

Giuseppe Carbone *Editor*

# Grasping in Robotics

# Mechanisms and Machine Science

Volume 10

*Series Editor*

Marco Ceccarelli

For further volumes:  
<http://www.springer.com/series/8779>

Giuseppe Carbone  
Editor

# Grasping in Robotics

*Editor*

Giuseppe Carbone  
University of Cassino  
Cassino  
Italy

ISSN 2211-0984

ISBN 978-1-4471-4663-6

DOI 10.1007/978-1-4471-4664-3

Springer London Heidelberg New York Dordrecht

ISSN 2211-0992 (electronic)

ISBN 978-1-4471-4664-3 (eBook)

Library of Congress Control Number: 2012952025

© Springer-Verlag London 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media ([www.springer.com](http://www.springer.com))

# Preface

This book is an attempt to address the wide topic of grasping in robotics with a multi-disciplinary approach. Each chapter has been authored by an expert or a team of experts in a specific area spanning from the mechanics of machinery to control theory, informatics, and mechatronics. Chapters have been divided into four sections with the aim to give a theoretical and historical background, to discuss main concepts for mechanical design, to illustrate main issues on control and motion planning for both industrial and non-conventional applications. Applications have been detailed by referring to industrial gripping solutions as proposed with experiences at the company Schunk GmbH, for grasping solutions in agriculture, micro-gripping solutions in precision assembly processes, and innovative design solutions such as in fusing grasping features into robot design.

This book project can be foreseen as a reference for young professionals/researchers to overview the most significant aspects in the field of grasping in robotics. Given the wideness of the topic, this book can be considered as a first edition and, as Editor, I shall be pleased to consider additional contents/suggestions for a future edition.

I wish to acknowledge all the authors for their significant contributions to this project. Particularly, I wish to thank Prof. Marco Ceccarelli, who has been also the mentor of my academic career. He has strongly stimulated and supported my research activities and this book project, since its early beginning. Also acknowledged is the professional assistance by the staff of Springer Science + Business Media that have supported this project with their help and advice in the preparation of the book.

I am grateful to my wife Annalisa, my daughters Mariagrazia, Manuela, and the just born Camilla. Without their patience and understanding it would not have been possible for me to work on this book.

Cassino, July 2012

Giuseppe Carbone

# Contents

## Part I Background

<b>1 Notes for a History of Grasping Devices .....</b>	<b>3</b>
Marco Ceccarelli	
<b>2 Stiffness Analysis for Grasping Tasks .....</b>	<b>17</b>
Giuseppe Carbone	
<b>3 Multibody Dynamics Formulation.....</b>	<b>57</b>
Parviz E. Nikravesh	
<b>4 Sensors and Methods for the Evaluation of Grasping.....</b>	<b>77</b>
Antonio Morales, Mario Prats and Javier Felip	

## Part II Mechanism Design

<b>5 Industrial Grippers: State-of-the-Art and Main Design Characteristics .....</b>	<b>107</b>
Penisi Osvaldo Hugo	
<b>6 Robotic Hands and Underactuated Finger Mechanisms .....</b>	<b>133</b>
Licheng Wu	
<b>7 Finger Orientation for Robotic Hands.....</b>	<b>159</b>
Minzhou Luo	
<b>8 Parallel Wrists for Enhancing Grasping Performance .....</b>	<b>189</b>
Massimo Callegari, Luca Carbonari, Giacomo Palmieri and Matteo-Claudio Palpacelli	

**Part III Control and Motion Planning**

<b>9 A Survey on Different Control Techniques for Grasping . . . . .</b>	223
Balaguer Carlos, Gimenez Antonio, Jardón Alberto, Cabas Ramiro and Martínez de la Casa Santiago	
<b>10 Hardware Control System of Robotic Hands . . . . .</b>	247
Zhan Qiang	
<b>11 Visual Servo Control of Robot Grasping . . . . .</b>	269
Theodor Borangiu	
<b>12 Path Planning for Grasping Tasks . . . . .</b>	305
João Carlos Mendes Carvalho and Sezimária F. Pereira Saramago	
<b>13 Grasp and Motion Planning for Humanoid Robots . . . . .</b>	329
Markus Przybylski, Nikolaus Vahrenkamp, Tamim Asfour and Rüdiger Dillmann	

**Part IV Applications**

<b>14 Hardware for Industrial Gripping at SCHUNK GmbH &amp; Co. KG . . . . .</b>	363
Christopher Parlitz	
<b>15 Grasping in Agriculture: State-of-the-Art and Main Characteristics . . . . .</b>	385
F. Rodríguez, J. C. Moreno, J. A. Sánchez and M. Berenguel	
<b>16 Microgrippers and their Influence on High Precision Assembly Processes . . . . .</b>	411
A. Raatz and R. J. Ellwood	
<b>17 A New Way of Grasping: PARAGRIP—The Fusion of Gripper and Robot . . . . .</b>	433
Tom Mannheim, Martin Riedel, Mathias Hüsing and Burkhard Corves	
<b>Index . . . . .</b>	465

# **Part I**

## **Background**

# Chapter 1

## Notes for a History of Grasping Devices

Marco Ceccarelli

**Abstract** In this chapter an evolution of grasping devices is briefly presented with notes by looking at design solutions. A history of grasping devices and their evolutions is outlined by using illustrative examples of both mechanical design and invented solutions.

### 1.1 Introduction

Grasping devices have evolved to help humans in manipulation for handling objects of different sizes, materials, and conditions. Grasping has been considered as an essential part of manipulation and only recently a specific attention has been addressed to grasping devices as independent mechanical/mechatronic designs with theory, practice, and application.

Even in the history of mechanical systems no particular attention has been devoted to grasping devices as independent systems. This is also due to the fact that grasping devices have been conceived, built, and used, as they still are, for helping humans in manipulation with two categories, namely tools and prostheses.

Tools are those systems that enable humans to enlarge grasping capability or to achieve difficult or dangerous grasps. Prostheses are systems that are developed to restore grasping capability in human arms when hands are lost.

Grasping devices are tools that are useful to perform the loading of objects in systems for manipulation, handling, or transportation. They are fundamentals

---

M. Ceccarelli (✉)

LARM: Laboratory of Robotics and Mechatronics, University of Cassino and South Latium,  
Via G. Di Biasio, 43 03043 Cassino, FR, Italy  
e-mail: ceccarelli@unicas.it

components in achieving successfully these tasks. Since antiquity grasping devices have been conceived and used to help or substitute human extremities in achieving or enhancing grasping tasks. Grasping devices can be considered mechanical interfaces, when they are not even the goal systems yet, that perform mechanical actions on objects to be grasped. A variety of grasping devices have been developed during machine evolutions and they are today available with mechatronic solutions that are based on actions and motions of elements that are in contact with the objects to be grasped.

From historical viewpoints, grasping devices were developed as tools since the beginning of technical skills in human society. When such a skill was so successful early prostheses were also conceived even in antiquity.

Although with scarce specific attention, even in the past, interest has been addressed to the study of evolution of grasping systems, but as part of manipulation systems. In fact, in historical surveys on developments of machines, in general grasping devices are not specifically considered but they are still included as part of machines as for example in [2, 9, 18, 22, 35].

Some specific attention is addressed recently mainly within the history of robots, as indicated for example in [8]. Even technical surveys are in general focused on topics developments but with a modern recent background, like for example in [4, 31, 12]. Even historical tracks of recent systems are usually outlined within technical problems more than a well-focused historical study, like for example in [11].

Nevertheless, grasping mechanisms have attracted specific technical interest in modern times in order to outline the specific mechanisms and to emphasize on specific design algorithms. In those mechanism atlas or mechanism surveys, like in [3, 13, 14, 17, 24, 25, 32] a historical consideration can be appreciated as referring to mechanisms from the past. Even commercial catalogs of grippers can appreciate somehow historical insights for recent development, like for example in [40, 41]. Catalogs of mechanisms for machines can also include mechanisms for grasping devices with specific mention to grasping purposes in the past and in modern time, like for example in [1, 5, 15] just to cite some.

Specific design algorithms very seldom are outlined with a historical background but sometimes they can also track the evolution of the grasping devices in terms of the rationale on which they have been designed, like for example in [6, 13, 14]. Recently, specific attention has been addressed to the specific field of robot hands even by using the above-mentioned approaches of mechanism survey and problem/algorithm outline, like for example in [20, 38].

Even books have been recently published as dealing specifically on grasping devices with specific short accounts on their history, like for example in [29, 37, 42].

In this chapter, the historical evolution of grasping devices as tools and prostheses is briefly presented with few significant examples to point out main facts and achievements that have permitted the modern technology of grasping devices. This presentation is based on illustrations that are representative of the discussed aspects for outlining a specific history of grasping devices as preliminary notes that can be used for future well-focused study of the history of grippers.

## 1.2 Grasping in Antiquity

Grasping tools were conceived and developed since Prehistory Times with the aim to help humans to grasp objects that were difficult for size, shape, materials, and conditions. Early grasping tools were developed just with two rigid links that act as fingers in independent operation or in fairly simple link design. Documents in archeology are very rare and the only remains and illustrations indicate a well-established technology for grippers and even very primitive artificial hands. In general, those first grasping devices were two-finger grippers with a very essential design both in size and shape to grasp a certain limited variety of objects. Nevertheless, the practical efficiency of those devices motivated further developments up to early prostheses. An iron hand was discovered as attached to an arm of an Egyptian mummy, and other prostheses were found as parts of human extremities. Most of those early prostheses were very likely with cosmetic functions more than really active artificial hands. Nevertheless, a first well-documented case of a functional artificial hand is reported by Plinius the Older in [33] as regarding a pretorian Roman official Marcus Sergius Silus, who had a replaced hand after its loss during the second Punic War (218–200 BC).

In Antiquity, grasping devices were developed with specific functions not only made of perishable materials. There were grasping tools for construction, metal working, theater plays, and in general for production and manufacturing of handicraft products.

In Fig. 1.1 a fairly simple gripper is indicated among tools of a blacksmith. The fingers are properly curved to facilitate the grasp by extremity points and to handle large object. The operation is also indicated in a moment of blacksmith action. Since the scene is related to metallurgic manufacturing, it is very likely that those kinds of grippers were made of iron. In addition, the fact that those tools were illustrated in an artistic representation, indicates that grippers of that kind were very likely common in a blacksmith shop in diary work.

Figure 1.2 gives examples of the grasping in cranes with alternative solutions. Namely, in Fig. 1.2a a gripper mechanism is attached at the end of a cable but with no clear indication on how the closure is ensured. This is perhaps because such a technology of closure regulation was still of value for advanced crane efficiency,

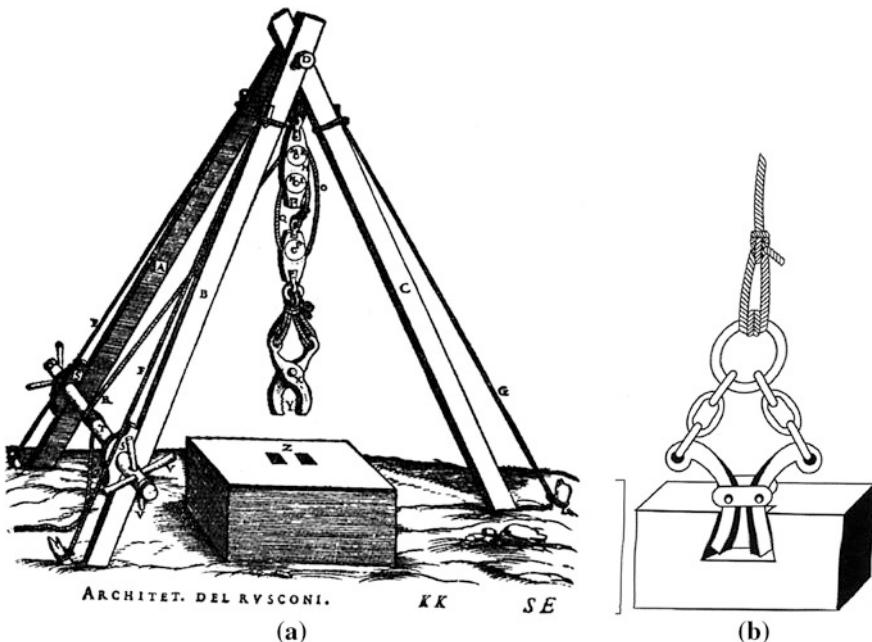


**Fig. 1.1** A gripper and its use by a blacksmith for lock constructions form a Roma basso-rilievo [21]

and therefore worth full to be kept secret to a general public or competing technicians. In Fig. 1.2b an inner grasp is obtained. This indicates that in Antiquity they were well-known of alternative solutions of grasping by using also very different mechanisms with no intuitive (natural) solutions. This may suppose theoretical studies and even creative approaches for inventions in practical applications.

Figure 1.3 shows medical grippers with fairly surprising modern-like solutions in terms of design specificity as function of the specific task. In Fig. 1.3a, a two-link gripper is shown whose operation is based on the elastic connection of the two links. In Fig. 1.3b, the gripper has two long actuating links that help to have large force at the grasp with curved fingers, whose aim is to be inserted deep in a body under treatment. The fingers show fingertips with proper rough surface that is aimed for deep grasp of textures, and therefore indicating a relevant expertise in how to use friction in grasping stability. Ergonomics of the grasp can be recognized in such a link design. Figure 1.3c shows a very early use of complex mechanisms to enhance both force efficiency and finger motion resolution. This example can indicate a knowledge on multi-link mechanisms and more likely a well aware expertise how to use their capabilities in motion and force transmission.

The above few examples show that in Antiquity there was a well-established use of grippers in several field of applications, whose quality is indicative of a well-developed technology and theoretical backgrounds, although there are no evidence of design procedures or theoretical studies for design purposes.



**Fig. 1.2** Two-finger grasp in Roman crane applications by: **a** gripper mechanism in Vitruvius's work [19]; **b** three pieces inner gripper device in a modern reconstruction [21]



**Fig. 1.3** Roman medical grippers of first century A.D. [21]: **a** with elastic actuation; **b** with curved fingers; **c** with linkage mechanism

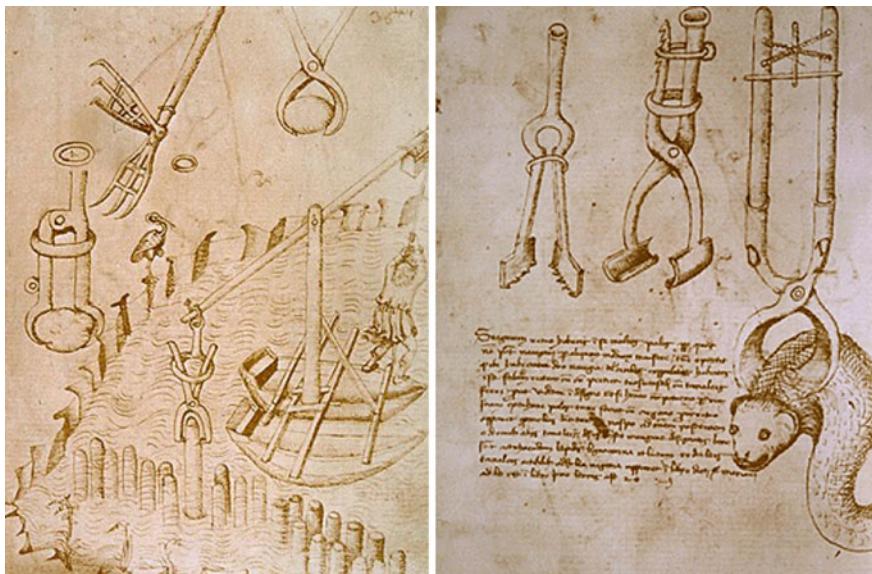
### 1.3 Evolution of Grippers

Since always grippers have been considered and used as technical systems for applications in manufacturing activities both in industrial and handcraft frames. Gripper systems for prosthesis applications have been identified as artificial hands, although at least in the very past they did not differ so much from the traditional grippers.

Grippers were developed for grasping objects in order to help the human handling. Grippers are usually referred to structures with two fingers whose extremities (not necessarily fingertips) are used to be in contact with object surface to perform the grasping action.

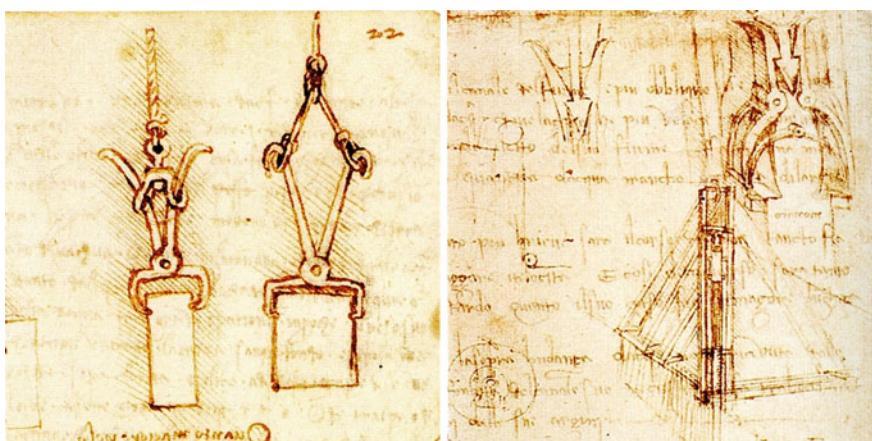
A rational design and operation of two-finger grippers can be thought as started at early Renaissance when special (new) applications required new attention to the object grasping. Thus, specific designs were developed for specific applications with details addressing care both in size and shape. Attention was even addressed to early control systems to maintain the grasp with firm configurations. All the above aspects are related to an activity for gripper design with early rational processes for design but event for a user-oriented operation.

Emblematic examples on the above approaches are reported in following figures [10]. In particular, in Fig. 1.4 interesting studies are indicated for early studies on the variety of gripper designs even for different size scales. In Fig. 1.4a, grippers are used for grasping and handling heavy loads such as pillars for building a peer. Besides different shapes for the extremities, interesting indications are drawn for solutions to maintain the grasp during the whole operation. In Fig. 1.4b, main attention is addressed to the shape and function of the extremities with a kind of fingertips, which have the goal to perform a contact with a living animal without damaging it. Different solutions are indicated, perhaps as function of different types of animals. Even the systems for regulating the grasp are indicated with solutions that give the possibilities to adjust the gripper configuration in order to regulate the opening and grasp force of the fingers, as shown in the two drawings

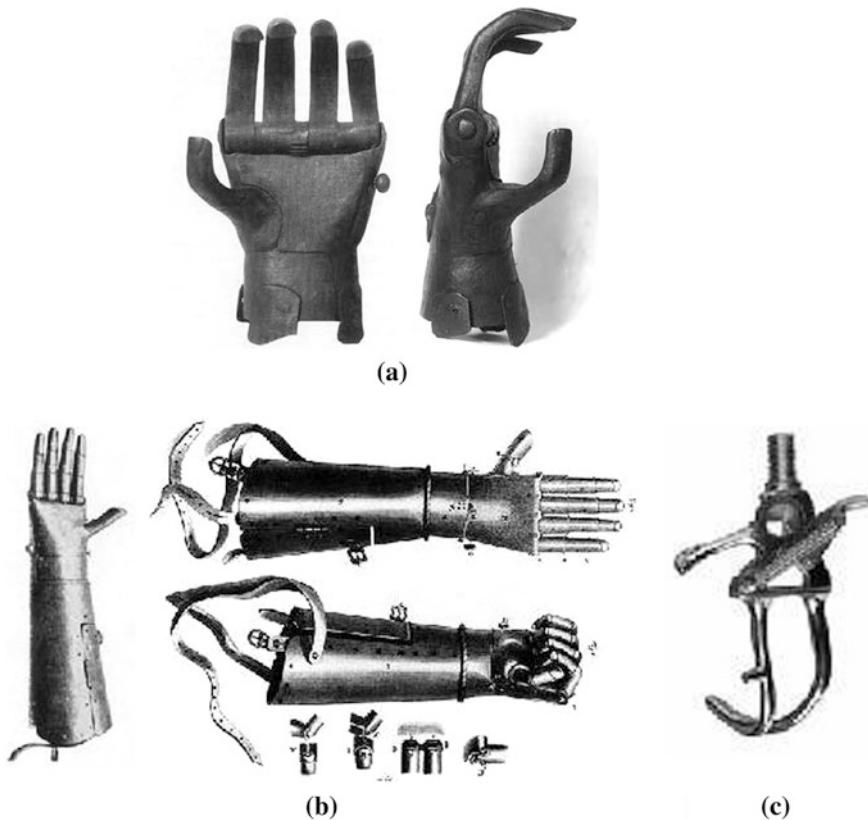


**Fig. 1.4** Mechanisms for grippers by Mariano di Jacopo (il Taccola) (1382–1458?) [10]

at the right side. Similar attention can be recognized in the drawings by Leonardo in Fig. 1.5 for grasping tools in crane applications. A certain automation of the grasping control is indicated also in the specific features of these examples, by using mechanical solutions as referred to closure devices in the actuating handles. The mechanical design of grippers was also adapted for prosthesis with two-finger functioning. Examples are illustrated in Fig. 1.6 a, b in which mechanical hands are designed with human-like shape but the operation of the fingers is obtained as a



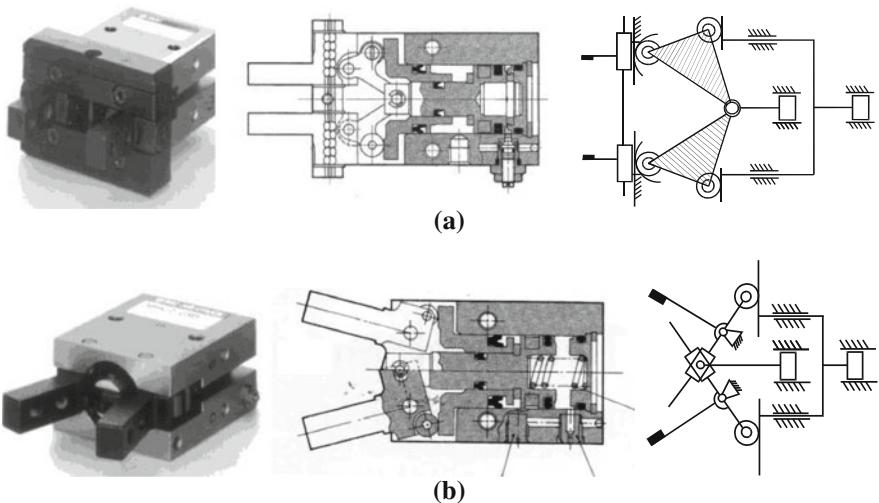
**Fig. 1.5** Mechanisms for crane grippers by Leonardo Da Vinci (1452–1519) [10]



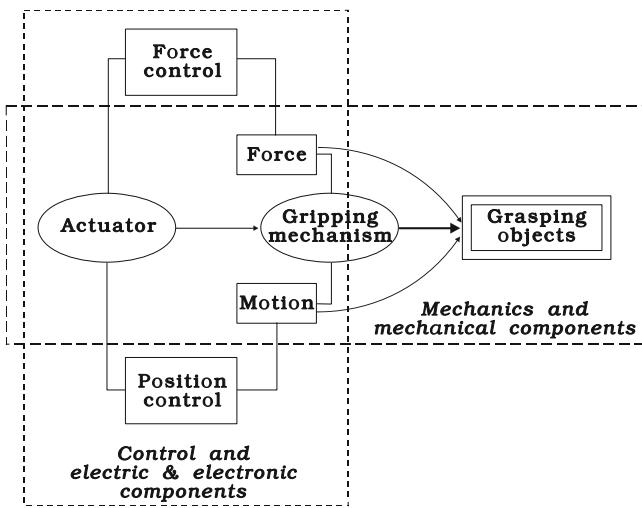
**Fig. 1.6** Mechanisms for grippers as prosthetic hands, [16]: **a** Stibbert Hand in fifteenth century; **b** in Eisern Hand in early sixteenth century; **c** a modern hook design

two-finger gripper. The two-finger functionality has been used and is still used in hook prosthesis as the modern one shown in Fig. 1.6 c, in which the fingers are properly curve shaped to widen the grasping possibilities. But the fingers act like in a mechanical gripper with just two fingers.

Mechanical grippers have evolved mainly during nineteenth century with solutions that were based on the operation of linkage mechanisms with the aim to get better efficiency in terms of actuation power. Modern industrial grippers are still based on those mechanisms, like in the examples of Fig. 1.7, [36] in which the function of a mechanism is properly combined with an actuator but within a mechatronic design according to the scheme in Fig. 1.8 [7]. Thus, the mechanical components give still the main appearance in modern industrial grippers but the overall design of a gripper and mainly its operation is more and more influenced by the actuation technology, including control systems and sensor equipment, as noted in [9]. Nevertheless, the grasping aim is still emphasized by the mechanical



**Fig. 1.7** Examples of mechanical design and mechanism chain for modern industrial grippers [36]: **a** model MHC2-10; **b** MHQG2-10



**Fig. 1.8** Current mechatronic layout for modern grippers design and operation [7]

design of finger contacts and fingertips that can be properly shaped as function of the objects or even they can be easily interchanged for such versatile purpose.

The design and operation of grippers have also required development of theoretical background to properly understand and apply the mechanics of grasp. Thus, early studies with analytical developments were carried out during nineteenth century both to study the grasp and to design mechanisms for grasping.

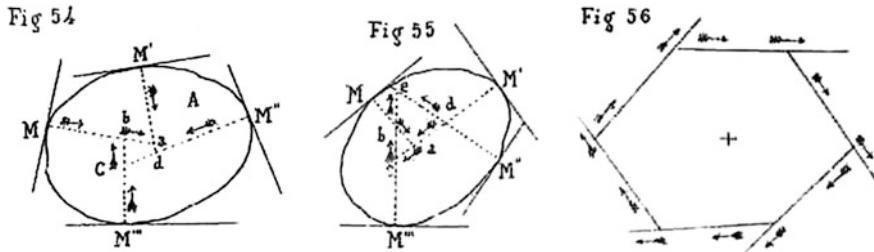


Fig. 1.9 Sketches for the analysis of planar multi-contact grasp in Masi [27]

Some specific attention was addressed by looking since then at solutions in nature, namely at clams and fingers of animals, like for example in the pioneering work [34]. An example of such an interest on the mechanics of grasp is reported in Fig. 1.9 [27], in which number of contacts and force closure form of grasping configurations are considered to outline conditions to ensure a stable static equilibrium of a grasped object among grasping fingers. Such an attention to the static equilibrium of a grasped object has been used both to enhance the finger-object interaction even with the use of new materials and fingertip shapes, and to formulate control process and hardware in order to obtain more versatile and stable grasp applications with a large variety of materials and size of objects. This is still a topic of great interest as indicated in its main aspects in [26].

In addition, the study of mechanism applications has stimulated research on suitable topology of mechanisms and catalogs of mechanisms for gripper designs. Mechanisms for gripper designs have been indicated since early handbooks of machinery, but consistent collections were outlined only since late 1970s as for example in [13, 14, 24, 25] and they are still subject of attention and research, mainly in topology search algorithms. Examples of such mechanisms catalogs are in the pioneering works of [13, 14, 25, 32]. But the topic has addressed attention also in other frames and even for teaching purposes, like in the example in Fig. 1.10 [17]. Thus, several mechanisms are attempted and indeed used as core of the grasping functioning of grippers, still nowadays even with new solutions within a mechatronic design.

From simple mechanisms in early grippers at Renaissance, the functioning of grippers has required more efficient (complex) mechanisms that have been proposed mainly for industry applications in the nineteenth century and in modern time within a mechatronic design.

210  
SISTEMATIZAREA STRUCTURALA A MECANISMELOR PLANE MONOCONTUR UTILIZABILE IN PREHENSOARE

Grafuluri lanturiilor cinematice plane monocontur, distincte din punct de vedere structural	Grafuluri variante structurale distincte ale fiecărui lant, în funcție de elementul de referință (boala)	Grafuluri variante ale structuralelor distincte ale mecanismelor plane monocontur folosite în prezență	Scheme structurale de mecanisme plane monocontur utilizabile în prezență (exemplu)
1	2	3	4
A	A I	A I 1 A I 2	1 1' 2 2' 3 3' 4 4' 1 1' 2 2' 3 3' 4 4'
B	B I	B I 1 B I 2	1 1' 2 2' 3 3' 4 4' 1 1' 2 2' 3 3' 4 4'
	B II	B II 1 B II 2 B II 3 B II 4 B II 5 B II 6 B II 7	5 5' 6 6' 7 7' 8 8' 9 9' 10 10' 11 11'

Fig. 1.10 A mechanism catalog for two-finger grippers in Dutita et al. [17]

## 1.4 Grippers Towards Artificial Hands

Experience and expertise with two-finger grippers have been transferred also in attempts for designing artificial hands either in prosthesis applications or in automation devices. This trend has been accelerated and indeed possible with successful designs in modern time because of the availability of multi-DOF mechanisms and control systems, at the most since 1940s.

In developing grippers, the reference in nature has been always the human hand [39], whose operation is characterized by a complex structure with a large variety of grasping modes. The anatomy and taxonomy of human hand have been an inspiration for gripper design and operation mainly when looking at the two-finger grasp which is the most used mode both for simplicity and efficiency, as reported for example in the reference works, Nof [30] and Mason and Salisbury [28]. But even the full structure and its operation have been of inspiration and replicated since the early developments of automata both for better understanding the human functioning and for attempting human hand construction.

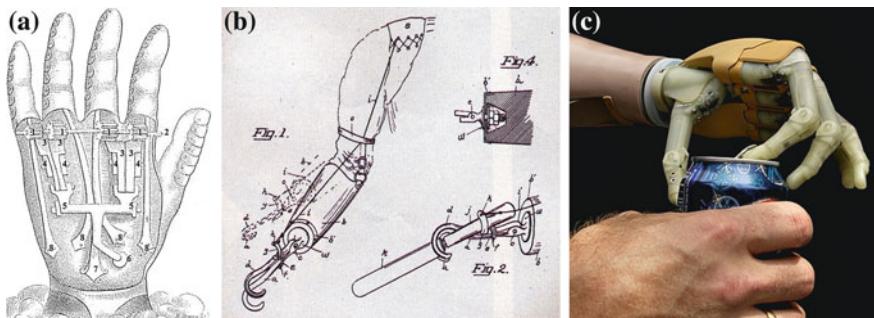
Thus, this grippers evolution can be recognized in activities for studying the biomechanics of human grasping with consequent developments of models and algorithms mainly with analysis purposes and in activities for designing and building human-like artificial hands both for prosthesis and humanoid robots or automata. The study of hand anatomy and its operation has given also the possibility to develop challenging designs for prosthesis.

Emblematic examples are shown in Fig. 1.11 [16]: from a mechanical structure as in Fig. 1.11a with a solution from sixteenth century, through attempts of arm-hand in Fig. 1.11b during nineteenth century, up to modern mechatronic solutions with high-level multi-functionality like the case I-Limb Hand example in Fig. 1.11c.

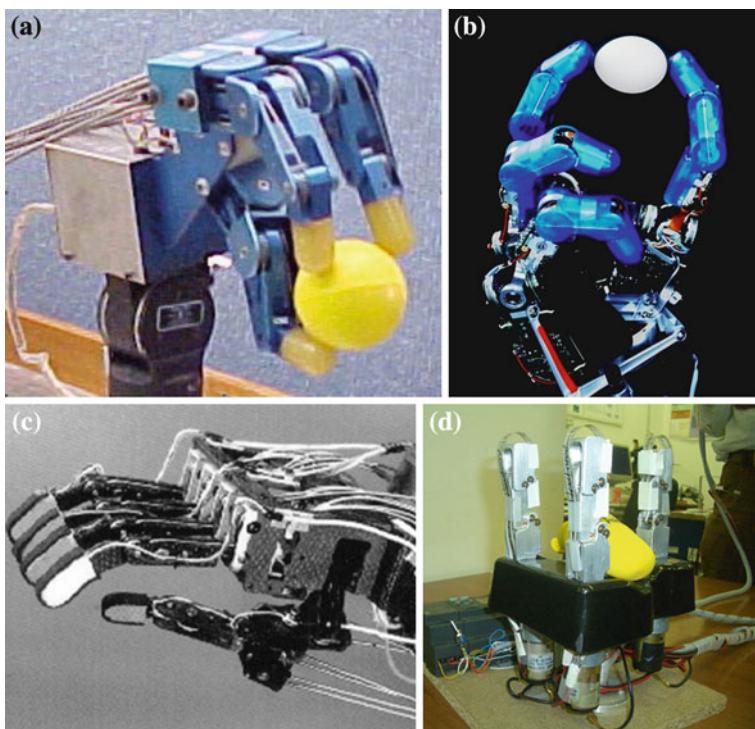
Unfortunately, prosthesis design has been often focused mainly on technical aspects for functionality in order to replace a human hand but without considering the psychology of the person and his/her capability in operating such a system, so that acceptance and real applications have been very limited. Thus, although today artificial hands in their design solutions have reached advanced mechatronic solutions with even AI features, prosthesis of human hand for practical implementation is still today with a certain primitive design and functionality, such as with hook-like solutions (see Figs. 1.6c and 1.11b, as also noted in [23]).

More successful has been the development of artificial hands with human-like appearance and functionality for implementation on robot structures and at the most in humanoid robots.

In modern time, the activity for robot artificial hands has evolved very rapidly from early designs in 1960s to very compact human-size prototypes for humanoid robots of today solutions. The gripper background is less and less evident, and today technology transfer is experienced even more in opposite directions, whereas the industrial grippers benefit of new solutions of hardware and software conceived for artificial hands of humanoid robots.



**Fig. 1.11** Examples of prosthesis designs over time [16]: **a** mechanical Lorraine hand in 1550; **b** arm-hand as developed at the end of nineteenth century; **c** a modern mechatronic prototype



**Fig. 1.12** Modern robotic hands [16]: **a** Salisbury hand built at Stanford University in USA in 1982; **b** DRL hand built at Munich University in Germany in 1997; **c** WAM-8R hand built at Waseda University Japan in 1985; **d** LARM hand built at Cassino University in Italy Cassino in 2003

Emblematic examples of artificial hands for humanoid robots are reported in Fig. 1.12, even with a kind of historical evolution.

In Fig. 1.12d the prototype of LARM hand with low-cost user-oriented operation features, [11], is shown as from direct experience of the author and his team with the aim to show that such Robotic area of advanced solutions has attracted interest and indeed motivated successful activity in Robotic labs worldwide at different levels depending of resource and applications.

Artificial hands have centered the attention in research and advanced applications of grasping, but grippers are still the most used solutions both in industrial and nonindustrial fields with features that nevertheless require further developments to fulfill conditions for an evolving/updating successful use.

## 1.5 Conclusions

In this chapter, a brief account is presented for a history of grippers as based both on practical empirical needs and experiences as well as on conceptual design, theory and algorithms, with the aim at developing and implementing practical applications of grasping devices in industrial frames and prosthesis. This short note can be considered just a first attempt to consider properly the role of gripper evolution in a more comprehensive history of grippers and artificial hands.

## References

1. Artobolevsky II (1975) Mechanisms in modern engineering, vol 5. Mir Publishers, Moscow, pp 1975–1980
2. Bautista Paz E, Ceccarelli M, Echavarri Otero J, Munoz Sanz JJ (2010) A brief illustrated history of machines and mechanisms. Science and Engineering, Book series on history of machines and machine science, vol 10. Springer, Dordrecht. doi:[10.1007/978-90-481-2512-8](https://doi.org/10.1007/978-90-481-2512-8)
3. Belfiore NP, Pennestrì E (1997) An atlas of linkage-type robotic grippers. Mech Mach Theory 32(7):811–833
4. Bicchi A (2000) Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity. IEEE Trans Robot Autom 16(6):652–662
5. Borgnis GA (1818–21) *Traité complet de mécanique appliquée aux Arts*. Bachelier, 1818–1821, Paris, 9 volumes
6. Ceccarelli M (1994) Design problems for industrial robot two-finger grippers. In: 3rd international workshop on robotics in alpe-adria region, Bled, pp 117–120
7. Ceccarelli M (1999) Grippers as mechatronic devices. Advances in multibody systems and mechatronics, Duisburg, pp 115–130
8. Ceccarelli M (2001) A historical perspective of robotics toward the future. *Fuji Int J Robotics Mechatron* 13(3):299–313
9. Ceccarelli M (2004) Fundamentals of mechanics of robotic manipulation. Kluwer/Springer, Dordrecht
10. Ceccarelli M (2008) Renaissance of machines in italy: from brunelleschi to galilei through francesco di giorgio and leonardo. *Mech Mach Theory* 43:1530–1542

11. Ceccarelli M, Carbone G (2008) Design of LARM hand: problems and solutions. *J Control Eng Appl Inform* 10(2):39–46
12. Chelpanov IB, Kolashnikov SN (1983) Problems with the mechanics of industrial robot grippers. *Mech Mach Theory* 18(4):295–299
13. Chen FY (1982) Gripping mechanisms for industrial robots. *Mech Mach Theory* 17(5):299–311
14. Chen FY (1982) Force analysis and design considerations of grippers. *The Industrial Robot*, 9(4):243–249
15. Chironis F (1961) Mechanisms, linkages, and mechanical controls. McGraw-Hill, New York
16. D'Aliesio E, Ceccarelli M (2011) A history of artificial hands. In: Proceedings of IFToMM-FelBIM international symposium on mechatronics and multibody systems MUSME 2011, Valencia, paper 37
17. Dudita F, Doaconescu D, Gogu G (1987) Curs de Mecanisme—Fascicula 4. University of Brasov, Brasov
18. Ferguson ES (1962) Kinematics of mechanisms from the Time of Watt. Contributions from the Museum of History and Technology, Washington, paper 27, pp 186–230
19. Fra' Giocondo (1511) *De Architectura*' by Marcus Polione Vitruvius (M. Vitruvius per Iocundum solito castigator factus cum figuris et tabula...) Venetii
20. Fukuda T (1993) Stae of art and future trends of R&D in robot hansa. *J Robotics Mechatron* 5(1):8–11
21. Galli M, Pisani Sartorio G (eds) (2009) *Machina—Tecnologia dell'antica Roma*. Palombi Editori, Roma (in Italian)
22. Hartenberg RS, Denavit J (1956) Men and machines: an informal history. *Machine Design*, 75–82(May 3, 1956); 101-109(June 14, 1956) ; 84-93(July12, 1956)
23. Kuniholm J (2009) Open Arms. *IEEE spectrum* 46:37–41
24. Lundstrom G (1973) Industrial robots—gripper review. *Ind Robot*, 1(2):72–82
25. Lundstrom G (1977) Industrial robot grippers. International Fluidics Service Ltd, Bedford
26. Markenscoff X, Ni L, Papadimitriou CH (1990) The geometry of grasping. *Int J Robotics Res*, 9(1):61–74
27. Masi F (1897) *Teoria dei Meccanismi*, Bologna (In Italian)
28. Mason MT, Salisbury JK (1986) Robot hands and the mechanics of manipulation. MIT Press, Cambridge
29. Monkman GJ, Hesse S, Steinmann R, Schunk H (2007) Robot grippers. Wiley-VCH, Wienheim
30. Nof SY (ed) (1985) *Handbook of industrial robotics*. Wiley, New York
31. Pertin-Troccaz J (1988) Grasping: a state of art. In: Khatib O, Craig JJ, Lozano-Perez T (eds) *The robotics review*, vol 1. The MIT Press,Cambridge, pp 71–98
32. Pham DT, Hegbotham WB (eds) (1986) Robot grippers. IFS Publications Ltd, Bedford
33. Plinius GS (1827) *Naturalis historiae liber VII, Bibliotheca classica latina sive collectio auctorum classicorum latinorum cum notis et indicibus*, Lemaire 3:121
34. Reuleaux F (1875) *Theoretische kinematic*. Braunschweig
35. Rosheim ME (1994) Robot evolution: the developments of anthropotics. Wiley, New York
36. SMC Corporation (1996) Pinze pneumatiche serie MH Cat.I 305A (In Italian)
37. Staretu I (1996) Grasping systems. Editura Lux Libris, Brasov (In Romanian)
38. Tanie K (1985) Design of robot hands. In: Nof SY (ed) *Handbook of industrial robotics* . Wiley, New York, pp 112–137 (Chapter 8)
39. Taylor CL, Schwarz MD (1955) The anatomy and mechanics of the human hand. *J Artif Limbs* 2:22–35
40. TBK (1994) Robot hand'93–94. Takano Bearing Co., Tokyo
41. TBK (1997) Robot hand'98–99. Takano Bearing Co.,Tokyo
42. Wolf A, Steinmann R, Schunk H (2005) *Grippers in motion*. Springer, Berlin

# Chapter 2

## Stiffness Analysis for Grasping Tasks

Giuseppe Carbone

**Abstract** This section addresses key aspects that are related with stiffness properties when dealing with grasping tasks. Main theoretical aspects are formulated for computing the Cartesian stiffness matrix via a proper stiffness analysis and modeling. Basic concepts are given for the comparison of stiffness performance for different robotic architectures and end-effectors by referring both to local and global properties. Cases of study are described for clarifying the effectiveness and engineering feasibility of the proposed formulation for stiffness analysis. Then, an experimental set-up and tests are proposed for the experimental validation of stiffness performance.

### 2.1 Introduction

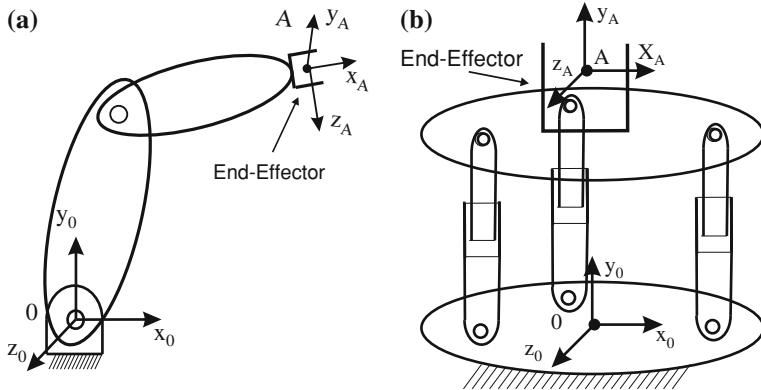
A load applied on a body produces deformations of the body itself that are known as compliant displacements. Stiffness can be defined as the property of a mechanical system in sustaining loads without too large compliant displacements. Stiffness can be also defined quantitatively as the amount of force that can be applied in one direction per unit of compliant displacement of the body in the same direction, or the ratio of a steady force acting on a body to the resulting compliant displacement [1–8].

Stiffness plays a key role both in the design and control of any robotic system for grasping tasks. Thus, stiffness is widely investigated for any robotic system

---

G. Carbone (✉)

LARM: Laboratory of Robotics and Mechatronics, University of Cassino and South Latium,  
Via G. Di Biasio, 43 03043 Cassino, FR, Italy  
e-mail: carbone@unicas.it

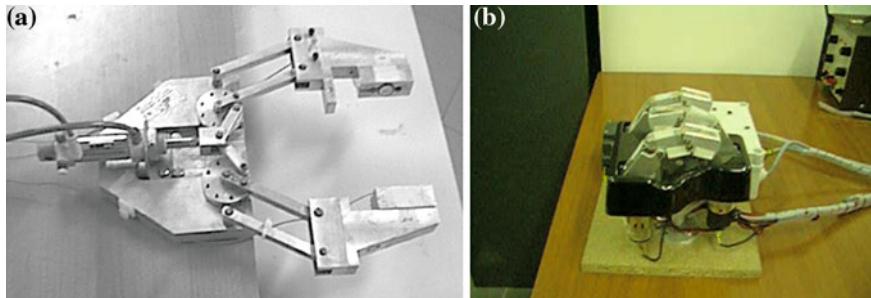


**Fig. 2.1** Schemes of multibody robotic systems: **a** A 2R serial manipulator; **b** A parallel manipulator with three RPR legs

spanning from conventional serial robots to non-conventional parallel manipulators, such as those that are schematized in Fig. 2.1. Few examples in a wide literature can be found in [6–28].

