer is judged high, but stability cannot be guaranteed based on analytical solution *a priori* before the application of the controller to the final system. The stability conditions structured in research efforts so far are examined only with respect to certain components of disturbances in the system, and controller parameters are adjusted either through iterative or graphical methods, or intuitively [26]. In [26], sufficient analytical conditions are first structured for ensuring stability (stiffness-effect) for small displacements around the equilibrium point where there is an elastic operating region under the effect of the gravitational term of the object and with respect to its mass, contact points, and elastic terms of the fingertips. Subsequently, for tuning the controller parameters (which incorporates a Disturbance Observer) for ensuring stability, a Lyapunov function for the system is structured from which satisfaction of the corresponding stability conditions is also required. Asymptotic stability is proven with LaSalle's invariance principle.

For comparing stability between different grasps as well as generally for stability analysis, the calculation of a corresponding appropriate metric is judged useful, such as the real part of the dominant eigenvalues of the linearized grasp model. A more useful approach in several applications would be the calculation of the vector basis of generalized attractive forces around the equilibrium point, thus providing information about the range of force magnitude for which the system is stable. Efficient algorithms for calculating this quantity have not yet been developed [8].

# Chapter 3

## Modeling & Control of Robotic Grasping: Theoretical Background

A primary goal of a robotic hand is the manipulation of objects in all 6 dimensions. In the past, many algorithms have been proposed based on the pseudoinverse of the grasp matrix with some weighting factor, combined with internal-force control. Most of these methodologies require robust contact detection/tracking as well as switching controllers. By introducing the concept of the *virtual object* (*virtual-object*), object-level robotic grasping control techniques have been developed and are reported in the literature [27, 28, 29]. The controller takes as input the desired object frame and the desired grasping forces. Object-level control generally has the following advantages:

- Easy definition of grasping forces to compensate for the inertial characteristics of the robot–object system
- Easy definition of external forces acting on the object
- Avoidance of unnecessarily large internal forces

Essentially, the goal of this work is to develop a robust and intuitive object-level control law that does not require robust contact detection/tracking.

### 3.1 Modeling of the Robotic Hand

In this section, we successively construct the kinematic, differential, and dynamic model of the robotic hand.

#### 3.1.1 Kinematic Model of the Robotic Hand

## Forward Kinematics

To describe the position and orientation of a kinematic chain consisting of prismatic and revolute joints with the minimum possible number of parameters, the Denavit–Hartenberg method is followed [30].

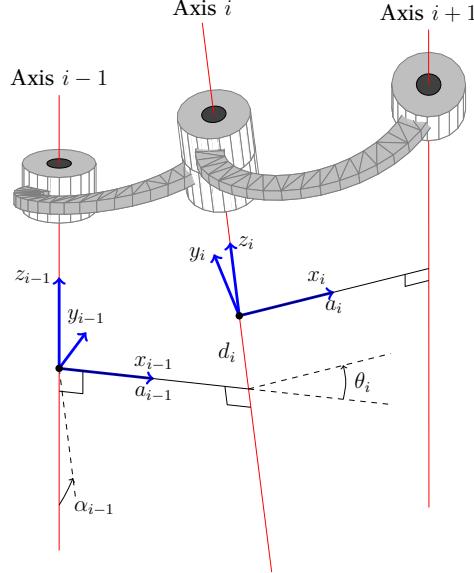


Figure 3.1: Denavit–Hartenbeg Parameters [31]

Link $i$	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	0	0	0	$\theta_1$
2	$l_1$	$\pi/2$	0	$\theta_2$
3	$l_2$	0	0	$\theta_3$
4	$l_3$	0	0	$\theta_4$

Table 3.1: Denavit–Hartenberg parameter table for an anthropomorphic 4-DOF finger

Thus, for describing the kinematic relationship of the fingertip with respect to the common reference frame, we have the following successive homogeneous transformations:

$$T_i(\theta_i) = A_{\Sigma_i}^O [Rot_z(\theta_{i1}) * Rot_x(\pi/2)] [Rot_z(\theta_{i2}) * Tra_x(l_1)] \\ [Rot_z(\theta_{i3}) * Tra_x(l_2)] [Rot_z(\theta_{i4}) * Tra_x(l_3)] \quad (3.1)$$

where  $A_{\Sigma_i}^O$  is the homogeneous transformation of the base frame  $\Sigma_i$  of finger  $i$  with respect to the common reference frame  $O$ , and  $\theta_i = [\theta_{i1}, \theta_{i2}, \theta_{i3}, \theta_{i4}]^T$ . The matrix  $T_i(\theta_i)$

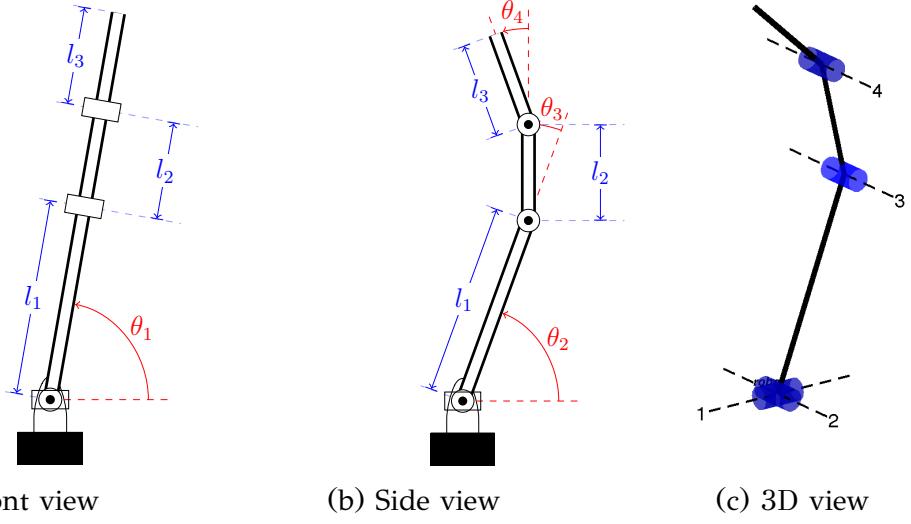


Figure 3.2: Kinematic structure of an anthropomorphic 4-DOF finger

consists of the rotation matrix  $R_i(\theta) \in \mathbb{R}^{3 \times 3}$  and the position  $x_i(\theta_i) = [x_{i1}, x_{i2}, x_{i3}]^T \in \mathbb{R}^3$ .

$$T_i(\theta_i) = \begin{bmatrix} R_i(\theta_i) & x_{i1}(\theta_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

For  $N$  fingers, the kinematic matrix of the entire hand is:

$$T(\theta) = \begin{bmatrix} T_1(\theta_1) & & & \mathbf{0} \\ & \ddots & & \\ \mathbf{0} & & & T_N(\theta_N) \end{bmatrix} \quad (3.3)$$

where  $\theta = [\theta_1^T, \theta_2^T, \dots, \theta_N^T]^T \in \mathbb{R}^{4N}$ .

## Differential Kinematics

At this stage, we aim to determine the differential mapping between the generalized fingertip velocity and the joint velocities.

By differentiating the final position of each finger  $x_i(\theta_i)$ , we have:

$$\dot{x}_i(\theta_i) = \frac{\partial x_i}{\partial \theta_i} \dot{\theta}_i \quad (3.4)$$

and for the angular velocity of the fingertip:

$$[\omega_i \times] = R_i^T \dot{R}_i \quad (3.5)$$

where  $[r \times]$  is the skew-symmetric operator  $[r \times] : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$ :

$$[r \times] = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix} \quad (3.6)$$

Expanding  $\dot{R}_i$  yields:

$$[\omega_i \times] = R_i^T \frac{\partial R_i}{\partial \theta_i} \dot{\theta}_i \quad (3.7)$$

Finally, we can define an inverse skew operator  $[[r \times] \times^{-1}] : \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}^3$ :

$$[[r \times] \times^{-1}] = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} \quad (3.8)$$

Thus, the Jacobian matrix for finger  $i$  is:

$$J_i(\theta_i) = \begin{bmatrix} \frac{\partial x_i}{\partial \theta_i} \\ [[(R_i^T \frac{\partial R_i}{\partial \theta_i}) \times^{-1}]] \end{bmatrix} \quad (3.9)$$

The Jacobian matrix of the entire hand is defined as:

$$J_H(\theta) = \begin{bmatrix} J_1(\theta_1) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & J_N(\theta_N) \end{bmatrix} \quad (3.10)$$

and the velocity relationship becomes:

$$\begin{bmatrix} \dot{x} \\ \omega \end{bmatrix} = J_H(\theta) \dot{\theta} \quad (3.11)$$

### 3.1.2 Dynamic Model of the Robotic Hand

The dynamic model of the robotic hand with  $N$  fingers of four degrees of freedom is described by the following differential equation, which is derived via the Lagrange method.

The Lagrangian function is defined for finger  $i$ :

$$L_i = K_i - P_i \quad (3.12)$$

where  $K_i = \sum_{j=1}^4 K_{ij}$  and  $P_i = \sum_{j=1}^4 P_{ij}$  are the total kinetic and potential energy of each finger, respectively. The kinetic energy of each finger is the sum of the kinetic energy of the center of mass of each link due to linear motion and the kinetic energy due to rotation:

$$K_{ij} = \frac{1}{2} m_{ij} \dot{x}_{cm\_ij}^T \dot{x}_{cm\_ij} + \frac{1}{2} \omega_{ij}^T I_{ij} \omega_{ij} \quad (3.13)$$

$$P_{ij} = m_{ij} g h_{ij} \quad (3.14)$$

with  $m_{ij}$  and  $\dot{x}_{cm\_ij}$  being the mass and linear velocity, respectively, of the center of mass  $cm$  of link  $ij$ .  $\omega_{ij}$  and  $I_{ij}$  are the angular velocity and inertia tensor of link  $ij$ .  $h_{ij}$  is the height of the link  $ij$  center of mass.

The Lagrange dynamic model is expressed as:

$$\frac{d}{dt} \left( \frac{\partial L_i}{\partial \dot{\theta}_i} \right) - \frac{\partial L_i}{\partial \theta_i} = \tau_i \quad (3.15)$$

From (3.12)–(3.15), the dynamic model of robotic finger  $i$  follows:

$$M_i(\theta_i)\ddot{\theta}_i + C_i(\theta_i, \dot{\theta}_i)\dot{\theta}_i + g_i(\theta_i) = \tau_i + \tau_{i\_ext} \quad (3.16)$$

where  $\theta_i, \tau_i = [\tau_{i1}, \tau_{i2}, \tau_{i3}, \tau_{i4}]^T$ , and  $\tau_{i\_ext} \in \mathbb{R}^4$  are the joint angles, joint torques, and externally applied joint torque, respectively.  $M_i(\theta_i) \in \mathbb{R}^{4N \times 4N}$ ,  $C_i(\theta_i, \dot{\theta}_i)$ , and  $g(\theta_i) \in \mathbb{R}^{4N}$  are the inertia matrix, the centrifugal and Coriolis terms, and the gravity term, respectively. All these terms are expressed in joint space.

Finally, for a robotic hand of  $N$  fingers, the dynamic equation is:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau + \tau_{ext} \quad (3.17)$$

with:

$$M(\theta) = \begin{bmatrix} M_1(\theta_1) & & & \mathbf{0} \\ & \ddots & & \\ \mathbf{0} & & M_N(\theta_N) & \end{bmatrix} \quad (3.18)$$

$$C(\theta, \dot{\theta}) = \begin{bmatrix} C_1(\theta_1, \dot{\theta}_1) & & & \mathbf{0} \\ & \ddots & & \\ \mathbf{0} & & C_N(\theta_N, \dot{\theta}_N) & \end{bmatrix} \quad (3.19)$$

$$g(\theta) = \begin{bmatrix} g_1(\theta_1) & & & \mathbf{0} \\ & \ddots & & \\ \mathbf{0} & & g_N(\theta_N) & \end{bmatrix} \quad (3.20)$$

and  $\tau = [\tau_1^T, \tau_2^T, \dots, \tau_4^T]^T \in \mathbb{R}^{4N}$ .

### 3.1.3 Basic Properties of the Robotic Hand Dynamic Model

The dynamic equation (3.16) and, by extension, (3.17) are nonlinear. Therefore, since the tools of classical control theory for linear time-invariant (LTI) systems cannot be used (at least not directly), new approaches are sought based on the basic structural properties of the system:

- $M(\theta) > 0$  &  $M(\theta)^T = M(\theta)$ .
- $(\dot{M} - 2C)^T = -(\dot{M} - 2C)$ , i.e., a skew-symmetric matrix.

### 3.1.4 Passivity-Based Robotic Control

From nonlinear systems theory, we have the following definition for passive systems [32].

Consider the system:

$$\dot{x} = f(x, u) \quad (3.21)$$

$$y = h(x) \quad (3.22)$$

where  $f(0, 0) = 0$  and  $h(0) = 0$ . This system is passive if there exists a positive semidefinite function  $V(x)$  (storage function) such that:

$$u^T y \geq \dot{V} = \frac{\partial V}{\partial x} f(x, u). \quad (3.23)$$

The system is zero-state observable if no solution of  $\dot{x} = f(x, 0)$  belongs to the set  $\{x | h(x) = 0\}$  except for the trivial solution  $x(t) = 0$ .

Interpreting the storage function  $V(x)$  as the energy of the system, we observe that a passive system has a stable equilibrium at the origin ( $x(t) = 0$ ). Control therefore relies on feedback that enforces passivity.

For global stabilization of a passive system at the origin, given a radially unbounded positive definite storage function and zero-state observability, one may apply a control law of the form  $u = -\phi(y)$ , where  $\phi(y)$  is locally Lipschitz such that  $\phi(0) = 0$  and  $y^T \phi(y) > 0$  for all  $y \neq 0$ :

$$\dot{x} = f(x, -\phi(y)) \quad (3.24)$$

Using  $V(x)$  as a Lyapunov candidate, we obtain:

$$\dot{V} = \frac{\partial V}{\partial x} f(x, -\phi(y)) \leq -y^T \phi(y) \leq 0 \quad (3.25)$$

so  $\dot{V}$  is negative semidefinite. Note that  $\dot{V} = 0$  if and only if  $y = 0$ . By LaSalle's invariance principle, asymptotic stability for  $y = 0$  follows, implying  $u(t) = 0$  and finally  $x(t) = 0$ .

In the control problem for the robotic system, we wish to stabilize the system at a reference point  $q = q_r$ . The tracking error dynamics  $e = q - q_r$ ,  $\dot{e} = \dot{q}$  follow from (3.17):

$$M(\theta)\ddot{e} + C(\theta, \dot{\theta})\dot{e} + g(\theta) = \tau \quad (3.26)$$

The point  $(e = 0, \dot{e} = 0)$  is not an equilibrium point of the open-loop system.

Let the control input include gravity compensation, a position-error term, and a damping term:

$$\tau = g(q) - \phi_p(e) - D\dot{e} \quad (3.27)$$

where  $\phi_p(0) = 0$ ,  $D$  is a positive definite ( $D > 0$ ) velocity-feedback gain matrix, and  $e^T \phi_p(e) > 0$  for all  $e \neq 0$ . The closed-loop equation is:

$$M(\theta)\ddot{e} + C(\theta, \dot{\theta})\dot{e} + D\dot{e} + \phi_p(e) = \tau_{ext} \quad (3.28)$$

The system energy (storage function) is the sum of kinetic energy and potential:

$$V = \frac{1}{2}\dot{e}^T M(\theta)\dot{e} + V_d(e) \quad (3.29)$$

where:

$$V_d(e) = \int_0^e \phi_p^T(\sigma)d\sigma \quad (3.30)$$

The total derivative of the system energy is:

$$\dot{V} = \frac{1}{2}\dot{e}^T (\dot{M} - 2C)\dot{e} - \dot{e}^T D\dot{e} - \dot{e}^T \phi_p(e) + \dot{e}^T \tau_{ext} + \phi_p^T(e)\dot{e} \leq \dot{e}^T \tau_{ext} \quad (3.31)$$

and for  $\tau_{ext} = 0$ :

$$\dot{V} = \frac{1}{2}\dot{e}^T (\dot{M} - 2C)\dot{e} - \dot{e}^T D\dot{e} \leq 0 \quad (3.32)$$

Defining the output as  $y = \dot{e}$ , we observe that the system with input  $\tau$  and output  $y$  is passive with storage function  $V$ . By LaSalle's invariance principle,  $\dot{V} = 0 \Rightarrow \dot{e} = 0 \Rightarrow \ddot{e} = 0 \Rightarrow \phi_p(e(t)) = 0 \Rightarrow e(t) = 0$ .

## 3.2 Static Grasp Analysis

In this section, we formulate the relations of the static grasp model for point contacts with friction. First, we define the Grasp Force Transformation Matrix through the static equilibrium relations. Then, we provide the definition of the stiffness effect and, by extension, stability conditions for robotic grasping.

### 3.2.1 Static Equilibrium & Grasp Force Transformation Matrix

The static equilibrium relations with respect to forces and torques are:

$$\sum_{i=1}^N f_{ci}^o + f_{ext} = 0 \quad (3.33)$$

$$\sum_{i=1}^N (r_{ci} \times f_{ci}^o) + n_{ext} = 0 \quad (3.34)$$

where  $f_{ci}$  are the forces exerted by fingertip  $i$ ,  $f_{ext}$  and  $n_{ext}$  are the total externally applied force and torque respectively, and  $r_{ci}$  is the fingertip position vector. All these

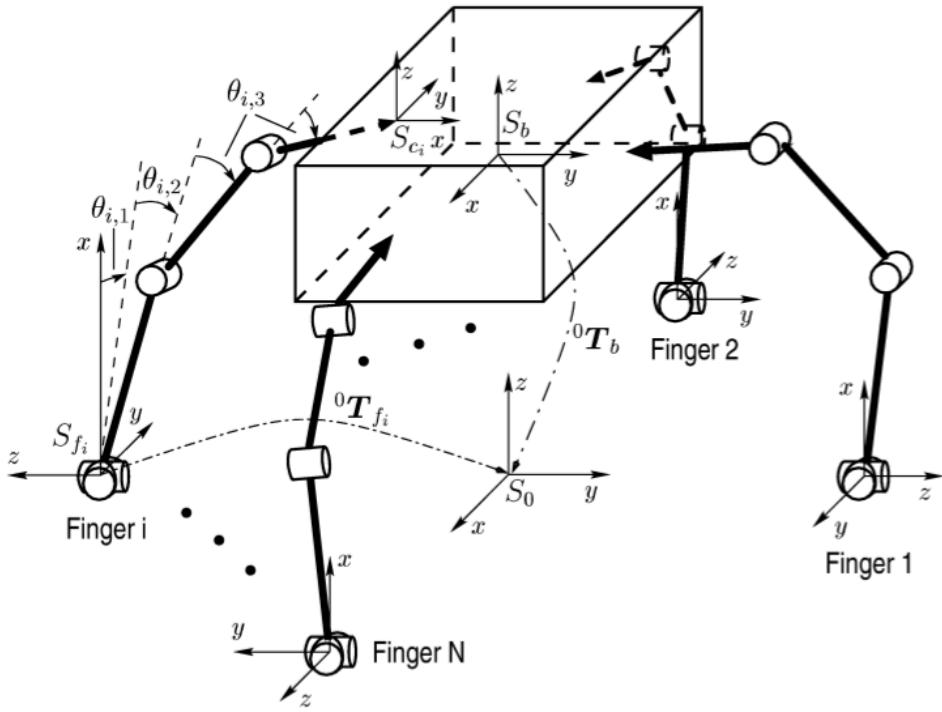


Figure 3.3: Schematic diagram of an object grasped by an  $N$ -finger hand with point contacts and friction [16]

quantities are expressed with respect to the local object frame. Writing these relations in matrix form yields:

$$\begin{bmatrix} I_{3 \times 3} & I_{3 \times 3} & \dots & I_{3 \times 3} \\ [r_{c1} \times] & [r_{c2} \times] & \dots & [r_{cN} \times] \end{bmatrix} \begin{bmatrix} f_{c1}^o \\ f_{c2}^o \\ \vdots \\ f_{cN}^o \end{bmatrix} = \begin{bmatrix} -f_{ext} \\ -n_{ext} \end{bmatrix} \quad (3.35)$$

We summarize the relation in the following form, expressing fingertip forces with respect to the corresponding frame they are expressed in:

$$f_c^o = \begin{bmatrix} R_o^T & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & R_o^T \end{bmatrix} f_c \quad (3.36)$$

$$\underbrace{\begin{bmatrix} R_o^T & R_o^T & \dots & R_o^T \\ [r_{c1} \times] R_o^T & [r_{c2} \times] R_o^T & \dots & [r_{cN} \times] R_o^T \end{bmatrix}}_G \underbrace{\begin{bmatrix} f_{c1} \\ f_{c2} \\ \vdots \\ f_{cN} \end{bmatrix}}_{f_c} = \underbrace{\begin{bmatrix} -f_{ext} \\ -n_{ext} \end{bmatrix}}_{F_{ext}} \quad (3.37)$$

$$F_{ext} = G f_c \quad (3.38)$$

where  $G \in \mathbb{R}^{6 \times 3N}$  is the grasp force transformation matrix.

The general least-squares solution for  $\min\|F_{ext} - Gf_c\|$  is:

$$f_c = [G]^+ F_{ext} + (\mathbf{I} - [G]^+ G)\xi \quad (3.39)$$

where  $[G]^+ = G^T(GG^T)^{-1}$  is the Moore pseudoinverse, and  $\xi \in \mathbb{R}^{4N}$  is an arbitrary vector.

Direct force control based on (3.39) is usually avoided, because direct measurements of fingertip forces are practically difficult; and when available, the acquired signals are noisy and generally hard to process [21, 18, 16, 19, 14, 13]. For this reason, indirect force-control schemes are preferred, typically expressed through impedance-control techniques (impedance) or, in a simplified form, stiffness control.

The dual relation of (3.38) yields the relationship between object and fingertip velocities:

$$\dot{x} = G\dot{x}_o \quad (3.40)$$

and a full description of the kinematic constraints for the hand–object grasp is given by:

$$J_H(\theta)\dot{\theta} = G\dot{x}_o \quad (3.41)$$

### 3.3 Passivity-Based Object-Level Impedance Control

In this section, we describe an object-level grasping control method based on passivity (Passivity Based Object Level Impedance Control), which has been proposed by Wimboeck, Ott, and Hirzinger for controlling the impedance of a manipulated object using a virtual frame. This scheme is referred to as Intrinsically Passive Control (IPC).

In addition, we present a technique for dealing with redundant degrees of freedom by defining an appropriate subtask in the null space of the robotic hand.

For modeling the dynamics and designing the control scheme, we assume the following regarding grasping and object manipulation:

- Point contacts with friction
- Internal forces are selected so as to satisfy the constraints of the friction model
- To allow object motion in 6D, contacts between object and robotic hand are restricted to the fingertips

#### 3.3.1 Virtual Object Frame

For fingertip control, a virtual frame is defined at the center of the contact points:

$$x_o(\theta) = \frac{\sum_{i=1}^N x_i(\theta)}{N} \quad (3.42)$$

where  $x_i(\theta) \in \mathbb{R}$  represent the Cartesian positions of the fingertips for finger  $i$  with respect to the base frame (frame 0).

The orientation  $R_o = [r_{1,o}, r_{2,o}, r_{3,o}]$  of the virtual frame is also defined based on the Cartesian fingertip positions. For  $N = 4$  fingers, the unit vector  $r_{1,o}$  is defined to be parallel to the plane formed by the axes defined by fingers  $i = 1, 3$  and  $i = 2, 4$ , respectively:

$$\tilde{r}_{1,o} = \frac{x_1 - x_3}{\|x_1 - x_3\|} + \frac{x_2 - x_4}{\|x_2 - x_4\|}, \quad r_{1,o} = \frac{\tilde{r}_{1,o}}{\|\tilde{r}_{1,o}\|} \quad (3.43)$$

The unit vector  $r_{3,o}$  is defined to be orthogonal to this plane, and finally  $r_{2,o}$  is defined such that the orientation frame satisfies the right-hand rule,  $R_o \in SO(3)$ :

$$\tilde{r}_{3,o} = [(x_1 - x_3) \times](x_2 - x_4), \quad r_{3,o} = \frac{\tilde{r}_{3,o}}{\|\tilde{r}_{3,o}\|} \quad (3.44)$$

$$r_{2,o} = [r_{3,o} \times]r_{1,o} \quad (3.45)$$

Finally, the virtual object frame  $H_o$  is defined as:

$$H_o = [R_o, x_o] \in SE(3) \quad (3.46)$$

This representation exhibits singularities in the case where the fingertips satisfy  $x_j - x_{j+2} = 0$  or  $(x_1 - x_3) \parallel (x_2 - x_4)$ . These singularities do not occur for common convex objects with geometries such as rectangular parallelepipeds, as well as cylindrical or spherical objects.

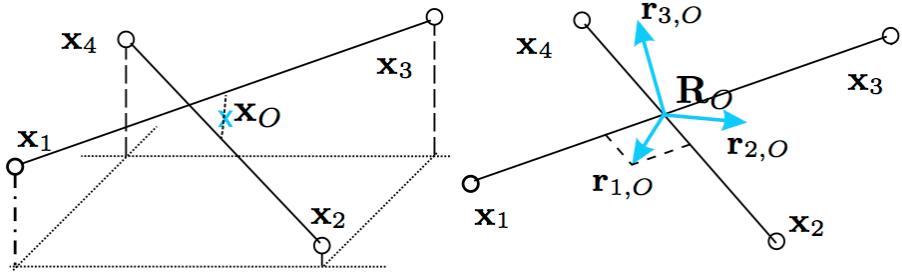


Figure 3.4: Virtual frame construction [27]

Once the virtual frame is defined, a central point in the overall effort of constructing a suitable control scheme is relating it to the actual frame of the manipulated object.

Thus, the virtual frame is considered to be kinematically associated with the object frame if and only if the object–finger contact points remain fixed throughout the manipulation stage, neglecting any rolling or sliding phenomena.

### 3.3.2 Object-Level Control Law

The control signal, i.e., the joint torque, is formed as follows:

$$\tau = -D(\theta)\dot{\theta} - \frac{\partial V_d}{\partial \theta}(\theta) + g(\theta) \quad (3.47)$$

where  $D(\theta)$  is a positive definite dynamic damping matrix. We define a desired potential-energy function which includes the following superposed terms:

$$V_d(\theta) = V_{o,t}(\theta) + V_{o,r}(\theta) + V_{Conn}(\theta) \quad (3.48)$$

where, through partial differentiation with respect to the joint angles  $\theta$ , the individual potentials  $V_{o,t}(\theta)$ ,  $V_{o,r}(\theta)$ , and  $V_{Conn}(\theta)$  yield, respectively: translational stiffness, rotational stiffness, and connection stiffness between the fingertips and the center of the virtual frame.

From passivity-based control theory, if we choose  $V_d(\theta)$  to be positive semidefinite, passivity is ensured for the closed-loop system. If it is positive definite, stability is also ensured. For asymptotic stability, one may refer to LaSalle's theorem.

The potential functions are designed as follows:

$$V_{o,t}(\theta) = \frac{1}{2}(x_o - x_{o,des})^T R_o K_{o,t} R_o^T (x_o - x_{o,des}) \quad (3.49)$$

$$V_{o,r}(\theta) = 2\epsilon_b^T K_{o,r} \epsilon_b \quad (3.50)$$

$$V_{Conn}(\theta) = \frac{1}{2} \sum_{i=1}^N K_{Conn,i} [||\Delta x_i|| - l_{i,des}]^2 \quad (3.51)$$

where  $x_{o,des}$  and  $\epsilon_b$  are the desired position and the rotation error of the virtual frame, represented by the vector part of the unit quaternion obtained from the product  $R_o^T R_{o,des}$ , with  $R_{o,des}$  being the desired rotation matrix.  $K_{o,t}$  and  $K_{o,r}$  are the stiffness matrices for linear displacement and rotation with respect to the object frame, and  $K_{Conn} = diag\{K_{Conn,1}, K_{Conn,2}, \dots, K_{Conn,N}\}$  is the stiffness matrix of the virtual connection springs between the fingertips and the frame center. All three matrices are chosen positive definite. The stiffness components resulting from the potentials (3.49)–(3.51) are illustrated in Figure 3.6.

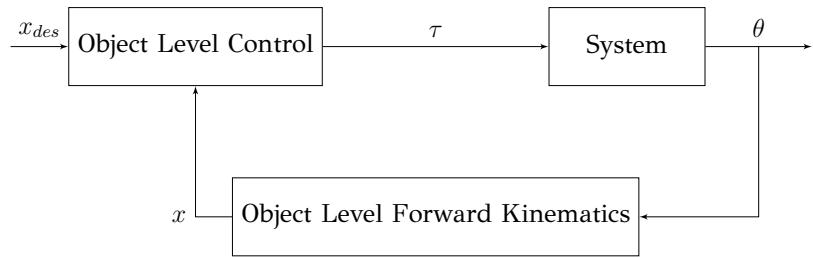


Figure 3.5: Intrinsically Passive Control scheme [29]

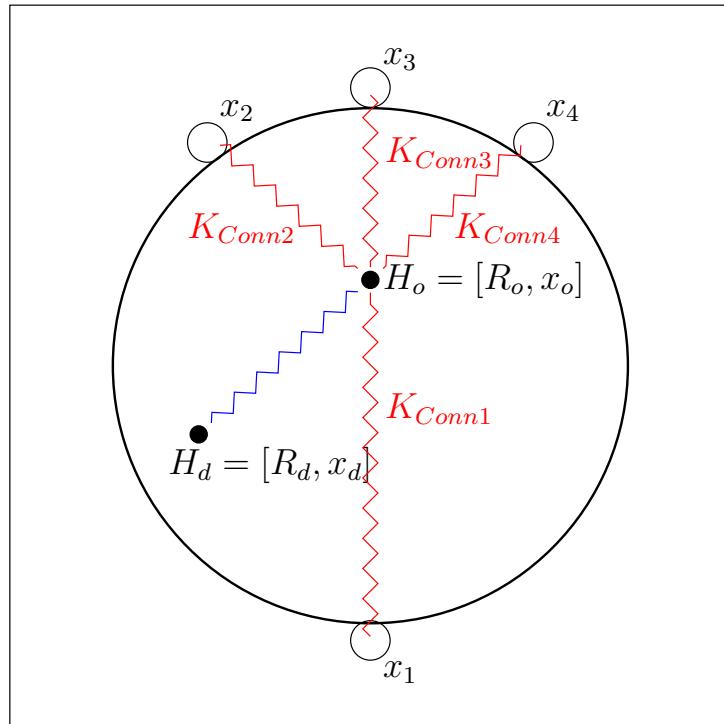


Figure 3.6: The fingertips  $x = [x_1, x_2, x_3, x_4]^T$  hold a spherical/cylindrical object using the virtual springs that define the control, as described in (3.48)

Next, we present in detail how the partial derivatives of the individual potentials with respect to the joint angles  $\theta$  are computed for inclusion in the control scheme, starting from the rotational-stiffness potential.

### 3.3.3 Rotational Stiffness

The relationship between the time derivative and the partial derivative  $\frac{\partial V_{o,r}(\theta)}{\partial \theta}$  is:

$$\dot{V}_{o,r}(\theta) = \dot{\theta}^T \frac{\partial V_{o,r}(\theta)}{\partial \theta}. \quad (3.52)$$

Differentiating in time the potential  $V_{o,r}$  as defined in (3.50) yields:

$$\dot{V}_{o,r}(\theta) = 4\epsilon_b^T K_{o,r} \dot{\epsilon}_b. \quad (3.53)$$

From the quaternion rate  $\dot{\epsilon}_b$ , we can obtain the relation that yields the corresponding angular velocity  $\omega_o$  of the frame. The relation implementing this differential mapping is:

$$\dot{\epsilon}_b = J_{\omega\epsilon}(\theta)\omega_o \quad (3.54)$$

The conversion from a rotation matrix  $R$  to the corresponding quaternion  $\epsilon = [\epsilon_0 \ (\epsilon_{b1} \ \epsilon_{b3} \ \epsilon_{b3})]$  is not unique; a special selection algorithm is used each time depending on the values of the elements of the rotation matrix [33]. The Jacobian matrix  $J_{\omega\epsilon}(\theta)$  is [33]:

$$J_{\omega\epsilon}(\theta) = \frac{1}{2} \begin{bmatrix} \epsilon_0 & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \epsilon_0 & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \epsilon_0 \end{bmatrix}^T \quad (3.55)$$

For the angular velocity of the frame, the following relation holds:

$$[\omega_o \times] = R_o^T \dot{R}_o \quad (3.56)$$

We expand the second term using the chain rule so that the differential kinematic relations between virtual frame–fingertips and fingertips–joints appear explicitly:

$$\dot{R}_o = \frac{\partial R_o}{\partial x^T} \frac{\partial x}{\partial \theta^T} \dot{\theta} \quad (3.57)$$

The term  $\frac{\partial R_o}{\partial x^T}$  can be computed by partially differentiating (3.43), (3.44), and (3.45) with respect to the vector  $x^T$ . The term  $\frac{\partial x}{\partial \theta^T}$  is the hand Jacobian matrix  $J_H = \frac{\partial x}{\partial \theta^T}$  (3.10). Finally, we obtain:

$$\omega_o = J_{o,r}(\theta)\dot{\theta}. \quad (3.58)$$

Substituting into (3.54) and then into (3.53) gives:

$$\dot{V}_{o,r}(\theta) = \dot{\theta}^T J_{o,r}^T 4J_{\omega\epsilon}^T K_{o,r} \epsilon_b \quad (3.59)$$

and from (3.52) and (3.59) we finally obtain:

$$\frac{\partial V_{o,r}(\theta)}{\partial \theta} = J_{o,r}^T 4 J_{\omega e}^T K_{o,r} \epsilon_b \quad (3.60)$$

### 3.3.4 Translational Stiffness

This time, we differentiate the translational-displacement potential term (3.49) with respect to time:

$$\dot{V}_{o,t}(\theta) = (x_o - x_{o,des})^T R_o K_{o,t} \frac{d}{dt} \{R_o^T (x_o - x_{o,des})\} \quad (3.61)$$

We expand the term  $\frac{d}{dt} \{R_o^T (x_o - x_{o,des})\}$ :

$$\frac{d}{dt} \{R_o^T (x_o - x_{o,des})\} = \dot{R}_o^T (x_o - x_{o,des}) + R_o^T \frac{\partial x_o}{\partial x^T} \frac{\partial x^T}{\partial \theta^T} \dot{\theta} \quad (3.62)$$

From (3.56) and (3.58), in combination with properties of skew matrices  $[v \times]^T = -[v \times]$  and  $[v \times]w = -[w \times]v$ , we obtain a common factor of the joint angular velocity  $\dot{\theta}$ :

$$\frac{d}{dt} \{R_o^T (x_o - x_{o,des})\} = [(R_o^T (x_o - x_{o,des})) \times] J_{o,r} + R_o^T \frac{\partial x_o}{\partial x^T} \frac{\partial x^T}{\partial \theta^T} \dot{\theta} \quad (3.63)$$

The partial derivative of the frame position with respect to fingertip positions from (3.42) is  $\frac{\partial x_o}{\partial x^T} = \frac{1}{N} I_{3N \times 3}$ , and  $\frac{\partial x^T}{\partial \theta^T}$  is the hand Jacobian matrix  $J_H = \frac{\partial x}{\partial \theta^T}$ . Therefore, the derivative of the translational-displacement potential becomes:

$$\dot{V}_{o,t}(\theta) = \dot{\theta}^T J_{o,t}^T K_{o,t} R_o^T (x_o - x_{o,des}) \quad (3.64)$$

from which the partial derivative (the term added to the joint-torque control signal) is:

$$\frac{\partial V_{o,t}(\theta)}{\partial \theta} = J_{o,t}^T K_{o,t} R_o^T (x_o - x_{o,des}) \quad (3.65)$$

with:

$$J_{o,t} = [\frac{1}{N} J_H^T I_{3N \times 3} R_o - J_{o,r}^T ((R_o^T (x_o - x_{o,des})) \times)]^T \quad (3.66)$$

Summing (3.65) and (3.60) essentially forms in space a 6D spring, with respect to linear and rotational displacements, associated with the object.

### 3.3.5 Connection Stiffness

For controlling the internal forces (grasping forces), virtual springs are defined between the fingertips  $x_i$  and the position  $x_o$  of the virtual frame, with an adjustable equilibrium point  $l_{i,des}$  so as to satisfy friction constraints (see (3.48)).