Given the peculiarities of grasping tasks, one should carefully study the stiffness performance of the robotic system but also of the robot extremity, which is generally denominated as end-effector. In fact, usually only the end-effector is directly interacting with the environment and the objects that have to be manipulated. There are many different types of end-effectors with different sizes, shapes, operation, and actuators principles, as stated for example in [2, 29–31]. Stiffness can be considered of particular significance for all those mechanically achieving the grasp. They may range from dedicated mechanical grippers having two fingers (widely used in industrial applications) up to versatile multi-fingered robotic hands (widely investigated for mimicking the high multi-purpose operation of human hands). Some examples from an extensive literature are reported in [32–40].

Several grasping devices have been also designed and built at LARM in Cassino, [41–49]. For example, Fig. 2.2 shows two prototypes at LARM: a two-finger gripper, Fig. 2.2a, and a robotic hand with three fingers, Fig. 2.2b, respectively.



**Fig. 2.2** Example of grasping devices that have been designed and built at LARM in Cassino: **a** A two-finger gripper; **b** The LARM Hand IV with three fingers

Many researchers have investigated stiffness with different approaches and focuses. Most of the published works on stiffness can be classified into three main categories. The first category deals with stiffness analysis and determination of overall stiffness. Given the stiffness of main components motors, joints, links, the overall stiffness has to be determined as reported, for example, in [7–18]. Once a proper stiffness model has been defined it can be used for controlling the grasp stability, or in compliance control algorithms as proposed for example in [19–21]. Moreover, a proper stiffness model can be used also for design purposes, for example, in order to find an optimum compromise between weight of links and stiffness performance as proposed in [22–24]. A second category studies the inverse decomposition of a stiffness matrix into constituent stiffness parameters that are often assumed to be simple linear springs, as proposed for example in [25]. In a third research line, mathematical properties of the stiffness matrix are investigated, mainly with the aim of finding intrinsic properties that are independent from the coordinate frame in which the stiffness matrix is expressed, [26–28].

Although stiffness is widely investigated there are still open problems. For example, experimental determinations and evaluations of stiffness performance are prescribed in standard codes for robotic manipulators, [50–52] that should be extended also to non-conventional robotic systems, grippers, and hands. Still an open issue can be considered also the formulation of computationally efficient algorithms that can give direct engineering insight of the design parameter influence and can be translated into experimental tests for experimental determinations. Moreover, still few preliminary comparisons of numerical results with experimental experience have been proposed, [53, 54].

## 2.2 Stiffness Modelling and Analysis

Usually, stiffness analysis of a robotic system is aiming to determine the stiffness performance through the computation of a  $6 \times 6$  Cartesian stiffness matrix  $K$ . This stiffness matrix  $K$  expresses the relationship between the compliant displacements  $\Delta S$  occurring to a frame fixed at the end of the kinematic chain when a static wrench  $W$  acts upon it and  $W$  itself. Considering Cartesian reference frames,  $6 \times 1$  vectors can be defined for the compliant displacements  $\Delta S$  and the external wrench  $W$  as

$$\begin{aligned} \Delta S &= (\Delta x, \Delta y, \Delta z, \Delta \alpha, \Delta \gamma, \Delta \delta)^t; \\ W &= (F_x, F_y, F_z, T_x, T_y, T_z)^t \end{aligned} \quad (2.1)$$

where  $\Delta x$ ,  $\Delta y$ ,  $\Delta z$ ,  $\Delta \alpha$ ,  $\Delta \gamma$ , and  $\Delta \delta$  are the linear and angular compliant displacements on the robotic system extremity;  $F_x$ ,  $F_y$ , and  $F_z$  are the force components acting on the robotic system extremity along  $X$ ,  $Y$ , and  $Z$  directions, respectively;  $T_x$ ,  $T_y$ , and  $T_z$  are the torque components acting upon the same point on the robotic system extremity about  $X$ ,  $Y$ , and  $Z$  directions, respectively.

Compliant displacements have usually negative effects on a robotic device for grasping tasks, since compliance detrimentally affect accuracy, repeatability, and payload capability. Additionally, in dynamic conditions, the presence of large compliant displacements can affect fatigue strength, can produce vibrations and energy losses. However, in some cases, compliant displacements can even have a positive effect if they are properly controlled, [55,56]. In fact, they can enable the correction of misalignment errors encountered; for example, when parts are mated during assembly operations [5] or in peg into hole tasks [21], or in deburring tasks, [57], or in the operation of a prosthesis [58]. It is to note also that a stiffer behavior is often obtained at cost of an higher own weigh of a robotic system that can rise manufacturing costs and detrimentally affect dynamic performance and power consumption. Thus, a proper stiffness modeling and analysis is of key significance to identify optimal trade-off solutions both at design and control stage.

Provided that the assumptions of small compliant displacements hold, one can write

$$\mathbf{K}(q) : \Re^r \rightarrow \Re^r, \quad \mathbf{W} = \mathbf{K} \Delta \mathbf{S} \quad (2.2)$$

where  $\mathbf{K}$  is the so-called  $6 \times 6$  Cartesian or spatial stiffness matrix.

It is worth noting that according to the definition in Eq. (2.2), the stiffness matrix  $\mathbf{K}$  is in general posture dependant. Moreover, the stiffness matrix  $\mathbf{K}$  is generally nonsymmetric and its entries depend on choice of reference frame, since it is not reference frame invariant, as demonstrated for example in [2, 7, 16, 26–28].

The computation of stiffness matrix  $\mathbf{K}$  can be achieved with different approaches such as finite element methods (FEM) or methods based on models with lumped parameters (MLP). FEM methods can be used for a stiffness analysis of multibody robotic systems, although with very difficult numerical implementation. In fact, even if FEM methods could be more accurate than MLP methods they are time consuming and they require a complete recalculation at each configuration/loading condition under analysis. Therefore, the stiffness analysis of robotic systems is usually carried out by means of MLP methods that are based on using lumped stiffness parameters for taking into account the stiffness properties of links and joints with configuration dependant relationships. Therefore, main advantages of MLP methods can be understood in reduced computational efforts and possibility to use the same stiffness model for the analysis of several different configurations. These aspects give the possibility to investigate the stiffness performance through the whole workspace of a robotic system in a reasonable amount of computational time. Moreover, MLP methods can be conveniently used for developing parametric models within optimal design procedures.

Equation (2.2) defines  $\mathbf{K}$  as a  $6 \times 6$  matrix whose components are the amount of forces or torques that can be applied per unit of compliant displacements of the end effector for a robotic system. However, the linear expression in Eq. (2.2) is valid only for small magnitude of the compliant displacements  $\Delta \mathbf{S}$ . Moreover, Eq. (2.2) is valid only in static (or quasi static) conditions. The entries of a  $6 \times 6$

stiffness matrix can be obtained through the composition of suitable matrices. A first matrix  $C_F$  gives all the wrenches  $\mathbf{W}_L$ , acting on the links when a wrench  $\mathbf{W}$  acts on the manipulator extremity as

$$\mathbf{W} = \mathbf{C}_F \mathbf{W}_L \quad (2.3)$$

with the matrix  $C_F$  representing the force transmission capability of the manipulator mechanism.

A second matrix  $K_p$  gives the possibility to compute the vector  $\Delta v$  of all the deformations of the links when each wrench  $\mathbf{W}_{Li}$  on a  $i$ th link given by  $\mathbf{W}_L$ , acts on the legs according to

$$\mathbf{W}_L = \mathbf{K}_p \Delta v \quad (2.4)$$

with the matrix  $\mathbf{K}_p$  grouping the lumped parameters of the robotic system.

A third matrix  $C_K$  gives the vector  $\Delta S$  of compliant displacements of the manipulator extremity due to the displacements of the manipulator links, as expressed as

$$\Delta v = \mathbf{C}_K \Delta S \quad (2.5)$$

Therefore, the stiffness matrix  $K$  can be computed as

$$\mathbf{K} = \mathbf{C}_F \mathbf{K}_p \mathbf{C}_K \quad (2.6)$$

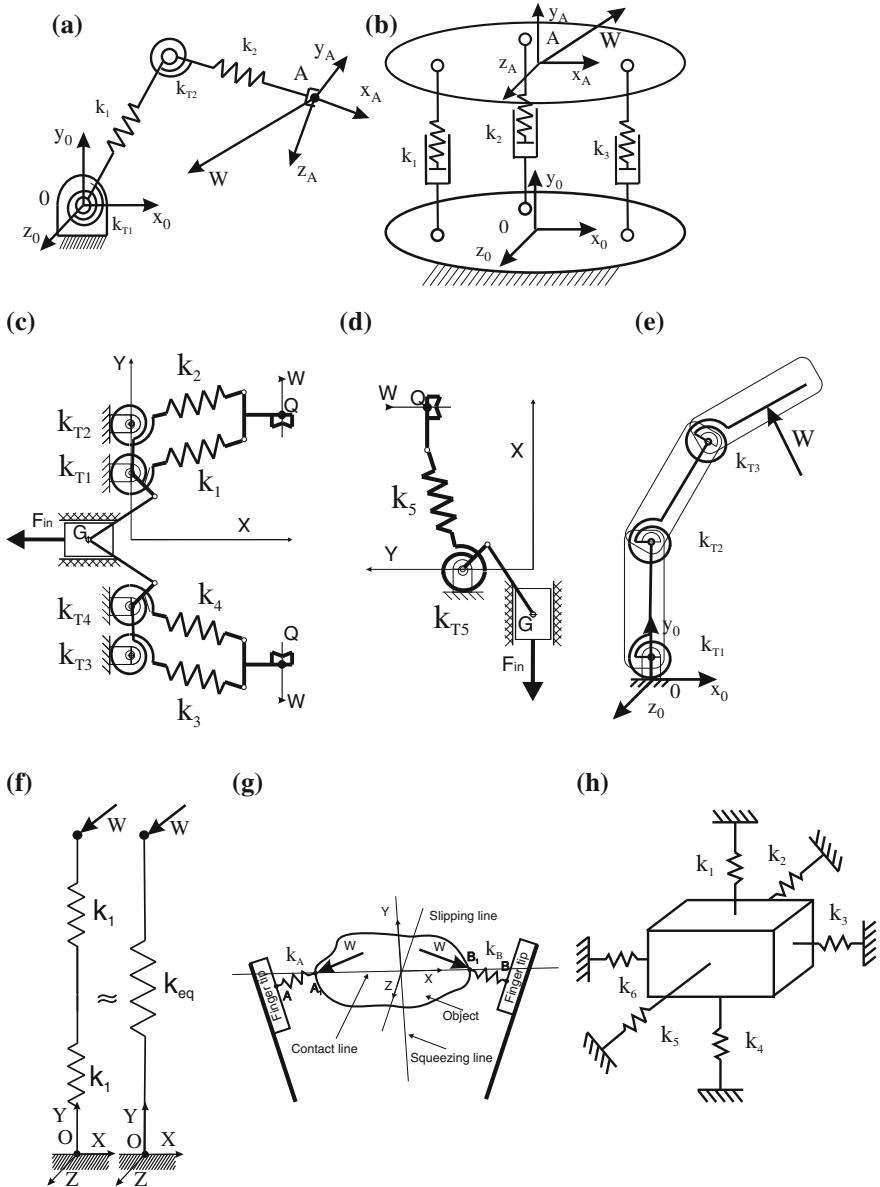
with matrix  $C_F$  giving the force transmission capability of the mechanism;  $K_p$  grouping the spring coefficients of the deformable components;  $C_K$  considering the variations of kinematic variables due to the deformations and compliant displacements of each compliant component.

Matrices  $C_K$  and  $C_F$  can be computed, for example, as a Jacobian matrix  $J$  and its transpose, respectively, as often proposed in the literature. Thus, one can compute the Cartesian stiffness matrix as

$$\mathbf{K} = J^t \mathbf{K}_p J \quad (2.7)$$

Nevertheless, this is only an approximate approach as pointed out, for example, in [59]. A more accurate computation of matrices  $C_K$  and  $C_F$  can be obtained as reported, for example in [2, 7, 13]. The  $K_p$  matrix can be computed as a diagonal matrix whose components are the lumped stiffness parameters of links, joints, and motors that compose a multibody robotic system. The lumped stiffness parameters can be estimated by means of analytical and empirical expressions or by means of experimental tests.

The lumped stiffness parameters can be graphically represented as linear or torsion springs. Examples of stiffness models with lumped parameters are shown in Fig. 2.3. In particular, Fig. 2.3a shows a stiffness model with lumped parameters for the 2R serial manipulator in Fig. 2.1a. This model considers the stiffness of the two motors and joints by means of two lumped parameters  $k_{T1}$ ,  $k_{T2}$ .



**Fig. 2.3** Examples of stiffness models with lumped parameters: **a** For the 2R serial manipulator in Fig. 2.1a; **b** For the parallel manipulator in Fig. 2.1b; **c** For the two finger gripper in Fig. 2.2a; **d** For the symmetric part of the two finger gripper in Fig. 2.3c; **e** For one finger of the LARM Hand IV in Fig. 2.2b; **f** Two linear springs and their equivalent as a single linear spring; **g** For the grasping of a generic object with a two finger gripper; **h** For representing a generic stiffness matrix with a six linear springs model

Additionally, the axial stiffness of the links is considered by means of the lumped parameters  $k_1, k_2$ . These two scalar lumped parameters can be replaced with two  $6 \times 6$  stiffness matrices  $K_1, K_2$  that can be even obtained by means of finite element softwares for taking into account the whole stiffness behavior of the links. Similarly, Fig. 2.3b shows a stiffness model with lumped parameters for the parallel manipulator in Fig. 2.1b. This model considers the stiffness of the two prismatic actuators and links by means of two lumped parameters  $k_1, k_2$ . These two scalar lumped parameters can be replaced with two  $6 \times 6$  stiffness matrices  $K_1, K_2$  that can be even obtained by means of finite element softwares for taking into account the whole stiffness behavior of the links.

Figure (2.3c) shows a stiffness model for the two-finger gripper in Fig. 2.2a. This model takes into account the stiffness of joints and links by means of the lumped parameters  $k_{T1}, k_{T2}, k_{T3}, k_{T4}$ , and  $k_1, k_2, k_3, k_4$ , respectively. It is to note that the two-finger gripper in Fig. 2.2a has a symmetric design. Thus, one can study only half of the model. Additionally, the effect of the lumped parameters  $k_{T1}, k_{T2}$  and  $k_1, k_2$ , can be combined in the lumped parameters  $k_{T5}$  and  $k_5$ , respectively, as shown in the simplified scheme of Fig. 2.3d. Similarly, Fig. 2.3e shows a simplified stiffness model for the robotic hand in Fig. 2.2b. In this simplified model, the stiffness properties of the motor and the driving mechanism as well as the flexional properties of the links have been combined in the lumped parameters  $k_{T1}, k_{T2}, k_{T3}$ . It is to note that it is advisable to keep the number of chosen lumped stiffness parameters as equal to the desired rank of the stiffness matrix (that is equal to six in the spatial case, to three in the planar case). In fact, choosing a different number of lumped stiffness parameters yields to nonsquare matrices that can lead to computation problems. For this purpose, if the superposition principle holds, one can combine the effect of more compliance sources in a single lumped parameter as described in the example of Fig. 2.3f where the lumped stiffness parameters  $k_1$  and  $k_2$  have been combined in the equivalent lumped parameter  $k_{eq}$ , whose value can be obtained as

$$(k_{eq})^{-1} = (k_1)^{-1} + (k_2)^{-1} \quad (2.8)$$

The grasping of an object also should require proper stiffness models that need to take into account the contact forces, the position of contact points/areas, the curvature of surfaces in contact. For example, Fig. 2.3f shows a simplified stiffness model with lumped parameters for the grasping of a generic object with a two-finger gripper. The contact areas on the two fingertips are assumed to coincide with the contact points A and B along the contact line. The stiffness properties of the contact are lumped in the parameters  $k_1, k_2$  that are shown in Fig. 2.3f as ideal springs having no mass and length A-A<sub>1</sub> and B-B<sub>1</sub> equal to zero. The grasping model can become more complex as the one in Fig. 2.3f if one considers contact areas, multiple contact points, variable curvature of surfaces, and effect of friction on stiffness. In these cases, the scalar lumped parameters  $k_1, k_2$  can be replaced by  $6 \times 6$  stiffness matrices  $K_A$  and  $K_B$ . It is to note that any  $6 \times 6$  stiffness matrix

can be decomposed in a model with linear springs having scalar lumped parameters such as the general model with six linear springs that is shown in Fig. 2.3g.

Stiffness properties usually have different expressions according to the chosen reference frame. Thus, each stiffness model should clearly indicate the chosen reference frame. For example, in two-finger grasping models as in Fig. 2.3f a Cartesian reference frame can be chosen with axes coinciding with the contact line, squeezing line, and slipping line.

The Cartesian stiffness matrix  $K$  is posture dependent. Thus, one should define configuration(s) of a multibody robotic system where the stiffness matrix can be computed. The configuration(s) should be carefully chosen in order to have significant information on the stiffness performance of the system in its whole workspace. Then, the kinematic model can be used for computing the vector  $\theta$  that express input angles and strokes in the Joint Space for any posture. It is worth noting that the accuracy in the estimation of model data such as geometrical dimensions and values of lumped stiffness parameters can significantly affect the accuracy of the computed stiffness matrix. Thus, experimental tests should be carried out in order to validate model data and overall stiffness model.

Once the stiffness matrix has been computed, it is also necessary to give synthetic evaluation of the stiffness performance both for analysis and design purposes. Thus, an index of merit can be formulated by using properties of the stiffness matrix, so that it represents numerically the stiffness performance of a new multibody robotic system.

The current standard codes for stiffness evaluation of manipulators are given as short parts of the norms ANSI/RIA15.01.11-1990 [51] and ISO9283-1995 [52], which refer explicitly to serial chain industrial robots only. In particular, Sect. 8.6 in [51] and Sect. 10 in [52] are devoted to Static compliance with a very similar approach but only referring to a performance evaluation through measures of position compliant displacements. Then, a recommendation states to express the results in term of millimeters per Newton for displacements that are referred to the directions of a base coordinate system. Thus, the standard codes do not yet consider the stiffness matrix as a performance index for the elasto-static response of multibody robotic manipulators, but they still refer to a practical evaluation with a direct natural interpretation that is related to the compliance response of the stiffness of a manipulator structure. Of course, it is evident that compliant displacements can be considered as a measure of the manipulator stiffness since the fundamental relationship in Eq. (2.1). Nevertheless, the compliance response is system posture and wrench direction dependant since one can find a  $6 \times 1$  vector of compliant displacements at any posture for any wrench. Therefore, one should define a single local index of stiffness performance and then a global index expressing the stiffness performance in the overall workspace of a multibody robotic system.

A local stiffness index can be directly related with the Cartesian stiffness matrix by means of different mathematical operators that can be applied to a matrix, as proposed for example in [54].

Compliant displacements can also provide an insight on local stiffness performance due to their simple physical interpretation, as indeed suggested by ISO and ANSI codes [51, 52]. In fact, one can compute the compliant displacements for a given configuration by multiplying the computed stiffness matrix for a given external wrench  $\mathbf{W}_{\text{Given}}$ . Reasonable choices for  $\mathbf{W}_{\text{Given}}$  can be a unit vector or a vector equal to the expected payload for a multibody robotic system as proposed for example in [7]. The first choice gives a measure of the compliant displacements per unit of external wrench. The second choice provides a measure of the maximum compliant displacements for the system in specific applications. Nevertheless, compliant displacements have usually six components. Thus, they cannot be treated as a single merit index.

Eigenvalues and eigenvectors of a stiffness matrix are also very useful for their physical interpretation with respect to local stiffness performance. In fact, the eigenvectors are related with the maximum and minimum eigenvalue and they provide the directions of maximum and minimum stiffness performance, respectively. Moreover, a smaller difference among the eigenvalues stands for a smaller anisotropic stiffness behavior at a given posture. Nevertheless, eigenvalues and eigenvectors cannot be treated as a single merit index. But their values can be used for drawing graphical local representations of the stiffness performance such as compliance/stiffness ellipses and ellipsoids, as reported for example in [8]. These graphical local representations also provide a graphical tool for the comparison of stiffness performance along and about different directions. The graphical representations can be very useful when specific design requirements arise. In particular, they are useful if there is a need of the best stiffness performance only in a given direction or if equal stiffness is preferred in all directions.

Other graphical tools for a comparison of stiffness performance can be obtained through the definition of the so-called center of stiffness or the center of compliance and by means of stiffness or compliant axes that can be used for defining directions and orientations in which a robotic system acts as a simple spring, as mentioned for example in [22].

A local index of stiffness performance is neither suitable for an accurate design analysis nor useful for a comparison of different designs. In fact, even if a robotic system has suitable stiffness for a given system posture it can have inadequate stiffness at other postures. Therefore, one should look at stiffness performance at all points of workspace or define a single global stiffness index over the whole workspace yet.

A global index of stiffness performance for a robotic system can be defined with graphical methods that are based on plotting curves connecting postures having the same value of the local stiffness index (iso-stiffness curves or surfaces), as proposed for example in [6]. Nevertheless, the number of iso-stiffness curves or surfaces that one can plot is graphically limited. Moreover, few curves or surfaces usually do not provide sufficient insight of the overall stiffness behavior of a robotic system. These aspects significantly reduce the effectiveness of iso-stiffness curves or surfaces.

Global stiffness indices can be defined also in a mathematical form by using minimum, maximum, average, or statistic evaluations of a local stiffness index. For example, one can compute a global index in the form

$$GI_{MN} = \frac{\int \max_{i=1,\dots,6} \{ \sqrt{\lambda_i^*} \} dV}{L^3} \quad (2.9)$$

where  $\{ \sqrt{\lambda_i^*} \}$  is the set of nonnegative eigenvalues of  $KK^T$   $V$  is the workspace volume;  $L$  is a characteristic length that is used in order to obtain information that is independent from the workspace volume. Alternatively to  $L^3$ , the denominator can be expressed as the volume  $V$  of workspace. Moreover, the dimensional inconsistency can be solved by using a proper dimensionless value of the merit index (that is indicated with a superscript  $*$ ) that can be obtained by dividing the length entries by a characteristic length  $L$ . This global index can be useful when a design goal is to maximize the stiffness performance along or about one or more specific direction(s). A similar global stiffness index can be defined by referring to the minimum eigenvalue as

$$GI_{mN} = \frac{\int \min_{i=1,\dots,6} \{ \sqrt{\lambda_i^*} \} dV}{L^3} \quad (2.10)$$

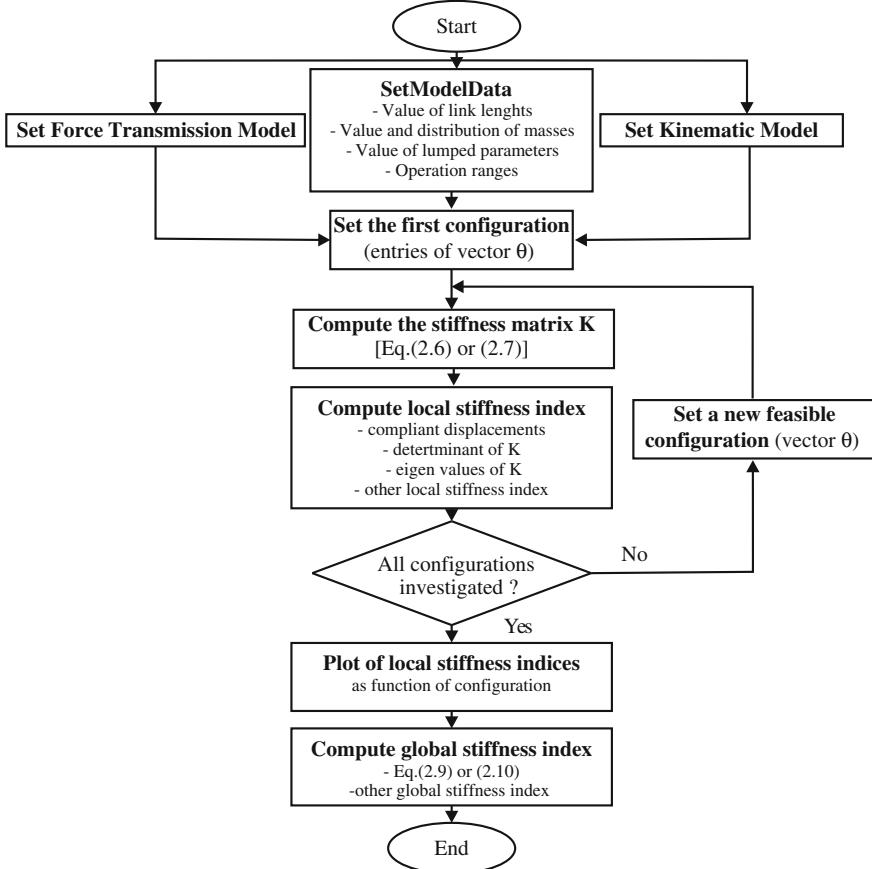
This global stiffness index can be useful to detect and avoid design with weak stiffness performance along or about a specific direction. A global index can be defined also as the difference between  $GI_{MN}$  and  $GI_{mN}$ .

It is to note that the integration operator in Eqs. (2.9) and (2.10) is usually numerically calculated, since the analytical expression of  $\{ \sqrt{\lambda_i^*} \} dV$  is usually not available. Thus, for comparison purposes it is advisable to have the same number of calculation configurations.

## 2.3 Numerical Computation of Stiffness Performance

The models and formulations in the previous section can be used to get a numerical insight of the stiffness performance for a robotic system. In particular, a numerical algorithm can be composed of a first part in which all the model data are provided such as the numerical values of the geometrical dimensions, masses, and lumped stiffness parameters. A second part defines the kinematic model, the force transmission model, and the lumped parameter model through the matrices  $C_F$ ,  $C_K$ , and  $K_p$ , respectively. Then, a third part can compute an expression of the stiffness matrix  $K$  by means of Eqs. (2.6) or (2.7), as shown in the flowchart of Fig. 2.4.

It is worth noting that the matrices  $C_F$ , and  $C_K$  as well as the Jacobian matrix and its transpose are configuration dependant. Thus, they may have nonlinear components in a complex formulation. In general, this makes very difficult to



**Fig. 2.4** A flowchart for the proposed numerical computation of stiffness performance

identify a close-form formulation of the stiffness matrix  $K$ . Thus, often the matrices  $C_F$  and  $C_K$  or the Jacobian matrix are numerically computed at a given configuration, and then combined into Eqs. (2.6) or (2.7) to calculate the stiffness matrix  $K$ . However, one should carefully define the configuration(s) where the stiffness matrix will be computed. The configuration(s) should be carefully chosen in order to have significant information on the stiffness performance of the system in its whole workspace. Careful attention should be addressed also to avoid numerical singularities for the stiffness matrix  $K$  that might occur also due to nonlinear terms. However, a linearization might be possible under the assumption of small compliant displacements.

Once the stiffness matrix has been computed, it can compute local stiffness indices at a given configuration. Different local stiffness indices can be chosen as also mentioned in the previous section. Values of local stiffness indices allow to compare the stiffness performance of a robotic system at different configurations.

An iterative loop can be defined to investigate a significant (but finite) number of configurations within the workspace of a robotic system. In some cases, a robotic system can have few trajectories that are mostly used during its operation. In these cases, the kinematic model can be used together with a proper path planning strategy for properly computing the time evolution of all the entries in the vector  $\theta$  that express input angles and strokes in the joint space as function of time for a given trajectory. Thus, the vector  $\theta(t)$  can be used for computing the stiffness matrix as function of time for a given end-effector trajectory.

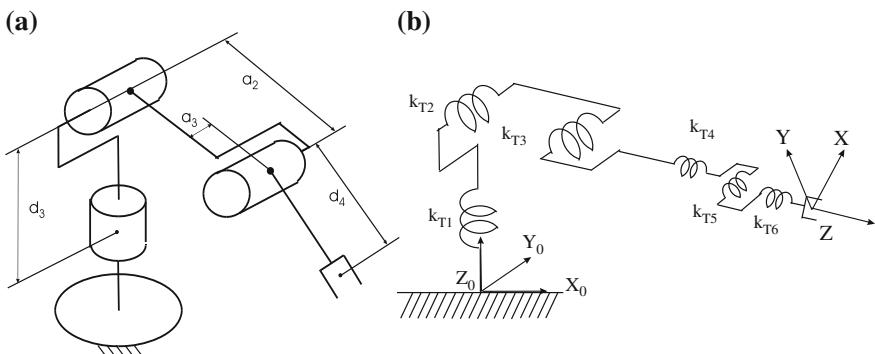
After completing the calculation of the local stiffness indices one can plot their values as function of the configuration. Additionally, one can plot other graphical representations of stiffness properties such as the iso-stiffness curves. Finally, one can compute the global stiffness parameter, for example, by using Eqs. (2.9) or (2.10). These global values are useful for quantitative comparisons of different robotic systems especially at design stage.

It is to note that a proper stiffness analysis of a robotic system for grasping tasks should start from the identification of the main sources of compliance. In fact, there is always a trade-off between accuracy of the model and computational costs. For this reason, very often the stiffness analysis is limited to the robot architecture while one should carefully consider also the end effector.

## 2.4 Cases of Study for Stiffness Modelling and Analysis

### 2.4.1 A 6R Serial Manipulator

A 6 DOFs PUMA-like manipulator has been considered as a case of study for the above-mentioned formulation as specifically applied to a serial type robotic system. Main design parameters for a PUMA-like manipulator are shown in Fig. 2.5a.



**Fig. 2.5** Models for a PUMA-like manipulator: **a** Main dimensional parameters; **b** Lumped stiffness parameters

**Table 2.1** Main design parameters and workspace ranges for a PUMA 562

$a2$ (mm)	$a3$ (mm)	$d3$ (mm)	$d4$ (mm)	$x$ (mm)	$y$ (mm)	$z$ (mm)	$\phi$ (°)	$\psi$ (°)	$\theta$ (°)
431.8	20.3	125.4	431.8	529.2	472.4	625.0	180	180	180

Values of the design parameters  $a2$ ,  $a3$ ,  $d3$ , and  $d4$  are reported in Table 2.1. These values have been defined by referring to a PUMA 562 design and the mobility ranges for the joint angles have been assumed equal to 180° for the first three joints (of the arm) and 90° for the last three joints (of the wrist).

A simplified stiffness model for a PUMA-like manipulator is shown in Fig. 2.5b. In this model, the links have been considered as rigid bodies. In fact, in this type of robots, the payloads are limited and the compliant displacements are in general due to the flexibility of joints only. The link compliant displacements are much smaller than the compliant displacements that are due to the compliance of motors as pointed out for example in [44]. Thus, in the model of Fig. 2.5b the lumped parameters  $kT1$  to  $kT6$  take into account the stiffness motors and joints only.

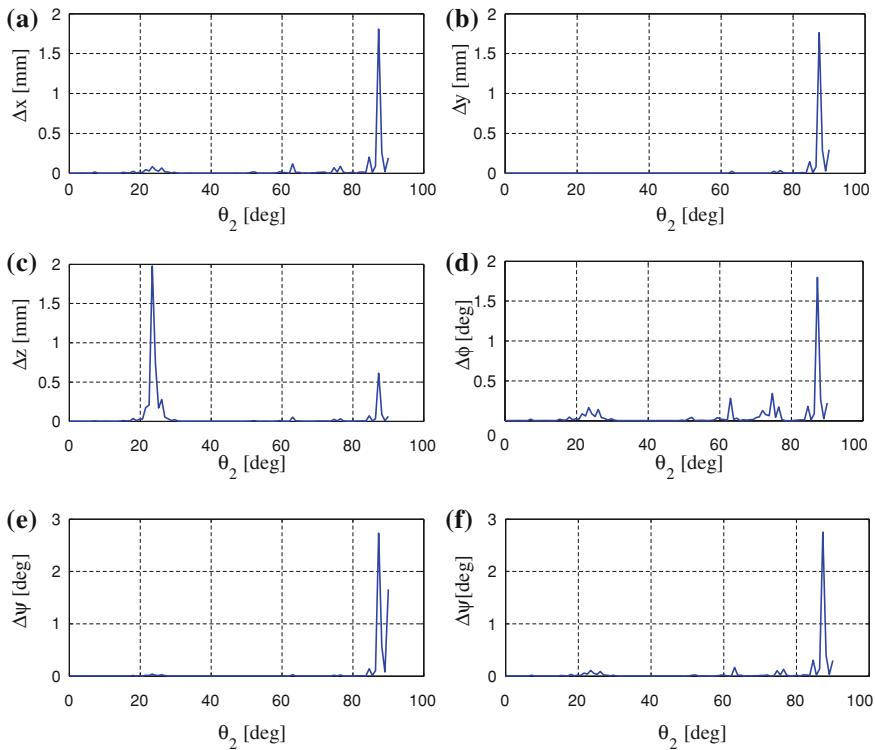
Moreover, if the only contributions to the overall compliance are given by motor compliances, the stiffness matrix  $K$  can be computed through Eq. (2.7) where  $J$  is the well-known Jacobian matrix of the PUMA-like robot. The matrix  $KP$  in Eq. (2.7) can be computed as a diagonal matrix with lumped stiffness parameters of the motors that can be set as  $kT1 = kT2 = kT3 = 5 \times 10^6$  Nm/rad and  $kT3 = kT4 = kT5 = 5 \times 10^4$  Nm/rad as reasonable values by referring to a PUMA 562. The stiffness matrix of the PUMA-like robot that can be computed through Eq. (2.7) as function of the input joint angles. The expressions of the input joint angles can be computed as functions of the coordinates ( $x$ ,  $y$ ,  $z$ ,  $\phi$ ,  $\psi$ ,  $\theta$ ) for the position and orientation of the end effector from the well-known inverse Kinematics of PUMA-like robot.

It is worth noting that the accuracy in the estimation of model data such as geometrical dimensions and values of lumped stiffness parameters can significantly affect the accuracy of the stiffness matrix that is computed through Eq. (2.7). Thus, experimental tests should be carried out in order to validate stiffness model and model data.

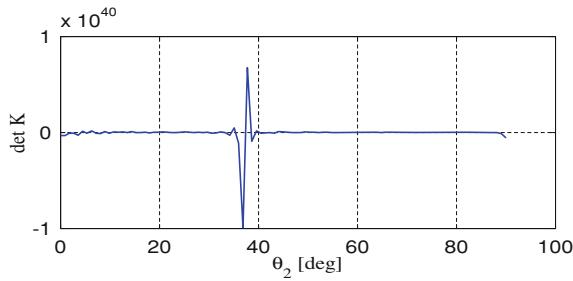
Results of the proposed stiffness analysis as applied to the PUMA-like architecture are reported in Table 2.2 and Figs. 2.6 and 2.7.

**Table 2.2** Maximum values of compliant displacements and values of global stiffness indices within the feasible workspace of PUMA 562

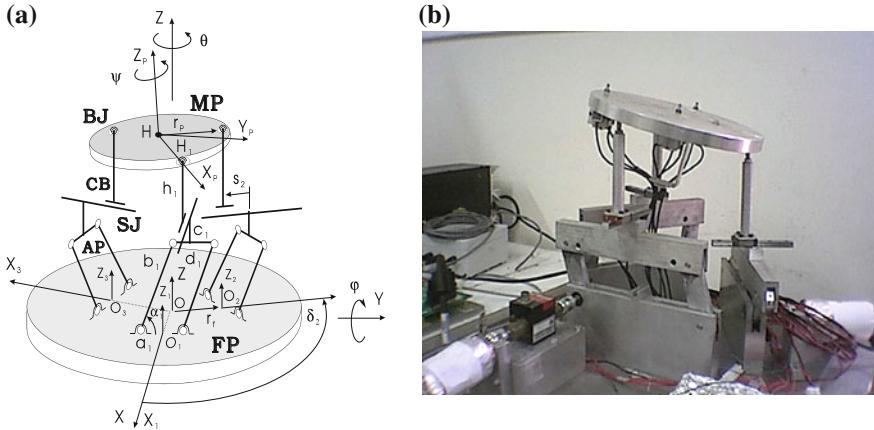
$\Delta x$ (mm)	$\Delta y$ (mm)	$\Delta z$ (mm)	$\Delta\phi$ (°)	$\Delta\psi$ (°)	$\Delta\theta$ (°)	GIMn	GImn
1.81	1.76	1.98	1.80	2.73	2.75	5.6202e + 034	5.3184e + 017



**Fig. 2.6** Compliant displacements of Puma-562 as a function of the input angle  $\theta_2$ : **a** Linear compliant displacement along X-axis; **b** Linear compliant displacement along Y-axis; **c** Linear compliant displacement along Z-axis; **d** Angular compliant displacement about X-axis; **e** Angular compliant displacement about Y-axis; **f** Angular compliant displacement about Z-axis



**Fig. 2.7** Determinant of the matrix  $K$  for the case in Fig. 2.6



**Fig. 2.8** CaPaMan (Cassino Parallel Manipulator) design: **a** A kinematic diagram; **b** A built prototype at LARM

### 2.4.2 A 3 DOF Parallel Manipulator

The Cassino parallel manipulator (CaPaMan) has been considered to test the engineering feasibility of the above-mentioned formulation as specifically applied to parallel architectures which can be different from a general Gough-Stewart platform. CaPaMan architecture has been conceived at LARM in Cassino since 1996, where a prototype has been built for experimental activity. A schematic representation of the CaPaMan manipulator is shown in Fig. 2.8a, and the prototype is shown in Fig. 2.8b. Indeed, by using the existing prototype, simulations have been carried out to compute its stiffness performance.

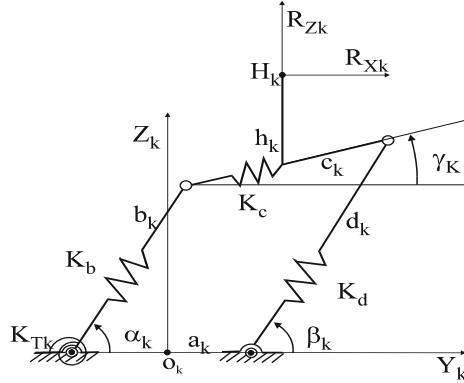
Kinematics of CaPaMan manipulator has been already investigated in previous works at LARM. In particular, matrices A and B have been formulated in the form

$$A = \begin{bmatrix} (D - F)b_1 c\alpha_1 & (D + 2F)b_2 c\alpha_2 & -(2D + F)b_3 c\alpha_3 \\ (D + F)b_1 c\alpha_1 & -Db_2 c\alpha_2 & -F b_3 c\alpha_3 \\ b_1 c\alpha_1 & b_2 c\alpha_2 & b_3 c\alpha_3 \end{bmatrix} \quad (2.11)$$

$$B = \begin{bmatrix} (6E/\sqrt{3}) & 0 & 0 \\ 0 & \sqrt{E(9r_p^2 - 4E)} & 0 \\ 0 & 0 & 3 \end{bmatrix} \quad (2.12)$$

in which

$$\begin{aligned} E &= z_1^2 + z_2^2 + z_3^2 - z_1 z_2 - z_2 z_3 - z_1 z_3 \\ D &= 2z_2 - z_1 - z_3; \quad F = 2z_3 - z_1 - z_2 \end{aligned} \quad (2.13)$$



**Fig. 2.9** A scheme for stiffness evaluation of a CaPaMan leg

with

$$z_k = b_k \sin \alpha_k; \quad \text{for } k = 1, 2, 3 \quad (2.14)$$

By modeling each leg of CaPaMan as in Fig. 2.9, the stiffness matrix of CaPaMan can be computed by using Eq. (2.6) with

$$C_F = M_{FN}; \quad C_K = C_P^{-1} A_d^{-1} \quad (2.15)$$

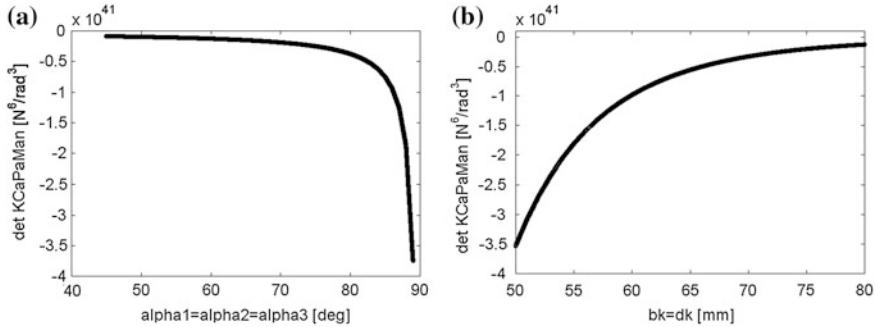
where  $M_{FN}$  is a  $6 \times 6$  transmission matrix for the static wrench applied on  $H$  and transmitted to points  $H_1$ ,  $H_2$  and  $H_3$  of each leg;  $K_p$  is a  $6 \times 6$  matrix with the lumped stiffness parameters of the three legs;  $C_p$  is a  $6 \times 6$  matrix giving the displacements of the links of each leg as a function of the displacements of points  $H_1$ ,  $H_2$ , and  $H_3$ ;  $A_d$  is a  $6 \times 6$  matrix that has been obtained by using the Direct Kinematics of the CaPaMan to give the position of point  $H$  on the movable plate as function of the position of points  $H_1$ ,  $H_2$ , and  $H_3$  in the form

$$\mathbf{X}_H = A_d \mathbf{v} \quad (2.16)$$

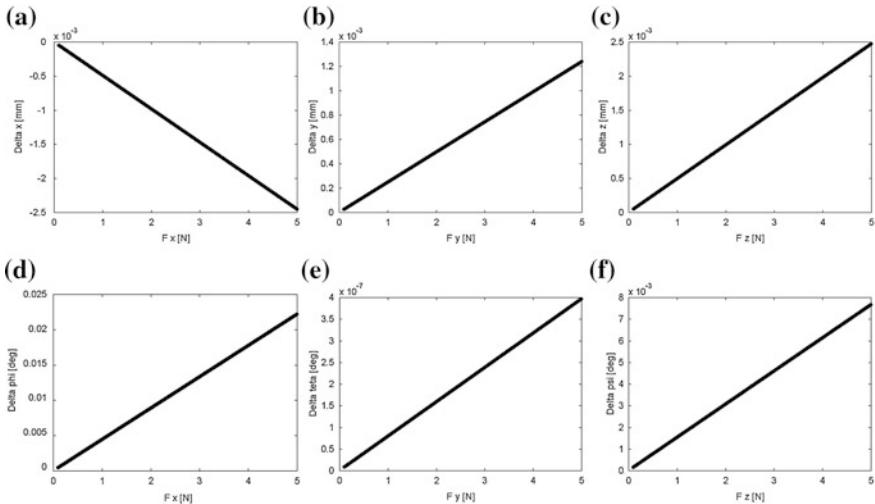
with  $\mathbf{v} = [y_1, z_1, y_2, z_2, y_3, \Delta z_3]^T$  and  $\mathbf{X}_H = [x_H, y_H, z_H, \varphi, \theta, \psi]^T$ . The derivation of matrices  $M_{FN}$ ,  $K_p$ ,  $A_d$ , and  $C_p$  for CaPaMan can be found in [18].

The lumped stiffness parameters have been assumed as  $k_{bk} = k_{dk} = 2.6 \times 10^6 \text{ N/m}$  and  $k_{Tk} = 58.4 \times 10^3 \text{ Nm/rad}$ ; the couplers  $c_k$  have been assumed rigid bodies because of the massive design that has been imposed to have a fix position of the sliding joints. Further, details on the derivation of the matrices in Eqs. (2.15) and (2.16) can be found in [18]. In the numerical example, for evaluation and design purposes we have assumed  $r_p = r_f$ ,  $a_k = c_k$ ,  $b_k = d_k$ .

Results of numerical simulations are shown in Figs. 2.10 and 2.11, while main numerical results for stiffness performance are summarized in Table 2.3.



**Fig. 2.10** Plots of the determinant of  $K_{CaPaMan}$ : **a** Versus  $\alpha_1 = \alpha_2 = \alpha_3$ ; **b** Versus  $b_k = d_k$  ( $k = 1,2,3$ ) with  $\alpha_1 = \alpha_2 = \alpha_3 = 60^\circ$



**Fig. 2.11** Compliant displacements of CaPaMan as a function of  $F_x = F_y = F_z$  when  $N_x = N_y = N_z = 0$  for  $\alpha_1 = \alpha_2 = \alpha_3 = 60^\circ$ : **a**  $\Delta x$ ; **b**  $\Delta y$ ; **c**  $\Delta z$ ; **d**  $\Delta\phi$ ; **e**  $\Delta\theta$ ; **f**  $\Delta\psi$

**Table 2.3** Maximum values of compliant displacements and values of global stiffness indices within the feasible workspace of CaPaMan with  $\mathbf{W} = (1.0; 1.0; 1.0; 0.0; 0.0; 0.0)^T$

$\Delta x$ (mm)	$\Delta y$ (mm)	$\Delta z$ (mm)	$\Delta\phi$ ( $^\circ$ )	$\Delta\psi$ ( $^\circ$ )	$\Delta\theta$ ( $^\circ$ )	GIMn	GImn
-0.0982	0.0309	0.0357	5.7932	0.0000	1.8620	4.8367e + 019	1.2882e + 017

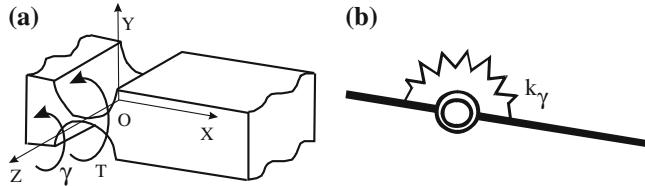
**Table 2.4** Illustrative value of forces as a function of the dimension L of an iron grasped object

Law	Type of force	Force [N] (at L = 1 mm)	Force [N] (at L = 100 mm))
$k_v L$	Van der Waals	0.119 $k_v = 1.19 \cdot 10^2$	$1.19 \cdot 10^{-9}$ $k_v = 1.19 \cdot 10^{-10}$
$k_e L^2$	Electro-static	0.014 $k_e = 1.40 \cdot 10^4$	$1.36 \cdot 10^{-6}$ $k_e = 1.36 \cdot 10^{-12}$
$k_G L^3$	Gravity	$3.00 \cdot 10^{-5}$ $k_G = 3.00 \cdot 10^4$	30.0 $k_G = 3.00 \cdot 10^4$
$k_I L^4$	Inertia	$3.06 \cdot 10^{-9}$ $k_I = 3.06 \cdot 10^3$	0.306 $kI = 3.06 \cdot 10^3$
$k_m L^4$	Magnetic force	$1.00 \cdot 10^{-6}$ $k_m = 1.00 \cdot 10^5$	$1.00 \cdot 10^{-1}$ $k_m = 1.00 \cdot 10^5$
$k_g L$	Grasping force	1 $k_g = 1 \cdot 10^3$	300 $kg = 3 \cdot 10^3$

### 2.4.3 A Two-Finger Milli-Gripper

The modeling and analysis of any grasping device should take into account its peculiarities and constraints. For example, the modeling and analysis of a milli-gripper should take into account many aspects including expected accuracy, dimensions, displacement ranges, acting forces, and loads. For example, the required accuracy can be considered inversely proportional to the geometric dimensions since smaller objects require greater positioning accuracy. The displacement and force capability are directly proportional to the geometric dimensions, since bigger objects require larger displacements of fingers and greater grasping force. For these reasons, in manipulative tasks at small scale the needed accuracy is high, while displacement and force capability are of small magnitude in agreement with the dimensions of the handled objects, [47].

A stiffness model for a two-finger milli-gripper is quite similar to the one shown in Fig. 2.3, even if important differences must be considered such as the scale of the acting forces and the kind of contact between the object and the fingers. As regards the scaling of the acting forces, Table 2.4 can be deduced by using dimensional analysis, similitude laws and experimental measurements as shown for example in [47]. Table 2.4 illustrates the main forces and their proportionality to the geometric dimension L of a grasped object by using coefficients  $k_v$ ,  $k_e$ ,  $k_G$ ,  $k_I$ ,  $k_m$ , and  $k_g$  that summarize have been computed from theoretical and experimental results in the literature. Moreover, Table 2.1 shows numerical examples, giving the intensity of different types of forces that have been evaluated in the case of L having a size of 1 and 100 mm. These examples show that, when the grasped object is of millimeter order, forces due to gravity and inertia can be neglected. The action of Van der Waals forces and electrostatic forces is still negligible as compared with the required grasping force at millimeter scale. A further scale reduction can make adhesive forces more significant than grasping forces, so that the releasing of the object becomes the most critical phase at micro- or even nano-scale [59].

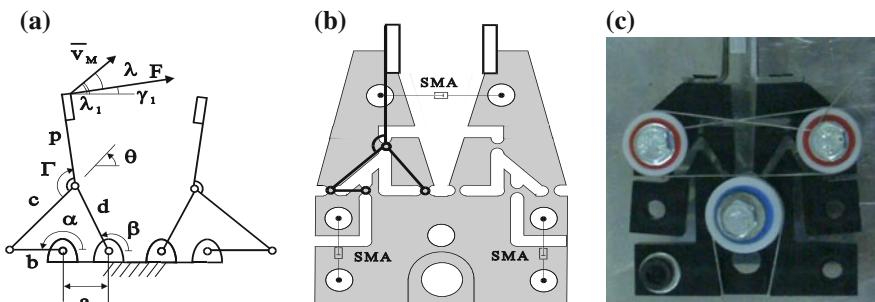


**Fig. 2.12** A flexural joint: **a** Manufacturing scheme; **b** Kinematic model

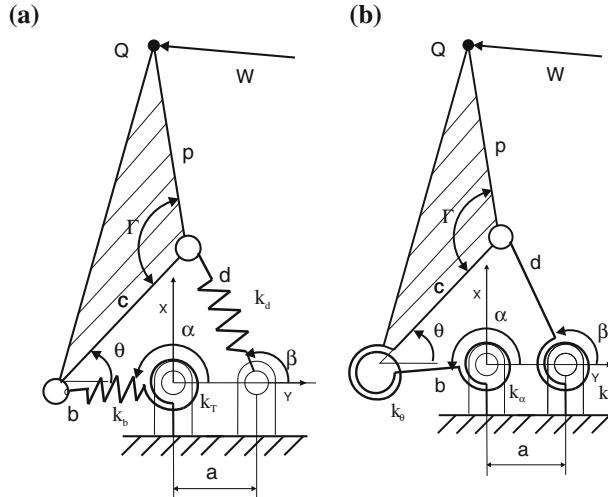
The operation of a mechanical milli-gripper strongly depends on the design and behavior of the driving mechanism, which transmits the motion and force to the gripping fingers. Theoretically, a milli-gripper could have the same mechanism type of a conventional gripper. However, these mechanisms are not always feasible due to the small dimensions. In fact, for example, conventional joints cannot be easily miniaturized. This problem can be solved, for example, by using flexural joints. In fact, flexural joints can be manufactured from a single piece of material by using milling machines to provide a monolithic mechanism, which eliminates interface wear and allow very high miniaturization, as pointed out in [44, 46, 47].

Figure 2.12a and b shows a design scheme and a kinematic model for a flexural joint obtained by manufacturing two notches on a single piece of material to have rotations about the Z-axis related with the stiffness parameter  $k_\gamma$ . The stiffness about the X-axis and the Y-axis is much higher than about the Z-axis. Therefore, the rotations about the X-axis and the Y-axis can be neglected and the flexural joint allows only a rotation  $\gamma$  of few degrees about the Z-axis when a torque  $T$  is applied. Actual misalignment of the actuation force or any unexpected forces may cause rather large parasitic deflections in other direction than the desired one. However, generally, this is enough for the microworld applications.

Figure 2.13 shows a design solution that has been proposed at LARM in Cassino. It uses flexural joints to obtain four-bar Chebichev type driving mechanism that allows an approximately straight-line motion of the fingers as shown in the kinematic chain of Fig. (2.1c). Considering this specific planar case, one can



**Fig. 2.13** A milli-gripper design: **a** Kinematic chain with design parameters; **b** A design scheme considering shape memory alloy (SMA) actuators; **c** A built prototype at LARM



**Fig. 2.14** Stiffness models with lumped parameters of the milli-gripper shown in Fig. 2.13: **a** With two linear and one torsional lumped parameters; **b** With three torsional lumped parameters

define the stiffness matrix  $K$  as a  $3 \times 3$ . The vector of compliant displacements can be defined as  $\Delta S = [\Delta x, \Delta y, \Delta \theta]^t$ . The external acting wrench acting on the point Q can be defined as  $W = [F_x, F_y, 0]^t$  when the external moment is assumed to be equal to zero. Referring to Fig. 2.13a a denotes the length of the frame, b and d are respectively the input and the output length links, and c is the length of the coupler whose fingertip point Q is located by the length p and the angle  $\Gamma$ .

Stiffness models of the milli-gripper in Fig. 2.13 can be proposed as shown in Fig. 2.14 by considering three lumped parameters. In particular, Fig. 2.14a shows a stiffness model with two linear lumped parameters that express the linear stiffness of the input and output links and one angular lumped parameter that express the stiffness of the input actuator and input flexural joint. Figure 2.14b shows a stiffness model with three angular lumped parameters that express the angular stiffness of the corresponding three flexural joints.

The bending of links and the stiffness of the actuator(s) can be also taken into account as additive components of these angular lumped parameters.

If OXY is a fixed reference and assuming that  $\alpha$  is the input angle, positive counterclockwise,  $\beta$  the output angle, and  $\theta$  the angle between the generic position of c and X-axis, from the loop closure equations one can write

$$\begin{aligned} x &= b \cos \alpha + c \cos \theta - p \cos(\Gamma - \theta) \\ y &= b \sin \alpha + c \sin \theta + p \sin(\Gamma - \theta) \\ \theta &= 2 \tan^{-1} \frac{\sin}{\alpha} - (i \sin^2 \alpha + B^2 - D^2)^{1/2} B + D \end{aligned} \quad (2.17)$$

with  $x$  and  $y$  being the position of the point Q along the X- and Y-axis, respectively and with

$$\begin{aligned} B &= \cos\alpha - \frac{a}{b}; \quad C = \frac{a^2 + b^2 - c^2 + d^2}{2bd} - \frac{a}{d}\cos\alpha; \\ D &= \frac{a}{c}\cos\alpha - \frac{a^2 + b^2 + c^2 - d^2}{2bc} \end{aligned} \quad (2.18)$$

Thus, the kinematic equations (2.17) and (2.18) can be used to obtain the matrix  $C_k$ .

If one refers to the stiffness model in Fig. 2.13a, the static equilibrium can be expressed by referring to the equilibrium of the coupler as

$$\begin{vmatrix} F_x \\ F_y \\ T_z \end{vmatrix} = \begin{vmatrix} k_b\cos\alpha & k_d\cos\beta & -\frac{k_T}{b}\sin\alpha \\ k_b\sin\alpha & k_d\sin\beta & \frac{k_T}{b}\cos\alpha \\ k_b r_b & -k_d r_d & \frac{k_T}{b} r_T \end{vmatrix} \begin{vmatrix} \Delta b \\ \Delta d \\ \Delta\alpha \end{vmatrix} \quad (2.19)$$

with

$$\begin{aligned} r_b &= \frac{c}{2}\sin(\alpha + \theta) + p\cos(\alpha + \theta); \quad r_d = -\frac{c}{2}\sin(\alpha + \beta) + p\cos(\alpha + \beta); \\ r_T &= \frac{c}{2}\cos(\alpha - \theta) + p\sin(\alpha - \theta) \end{aligned} \quad (2.20)$$

where the moment of the forces has been computed about point Q.

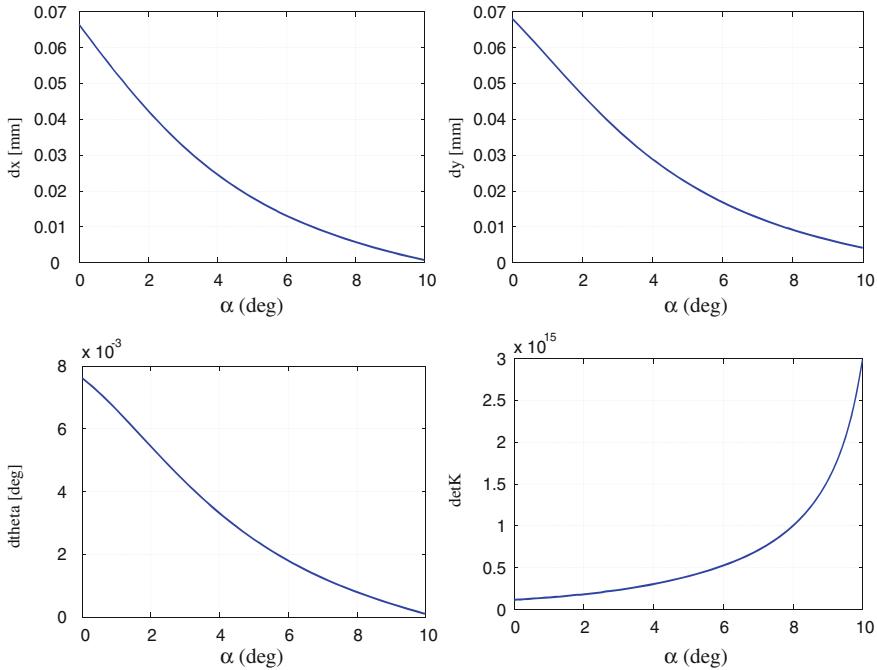
By using Eq. (2.19) one can compute the product of the matrices  $CF$  and  $K_p$  in the form

$$CFK_p = \begin{vmatrix} k_b\cos\alpha & k_d\cos\beta & -\frac{k_T}{b}\sin\alpha \\ k_b s\alpha & k_d\sin\beta & \frac{k_T}{b}\cos\alpha \\ k_b r_b & -k_d r_d & \frac{k_T}{b} r_T \end{vmatrix} \quad (2.21)$$

Thus, Eqs. (2.17–2.21) can be used to compute the stiffness matrix  $K$  of the milli-gripper as described in Eq. (2.6).

It is to note that it is usually not simple to find a close form expression for the stiffness matrix even for a rather simple mechanism. A simplified expression of the stiffness matrix one can obtained by referring to the stiffness model in Fig. 2.6b and by using the Jacobian matrix of the proposed mechanism. For example, if one assumes  $\Gamma = \pi$ ,  $a = 0.005$ ,  $b = 0.01$ ,  $c = 0.005$ ,  $d = 0.01$ ,  $p = 0.01$  m, the Jacobian matrix can be computed as

$$J = \begin{bmatrix} \frac{-0.01\sin\alpha+0.02\sin(\pi+\alpha\sin(-2\text{tex}\sin\beta+2\sin\alpha))}{\sqrt{1-4(\sin\beta-\sin\alpha)^2}} & \frac{-0.02\sin(\pi+\alpha\sin(-2\sin\beta+2\sin\alpha)\cos\beta)}{\sqrt{1-4(\sin\beta-\sin\alpha)^2}} & -0.005\sin\theta \\ 0.01\sin\alpha & \frac{-0.02\cos(\pi+\alpha\sin(-2\sin\beta+2\sin\alpha)\cos\beta)}{\sqrt{1-4(\sin\beta-\sin\alpha)^2}} & 0.005\cos\theta \\ \frac{2\sin(\alpha-\beta)}{\sin(\beta-\theta)} & 0 & 1 \end{bmatrix} \quad (2.22)$$



**Fig. 2.15** Simulation results by referring to the model in Eqs. (2.15, 2.16) versus the value of the input angle  $\alpha$  when  $\mathbf{W}_{\text{Given}} = [1, 1, 0]^T$ : **a** Linear compliant displacement along X-axis; **b** Linear compliant displacement along Y-axis; **c** Angular compliant displacement about Z-axis; **d** Determinant of the stiffness matrix  $K$

The matrix  $K_p$  can be defined according to the model in Fig. 2.14b as

$$K_p = \begin{vmatrix} k_\alpha & 0 & 0 \\ 0 & k_\beta & 0 \\ 0 & 0 & k_\theta \end{vmatrix} \quad (2.23)$$

Then, the Cartesian stiffness matrix can be easily obtained as in Eq. (2.7).

The stiffness matrix and compliant displacements can be computed at each configuration of the milli-gripper by means of Eqs. (2.17–2.21) or (2.22 and 2.23) by defining the values of the input angle and the values of the lumped stiffness parameters. For example, if one refers to the model in Fig. 2.14b, and one sets  $k_\alpha = k_\beta = k_\theta = 10^6$  Nm/rad as reasonable values for the proposed milli-gripper, one can use Eqs. (2.22) and (2.23) to compute the stiffness matrix  $K$  at any configuration. For example, the specific configuration when the input angle  $\alpha$  is equal to zero yields

$$K = 10^5 \begin{vmatrix} 4.1758 & -0.1215 & -1.9500 \\ -0.1215 & 0.4565 & -0.0979 \\ -1.9500 & -0.0979 & 1.0250 \end{vmatrix} \quad (2.24)$$

**Table 2.5** Maximum values of compliant displacements and values of global stiffness indices within the feasible workspace of the milli-gripper in Fig. 2.13

$\Delta x$ (mm)	$\Delta y$ (mm)	$\Delta z$ (mm)	$GIM_n$	$GIm_n$
0.068	0.069	7.551 e-003	2.6422e + 008	3.2462e + 007

One can compute the stiffness matrix  $K$  at any other configuration by setting different values of the input angle  $\alpha$ . A proper “for” loop can be implemented in any programming environment (such as Matlab) to span the whole operation range of the milli-gripper. The same code can be used to compute any local and global stiffness performance index as well as the compliant displacements at a given configuration for a given acting external wrench  $\mathbf{W}_{\text{Given}}$ .

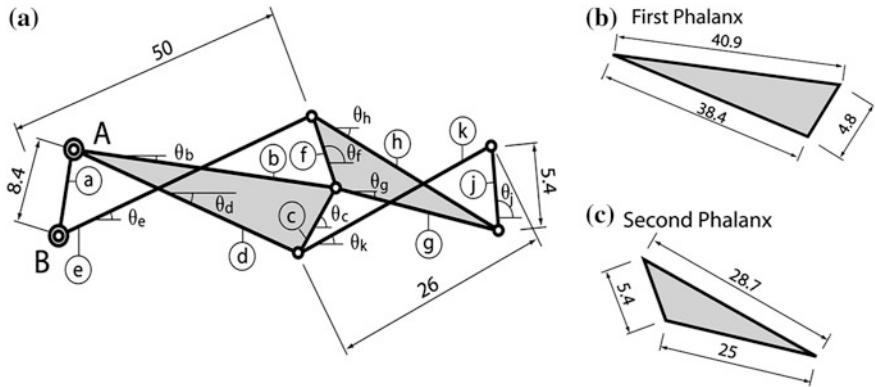
For example, if one assumes  $\mathbf{W}_{\text{Given}} = [1, 1, 0]^t$  then the compliant displacements versus the input angle  $\alpha$  can be computed as shown in Fig. 2.15. The compliant displacements in Fig. 2.15 can be seen as local indices of stiffness performance. Similarly, the determinant of the stiffness performance can give an useful graphical information of the local stiffness performance as it grows when stiffness performance is improving. Global stiffness indices can be computed such as proposed in Eqs. (2.9) and (2.10) for the above-mentioned case of study as reported in Table 2.5.

#### 2.4.4 LARM Hand IV

LARM Hand IV, Fig. 2.2b, is a robotic hand having three one-DOF human-like fingers that has been developed and built at LARM in Cassino. Figure 2.16 shows a scheme of a finger including a kinematic model of its driving mechanism. Each finger is basically composed of two four-bar linkage mechanisms as shown in Fig. 2.16. The first phalanx is the input bar of the first four-bar linkage mechanism. It is also the base frame of the second four-bar linkage mechanism. The second phalanx is the input bar of the second four-bar linkage mechanism and it is also the coupler of the first four-bar linkage mechanism. Then, the third phalanx is the coupler of the second four-linkage mechanism. Table 2.6 shows the main design parameters of a finger.

Referring to the scheme in Fig. 2.16, the angular velocities of the second and third phalanxes can be defined as  $\dot{\theta}_g = d\theta_g/dt$  and  $\dot{\theta}_j = d\theta_j/dt$ , respectively. Both  $\dot{\theta}_g$  and  $\dot{\theta}_j$  can be computed as function of the input angular velocity of the first phalanx  $\dot{\theta}_b = d\theta_b/dt$  in the form

$$\dot{\theta}_f = \dot{\theta}_g = \frac{b \sin(\theta_b - \theta_e)}{f \sin(\theta_e - \theta_f)} \dot{\theta}_b \quad (2.25)$$



**Fig. 2.16** Scheme of the LARM hand driving mechanism: **a** Complete system; **b** First phalanx; **c** Second phalanx

**Table 2.6** Main design parameters of a finger of LARM Hand IV

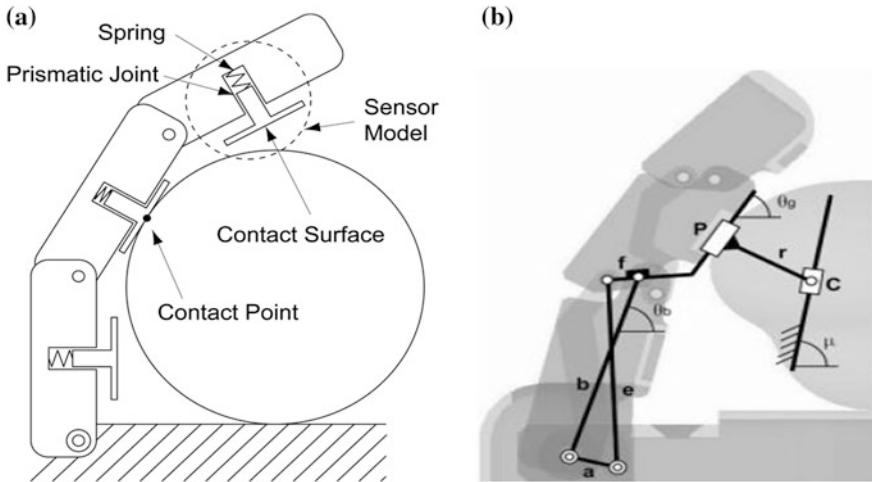
Label in Fig. 2.2	Frame	Phalanx 1				Phalanx 2				Phalanx 3	
		b	c	d	e	f	g	h	k	j	
Length (mm)	a	8.4	40.9	4.8	38.4	50	5.4	25	28.7	26	5.4

$$\dot{\theta}_j = \dot{\theta}_c + \frac{g \sin(\theta_g - \theta_k)}{j \sin(\theta_k - \theta_j)} (\dot{\theta}_f - \dot{\theta}_c) \quad (2.26)$$

where one can also replace the angular velocity  $\dot{\theta}_c$  with  $\dot{\theta}_b$ , since both  $\dot{\theta}_c$  and  $\dot{\theta}_b$  are angular velocities of the same rigid body (the first phalanx) with respect to the fixed reference frame. The time derivatives of Eqs. (2.25) and (2.26) can be also used for computing the angular accelerations on each phalanx. The above-mentioned kinematic equations can be used to derive the speed of each phalanx as function of the input speed or vice versa. Therefore, one can use Eqs. (2.25) and (2.26) for verifying that the speed of a phalanx provides a human-like behavior.

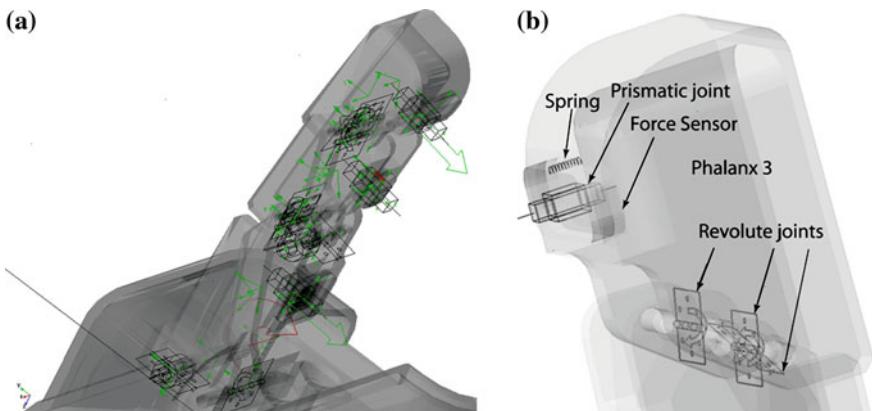
The LARM Hand IV is also equipped with three force sensors on each finger for measuring the grasping force on each phalanx. The location of these force sensors is shown in the scheme of Fig. 2.17a. It is worth noting that in the model of Fig. 2.3 each sensor has been modeled as a prismatic joint with a spring. The lumped stiffness parameter of each spring has been assumed as  $k = 10 \text{ N/mm}$  by referring to a piezoresistive low-cost force sensor. Thus, the grasping force that is measured on each phalanx can be computed  $F = k\Delta d$  where  $\Delta d$  is the compliant displacement on each spring.

Figure 2.17b shows a scheme of LARM hand grasping a cylindrical object. In this scheme, the object is in contact with the second phalanx of LARM Hand at point P. The same grasping conditions of Fig. 2.17 have been modeled in

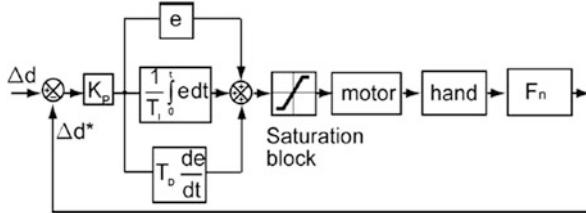


**Fig. 2.17** A finger of LARM Hand IV: **a** Model of the grasp and contact; **b** A scheme of the kinematic chain for the numerical solution in MSC.ADAMS environment

MSC.ADAMS environment as shown in the scheme of Fig. 2.18. The proposed model in Fig. 2.18 has been carefully designed and implemented in order to obtain a suitable numerical solution of the proposed grasping conditions. It is worth noting that a firm grasp is achieved when all forces are in equilibrium. Therefore, the input torque has to change as function of several parameters including the external force acting on the object, and position, size, shape of the grasped object. Numerical simulations of LARM Hand operation have been computed in MSC.ADAMS environment.



**Fig. 2.18** A MSC.ADAMS mode of LARM Hand IV: **a** A finger with main constraints; **b** Detail of the third phalanx



**Fig. 2.19** Control PID with saturation block for the contact force control of the finger

The contact of the elastic bar with the frame and rollers have been modeled with the characteristic of the contact data as: stiffness: 1.0E11 Pa; damping: 10.0E3 Ns/m; force exponent: 2.2 N; maximum penetration: 0.1 mm. The friction has been modeled as a Coulomb force with the followings parameters: Static coefficient 0.6; Dynamic coefficient 0.5; Stiction Transition Velocity 100 mm/s; Friction Transition Velocity 1000 mm/s.

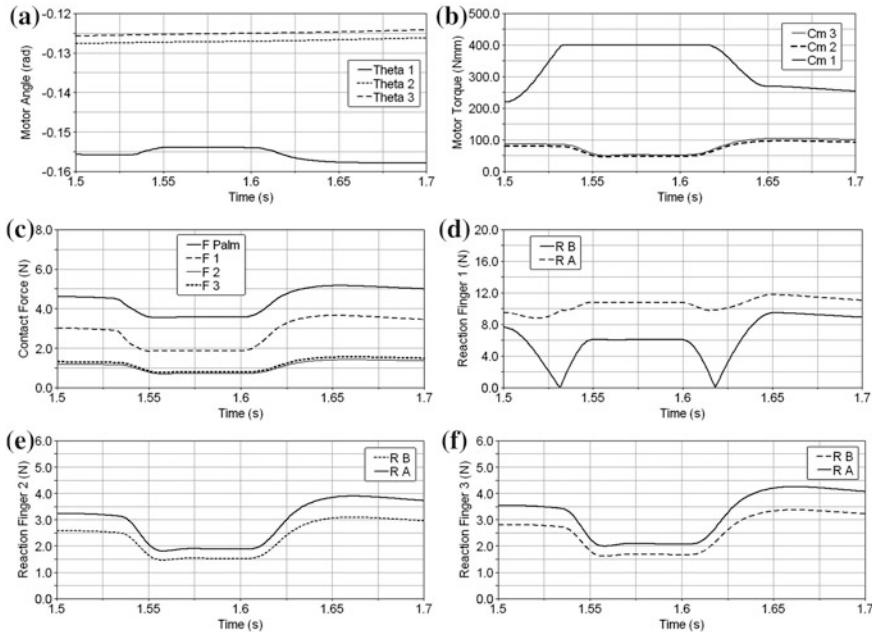
The operation mode is based on a control of the torque of the motor aiming to obtain a desired maximum value for the contact force. A close loop control is achieved by using as feedback the data that are obtained from force sensors on the finger. The force sensors are modeled as a linear spring as it is shown in Figs. 2.17a and 2.18b. The value of the contact force is obtained by evaluating the displacement of the prismatic joint of the force sensor. Figure 2.19 shows the scheme of the PID control with the input data that are obtained from the force sensor.

A saturation block has been added, as it is shown in Fig. 2.19. This block imposes upper and lower bounds on a signal. When the input signal is within the range specified by the lower limit and upper limit parameters, the input signal passes through unchanged. When the input signal is outside these bounds, the signal is set to the upper or lower bound. The system uses the measures that are provided by the force sensors to trigger the change from one control scheme to another. This trigger has been modeled with an “if” statements.

If the absolute value of the deformation of the spring that models a force sensor is lower than a threshold value the system remains in the speed control, but if the deformation is greater than this value the control switches to the contact force control. A hysteretic gap has been added to avoid an oscillation from one control to another in the transient phase between the two operation modes.

Several simulations have been carried out in MSC.ADAMS environment by implementing the proposed grasp model and control algorithm. Figures 2.20 and 2.21 show results of the simulation of LARM Hand IV while grasping a rigid cylinder of 50 mm of diameter, under various grasping conditions. The control has been designed to obtain a normal contact force of 1.5 N for the two parallel fingers and of 3 N for the opposite finger. A friction coefficient has been assumed at the contacts with low values, as in usual robotic devices.

The integrator is an algorithm that solves the differential equations of a dynamic problem over an interval of time during a simulation. The used integrator for the

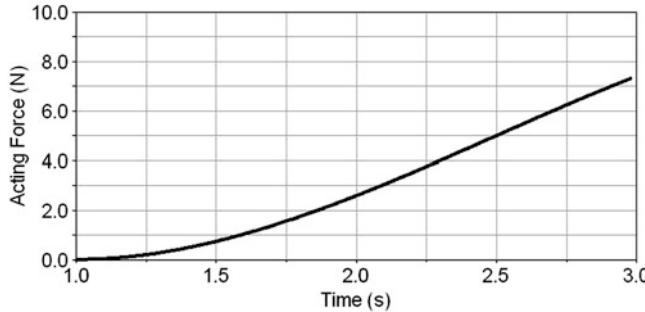


**Fig. 2.20** Simulation results of the LARM Hand grasp in Fig. 2.3 when an external disturbing force is applied to the object: **a** Input angle; **b** Motor torque; **c** Contact grasp forces; **d** Reaction forces on finger 1; **e** Reaction forces on finger 2; **f** Reaction forces on finger 3

simulation has been gear stiff integrator (GSTIFF), [60]. The GSTIFF integrator is the default in MSC.ADAMS environment since it provides good solutions for simulation of stiff models. The GSTIFF integrator uses a backwards differentiation formula to integrate differential and algebraic equations. In addition, it assumes a fixed time step that results in fixed coefficients for predicting the errors. The time interval 0.1 (10 intervals in 1 s) has been selected after several attempts for developing the simulation of the LARM Hand in order to obtain suitable numerical results with suitable computational efforts. The simulation time has been 20 s in a standard Pentium II computer.

The speed control is used from the initial position until the spring suffer a compliant displacement of  $\Delta d = 0.1$  mm. After this event occurs, the system switches to the contact force control. Results also show the transition from speed control to the contact force control. The PID constant  $K_p = 1E4$ ,  $K_i = 1E5$  and  $K_d = 100$  has been chosen in order to prevent upper-oscillation around the value of the maximum force value.

Results in Figs. 2.20 and 2.21 show that the control is suitable for keeping a firm grasp. In particular, Fig. 2.20a shows the values for the angles of the input bars of the finger. These angles show that the finger does not return to its original configuration with input angle at  $-0.155$  rad. However, the new input angle value of  $0.153$  rad is still a stable configuration. Figure 2.20b shows the motor torques that grow to their



**Fig. 2.21** Module of the external force that is applied to the centre of mass of the object that is grasped for the simulation, whose results are reported in Fig. 2.13

maximum value of 400 Nm in order to return to a new stable condition after about 0.15 s. The contact forces are shown in Fig. 2.20c. The control cannot avoid that the contact forces goes below the desired values of 3 N due to a restriction in the maximum torque that the input motor can provide. However, contact forces never go over 1.8 N. This value can be considered suitable for keeping the grasp and avoiding any damage to the objects to be grasped. Figures 2.20d-f show the reaction forces in the frame joints of the finger. In particular, plots in Fig. 2.20d show reaction forces ranging from 0 to about 12 N with sufficiently smooth time history, and Fig. 2.20e and d show reaction forces in the range from 1.5 to 4 N.

## 2.5 Experimental Determination of Stiffness Performance

Experimental determination of stiffness performance of multibody robotic systems can be performed for calibration purposes and operation characterizations by identifying the entries of stiffness matrix of the proposed formulation according to practical aims that can be related also to standard codes that are reported in [51, 52]. Nevertheless, the computation of the coefficients of the stiffness matrix requires carrying out experimental tests in which compliant displacements and wrenches will be measured contemporaneously.

One should note that the stiffness matrix  $K$  can be symmetric if and only if some conditions are satisfied on the external wrench and choice of reference frames for the representation of compliant displacements as demonstrated for example in [26–28]. Therefore, the computation of the  $6 \times 6$  Cartesian stiffness matrix  $K$  in the most general case requires the identification of its overall  $36 k_{ij}$  entries. The identification of all these 36 entries can be achieved if wrenches and compliant displacements are measured in at least six experiments for a given manipulator in a given configuration. In fact, with six experiments Eq. (2.2) can be used to give as many equations as the 36 unknown entries of  $K$  in the form

$$\begin{pmatrix} {}^1\Delta x & {}^1\Delta y & {}^1\Delta z & {}^1\Delta \varphi & {}^1\Delta \psi & {}^1\Delta \theta & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & {}^1\Delta x & {}^1\Delta y & {}^1\Delta z & {}^1\Delta \varphi & {}^1\Delta \psi & {}^1\Delta \theta & k_{11} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ {}^6\Delta x & {}^6\Delta y & {}^6\Delta z & {}^6\Delta \varphi & {}^6\Delta \psi & {}^6\Delta \theta & \dots & 0 & 0 & 0 & 0 & 0 & 0 & k_{61} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & k_{62} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & k_{63} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & k_{64} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & k_{65} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & {}^6\Delta x & {}^6\Delta y & {}^6\Delta z & {}^6\Delta \varphi & {}^6\Delta \psi & {}^6\Delta \theta & k_{66} \end{pmatrix} \begin{pmatrix} k_{11} \\ k_{12} \\ k_{13} \\ k_{14} \\ k_{15} \\ k_{16} \\ \vdots \\ \vdots \\ \vdots \\ k_{61} \\ k_{62} \\ k_{63} \\ k_{64} \\ k_{65} \\ k_{66} \end{pmatrix} = \begin{pmatrix} {}^1F_x \\ {}^1F_y \\ {}^1F_z \\ {}^1N_x \\ {}^1N_y \\ {}^1N_z \\ \vdots \\ \vdots \\ \vdots \\ {}^6F_x \\ {}^6F_y \\ {}^6F_z \\ {}^6N_x \\ {}^6N_y \\ {}^6N_z \\ 0 \end{pmatrix} \quad (2.27)$$

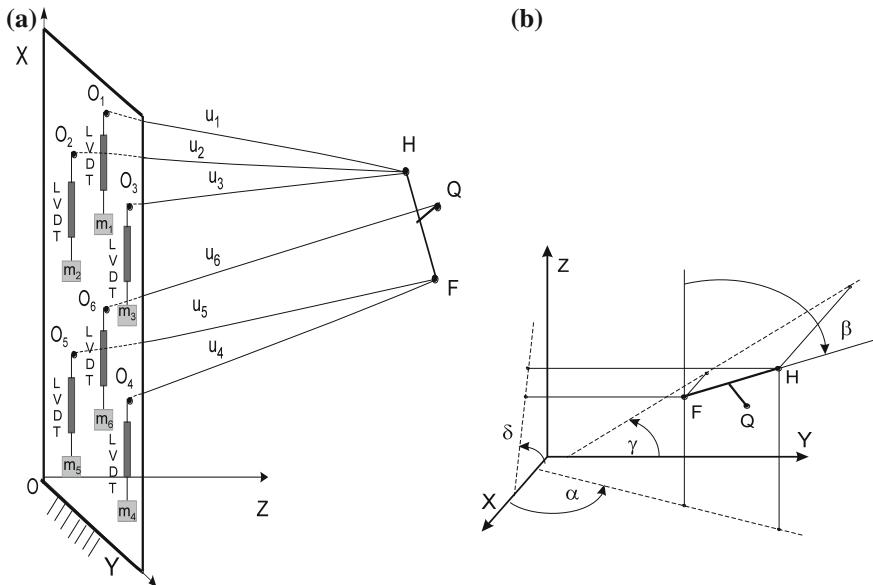
where the  $k_{ij}$  coefficients refer to the stiffness matrix as

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} \end{bmatrix} \quad (2.28)$$

The numerical solution of Eq. (2.28) provides the required values of the 36 coefficients of the stiffness matrix in Eq. (2.14) once the wrenches (due to known masses) and compliant displacements (due to those wrenches) that have been measured in six experiments are available for a given configuration. These experiments can be carried out by means of Milli-CaTraSys that has been conceived and built at LARM as schematized in Fig. 2.22.

Milli-CaTraSys is a wire tracking system whose scheme is reported in Fig. 2.23. It is composed of six LVDT sensors that can measure the distances  $u_i$  ( $i = 1, \dots, 6$ ). Then, the distances  $u_i$  are used as radii of arcs from corresponding center points  $O_i$ . The position of H on the robot end effector is defined as the position of the point at which three arcs intersect. Similarly, it is possible to measure the position of points F and Q. Then, the vectors connecting H, F, and Q can be used to determine the end-effector orientation through the orientation angles  $\alpha$ ,  $\gamma$  and  $\delta$ , as shown in Fig. 2.22b. Thus, a trilateration technique can be used with Milli-CaTraSys in order to measure both position and orientation of its end effector that can be attached to a mobile body through reference points H, F and Q on it, as shown in Fig. 2.23. Moreover, that known masses can be attached on the free end of each wire as shown in Fig. 2.22a. Therefore, Milli-CaTraSys can measure the changes in position and orientation of its end effector (compliant displacements) while different known wrenches are applied. Several experimental trials can be carried out for each configuration just by applying different masses to the wires.

A virtual instrument in LabVIEW environment has been developed for acquisition and processing the data from the LVDT sensors. This virtual instrument has



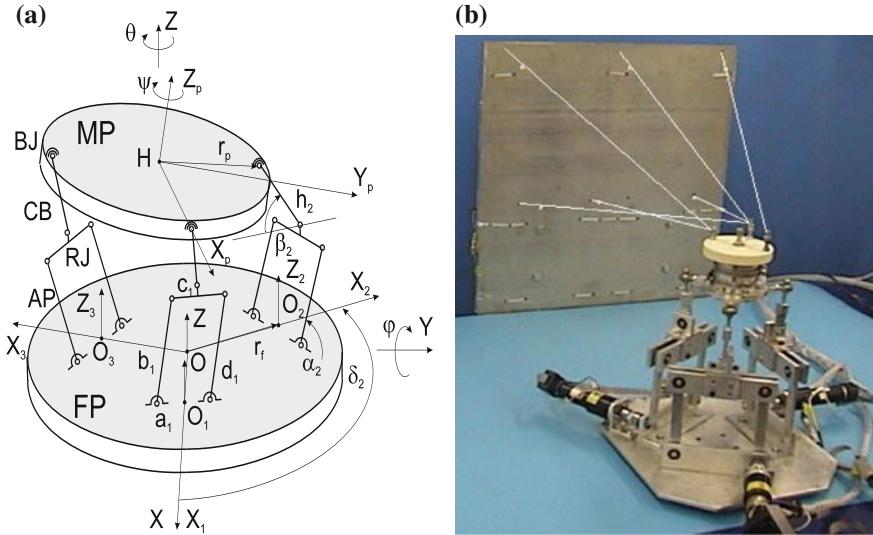
**Fig. 2.22** A scheme of Milli-CaTraSys: **a** With the reference frame, LVDTs and masses  $m_i$  as applied to the wires  $u_i$  ( $i = 1, \dots, 6$ ); **b** Orientation of the end-effector through the angles  $\alpha$ ,  $\gamma$  and  $\delta$

been used to measure the displacement of the cores inside the LVDT sensors. In this configuration of Milli-CaTraSys, the H point can be measured with any combination of three wires among the six available wires. In particular, the combination of three wires 1, 3, 5, and the combination of three wires 2, 4, 6 have been selected. In fact, these configurations are symmetric, have been proved to reduce computational costs, and increase the accuracy of measurement, as reported in previous experiences at LARM.