The virtual potential for the connection-stiffness term is formed spherically symmetric with respect to the fingertips:

$$V_{conn}(\theta) = \frac{1}{2} \sum_{i=1}^N K_{conn,i} [\|\Delta x_i\| - l_{i,des}]^2 \quad (3.67)$$

where  $\Delta x_i = x_i - x_o$  and  $K_{conn,i}$  is always chosen positive. Partially differentiating with respect to the joint angles  $\theta$  yields:

$$\frac{\partial V_{Conn}(\theta)}{\partial \theta} = J_H^T \sum_{i=1}^N \left[ \frac{\partial \Delta x_i^T}{\partial x} \frac{K_{conn,i} (\|\Delta x_i\| - l_{i,des})}{\|\Delta x_i\|} \Delta x_i \right] \quad (3.68)$$

The partial derivative  $\frac{\partial \Delta x_i^T}{\partial x}$  is computed as:

$$\begin{aligned} \frac{\partial \Delta x_1^T}{\partial x} &= [(1 - \frac{1}{N})I_{3 \times 3} \quad (-\frac{1}{N})I_{3 \times 3} \quad \dots \quad (-\frac{1}{N})I_{3 \times 3}] \\ \frac{\partial \Delta x_2^T}{\partial x} &= [(-\frac{1}{N})I_{3 \times 3} \quad (1 - \frac{1}{N})I_{3 \times 3} \quad \dots \quad (-\frac{1}{N})I_{3 \times 3}] \\ &\vdots \\ \frac{\partial \Delta x_N^T}{\partial x} &= [(-\frac{1}{N})I_{3 \times 3} \quad (-\frac{1}{N})I_{3 \times 3} \quad \dots \quad (1 - \frac{1}{N})I_{3 \times 3}] \\ &\quad \begin{matrix} 1 & 2 & \dots & N \end{matrix} \end{aligned} \quad (3.69)$$

Finally, the connection-stiffness term can be written more compactly in matrix form as:

$$\frac{\partial V_{Conn}(\theta)}{\partial \theta} = J_{conn}^T K_{Conn} \begin{bmatrix} \|\Delta x_1\| - l_{1,des} \\ \vdots \\ \|\Delta x_N\| - l_{N,des} \end{bmatrix} \quad (3.70)$$

with:

$$J_{Conn}^T = J_H^T \left[ \frac{\partial \Delta x_1^T}{\partial x} \frac{\Delta x_1}{\|\Delta x_1\|} \quad \dots \quad \frac{\partial \Delta x_N^T}{\partial x} \frac{\Delta x_N}{\|\Delta x_N\|} \right] \quad (3.71)$$

and:

$$K_{Conn} = blockdiag\{K_{Conn,1}, \dots, K_{Conn,N}\} \quad (3.72)$$

### 3.3.6 Control of Redundant Degrees of Freedom

Based on the controller design so far, a complete specification is essentially provided only for the fingertip positions  $x_i(\theta_i)$ . Since  $x_i$  is a vector in  $\mathbb{R}^3$  and each finger has 4 degrees of freedom, the robotic configuration is kinematically redundant by 1 degree of freedom.

Exploiting this fact, we define a subtask in the null space of the robotic finger in order to introduce an anthropomorphic coupling between the last two joints  $\theta_{i3}$  and  $\theta_{i4}$ , aiming to ensure a perpendicular projection of the fingertip onto the object surface, thereby maximizing the friction cone.

The desired control for redundant configurations is formed as:

$$\tau_d = \tau_{d,cart} + N(\theta)\tau_{d,N} \quad (3.73)$$

where  $\tau_{d,cart}$  is the desired Cartesian-level control at the fingertips (as defined in the previous subsections via the stiffness of the virtual frame),  $N(\theta)$  is the projection matrix onto the null space of the robotic grasp, and:

$$\tau_{d,N} = -K_N(\theta - \theta_N) - D_N\dot{\theta} \quad (3.74)$$

is the PD control law for the desired subtask.

The projection matrix can be obtained statically:

$$N(\theta) = V(\theta)^T V(\theta) \quad (3.75)$$

where for the matrix  $V(\theta)$  it holds:

$$V(\theta)J^T(\theta) = 0 \quad (3.76)$$

$V(\theta)$  is essentially a nullifying matrix of the Jacobian and can be computed in practice via singular-value decomposition of the Jacobian. Practically, it turns out that this type of static projection onto the Jacobian null space ultimately affects the Cartesian space of the end-effector [34]. To obtain dynamically consistent behavior, we first transfer the hand dynamic equation (3.17) from joint space to fingertip space. For fingertip velocity and acceleration, we have:

$$\dot{x} = J(\theta)\dot{\theta} \quad (3.77)$$

$$\ddot{x} = J(\theta)\ddot{\theta} + \dot{J}(\theta)\dot{\theta} \quad (3.78)$$

Substituting into (3.17) gives:

$$\ddot{x} - \dot{J}(\theta)\dot{\theta} + J(\theta)M^{-1}(\theta)(C(\theta, \dot{\theta})\dot{\theta} + g(\theta)) = J(\theta)M^{-1}(\theta)\tau_d \quad (3.79)$$

Thus, it follows that to ensure dynamically consistent null-space control, the projection matrix must satisfy:

$$J(\theta)M(\theta)^{-1}N(\theta) = 0 \quad (3.80)$$

Finally, the solution we use for the dynamic null-space projection matrix [35] is:

$$N(\theta) = (I - J^T(\theta)\Lambda(\theta)J(\theta)M^{-1}(\theta)) \quad (3.81)$$

where:

$$\Lambda(\theta) = (J(\theta)M^{-1}(\theta)J(\theta)^T)^{-1} \quad (3.82)$$

The null-space control law is designed, as mentioned, for coupling the last two degrees of freedom, as follows:

$$\tau_{d,N} = -K_{Null}(\theta_{i3} - \alpha\theta_{i4}) - D_{Null}\dot{\theta} \quad (3.83)$$

where  $\alpha \in [0, 1]$  is the coefficient by which we can tune the coupling relationship between  $\theta_{i3}$  and  $\theta_{i4}$ .

### 3.3.7 Damping-Term Design

In order to determine the transient behavior of the system, the dynamic damping term  $D(\theta)$  of the controller is designed based on the following methodology [34, 36, 27].

First, based on the differential mappings between the joint angular velocities  $\dot{\theta}$  and the workspace-variable velocities  $\dot{x}_o, \omega_{o,0}, \|\Delta\dot{x}\|$  as described by (3.66), (3.58), and (3.71), respectively, these differential relations can be grouped as:

$$\underbrace{\begin{bmatrix} \dot{x}_o \\ \omega_{o,0} \\ \|\Delta\dot{x}\| \end{bmatrix}}_{\dot{\bar{x}}} = \underbrace{\begin{bmatrix} J_{o,t} \\ J_{o,r} \\ J_{conn} \end{bmatrix}}_{J_{tot}} \dot{\theta} \quad (3.84)$$

defining  $\dot{\bar{x}} \in \mathbb{R}^{6+N}$  as the generalized differential coordinate vector and  $J_{tot}$  as the compact Jacobian.

In (3.77), (3.78), and (3.79), we see the transformation of the hand dynamic model from joint-angle space to the Cartesian space of position variables. In the present case, considering  $\bar{x}$  as the generalized position variables of the robotic grasping system, we similarly transform and obtain:

$$\ddot{\bar{x}} - J_{tot}(\theta)\dot{\theta} + J_{tot}(\theta)M^{-1}(\theta)(C(\theta, \dot{\theta})\dot{\theta} + g(\theta)) = J_{tot}(\theta)M^{-1}(\theta)\tau \quad (3.85)$$

With the control law as defined for the system via (3.47), (3.60), and (3.71), we have:

$$\ddot{\bar{x}} - J_{tot}(\theta)\dot{\theta} + J_{tot}(\theta)M^{-1}(\theta)(C(\theta, \dot{\theta})\dot{\theta} + g(\theta)) = J_{tot}(\theta)M^{-1}(\theta)(-D(\theta)\dot{\theta} - \frac{\partial V_d}{\partial \theta}(\theta) + g(\theta)) \quad (3.86)$$

Defining the following relation between joint-space damping and damping in generalized workspace coordinates:

$$D(\theta) = J_{tot}^T D_{\bar{x}}(\theta) J_{tot} \quad (3.87)$$

we finally obtain:

$$\ddot{\bar{x}} - J_{tot}(\theta)\dot{\theta} + J_{tot}(\theta)M^{-1}(\theta)(C(\theta, \dot{\theta})\dot{\theta}) = J_{tot}(\theta)M^{-1}J_{tot}^T(\theta)(-D_{\bar{x}}\dot{\bar{x}} - F_{\bar{x}}) \quad (3.88)$$

where  $F_{\bar{x}} = [f_o \ m_o \ f_{Conn}]^T = K_{\bar{x}}(\bar{x} - \bar{x}_{des}) = K_{\bar{x}}e_{\bar{x},o}$  with  $K_{\bar{x}} = \text{blockdiag}\{K_{o,t} \ K_{o,r} \ K_{Conn}\}$ .

For simplicity, only genuine second-order inertial phenomena are considered for designing the dynamic damping matrix, omitting the Coriolis terms that include products between joint angular velocities. The previous relation (3.88) is simplified as:

$$M_{H,\bar{x}}(\theta)\ddot{\bar{x}} = -D_{\bar{x}}\dot{\bar{x}} - F_{\bar{x}} \quad (3.89)$$

where:

$$M_{H,\bar{x}}(\theta) = (J_{tot}M(\theta)^{-1}J_{tot}^T)^{-1} \quad (3.90)$$

At this point, we can also introduce the dynamic effect of the object,  $F_o = M_o[\ddot{x}_o \ \omega_o \ O_{N \times 6}]^T$  (assuming that the object's gravitational force is compensated as explained in the next subsection), as an external force, so that:

$$M_{H,\bar{x}}(\theta)\ddot{\bar{x}} = -D_{\bar{x}}\dot{\bar{x}} - F_{\bar{x}} - F_o \quad (3.91)$$

and finally:

$$M_{\bar{x}}(\theta)\ddot{\bar{x}} = -D_{\bar{x}}\dot{\bar{x}} - F_{\bar{x}} \quad (3.92)$$

where:

$$M_{\bar{x}}(\theta) = M_{H,\bar{x}}(\theta) + \begin{bmatrix} I_{6 \times 6} \\ 0_{N \times 6} \end{bmatrix} M_o[I_{6 \times 6}, 0_{6 \times N}] \quad (3.93)$$

The tracking-error dynamics follow from (3.92) for zero external forces:

$$M_{\bar{x}}(\theta)e_{\bar{x},o}'' + D_{\bar{x}}e_{\bar{x},o}' + K_{\bar{x}}e_{\bar{x},o} = 0 \quad (3.94)$$

At this point, the damping term is chosen as a function of the inertia matrix  $M_{\bar{x}}$  and the proportional gain  $K_{\bar{x}}$  so that dynamically consistent damping results for all possible configurations.

Based on the double diagonalization methodology from linear algebra theory, it follows that for a positive definite symmetric  $n \times n$  matrix  $M_{\bar{x}}$  and a symmetric  $n \times n$  matrix  $K_{\bar{x}}$ , there exists a full-rank matrix  $Q$  such that  $M_{\bar{x}} = QQ^T$  and  $K_{\bar{x}} = QK_{d0}Q^T$  for an arbitrary diagonal matrix  $K_{d0}$ .

Choosing the damping matrix:

$$D_{\bar{x}}(\theta) = 2Q(\theta)D_{\xi}K_{d0}^{1/2}Q(\theta)^T \quad (3.95)$$

the tracking-error dynamics become:

$$Q(\theta)Q(\theta)e_{\bar{x},o}'' + 2Q(\theta)D_{\xi}K_{d0}^{1/2}Q(\theta)^T e_{\bar{x},o}' + Q(\theta)K_{d0}Q(\theta)^T e_{\bar{x},o} = 0 \quad (3.96)$$

Factoring out  $Q(\theta)$  and mapping the system to the new generalized-variable space  $w = Q^T(\theta)e_{\bar{x},o}$  yields the desired system behavior:

$$\ddot{w} + 2D_{\xi}K_{d0}^{1/2}\dot{w} + K_{d0}w = 0 \quad (3.97)$$

It follows that the diagonal elements of the inertia matrix  $M_{\bar{x}}$  dominate, essentially affecting the inertial characteristics of the system. Considering only the diagonal elements, we have:

$$M_{\bar{x}}(\theta) = (\text{blockdiag}\{\text{diag}\{J_{tot}M(\theta)^{-1}J_{tot}^T\}\})^{-1} + \begin{bmatrix} I_{6 \times 6} \\ 0_{N \times 6} \end{bmatrix} M_o[I_{6 \times 6}, 0_{6 \times N}] \quad (3.98)$$

Thus, we essentially choose  $Q(\theta) = \sqrt{M_{\bar{x}}}$ .

Finally, the damping term that determines the transient dynamic behavior of the closed system is selected by defining the following diagonal elements:

$$D_{\bar{x},ii}(\theta) = 2\xi_i \sqrt{M_{\bar{x},ii}(\theta)K_{\bar{x},ii}} \quad (3.99)$$

where  $\xi_i \in [0, 1]$  is the desired damping ratio for the corresponding generalized variable  $\bar{x}_i$ , and  $K_{\bar{x},ii}$  are the diagonal elements of the gain matrix  $K_{\bar{x}} = \text{blockdiag}\{K_{o,t}, K_{o,r}, K_{Conn}\}$ .

### 3.4 Extensions and Modifications of the Object-Level Impedance Controller

Internal-force control based on the connection-stiffness scheme  $V_{Conn}$ , as described in the previous subsection, yields satisfactory performance and robust, stable grasping [27, 28, 29]. This is particularly impressive considering that the system is essentially “blind,” since it does not rely on any type of sensory feedback (force–sensor–tactile), with the only assumption being that the manipulated objects will have regular 3D geometries.

The criticism that can be made of this internal-force control scheme consists of two main points:

- Particularly increased requirements on the contact friction cone in the case of manipulating objects with planar surfaces, as shown in Figure 3.7, since the direction of the virtual spring  $V_{Conn,i}$  tends to become tangential to the object surface.
- Practically difficult real-time tuning of the equilibrium points  $l_{i,des}$  for each connection  $V_{Conn,i}$  for different objects, resulting in a possible resultant force on the object and, consequently, a final steady-state positioning error.

#### 3.4.1 Internal-Force Control Based on Surface Characteristics (IPC-IF)

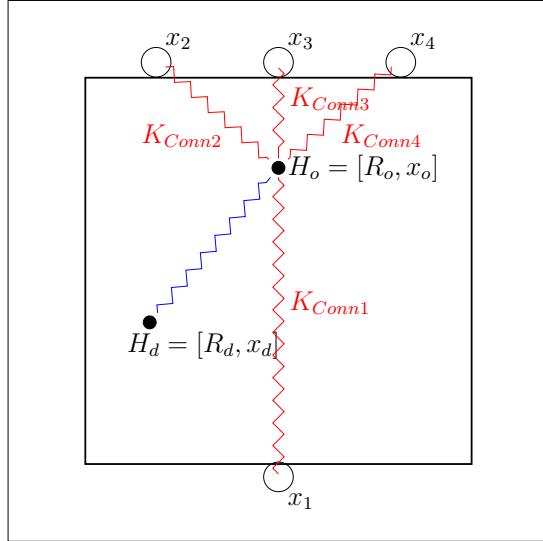


Figure 3.7: Large tangential force with respect to the object surface at fingertips  $x_2$  and  $x_3$  due to the spherically symmetric connection stiffnesses  $K_{Conn2}$  and  $K_{Conn4}$

The main argument in our approach is that in real integrated robotic-manipulation applications, there exist at least one or more sensor suites (Force, Tactile, Vision)

capable of providing information about the object surface, and specifically the direction normal to the surface.

Thus, we propose an extension of the Intrinsically Passive Controller (IPC) by applying internal-force control through exerting forces from the fingertips in a direction normal and inward with respect to the object and the surface defined by the contacts (IPC-IF) (see Figure 3.8).

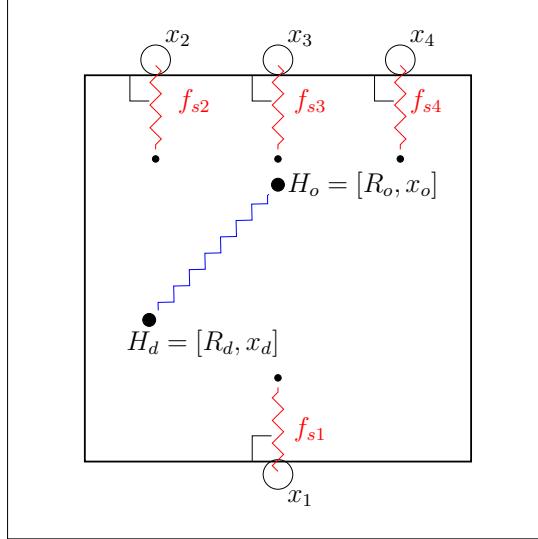


Figure 3.8: Intrinsic Passivity Control with internal-force control by applying normal pressure on the surfaces defined by the contacts (IPC-IF).

For appropriate tuning of these internal forces, we propose projection onto the null space of the grasp matrix.

In the static analysis (3.103), we have:

$$\begin{bmatrix} R_o^T & R_o^T & R_o^T & R_o^T \\ [r_{c1} \times] R_o^T & [r_{c2} \times] R_o^T & [r_{c2} \times] R_o^T & [r_{cN} \times] R_o^T \end{bmatrix} \begin{bmatrix} F_{s1} \\ F_{s2} \\ F_{s2} \\ F_{sN} \end{bmatrix} = \begin{bmatrix} -f_{ext} \\ -n_{ext} \\ \vdots \end{bmatrix} \quad (3.100)$$

with the vector  $r_{ci}$  defined as:

$$r_{ci} = \frac{x_i - x_o}{\|x_i - x_o\|} \quad (3.101)$$

The internal forces perpendicular to the object are:

$$F_s = n f_s \quad (3.102)$$

where  $n = \text{blockdiag}\{n_1, n_2, n_3, n_4\} \in \mathbb{R}^{12 \times 4}$  are the unit vectors  $n_i$  normal to the surface

at contact  $i$ , respectively. Thus, from (??), we have:

$$\underbrace{\begin{bmatrix} R_o^T n_1 & R_o^T n_2 & R_o^T n_3 & R_o^T n_4 \\ [r_{c1} \times] R_o^T n_1 & [r_{c2} \times] R_o^T n_2 & [r_{c3} \times] R_o^T n_3 & [r_{cN} \times] R_o^T n_4 \end{bmatrix}}_G \underbrace{\begin{bmatrix} f_{s1} \\ f_{s2} \\ f_{s3} \\ f_{s4} \end{bmatrix}}_{f_s} = \underbrace{\begin{bmatrix} -f_{ext} \\ -n_{ext} \end{bmatrix}}_{F_{ext}} \quad (3.103)$$

where  $G \in \mathbb{R}^{6 \times 4}$  is the resulting grasp matrix. From (3.39), the projection onto the null space of  $G$  is:

$$f_{s,Null} = \underbrace{(\mathbf{I} - [G]^+ G)}_{G_{s,Null}} f_s \quad (3.104)$$

Finally, the control law  $\tau_N$  for the internal forces becomes:

$$\tau_N(\theta) = J_H^T(\theta) (\mathbf{I} - [G]^+ G) f_s \quad (3.105)$$

### 3.4.2 Gravity Compensation

Having defined the stiffnesses of linear and rotational displacements by defining the force  $f_o$  and the impedance torque  $m_o$ , we can easily extend the existing structure by introducing gravity-compensation terms through the gravity vector  $G = [0, 0, mg]$ . The translational effect of gravity with respect to the frame  $H_o$  is:

$$f_g = R_o^T G \quad (3.106)$$

and the torque it induces is:

$$m_{o,g} = [r_{COG} \times] R_o^T G \quad (3.107)$$

where  $r_{COG}$  is the position vector of the object's center of gravity with respect to frame  $H_o$ .