## 2.6 Cases of Study for Stiffness Experimental Tests

### 2.6.1 CaPaMan 2bis

CaPaMan 2bis is a parallel manipulator that has been designed and built at LARM in Cassino. A kinematic scheme of CaPaMan 2bis is shown in Fig. 2.23, where the fixed platform is FP and the moving platform is MP. MP is connected to FP through three identical leg mechanisms and is driven by the corresponding articulation points. An articulated parallelogram AP, a revolute joint RJ and a connecting bar CB compose each leg mechanism. AP's coupler carries the RJ and CB transmits the motion from AP to MP through RJ; CB is connected to the MP by a spherical joint BJ, which is installed on MP. Each plane, which contains AP, is



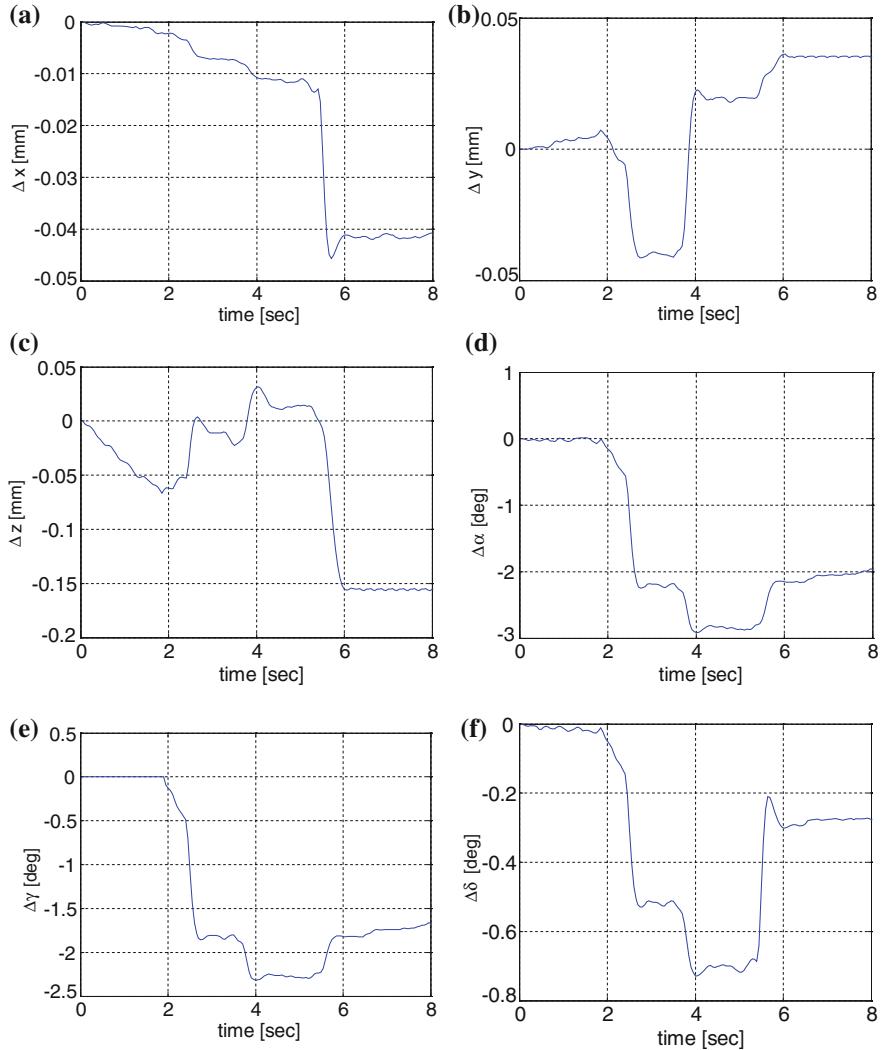
**Fig. 2.23** The CaPaMan 2bis: **a** A kinematic scheme; **b** A prototype with Milli-CaTraSys set up at LARM in Cassino

rotated of  $\pi/3$  with respect to the neighbour one. Design parameters of a  $k$ -th leg are identified through:  $a_k$ , which is the length of the frame link;  $b_k$ , which is the length of the input crank;  $c_k$ , which is the length of the coupler link;  $d_k$ , which is the length of the follower crank;  $h_k$ , which is the length of the connecting bar. The kinematic input variables are the crank angles  $\alpha_k$  ( $k = 1, 2, 3$ ). Sizes of MP and FP are given by  $r_p$  and  $r_f$ , respectively. Table 2.7 reports the sizes of main design parameters of CaPaMan 2bis.

Experimental tests have been carried out by applying six different wrenches for a given configuration of CaPaMan 2bis. In particular, Fig. 2.24 shows the measured compliant displacements when  $m_1 = m_2 = m_3 = m_4 = m_5 = m_6 = 0.03$  kg and CaPaMan 2bs is in its vertical configuration. Similar results have been obtained for six different masses distributions. It is worth noting that the plots of Fig. 2.24 show the measured compliant displacements versus time during a static experiment. This is necessary in order to find the stationary values of the measured compliant displacements after applying the external wrench. In particular, the stationary values for the experimental test that is reported in Fig. 2.24 are  $\Delta x = -0.041$ ,  $\Delta y = 0.035$ ,  $\Delta z = -0.155$ ,  $\Delta \alpha = -1.964^\circ$ ,  $\Delta \gamma = -1.667^\circ$  and  $\Delta \delta = -0.277^\circ$ .

**Table 2.7** Sizes of main design parameters for CaPaMan 2bis

$a_k = c_k$ (mm)	$b_k = d_k$ (mm)	$h_k$ (mm)	$r_p = r_f$ (mm)	$\alpha_k$ ( $^\circ$ )
100	100	50	65	45:135



**Fig. 2.24** Measured compliant displacements for a wrench given by  $m_1 = m_2 = m_3 = m_4 = m_5 = m_6 = 0.033$  kg when CaPaMan2bis is in its vertical configuration: **a**  $\Delta x$ ; **b**  $\Delta y$ ; **c**  $\Delta z$ ; **d**  $\Delta\alpha$ ; **e**  $\Delta\gamma$ ; **f**  $\Delta\delta$

The experimental analysis has given results, which confirm the numerical computations of the stiffness matrix. For example, when the three legs of CapaMan 2bis are inclined of  $45^\circ$  the stiffness matrix is measured as given by

$$K = 10^8 \begin{bmatrix} 0.013 & 0.002 & 0.040 & -10.13 & 0.031 & 3.058 \\ 0.003 & 0.001 & 0.010 & -2.595 & 0.012 & 0.782 \\ 0.000 & 0.000 & 0.001 & -0.093 & -0.013 & 0.028 \\ 0.000 & 0.000 & 0.000 & -0.027 & 0.000 & 0.008 \\ -0.000 & -0.000 & -0.001 & 0.195 & -0.001 & -0.059 \\ 0.000 & 0.000 & 0.000 & -0.051 & 0.000 & 0.015 \end{bmatrix} \quad (2.29)$$

when wrenches are obtained by using additional masses of 50 grams on each wire once in tension.

When two legs are inclined of  $60^\circ$  and one leg is in vertical configuration, the stiffness matrix is measured as

$$K = 10^5 \begin{bmatrix} 0.014 & -0.094 & 0.268 & 7.416 & 4.410 & -2.051 \\ 0.009 & -0.174 & 0.606 & 1.740 & 13.99 & -5.299 \\ 0.008 & -0.138 & 0.500 & 20.09 & 7.603 & -5.923 \\ -0.000 & 0.000 & -0.000 & 0.003 & 0.007 & -0.000 \\ -0.000 & 0.000 & -0.001 & 0.169 & -0.026 & -0.050 \\ -0.000 & 0.000 & -0.000 & -0.307 & -0.126 & 0.092 \end{bmatrix} \quad (2.30)$$

The determinant of the stiffness matrix in Eq. (2.29) is equal to  $5.987 \times 10^{20}$  while the determinant of the stiffness matrix in Eq. (2.30) is equal to  $-2.452 \times 10^{20}$ . By using this measure one can conclude that CaPaMan 2bis is stiffer in the first configuration, as intuitively expected.

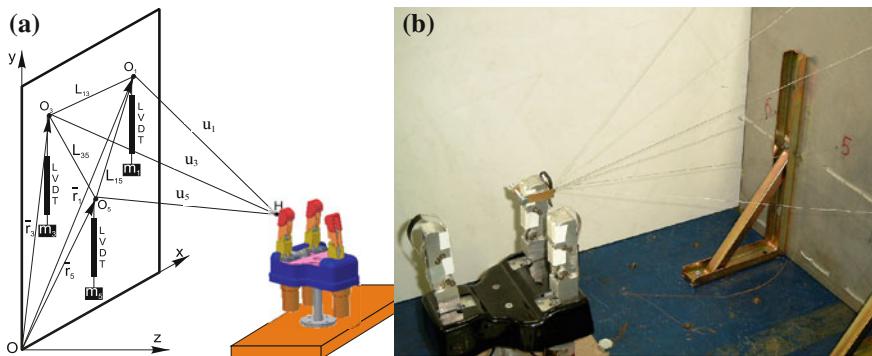
### 2.6.1.1 LARM Hand IV

The attached problem is to determine the performance of the LARM Hand in terms of operation properties. This has been obtained by using the system Milli-CaTraSys. A proper end effector has been installed on a finger of the LARM Hand as shown in Fig. 2.25 and Fig. 2.26 in order to provide a proper location of points F, Q, and H and suitable frame for attaching the wires. In particular, three wires have been attached to point H, two wires to point F, and one wire to point Q. This setup refers to the Gough-Stewart 3-2-1 parallel manipulator configuration. Experiments with two redundant wires have been also carried out to have a 3-3-3 configuration and to validate the use of the 3-2-1 configuration with redundant measurement data.

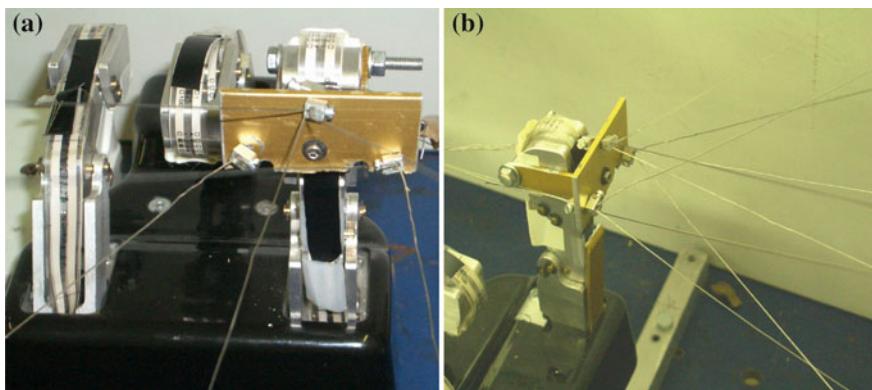
A calibration process has been carried out on the experimental setup, as proposed for example in [24–27], in order to determine a suitable initial configuration for the measuring system and to verify its accuracy.

As a result of the experimental calibration process, it has been determined an average accuracy of Milli-CaTraSys of about 0.1 mm when LVDT sensors are used with 100 mm range and an accuracy of Milli-CaTraSys of about 0.01 mm when LVDT sensors are used with 2.5 mm range.

Experimental tests have been carried out by means of the setup that is shown in Fig. 2.25. In the experimental tests, pretension of all wires has been obtained by



**Fig. 2.25** The experimental set-up for the test-bed operations with LARM Hand: **a** A scheme; **b** The laboratory set-up

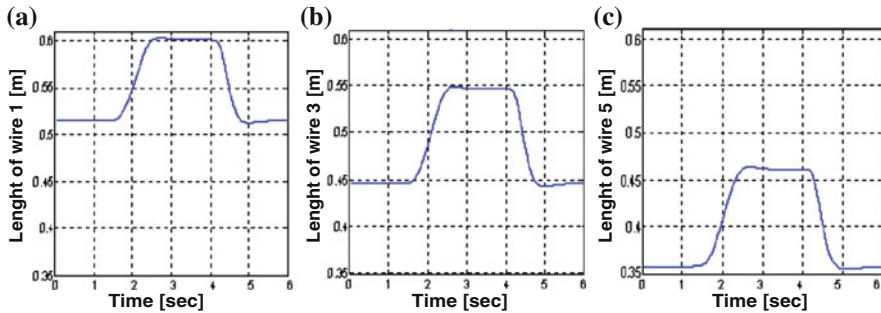


**Fig. 2.26** End-effector for Milli-CaTraSys that has been attached to the fingertip of LARM Hand: **a** A zoomed frontal view with connections for six wires in a 3-2-1 platform configuration; **b** A zoomed lateral view with connections for six wires in a 3-3-3 platform configuration

means of known masses  $m_i$  ( $i = 1, \dots, 6$ ) of 30 g. This pretension value has been set up experimentally since it keeps all the wires pulling during the whole duration of experimental tests. Moreover, this pretension produces negligible compliant displacements of the twisted iron wires that have been used in Milli-CaTraSys.

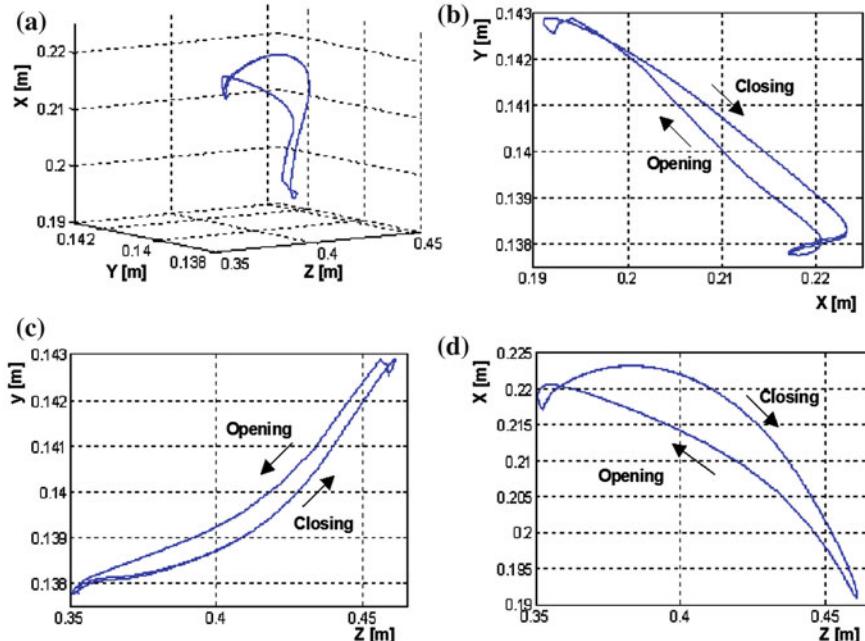
Experimental tests have been carried out by operating the LARM Hand in an open-close mode. Namely, the fingers of LARM Hand begin the tests in the fully open configuration; they start moving after about 1.6 s; they move to the fully closed configuration in about 1 s; they wait in the fully closed configuration about 2 s; they move back to the fully open configuration in about 1 s.

For example, Fig. 2.27 shows the plots of the measured lengths of the wires 1-3-5 during an experimental test. Then, the measured lengths of wires are converted through trilateration technique to give the position of the fingertip of LARM Hand.



**Fig. 2.27** Plots of the measured lengths of the wires of Milli-CaTraSys during an open-close operation mode for LARM Hand: **a** Wire 1; **b** Wire 3; **c** Wire 5

The position coordinates of the fingertip of LARM Hand during the experimental test in Fig. 2.27 are reported in Fig. 2.28 as function of time. In particular, Fig. 2.28a shows a 3D view of the operation workspace of a fingertip motion of LARM Hand in Cartesian coordinates. Figure 2.28b–d show the projection of the operation workspace of a fingertip motion of LARM Hand onto XY, YZ, and XZ planes, respectively. The measured motion ranges have been about 0.005 in X direction, 0.032 in Y direction, 0.113 m in Z direction, respectively.



**Fig. 2.28** Workspace of a fingertip motion of LARM Hand during an open-close operation mode: **a** A 3D view; **b** Projection onto XY plane; **c** Projection onto YZ plane; **d** Projection onto XZ plane

One should note that a nonzero value of the motion range in X direction is an evidence of a slightly nonplanar motion of the finger mechanism. This motion can be although as given by a certain clearance in the joints. Moreover, due to the presence of this clearance, the motion in the closing phase does not coincide with the motion in the opening phase as shown in Fig. 2.28. However, the above-mentioned motion ranges show a good match with those referring to a cylindrical grasping mode by an average human hand.

Additional experimental tests with the same set up have been carried in static conditions by applying known wrenches on the LARM Hand fingertip. Maximum compliant displacement that have been measured on the fingertip of LARM Hand has been of 0.1 mm for a maximum force of 3 N in the same direction. Thus, a stiffness coefficient can be computed as at least 30,000 N/m. The above-mentioned values of maximum compliant displacement and stiffness coefficient give a proof of a quite good stiffness behavior for LARM Hand operation, whose average grasping force is 3 N. A more accurate estimation of stiffness behavior can be obtained by computing a  $6 \times 6$  Cartesian stiffness matrices for LARM Hand through a general procedure that has been proposed in [28].

## 2.7 Conclusions

This section has given fundamentals of stiffness modeling and analysis by referring to robotic systems for grasping tasks. A formulation has been detailed for the computation of the Cartesian stiffness matrix  $K$ . Considerations on local and global stiffness properties have been reported also within a numerical procedure. Cases of study have been proposed for the stiffness analysis of manipulators having serial or parallel architecture. Stiffness modeling and analysis has been reported also for mechanical end-effectors such as two-finger grippers or robotic hands, given their significance in grasping tasks. A procedure for experimental validation has been outlined and two cases of study have been reported.

## References

1. Riven EI (1999) Stiffness and damping in mechanical design. Marcel Dekker Inc., New York
2. Ceccarelli M (2004) Fundamentals of mechanics of robotic manipulation. Kluwer, Dordrecht
3. Tsai LW (1999) Robot analysis: the mechanics of serial and parallel manipulators. Wiley, New York, pp 260–297
4. Duffy J (1996) Statics and kinematics with applications to robotics. Cambridge University Press, Cambridge, pp 153–169
5. Nof SY (ed) (1985) Handbook of industrial robotics. Wiley, New York
6. Merlet J-P (2006) Parallel robots. Springer, Dordrecht
7. Carbone G (2003) Stiffness evaluation of multibody robotic systems. Ph D Dissertation, LARM, University of Cassino, Cassino
8. Gosselin C (1990) Stiffness mapping for parallel manipulators. IEEE Trans Robot Autom 6(3):377–382

9. Tahmasebi F, Tsai LW (1992) Jacobian and stiffness analysis of a novel class of six-dof parallel manipulators. In: Proceedings of the ASME 22nd biennial mechanism conference, Scottsdale, vol 47, pp 95–102
10. Gosselin CM, Zhang D (2002) Stiffness analysis of parallel mechanisms using a lumped model. *Int J Robot Autom* 17(1):17–27
11. Simaan N, Shoham M (2003) Stiffness synthesis of a variable geometry six-degrees-of-freedom double planar parallel robot. *Int J Robot Res* 22(9):757–775
12. Kim HY, Streit DA (1995) Configuration dependent stiffness of the Puma 560 manipulator: analytical and experimental results. *Mech Mach Theory* 30(8):1269–1277
13. Ceccarelli M, Carbone G (2005) Numerical and experimental analysis of the stiffness performance of parallel manipulators. In: 2nd international colloquium collaborative research centre 562, Braunschweig, pp 21–35
14. Chakarov D (2001) Analysis and synthesis of the stiffness of a hybrid manipulator with redundant actuation. In: Proceedings of the 5th Magdeburg days of mechanical engineering, Magdeburg, pp 119–127
15. Ciblak N, Lipkin H (1999) Synthesis of cartesian stiffness for robotic applications. In: Proceedings of the IEEE international conference on robotics and automation ICRA'99, Detroit, vol 3, pp 2147–2152
16. Pigoski T, Griffis M, Duffy J (1998) Stiffness mappings employing different frames of reference. *Mech Mach Theory* 33(6):825–8381998
17. Carbone G, Ceccarelli M (2004) A stiffness analysis for a hybrid parallel-serial manipulator. *Robot Int J* 22:567–576
18. Ceccarelli M, Carbone G (2002) A stiffness analysis for CaPaMan (Cassino parallel manipulator). *Mech Mach Theory* 37(5):427–439
19. Svinin MM, Hosoe S, Uchiyama M, Luo ZW (2002) On the stiffness and stiffness control of redundant manipulators. In: IEEE international conference on robotics & automation ICRA 2002, Washington, pp 2393–2399
20. Cutkosky MR, Kao I (1989) Computing and controlling the compliance of a robotic hand. *IEEE Trans Robot Autom* 5(2):151–165
21. Tsumugiwa T, Yokogawa R, Hara K (2002) Variable impedance control with virtual stiffness for human-robot cooperative peg-in-hole task. In: Proceedings of the IEEE/RSJ international conference on intelligent robots ans systems IROS'02, Lausanne, pp 1075–1081
22. Chakarov D (1998) Optimization synthesis of parallel manipulators with desired stiffness. *J Theor Appl Mech* 28(4). Available. <http://www.imbm.bas.bg/IMBM/LMS/Chakarov/TPM98.pdf>
23. Liu X-J, Jin Z-L, Gao F (2000) Optimum design of 3-dof spherical parallel manipulators with respect to the conditioning and stiffness indices. *Mech Mach Theory* 35(9):1257–12672000
24. Carbone G, Ottaviano E, Ceccarelli M (2007) An optimum design procedure for both serial and parallel manipulators. *IMechE Part C J Mech Eng Sci* 221(7):829–843
25. Huang S, Schimmels JM (2000) The bounds and realization of spatial compliances achieved with simple springs connected in parallel. *IEEE Trans Robot Autom* 14(3):466–475
26. Zefran M, Kumar V (1997) Affine connections for the cartesian stiffness matrix. In: Proceedings of the IEEE international conference on robotics and automation ICRA'97, Albuquerque, vol 2, pp 1376–1381
27. Howard S, Zefran M, Kumar V (1998) On the  $6 \times 6$  Cartesian stiffness matrix for three-dimensional motions. *Mech Mach Theory* 33(4):389–408
28. Chen S-F, Kao I (2000) Geometrical method for modeling of asymmetric  $6 \times 6$  cartesian stiffness matrix. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems IROS2000, Takamatsu, pp 1217–1222
29. Pham DT, Heginbotham WB (1986) Robot grippers. IFS Publications Ltd, Bedford
30. Rosheim ME (1996) Robot surrogate: work in progress. In: International conference on robotics and automation, Minnesota, pp 399–403
31. Cutkosky MR (1989) On grasp choice, grasp model, and the design of hands for manufacturing tasks. *IEEE Trans Robot Autom* 5(3):269–279

32. Iberal T (1997) Human prehension and dexterous robot hands. *Int J Robot Res* 6(3):285–299
33. Fukaya N, Toyama S, Asfour T, Dillmann R (2000) Design of the TUAT/Karlsruhe humanoid hand. In: IEEE/RSJ international conference on intelligent robots and systems, Takamatsu, vol 3, pp 1754–1759
34. Butterfass J, Grebenstein M, Liu H, Hirzinger G (2001) DLR-Hand II: next generation of a dexterous robot hand. In: IEEE international conference on robotics and automation, Seoul Korea, pp 109–114
35. Zhang Y, Han Z, Zhan H, Shang X, Wang T, Guo W (2001) Design and control of the BUAA four-fingered hand. *IEEE Trans Robot Autom* 3:2517–2522
36. Gosselin CM, Mountambault S, Gosselin CJ (1993) Manus Colobi: preliminary results on the design of a mechanical hand for industrial applications. In: 19th ASME design automation, Albuquerque, vol 65–1, pp 585–592
37. Dechev N, Cleghorn WL, Nauman S (1999) Multiple finger, passive adaptive grasp prosthetic hand. *Mech Mach Theory* 36:1157–1173
38. Townsend WT (2000) The Barretthand grasper—programmably flexible parts handling and assembly. *Ind Robot Int J* 27(3):181–188
39. Venkataraman ST, Iberall T (eds) (1989) Dexterous robot hands. Springer, New York
40. Boudreault E, Gosselin C (2006) Design of sub-centimetre underactuated compliant gripper. In: Proceedings of IDETC/CIE 2006 ASME 2006, 30th mechanisms & robotics conference, Philadelphia. Paper DETC2006-99415
41. Ceccarelli M, Luyckx I, Vanaelten W (1996) Grasp forces in two-finger grippers: modeling and measuring. In: 5th international workshop on robotics in Alpe-Andria Danube region RAAD'96, Budapest, pp 321–326
42. Ceccarelli M, Nava Rodriguez NE, Carbone G (2006) Design and tests of a three-finger hand with 1-dof articulated fingers. *Robot Int J* 24(2):183–196
43. Carbone G, González A (2011) Numerical simulation of the grasp operation by LARM hand IV, A three finger robotic hand. *Robot Comput Integr Manuf* 27(2):450–459
44. Carbone G, Jeckel M, Havlík S, Ceccarelli M (2003) An optimum multi-objective design procedure for microgripping mechanisms. In: 12th international workshop on robotics in Alpe-Andria-Danube region RAAD 2003, Cassino, paper 055RAAD03
45. Penisi OH, Carbone G, Ceccarelli M (2002) Optimum design and testing of mechanisms for two-finger grippers. *Int J Mech Control* 03(01):9–20
46. Ceccarelli M, Carbone G, Kerle H (2001) Designing mechanisms for two-finger microgrippers. In: CD proceedings of the 10th workshop on robotics and Alpe-Adria-Danube region RAAD'01, Wien, paper RD-021
47. Carbone G, Ceccarelli M, Kerle H, Raatz A (2001) Design and experimental validation of a microgripper. *Fuji Int J Robot Mechatron* 13(3):319–325
48. Ceccarelli M, Carbone G (2010) Design and operation of fingered hands and two-finger grippers for space applications as from experiences at LARM. In: IX international scientific-technical conference. Control vibration technologies and machines VIBRATION 2010, Kursk, pp 208–216 (in Russian) ISBN 978-5-7681-0561
49. Lanni C, Carbone G, Havlík Š, Ceccarelli M (2007) Experimental validation of a milli-gripper based on Chebyshev mechanism. In: Proceedings of the 16th international workshop on robotics in Alpe-Adria-Danube region, Ljubljana, CD Proceedings, paper n. FP\_GG2, pp 42–51
50. Vaishnav RN, Magrab EB (1987) A general procedure to evaluate robot positioning errors. *Int J Robot Res* 6(1):59–74
51. ANSI, American National Standards Institute (1990) American national standard for industrial robots and robot systems: point-to-point and static performance characteristics—evaluation. ANSI/RIA 15.05-1-1990, New York
52. UNI, Italian National Institute for Standards (1995) Manipulating industrial robots: performance criteria and related test methods. UNI EN 29283 (= ISO 9283), Milan

53. Carbone G, Ceccarelli M (2004) A procedure for experimental evaluation of Cartesian stiffness matrix. In: 15th CISIM-IFToMM symposium on robot design, dynamics and control, paper Rom04-24, Montreal
54. Carbone G, Ceccarelli M (2010) A comparison of indices for stiffness performance evaluation. *Front Mech Eng* 5(3):270–278. doi:[10.1007/s11465-010-0023-z](https://doi.org/10.1007/s11465-010-0023-z)
55. Pratt GA, Williamson MM, Dillworth P, Pratt J, Ulland K, Wright A (1995) Stiffness isn't everything. In: Preprints of the 4th international symposium on experimental robotics ISER'95, Stanford, available. [http://www.ai.mit.edu/projects/leglab/publications/stiffness\\_isnt\\_everything.pdf](http://www.ai.mit.edu/projects/leglab/publications/stiffness_isnt_everything.pdf)
56. Schimmels JM (2001) Multidirectional compliance and constraint for improved robotic deburring. Part 1: improved positioning. *Robot Comput Integr Manuf* 17(4):277–286
57. English CE, Russell D (1999) Mechanics and stiffness limitations of a variable stiffness actuator for use in prosthetic limbs. *Mech Mach Theory* 34(1):7–25
58. Alici G, Shirinzadeh B (2003) Exact stiffness analysis and mapping for a 3-SPS + S parallel manipulator. In: 7th international conference on automation technology AUTOMATION 2003, Taiwan, paper F120
59. Zhou Y, Nelson BJ (1998) Adhesion force modeling and measurement for micromanipulation, microrobotics and micromanipulation. In: Sulzmann A, Nelson BJ (eds) International society for optical engineering, vol 3519. SPIE, Boston, pp 169–180
60. MSC.ADAMS (2010) Documentation and Help. User CD-ROM

# Chapter 3

## Multibody Dynamics Formulation

Parviz E. Nikravesh

**Abstract** In this chapter, a methodology for automatic generation of the equations of motion for rigid multibody systems is reviewed. The methodology is based on two formulations: body-coordinate formulation that uses Newton–Euler equations; and joint-coordinate formulation that employs relative coordinates. A systematic transformation process from the body into the joint-coordinate formulation is described. This methodology provides an effective and efficient formulation for constructing the equations of motion for multibody systems, in general, and for robotic devices in particular. A simple example is provided to demonstrate the implication of the methodology. In the final section, a list of some available commercial and free multibody dynamic software is presented.

### 3.1 Introduction

Derivation of equations of motion for computational multibody dynamics has been the objective of many research activities in the last four decades. The scope of these activities has been broad. Some techniques allow us to generate the equations of motion in terms of a large set of differential–algebraic equations (DAEs). Other techniques yield a minimal set of ordinary differential equations of motion. Many other *in-between* approaches provide us with various alternatives. Each formulation has its own advantages and disadvantages depending on the application and our priorities.

---

P. E. Nikravesh (✉)

Department of Aerospace and Mechanical Engineering, University of Arizona,  
Tucson, AZ 85721, USA  
e-mail: pen@email.arizona.edu

Although it is possible to construct the equations of motion for any multibody system using a classical approach such as the Lagrangian method, implementation of the process into a computer program can be time consuming, complicated, as well as computationally inefficient. Therefore, for multibody systems with large number of components, it is desirable to have a process for constructing the equations of motion that would not require too much programming effort, it is simple, and preferably computationally efficient.

Possibly, the simplest method to construct the equations of motion is in the form of a large set of DAE. The configuration of a rigid body is described by a set of translational and rotational coordinates. Algebraic constraints are introduced to represent kinematic joints connecting bodies, and the Lagrange multiplier technique is used to describe joint reaction forces. Although such formulations are easy to implement into a computer program, their main drawback is their computational inefficiency. One such formulation is called the body-coordinate formulation, which is the first method discussed in this chapter. This formulation has also been referred in the literature as the absolute- or Cartesian-coordinate formulation [1].

The body-coordinate equations of motion can be transformed into other generalized coordinates [2,3]. In this chapter, a systematic process for such transformation based on a simple velocity relationship is reviewed. The velocity relationship describes the body-coordinate velocities in terms of a new set of generalized velocities. If the new set of generalized velocities contains a smaller number of velocities than the original set, then the process yields a smaller number of equations. In this process, some of the kinematic constraints and their attributes vanish from the transformed equations. If the new set contains as many generalized velocities as the number of degrees of freedom (DOF), then the resultant equations will be in the form of second-order ordinary differential equations without any algebraic constraints. If the new set of generalized coordinates (velocities) is based on the DOF associated with the kinematic joints of the system, the method is called the joint-coordinate formulation [4–6].

The joint-coordinate formulation of equations of motion exhibits numerous advantageous features, in particular for robotic devices. Since the foundation of this formulation is the body coordinates, the absolute kinematics of each component is available at any given time. Furthermore, since the final equations are in terms of the joint coordinates, the relative kinematics between the bodies is also available at any given time. This dual feature provides the capability of incorporating models for control systems into the equations of motion. In addition, the formulation is computationally very efficient.

In this chapter, the body-coordinate formulation for general multibody systems is first reviewed. Then, the transformation process from the body-coordinate into the joint-coordinate formulation is presented. A simple example of a robot arm is provided to assist the reader in understanding the process of constructing the equations of motion. It should be noted that in this chapter matrix notation is used in order to keep our attention on concepts without losing sight of details. Readers should find the notation effective and simple to follow.

## 3.2 Body-Coordinate Formulation

Body-coordinate method, also known as absolute or Cartesian coordinate method, provides the simplest computational formulation to construct the equation of motion for multibody systems [1]. The formulation is simple to learn and easy to implement in either special- or general-purpose computer programs, however, the computational efficiency is not the best. This formulation yields a *large* set of *loosely coupled* DAEs of motion.

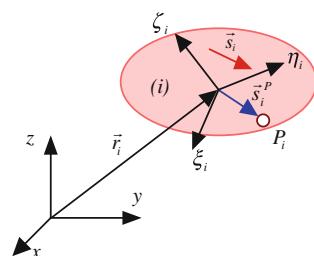
### 3.2.1 Configuration of a Rigid Body

As shown in Fig. 3.1, a typical rigid body  $i$  is positioned in an inertial (nonmoving)  $x - y - z$  frame by vector  $\vec{r}_i$ . This vector locates the origin of a body-attached  $\xi_i - \eta_i - \zeta_i$  frame, which for convenience is at the mass center of the body. This reference frame may or may not coincide with the body principal axes. The  $x - y - z$  components of vector  $\vec{r}_i$  represent the translational coordinates of the body and are denoted by the algebraic vector  $\mathbf{r}_i = \{x \ y \ z\}_i^T$ . Angular orientation of  $\xi_i - \eta_i - \zeta_i$  frame with respect to  $x - y - z$  frame can be described by a set of Euler angles or Euler parameters [1] (or any other desirable rotational coordinates). The rotational coordinates are used to construct a  $3 \times 3$  rotational transformation matrix, denoted as  $\mathbf{A}_i$ , where  $\mathbf{A}_i^T = \mathbf{A}_i^{-1}$ . This matrix transforms components of a vector from  $\xi_i - \eta_i - \zeta_i$  to  $x - y - z$ .

A vector or a point can be defined attached to a rigid body. As shown in Fig. 3.1, vector  $\vec{s}_i$  is defined on a body representing a particular axis, and vector  $\vec{s}_i^P$  locates point  $P$  from the origin of the body. Body-fixed components of these vectors are constants and are respectively denoted as  $\dot{\mathbf{s}}_i = \{s_{(\xi)} \ s_{(\eta)} \ s_{(\zeta)}\}_i^T$  and  $\dot{\mathbf{s}}_i^P = \{\xi^P \ \eta^P \ \zeta^P\}_i^T$ . Obviously, the following transformations can be performed:

$$\dot{\mathbf{s}}_i = \mathbf{A}_i \dot{\mathbf{s}}_i, \quad \dot{\mathbf{s}}_i^P = \mathbf{A}_i \dot{\mathbf{s}}_i^P \quad (3.1)$$

**Fig. 3.1** A rigid body positioned in an inertial frame



The translational and rotational velocity vectors for a body are defined as  $\dot{\mathbf{r}}_i = \{\dot{x} \quad \dot{y} \quad \dot{z}\}_i^T$  and  $\boldsymbol{\omega}_i = \{\omega_{(x)} \quad \omega_{(y)} \quad \omega_{(z)}\}_i^T$  or  $\boldsymbol{\phi}_i = \{\phi_{(\xi)} \quad \phi_{(\eta)} \quad \phi_{(\zeta)}\}_i^T$ , where  $\boldsymbol{\omega}_i = \mathbf{A}_i \boldsymbol{\phi}_i$ . In order to simplify the discussion, we adopt the  $x - y - z$  components of angular velocity vector in our formulations. Similarly, the acceleration vectors are defined as  $\ddot{\mathbf{r}}_i = \{\ddot{x} \quad \ddot{y} \quad \ddot{z}\}_i^T$  and  $\dot{\boldsymbol{\omega}}_i = \{\dot{\omega}_{(x)} \quad \dot{\omega}_{(y)} \quad \dot{\omega}_{(z)}\}_i^T$ .

The first and second time derivatives of a body-fixed vector, either  $\mathbf{s}_i$  or  $\mathbf{s}_i^P$ , are computed as:

$$\begin{aligned}\dot{\mathbf{s}}_i &= \tilde{\boldsymbol{\omega}}_i \mathbf{s}_i = -\tilde{\mathbf{s}}_i \boldsymbol{\omega}_i \\ \ddot{\mathbf{s}}_i &= -\tilde{\mathbf{s}}_i \dot{\boldsymbol{\omega}}_i + \tilde{\boldsymbol{\omega}}_i \dot{\mathbf{s}}_i\end{aligned}\tag{3.2}$$

Position, velocity, and acceleration of a point  $P$  attached to body  $i$  are expressed as:

$$\begin{aligned}\mathbf{r}_i^P &= \mathbf{r}_i + \mathbf{s}_i^P \\ \dot{\mathbf{r}}_i^P &= \dot{\mathbf{r}}_i + \dot{\mathbf{s}}_i^P = \dot{\mathbf{r}}_i - \tilde{\mathbf{s}}_i^P \boldsymbol{\omega}_i \\ \ddot{\mathbf{r}}_i^P &= \ddot{\mathbf{r}}_i + \ddot{\mathbf{s}}_i^P = \ddot{\mathbf{r}}_i - \tilde{\mathbf{s}}_i^P \dot{\boldsymbol{\omega}}_i + \tilde{\boldsymbol{\omega}}_i \dot{\mathbf{s}}_i^P.\end{aligned}\tag{3.3}$$

### 3.2.2 Kinematic Joints

In a typical multibody system, kinematic joints appear between some of the bodies. A kinematic joint presents conditions on the motion of the two connected bodies. These conditions can be described by algebraic equations, denoted as  $\Phi$ , between coordinates of points and components of vectors defined on those bodies.

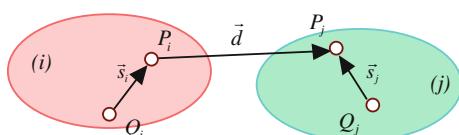
Assume vector  $\vec{s}_i$  is defined on body  $i$ , vector  $\vec{s}_j$  is defined on body  $j$ , and vector  $\vec{d}$  connects a point on body  $j$  to a point on body  $i$ , as shown in Fig. 3.2. Components of vector  $\vec{d}$  are computed as:

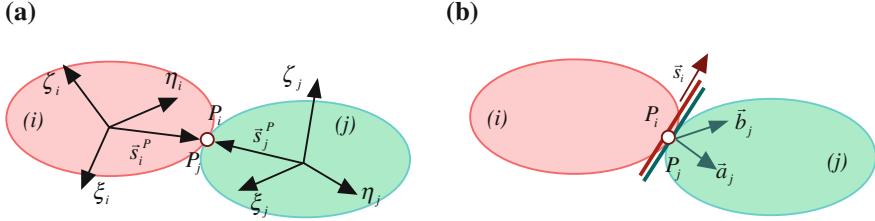
$$\mathbf{d} = \mathbf{r}_j^P - \mathbf{r}_i^P\tag{3.4}$$

For vectors  $\vec{s}_i$  and  $\vec{s}_j$  to remain perpendicular, we must enforce the following:

$$\Phi^{(n1)} \equiv \mathbf{s}_i^T \mathbf{s}_j = 0\tag{3.5}$$

**Fig. 3.2** Vectors defined on and between bodies





**Fig. 3.3** **a** A spherical joint and **b** a revolute joint

This constraint provides only one algebraic equation (*normal, type 1*). Similarly, for vectors  $\vec{s}_i$  and  $\vec{d}$  to remain perpendicular, we must enforce the following single algebraic equation (*normal, type 2*):

$$\Phi^{(n2)} \equiv \mathbf{s}_i^T \mathbf{d} = 0 \quad (3.6)$$

In order for two points to remain together; e.g.,  $P_i$  on body  $i$  and  $P_j$  on body  $j$  as shown in Fig. 3.3a, we must enforce the following three algebraic equations:

$$\Phi^{(s)} \equiv \mathbf{r}_j^P - \mathbf{r}_i^P = \mathbf{0} \quad (3.7)$$

This equation can be used to represent a *spherical* joint between two bodies.

Another useful constraint is for the length of a vector between two bodies to remain a constant (*spherical-spherical*):

$$\Phi^{(s-s)} \equiv \mathbf{d}^T \mathbf{d} - \ell^2 = 0 \quad (3.8)$$

By employing different combination of the constraints in Eqs. (3.5–3.8), we can represent a variety of kinematic joints. As an example, consider a revolute (pin) joint between two bodies as shown in Fig. 3.3b. We define point  $P_i$  on body  $i$  and point  $P_j$  on body  $j$  anywhere along the joint axis. These two points must remain together. We define vector  $\vec{s}_i$  along the joint axis on body  $i$ , and vectors  $\vec{a}_j$  and  $\vec{b}_j$  perpendicular to the joint axis (and to each other) on body  $j$ . Vector  $\vec{s}_i$  must remain perpendicular to  $\vec{a}_j$  and  $\vec{b}_j$ . Hence, the following five algebraic constraints must be enforced to form a revolute joint:

$$\Phi \equiv \begin{cases} \mathbf{r}_j^P - \mathbf{r}_i^P = \mathbf{0} \\ \mathbf{s}_i^T \mathbf{a}_j = 0 \\ \mathbf{s}_i^T \mathbf{b}_j = 0 \end{cases}$$

Similarly, other types of kinematic joints, such as cylindrical, prismatic, or universal, can be defined through constraint equations.

The first time derivative of a position constraint yields the corresponding velocity constraint. For our four fundamental constraints, the velocity constraints are:

**Table 3.1** Sub-Jacobians and the quadratic velocity terms for four fundamental constraints

Eq.	$\mathbf{D}_i$		$\mathbf{D}_j$		$\dot{\mathbf{D}}\mathbf{v}$
	$\dot{\mathbf{r}}_i$	$\boldsymbol{\omega}_i$	$\dot{\mathbf{r}}_j$	$\boldsymbol{\omega}_j$	
3.5	$\mathbf{0}$	$-\mathbf{s}_i^T \tilde{\mathbf{s}}_i$	$\mathbf{0}$	$-\mathbf{s}_i^T \tilde{\mathbf{s}}_j$	$\mathbf{s}_i^T \tilde{\boldsymbol{\omega}}_j \dot{\mathbf{s}}_j + \mathbf{s}_j^T \tilde{\boldsymbol{\omega}}_i \dot{\mathbf{s}}_i + 2 \dot{\mathbf{s}}_j^T \dot{\mathbf{s}}_i$
3.6	$-\mathbf{s}_i^T$	$\mathbf{s}_i^T (\tilde{\mathbf{d}} + \tilde{\mathbf{s}}_i^P)$	$\mathbf{s}_i^T$	$-\mathbf{s}_i^T \tilde{\mathbf{s}}_j^P$	$\mathbf{d}^T \tilde{\boldsymbol{\omega}}_i \dot{\mathbf{s}}_i + \mathbf{s}_i^T (\tilde{\boldsymbol{\omega}}_j \dot{\mathbf{s}}_j^P - \tilde{\boldsymbol{\omega}}_i \dot{\mathbf{s}}_i^P) + 2 \dot{\mathbf{d}}^T \dot{\mathbf{s}}_i$
3.7	$\mathbf{I}$	$-\tilde{\mathbf{s}}_i^P$	$-\mathbf{I}$	$\tilde{\mathbf{s}}_j^P$	$\mathbf{s}_j^P \boldsymbol{\omega}_j - \mathbf{s}_i^P \boldsymbol{\omega}_i$
3.8	$-2\mathbf{d}^T$	$2\mathbf{d}^T \tilde{\mathbf{s}}_i^P$	$2\mathbf{d}^T$	$-2\mathbf{d}^T \tilde{\mathbf{s}}_j^P$	$2\mathbf{d}^T (\tilde{\boldsymbol{\omega}}_j \dot{\mathbf{s}}_j^P - \tilde{\boldsymbol{\omega}}_i \dot{\mathbf{s}}_i^P) + 2 \dot{\mathbf{d}}^T \dot{\mathbf{d}}$

$$\begin{aligned}\dot{\Phi}^{(n1)} &\equiv -\mathbf{s}_j^T \tilde{\mathbf{s}}_i \boldsymbol{\omega}_i - \mathbf{s}_i^T \tilde{\mathbf{s}}_j \boldsymbol{\omega}_j = 0 \\ \dot{\Phi}^{(n2)} &\equiv -\mathbf{d}^T \tilde{\mathbf{s}}_i \boldsymbol{\omega}_i + \mathbf{s}_i^T (\dot{\mathbf{r}}_i - \tilde{\mathbf{s}}_j^P \boldsymbol{\omega}_j - \dot{\mathbf{r}}_i + \tilde{\mathbf{s}}_i^P \boldsymbol{\omega}_i) = 0 \\ \dot{\Phi}^{(s)} &\equiv \dot{\mathbf{r}}_i - \tilde{\mathbf{s}}_i^P \boldsymbol{\omega}_i - \dot{\mathbf{r}}_j + \tilde{\mathbf{s}}_j^P \boldsymbol{\omega}_j = \mathbf{0} \\ \dot{\Phi}^{(s-s)} &\equiv \mathbf{d}^T (\dot{\mathbf{r}}_j - \tilde{\mathbf{s}}_j^P \boldsymbol{\omega}_j - \dot{\mathbf{r}}_i + \tilde{\mathbf{s}}_i^P \boldsymbol{\omega}_i) = 0\end{aligned}$$

We consider an array of velocities for the two bodies containing their translational and rotational velocities. The coefficient matrix for the velocity array will be referred to as the *Jacobian* matrix. Each velocity equation, whether it contains one or several algebraic equations, can be described in a general form as:

$$\mathbf{D}\mathbf{v} = \mathbf{D}_i \mathbf{v}_i + \mathbf{D}_j \mathbf{v}_j = \mathbf{0} \quad (3.9)$$

where  $\mathbf{v}_i$  and  $\mathbf{v}_j$  each contain six velocity components, and  $\mathbf{D}_i$  and  $\mathbf{D}_j$  are the corresponding sub-Jacobians.

The time derivative of a velocity constraint yields the corresponding acceleration constraint, which can be expressed in the following general form:

$$\mathbf{D}\dot{\mathbf{v}} + \dot{\mathbf{D}}\mathbf{v} = \mathbf{0} \quad (3.10)$$

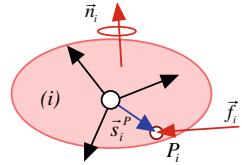
Acceleration constraints contain quadratic velocity terms; i.e.,  $\dot{\mathbf{D}}\mathbf{v}$ . In Table 3.1 entries for the sub-Jacobian matrices and the quadratic terms for the four fundamental constraints.

### 3.2.3 Newton–Euler Equations

In the body-coordinate formulation, Newton–Euler equations provide the simplest description of the equations of motion. For a body with a mass  $m_i$ , if the sum of forces acting on the body is denoted as  $\vec{f}_i$  as shown in Fig. 3.4, the Newton equations of motion describing the translation of the mass center are written as:

$$m_i \ddot{\mathbf{r}}_i = \mathbf{f}_i \quad (3.11)$$

**Fig. 3.4** A force and a moment acting on a body



The Euler's equations describing the rotation of the body are normally expressed in the body-attached  $\xi_i - \eta_i - \zeta_i$  reference frame as:

$$\mathbf{J}_i \dot{\boldsymbol{\phi}}_i = \mathbf{n}_i - \tilde{\boldsymbol{\phi}}_i \mathbf{J}_i \boldsymbol{\omega}_i$$

where  $\mathbf{J}_i$  is a  $3 \times 3$  constant *rotational inertia* matrix. The Euler equations can also be described in the  $x$ - $y$ - $z$  frame as:

$$\mathbf{J}_i \dot{\boldsymbol{\omega}}_i = \mathbf{n}_i - \tilde{\boldsymbol{\omega}}_i \mathbf{J}_i \boldsymbol{\omega}_i \quad (3.12)$$

where  $\mathbf{J}_i = \mathbf{A}_i \mathbf{J}_i \mathbf{A}_i^T$  is no longer a constant matrix.

In order to be consistent with the constraint equations at velocity and acceleration levels, we choose the Euler equations from Eq. (3.12). Appending Eq. (3.12) to Eq. (3.11) provides six equations of motion for a free moving body:

$$\begin{bmatrix} m_i \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_i \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{r}}_i \\ \dot{\boldsymbol{\omega}}_i \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_i \\ \mathbf{n}_i - \tilde{\boldsymbol{\omega}}_i \mathbf{J}_i \boldsymbol{\omega}_i \end{Bmatrix}$$

or,

$$\mathbf{M}_i \dot{\mathbf{v}}_i = \mathbf{g}_i \quad (3.13)$$

If a kinematic joint is present between two bodies, the *reaction force/moment* between the bodies must be included in the equations of motion. If bodies  $i$  and  $j$  are connected by a joint, and the Jacobian submatrices for the joint are described as  $\mathbf{D}_i$  and  $\mathbf{D}_j$ , then Eq. (3.13) is revised for the two bodies as:

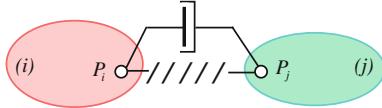
$$\begin{bmatrix} \mathbf{M}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_j \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{v}}_i \\ \dot{\mathbf{v}}_j \end{Bmatrix} = \begin{Bmatrix} \mathbf{g}_i \\ \mathbf{g}_j \end{Bmatrix} + \begin{bmatrix} \mathbf{D}_i^T \\ \mathbf{D}_j^T \end{bmatrix} \boldsymbol{\lambda} \quad (3.14)$$

where  $\boldsymbol{\lambda}$  is an array of Lagrange multipliers containing as many elements as the number of constraints describing the joint.

### 3.2.4 Force Elements

Typical forces that may act on a body are gravitational, spring, damper, and actuator forces. These forces are either constants or position and velocity dependent. Computation of these forces is often a simple and systematic task.

**Fig. 3.5** A point-to-point spring-damper between two bodies



As an example, assume a force element between two bodies is the linear point-to-point spring-damper element shown in Fig. 3.5. In a given configuration (known coordinates) and for known velocity, the force of this element can be determined and applied on the two bodies. A vector connecting the two attachment points is first computed as  $\mathbf{d} = \mathbf{r}_j^P - \mathbf{r}_i^P$ . The length of this vector is  $\ell = (\mathbf{d}^T \mathbf{d})^{1/2}$ . A unit vector along this axis is determined as  $\mathbf{u} = \mathbf{d}/\ell$ . The force of the spring, assuming that the undeformed length and stiffness are known, is computed as  $f^{(s)} = k(\ell - \ell^{(0)})$ . For the damping force we compute  $\dot{\ell} = \mathbf{d}^T \dot{\mathbf{d}}/\ell$ . Assuming that the damping coefficient is known, the damper force is computed as  $f^{(d)} = c \dot{\ell}$ . The force of the element on body  $i$  and body  $j$  is determined as  $\mathbf{f}_i = (f^{(s)} + f^{(d)})\mathbf{u}$  and  $\mathbf{f}_j = -(f^{(s)} + f^{(d)})\mathbf{u}$ . Since these forces have moment arms with respect to the body mass centers, their moments must be included in the rotational equations; i.e.,  $\mathbf{n}_i = \tilde{\mathbf{s}}_i^P \mathbf{f}_i$  and  $\mathbf{n}_j = \tilde{\mathbf{s}}_j^P \mathbf{f}_j$ .

A rotational spring damper, or a rotational actuator (motor) can be defined about the axis of a revolute joint. To model a rotational force element, a relative angle must be defined between the two bodies. Using this angle and its time derivative, in case of a rotational damper, a moment can be determined similar to the force of a point-to-point element. The moment is then applied in opposite directions on the two bodies. Inclusion of a rotational element in the equations of motion is a trivial task in the joint-coordinate formulation, but it requires more calculations in the body-coordinate method.

### 3.2.5 Complete Set of Equations of Motion

Assume that a multibody system contains  $n_b$  rigid bodies. An array of coordinates,  $\mathbf{c}$ , is defined containing  $n_v$  coordinates. The number of coordinates could be either  $6 \times n_b$  or  $7 \times n_b$  depending whether we use Euler angles or Euler parameters. In our discussion, let us assume we are using Euler angles, and therefore  $n_v = 6 \times n_b$ . Arrays of velocities and accelerations are defined as  $\mathbf{v}$  and  $\dot{\mathbf{v}}$  each containing  $n_v$  elements.

The position constraints are  $n_c$  nonlinear algebraic equations as:

$$\Phi(\mathbf{c}) = \mathbf{0} \quad (3.15)$$

The  $n_c$  velocity constraints are written as:

$$\dot{\Phi} \equiv \mathbf{D} \mathbf{v} = \mathbf{0} \quad (3.16)$$

Similarly, the  $n_c$  acceleration constraints are expressed as:

$$\ddot{\Phi} \equiv \mathbf{D}\ddot{\mathbf{v}} + \dot{\mathbf{D}}\mathbf{v} = \mathbf{0} \quad (3.17)$$

If all  $n_c$  constraints are independent, then the system has  $n_{\text{dof}} = n_e - n_c$  DOF.

The Newton–Euler equations are expressed as  $n_v$  equations as

$$\mathbf{M}\mathbf{v} - \mathbf{D}^T\lambda = \mathbf{g} \quad (3.18)$$

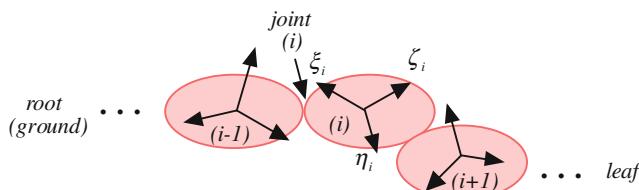
where there are  $n_c$  multipliers in the array  $\lambda$ . We note that Eqs. (3.15–3.18) represent a set of mixed DAEs.

### 3.3 Joint-Coordinate Formulation

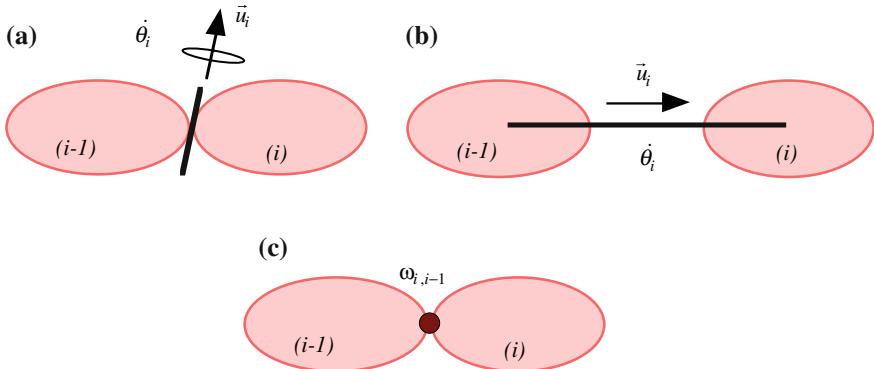
In this formulation, the equations of motion from the body coordinates are transformed into a smaller set [4–6]. The generalized coordinates in this formulation are mostly relative coordinates associated with the kinematic joints that connect the bodies. This formulation is first described for *open-chain* systems (no closed kinematic loops), and then extended to *closed-chain* systems.

#### 3.3.1 Open-Chain Systems

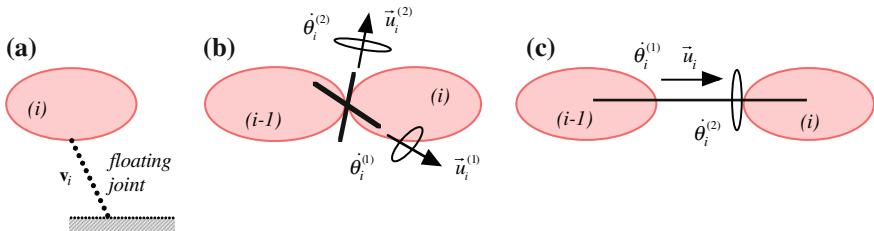
An open-chain multibody system can be viewed as a *tree*. In order to negotiate through every component of a system, we start from the *ground* (or the root) and move to a body via its joint, and then to another body and so on until we reach a *leaf* where no further bodies could be reached. A leaf represents the end of a *branch*. Bodies are numbered in any desired order; however, numbering in ascending order provides easier bookkeeping. As shown in Fig. 3.6, for a typical body  $i$ , the attached body closer to the root is referred to as body  $i-1$ , or the body before body  $i$ . The joint between these two bodies is said to belong to body  $i$ , and therefore all of its attributes carry the index of body  $i$ . Each body will be assigned a body-fixed reference frame with the origin at the mass center as in the body-coordinate formulation.



**Fig. 3.6** Body and joint indices



**Fig. 3.7** Joint velocities: **a** revolute; **b** prismatic; and **c** spherical



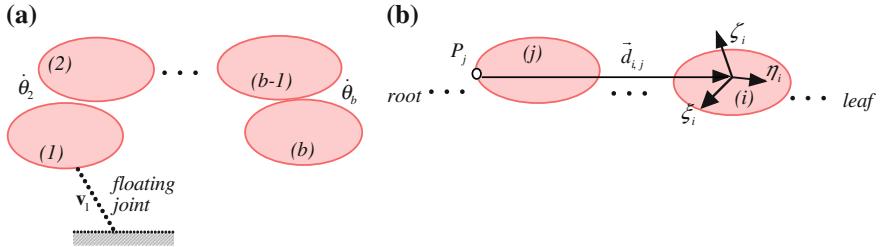
**Fig. 3.8** Joint velocities: **a** floating joint; **b** universal; and **c** cylindrical

In an open-chain system, each kinematic joint is represented by a joint coordinate and an associated joint velocity. The number of components in a joint velocity is equal to the number of DOF of the joint. Figure 3.7 shows three typical kinematic joints: a revolute joint with 1 rotational velocity about the joint axis; a prismatic joint with 1 translational velocity along the joint axis; and a spherical joint with a relative angular velocity vector containing three components. If a body does not own an actual joint, it is assigned a floating joint with 6-DoF,  $\mathbf{v}_i = \{\mathbf{r}_i^T \quad \boldsymbol{\omega}_i^T\}^T$ , as depicted in Fig. 3.8a. Single DOF joints can be combined to represent joints with more than 1 DOF. For example, as shown in Fig. 3.8b and c, a universal joint is represented by two revolute joints, and a cylindrical joint is represented by one revolute joint and one prismatic joint.

For the system shown in Fig. 3.9a with  $b$  bodies, whether being a single branch or multi-branch tree, an array of joint velocities is defined as:

$$\dot{\theta} \equiv \{\mathbf{v}_1^T \quad \dot{\theta}_2 \quad \dots \quad \dot{\theta}_b\}^T \quad (3.19)$$

This array contains as many velocity components as the number of DOF. A useful definition is vector  $\vec{d}_{i,j}$ , shown in Fig. 3.9b, connecting the origin of body  $i$  to a point on joint  $j$ . This vector is computed as:



**Fig. 3.9** **a** A single branch system; **b** Definition of vector  $\vec{d}_{i,j}$

$$\mathbf{d}_{i,j} = \mathbf{r}_i - \mathbf{r}_j^P \quad (3.20)$$

By deriving the velocity relationships, it can be shown that a *velocity transformation matrix* between the *body velocities* and the *joint velocities* exists as:

$$\mathbf{v} = \mathbf{B} \dot{\boldsymbol{\theta}} \quad (3.21)$$

Matrix  $\mathbf{B}$  is orthogonal to the system Jacobian; i.e.,

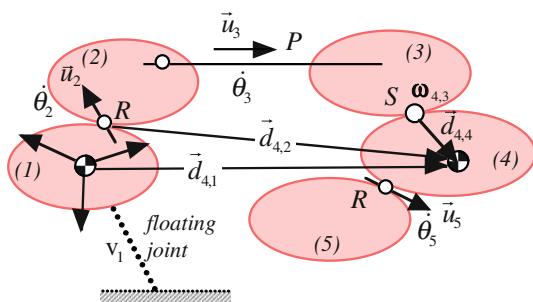
$$\mathbf{D}\mathbf{B} = \mathbf{0} \quad (3.22)$$

The structure of matrix  $\mathbf{B}$  can be demonstrated through a simple example. Consider the open-chain single-branch system shown in Fig. 3.10 containing five bodies, one floating joint, two revolute joints, one prismatic joint, and one spherical joint. The velocity equation of Eq. (3.21) can be written for all five bodies. For example, for body 4 we can write

$$\mathbf{v}_4 = \mathbf{F}_{4,1}\mathbf{v}_1 + \mathbf{R}_{4,2}\dot{\theta}_2 + \mathbf{P}_{4,3}\dot{\theta}_3 + \mathbf{S}_{4,4}\omega_{4,3}$$

This equation states that in order to reach body 4 starting from the ground, we need to go through the floating joint 1, revolute joint 2, prismatic joint 3, and spherical joint 4. The matrices used in this equation are called *block matrices* and they are described in Table 3.2. The description of the  $\mathbf{d}_{i,j}$  vectors that appear in the block-matrices for our example is shown in Fig. 3.10. For a floating joint we have

**Fig. 3.10** An example for constructing the velocity transformation matrix



**Table 3.2** Elementary block-matrices

Type	Size	ID	Entries	Time derivative
Floating	$6 \times 6$	$\mathbf{F}_{i,j}$	$\begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{d}}_{i,j} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$	$\begin{bmatrix} \mathbf{0} & -\mathbf{d}_{i,j} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$
Revolute	$6 \times 1$	$\mathbf{R}_{i,j}$	$\begin{bmatrix} -\tilde{\mathbf{d}}_{i,j}\mathbf{u}_j \\ \mathbf{u}_j \end{bmatrix}$	$\begin{bmatrix} -(\mathbf{d}_{i,j} + \tilde{\mathbf{d}}_{i,j}\tilde{\omega}_j)\mathbf{u}_j \\ \tilde{\omega}_j\mathbf{u}_j \end{bmatrix}$
Prismatic (translational)	$6 \times 1$	$\mathbf{P}_{i,j}$	$\begin{bmatrix} \mathbf{u}_j \\ \mathbf{0} \end{bmatrix}$	$\begin{bmatrix} \tilde{\omega}_j\mathbf{u}_j \\ \mathbf{0} \end{bmatrix}$
Spherical	$6 \times 3$	$\mathbf{S}_{i,j}$	$\begin{bmatrix} -\tilde{\mathbf{d}}_{i,j} \\ \mathbf{I} \end{bmatrix}$	$\begin{bmatrix} -\mathbf{d}_{i,j} \\ \mathbf{0} \end{bmatrix}$

$\mathbf{d}_{1,1} = \mathbf{0}$ . We note that the row and column indices in this matrix correspond to the body and joint indices, respectively. A more detailed example for constructing the velocity transformation matrix is provided in Sect. 3.4.

The time derivative of Eq. (3.21) yields the acceleration transformation as:

$$\dot{\mathbf{v}} = \mathbf{B}\dot{\boldsymbol{\theta}} + \dot{\mathbf{B}}\dot{\boldsymbol{\theta}} \quad (3.23)$$

Substituting Eqs. (3.21) and (3.23) into the constraints of Eqs. (3.16) and (3.17), and then using Eq. (3.22) reveals that all of the constraints disappear when we transform the constraints to the joint coordinates. Substituting Eq. (3.23) into Eq. (3.18) and pre-multiplying the result by  $\mathbf{B}^T$ , then using Eq. (3.22) yields the equations of motion as:

$$\mathbf{M}\ddot{\boldsymbol{\theta}} = \mathbf{f} \quad (3.24)$$

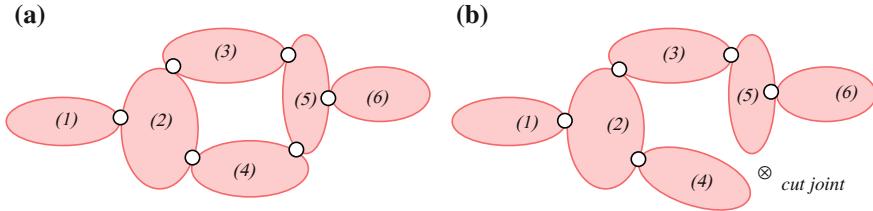
where,

$$\mathbf{M} = \mathbf{B}^T \mathbf{M} \mathbf{B}, \mathbf{f} = \mathbf{B}^T (\mathbf{g} - \mathbf{M} \mathbf{B} \dot{\boldsymbol{\theta}}) \quad (3.25)$$

Equation (3.25) represents as many second-order differential equations as the number of DOF.

### 3.3.2 Closed-Chain Systems

A multibody system containing one or more closed chains can temporarily be transformed into an open-chain system (known as a *reduced* system) by removing one kinematic joint from each loop. The removed joints are called *cut-joints* and they are denoted by the symbol “ $\otimes$ ”. Figure 3.11 shows schematically a system before and after the cut-joint process. The reduced system is first modeled by joint-coordinate formulation, where on joint coordinates are defined for the cut joints, and then the cut joints are put back into the system. When a cut joint is put back in a loop, the joint coordinates in that loop become dependent on one another. Therefore, for the cut joints, kinematic constraints can be written among the body coordinates, velocities, and accelerations as:



**Fig. 3.11** **a** A closed-chain system and **b** its reduced open-chain system

$$\begin{aligned}\Phi^\otimes(\mathbf{c}) &= \mathbf{0} \\ \dot{\Phi}^\otimes \equiv \mathbf{D}^\otimes \mathbf{v} &= \mathbf{D}^\otimes \mathbf{B} \dot{\theta} = \mathbf{C} \dot{\theta} = \mathbf{0} \\ \ddot{\Phi}^\otimes \equiv \mathbf{D}^\otimes \dot{\mathbf{v}} + \dot{\mathbf{D}}^\otimes \mathbf{v} &= \mathbf{C} \ddot{\theta} + \dot{\mathbf{C}} \dot{\theta} = \mathbf{0}\end{aligned}\quad (3.26)$$

where,

$$\begin{aligned}\mathbf{C} &\equiv \mathbf{D}^\otimes \mathbf{B} \\ \dot{\mathbf{C}} &= \mathbf{D}^\otimes \dot{\mathbf{B}} + \dot{\mathbf{D}}^\otimes \mathbf{B}\end{aligned}\quad (3.27)$$

Since the position constraints are nonlinear in the coordinates, the first equation in Eq. (3.26) is kept in terms of the body coordinates. For computational purposes, the term  $\dot{\mathbf{C}} \dot{\theta}$  can be expressed as  $\dot{\mathbf{C}} \dot{\theta} = \mathbf{D}^\otimes \dot{\mathbf{B}} \dot{\theta} + \dot{\mathbf{D}}^\otimes \mathbf{v}$ .

In the presence of constraints in a closed-chain system, the open-chain equations of motion; i.e., Eq. (3.24), are revised with the aid of Lagrange multipliers to obtain:

$$\mathbf{M} \ddot{\theta} - \mathbf{C}^T \otimes \lambda = f \quad (3.28)$$

Equations (3.26–3.28) provide mixed DAEs of motion for a closed-chain system. The number of equations, however, is much smaller than that of the body-coordinate formulation.

The equations of motion for a closed-chain system can be reduced to a minimal set by applying a second step of velocity transformation. For the multibody system of interest, we define a subset of the joint velocities that have been defined for the reduced system as an independent set of joint velocities,  $\dot{\theta}^{(i)}$ . A closed-chain velocity transformation matrix  $\mathbf{E}$  can be expressed as:

$$\dot{\theta} = \mathbf{E} \dot{\theta}^{(i)} \quad (3.29)$$

Matrix  $\mathbf{E}$  can be obtained from the  $\mathbf{C}$  matrix as:

$$\dot{\Phi}^\otimes \equiv \mathbf{C} \dot{\theta} = \mathbf{C}^{(i)} \dot{\theta}^{(i)} + \mathbf{C}^{(d)} \dot{\theta}^{(d)} = \mathbf{0}$$

where  $\dot{\theta}^{(d)} = -\mathbf{C}^{(d)-1} \mathbf{C}^{(i)} \dot{\theta}^{(i)}$  contains dependent joint velocities. This yields:

$$\dot{\theta} \equiv \begin{bmatrix} \dot{\theta}^{(i)} \\ \dot{\theta}^{(d)} \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{C}^{(d)-1} \mathbf{C}^{(i)} \end{bmatrix} \dot{\theta}^{(i)} \quad (3.30)$$

Therefore, the closed-chain velocity transformation matrix becomes:

$$\mathbf{E} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{C}^{(d)-1} \mathbf{C}^{(i)} \end{bmatrix} \quad (3.31)$$

This matrix has the following characteristics:

$$\mathbf{C} \mathbf{E} = \mathbf{0} \quad (3.32)$$

In most practical applications, matrix  $\mathbf{C}$  associated with a closed chain is very small in dimensions. Therefore,  $\mathbf{E}$  can efficiently be evaluated in numerical form.

The acceleration transformation equation is written as:

$$\ddot{\theta} = \mathbf{E} \ddot{\theta}^{(i)} + \dot{\mathbf{E}} \dot{\theta}^{(i)} \quad (3.33)$$

It can be found that the term  $\dot{\mathbf{E}} \dot{\theta}^{(i)}$  can be computed as  $\dot{\mathbf{E}} \dot{\theta}^{(i)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{C}^{(d)-1} \dot{\mathbf{C}}^{(i)} \dot{\theta} \end{bmatrix}$ .

Substituting Eq. (3.33) into Eq. (3.28), pre-multiplying the result by  $\mathbf{E}^T$ , and then employing Eq. (3.32) yield:

$$\mathbf{M}^{(i)} \ddot{\theta}^{(i)} = \mathbf{f}^{(i)} \quad (3.34)$$

where,

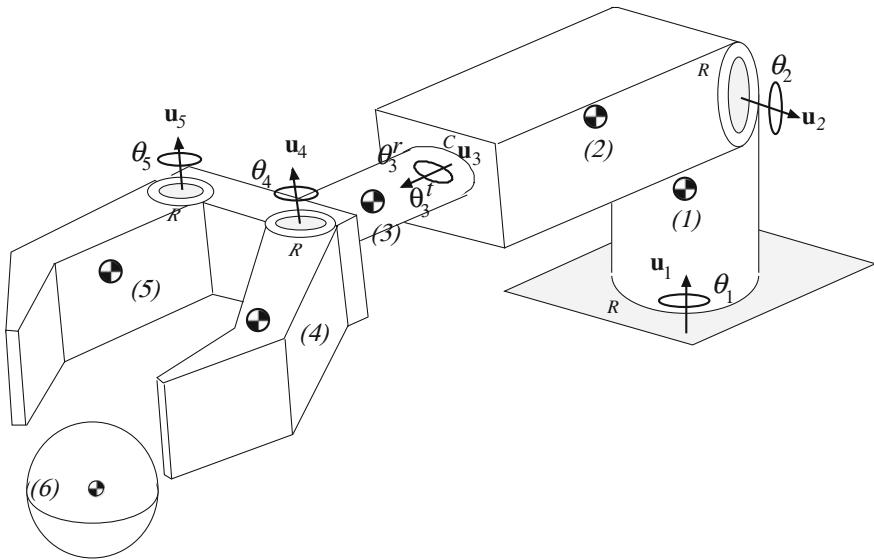
$$\begin{aligned} \mathbf{M}^{(i)} &= \mathbf{E}^T \mathbf{M} \mathbf{E} \\ \mathbf{f}^{(i)} &= \mathbf{E}^T (\mathbf{f} - \mathbf{M} \mathbf{E} \ddot{\theta}^{(i)}) \end{aligned} \quad (3.35)$$

Equation (3.35) represents a set of highly nonlinear second-order differential equations. There are no constraints or any associated Lagrange multipliers in these equations.

### 3.4 Example

In this section, an example is provided to demonstrate some of the important aspects of modeling a multibody system. The example is a robot arm containing 6-DOF. The system contains four revolute joints ( $R$ 's) and one cylindrical joint ( $C$ ). The links of the robot are numbered as shown in Fig. 3.12. We consider this system for joint-coordinate formulation.

The mass of each motor/actuator is distributed to the mass/inertia of the attached links. The mass center for each link is identified. Body-fixed reference frames are defined for each link (not shown). A  $6 \times 6$  mass matrix Eq. (3.13) is constructed for each link. A  $6 \times 1$  array of force/moment Eq. (3.13) is constructed



**Fig. 3.12** A robot as an open-chain system

for each body. These arrays contain the gravitational force, the torque (or force) of each actuator, the gyroscopic velocity terms, and any other applied forces/moment such as viscous friction. When the array of force/moment is transformed into the joint coordinate space, it finds a very simple form, which will be discussed later.

For the joint-coordinate formulation, unit vectors and joint coordinates are defined as shown. The array of joint coordinates is defined as

$$\boldsymbol{\theta} = \{ \theta_1 \quad \theta_2 \quad \theta_3^t \quad \theta_3^r \quad \theta_4 \quad \theta_5 \}^T$$

Note that  $\theta_3^t$  and  $\theta_3^r$  represent the translational (as a prismatic joint) and rotational (as a revolute joint) DOF associated with the cylindrical joint. The velocity transformation matrix  $\mathbf{B}$  is constructed as

$$\begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{v}_4 \\ \mathbf{v}_5 \end{Bmatrix} = \begin{bmatrix} \mathbf{R}_{1,1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}_{2,1} & \mathbf{R}_{2,2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}_{3,1} & \mathbf{R}_{3,2} & \mathbf{P}_{3,3} & \mathbf{R}_{3,3} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}_{4,1} & \mathbf{R}_{4,2} & \mathbf{P}_{4,3} & \mathbf{R}_{4,3} & \mathbf{R}_{4,4} & \mathbf{0} \\ \mathbf{R}_{5,1} & \mathbf{R}_{5,2} & \mathbf{P}_{5,3} & \mathbf{R}_{5,3} & \mathbf{0} & \mathbf{R}_{5,5} \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3^t \\ \dot{\theta}_3^r \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{Bmatrix}$$

Using the block matrices from Table 3.2 we can compute all the terms in matrix  $\mathbf{B}$  based on the components of the unit vectors and the  $\mathbf{d}_{i,j}$  vectors. Figure 3.13 depicts some of these vectors for clarification.

Constructing the mass matrix for the equations of motion in Eq. (3.24) requires computing  $\mathbf{M} = \mathbf{B}^T \mathbf{M} \mathbf{B}$ . Although the three matrices,  $\mathbf{B}^T$ ,  $\mathbf{M}$ , and  $\mathbf{B}$  can be

multiplied numerically, closed form multiplication of these matrices and further simplification would reveal the inner structure of the resultant  $6 \times 6$  matrix. Obviously, some of the terms in this matrix are functions of the joint coordinates.

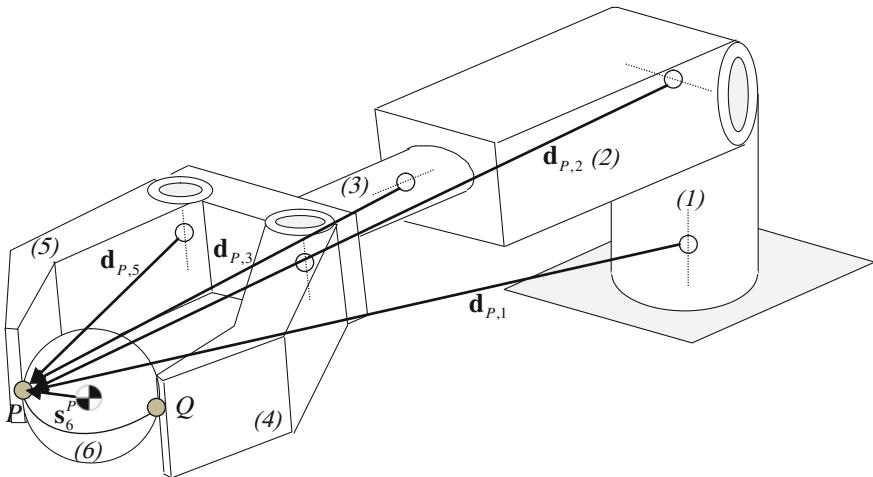
To compute the array of generalized forces; i.e.,  $f = \mathbf{B}^T(\mathbf{g} - \mathbf{MB}\dot{\theta})$ , we need the  $\mathbf{g}$  array from the body-coordinate step and the  $\dot{\mathbf{B}}\dot{\theta}$  array. It is most convenient to compute the  $6 \times 1$  array  $\mathbf{B}^T\mathbf{MB}\dot{\theta}$  numerically. Similarly, the  $6 \times 1$  array  $\mathbf{B}^T\mathbf{g}$  can be computed numerically, however, it is important to recognize the structure of this array.

Assume that actuators control each DOF of the robot; i.e., there are one linear and five rotational actuators. If we consider the contribution of these actuators in the  $\mathbf{g}$  array as  $\mathbf{g}^{(a)}$ , then the product  $\mathbf{B}^T\mathbf{g}^{(a)}$  after simplification becomes  $\mathbf{B}^T\mathbf{g}^{(a)} = \{n_1 \ n_2 \ f_3 \ n_3 \ n_4 \ n_5\}^T$ , where  $n_i$  is the torque of the  $i$ -th rotational actuator and  $f_3$  is the force of the linear actuator. This indicates that for the moments and forces that act respectively about and along the joint axes, their contribution can be included in the generalized force array directly. This is an important feature of the joint-coordinate formulation when the actuator forces are adjusted by a control system. For the robot without contact with any objects the equations of motion of Eq. (3.24) yield six second-order differential equations.

Next, we consider the robot to pick up an object, such as the spherical object denoted as body 6. We assume that links 4 and 5 contact the sphere at points  $Q$  and  $P$ , as shown in Fig. 3.13. Since a closed chain has been created, the two contact points can be considered as cut joints. Each contact point, or cut joint, can be represented as a spherical joint represented by constraints of Eq. (3.7). For the spherical joint at  $P$  bodies 5 and 6 are involved, and for the spherical joints at  $Q$  bodies 4 and 6 are involved. The Jacobian matrix  $\mathbf{D}^\otimes$  for these two cut joints is constructed in body coordinates. Then, based on Eq. (3.26), the joint-coordinate Jacobian  $\mathbf{C}$  is constructed as  $\mathbf{C} = \mathbf{D}^\otimes \mathbf{B}$ . If the elements of  $\mathbf{D}^\otimes$  and  $\mathbf{B}$  are multiplied in closed form, matrix  $\mathbf{C}$  is found to be

$$\mathbf{C} = \begin{bmatrix} \tilde{\mathbf{d}}_{P,1}\mathbf{u}_1 & | & \tilde{\mathbf{d}}_{P,2}\mathbf{u}_2 & | & \mathbf{u}_3 & | & \tilde{\mathbf{d}}_{P,3}\mathbf{u}_3 & | & \mathbf{0} & | & \tilde{\mathbf{d}}_{P,5}\mathbf{u}_5 & | & -\mathbf{I} & | & \tilde{\mathbf{s}}_6^P \\ \tilde{\mathbf{d}}_{Q,1}\mathbf{u}_1 & | & \tilde{\mathbf{d}}_{Q,2}\mathbf{u}_2 & | & \mathbf{u}_3 & | & \tilde{\mathbf{d}}_{Q,3}\mathbf{u}_3 & | & \mathbf{d}_{Q,4}\mathbf{u}_4 & | & \mathbf{0} & | & -\mathbf{I} & | & \tilde{\mathbf{s}}_6^Q \end{bmatrix} \quad ()$$

Some of the  $\mathbf{d}$  vectors in this matrix are depicted in Fig. 3.13. These vectors originate from the joint definition points that are used in defining the joint constraints in body coordinates, and they end up either at  $P$  or  $Q$ . The first six columns of this matrix correspond to the six joint coordinates of the robot. The last two columns (actually six columns in expanded form) represent three translational and three rotational coordinates of body 6. Each row of the matrix represents three algebraic rows. Therefore, in expanded form, this is a  $6 \times 12$  matrix resulting in  $12-6 = 6$  DOF. Four DOF are associated with the revolute joints 1 and 2, and the cylindrical joint 3. One DOF belongs to the planar four-bar system formed by links 3–6, and the final DOF corresponds to the rotation of body 6 about the axis  $PQ$ .



**Fig. 3.13** Some of the position vectors used in the construction of matrix  $\mathbf{C}$

The complete set of equations of motion for the robot holding an object is formed by constructing six algebraic constraints, in body coordinates (Eq. 3.26), for the two spherical joints, their first and second time derivatives of the constraints in joint coordinates (Eq. 3.26), and six equations of motion (Eq. 3.28). This is a mixed set of DAEs. It is recommended that the equations be kept in this form, and not transformed into a smaller set (Eq. 3.34).

### 3.5 Analysis

Typical types of analysis in multibody dynamics are kinematic, inverse dynamic, and forward dynamic. It is also possible to perform static, static equilibrium, or steady-state analysis as well.

In a kinematic analysis time functions, as many as the number of system's DOF, are defined. A time function should provide values for a given coordinate and its time derivatives at any instant of time. This process transforms the equations of motion into algebraic equations (nonlinear and linear) that must be solved for the remaining coordinates, velocities, and accelerations.

Inverse dynamics is an extension of kinematic analysis, where after the complete kinematics of the system is determined, unknown reaction forces, as well as the forces/torques associated with the kinematic time functions can be determined. This process, particularly in applications such as robotics, provides the means to determine the required actuator forces/torques to derive the system according to a desired scenario.

In a forward dynamic analysis, the motion of the system in response to a set of applied (external) loads is determined by numerically solving differential

equations of motion. Depending on the method of formulating the equations of motion, the equations could be pure differential equations (but highly nonlinear), or mixed DAEs. Forward dynamics is the most common form of analysis in multibody systems. It should be noted that forward dynamic procedures could also be applied to analyze a system for kinematics (and inverse dynamics) by treating the corresponding algebraic equation as special form of differential equations. Interested reader may refer to [7] for detailed description and algorithms for various analysis methods in multibody dynamics.

### 3.6 Multibody Dynamics Software

In this section, a list for some available commercial and free multibody software is provided. The list contains the name of the product, the developer and/or distributor, and the URL. A software is marked *freeware* only if its website has stated that clearly. The list is only for general information. For detailed information for each software, interested reader should refer to its website. The products are listed alphabetically.

#### **ADAMS**

MSC software

<http://www.mscsoftware.com/Products/CAE-Tools/Adams.aspx>

#### **DCAP**

ESA

[http://www.esa.int/TEC/mechanisms/SEMZX356JGG\\_0.html](http://www.esa.int/TEC/mechanisms/SEMZX356JGG_0.html)

#### **Dymola**

Dassault Systèmes

<http://www.3ds.com/products/catia/portfolio/dymola>

#### **HOTINT/MBS (Freeware)**

Institute of Technical Mechanics, Johannes Kepler University of Linz

<http://tmech.mechatronik.uni-linz.ac.at/staff/gerstmayr/hotint.html>

#### **LMS Virtual.Lab Motion**

LMS

<http://www.lmsintl.com/simulation/virtuallab/motion>

#### **MADYMO**

Advanced Simulation Technologies Ltd.

<http://www.advancesimtech.com/software/madymo/>

#### **MBDyn (Freeware)**

Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano

<http://www.aero.polimi.it/mbdyn/>

**MBS3D** (Freeware)

INSIA and ETSII, Technical University of Madrid

<http://mat21.etsii.upm.es/mbs/mbs3d/index.htm>**MBSim** (Freeware)

BerliOS Developer

<http://mbsim.berlios.de/index.html>**MotionSolve**

Altair

<http://www.altairhyperworks.com/Product,17,MotionSolve.aspx>**RecurDyn**

MotionPort

<http://www.motionport.com/index.aspx?page=RecurDyn>**ROBOTRAN** (Freeware)

CEREM

<http://www.robotran.be/>**Simbody** (Freeware)

SimBios

<https://simtk.org/home/simbody>**SIMPACK**

SIMPACK AG

<http://www.simpack.com/>**Solfec** (Freeware)

Civil Engineering Department, University of Glasgow

<http://code.google.com/p/solfec/>**SolidWorks**

Dassault Systèmes SolidWorks Corp.

[http://www.solidworks.com/sw/products/10169\\_ENU\\_HTML.htm](http://www.solidworks.com/sw/products/10169_ENU_HTML.htm)**SPACAR** (Freeware)

Laboratory of Mechanical Automation, University of Twente

<http://www.wa.ctw.utwente.nl/software/spacar/2011/intro/index.html>**Universal Mechanism**

Laboratory of Computational Mechanics, Bryansk State Technical University

<http://www.umlab.ru/>**Vortex**

CM Labs

<http://www.vxsim.com/en/software/index.php>

**Working Model (2D)**

Design Simulation Technologies

<http://workingmodel.design-simulation.com/>**References**

1. Nikravesh PE (1988) Computer-aided analysis of mechanical systems. Englewood Cliffs, Prentice-Hall
2. Nikravesh PE, Affifi HA (1994) Construction of the equations of motion for multibody dynamics using point and joint coordinates. Computer-aided analysis of rigid and flexible mechanical systems. NATO ASI Series E: Applied Sciences. Kluwer academic publishers, Dordrecht, vol 268, pp 31–60
3. Garcia de Jalon J, Bayo E (1994) Kinematic and Dynamic Simulations of Multibody Systems. Springer-Verlag, New York
4. Jerkovsky W (1978) The structure of multibody dynamics equations. *J Guidance Control* 1(3):173–182
5. Kim SS, Vanderploeg MJ (1986) A general and efficient method for dynamic analysis of mechanical systems using velocity transformation. *ASME J Mech Trans Auto Des* 108: 176–182
6. Nikravesh PE, Gim G (1993) Systematic construction of the equations of motion for multibody systems containing closed kinematic loops. *ASME J Mech Des* 115(1):143–149
7. Nikravesh PE (2008) Planar multibody dynamics: formulation, programming, and applications. CRC Press, Taylor & Francis Group, Boca Raton

# **Chapter 4**

## **Sensors and Methods for the Evaluation of Grasping**

**Antonio Morales, Mario Prats and Javier Felip**

**Abstract** Evaluation of robot grasping consists in assessing whether a grasp on a target object meets certain desirable contact properties related with its robustness, reliability, and stability. Traditionally, most works on robot grasp planning have focused on pre-grasp analysis, where the target objects and tasks are considered in order to provide a set of feasible candidate grasps. Evaluation has been restricted to rank them in order to ease the selection. Form and force closure criteria have been defined to determine the grasp feasibility and several quality metrics have been developed to evaluate the goodness of a given grasp. However, grasp evaluation is not only concerned with the planning stage. It is also relevant while the grasping action is occurring or has been completed. For this kind of approaches a variety of sensor modalities can be used in order to assess and improve the state of the grasp. The most common sensors are contact based, though visual input and proximity sensors are also used. The aim of sensing capabilities in this stage is to provide feedback in order to improve or adapt the robot hand to the objects and environment conditions. In addition sensing information can also be used to provide post-grasp quality tests. This chapter gives an overview of sensor-based grasp evaluation. On the first place a survey on the major sensor modalities and technologies is provided, with special focus on contact-based sensors. Commercially available sensors and their applications are described. This survey is completed with a number of illustrative cases in which sensor feedback is used to evaluate the goodness of grasping and to improve its execution.

---

A. Morales (✉) · M. Prats · J. Felip  
Robotic Intelligence Lab, University Jaume I of Castellón,  
Avd. Sos Baynat S/N E-12061 Castellón de la Plana, Spain  
e-mail: Antonio.Morales@uji.es

## 4.1 Introduction

The field of robot grasping and manipulation comprehends all of those techniques that allow robots, mostly manipulators arms endowed with simple grippers or complex hands, to interact with objects in its environment with the purpose to bringing them to a desired state. Most specifically, *robot grasping* aims to manipulate objects by immobilizing them with the robot effector through the fixtures and forces exerted at the physical contacts. Within this goal, the ability to determine whether a given grasp is reliable, robust or even feasible is a key part on the planning, analysis, and execution of grasping tasks.

Most of the theoretical and analytical work in the field of robot grasping have focused on (1) determining on which conditions a given configuration of the robot effector ensures the immobilization of the object and (2) computing stable configurations given a certain object [34]. In fact, works on *grasp analysis* are devoted to establishing which conditions contact configurations must meet in order to ensure full constraining of the object. Note that a grasp of an object is abstracted as a set of contacts on the object, which is a physically meaningful representation of the effector configuration.