We introduce these terms into (3.60) and (3.65), obtaining respectively:

$$\frac{\partial V_{o,r}(\theta)}{\partial \theta} = J_{o,r}^T \underbrace{(4J_{\omega e}^T K_{o,r} \epsilon_b + m_{o,g})}_{m_o} \quad (3.108)$$

$$\frac{\partial V_{o,t}(\theta)}{\partial \theta} = J_{o,t}^T \underbrace{(K_{o,t} R_o^T (x_o - x_{o,des}) + f_{o,g})}_{f_o} \quad (3.109)$$

The addition of gravity compensation is illustrated in Figure 3.9.

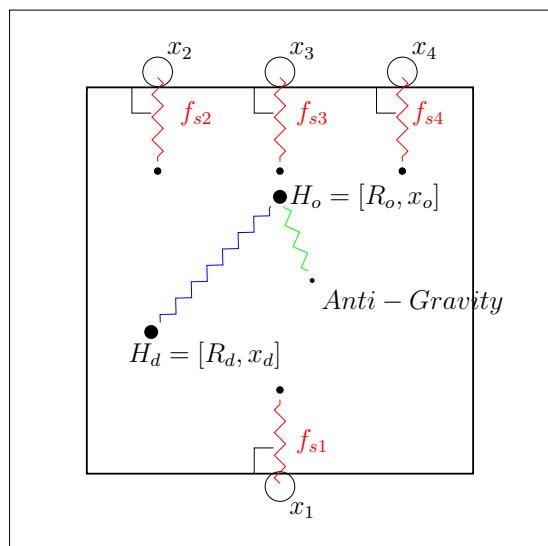


Figure 3.9: Internal-force control with normal pressure on the object surfaces in combination with gravity-compensation terms.

# Chapter 4

## Implementation & Simulation Results

In this chapter we present the implementation methodology and the results of the dynamic simulation that we carried out for the control of a robotic hand with kinematic and dynamic characteristics analogous to those of the DLR Hand 2 developed by the German Aerospace Center. First, we implement the Intrinsically Passive Control (IPC) scheme using a virtual frame, and we obtain measurements for both translational and rotational displacement scenarios, as well as for the application of external forces. Next, we introduce the proposed internal-force control scheme based on object surface characteristics (IPC–IF), repeating the same set of measurements and comparing the results against the previous case. Finally, we implement the gravitational compensation controller.

### 4.1 Construction of the simulation environment

Examining the historical development of automatic control systems, we observe that initially—and especially before their mathematical foundation began to develop after the middle of the 19th century—they were primarily empirical constructions, involving direct implementation and tuning of the controller on the final system. Although the mathematical foundations of automatic control progressed rapidly with the maturation of classical theory up to the 1950s, controller adaptation and final tuning still had, in most cases, to be carried out directly on the final system, making the process potentially dangerous, costly, and in some cases even impossible, considering that applications may occur in fields such as aerospace or nuclear technology.

To address these issues, it was evident that introducing an arrangement capable of simulating subsystems or the entire system was necessary for the testing and tuning of controllers. A characteristic and extreme example is the lunar module simulation system of the Apollo program in the 1960s, which included all vehicle systems, with the dynamic computations performed by a vacuum-tube analog computer that fully occupied a four-story building. Subsequently, digital systems—with abundant computational power, ease of programming, and the ability to reuse software—became the core for building simulation environments.

Today, simulation is common practice for the construction of any integrated system, incorporating specially tailored hardware components (Hardware in the Loop – HiL) and/or software components (Software in the Loop – SiL) in the loop, improving final quality, reducing development time and cost, and maintaining the process in a safe and controlled environment.

For the simulation of multi-body dynamics systems, there are many available software options, both professional and non-professional. Professional applications typically offer highly specialized solutions tailored to each problem and are commercially available at some cost. While most of these systems can simulate constrained kinematic mechanical structures as well as a variety of parallel physical phenomena, the list of available applications becomes significantly smaller when collision detection and handling between simulated bodies (Collision Detection & Handling) is a prerequisite. This is the main feature of physics engines, which carry out all computations for rigid-body motion and at the same time can detect and handle contact and collision phenomena between bodies. Collision-event handling is of obvious usefulness for robotics systems, especially for walking robots or manipulation systems.

In general, physics simulators can be found in applications such as:

- 3D animation applications such as games, movies, and any graphics application.
- Scientific-industrial applications for modeling and R&D of technological applications such as robotics, aerospace, and automotive engineering, as well as the development of educational vehicle-operator simulators of all types.

#### 4.1.1 DLR Hand 2

In this subsection we present some of the basic characteristics of the DLR Hand 2 robotic hand, developed by the German Aerospace Center (DLR German Space Center), which we use as a reference for constructing the simulation environment of the robotic hand, serving as the testbed for the control schemes implemented in this work.

The DLR Hand 2 is a multi-jointed anthropomorphic robotic hand with four fingers, where each finger has four degrees of freedom, with the last two being coupled. The kinematic structure of each finger is illustrated in Figure 3.2. In addition, the DLR Hand 2 has two additional degrees of freedom for moving the thumb as well as the 4th finger, which were deemed necessary to enable both power grasps and dexterous fingertip manipulation. The overall size of the robotic hand is approximately 1.5 times that of a human hand.

To realize these motions, a joint actuation system is employed, consisting of DC motors, timing belts for the coupled joints, and harmonic drives for the remaining joints. Ultimately, each finger can exert tangential fingertip forces of up to 30 N.

The sensory suite of the DLR Hand 2 consists of the following sensors on each finger:

- 3 joint position sensors, specifically designed based on an inductive synthetic potentiometer.
- 3 joint torque sensors based on measuring elastic deformation.
- 3 position–velocity sensors (analog Hall sensors) with interpolation.
- 1 six-axis force sensor at the fingertip based on measuring elastic deformation.
- 3 motor temperature sensors.
- 3 temperature sensors for temperature compensation.

The electronics of the hand are fully integrated into the main body, which was made possible through the use of flexible printed circuit boards (flexible PCBs). In each link of each finger there is at least one board: an 8-channel, 12-bit analog-to-digital converter for reading sensor signals. It is also important to note that the power electronics of each motor are placed near the motor. Thus, the DLR Hand 2 achieves complete autonomy, being capable of being mounted on any robotic arm without requiring special infrastructure.

Control is performed by an external computer. For communication between the subsystems and the external computing system, an advanced fully integrated serial control system is used, aiming to minimize internal wiring. The structure of this system is hierarchical, consisting initially of a telecommunication controller at the base of each finger that processes the signals of the corresponding finger (40 channels, 12-bit resolution), and a telecommunication controller at the base of the hand responsible for communication between each finger controller and the external computing system.

The total weight of each of the 4 identical fingers is 375 gr. The mass distribution density, assuming simplistically that it is constant over the total link length  $l_1 + l_2 + l_3 = 155\text{mm}$ , is  $d = 2.49\text{gr}/\text{mm}$ . In Table 4.1 we list the basic geometric–inertial characteristics of the DLR Hand 2.

Link $i$	Length [mm]	Mass [gr]
1	75	187
2	40	94
3	40	94

Table 4.1: Table of basic geometric–inertial characteristics of the DLR Hand 2

#### 4.1.2 Open Dynamics Engine (ODE)

A popular open-source solution for rigid-body physics simulation based on a simple and efficient API (Application Programming Interface) in C or C++ is the Open

Dynamics Engine (ODE), which is widely used in both areas mentioned above: 3D animation as well as simulators for scientific and industrial applications. In robotics in particular, it is used extensively for application development, either as part of a customized solution or as part of a complete ready-made simulation program for robotics systems, such as Webots, V-Rep, and ROS.

Some features that make ODE suitable for multi-body dynamics simulation are:

- Stable and robust numerical integration for computing body motion.
- The ability to apply torque control signals directly to the desired joints.
- Easy tuning and definition of every system parameter.

As with any physics simulator, the manufacturer makes a fundamental design trade-off between accuracy and stability—speed—robustness. ODE is a simulator clearly oriented towards stability and computational speed, and therefore special care is required when drawing conclusions from simulation results. For the same reason, ODE uses a highly simplified model for handling friction phenomena, which deviates even from the classical static friction cone model [37].

To mitigate these characteristics, we follow the rules below:

- Small world time step to improve accuracy and numerical robustness.
- We ensure that mass ratios, especially between bodies connected by joints, are as close as possible to unity, and in no case greater than 10.

The simulation parameters we selected are shown in Table 4.2. Since we use a small time step, in practice the error reduction parameter (ERP) can remain relatively low. We also assume perfectly rigid contacts by selecting a zero value for ODE’s constraint force mixing (CFM) parameter.

World Time Step	656.25 $\mu$ sec
ERP	0.2
CFM	$\sim 0$

Table 4.2: ODE parameter values

#### 4.1.3 Open Dynamics Engine S-Function Block – Robotic hand simulation

As the primary tool for developing the dexterous robotic manipulation controller, we chose Mathworks Simulink. Some reasons contributing to this choice are:

- Ready-made block libraries with standard as well as advanced automatic control tools.

- The graphical block-diagram environment enables easy modifications, easy debugging, and an overall view of all system parameters.
- Excellent flexibility for developing custom and specialized solutions, with the ability to implement blocks in C, C++, or Matlab code via the S-Function interface.
- Automated C/C++ code generation for improved performance and easy transfer of the implementation to the final real-time target machine.

We note that the reason we do not use the SimMechanics libraries and ready-made Simulink blocks for simulating our system is the absence of collision detection and handling. Although such collision handling is absent in its basic form, it is possible to implement it in a customized way using Simulink components [38].

Ultimately, the solution we propose and implement is the construction of a custom block as an S-Function MEX file that uses the ODE API to simulate the robotic system. A similar effort [39] contributed significantly in this direction.

As already mentioned, we base our model on the kinematic–dynamic characteristics of the DLR Hand 2. We simulate each robotic link using ODE’s capsule-shaped geometric primitive. The radius of each capsule is chosen small so as to approximate point contacts with friction as faithfully as possible. In its internal operation, ODE does not treat quantities with their strict physical meaning, but rather in a relative manner. Based on this, we define the absolute values of the relevant quantities according to the manufacturer’s guidelines for maximum simulation reliability, as shown in Table 4.3.

All fingers are placed such that their bases lie on the same plane, with their relative positions defined for fingers 1, 2, 3, and 4 as:  $(-0.7, 0, 0)$ ,  $(0.6, -0.4, 0)$ ,  $(0.7, 0, 0)$ , and  $(0.6, 0.4, 0)$ .

Link $i$	Length	Mass	Capsule link radius
1	1	2.49	0.04
2	0.53	1.25	0.04
3	0.53	1.25	0.02

Table 4.3: Table of basic geometric–inertial characteristics of the simulated robotic hand

#### 4.1.4 Overall implementation structure

Our system is structured in Simulink in hierarchical levels. The highest hierarchical level, which provides the overall view of the system, is shown in Figure 4.1, describing the interconnection between the main building blocks:

- Mex C S-Function Block: Open Dynamics Engine simulation of the robotic hand and environment.

- Controller.
- Object-level path planning.
- Surface feature extraction system – extraction of the object surface normal vector  $n_i$ .
- System for applying external forces and torques on the manipulated object.

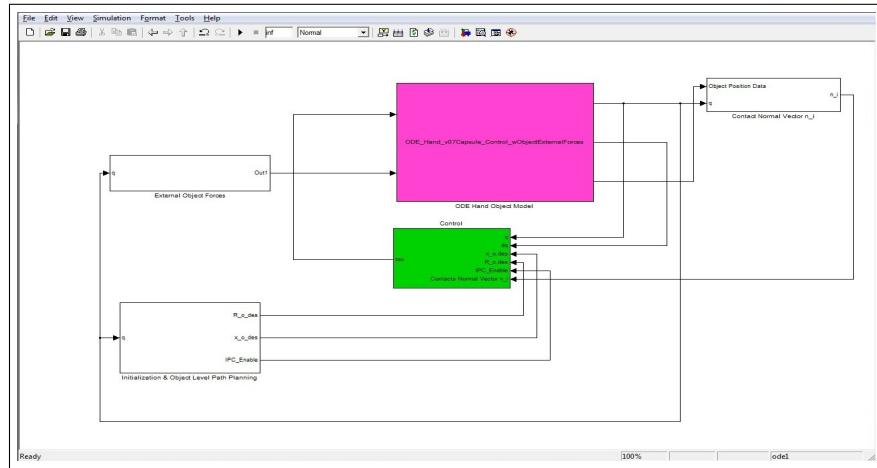


Figure 4.1: Top-level Simulink implementation view including the system model, the controller, and the auxiliary tools.

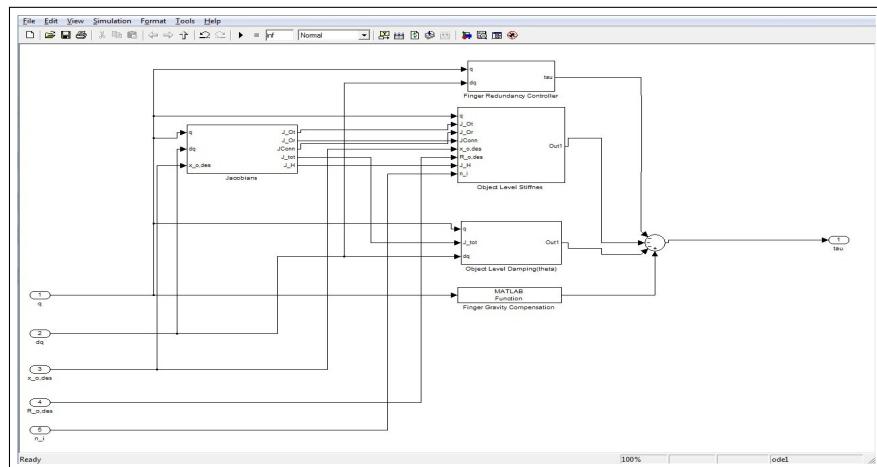


Figure 4.2: Overall view of the Simulink controller structure.

The graphical visualization is performed in real time as the Simulink simulation runs. The initial grasp configuration with a spherical object under manipulation is shown in Figure 4.4.

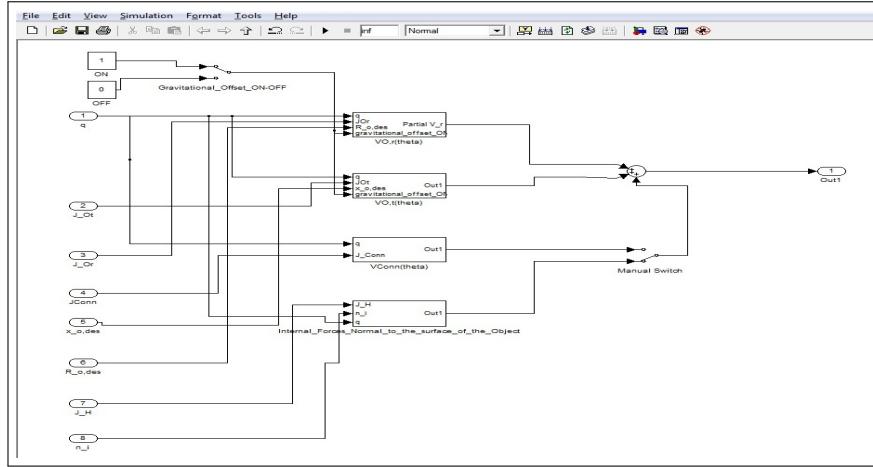


Figure 4.3: View of the Simulink internal stiffness terms (rotational and translational).

## 4.2 IPC controller simulation results

In this section we present the results obtained from applying the IPC robotic grasp controller, described in detail in Section 3.3 of the previous chapter. First, we apply passivity-based control as described in Chapter 3, including internal-force control via connection stiffness with the virtual frame. We obtain the required measurements by first exciting the system with step inputs and then by measuring stiffness through applying external forces and torques on the object.

The stiffness gains for translational–rotational motion, the damping ratio, and the equilibrium points for connection stiffness are defined as:

$$K_{o,r} = \begin{bmatrix} 700 & 0 & 0 \\ 0 & 700 & 0 \\ 0 & 0 & 700 \end{bmatrix}, \quad K_{o,t} = \begin{bmatrix} 200 & 0 & 0 \\ 0 & 200 & 0 \\ 0 & 0 & 200 \end{bmatrix}$$

$$\xi = 0.9, \quad l_{i,des} = 0.15m, \quad i = 1, 2, 3, 4$$

The gains for the null-space controller of the robotic hand are selected to be of low bandwidth, affecting the task space as little as possible, with the following values:

$$K_N = 50, \quad D_N = 0.13K_N^{1/2}$$

### 4.2.1 Step response – translational and rotational motion

First, we examine two manipulation scenarios for a sphere with diameter 0.54 and weight 0.8 (the physical values do not have strict meaning for ODE, as already mentioned). One scenario concerns rotational displacement about the vertical  $z$  axis,  $rotz(0.6\ rad)$ , and the other concerns linear displacement along the  $x$  axis,  $tranx(0.40\ m)$ . Essentially, both manipulations are step inputs to the system,

in the classical control-theory sense, highlighting fundamental characteristics of the final closed-loop system such as transient response, overshoot, steady-state position error, and oscillations.

The values of the gains  $K_{Conn}$  for connecting the fingertips to the virtual frame are:

$$K_{Conn} = \begin{bmatrix} 800 & 0 & 0 & 0 \\ 0 & 800 & 0 & 0 \\ 0 & 0 & 800 & 0 \\ 0 & 0 & 0 & 800 \end{bmatrix}$$

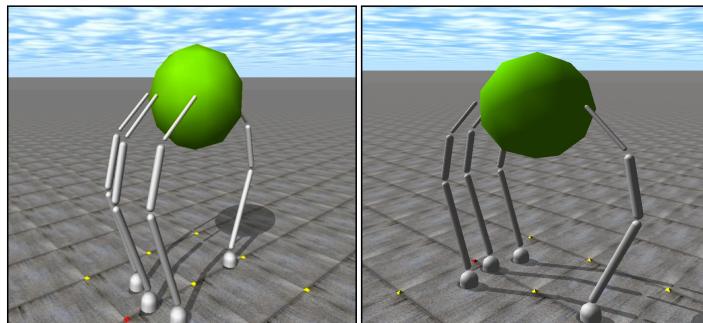


Figure 4.4: General view of the initial grasp configuration with a spherical object under manipulation.

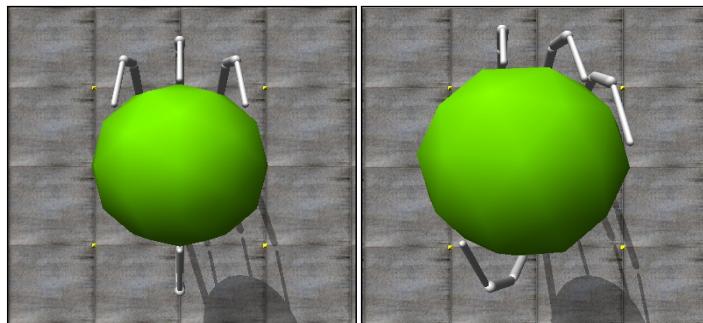


Figure 4.5: Top view – final pose after rotational displacement about the  $z$  axis (rotation 0.6 rad).

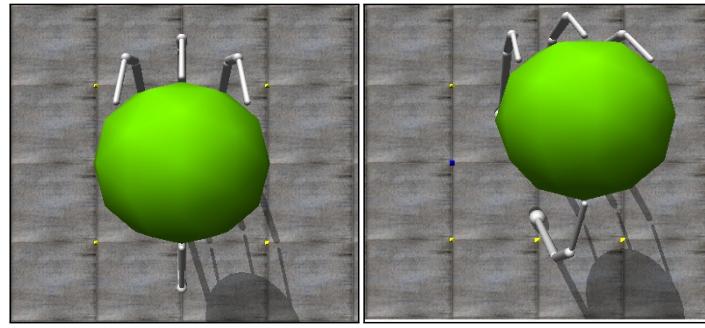


Figure 4.6: Top view – displacement along the  $x_o$  axis (translation 40 cm).

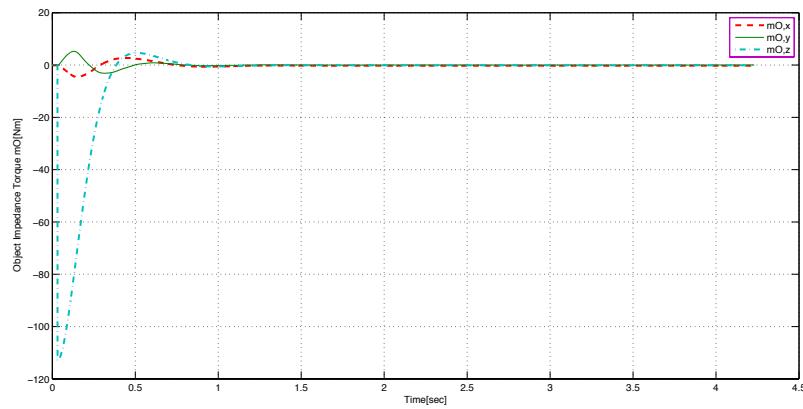


Figure 4.7: Impedance torque  $m_O$  during rotation about the  $z$  axis (rotation 0.6 rad).

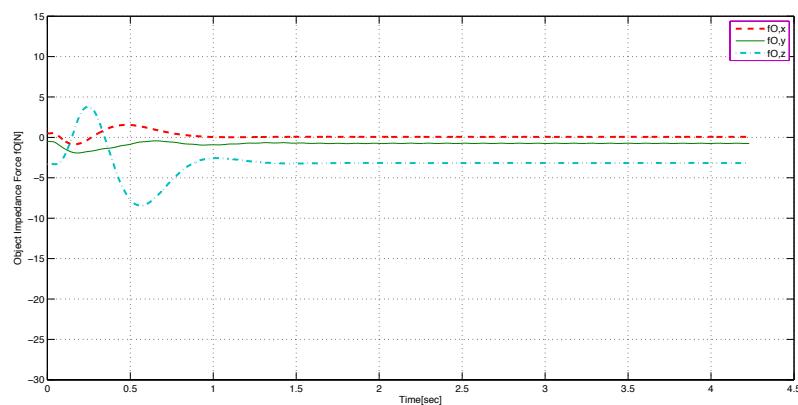


Figure 4.8: Impedance force  $f_O$  during rotation about the  $z$  axis (rotation 0.6 rad).

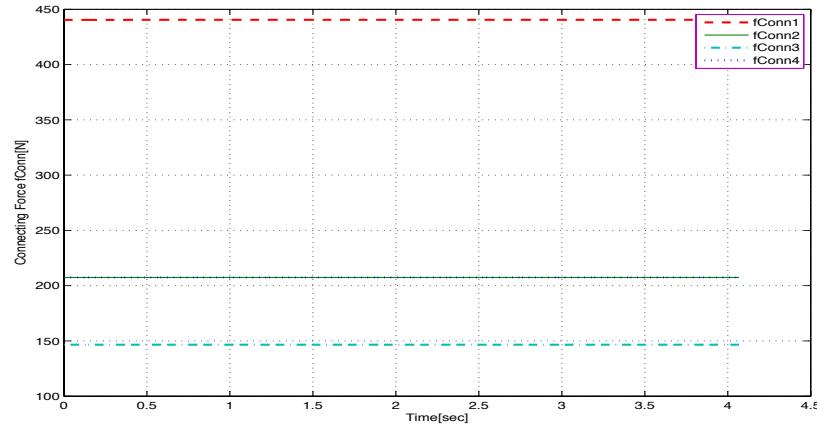


Figure 4.9: Fingertip-to-virtual-frame connection force  $f_{Conn}$  during rotation about the  $z$  axis (rotation 0.6 rad).

Figure 4.10: Applied joint torques of fingers 2,3,4,1–thumb, during rotational motion about the  $z$  axis (rotation 0.6 rad).

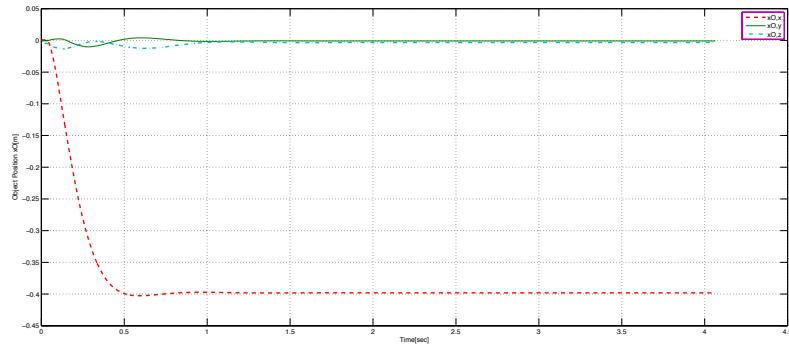


Figure 4.11: Object position  $xO$  during displacement along the  $x_o$  axis (translation 40 cm).

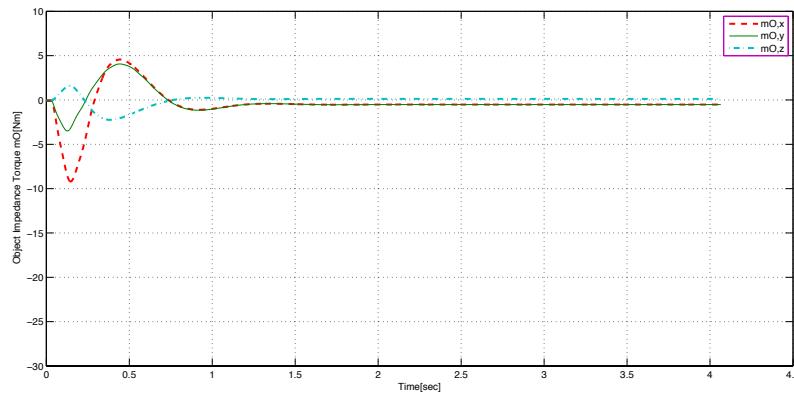


Figure 4.12: Impedance torque  $m_O$  during displacement along the  $x_o$  axis (translation 40 cm).

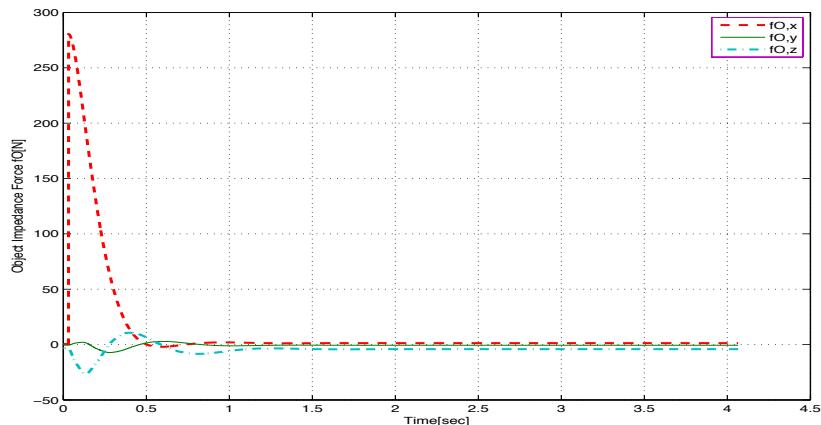


Figure 4.13: Impedance force  $f_O$  during displacement along the  $x_o$  axis (translation 40 cm).

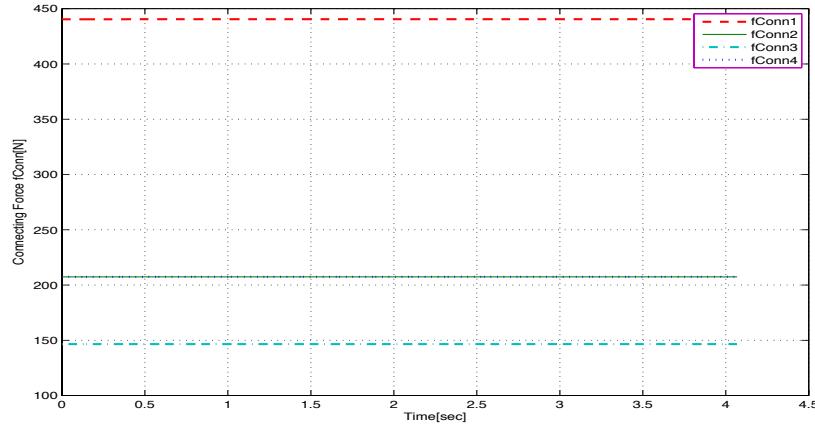


Figure 4.14: Fingertip-to-virtual-frame connection force  $f_{Conn}$  during displacement along the  $x_o$  axis (translation 40 cm).

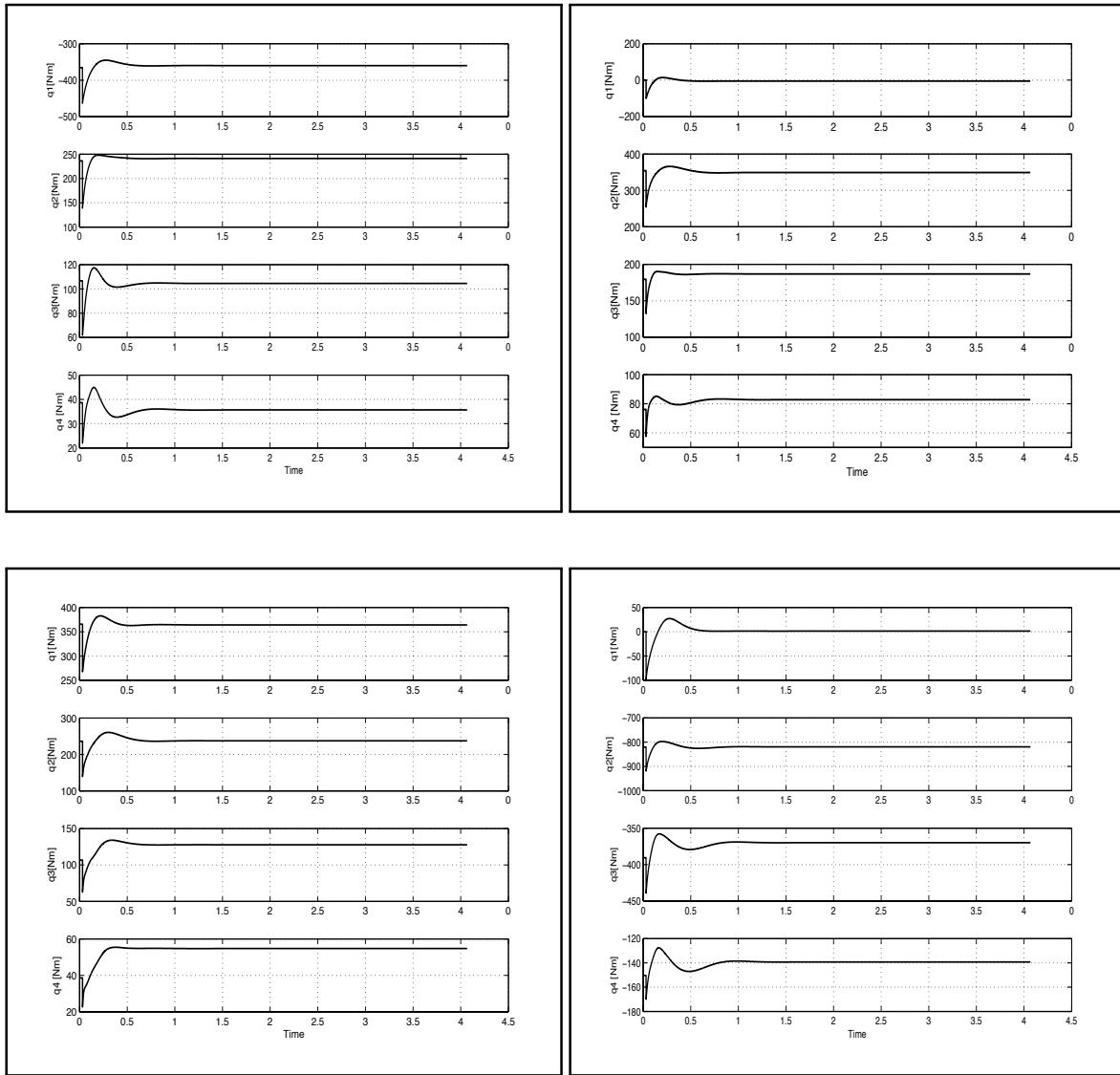


Figure 4.15: Applied joint torques of fingers 2,3,4,1–thumb, during translational motion  $x$ , 0.40m, as produced by the control law.

Based on the measurements, we observe that our system is able to execute the desired motions within the workspace successfully, with small or even zero coupling between different degrees of freedom. Our main criticism focuses on the presence of steady-state position errors in the final state, as can be seen in Figures 4.8, 4.12, and 4.13.

Specifically, we observe steady-state error:

- with respect to the impedance force  $f_o$  (3.8 N on the  $z$  axis, 0.7 N on the  $y$  axis, zero error on the  $x$  axis; see Figures 4.8, 4.13)
- with respect to the impedance torque  $m_o$  (0.1 N·m on the  $x$  axis as well as on the  $y$  axis, zero error on the  $x$  axis; see Figure 4.12).

Our initial suspicion is that the internal-force control mechanism, through the connection stiffness with the virtual frame, introduces a non-zero resultant force on the object, which shifts the static equilibrium point of the system away from the desired one.

Regarding the internal force space, it is interesting to observe in Figures 4.9, 4.14, 4.10, 4.10 the significantly stronger effort exerted by the thumb (*finger1*) at each instant.

### 4.2.2 Stiffness measurement

In this measurement scenario, we sequentially apply increasing forces/torques to the object at the frame  $H_o$  in order to extract the resulting system stiffness  $K_{\bar{x}}$ .

We begin by applying an increasing force sequentially along the  $x$ ,  $y$ , and  $z$  axes and observing the resulting translations–rotations. We increase the force by 20 N every 0.7 s. The 0.7 s interval is considered sufficient for the system to reach its final equilibrium position. Then, following the same logic, we apply a torque that increases sequentially by 3 Nm every 0.7 s along each axis.

The translational and rotational stiffness values are obtained via least-squares fitting from the measured values:

$$K_{tr,measured} = \begin{bmatrix} 697.9 & 0 & 0 \\ 0 & 703.2 & 0 \\ 0 & 0 & 710.2 \end{bmatrix}, \quad K_{rot,measured} = \begin{bmatrix} 189 & 0 & 0 \\ 0 & 195 & 0 \\ 0 & 0 & 192 \end{bmatrix}$$

We observe that these values are quite close to the values defined in the controller ( $k_{translation} = 700$ ,  $k_{rotation} = 200$ ), with small deviations but consistent linear behavior, as shown in Figure 4.18. At the same time, we observe small coupling in some directions, as can be seen in Figures 4.16, 4.17, 4.20.