Seminal works established the *form-closure* and *force-closure* conditions as necessary for grasp stability. Later works introduced the mathematical apparatus to analyze the requirements of these conditions in the cases of 2D and 3D objects, considering friction-less, friction, and soft contact models [4]. A relevant topic on most of these works was to establish the minimum number of contacts necessary to achieve the closure conditions [5].

*Grasp planning*, also referred often as *grasp synthesis*, goes a step further, and tries to determine which contact configurations meet the closure and equilibrium conditions given the shape of the target object. Here, most of the works focus on designing theoretical and computational algorithms to determine all stable contact configurations. These algorithms commonly produce a large amount of solutions, often infinite. In order to discriminate and rank them, a number of *quality metrics* have been proposed which prioritize different aspects of the found solutions [7].

Most of the works mentioned above focus on the evaluation of the grasps in the planning stage, before the actual execution of the computed grasp. The execution usually consists of an *approximation* phase where the robot arm moves the end-effector toward the vicinity of the object; a *closing* phase, during which the robot hand approaches the object, makes the first contact and finally closes the fingers; and finally in the *transportation* phase in which the object is moved to the desired destination.

The successful execution of these phases can be a challenging control problem especially when uncertainty and inaccuracy of the knowledge of the robot environment increases. These uncertain conditions, common when addressing service robot applications, can cause that the final executed grasp is completely different from the initial candidate, which causes the planning analyzes useless [14].

Methodologies able to robustly execute grasps in the presence of uncertainty are necessary. Some engineering work has addressed this problem by designing robot hands whose compliant mechanism adapts to unknown objects and constraint their mobility [11].

This chapter discusses the use of sensor feedback to evaluate the properties of a grasp while it is being executed in order to control the hand toward a more robust contact configuration. The focus is placed on contact-based sensors, though visual and combined approaches are also described.

#### **4.1.1 Chapter Outline**

In the first place, Sect. 4.2 surveys the sensor modalities most often used on robot grasping and manipulation with special emphasis on contact-based sensors and the commercially available technologies (Sect. 4.2.2), though other non contact modalities like vision are also referred (Sect. 4.2.1).

The remaining of the chapter is devoted to describe several cases of the use of sensor information to evaluate grasp executions. The first case (Sect. 4.3) shows how vision-based features can be used to identify reliable grasps based on the experience on previous executions. An experimental test is implemented in order to measure the practical reliability of a grasp. The next two cases show how different sensor modalities can be combined and used in order to monitor the goodness of an interaction during manipulation task. In the first case, (Sect. 4.4.1) tactile and force feedback are used to extract a book from a bookshelf. In the second, (Sect. 4.5) vision, force, and tactile readings are applied to open a door with a robot hand. All three cases are explained in details and results of their effectiveness are provided.

## **4.2 Sensor Modalities**

During the manipulation process, there are three main senses used: sight, touch, and hearing. Eyesight has been widely studied by the scientific community, nevertheless vision for manipulation is a small part of what has been done in that field. Vision is the most suitable sensory cue to locate objects in the workspace and identify obstacles on its surroundings. It can also be used to detect contacts and to track the relation between the hand and the manipulated object. On the other hand, vision is not 100 % reliable due to environmental uncertainties, sensing errors, calibration errors, and so on. Another sensor modality is used to deal with that lack of accuracy: the sense of touch. In robotic systems, sense of touch is implemented using three main types of sensors: tactile, force torque, and kinesthetic sensors. Touch information is devoted to local features while vision information is used for global scene understanding. Hearing is a complementary sensor cue that can ease

the detection of contacts by rising the contact likelihood after the hearing of a contact. It can also be used to infer the material of objects and to barely guess the location where a contact happened.

Regarding physical interaction with objects, another possible classification of the different sensory cues can be used: contact and contactless.

#### **4.2.1 Contactless Sensors**

This category is populated by sensors that do not need contact between the sensor and the environment to detect contacts, mainly vision and sound sensors. Other types of sensors such as IR sensors have also been used for grasping and pre-contact [18] detection but not strictly to detect contacts.

Although there is a wide variety of vision sensor types, all of them share a common element: a camera. There is a lot of information from the objects and the environment that can be extracted through visual sensors. Nevertheless, the information that is most relevant to grasping and manipulation is related with objects and obstacles location and their 3D structure. The classical and common approach to obtain 3D information of the environment is the stereo reconstruction system. This method is based in the disparity between two camera images to extract depth information and has been widely addressed. In fact, most of the humanoid robots with ability to grasp objects implement a stereo vision system on their head. Moreover there are several commercial stereo systems that implement all the stereo reconstruction algorithms on hardware.

Another approach is to actively project a pattern to the scene and observe its deformation to obtain the 3D structure, this method is used by the Kinect sensor. Since its release in November 2010 its impact in the robotics community has been quite high. Due to its low cost, high reliability, precision and speed, and lots of roboticists found on Kinect a shortcut to easily get 3D visual information from a scene. With high speed and quality 3D reconstruction enabled, the field of 3D point cloud processing has been on the bleeding edge of perceptual systems, one of the most active groups in this field is the Point Cloud Library (PCL) from Willow Garage [36] that started their project as a ROS package and now is a standalone library that eases the processing of point clouds and offers a lot of tools to use with point clouds, PCL implements also an interface to the Kinect sensor.

There are other range sensors that are also laser-based like LIDAR laser scanners or time-of-flight cameras. These alternatives make sense on applications in which requirements cannot be fitted by the Kinect sensor (i.e range, exteriors, and underwater).

Sound sensors are scarcely used in the robotics community but they can be used to detect contacts and determine precisely the instant when the contact occurred.

It is possible to use visual sensors to determine the grasp quality, but it is not easy neither confident to do it from a still view of the object grasped. A possible idea to detect grasp stability from a visual system could be to track the object and the hand

at the same time while lifting the object and detect mismatches between both movements [32]. Other sensors should be used to detect whether the grasp is stable.

#### 4.2.2 Contact Sensors

Several attempts to mimic the human sense of touch exist [43], using different tactile technology, such as conductive elastomer, elastomer-dielectric capacitive, light-based sensors, organic field-effect transistors, piezoelectric, quantum tunnelling composite, or force sensing resistors, among others [16]. Unfortunately, each type of sensor is focused to a particular application, and it is still not possible to achieve the versatility of the human receptors.

It is worth mentioning that *tactile sensing* and *contact sensing* refer to different concepts. Tactile sensing includes research on skin-like sensors to measure pressure distribution, temperature, softness, etc., whereas contact sensing refers to the perception of forces and torques generated through the contact points during manipulation. Both modalities belong to the touch sense. Often tactile sensing is addressed in robotics by tactile sensors and contact sensing is detected by force-torque sensors. Some tactile sensors are able to provide both tactile and contact information, but, normally, they cover only a small surface of the hand, such as the fingertips or the palm. If the contact is performed with a part which is not covered by the tactile sensors, then there is no available information. In contrast, a force sensor placed at the robot wrist is able to provide the force generated by contacts on any part of the hand, ensuring that contact detection will be always available.

In the robotics literature, tactile feedback has been used mainly for event-driven manipulation [17], which deals with switching between different task states according to a set of sensor-based conditions. Among the interesting events that can be detected with tactile sensors, it is worth mentioning perceiving a contact, an incipient slip, the contact type, changes in the object properties, etc. For example, Cutkosky and Hyde [8] used force and tactile-based events for switching between different manipulation phases. Bicchi et al. [6] and Dario and De Rossi [9] also focused on tactile-based event detection, particularly slip and changes in the object texture. Petrovskaya et al. [31] used tactile feedback in order to recover the position and orientation of an object, following an active sensing approach.

Force feedback represents a fundamental requirement for the success of any task that involves physical interaction with the environment but when trying to use it to assess the quality of a grasp, the force information coming from a wrist sensor is not that important. On the other hand, force sensors could be used to detect contact positions if they are placed on each finger and use that contact information to evaluate grasp stability [19].

Force-torque sensors have been used in industrial processes like load cells. The demands of these systems by industries have driven the force-torque sensing technology to a great development and robustness. Usually force-torque sensors

are 6D: force and torque. There are also 12D sensors that measure also accelerations. Mainly available from JR3 Inc. [21] and ATI Industrial Automaton Inc. [2]. At the beginning force-torque sensors were designed for the wrist of the manipulators. Nowadays, it exists a wide variety of sensor sizes that can be mounted on the robot base or between the hand phalanxes.

The application of tactile sensors to grasp evaluation is more straightforward. Tactile sensors can give information of the surface in contact with the object and use it to determine the stability of the grasp for example using a classification method.

#### 4.2.2.1 Commercial Tactile Sensors

Deploying robots in unstructured environments, requires some adaptation to unexpected events and also management of the inherent uncertainty of real and changing environments. That has driven the different types of sensors to be a must in any robotics platform that is intended to deal with real unstructured scenarios. Along the last decade some companies were found to fill in the tactile sensing business bracket. Commercial tactile sensors enable the robotics research groups all over the world to implement a tactile sensing system into their current robotic platforms without the need of designing and implementing their own sensing systems. There are several technologies used to design tactile sensors each one with its own advantages and drawbacks.

Commercial tactile sensors are usually designed as arrays of sensor cells called “tactels”. Each tactel is the sensing unit and it is sensitive to pressure. Mostly the manufacturers have some defined sizes for their sensors, but it is often possible to order a custom sized sensor to fit any platform needs.

The main features of tactile sensors are:

- **Sensitivity.** The smallest variation of pressure that can be detected by the sensor.
- **Repeatability.** The variation of the sensed value for the same applied pressure.
- **Sample rate.** The speed at which the sensor outputs the data.
- **Range of pressure.** The range of possible values that can be measured.
- **Spatial resolution.** The density of tactels along the sensor surface.

The material and the technology used to build the tactile sensors may influence other issues that are quite common in tactile sensing: hysteresis and low pass filtering. An hysteresis cycle makes possible to have two different sensed values for the same pressure, depending if the pressure is increasing or decreasing. The occurrence of this effect is mainly determined by the technology used to build the tactile sensors. On the other hand, the low pass filtering effect is caused mainly by the material used to cover the tactile sensors. The low pass filter effect acts as a convolution operation over the pressure image that is generated by the tactile sensors causing the loss of details and blurring the data.

These two problems have to be taken into consideration before choosing a sensor system for a determined robotics application. For instance, if the tactile sensors are going to be used to detect details and texture, the low pass filtering effect that appears on most of the soft tactile sensors would cause the loss of the surface details. On the other hand if the application is intended to improve, detect, and perform good grasps without taking into consideration details of the grasped surface, a soft material on the fingertips will get better results easing grasp compliancy and increasing friction. For the improvement and evaluation of grasping, the sensitivity of the sensors is the most important feature. Having a very sensitive sensor will enable the detection of contacts with small and lightweight objects and act consequently.

There are several tactile sensor technologies: resistive, capacitive, piezoelectric, quantum tunneling composite (QTC), ultrasonic, and optical. For almost all the technologies, there is a manufacturer that sells a commercial version. See Table 4.1 for a summary of available products.

### Resistive Sensors

The resistive technology is the most common solution used. It is generally sensitive and cheap but less accurate than other technologies. Commercially available from Interlink [20], it has been used in experimental tactile sensors due to their low cost, noise, and good sensitivity. The main drawback of the Interlink FSR sensors is that there is only one sensing unit. If the desired application needs an array of sensors covering a surface (i.e. fingertips) a manual assembly is required. These sensors were used in early hand tactile systems like DLR Three-fingered hand. Weiss Robotics [46] has designed a resistive  $6 \times 14$  matrix of sensors with a spatial resolution of 3.4 mm, these sensors are covered with a rubber polymer that protects the sensor array and increases the friction with objects. On the other hand, the rubber cover acts as a low pass filter and hides the possible details (i.e. texture) of the contacted surfaces. Weiss tactile sensors are the embedded tactile sensing system for Schunk Dextrous Hand [38] and they are integrated on the hand platform and software.

### QTC Sensors

A very similar approach to FSR is used by Peratech [30] and their QTC-based system. Instead of having a resistor sensitive to pressure, they use a polymer that changes gradually from a perfect isolator to a total conductor. The major advantages are that QTC allows more flexibility regarding the sensor geometry and better sensitivity. This tactile solution is the one implemented by Shadow Robotics [39] in the Shadow Dextrous Hand. The spatial resolution is quite similar to the other tactile sensing systems.

**Table 4.1** Commercial tactile sensors

Manufacturer	Technology	Sensitivity	Sample rate (Hz)
Interlink FSR	Resistive	10 g	60
Weiss Robotics	Resistive	50 g	230
PPS RoboTouch	Capacitive	0–140 KPa	30
PeraTech QTC	QTC	0.01–2.7 MPa	20
Tekscan Flexi-Force	Resistive	Variable	Variable
Syntouch BioTac	Multimodal	No info	No info
Fastenica Skilsens	Optical	No info	No info

### Capacitive Sensors

Regarding capacitive technology, the tactile solutions from Pressure Profile Systems [35] are the most relevant commercial sensor systems. Although the capacitive technology allows more density of tactels, the sensor arrays from PPS offer similar spatial resolution than resistive solutions. The PPS sensory systems are integrated in the Barrett Hand [3], the Twendy One [45] hand, and the PR2 gripper [47]. Capacitive sensors are very sensitive but they suffer of severe hysteresis and stray capacity.

### Optical Sensors

In the past, Kinotex by Tactex was the most important tactile sensing system using optical-based system. Unfortunately in November 2008, the company went bankrupt and the production stopped. Nowadays, the commercial solution using optical technology is Skilsens by Fastenica [13]. Optical-based tactile systems are very sensitive, flexible, and fast but bulky. The size is the main reason that makes this type of sensors difficult to be integrated in state-of-the-art robotic hands. However, the Skilsens system is integrated in two fingers of the prosthetic hand Smart Hand.

### Other Tactile Sensors

In an innovative design, Syntouch [42] has designed a finger-shaped multimodal tactile sensor. This sensor can detect contact locations and force direction using impedance sensing electrodes. It can also detect microvibrations with a liquid pressure sensor and temperature with a thermistor. The sensor is formed of a rigid core where all the sensing units are attached and a conductive fluid covered by an elastic skin. Any change in the elastic skin produces a displacement in the conductive liquid that can be detected by the electrodes to infer the force direction and contact locations. This sensory system seems to be the one implemented in the forthcoming Robonaut hand.

Other commercial tactile solutions can be found: Tekscan flexi-force [44] is a pressure sensitive ink-based sensor system that is mostly suitable for factory environments with very low sensitivity. A piezoelectric-based system with is used by the Fraunhofer institute [15] to cover robotic arms with a tactile skin with the aim of detecting collisions and easing the integration of such robot arms in the household environments.

## 4.3 Grasp Evaluation from Vision

Vision is the most used sensor modality in robot grasping. The reason is that it is able to provide the richest information from unstructured environments. On the contrary, it provides a raw data that can be extremely complex to analyze and occasionally requires large computational resources. In order to make the visual analyzes feasible several types of assumptions are usually taken.

In the realm of robot grasping, visual input can aid on the three main phases of grasping actions: grasp planning, arm approaching, and grasp execution. In the case of grasp planning it plays a leading role in most of the works that show implementations of robots able to grasp. In the second case, the approaching of the robot arm to the target object, it is also a widely developed topic under the name of *visual servoing*. However, there are few cases of applications of visual feedback while the hand is contacting the object.

In this section, we discuss a case in which visual cues can be used to evaluate, or more precisely, predict the stability of a grasp. This case is considered as a part of a grasp planner, since it evaluates or filters the candidate grasps according to specified visual features. However, the grasp is evaluated from past experiences on similar grasps.

### 4.3.1 Grasp Synthesis from Visual Input

The approach applies a simple 2D visual analysis. Objects are assumed to be planar, or more precisely, 2D shapes with a constant height. The 2D contour of the target object is extracted from the images using segmentation techniques. Potential grasp regions are found on the 2D object contours which are modeled by straight lines. From triplets of grasping regions, up to three different thumb placement configurations are generated for the Barrett Hand that satisfy hand kinematics and force closure constraints (assuming a soft finger with a minimum, but unknown, friction coefficient) [26, 27]. The center of intersection of the friction cones in the plane defines a point used as grasp force focus and its projections on the three grasping segments determine both the contact points of the fingers on the object and the force directions. Typically 30–300 configurations are generated for each contour. Each of these contact point triplets are adapted to the physical and kinematical constraints of the real robot hand resulting in up to three hand configurations, one per thumb positioning, for each grasp triplet.

### 4.3.2 Quality Metrics for Hand Configurations

Using this procedure each hand configuration is described by a number of parameters all of them determined from the input images through the implementation phases of the grasp planning:

**Grasping regions.** The portions of the object contour where the three fingers are placed (**bold** in the images).

**Contact points.** The three points where the fingers are supposed to touch the object, each lying on one of the three grasping regions ( $P_1, P_2, P_3$ ).

**Force directions.** The real force directions  $F_1, F_2, F_3$  exerted by the fingers of the Barrett Hand are usually different from the ideal normal directions  $N_1, N_2, N_3$ .

**Force focus.** The intersection of the directions of the force lines.

**Finger extensions.** The opening of the fingers.

**Finger spread.** The opening angle  $\theta$  of the two fingers in opposition to the thumb.

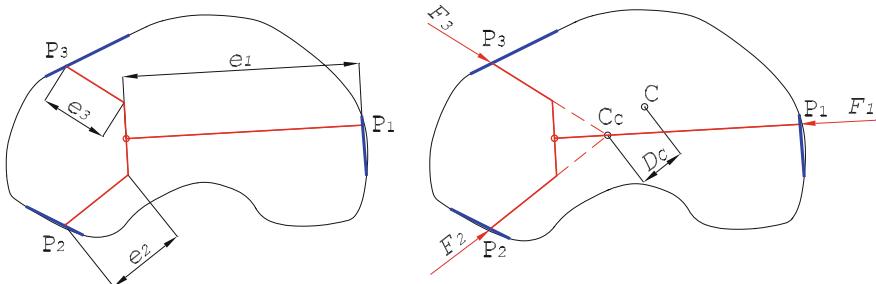
These parameters do not indicate anything about the quality of a given hand configuration. Up to nine quality metrics which evaluate different aspects of the hand configuration are defined using the above parameters. They are adapted from existing quality metrics or novel. Some of them considered the characteristics of the hand, on some others only visual aspects of the analysis.

To illustrate the reader two representative criteria are described in more detail (see also Fig. 4.1):

**REAL FOCUS CENTERING:** This feature aims to measure the effect of gravitational and inertial forces endorsing grasps with short distances between the focus of the grasp  $C_C$  and the center of mass of the object  $C$ . The latter is the centroid of the two-dimensional shape described by the extracted object contour, assuming that the object has uniform mass distribution. The metric definition is  $q_{RFC} = \frac{4D_C}{M_L+m_L}$ , where  $M_L$  and  $m_L$  are the sizes of the major and minor inertia axes computed from the shape contour.

**FINGER EXTENSION:** Ideally fingers should be equally extended. If the fingers contact the object with different extensions, they probably exert a torque out of the horizontal plane of the object. This metric estimates the risk given by the differences in the finger extensions:  $q_{FE} = \frac{1}{\eta^2} ((e_1 - e_2)^2 + (e_2 - e_3)^2 + (e_3 - e_1)^2)$ . The coefficient  $1/\eta^2$  is half of the possible maximum value, which depends on the construction of the hand for the above sum, so that results of the metric lie in the interval  $[0, 2]$ .

In order to be comparable all these metrics are defined over similar ranges, and have the best value at 0. In cases, like the FINGER EXTENSION metric this is achieved applying the ‘halfway’ method, by means of which, a grip halfway the best and the worst will have normalized a quality of 1.



**Fig. 4.1** Finger extension and real focus centering metrics

### 4.3.3 Experimental Grasp Evaluation

The above hand configuration metrics are prediction rules, that must be related with experimental grasp stability in the real robot. In order to do this, we define a experimental protocol to test the stability of a grasp after being executed. A single object is grasped using one of the candidate hand configurations. After closing the hand the object is lifted and three consecutive shaking movements are executed with an increasing acceleration profile. After each movement the tactile sensors are used to check whether the object has been dropped. A reliability class is assigned from A to E where *E* indicates that the robot did not lift the object, *D*, *C*, *B* indicate that the object was dropped on some of the three phases, and *A* indicates that the object was hold successfully.

Finally, a large sample database has been built by carrying out exhaustive series of experiments. Four real objects were used: two with simple shapes and two with more complex shapes. For each object, a subset of the feasible hand configurations was applied several times. Three different combinations of physical properties were tested: light objects and low friction (light/low), heavy objects and high friction (heavy/high); and light objects and high friction (light/high). At the end, more than eight hundred execution samples were recorded. Table 4.2 shows the number of different grips executed and the percentages of grips that resulted in each class. We define  $\Omega = \{A, B, C, D, E\}$  as the set of reliability classes.

### 4.3.4 Prediction Scheme

A step further is to use the collected date in order to predict the most likely reliability class of a candidate grip. In theoretical terms a data set of previous experience is composed of  $N$  executed triplets. Each grip  $g_i, i = 1 \dots N$  is described by the nine visual features  $q_1, \dots, q_9$  introduced in Sect. 4.3.2. The 9-dimensional space  $G_S$  is formed by the ranges of the values of the features. Moreover, we have also recorded the performance of the grip and have assigned it to a class  $\omega_i \in \Omega$  for each  $g_i$ .

**Table 4.2** Sample data sets

	E	D	C	B	A	Total
Light	102	84	33	27	18	264
Low	38.6 %	31.8 %	12.5 %	10.2 %	6.8 %	(22)
Light	51	97	56	38	118	360
High	14.2 %	26.9 %	15.6 %	10.6 %	32.8 %	(34)
Heavy	95	92	29	2	2	220
High	43.1 %	41.8 %	13.2 %	0.9 %	0.9 %	(23)

Sample distributions among classes for the different data sets. The figures in brackets in the “Total” column indicate the number of different grip configurations really tested. This table has been previously published on [25]. Copyright hold by World Scientific Publishing Co.

A prediction function has the form  $F(g) = \bar{\omega}$  where  $g \in G_S$  and  $\bar{\omega} \in \Omega$ . There exists a wide bibliography on the building of such functions based on the Bayesian decision theory [12]. We have chose the approach of the nonparametric techniques, in particular the *voting k-nearest neighbor (KNN) rule* [12, 23], for modeling this function. The nonparametric techniques do not assume any density distribution of the features and the classes. To predict the class of a *query* point  $g_q$ , the KNN rule counts the K-nearest neighbors and chooses the class that most often appears, the most voted. The contribution of each of the KNN points is weighted according to its distance to the query point. This gives more importance to the closer points. The kernel function used is  $K(d) = \frac{1}{1+(d/T)}$ , where T is an adjustable parameter, and  $d$  is the distance.

We define  $\text{KNN}(g_q) = \{(g_i, \omega_i), i = 1 \dots k, g_i \in G_S, \omega_i \in \Omega\}$  as the  $k$  closest points to  $g_q$  and  $d_i$  their corresponding distances from  $g_q$ . The probability corresponding to a class  $\bar{\omega}$  are computed using this expression:

$$p(\bar{\omega}, g_q) = \sum_{g_i \in \text{KNN}(g_q)} \frac{K(d_i)}{\sum_{g_j \in \text{KNN}(g_q)} K(d_j)} \quad (4.1)$$

$$\omega_i = \bar{\omega}$$

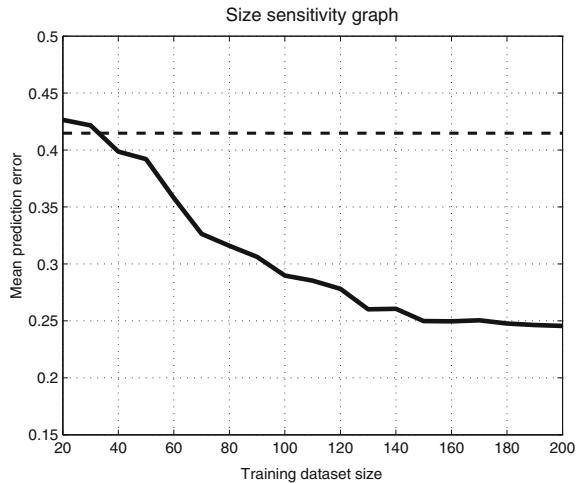
Function  $p$  is also an expression of the posterior probability [23]. To conclude, our predictor would be formally defined by the expression

$$F(g_q) = \text{argmax}_{\omega \in \Omega} \{p(\omega, g_q)\} \quad (4.2)$$

That is, the class predicted  $\omega$  is the one with the largest probability  $p(\omega, g_q)$ .

In order to validate the method a *n-fold cross validation* analysis is carried out. This consists of the following steps: (1) given the whole data set, remove all the points of a particular grasp (similar hand configuration) and use this subset as validation set; (2) use the remaining samples for predicting the outcomes of the validation set and compute the mean error; and (3) repeat steps (1) and (2) for all configurations.

**Fig. 4.2** Evolution of the error when the size of the available data set varies. The *Solid black line* represents the errors obtained by the KNN prediction method, while the *dashed line* is the threshold of the random error. Figure previously published on [24] by IOS Press



The error metric is based on the concept of *misclassification error distance*. The distance between two consecutive classes is defined as 1. In this way define the error distance  $e(g_q) = \{0, \dots, 4\}$  for the prediction of a given query grip. Given a set of predictions  $G = \{g_i, i = 1 \dots n\}$ , we define the average error metric  $\bar{e}(G) = \sum e(g_i)/4$ .

In order to asses sensitivity of the error with respect to the size of the training data. Instead of using the whole remaining data set, we chose randomly a subset of given size. Figure 4.2 shows the results of this experiment. We compare these results against the theoretical figures that would be obtained by a prediction method that would have chosen *randomly* the predicted class.

This method was exhaustively compared with an Artificial Neural Network approach (ANN), and with a analytical bayesian approach. It outperformed them in most of the cases [25].

#### 4.4 Evaluation with Force Torque and Tactile Sensing

Force feedback represents a fundamental requirement for the success of any task that involves physical interaction with the environment. When combined with the touch sense, dexterous manipulation can be performed, even when deprived of visual information (indeed, blindness does not impede dexterity in humans). If tactile feedback is also removed, the use of force information allows humans to robustly perform manipulation tasks which do not require special dexterity, such as performing compliant motion.

In this section, we show two different use cases of tactile-force sensing in the context of manipulation and grasping.

#### 4.4.1 Force-Tactile Physical Interaction for Book Grasping

Consider the problem of extracting a book from a shelf, while standing among other books. The approach is to do it as humans do: one finger is placed on the top of the target book and used to make contact and pull back the book, making it tilt with respect to the base. We consider a Barrett Hand with tactile sensors installed at the fingertips, each one containing an array of  $8 \times 5$  cells. The book grasping process can be decomposed into two different physical interaction tasks: one in charge of tilting the book, and another one in charge of the grasping and extraction part. They are specified as follows:

##### 4.4.1.1 Book Tilting Task

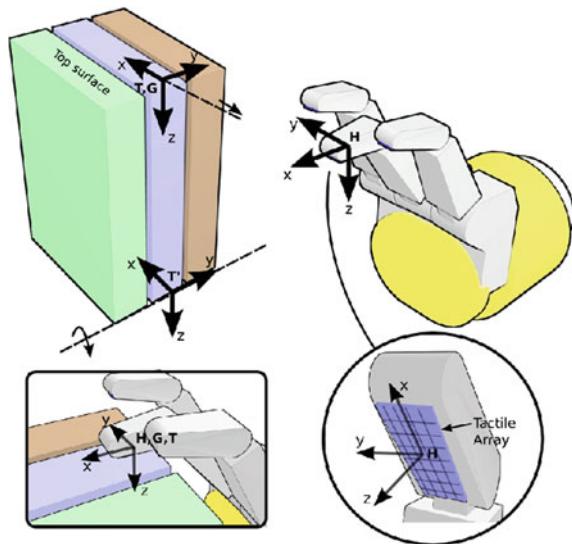
In Fig. 4.3, the specification of the book tilting task, according to our Physical Interaction Framework [33], is shown. It is composed of a hand frame,  $H$ , a grasp frame,  $G$ , and a task frame,  $T$ . There are two possibilities for the task frame in this case. The first is to set it to the book base (frame  $T'$  in Fig. 4.3), so that the task is described as a rotation velocity around this frame. The second possibility is to set the task frame on the top edge of the book (frame  $T$  in Fig. 4.3), so that the task is described as a negative translational velocity along  $X$  direction. We have opted for the second solution, because it avoids the use of the book model. In the first case, the height of the book should be known in order to transform the task velocity from the task frame to the hand frame. By adopting the second solution, we make the approach general for any book size. Two references are set in the task frame, a velocity reference  $v_T^*$ , and a force reference  $f_T^*$ . The first one is set to a negative velocity in  $X$  axis, in order to perform the task motion, whereas  $f_T^*$  is set to a force along  $Z$  axis. This force is needed in order to make enough pressure on the book surface and avoid slip. Therefore,  $Z$  is set to be a force-controlled DOF through a selection matrix  $S_f = \text{diag}(0, 0, 1, 0, 0, 0)$ .

A one-finger precision hand posture is selected, where one of the fingers is slightly more closed than the other ones, so that the hand can easily make contact on the top of the book with one finger, as shown in Fig. 4.3. The hand frame is set to the inner part of the middle finger fingertip, just in the center of the tactile sensor. The fingertip has to make contact on the top of the book. Therefore, the grasp frame is set to the book top surface. The desired relationship between the hand and the grasp frame,  ${}^H\mathbf{M}_G^*$ , is set to the identity.

This task is performed by means of force-tactile control: tactile information is used as feedback to a grasp controller in order to continuously improve the contact between the hand and the book, whereas force feedback is used to perform the task motion, coping with uncertainties and ensuring that a suitable force is performed on the book surface so that there is no slip.

Contact on the top of the book is performed with the tactile array. A qualitative relative pose between the sensor surface and the book can be estimated according to

**Fig. 4.3** The physical interaction frames involved in the book grasping task. The tactile array is used to estimate the relationship between the hand and the grasp frame,  ${}^H\mathbf{M}_G$



the sensor cells that are activated. For example, if there is contact with the upper part of the sensor, but not with the lower part, we can deduce that the sensor plane is tilted around  $Y$  axis with respect to the book top plane. We consider that the finger is completely aligned with the book surface when there are cells activated on each of the four  $XY$  quadrants of the hand frame, i.e., all the tactile sensor surface is in contact.

Taking as input this qualitative description of the relative pose, a tactile-based grasp controller regulates the rotation around  $X$  and  $Y$  axis of the hand frame in order to achieve contact on each of the  $XY$  quadrants of the hand frame. With this approach, the behavior of the robot is completely reactive to the tactile sensor readings. The goal is to keep the sensor plane always parallel to the book top plane, thus ensuring that  ${}^H\mathbf{M}_G = \mathbf{I}_{4 \times 4}$ .

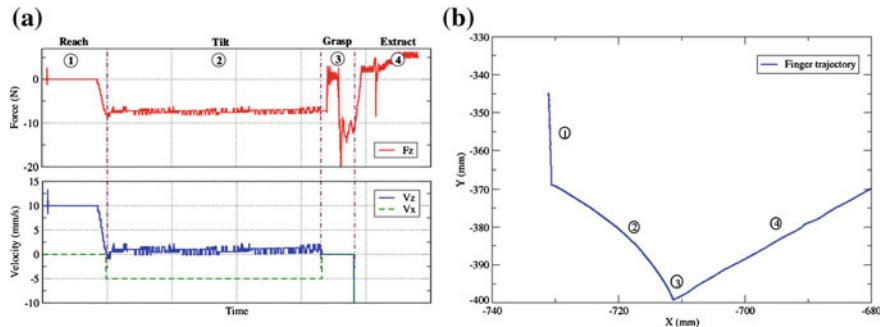
According to the task description, the task motion is performed by moving the hand along negative  $X$  axis of the task frame, while applying a force along  $Z$  axis, i.e.  $\mathbf{f}_T^* = (\mathbf{0}, \mathbf{0}, \mathbf{f}_z^*, \mathbf{0}, \mathbf{0}, \mathbf{0})$  and  $\mathbf{v}_T^* = (-\mathbf{v}_x^*, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})$ . This motion makes the book tilt with respect to the base, as shown in Fig. 4.4. Note that, as the fingertip moves backwards and the book is tilted, the tactile sensor may lose contact with the lower part. This situation is detected by the qualitative pose estimator, and corrected by the grasp controller, so that the hand frame is always aligned with the grasp frame, ensuring that task motion can successfully be transformed to end-effector coordinates.

#### 4.4.1.2 Extraction Task

For the extraction task, the task frame is kept on the top surface of the book, but its rotation is updated in order to be aligned with the horizontal plane. A special



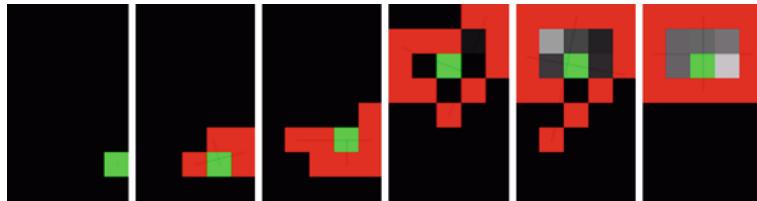
**Fig. 4.4** The robot grasping the book by means of force and tactile-based continuous estimation of hand-to-object relative pose



**Fig. 4.5** Results of the book grasping task. **a** Force and pressure during tilting. **b** Fingertip trajectory, represented in a world frame aligned with the floor

preshape is adopted, called *three-finger spherical* preshape, where the hand frame is kept at the inner part of the middle finger fingertip, as in the previous case. This preshape allows to grasp the book using two fingers, whereas the third one remains in contact with the top of the book. A translational velocity reference is set along the extraction direction, and the interaction task is performed with velocity-force control.

Figure 4.5a represents the forces that appear during the task, and the consequences on the robot's behavior. The first stage corresponds to the initial reaching movement, when the fingertip is moving along the Z direction of the hand frame, searching for the contact. Thus, no forces are present, and the robot moves with a constant velocity along Z axis. When contact is made, the opposite force suddenly increases, until it reaches a reference value of 8N. At this point, the velocity along Z direction is approximately zero, and the robot starts moving along the X direction. This makes the robot perform the rotation movement of the book. In order to avoid the sliding of the book during this stage, the pushing force is kept constant. As the book is tilted, the tactile-based grasp controller updates the finger orientation in order to keep a suitable contact. Figure 4.6 shows a sequence in which the tactile controller establishes a wide contact area starting from a initially weak contact. Step 3 corresponds to the point where the hand adopts a new posture in order to grasp the book. Finally, during the fourth phase, the velocity reference is set to constant values, and the book is actually extracted from the bookshelf. The



**Fig. 4.6** Evolution of the tactile pattern during the book tilting task. *Red pixels* represent the boundary of the contact area, whereas the *green point* is the contact centroid

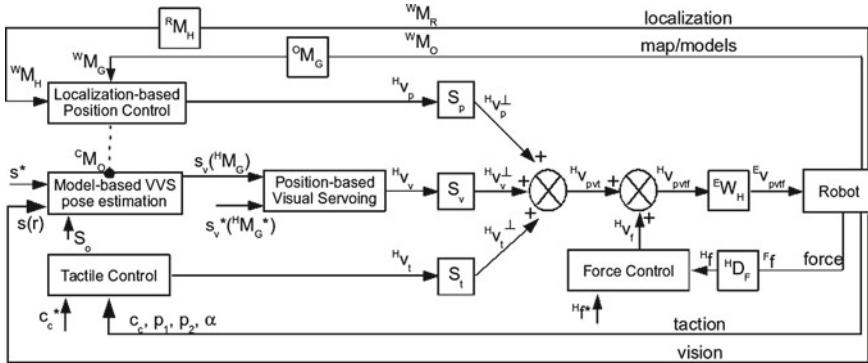
positive force that appears during this step is due to the book's weight. Figure 4.5b shows the trajectory followed by the fingertip during the execution of the whole task, represented within the XY plane of the robot's base coordinate system, which is aligned with the horizontal plane.

## 4.5 Grasp Evaluation Combining Vision, Tactile, and Force Torque

Due to the local nature of the information that tactile sensors provide, it is common to use them in combination with force and vision sensors. Vision-tactile-force combination was already addressed in Allen et al. [1], where some guidelines for detecting useful manipulation events with these sensors were given, without addressing the robot control problem. In Son et al. [40], an approach for combining vision and tactile feedback in a control law was proposed, where force along one direction was also considered, and measured from the tactile sensors. A recent paper, Schmid et al. [37], makes a comparison between tactile-alone, force-alone, and force-tactile integration in the task of opening a door, where vision is used in a previous step in order to estimate the door handle pose.

In this section, we describe a position-vision-tactile hybrid control modified at the low level by a force controller, as shown in Fig. 4.7. In contrast to classical sensor fusion approaches based on kalman filters, hybrid control performs the integration of the sensor signals at the control level, which is more suitable in cases dealing with disparate sensors [29]. In fact, vision, tactile, and force sensors produce fundamentally different quantities, whereas traditional sensor fusion techniques require a common representation among the various sensors being integrated.

In the context of this controller, position control is understood as open-loop motion based on environment information obtained by mobile robot localization algorithms, either based on laser, sonar, odometry, an intelligent environment, etc. According to the hybrid control concept, only one sensor between localization, vision, and tactile sensors is used for a given direction. Our approach is to use the one which provides the most accurate and robust information for that direction.



**Fig. 4.7** Our control approach, integrating position, vision, tactile, and force feedback

Thus, we establish a sensor hierarchy where tactile information is preferred over vision feedback, which is also preferred over localization information. The reason is that tactile sensors provide the most robust and detailed information about the object position, although at the contact level, whereas vision provides more global, but less accurate data, and localization is normally the most inaccurate source.

The cartesian DOF's assigned to each sensor are set on-line by three selection matrices,  $S_p$ ,  $S_v$  and  $S_t$ , which must be mutually orthogonal, i.e.  $S_p \perp S_v \perp S_t \perp S_p$ . If a given cartesian direction must be explicitly controlled by force, it can be set to 0 on all the selection matrices, so that the force controller will fully take charge of it. However, it is worth noting that all the cartesian DOF's are under the effects of a force controller, even if there is no explicit reference for them. This allows to robustly perform physical interaction tasks even if visual or tactile information is not available.

Being  $v_H^p$ ,  $v_H^v$  and  $v_H^t$ , the control velocity computed respectively by the position controller, the vision controller, and the tactile controller, all of them given in a hand frame,  $H$ , then the result of the preliminary sensor integration is given by:

$$v_H^{pvt} = S_p \cdot v_H^p + S_v \cdot v_H^v + S_t \cdot v_H^t \quad (4.3)$$

It is worth mentioning that, in our approach, the selection matrices act on the control velocities, and not on the input errors as in the original hybrid control concept. This is because the tactile and vision errors are not necessarily defined in the cartesian space, and thus, the selection matrices cannot be applied directly on them. Instead, they are applied after the corresponding controllers, where all the control signals are given in a common frame. Note that this is the common practice in hybrid vision-force control approaches [28]. This control velocity is then modified by a force controller which acts on all the degrees of freedom, ensuring that any force generated by a misalignment of the controlled frame,  $H$ , with respect to the environment will be kept inside a given range. If  $v_H^f$  is the hand

velocity computed by the force controller, the final velocity signal, given in the robot end-effector frame can be computed as:

$$\mathbf{v}_E^{pvtf} = {}^E\mathbf{W}_H \cdot (\mathbf{v}_H^{pvt} + \mathbf{v}_H^f) \quad (4.4)$$

where  ${}^E\mathbf{W}_H$  is the twist transformation matrix between the hand frame  $H$ , and the end-effector frame,  $E$ . A suitable cartesian controller must then transform cartesian velocities into joint velocities.

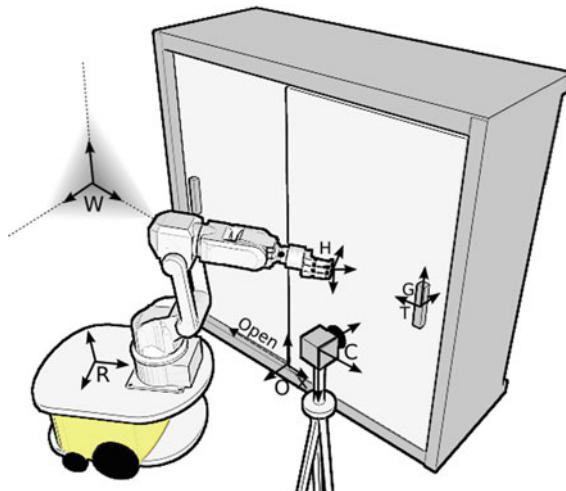
This approach leads to a very natural behavior, where force is the most important sense, followed by tactile, vision, and localization sensors, in this order. Under a blind situation, the task can still be performed by position-tactile-force integration. If tactile feedback is not available, as for example in the phase of reaching an object, then position-vision-force can successfully guide the hand. In the worst case where tactile and vision are unavailable, position-force can still be used.

#### 4.5.1 An Example on Door Opening

We consider the task of opening a cabinet door of sliding type over 25 cm. The robot is manually moved in front of a cabinet as shown in Fig. 4.8. The camera is placed in order to get a partial view of the cabinet door, and a coarse estimation of the homogeneous matrix describing the relationship between the camera frame and the robot base frame,  ${}^R\mathbf{M}_C$ , is previously calibrated by attaching a pattern to the robot hand and computing its pose with the Dementhon algorithm [10], and then making use of the robot kinematic model. Note that this step would not be necessary in a humanoid system, for example, where the eye-to-hand relationship can be approximately computed through robot kinematics. The door pose in the camera frame,  ${}^C\mathbf{M}_O$ , is coarsely calibrated in our case, although it could be also computed from robot laser and sonar-based localization algorithms.

We assume that the cabinet has been previously recognized, and that a model of the door is available, for planning and vision-based tracking purposes. It is also assumed that the task has been previously planned or manually specified, leading to (see also Figs. 4.8 and 4.11):

- A task-oriented hand preshape suitable for performing the particular grasp and task. In this case, a hook precision preshape is planned.
- A hand frame,  $H$ , associated with the task-oriented hand preshape, attached to the part of the hand used for the grasp, and known with respect to the manipulator end-effector frame through hand kinematics. This homogeneous transformation is denoted by  ${}^E\mathbf{M}_H$ .
- A grasp frame,  $G$ , attached to the part of the object where the hand must be moved to, and expressed with respect to the object frame,  $O$ , through the homogeneous transformation matrix  ${}^O\mathbf{M}_G$ .  ${}^H\mathbf{M}_G^*$  and  $\mathbf{S}_e$  are set to the identity matrix so that a rigid hand-handle relative pose is desired.



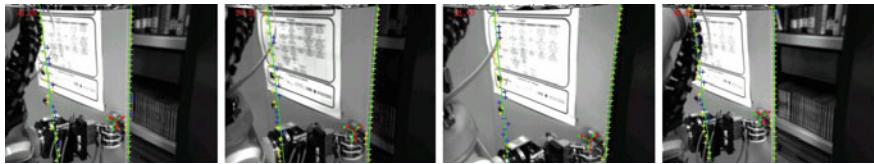
**Fig. 4.8** We consider a mobile manipulator which has to perform a physical interaction task with an object. The mobile manipulator is located at frame  $R$  given with respect to the world frame,  $W$ . The camera is at frame  $C$ , and calibrated with respect to  $R$ . The object frame,  $O$  is supposed to be known in world coordinates. Finally, the physical interaction frames,  $H$ ,  $G$  and  $T$  are computed by the physical interaction planner, and given with respect to  $R$  and  $O$  respectively

- A task frame,  $T$ , placed at the same position that the grasp frame, so that the interaction task can be specified as a desired velocity along  $Z$  axis,  $\mathbf{v}_T^* = (\mathbf{0}, \mathbf{0}, v_z^*, \mathbf{0}, \mathbf{0}, \mathbf{0})$ . Therefore,  $\mathbf{S}_T = \mathbf{0}$ .