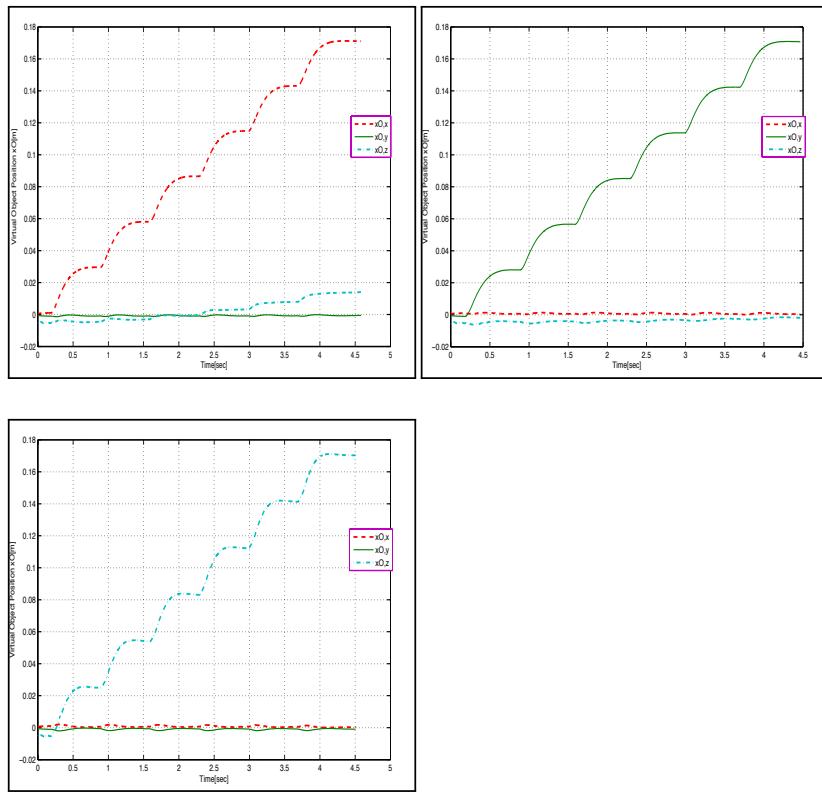


Figure 4.16: Virtual frame position  $x_o$  under applied forces along the  $x$ ,  $y$ ,  $z$  axes, respectively.

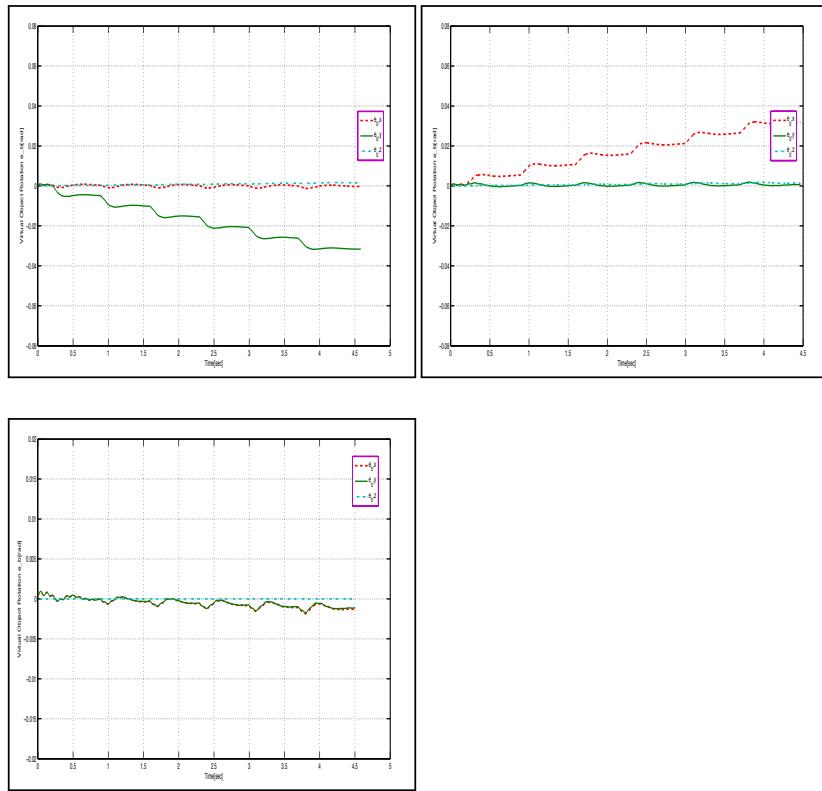


Figure 4.17: Virtual frame quaternion vector  $e_b$  under applied forces along the  $x$ ,  $y$ ,  $z$  axes, respectively.

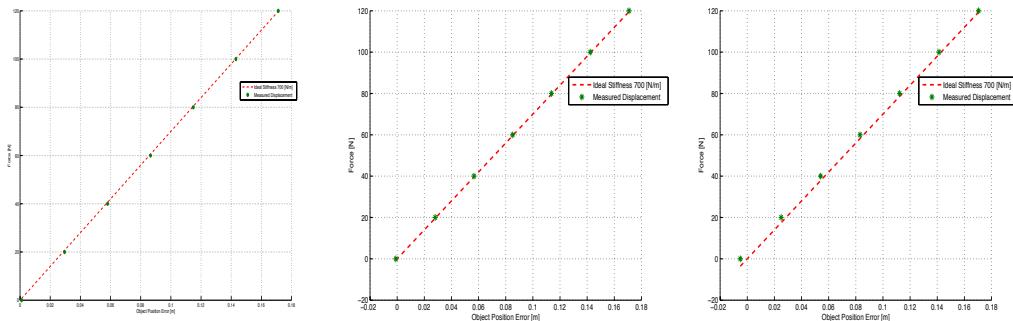


Figure 4.18: Object position  $x_o$  after applying forces along the  $x$ ,  $y$ ,  $z$  axes of frame  $H_o$ , respectively.

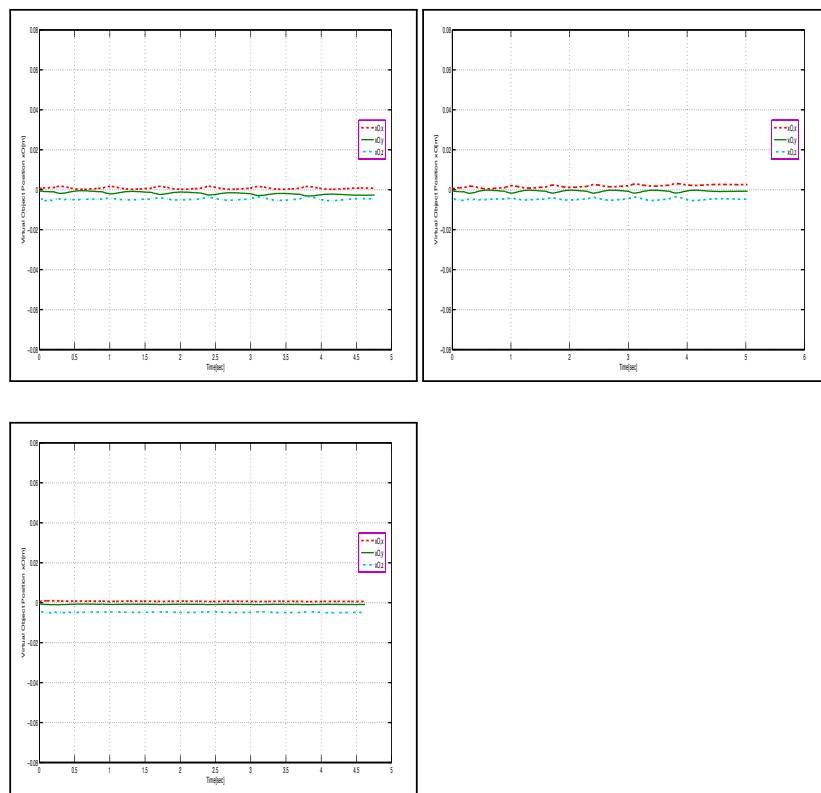


Figure 4.19: Virtual frame position  $x_o$  under applied torques along the  $x$ ,  $y$ ,  $z$  axes, respectively.

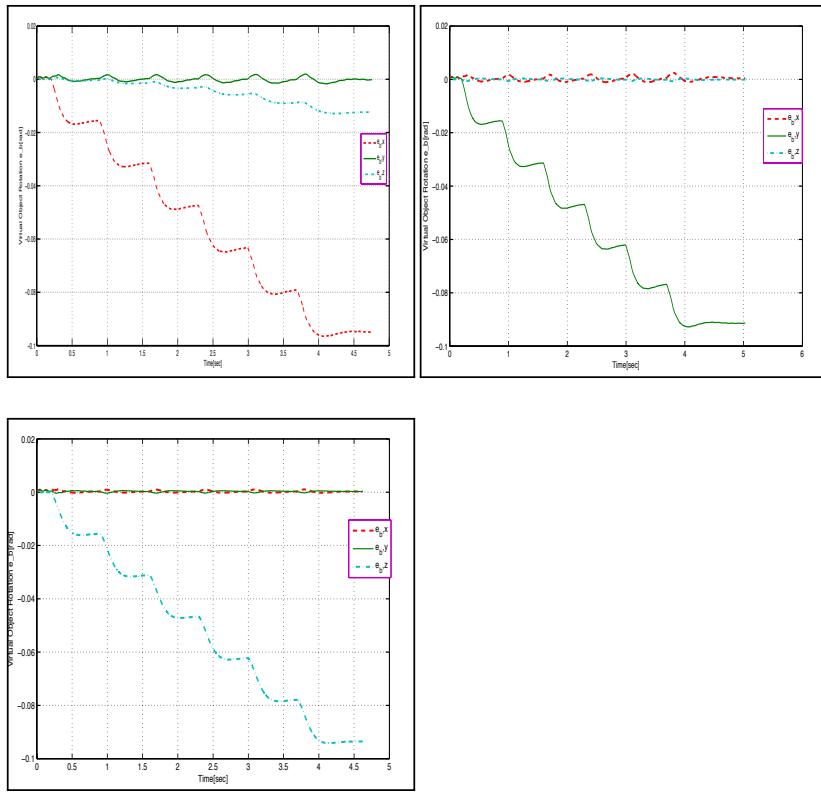


Figure 4.20: Virtual frame quaternion vector  $e_b$  under applied torques along the  $x$ ,  $y$ ,  $z$  axes, respectively.

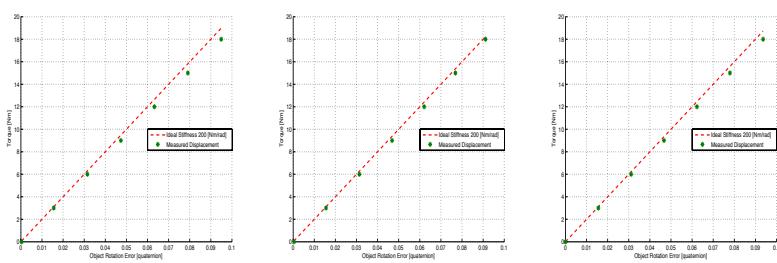


Figure 4.21: Object quaternion after applying torques along the  $x$ ,  $y$ ,  $z$  axes of frame  $H_o$ , respectively.

## 4.3 Simulation results of the IPC–IF robotic grasp controller with internal force compensation

In this section we repeat the same set of measurements/tests (step response–stiffness measurement) based on the proposed extension of IPC for internal force compensation (IPC–IF), analyzed in detail in Subsection 3.4.1. By nulling the internal force space on the object, we expect more consistent position tracking and smaller deviations in the stiffness values defined in the controller space.

### 4.3.1 Step response – translational and rotational motion

As in the previous section, we examine two manipulation scenarios for a sphere: rotational displacement about the vertical  $z$  axis,  $rotz(0.6rad)$ , and linear displacement along the  $x$  axis,  $tranx(0.40m)$ .

This time it is interesting to observe the improved characteristics with respect to reducing the final steady-state position error, as seen clearly in Figures 4.23, 4.27, 4.28, as well as the reduced coupling between different motion directions. Specifically, we observe a significant reduction of the steady-state impedance force error  $f_o$  along the  $z$  axis, from 3.8 N (with the basic IPC controller) to 2 N with the proposed IPC–IF controller.

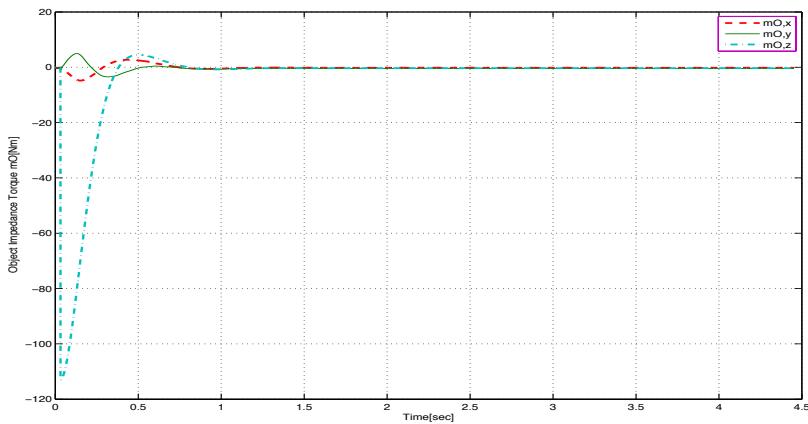


Figure 4.22: Impedance torque  $m_O$  during rotation about the  $z$  axis (rotation 0.6 rad), with the IPC–IF controller.

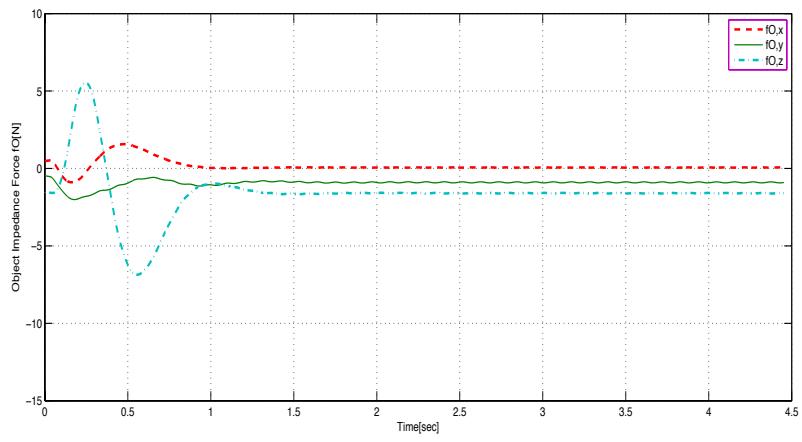


Figure 4.23: Impedance force  $f_O$  during rotation about the  $z$  axis (rotation 0.6 rad), with the IPC–IF controller.

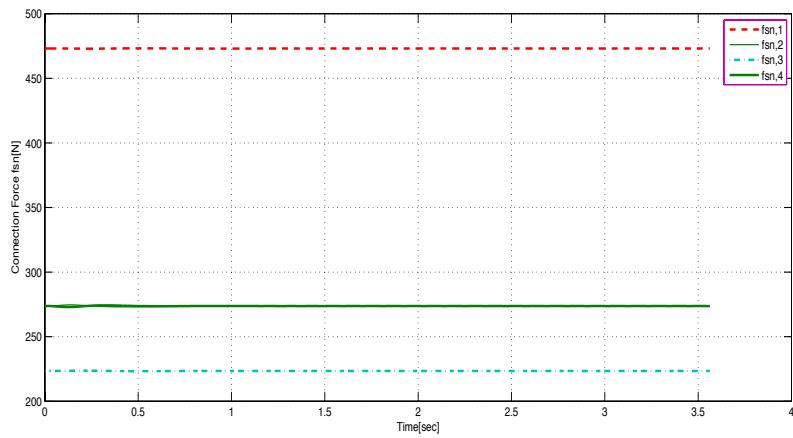


Figure 4.24: Magnitude of internal forces  $f_{sn}$  during rotation about the  $z$  axis (rotation 0.6 rad), with the IPC–IF controller.

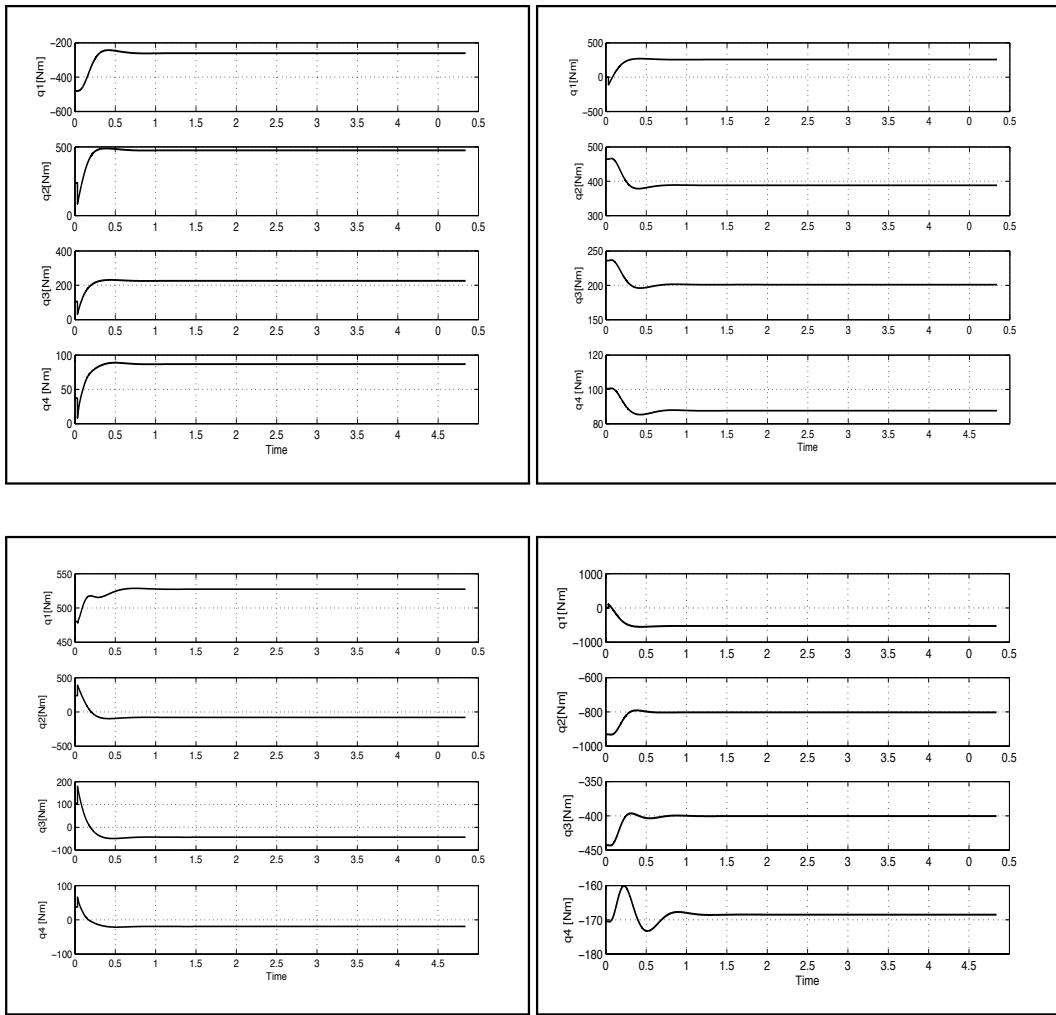


Figure 4.25: Applied joint torques of fingers 2,3,4,1–thumb, during rotational motion about the  $z$  axis (rotation 0.6 rad), with the IPC–IF controller.