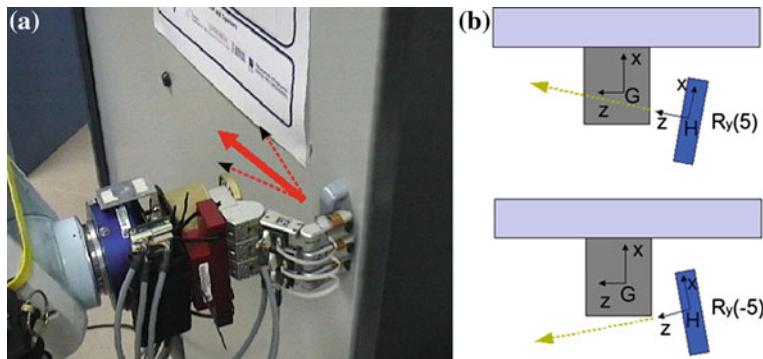
The vision controller is based on the pose estimation given by the Virtual Visual Servoing (VVS) method [22, 41] using the natural object edges. However, only the left and right edges of the door are visible from the camera position, leading to a feature set which is not rich enough for getting a full rank interaction matrix. Therefore, the pose estimation is unable to converge to the real pose on some directions, and, thus, the vision controller cannot reliably control these directions (Fig. 4.9).

Similarly, the hook precision nonprehensile grasp only finds position constraints along the frontal direction ( $X$  axis of the hand frame), and the opening direction ( $Z$  axis of the hand frame). As the rest of directions are not position-constrained, misalignments on these directions do not generate external forces, and, thus, cannot be detected and controlled with force feedback.

Tactile sensors can take control of these directions when the hand is in contact. In order to study the benefits that tactile sensors provide, the door opening task has been reproduced several times, with three different sensor combinations and manually set rotational errors in the initial estimation of the object pose, simulating localization errors. Note that these errors are added to those already existing due to the poor calibration of the initial camera-robot and camera-object transformation.



**Fig. 4.9** The vision effects of introducing an error around Y axis in the pose of the grasp frame. VVS is able to track the articulated pose, but errors on the rest of directions cannot be corrected (note that the left edge estimation does not correspond to the real one)

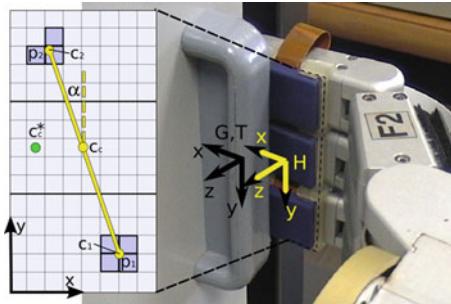


**Fig. 4.10** The task is to push open a sliding door under manually introduced errors in the initial estimation of the grasp link. For a rotational positive error around Y axis of the grasp frame,  $G$ , the hand motion has a positive component along the X axis of the real grasp frame, which finally leads to a frontal contact if not corrected. For a negative error, the robot pushes along a direction which has a negative component along the X axis of the real grasp frame

In particular, an error of  $\pm 5^\circ$  has been forced on each of the axis of the grasp frame,  $G$ , so that there is an important misalignment between the real orientation of the handle, and the one that the robot estimates (see Fig. 4.10). For each error (positive and negative), on each axis, the task has been executed, first by using only the force sensor, then adding the vision modality, and finally by a combination of vision, force, and tactile sensors. Therefore, a total of 18 trials have been performed, 6 for force-alone, 6 for vision force, and 6 for vision-force-tactile, and a trial was considered as a failure when the robot was unable to open the door over 25 cm.

#### 4.5.2 Tactile Controller

The tactile controller developed for this experiment looks for the alignment between the robot fingertips and a planar surface, such as a handle. Although it has been specifically designed for performing alignment tasks, it could be easily adapted for a different purpose, as long as a control velocity for the set of cartesian



**Fig. 4.11** The Barrett Hand in a hook precision preshape, with the tactile sensors installed at the fingertips.  $H$  is the hand frame, and  $G$  denotes the grasp frame. The biggest contact blob on sensors 1 and 2 is selected, and the centroids of these contacts are computed, together with the maximum pressure on each sensor and the angle between the contact line and the vertical

directions which can be robustly and accurately controlled with tactile information is provided. Depending on the sophistication of the tactile sensors, the sensor distribution, the hand configuration, and the task, more or less directions could be controlled.

We consider a Barrett Hand with one tactile array sensor on each fingertip, providing pressure distribution and magnitude information in a  $8 \times 15$  pressure matrix, as shown in Fig. 4.11. First, the biggest contact blob on sensors 1 and 2 is selected and its centroid is computed, giving the points  $\mathbf{c}_1 = (\mathbf{c}_{1x}, \mathbf{c}_{1y})$  and  $\mathbf{c}_2 = (\mathbf{c}_{2x}, \mathbf{c}_{2y})$  in the sensor frame. The maximum pressure sensed on each of the two contact blobs are denoted as  $p_1$  and  $p_2$ . The point  $\mathbf{c}_c = (\mathbf{c}_{cx}, \mathbf{c}_{cy})$  is computed as the middle point between  $\mathbf{c}_1$  and  $\mathbf{c}_2$ . Finally,  $\alpha$  is computed as the angle between the line joining  $\mathbf{c}_1$  and  $\mathbf{c}_2$  and the vertical.

Three cartesian DOF's at the hand frame ( $H$  in Fig. 4.11) are controlled in order to accomplish three goals:

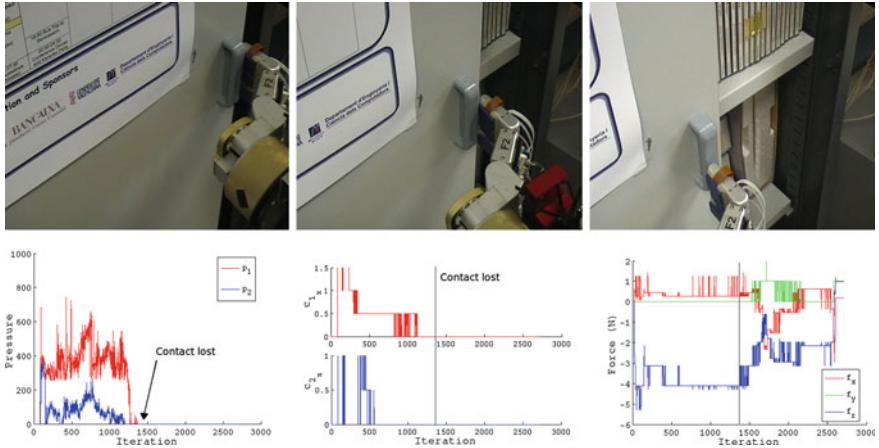
- First, rotation around X axis is controlled in order to guarantee that the pressure is equally distributed between the tactile sensors, thus ensuring that all the tactile sensors keep the contact:

$$w_x = k_p(p_2 - p_1) \quad (4.5)$$

- Second, rotation around Z axis is also controlled in order to regulate  $\alpha$  to zero. The goal is to be aligned with the handle.

$$w_z = k_z\alpha \quad (4.6)$$

- Finally, translation along X axis is controlled in order to bring the point  $\mathbf{c}_c$  toward a reference  $\mathbf{c}_c^* = (\mathbf{c}_{c_x}^*, \mathbf{c}_{c_y}^*)$ , which indicates the part of the tactile sensor where to keep the contact:



**Fig. 4.12** Vision-force performance for an error of  $-5^\circ$  in Y axis of the handle pose. *Top row*: three snapshots of the interaction task, where it is shown how contact is lost during execution. *Bottom row* from left to right: pressures at the fingertips ( $p_1$  and  $p_2$ ), X component of the contact centroids ( $c_{1x}$  and  $c_{2x}$ ) and forces in the hand frame ( $\mathbf{f}_H$ )

$$v_x = -k_c \left( \mathbf{c}_{\mathbf{c}_x} - \mathbf{c}_{\mathbf{c}_x}^* \right) \quad (4.7)$$

$k_p$ ,  $k_x$  and  $k_c$  are the control gains for each controlled direction. The velocity on the rest of directions is set to zero:

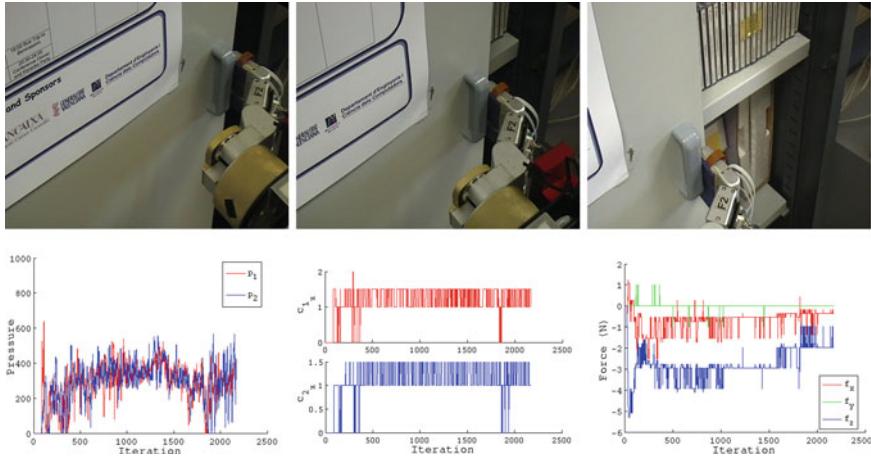
$$\mathbf{v}_H^t = (\mathbf{v}_x, \mathbf{0}, \mathbf{0}, \mathbf{w}_x, \mathbf{0}, \mathbf{w}_z) \quad (4.8)$$

The selection matrix for the tactile controller is set to  $\mathbf{S}_t = \text{diag}(\mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1})$  for this particular case. In the cases where tactile information is not available, such as in the phase of reaching,  $\mathbf{S}_t$  can be set to zero so that the hand is controlled by position-vision-force integration.

#### 4.5.2.1 Results

Simple force control succeeded in only 3 out of 6 trials, whereas vision force completed the task in 5 experiments, and vision-force-tactile performed well in all the 6 cases. In addition, the vision-force-tactile combination was the only one able to avoid undesired forces in directions other than the task direction.

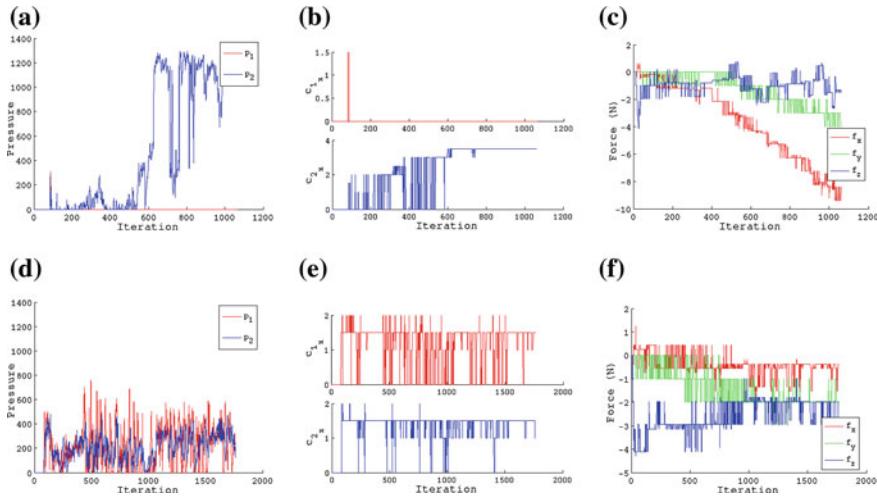
Detailed results for the interesting case of a rotation error around Y axis are shown in Figs. 4.12, 4.13 and 4.14. In the case of Figs. 4.12 and 4.13, the introduced error is manifested in a misalignment that makes the robot push along a direction which has a negative component along the X axis of the real grasp frame,



**Fig. 4.13** Vision-tactile-force performance for an error of  $-5^\circ$  in Y axis of the handle pose. Pressure is balanced and contact is kept until the end of the task. *Top row* three snapshots of the interaction task. *Bottom row* from left to right: pressures at the fingertips ( $p_1$  and  $p_2$ ), X component of the contact centroids ( $c_{1x}$  and  $c_{2x}$ ) and forces in the hand frame ( $\mathbf{f}_H$ )

$G$ , as shown in Fig. 4.10b. As this direction is not position-constrained, force is not able to detect the misalignment. Similarly, the vision part has partial information and can only track the task DOF. Thus, the articulated pose estimation still contains the initialization error. An opening strategy using only vision and force sensors would easily lose contact, as shown in Fig. 4.12. However, the vision-tactile-force approach is able to perceive contact information, and controls the robot so that a contact is always present at the desired location in the fingertip (Fig. 4.13). The force disturbances in the case of vision force are due to a frontal contact of the finger with the handle, appearing at the moment of losing the contact, which generates a small frontal force.

It is also worth noting that vision-tactile-force is able to balance the pressure between the fingertips, whereas vision force is not able to detect and control this issue. This is clearly shown in Fig. 4.14, which illustrates the case of a positive rotation error around Y axis in the localization of frame  $G$ . In this case, the pushing direction has a small positive component in X axis, which slowly drives the fingertip toward the door, as shown in Fig. 4.10. If only vision sensors were considered, frontal collision could not be detected, causing important damage to the robot. Figure 4.14 shows the behavior of vision force and vision-force-tactile control in this case. Note that, under vision force, there is contact only with one fingertip since the very beginning (Fig. 4.14a, b), and vision force is not able to correct this misalignment. Consequently, the whole task force is made by only one finger, which has to support a high pressure, increasing the risk of sensor or mechanics damage. Vision-force-tactile, however, is able to balance the pressure,



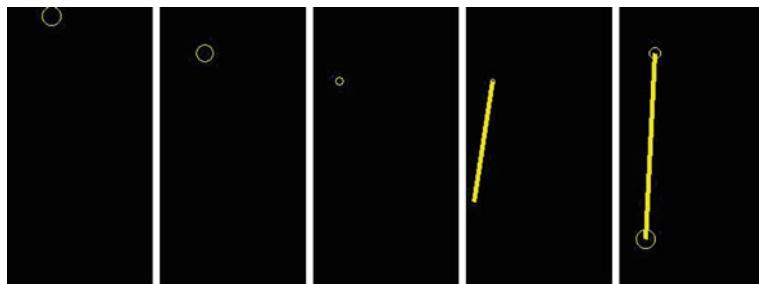
**Fig. 4.14** Results for an error of  $+5^\circ$  in Y axis of the handle pose. Vision-force control is not able to completely align the hand. Vision-force-tactile aligns the hand successfully and distributes the pressure between the fingertips. *Top row*: vision force; *Bottom row*: vision-tactile-force. From left to right column: pressures at the fingertips ( $p_1$  and  $p_2$ ),  $X$  component of the contact centroids ( $c_{1x}$  and  $c_{2x}$ ) and forces in the hand frame ( $\mathbf{f}_H$ )



**Fig. 4.15** A sequence showing the vision-tactile-force alignment process during task execution, starting from a coarse initial position

ensuring contact with all the fingers (Fig. 4.14d, e). Even if vision force is finally able to complete the task, note that, as a consequence of the initial introduced error, the fingertip finally makes frontal contact with the door, leading to a high force in the frontal direction that exists almost from the beginning of the execution (Fig. 4.14c). As expected, vision-force-tactile avoids this situation, keeping the pressure level, contact position, and forces inside a normal range.

Figure 4.15 shows a sequence of the vision-tactile-force execution, whereas Fig. 4.16 shows the corresponding *tactile image*. It can be appreciated how, starting from an initial position with important alignment errors, vision-tactile-force integration is able to correct them and converge to a reliable and safe configuration where all the fingertips are in contact and aligned with the handle.



**Fig. 4.16** A visual representation of the evolution of the contacts during the alignment task. A *circle* represents a point contact. The radius of the *circle* indicates the pressure magnitude. Two point contacts generate a line contact

## 4.6 Conclusion

The evaluation of a grasping action, and more generally of a manipulation action, is a fundamental skill in order to predict and assess the performance of the robot execution. Traditionally, this evaluation has been addressed in the grasp planning stage by the use of the *force* and *form-closure conditions* and the definitions of multiple quality metrics. However this approach has several limitations. In the first place, these analyzes are predictions which do not take into account the actual results of the execution of the actions. On the second place, these procedures can hardly take into account the inherent uncertainty if nonindustrial unstructured scenarios and applications.

The hypothesis defended in this chapter is that information provided by sensors offers a powerful mean to evaluate accurately the performance of grasping and manipulation actions. This assertion is supported by a series of experiments and implementations carried out by the authors. On the first place, it is shown how data-driven performance predictions can be learned and improved from visual features through an exhaustive testing and evaluation of grasping actions. On a second set of experiments it is shown how different modalities of contact, and later vision, can be combined to track and evaluate the correct execution of several manipulation tasks.

A second goal of this chapter is to analyze the potential uses of tactile sensors in the control of robot grasping. First, the different available technologies have been reviewed, and, second, their role in manipulation tasks have been illustrated. There has been a long-term promise that tactile sensing can be of great benefit for the realm of robot grasping. However, only in the recent years, the technological advances have been able to provide useful and practical designs of tactile sensors. It is reasonable to expect that in the coming years a great impulse will happen toward more robust and capable manipulation systems.

**Acknowledgments** This chapter describes research carried out at the Robotic Intelligence Laboratory of Universitat Jaume I. The research leading to these results has received funding

from the European Community's Seventh Framework Programme FP7/2007–2013 under grant agreement ICT-215821 (GRASP project).

## References

1. Allen PK, Miller A, Oh PY, Leibowitz B (1999) Integration of vision, force and tactile sensing for grasping. *Int J Intell Mach* 4(1):129–149
2. ATI Industrial Automation inc. <http://www.ati-ia.com/>
3. Barrett Technology Inc. <http://www.barrett.com/robot/products-hand.htm>
4. Bicchi A (2000) Hand for dexterous manipulation and robust grasping: a difficult road toward simplicity. *IEEE Trans Robot Autom* 16(6):652–662
5. Bicchi A, Kumar V (2000) Robotic grasping and contact: a review. In: IEEE international conference on robotics and automation, San Francisco, CA, pp 348–353
6. Bicchi A, Salisbury JK, Dario P (1989) Augmentation of grasp robustness using intrinsic tactile sensing. In: IEEE international conference on robotics and automation, vol 1. May 1989, pp 302–307
7. Chinellato E, Morales A, Fisher RB, del Pobil AP (2005) Visual quality measures for characterizing planar robot grasps. *IEEE Trans Syst Man Cybern* 35(1):30–41
8. Cutkosky MR, Hyde JM (1993) Manipulation control with dynamic tactile sensing. In: 6th international symposium on robotics research, Hidden Valley, Pennsylvania, Oct 1993
9. Dario D, De Rossi D (1985) Tactile sensors and the gripping challenge. *IEEE Spectr* 5(22):46–52
10. Dementhon D, Davis LS (1995) Model-based object pose in 25 lines of code. *Int J Comput Vis* 15(1/2):123–141
11. Dollar AM, Howe RD (2008) Simple, reliable robotic grasping for human environments. In: IEEE international conference on technologies for practical robot applications(TePRA 2008), pp 156–161, Nov 2008. doi:[10.1109/TEPRA.2008.4686691](https://doi.org/10.1109/TEPRA.2008.4686691)
12. Duda RO, Hart PE, Stork DG (2001) Pattern classification, 2nd edn. Wiley, New York
13. Fastenica s.r.l. <http://www.skilsens.com/index.html>
14. Felipe J, Morales A (2009) Robust sensor-based grasp primitive for a three-finger robot hand. In: IEEE/RSJ international conference on intelligent robots and systems(IROS 2009), pp 1811–1816, Oct 2009. doi:[10.1109/IROS.2009.5354760](https://doi.org/10.1109/IROS.2009.5354760)
15. Fraunhofer Institute. [http://www.iff.fraunhofer.de/en/iffdben/Produkt\\_detail.php?ProduktId=128](http://www.iff.fraunhofer.de/en/iffdben/Produkt_detail.php?ProduktId=128)
16. Howe RD, Cutkosky MR (1992) Touch sensing for robotic manipulation and recognition. MIT Press, Cambridge. ISBN 0-262-11171-3
17. Howe RD (1994) Tactile sensing and control of robotic manipulation. *J Adv Robot* 8(3):245–261
18. Hsiao K, Nangeroni P, Huber M, Saxena A, Ng AY (2009) Reactive grasping using optical proximity sensors. In: IEEE international conference on robotics and automation, Kobe, Japan, May 2009
19. Huber M, Grupen RA (2002) Robust finger gaits from closed-loop controllers. *IEEE/RSJ Int Conf Intell Robots Syst* 2:1578–1584. doi:[10.1109/IRDS.2002.1043980](https://doi.org/10.1109/IRDS.2002.1043980)
20. Interlink Electronics. <http://www.interlinkelec.com/>
21. JR3 inc. <http://www.jr3.com/>
22. Marchand E, Chaumette F (2002) Virtual visual servoing: a framework for real-time augmented reality. In: EUROGRAPHICS 2002, vol 21(3). Saarebrücken, Germany, pp 289–298
23. Mitchell TM (1997) Machine Learning. McGraw Hill, New York
24. Morales A, Chinellato E, Fagg AH, del Pobil AP (2004a) Active learning for robot manipulation. In: European conference on artificial intelligence (ECAI), Valencia, Spain, Aug 2004

25. Morales A, Chinellato E, Fagg AH, del Pobil AP (2004b) Experimental prediction of the performance of grasps tasks from visual features. *Int J Humanoid Robots* 1(4):671–691
26. Morales A, Sanz PJ, del Pobil AP (2002a) Vision-based computation of three-finger grasps on unknown planar objects. In: IEEE/RSJ international conference on intelligent robots and systems. Lausanne, Switzerland, Oct 2002, pp 1693–1698
27. Morales A, Sanz PJ, del Pobil AP, Fagg AH (2002b) An experiment in constraining vision-based finger contact selection with gripper geometry. In: IEEE/RSJ international conference on intelligent robots and systems. Lausanne, Switzerland, Oct 2002, pp 1711–1716
28. Nelson BJ, Morrow JD, Khosla PK (1995) Improved force control through visual servoing. In: American control conference, vol 1. 21–23 June 1995, pp 380–386
29. Nelson BJ, Khosla PK (1996) Force and vision resolvability for assimilating disparate sensory feedback. *IEEE Trans Robot Autom* 12(5):714–731
30. Peratech Ltd. <http://www.peratech.com/qtctechnology.php>
31. Petrovskaya A, Khatib O, Thrun S, Ng AY (2006) Bayesian estimation for autonomous object manipulation based on tactile sensors. In: IEEE international conference on robotics and automation. Orlando, FL, USA, May 2006, pp 707–714
32. Prats M, Martinet P, del Pobil AP, Lee S (2008) Robotic execution of everyday tasks by means of external vision/force control. *Intell Serv Robot* 1(3): 253–266, ISSN 1861–2776
33. Prats M, Sanz PJ, del Pobil AP (2010) A framework for compliant physical interaction. *Auton Robots* 28(1):89–111
34. Prattichizzo D, Trinkle JC (2008) Handbook of robotics, chapter Grasping. Springer, Heidelberg, pp 671–700
35. Pressure Profile Systems Inc. <http://www.pressureprofile.com/products-robotouch.php>
36. Bogdan Rusu R, Cousins S (2011) 3D is here: Point Cloud Library (PCL). In: IEEE international conference on robotics and automation (ICRA). Shanghai, China, 9–13 May 2011
37. Schmid AJ, Gorges N, Göger D, Wörn H (2008) Opening a door with a humanoid robot using multi-sensory tactile feedback. In: IEEE international conference on robotics and automation. Pasadena, CA, May 2008, pp 285–291
38. Schunk. <http://www.schunk-modular-robotics.com/left-navigation/service-robotics/components/actuators/robotic-hands/sdh.html>
39. Shadow Robot Company Ltd. <http://www.shadowrobot.com/tactile/overview.shtml>
40. Son JS, Howe RD, Wang J, Hager GD (1996) Preliminary results on grasping with vision and touch. In: IEEE/RSJ international conference on intelligent robots and systems, vol 3. Osaka, Japan, Nov 1996, pp 1068–1075
41. Sorribes JJ, Prats M, Morales A (2010) Visual tracking of a jaw gripper based on articulated 3D models for grasping. In: Proceedings of the IEEE international conference on robotics and automation, Anchorage, USA
42. Syntouch. <http://www.syntouchllc.com/technology.htm>
43. Tegin J, Wikander J (2005) Tactile sensing in intelligent robotic manipulation—a review. *Ind Robot: Int J* 32(1):64–70
44. Tekscan Inc. <http://www.tekscan.com/custom-OEM-force-sensors>
45. Twendy-one. <http://twendyone.com/>
46. Weiss Robotics. <http://www.weiss-robotics.de/>
47. Willow Garage. <http://www.willowgarage.com/pages/pr2/overview>

## **Part II**

# **Mechanism Design**

# Chapter 5

## Industrial Grippers: State-of-the-Art and Main Design Characteristics

**Penisi Osvaldo Hugo**

### 5.1 Introduction

The connecting device between the robot or manipulator and the load or object to be manipulated is known as gripper. Robot grippers are a kind of end-of-arm tooling used to move parts from one location to another.

Generally, a gripper in industry is a specific device that is used only to manipulate one or a few objects of similar shape. When in a repetitive operation [7] the length or weight varies the gripper must be changed.

In human beings, the hand is the typical element for the action of grasping and manipulating objects. It has a very complex structure, with a considerable number of articulations and rigid bars which conduct to a system of 22 degrees of freedom (DOF), including those of the wrist. As it is an element with multiple degrees of freedom it adapts perfectly to grasping and moving objects of different shapes and sizes, thus becoming an optimum universal gripping system.

According to statistical studies, from 60 to 70 % of man's grasping of objects of cylindrical, parallelepiped, and pyramidal shapes is performed with only two fingers [4, 13]. This is why two-finger grasp is well spread in the industrial applications and mainly in [4, 13] automated assembly. Next in percentage importance are the three-finger grasps.

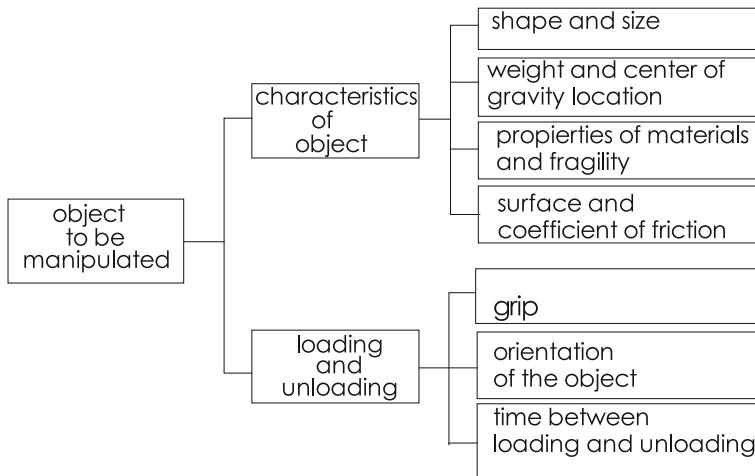
When versatility and ability are necessary, the following items must be taken into account in the design of grippers:

- The mechanism's synthesis;
- The grasping mechanics;
- The optimization of an index contemplating the factors included in the grasps;

---

P. O. Hugo (✉)

IMA: Institute of Applied Mechanics, National University of San Juan,  
Av. Libertador San Martín 1109 oeste 5400 San Juan, Argentina  
e-mail: openisi@unsj.edu.ar



**Fig. 5.1** Characteristics of the object to be manipulated [7]

- The gripper's sensors;
- The object's trajectory.

Firms concerned with the automatization of routine tasks, such as the manipulation of objects produce a wide variety of grippers as far as designs, kinematics chains, and different types of actuators and their functions. There is a great tendency toward the use of linear pneumatic actuators, and in a smaller scale, the rotation pneumatic or electrical actuators.

In the robot industry, grippers are specialized devices used to manipulate one or a few objects of a similar shape, size, and weight in a repetitive operation with minimum requirements of gripping ability and versatility. Thus, it is reasonable to think that not only the shape of the object is important in the gripper's design but also in the trajectory and the task to be performed, as well as the material being manipulated. Therefore, the chart of Fig. 5.1 indicates the characteristics to be considered in the gripper's design, which are the functions of the object to be manipulated, such as is introduced in [7].

In Fig. 5.1, the object to be manipulated must be considered from two fundamental aspects. First, from its geometric and physical characteristics and its mechanical properties and second, from its operative characteristics, as the grip, the space orientation and its manipulation time. The shape and size give information to define the grasp and size of the gripper. The weight and location of the gravity center provides information about which must be the contact point between the fingers and the object and about the minimum force necessary to be able to manipulate the object safely. The knowledge of the surface helps to determine the friction coefficient to be used to consider the necessary grasping force. The grip indicates the way to grasp the object. The orientation of the object indicates the trajectory to be realized with the object to be grasped and placed in the end

position. Time determines the kinematics parameters to be fulfilled. These influence the grasped object's dynamics and, consequently the necessary force to avoid a relative displacement between gripper and object and the damage due to excessive force. Taking this into consideration, we can state that a gripper which is able to manipulate objects of different shape and material can be called a universal gripper. But this is not seen in the different gripper suppliers in the world.

Looking at the gripper from the point-of-view of the mechanical design, a kinematics chain joins the actuator with the gripper's finger, which generally responds to configurations of sliders and bars articulated with roto-translatory movement.

Therefore, there is a wide variety of grippers. For a selection or design, it is necessary to have a classification of them by qualitative and quantitative characteristics.

From a practical point-of-view, there are two principal reasons which justify the classification of gripper mechanisms, one is the designer's possibility to easily find a group of solutions and second the search of design alternatives of the licensed kinematics structures, as stated in [1].

As the important element in the gripper is the finger's movement mechanism, it is necessary to make its detailed analysis.

Nevertheless, we must not forget that the grippers function is to grasp objects safely in the trajectory between the grasping of the object and its placement in the final position. This is why it is necessary to take into account the grasping mechanics and it can be observed that industrial grippers respond correctly to the model proposed by [5]

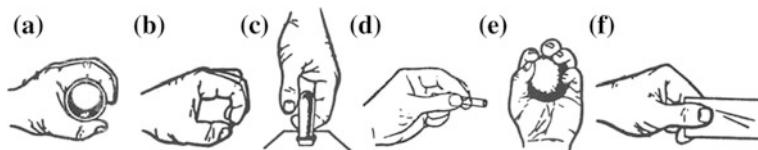
Robot grippers are often selected according to the amount of force that the gripper applies to a part. And this variable is important for a classification and selection of a useful gripper for a specific application. Each gripper mechanism has specific kinematics characteristics given by the size of the bars, the kinematics schemes, and the initial and final angular positions. This group of variables provides a relation between the actuator's force and the gripper's force to grasp the object. This relation is known as "grasping index".

Next, there is an analysis of each of the issues mentioned taking into account the different designs of grippers available in industry, from the point-of-view of the actuation mechanisms of the fingers.

## 5.2 Configurations of Grasping

The aim of grasping an object is to maintain it attached to the gripper in such a way that the set (gripper-object) does not have relative movement in all its trajectory. Thus, the object is not damaged due to the fact that the actuating forces on it do not overpass its mechanic strength.

The analysis of the grasp between the object and the gripper and its balance is known as grasping mechanics. The elimination of the relative movement and the



**Fig. 5.2** Different types of human grasp: **a** cylindrical; **b** between ends; **c** hook; **d** palmary; **e** spherical; **f** lateral

object's stability in the gripper is given by the balance of the actuating forces on the object.

There are different ways of grasping an object and, as it has already been mentioned the universal gripper is the human hand, therefore, it is very interesting to know how a hand grasps the different objects.

The different ways of grasping are mentioned in the bibliography, such as:

cylindrical

lateral

palmary

spherical

between ends

as a hook

Each one of these can be observed in Fig. 5.2.

In effect in everyday life, the selection of the grasping mode is influenced more by the kind of manipulation to be carried out than by the object's shape. Frequently, during the manipulation of an object with a human hand the grasping mode is changed, consciously or intuitively, to obtain the necessary force, coupling, or precision in the movement to be performed.

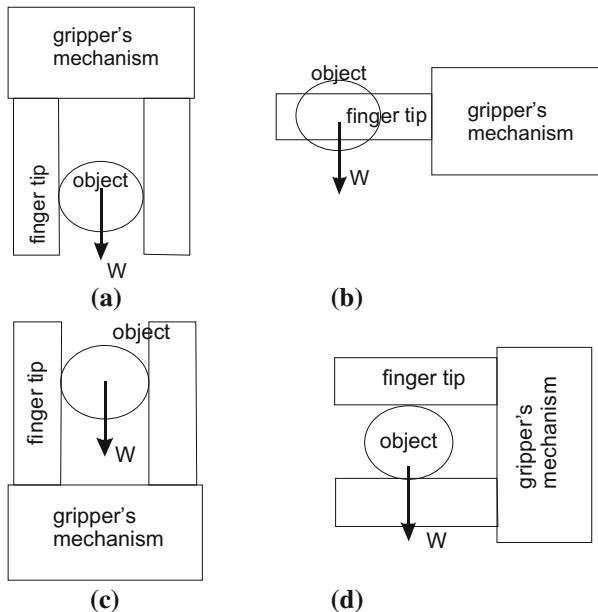
When the grasp must take place between two fingers, it is possible to consider different configurations, according to the relative position between the fingers' hold and the intervening forces. These can be the following [12]:

- Planar grasp, is when all forces actuating on the object are located on the same plane;
- Spatial grasp, is when the forces are located on different planes;
- Palmary grasp, is when the object's weight is directed to the gripper's palm or its ground.
- Grasp with the load on one finger, is when the object's weight is on one finger.

These configurations may be seen in Fig. 5.3 in which  $W$  is the object's weight.

If the object–gripper set is considered in movement, its direction may change the configuration from planar to spatial.

In Fig. 5.4, the distribution of the actuating forces can be seen. These define if the grasp may be considered planar or not. In them, the grasping force is the fingers' force on the object. The movement provides the direction of the inertia



**Fig. 5.3** Example of grasp configurations [12]: **a** planar; **b** spatial; **c** palmary; **d** grasp with load on one finger

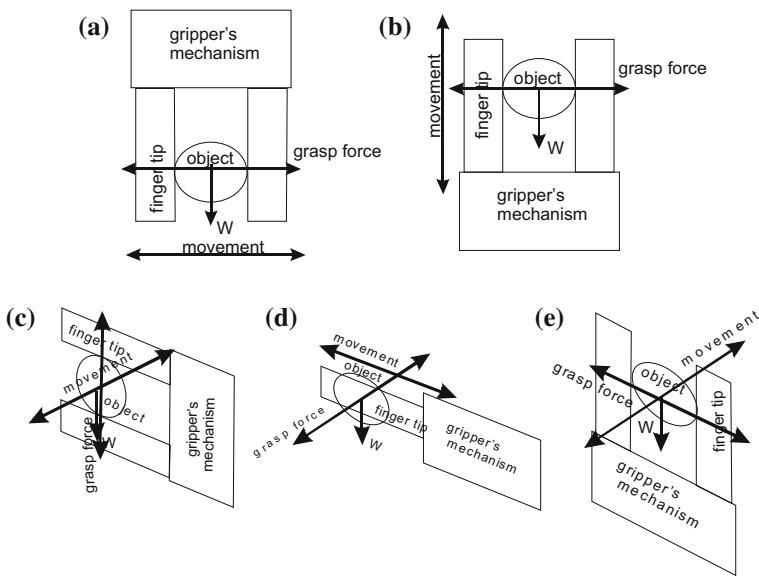
forces due to it. If weight  $W$  and the movement's direction are found on the gripper's plane, the grasp can be considered planar, as in Fig. 5.4a, b.

Any of the two,  $W$  or movement, if they occur outside the gripper's plane make the grasp configuration be considered as spatial. The configuration may change depending on the trajectory described by the set object–gripper. The study of the object's balance in the gripper must be considered fundamentally spatial.

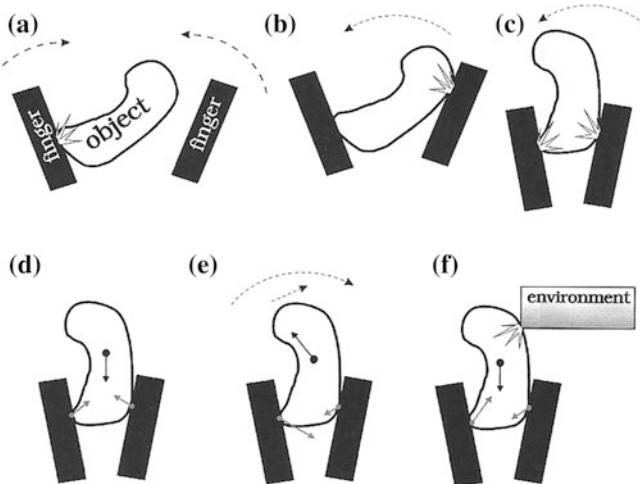
### 5.3 Grasp and its Different Phases

The different phases of grasping objects with two fingers are seen in Fig. 5.5. Here, the fingers' movement is shown as dashed lines and the moments, the external actions on the object, the forces applied by the fingers and the object's weight as continuous lines. These phases have the following characteristics: [5]

- The first phase of initial impact of one finger and the object—closing movement of the gripper; Fig. 5.5a;
- The second phase of impact of the second finger and dragging of the object while the closing movement continues, until a static position of the object in the fingers is reached, due to forces or to geometric links; Fig. 5.5b;

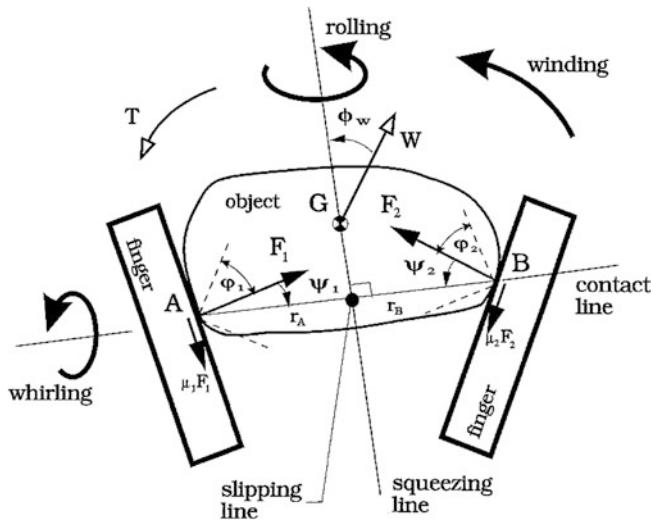


**Fig. 5.4** Examples of spatial grasps [12]: **a, b**, planar grasp; **c, d, e** spatial grasps



**Fig. 5.5** Phases of grasping mechanics, [5]: **a** initial impact; **b** thrust; **c** grasping impact; **d** static grasping; **e** dynamic grasping; **f** grasping with impact against the environment

- (c) The third phase of impact of the final grasp when the gripper's closure is produced and the grasping force is applied, stopping the fingers' movement; Fig. 5.5c;
- (d) The fourth phase defined as static grasp when the object remains grasped in a static equilibrium between the fingers; Fig. 5.5d;



**Fig. 5.6** Model of grasp with two-finger gripper [5]

- (e) The fifth phase defined as dynamic grasp in which the object remains in equilibrium between the fingers while the dynamics of the object's transportation takes place; Fig. 5.5e;
- (f) The sixth phase when an external perturbation of the movement takes place as for example an occasional impact with the environment; Fig. 5.5f.

In order to model these phases, the model in Fig. 5.6 has been proposed in which friction cones are considered between the fingers' contact surfaces and the object [5]. Friction forces must be in equilibrium with all the actuating forces on the object, such as weight, inertia, and external actions with the environment, reaching a static equilibrium of the object in the gripper. This model represents a planar grasp, because all the actuating forces are found on the plane defined by the two forces produced by the fingers.

Figure 5.6 is the model's scheme, where forces such as weight  $P$  is shown, where the actions of the inertia force due to the manipulator's movement and the external actions may be included; as well as the forces produced by fingers  $F_1$  y  $F_2$ ; the external actions represented by the  $T$  pair in which the actions of the moments of inertia due to the manipulator's movement and external movements may be considered. Friction forces generated between the fingers' contact surfaces and the object are indicated with  $\mu_1 F_1$  y  $\mu_2 F_2$  in the direction of the tangent which is common to the contact surfaces, being  $\mu_1$  y  $\mu_2$  the friction coefficients.

Forces  $F_1$  y  $F_2$  are applied by the fingers, generated by the gripper's actuator, transmitted through the mechanism to the finger.

Angles  $\varphi_1$ ,  $\varphi_2$  form forces  $F_1$  y  $F_2$  with the contact line, which is defined by the two contact points A and B between the object and the fingers. Angle  $\varphi_p$  indicates

the P's orientation with respect of the displacement line formed by the perpendicular line to the contact line which runs across the object's barycenter.

Angle  $\psi$  corresponds to the friction cone between the contact surfaces.

The equilibrium condition in the planar grasp for this model is given by the equilibrium of the actuating forces in two directions: that of displacement and that of contact line and the other condition is one of moments with respect of the intersection of these two lines, expressed as:

$$F_1 \sin \varphi_1 + F_2 \sin \varphi_2 + P \cos \varphi_p + \mu_1 F_1 \cos \varphi_1 + \mu_2 F_2 \cos \varphi_2 = 0 \quad (5.1)$$

$$F_1 \cos \varphi_1 + F_2 \cos \varphi_2 + P \sin \varphi_p + \mu_1 F_1 \sin \varphi_1 + \mu_2 F_2 \sin \varphi_2 = 0 \quad (5.2)$$

$$T - r_G P \sin \varphi_p - r_A F_1 (\sin \varphi_1 - \mu_1 \cos \varphi_1) + r_B F_2 (\sin \varphi_2 - \mu_2 \cos \varphi_2) = 0 \quad (5.3)$$

In this last equation,  $r_A$ ,  $r_B$ ,  $r_G$  are the distances between the corresponding points and the centers of moments which have been considered as the intersection points of displacement and the contact lines.

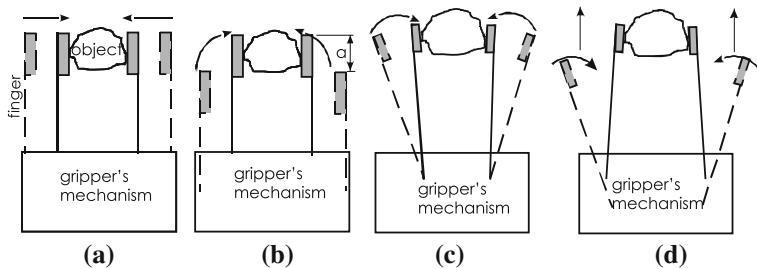
## 5.4 Gripper Mechanisms

Always considering the two-finger grasp, in the previous instant to when the finger contacts the object and the time that passes until the static grasp is produced the trajectory described by the fingers are defined by type of gripper mechanism used to command the fingers. These trajectories are of four types, as for example:

Rectilinear translation,  
Curvilinear translation,  
rotation,  
roto-translation.