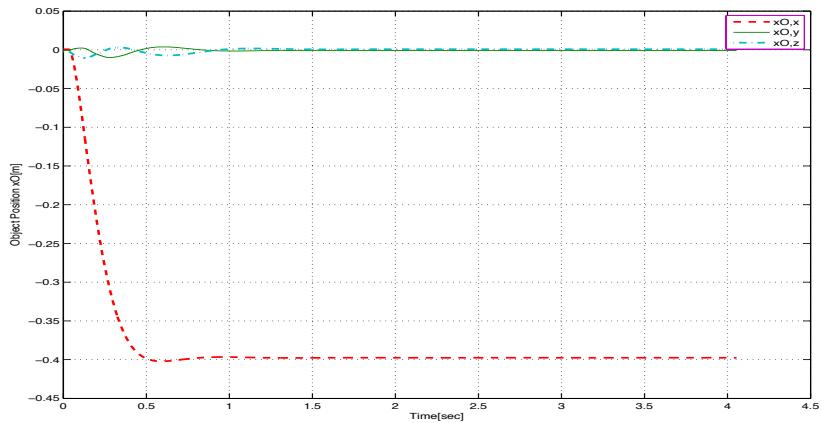


Figure 4.26: Object position  $x_O$  during displacement along the  $x_o$  axis (translation 40 cm), with the IPC–IF controller.

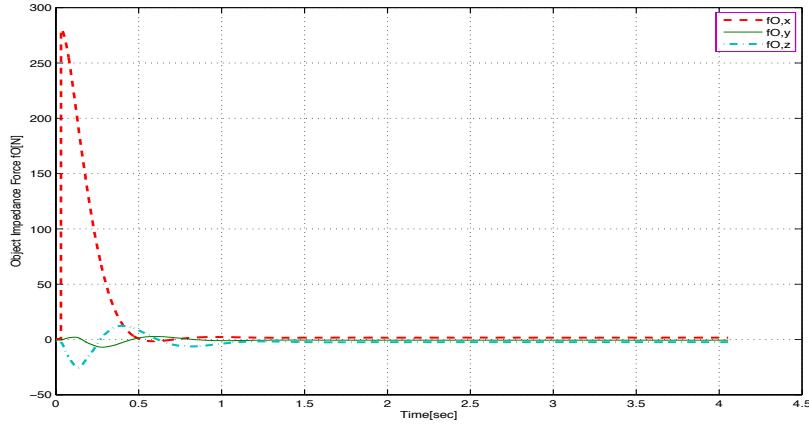


Figure 4.27: Object impedance force  $f_O$  along the  $x_o$  axis (translation 40 cm), with the IPC–IF controller.

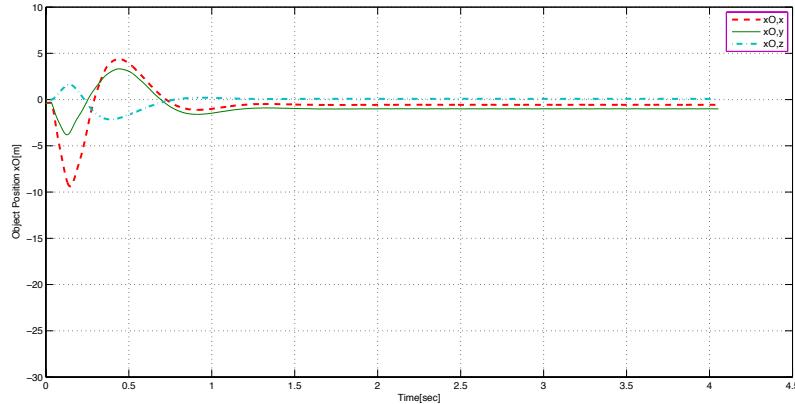


Figure 4.28: Object impedance torque  $m_O$  along the  $x_o$  axis (translation 40 cm), with the IPC–IF controller.

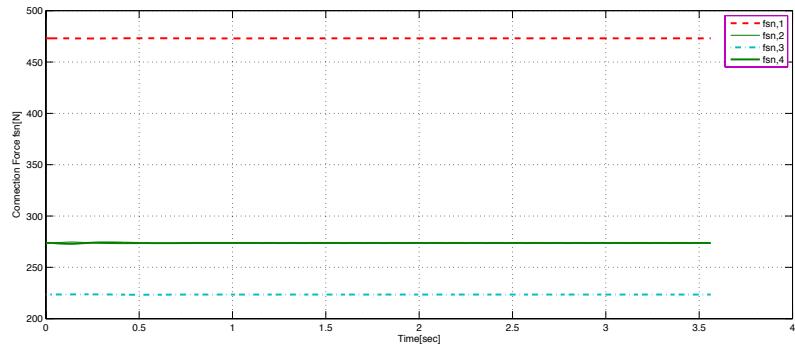


Figure 4.29: Magnitude of internal forces  $f_{sn}$  during displacement along the  $x_o$  axis (translation 40 cm), with the IPC–IF controller.

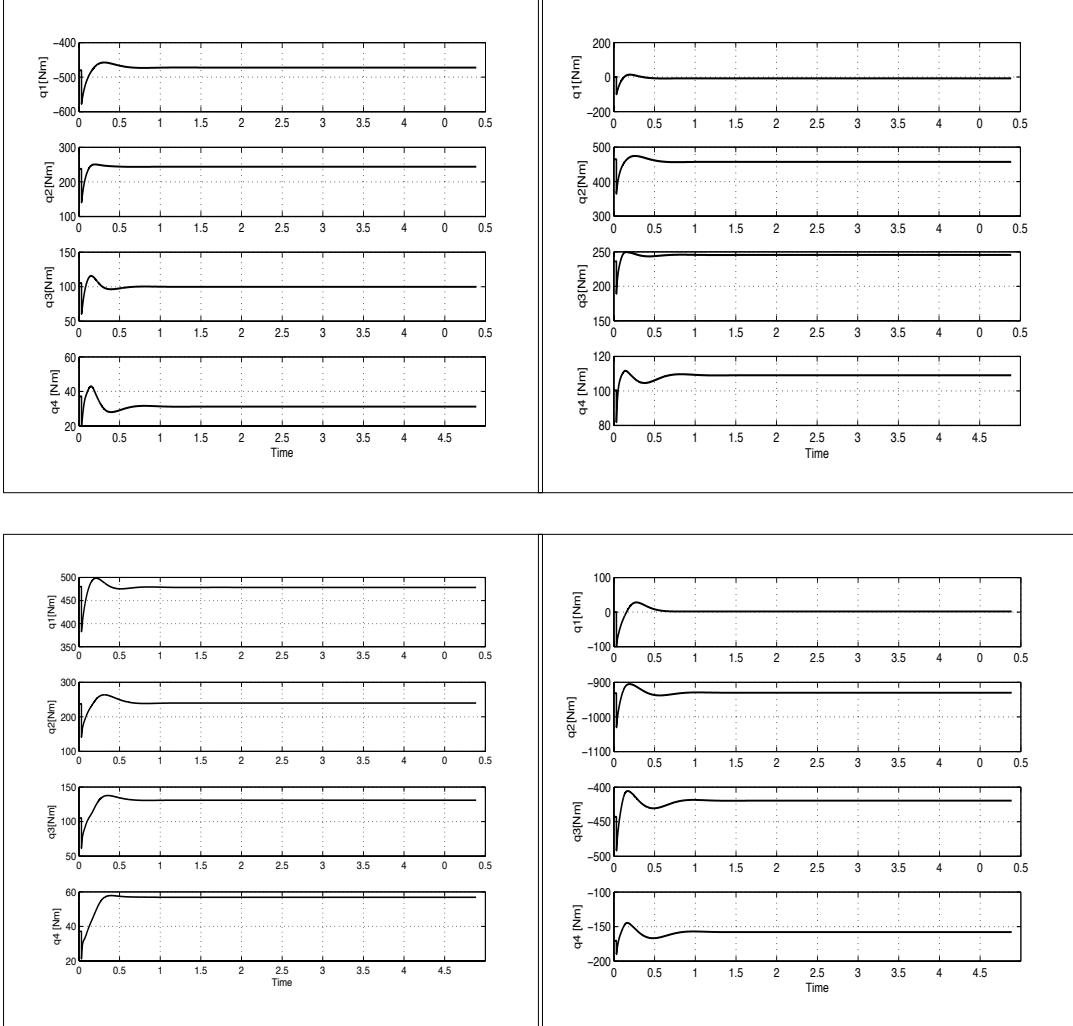


Figure 4.30: Applied joint torques of fingers 2,3,4,1–thumb, during displacement along the  $x_o$  axis (translation 40 cm), with the IPC–IF controller.

### 4.3.2 Stiffness measurement

In this measurement scenario, in the same way as in the previous section, we sequentially apply increasing forces/torques at frame  $H_o$  on the object in order to observe the resulting object stiffness  $K_{\bar{x}} = \text{blockdiag}\{K_{\text{translational}}, K_{\text{rotational}}\}$ .

We begin by applying an increasing force sequentially along the  $x$ ,  $y$ , and  $z$  axes and observing the resulting translations–rotations. We increase the force by  $20N$  every  $0.7sec$ . Then, we apply a torque that increases sequentially by  $3Nm$  every  $0.7sec$  along all three axes.

The measured stiffness values (translational–rotational) in this case are:

$$K_{tr,\text{measured}} = \begin{bmatrix} 699 & 0 & 0 \\ 0 & 704 & 0 \\ 0 & 0 & 698 \end{bmatrix}, \quad K_{rot,\text{measured}} = \begin{bmatrix} 191 & 0 & 0 \\ 0 & 196 & 0 \\ 0 & 0 & 192 \end{bmatrix}$$

Indeed, as expected, these values are significantly closer to the desired ones ( $k_{\text{translational}} = 700$ ,  $k_{\text{rotational}} = 200$ ) and, at the same time, coupling between different directions is significantly smaller, as can be seen in Figures 4.31, 4.32, 4.34, 4.35.

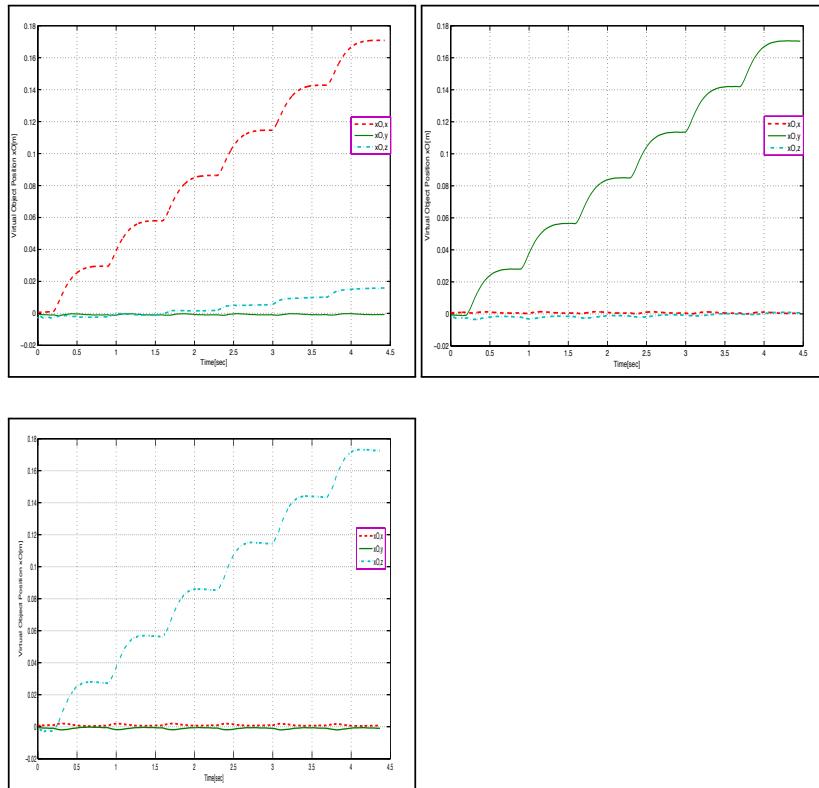


Figure 4.31: Virtual frame position  $x_o$  under applied forces along the  $x$ ,  $y$ ,  $z$  axes, respectively, with the IPC–IF controller.

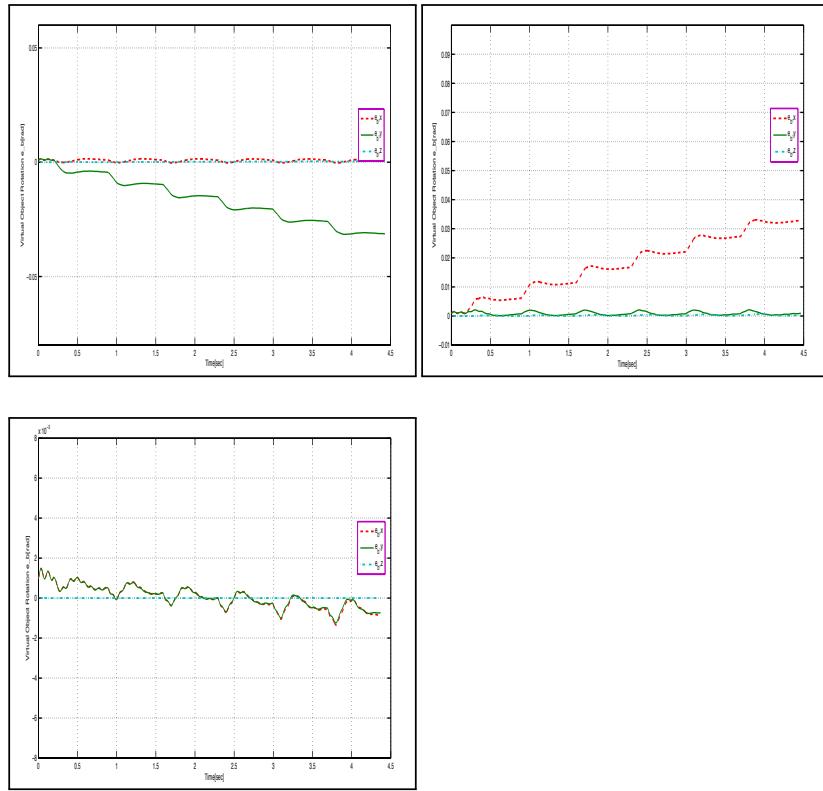


Figure 4.32: Virtual frame quaternion vector  $e_b$  under applied forces along the  $x$ ,  $y$ ,  $z$  axes, respectively, with the IPC–IF controller.

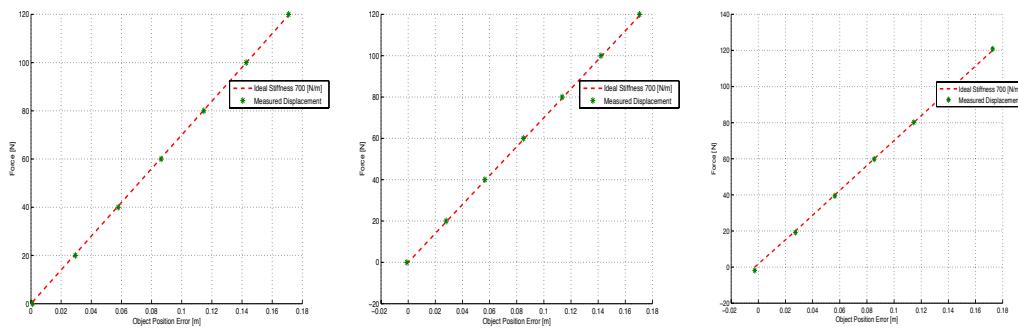


Figure 4.33: Object position  $x_o$  after applying forces along the  $x$ ,  $y$ ,  $z$  axes of frame  $H_o$ , respectively, with the IPC–IF controller.

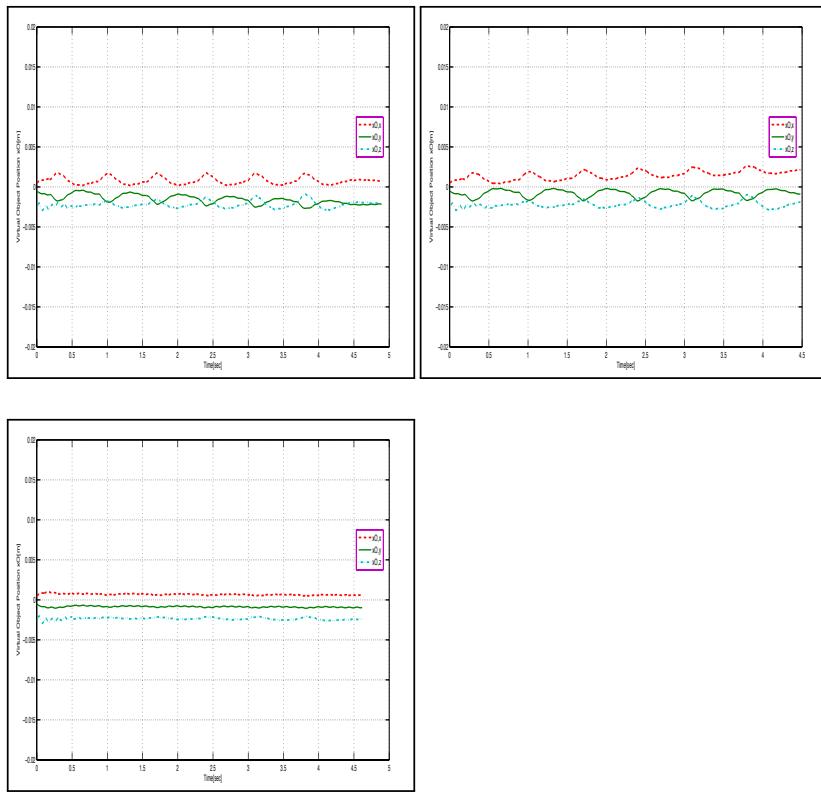


Figure 4.34: Virtual frame position  $x_o$  under applied torques along the  $x$ ,  $y$ ,  $z$  axes, respectively, with the IPC–IF controller.

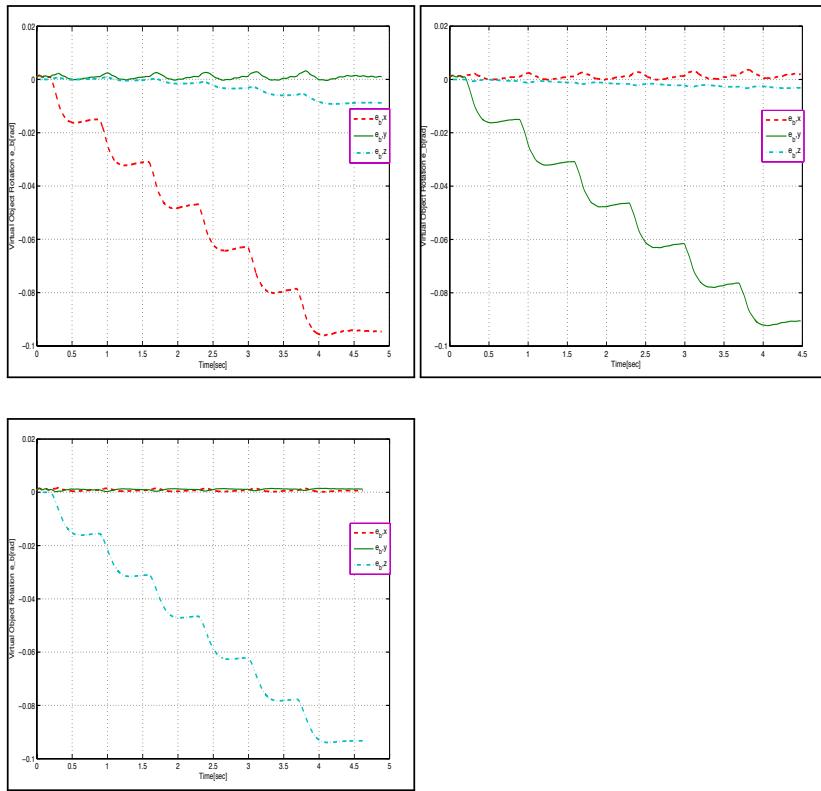


Figure 4.35: Virtual frame quaternion vector  $e_b$  under applied torques along the  $x$ ,  $y$ ,  $z$  axes, respectively, with the IPC–IF controller.

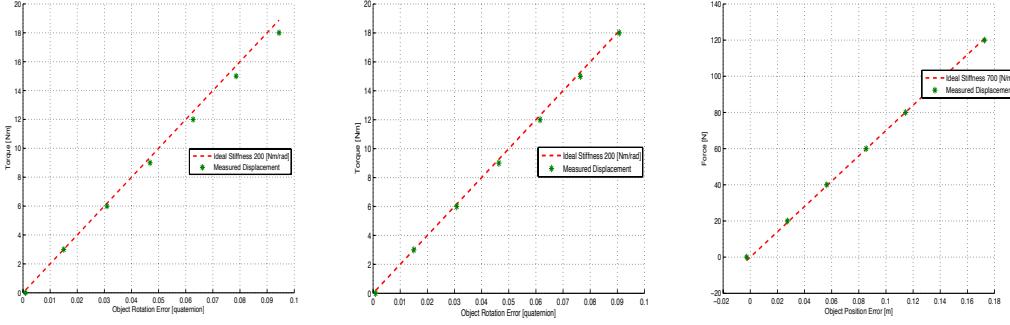


Figure 4.36: Object quaternion after applying torques along the  $x$ ,  $y$ ,  $z$  axes of frame  $H_o$ , respectively, with the IPC–IF controller.

## 4.4 Comparison of controllers IPC and IPC–IF

In order to perform a final comparison between the two implemented techniques, we compute the mean square errors with respect to the apparent translational and rotational stiffness, which are presented in Tables 4.4 and 4.5.

We observe in each case that the IPC–IF controller has significantly lower error, which in practice translates to a more consistent stiffness matrix, as we have already computed.

	Translational Stiffness $x$	Translational Stiffness $y$	Translational Stiffness $z$
IPC	1.29	1.36	3.58
IPC – IF	0.62	1.23	0.68

Table 4.4: Mean square errors ( $\sum_{i=1}^N e^2/N$ ) with respect to the apparent translational stiffness of the robotic grasp.

	Rotational Stiffness $x$	Rotational Stiffness $y$	Rotational Stiffness $z$
IPC	1.67	0.49	0.87
IPC – IF	1.19	0.26	0.73

Table 4.5: Mean square errors ( $\sum_{i=1}^N e^2/N$ ) with respect to the apparent rotational stiffness of the robotic grasp.

Regarding final position errors, we should note that performance does not differ between the two controllers to the extent initially expected. As already noted, we observe a significant reduction in the steady-state impedance force error  $f_o$  along the  $z$  axis, from 3.8 N (with the basic IPC controller) to 2 N with the proposed IPC–IF controller. Since in all other cases the steady-state position error remains practically the same, we can state that in this case as well, the IPC–IF controller is superior.

## 4.5 Gravitational compensation

At this point we introduce gravitational effects for the object in our model. We obviously expect steady-state position–orientation errors in the final state, as observed by the tilt of the cylindrical object in Figure 4.37a.

To compensate for these, we apply the scheme proposed in Subsection 3.4.2, which essentially introduces additional stiffness spring terms permanently aligned with the gravity vector, aiming to cancel it in the object frame.

In the experimental scenario, initially there is no effect from gravity until time  $0.3sec$ , when the gravitational effect on the object is activated. We observe the deviations and errors that arise until the activation of the gravitational compensation term, which occurs at time  $1.3sec$ .

(a) Tilt – steady-state position error      (b) Correction – recovery of orientation

Figure 4.37: Tilt – steady-state position error in the presence of gravity. Introduction of the gravitational compensation term with recovery of position–orientation.

Figure 4.38: Object position  $xO$  during application of the gravity term at  $0.3sec$  and the compensating term at  $1.3sec$ .

Figure 4.39: Translational impedance force  $f_O$  during application of the gravity term at  $0.3sec$  and the compensating term at  $1.3sec$ .

Figure 4.40: Rotational impedance  $m_O$  during application of the gravity term at  $0.3sec$  and the compensating term at  $1.3sec$ .

Figure 4.41: Quaternion vector  $e_b$  during application of the gravity term at  $0.3sec$  and the compensating term at  $1.3sec$ .

# Chapter 5

## Conclusions & Future Work

In this final chapter, we summarize the main points of this diploma thesis, the conclusions that arise, as well as possible future directions and extensions.

### 5.1 Summary

This diploma thesis addresses a particularly demanding topic in the field of robotics, namely dexterous robotic manipulation using anthropomorphic robotic-hand configurations (with parallel cooperating kinematic chains).

First, an extensive literature review is presented on fundamental topics related to dexterous robotic manipulation, with an emphasis on the employed control techniques.

For the study and development of solutions at the control level of robotic grasping, we developed a custom simulation environment within Simulink, using the open-source Open Dynamics Engine libraries compiled into a MEX C++ S-Function block, with the goal of building a multibody-dynamics simulation platform with contact-collision handling.

On this platform, we initially apply an object-level dexterous-grasping control technique based on passivity, in combination with dynamic damping control at the object level. The results of these simulations demonstrate reliable execution of dynamic simulations based on the ODE API, always within the strict assumptions set from the outset, primarily regarding the physics of contacts between bodies. Based on these results, the particular algorithm is also evaluated, and although it exhibits overall satisfactory behavior, our critique focuses mainly on the following points,

- Large tangential pressure on cylindrical or cubic objects at the fingertips, resulting in possible loss of support due to motion outside the friction constraints of the contact model.
- Presence of steady-state final-position errors.
- Presence of parasitic coupling between motions in different directions.
- Error in the measured stiffness compared to the stiffness defined at the controller level.

We attribute these effects to the internal-force control technique through defining stiffness coefficients between the fingertips and a virtual object frame. The forces generated by these connections are not mapped into the null space of the grasp matrix, resulting in a resultant external force on the manipulated object, which is also a cause of the aforementioned problems.

On this issue, we propose a solution based on the geometry of the object surface, and specifically the projection (into the internal subspace of the grasp matrix) of forces normal to the contact surface. We consider this to be a realistic approach, since acquiring the surface-normal direction is possible through vision–touch sensor suites which, in most cases, are present in robotic applications targeting dexterous manipulation. In this way, we keep fingertip forces within the friction constraints of the contact model, avoiding possible slipping, while simultaneously achieving explicit control of the internal forces. The performed simulations verify exactly these hypotheses, yielding smaller steady-state position errors, smaller couplings between motions in different directions, as well as a consistent stiffness matrix compared to the one defined.

Finally, we introduce the gravitational effect into the system, which, as an external force, creates steady-state position errors. For the fingers we have already introduced a dynamic gravity term, but not for the object. By extending the stiffnesses with respect to translation–rotation, we introduce an object-level gravity-compensation term, geometrically consistent, with successful results.

## 5.2 Extensions – Future Directions

### Simulation

The simulation of dynamic systems constitutes an extremely large and critical part of systems engineering for building complete, successful applications. As already mentioned, Open Dynamics Engine is a particularly popular solution for simulating robotic systems, mainly due to its robustness, stability, flexibility, and speed. Nevertheless, it exhibits significant limitations and simplifications, while at the same time it is restricted to simulations of perfectly rigid bodies. In contrast, a significant part of robotics research is oriented toward soft–flexible robotic components (Soft Robotics). Although for such cases one can seek customized commercial solutions in industry, the problem of realistic multibody dynamic simulation with simultaneous collision handling remains open.

Therefore, today one could develop a new open-source software package for dynamic simulation, tailored to the current needs of the robotics community. It is interesting that, although there have been numerous efforts to create robotic development platforms (Webots, V-REP, ROS, etc.), they are all based on simplified physics engines oriented primarily toward entertainment and graphics applications, with the main ones being ODE, Bullet, and PhysX, where the objective is fast and plausible response rather than dynamic simulation with absolute accuracy of results.

### Control

Regarding control, one can first observe that in our system we assume a perfect joint-torque controller, where our command signal translates perfectly into the torque ultimately applied at the joints. This naturally removes a significant portion of the simulation’s reliability. Therefore, an obvious extension would be the creation of a low-level torque-control subsystem that includes the dynamics of the actuation system.

Regarding the overall dynamic system of the robotic-grasp model, as already mentioned, in our analysis we made certain simplifying assumptions to facilitate the determination of the dynamic damping term. This results in a non-negligible inconsistency in the character of the transient characteristics for different configurations of the system. It would be useful to investigate, in this direction, a more complete control scheme that would also take into account the Coriolis terms (products of angular velocities) that we omitted in our analysis.

Subsequently, what becomes evident is that in our algorithm we assume the inertial characteristics of the manipulated object are known, which undermines the effort to build robotic systems capable of automated action in unstructured environments. It is possible to introduce an adaptive-control scheme that can tune the parameters of the system dynamically for different object inertial characteristics.

Although the system satisfies the characterization “dexterous”, in practice there is

significant room for improvement, considering that the object's orientation workspace is in fact limited. A particularly interesting prospect would therefore be an extension of the scheme to perform finger gaiting and/or suitable sliding–rolling at the fingertips, to increase dexterity, essentially introducing hybrid/discrete-event dynamics in this way.

Another line of work with substantial interest would be testing these algorithms on systems with different contact models that also include second-order effects.

## **Real System**

Theoretical analysis and simulation are inseparable and necessary parts of any effort to develop automatic-control systems. However, all of this has limited meaning if there is no application in the real world and under real conditions.

# Bibliography

- [1] O. Khatib B. Siciliano, editor. *Handbook of Robotics*. Springer, 2008.
- [2] M. Otter J. Stelter. G.Hirzinger, J. Bals. The dlr-kuka success story robotics research improves industrial robots. *IEEE Robotics & Automation Magazine*, 12(3):16–23, Sept. 2005 2005.
- [3] <http://www.businessinsider.com/the-drones-are-coming-2012-12>.
- [4] Tsuneo Yoshikawa. Multifingered robot hands: Control for grasping and manipulation. *Annual Reviews in Control*, 34(2):199 – 208, 2010.
- [5] <http://www.svrobo.org/>.
- [6] <http://spectrum.ieee.org/automaton/robotics/industrial-robots/next-big-thing-in-silicon-valley-robotics>.
- [7] <http://faculty.washington.edu/chudler/facts.html>.
- [8] A. Bicchi. Hands for dexterous manipulation and robust grasping:a difficult road toward simplicity. *IEEE Trans Rob Autom*, , 16(6):652–662, December 2000.
- [9] [http://www.ieee.ca/millennium/canadarm/canadarm\\_technical.html](http://www.ieee.ca/millennium/canadarm/canadarm_technical.html).
- [10] Eric Brown, Nicholas Rodenberg, John Amend, Annan Mozeika, Erik Steltz, Mitchell R. Zakin, Hod Lipson, and Heinrich M. Jaeger. Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences*, 107(44):18809–18814, 2010.
- [11] F. Lotti, P. Tiezzi, G. Vassura, L. Biagiotti, and C. Melchiorri. Ubh 3: an anthropomorphic hand with simplified endo-skeletal structure and soft continuous fingerpads. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 5, pages 4736 – 4741 Vol.5, april-1 may 2004.
- [12] Pham Thuc Anh Nguyen and Suguru Arimoto. Learning motion of dexterous manipulation for a pair of multi-dof fingers with soft-tips. *Asian Journal of Control*, 4(1):11–20, March 2002.

- [13] P. Payeur F. F. Khalil. Dexterous robotic manipulation of deformable objects with multi-sensory feedback - a review. *Robot Manipulators, Trends and Development, In-Teh(eds)*, Chap. 28:33 pages, March 2010.
- [14] V. Kumar A. Bicchi. Robotic grasping and contact: A review. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 1, pages 348–353, August 2000.
- [15] Andrey V.Savkin. *Hybrid Dynamical Systems*. Birkhäuser, 2002.
- [16] Thomas Schlegl, Martin Buss, and Günther Schmidt. Hybrid control of multi-fingered dexterous robotic hands. In Sebastian Engell, Goran Frehse, and Eckehard Schnieder, editors, *Modelling, Analysis, and Design of Hybrid Systems*, volume 279, pages 437–465. Springer Berlin Heidelberg, 2002.
- [17] B. Wang, L. Jiang, J.W. Li, H.G. Cai, and H. Liu. Grasping unknown objects based on 3d model reconstruction. In *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*, pages 461 – 466, july 2005.
- [18] Byoung-Ho Kim, Byung-Ju Yi, Sang-Rok Oh, and Il Hong Suh. Independent finger and independent joint-based compliance control of multifingered robot hands. *Robotics and Automation, IEEE Transactions on*, 19(2):185 – 199, April 2003.
- [19] Hanafiah Yussof and Masahiro Ohka. Application of stiffness control algorithm for dexterous robot grasping using optical three-axis tactile sensor system. In *Micro-NanoMechatronics and Human Science, 2009. MHS 2009. International Symposium on*, pages 472 – 476, nov 2009.
- [20] Jeremy A. Fishel and Gerald E. Loeb. Bayesian exploration for intelligent identification of textures. *Frontiers in Neurorobotics*, 6(4), 2012.
- [21] Gregory. P. Starr. An experimental investigation of object stiffness control using a multifingered hand. *Robotics and Autonomous Systems, Elsevier*, 10(1):33 –42, October 1992.
- [22] Y. Youm H.R. Choi, W.K. Chung. Control of grasp stiffness using a multifingered robot hand with redundant joints. *Robotica (1995), Cambridge University Press*, 13:351–362, 1994.
- [23] K. Kumazaki, M. Svinin, Z. Luo, T. Odashima, and S. Hosoe. A study on the stable grasping by a redundant multi-fingered robot hand. In *SICE 2002. Proceedings of the 41st SICE Annual Conference*, volume 1, pages 230 –231, aug 2002.
- [24] T. Schleg M. Buss. Multi-fingered regrasping using on-line grasping force optimization. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 2, pages 998 – 1003, April 1997.

- [25] Zhiwei Luo, T. Ito, N. Sugimoto, T. Odashima, and S. Hosoe. Virtual impedance control for preshaping of a robot hand. In *SICE 2002. Proceedings of the 41st SICE Annual Conference*, volume 3, pages 2058 – 2059, August 2002.
- [26] Akira Nakashima, Yuta Yoshimatsu, and Yoshikazu Hayakawa. Analysis and synthesis of stable grasp by multi-fingered robot hand with compliance control. In *Control Applications (CCA), 2010 IEEE International Conference on*, pages 1582 – 1589, sept. 2010.
- [27] G. Hirzinger T. Wimboeck, C. Ott. Passivity-based object-level impedance control for a multifingered hand. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 4621 – 4627, oct 2006.
- [28] G. Hirzinger T. Wimboeck, C. Ott. Impedance behaviors for two-handed manipulation: Design and experiments. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4182 – 4189, april 2007.
- [29] A. Albu-Schäffer G. Hirzinger T. Wimboeck, C. Ott. Comparison of object-level grasp controllers for dynamic dexterous manipulation. *The International Journal of Robotics Research*, 31(1):3 –23, September 2011.
- [30] Σπύρος Γ. Τζαφέστας. *Ρομποτική, Ανάλυση – Έλεγχος – Σχεδιασμός – Προγραμματισμός – Αίσθηση*. 2003.
- [31] <http://www.texample.net/tikz/examples/>.
- [32] Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, 3rd edition, 2002.
- [33] James Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. Stanford University, October 2006.
- [34] Udo Frese Gerd Hirzinger Alin Albu-Schäffer, Christian Ott. Cartesian impedance control of redundant robots: Recent results with the dlr-light-weight-arms. In *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA '03.*, volume 3, pages 3704 – 3709, 2003.
- [35] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *Robotics and Automation, IEEE Journal of*, 3(1):43 – 53, February 1987.
- [36] Gerd Hirzinger Alin Albu-Schäffer, Christian Ott. A passivity based cartesian impedance controller for flexible joint robots - part ii: Full state feedback, impedance design and experiments. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 3, pages 2666 – 2672 Vol.3, april-1 may 2004.
- [37] <http://www.ode.org/ode-latest-userguide.html>.

- [38] Fanny Ficuciello. *Modelling and Control for Soft Finger Manipulation and Human-Robot Interaction*. PhD thesis, Faculty of Engineering, University of Naples Federico II, 2010.
- [39] <http://code.google.com/p/ode-frank/>.