These can be observed in the examples of Fig. 5.7 [11]. In Fig. 5.7 a a finger's approximation to the object following a rectilinear trajectory is shown, the finger always being parallel to the original position, without varying its distance to the gripper's frame. In Fig. 5.7b the trajectory is curvilinear. Here, the finger although it is always parallel to the original position, there is a displacement of the finger in the direction indicated in the figure. Figure 5.7c shows an approximation movement toward the object, where the finger describes a circular trajectory around a fixed point of the mechanism. Figure 5.7d shows an approximation of the finger to the object in two movements such as in indicated in the previous.

Each kind of approximation has specific grasp characteristics, which must be taken into account at the moment the gripper grasps the object. These respond to the following:



**Fig. 5.7** Examples of fingers approximation movement [11]: **a** linear translation; **b** curvilinear translation; **c** rotation; **d** roto-translation

- The parallel approximation provides the possibility of grasping by its outer or interior surface parallelepiped, cylindrical objects, or those with parallel faces to both fingers, producing parallel or opposed grasp forces. If the directions of these forces do not coincide, a moment will be produced in the object on the plane defined by the forces. Once a gripper's position has been reached with respect of the object, a finger's point does not vary its distance to the frame, maintaining the relative position finger—object of the line defined by the points of each finger when they come near the object.
- The approximation by curvilinear translation has similar grasp characteristics to the previous one, with the sole difference that one point of each finger varies the position of the line which they define with respect of the predicted grasp line of the object. This makes the grasp location not precise, depending on the gripper's initial location.
- The approximation by rotation also produces a variation in the distance between a point and the theoretical object's line of contact. The grasp forces produce a non null component of them.
- The approximation by roto-translation produces similar effects to those of the approximation by rotation.

In brief, the precision needed in the relative gripper—object positioning will depend on these kinds of movements and on the possible stress and movements existing between them at the grasp moment.

## 5.5 Two-Finger Mechanisms in Industrial Grippers

Generally, the industrial gripper is a specific device used to manipulate only one or a few objects in a similar way, size and weight in a repetitive operation [7]. If only one of these characteristics changes, the gripper must also change.

There is a great variety of grippers which in order to be selected or designed need a classification by certain qualitative and quantitative characteristics.

From the practical point-of-view, there are two main reasons which justify research on the classification of gripper mechanisms, one consists of the designers' possibility of easily finding a group of solutions and second, the search of design alternatives of the licensed kinematic structures as stated in [1].

Several researchers have worked on this line, as for example [2, 4, 7–9, 15]; giving different examples of gripper mechanisms which can fulfill certain functions.

A general qualitative way of classifying grippers is given in [7], depending on the actuating principle of the grasp action as follows:

- mechanical;
- vacuum and magnetic action;
- universal.

The mechanical grippers are those which consist of fingers to grasp the object and the grasp force is that of contact and mechanical friction.

The vacuum grippers consist of vacuum cups and the object's grasp force is produced by suction in them.

The magnetic grippers are those which grasp the object by an electromagnetic force.

The universal grippers respond in general to multiple fingers trying to imitate the human hand, or the fingers are of deformable materials which can accommodate to the objects shape.

Fundamentally, there is an important development of the mechanical grippers due to the great variety of finger mechanisms and shapes to grasp objects of various materials which can be manipulated.

According to the object's shape and by the place where it can be grasped, the grippers may be classified as of:

- Internal action;
- External action.

Those objects which have holes by which they can be grasped are called of internal action. Those objects grasped by the external surface are called of external action.

Taking into account the fingers movement, the grippers may operate with the following movements:

- translation;
- rotation;
- roto-translation.

In the finger's trajectory of approximation to the object, the gripper mechanism can perform it with a parallel displacement to itself or by a rotation with respect to a fixed point of the gripper. Besides, a compound movement (roto-translation) can be used for the finger's movement.

Taking into account the kinematics components of finger conduction mechanisms, the grippers may have specific structures of:

bars;  
pinion–rack;  
cams;  
screw;  
wire and pulley;  
combination of these.

The bar structures may be of two or three points of articulation or sliders, being used for both kinds of finger movements.

The pinion–rack mechanism may be used to obtain a parallel or oscillating finger movement according to which is the input movement.

The cam mechanism in grippers is used for finger movement by translation, generally actuated by a rotation actuator.

When the mechanism consists of screws only, it is for finger movement by translation actuated in general by a rotation actuator, with a screw with right and left filet to command the finger's cube.

Some examples of possible mechanisms for industrial grippers which satisfy this classification are included in those stated [7]. Considering these as the most representative and, in some cases, they are still in use nowadays. They can be seen in Figs. 5.8, 5.9, 5.10, 5.11 and 5.12, which are described as:

Figure 5.8 is a group of possible mechanisms of grippers consisting of only rotation;

Figure 5.9 is a group of possible mechanisms consisting of only rotation and prismatic joints;

Figure 5.10 is a group of possible mechanisms of grippers consisting of pinion and rack joints;

Figure 5.11 is a group of possible mechanisms of grippers consisting cam and follower joints;

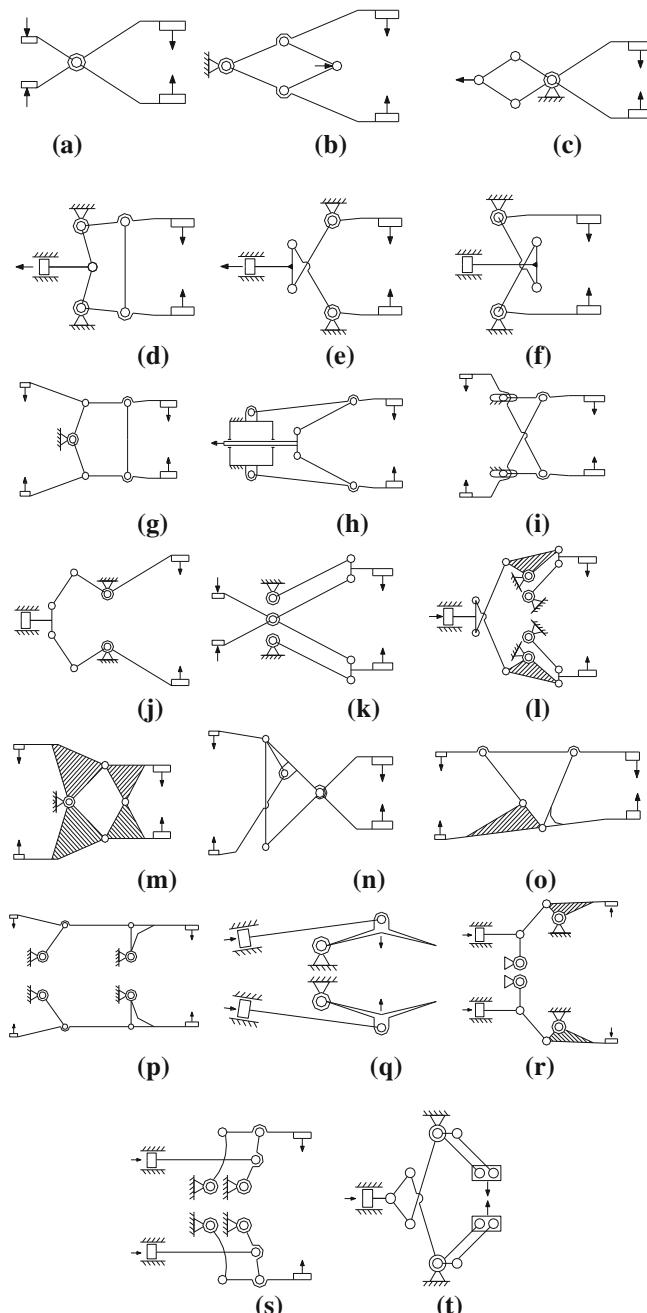
Figure 5.12 is a group of possible mechanisms of grippers actuated by screw mechanisms;

Next, some examples of different kinematics chains of industrial grippers of various origins may be observed.

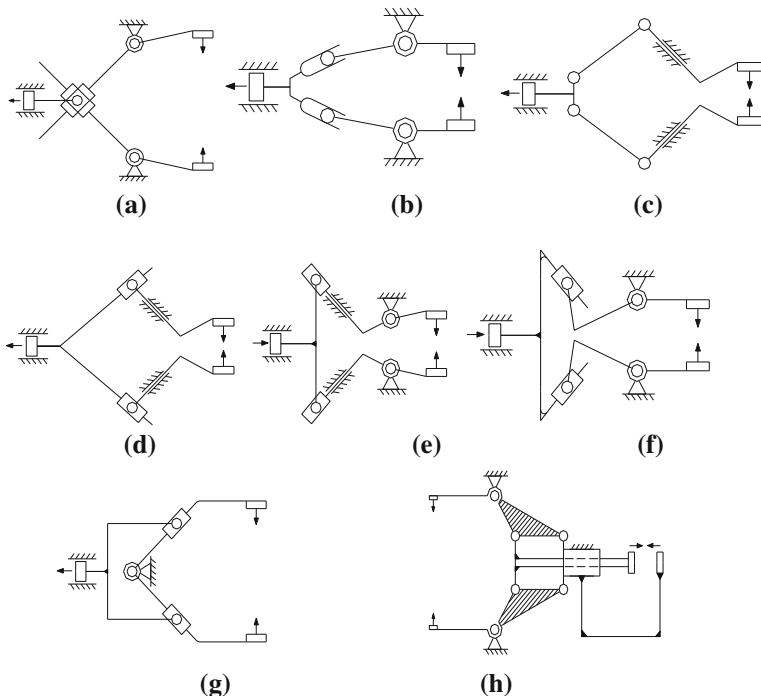
## 5.6 Characteristics and the Grasping Merit Index

To make an evaluation of the different grippers which can be designed and in order to determine which of the designs is more convenient for a specific application, it is necessary to take into account their characteristics and some comparison parameter as regards to how the fundamental gripper's object (secure grasp of the object) is fulfilled.

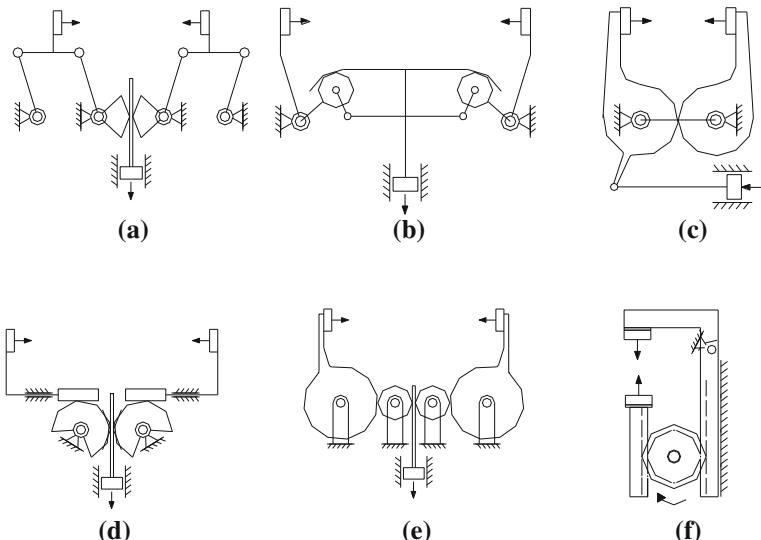
Each gripper mechanism presents specific characteristics, for example among the kinematics characteristics, the following can be named, connection point finger trajectory, link size, connection point finger velocity in different possible grasp



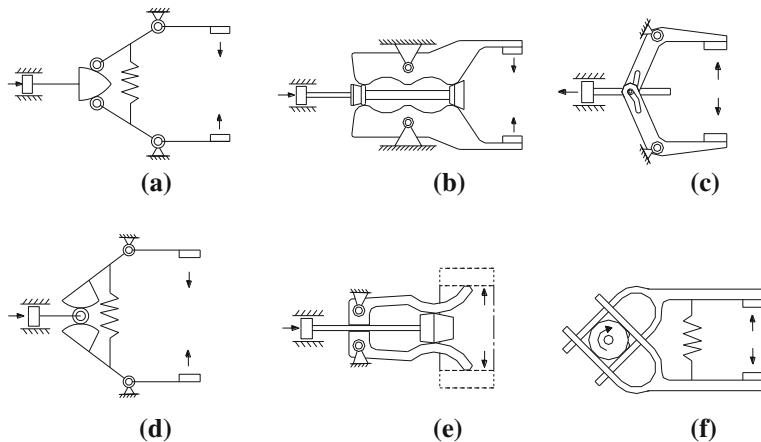
**Fig. 5.8** Kinematics schemes of industrial grippers proposed in [7] (a–t) different mechanisms of grippers consisting of only rotation joints



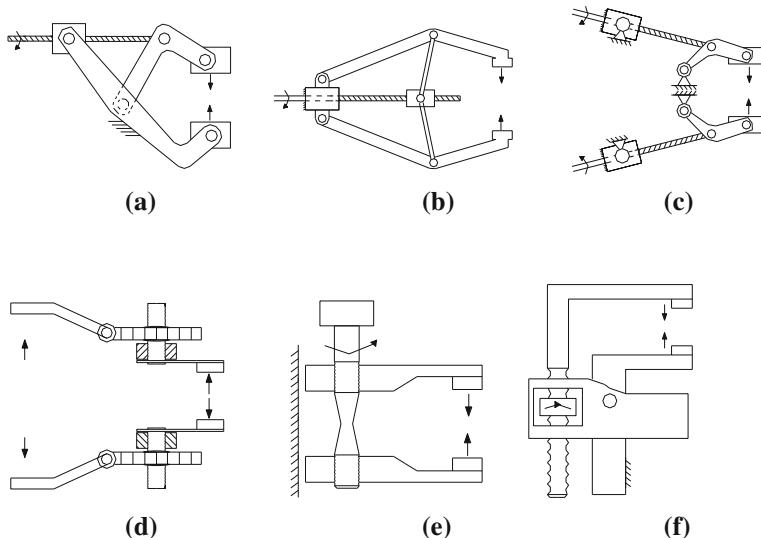
**Fig. 5.9** Kinematics schemes of industrial grippers proposed in [7] (a–h) different mechanisms of grippers consisting of only rotation and prismatic joints



**Fig. 5.10** Kinematics schemes of industrial grippers proposed in [7] (a–f) different mechanisms of grippers consisting of pinion and rack joints



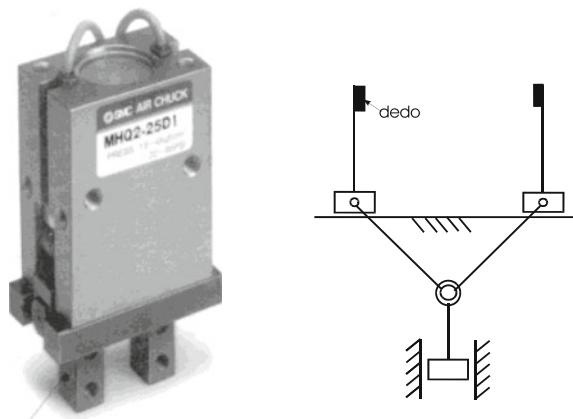
**Fig. 5.11** Kinematics schemes of industrial grippers proposed in [7] (a–f) different mechanisms of grippers consisting cam and followers joints



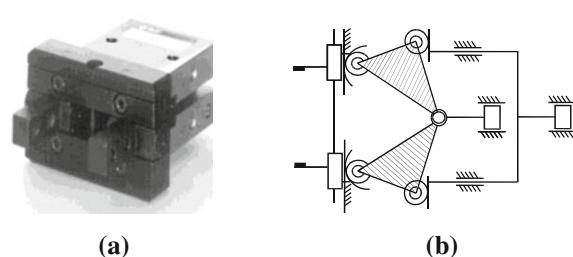
**Fig. 5.12** Kinematics schemes of industrial grippers proposed in [7] (a–f) different mechanisms of grippers actuated by screw mechanisms

positions, accelerations, which are given by the size of their links, the kinematics scheme, and the initial and final angular positions of the mechanism. A parameter which allows a comparison between grippers is called a Merit Index.

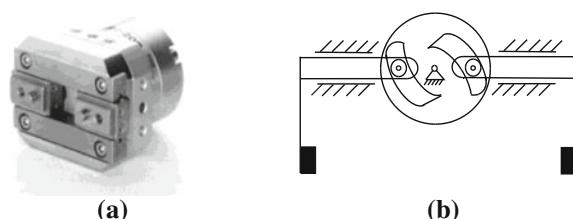
**Fig. 5.13** Example of an industrial gripper: **a** MHQ2-6 commercial model, I 305A Catalog, [14]; **b** kinematics scheme



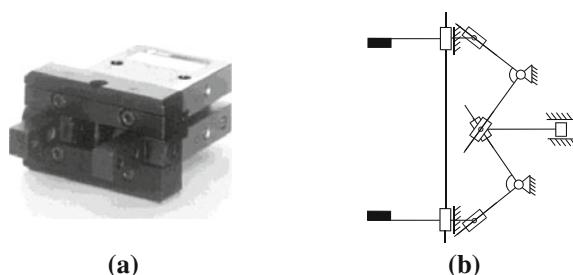
**Fig. 5.14** Example of an industrial gripper: **a** MHQG2-10 commercial model, I 305A Catalog, [14]; **b** kinematics scheme



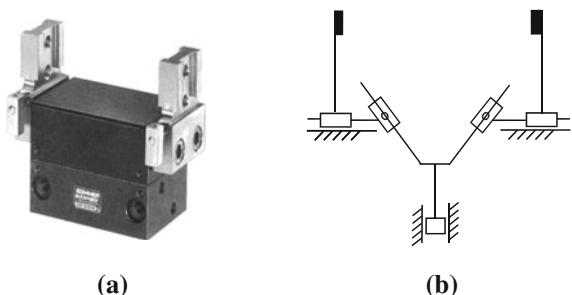
**Fig. 5.15** Example of industrial gripper: **a** MHR2-MDHR2-10 commercial model, I 305A Catalog, [14], **b** kinematics scheme



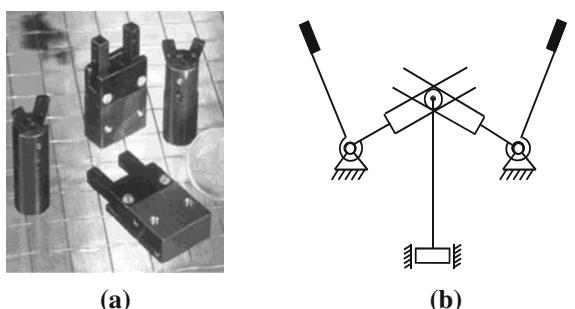
**Fig. 5.16** Example of industrial gripper; **a** MHQG2-32 commercial model, I 305A Catalog, [14]; **b** kinematics scheme



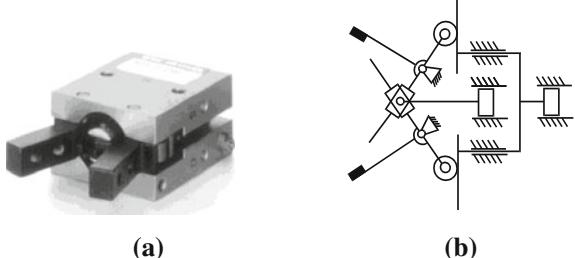
**Fig. 5.17** Example of an industrial gripper: **a** GP210 N commercial model, H920 Catalog, [17]; **b** kinematics scheme



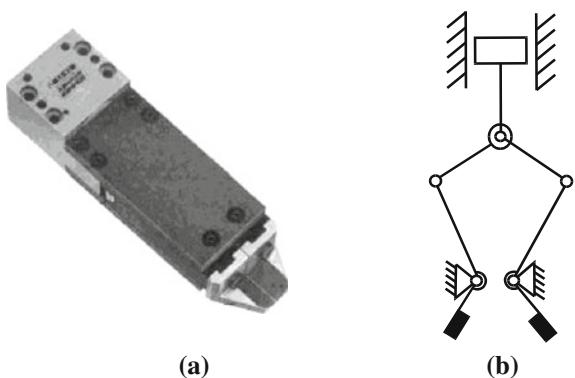
**Fig. 5.18** Example of an industrial gripper: **a** WGW-6 commercial model, 2.0 General Catalog, [18]; **b** kinematics scheme



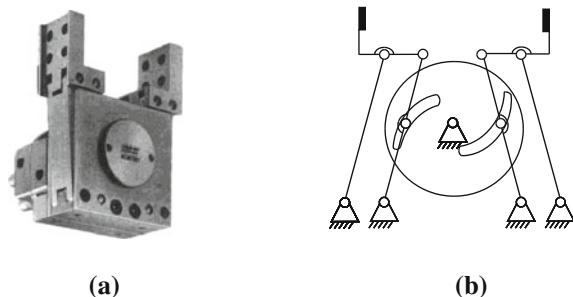
**Fig. 5.19** Example of an industrial gripper: **a** MHC2-10 commercial model, I 305A Catalog, [14]; **b** kinematics scheme



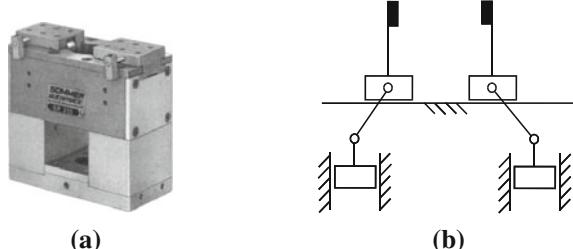
**Fig. 5.20** Example of an industrial gripper: **a** ZK22 commercial model, H920 Catalog, [17]; **b** kinematics scheme



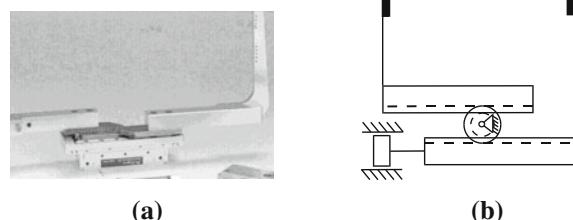
**Fig. 5.21** Example of an industrial gripper: **a** GP500 commercial model, H920 Catalog, [17]; **b** kinematics scheme



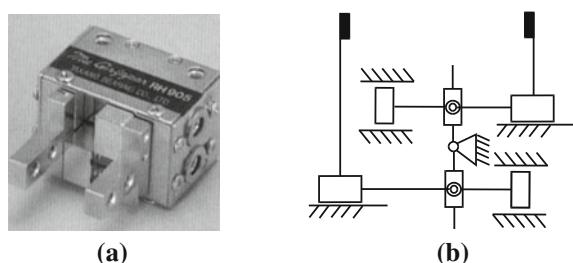
**Fig. 5.22** Example of an industrial gripper: **a** GP325 commercial model, H920 Catalog, [17]; **b** kinematics scheme



**Fig. 5.23** Example of an industrial gripper:  
**a** RH918ST140 commercial model, Robot Hand Catalog, [16]; **b** kinematics scheme



**Fig. 5.24** Example of an industrial gripper: **a** RH901 commercial model, Robot Hand Catalog, [16]; **b** kinematics scheme



## 5.7 Design Parameters for Two-Finger Grasp

When designing an element to grasp an object securely it is important to determine parameters to quantify the grasp. These must be able to consider the effects produced in the different grasp phases, taking into account the type of objects to be

manipulated, as regards shape, size, weights, trajectories, maximum velocities to be reached with the grasped object, maximum energy with which the fingers produce the contact impact with the object, the relative position of the gripper-object in which the grasp must take place.

Each one of these parameters must be quantified and/or controlled.

The relative position between the object and the gripper in the grasp moment will define the area in which the grasp can take place, the shape and size of the object on the grasp plane, as well as the initial movements of the object in the different axles as defined in paragraphs previous will also define which is the position of the object's gravity center with respect of the gripper's plane.

## 5.8 Merit Index in Literature

An important characteristic to define a gripper and to be able to compare with others, or to determine the actuator's size, is its mechanic efficiency. This is defined as the relationship between the necessary force needed by the finger to grasp the object securely and the necessary force in the actuator to achieve this.

A practical way to obtain an evaluation of this relationship is the application of the virtual work principle, such as is proponed in [7].

Taking into account the mechanism's efficiency  $\varepsilon$  and the symbology of Fig. 5.6 a grasping index grasping index (G.I.) may be defined [3] in the gripper as:

$$\text{G.I.} = \frac{F_1 \cos \varphi_1 + F_2 \cos \varphi_2}{P} = \varepsilon_1 \cos \varphi_1 + \varepsilon_2 \cos \varphi_2 \quad (5.4)$$

where:

$P$  is the force provided by the actuator of the mechanism driving the finger

$F_1$  and  $F_2$  are the forces applied by the fingers, when the referent to the model in Fig. 5.6 is made.

$\varepsilon_1$ ;  $\varepsilon_2$  represent the mechanical efficiency of each of the mechanisms which command each finger.

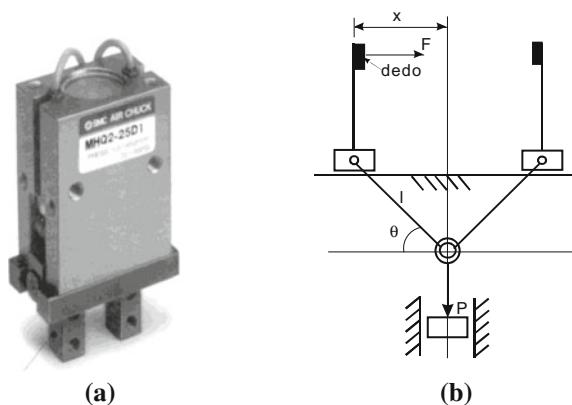
In the case  $F_1 = F_2 = F/2$  corresponding to a symmetric grasp by which  $\varphi_1 = \varphi_2 = \varphi$ , and the actuating mechanism of the fingers is commanded by a sole actuator, G.I. may be formulated as:

$$\text{G.I.} = \frac{F \cos \varphi}{P} = \varepsilon \cos \varphi \quad (5.5)$$

The G.I. index cannot be represented by only one number for each gripper because it is in effect variable for:

- each configuration of the kinematics scheme;
- angles  $\varphi_1$ ,  $\varphi_2$  or else  $\varphi$ .

**Fig. 5.25** Example of industrial gripper: **a** MHQ2-6 commercial model, I305A Catalog, [14]; **b** kinematics scheme



In the G.I. grasp index, the  $\varphi$  angle takes into account considerations of the grasp, of the mechanism, of the kinematics; and the efficiency  $\varepsilon$  considers the static.

The expression in Eq. (5.4) developed by a specific mechanism will depend on the mechanism's geometric parameters. For example for the gripper of Fig. 5.25 the expression is

$$\varphi = 0_y \quad \cos \theta = \frac{x}{l}, \quad \text{G.I.} = \frac{l}{2 \cos \theta} = \frac{l}{2x} \quad (5.6)$$

where  $\theta$  is the coupler's angle with respect of a line of reference parallel to the contact line,  $x$  is the distance between the grasp point and the sliding line which coincides with half the grasped object's size and  $l$  is the coupler's length which joins the actuator with the cube on which the finger is mounted.

As it can be observed, the G.I. index may also be expressed as the relationship between the gripper's mechanical characteristics, as for example, the bar's length and the size of the grasped object.

Efficiency expressions for various gripper mechanisms may be found in [7] with a different formula.

As efficiency depends on the geometric parameters of the mechanism, this is a good expression to use in the gripper design, from the synthetic point-of-view, because what must be considered is the size of the bars and its angles.

On the other hand, grasp kinematics can be thought of as related to the position and orientation of the finger with respect to the grasped object, and a satisfactory grasp is achieved when the capability is conveniently exploited and the finger motion to the grasp does not require manipulator small motion adjustment. In particular, the angular excursion of the finger and the trajectory accuracy of the finger's reference point are significant. The trajectory accuracy can be evaluated with reference to a linear path than can be considered as the optimal finger trajectory on the basis of the human grasp experience.

Moreover, the swinging angle of the input link is important in order to take into account the mechanism's excursion and also for the actuator's feature requirements, which are fundamentally power and motion capabilities.

For the motion aspects of the gripping mechanism, as may be found in [6], a so-called 'Capability Index' (C.I.) can be introduced in the form

$$\text{C.I.} = \frac{L_r}{l_F \varepsilon_{\max}} \cos \Delta\alpha_F \quad (5.7)$$

where:

- L is the gripper's capability under evaluation, which is related to the gripping mechanism between the fully open and fully closed configurations;
- r is the input angle ratio given as

$$r = \frac{\alpha_{1L} - \alpha_{1C}}{\alpha_{1L} + \alpha_{1C}} \quad (5.8)$$

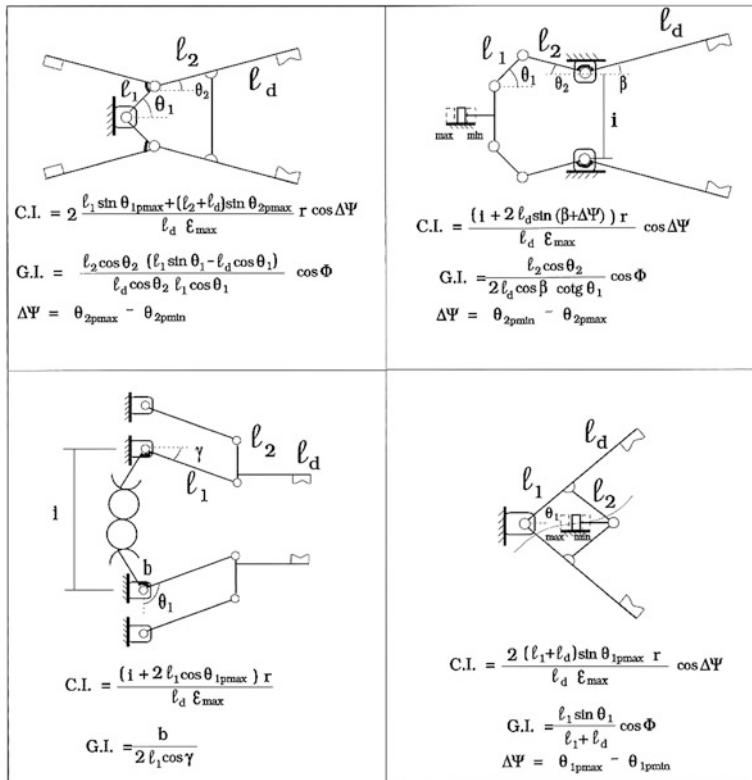
which is the ratio calculated by the angle difference of the gripping mechanism input link in the maximum capability configuration  $\alpha_{1L}$  and the minimum configuration  $\alpha_{1C}$ , and the average angle of the aforementioned input range angle;

- $l_F$  is the finger link length
- $\varepsilon_{\max}$  is the maximum geometric deviation of the trajectory of fingertip point M with respect to the ideal linear path within the permitted finger motion;
- $\Delta\alpha_F$  is the angular excursion of the finger link during the motion from the minimum capability configuration to the maximum capability of the gripping mechanism.

C.I. can be considered as a characteristic of two-finger grippers depending on the gripping mechanism proportions and on the size of the grasped object.

Assuming this C.I. formulation, a gripping mechanism is evaluated as more effective than another one when the C.I. is calculated greater, which may occur when:

- The configuration capability L is greater since the gripping mechanism allows the grasping of larger objects;
- The input angle ratio r is greater since the position control on the actuator can be transmitted with a greater precision on the finger motion;
- The size  $l_F$  of the finger link is smaller, since a smaller and lighter gripper is required in order to reduce as much as possible the manipulator payload and versatility;
- The deviation  $\varepsilon_{\max}$  is smaller, since the linear finger motion can be considered desirable also because no small manipulator adjustment motion is needed, mostly in industrial applications with small pieces grasp.



**Fig. 5.26** Examples of an analytical evaluation of gripping mechanisms through the Capability Index (C.I.) and the Grasping Index (G.I.) [6]

Analytical examples of indexes C.I. and G.I. may be seen in Fig. 5.26 for the gripping mechanisms here illustrated.

## 5.9 Kinematics Characteristics of a Gripping Mechanism

In brief, the kinematics characteristics of a gripping mechanism will be given by:

The size of each one of the gripper's bars;

- Configuration angle of the gripper in open and closed s, path;
- Size of the gripper's field of force to grasp objects which must be manipulated;
- Characteristic angles of the grasp mechanics;
- G. Index expression,
- Kind of actuator,
- Kinematics components,
- Grasp force.

**Table 5.1** Identification typology of the industrial grippers, using the characteristics of the proposed classifications

Figure/gripper model	Identification typology			
	Actuator	Finger's movement	Kinematics components	Kind of joints
Fig. 5.13 MHQ2-6	lin	TP	B	R-P
Fig. 5.14 MHQG2-10	lin	TP	B	R-P-SC
Fig. 5.15 MHR2-10	rot	TP	L	R-P-SC
Fig. 5.16 MHQG2-32	lin	TP	B	R-P
Fig. 5.17 GP210 N	lin	TP	B	P
Fig. 5.18 WGW6	lin	rot	B	R-P
Fig. 5.19 MHC2-10	lin	rot	B;L	R-P-SC
Fig. 5.20 ZK22	lin	rot	B	R
Fig. 5.21 GP500	rot	TD	B;L	R
Fig. 5.22 GP325	lin	TP	B	R-P
Fig. 5.23 RH918ST140	lin	TP	PC	R-SC
Fig. 5.24 RH901	lin	TP	B	R-P

lin: lineal; rot: rotation; TP: translation parallel; TD: translation in two directions; B: barras; PC: pinion and rack; L: cams; R: rotation joints; P: prismatic joints; SC: contact surfaces; A: active area of the piston of actuator

To introduce these characteristic gripper parameters Table 5.1 has been developed for the industrial gripper mechanisms, taken as examples corresponding to Fig. 5.13–5.24. In this Table 5.2, the value of the (G.I.) depends on the actuator's cylinder. This is taking into account considerations of the grasp mechanics ( $\cos\phi$ ) and uniting kinematic characteristics finger driving mechanism ( $F/p_{al}A$ ). This index has been established in this way because the catalogs do not have the actuators dimensions or the gripper bars. It is expressed with the data available, as the relationship between the grasp force  $F$  and the actuator's feeding pressure  $p_{al}$  by the cosine of the grasp angle  $\varphi$ ; the index being expressed, depending on the active area  $A$  of the piston actuator [10], [11]:

$$G.I. = \frac{F}{p_{al}A} \cos \varphi \quad (5.9)$$

Therefore, Table 5.2 gives the value of G.I. calculated for the gripper examples Figs. 5.13–5.24, as a merit quantification of the gripper's operation and design.

In this table we can find characteristic parameters of the industrial grippers, which may be useful in a selection or in a design process.

It shows the characteristics of the kind of pneumatic or electrical actuator according to its way of sliding. This defines the input mechanism for finger driving. This serves for the type synthesis.

The finger movement shows the approximation characteristics of the finger to the object, thus defining part of the mechanics which will take place between them.

**Table 5.2** Evaluation of the industrial grippers, using the characteristics of the proposed classifications. (A: active area of the actuator's piston)

Figure/gripper model	Numeric parameters of valuation					
	Trajectory plus (mm)	Maximum object	Grasp force (N)	Grasp angle $\Phi(^{\circ})$	Feeding pressure (N/cm <sup>2</sup> )	Grasp index (GI.A)
Fig. 5.13 MHQ2-6	4	6	3.3	0	60	0.05
Fig. 5.14 MHQG2-10	4	10	11	0	50	0.22
Fig. 5.15 MHR2-10	16	10	12	0	50	0.24
Fig. 5.16 MHQG2-32	20	32	88	0	50	1.76
Fig. 5.17 GP210 N	10	70	170	0	90	1.88
Fig. 5.18 WGW6	90°	14	4.3 (Nm)	20	60	0.07
Fig. 5.19 MHC2-10	40°	10	10 (Nm)	10	50	0.20
Fig. 5.20 ZK22	15°	3.5	1000 (Nm)	5	70	14.28
Fig. 5.21 GP500	54	90	1190	0	90	13.22
Fig. 5.22 GP325	25	100	300	0	90	1.66
Fig. 5.23 RH918ST140	140	140	100	0	60	1.66
Fig. 5.24 RH901	5	11	56	0	60	0.93

The kinematics components provide the necessary information to define the complexity of the mechanism or the necessary maintenance, as well as the kind of joints.

The trajectory plus the object's size which is able to grasp, indicate the gripper's opening capability, which will provide the amount of objects which can be manipulated with it.

The grasp force is indicating the maximum value that the gripper may have to support the object intended to be manipulated.

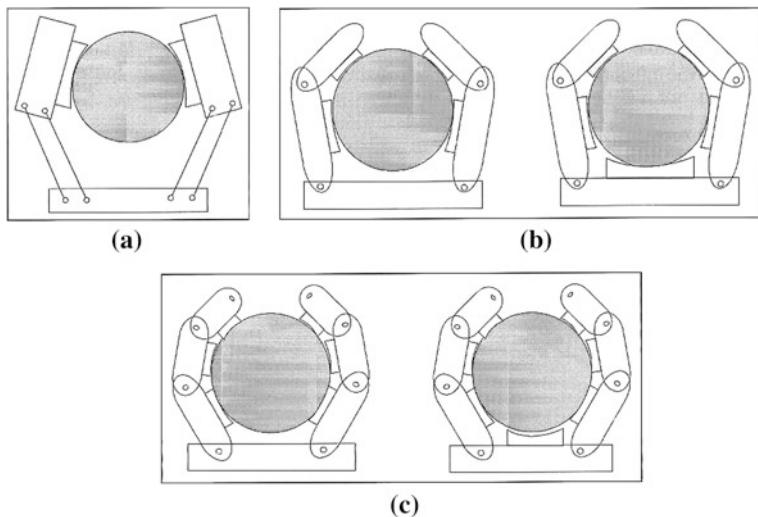
The grasp angle is the element which finally defines the grasp mechanics between gripper and object.

The feeding pressure is the maximum pressure recommended by the manufacturer for the correct gripper operation.

The grasp index is the one which enables comparisons to define which gripper is the best as regards the grasp mechanics, taking into account the object's size and the maximum feeding pressure.

This table is a contribution to designers. It provides the possibility to select from what has been industrially tested, which kind of design adapts better to the specific conditions of the problem to be solved

For objects of regular shape, there can be specific grasp configurations in which the values of the grasp forces necessary may diminish and the object's stability in the gripper may increase. To achieve this it is necessary to increase the contact points between object and gripper. There are several possible configurations to achieve this effect, among these, one is the use of passive contacts as the palm use,



**Fig. 5.27** Grasping configuration and mechanical designs for two-finger grippers [6]: **a** rigid fingers for two-grasp contacts; **b** articulated fingers without and with palm for four grasp contacts; **c** articulated fingers without and with palm for six grasp contacts

the other is the use of articulated fingers simulating the human hand, as is shown in Fig. 5.27, as may be found in [6].

In all cases, the configurations of planar mechanisms are used as grasp mechanisms. There is a specific characteristic when contact points are considered as flexible, these are installed on the fingers with a specific curving in order to adapt to the shape of the specific object.

## References

1. Belfiore NP, Pennestrì E (1997) An atlas of linkage-type robotic grippers. *Mech Mach Theory* 32(7):811–834
2. Belforte G (1985) Robot Grasp Organ. *Tech J AMMA* 2:128–136 (Torino)
3. Ceccarelli M (1994) Design problems for industrial robot two finger grippers. In: Third international workshop on robotics in Alpe-Adria region RAA94, Bled, pp 117–120
4. Ceccarelli M, Gradini G (1992) Robot's gripper mechanism: classification and optimization. *Autom Robot Tech* :1–20
5. Cecarelli M, Nieto J (1993) Grasping with a two-finger gripper. In: First Iberoamerican congress of mechanical engineering, vol 4, Madrid, pp 171–176
6. Cecarelli M (2004) Fundamentals of mechanics of robotic manipulation. Kluwer Academic Publishers, Dordrecht, ISBN 1-4020-1810-X
7. Chen F (1982) Gripping mechanisms for industrial robots: an overview. *Mech Mach Theory* 17(5):299–311
8. Kauffmann JM, Andre P, Taillard JP (1987) Robot component and systems. *Robot Technol* 4:297–310

9. Lundstrom G (1973) Industrial robot grippers. The Industrial Robot, Sweden, pp 72–82
10. Penisi O, Ceccarelli M, Carbone G (2003) Classification of industrial two-finger gripper mechanisms. Iberoamerican Mag Mech Eng 7(1):59–75, ISSN 1137-2729
11. Penisi O, Ceccarelli M, Carbone G (2002) Optimum design and testing of Mechanisms for two-finger grippers. Int J Mech Control 3(1):9–20, ISSN 1590-8844
12. Penisi O (2004) Grasp mechanics and gripper project. Ph.d. thesis, ISBN 987-43-8058-6
13. Pham DT, Heginbotham WB (eds) (1986) Robot grippers. IFS Publications Ltd, Bedford
14. SMC 1996 Corporation (1996) Pneumatic gripper, Series MH Cat.I 305A”, Italy, 1996
15. Tanie K (1985) Design of robot hands. Handbook of industrial robotics. Wily, New York, pp 112–137 (Chapter 8)
16. TBK (1993) Robot hand '94. Takano Bearing Co. Ltd., Tokyo
17. Techno Sommer (1997/98) Techno Sommer Automatic, “automation components from a single source”, New Hyde Park, NY 11042-5416, 1997/98
18. Weforma (1999) Dämpfungstechnik GmbH, “General catalogue 2.0 ideas for automation”, Stolberg

# Chapter 6

## Robotic Hands and Underactuated Finger Mechanisms

Licheng Wu

**Abstract** A real biologic hand has the features of compact volume, many fingers, many degree of freedoms, and strong grasp force, which brings big challenges to built a robotic hand-like real hand. In fact, all of the power supply, drivers, control system, sensors, and information processing system are need to be installed in the humanoid robot itself, which provides very strict power cost and real-time control as well as the humanoid appearance of the humanoid robot hand.

### 6.1 Introduction

A real biologic hand has the features of compact volume, many fingers, many degree of freedoms, and strong grasp force, which brings big challenges to built a robotic hand like real hand. In fact, all of the power supply, drivers, control system, sensors, and information processing system are need to be installed in the humanoid robot itself, which provides very strict power cost and real-time control as well as the humanoid appearance of the humanoid robot hand.

Researchers have investigated different types of robotic hands for achieving grasp and handling of objects. Several successful designs for humanoid and nonhumanoid hands have been developed in the past, like those that are used, for example, in the Stanford/JPL Hand [1], DLR Hand [2], BUAA Hand [3], Colobi Hand [4], Barrett Hand [5, 6], TUAT/Karlsruhe Hand [7], Turin Hand [8], TBM Hand [9], MA-I Hand [10], SARAH Hand [11, 12], MIT Hand [13] and the RCH-1

---

L. Wu (✉)

School of Information Engineering, Minzu University of China, Zhongguancun South Avenue, 27, Beijing 100081 Haidian district, P. R. China  
e-mail: wulichenggg@hotmail.com

Hand [14]. Most of the available multi-fingered prototypes have a high number of degrees of freedom (DOF), a complex actuation, and a high cost. Most of those hands are still used for enhanced applications, laboratory tests, or research projects. But even in the market, hands are available for industrial and nonindustrial applications, like for example the Barrett Hand [6], SARAH Hand [12], ROBO-SOFT Hand [15].

Many robotic hands focus on simulating the overall appearance and action of a biologic hand, while neglecting other equally important features such as size, weight, and real-time control. Conventional robotic hand, such as in [16], is relatively complex, large, heavy, and difficult to be installed in a robot arm with the size like a human. The complexity of conventional robotic hand is also very expensive and difficult to manufacture and maintain. However, although the robotic hand has many normal DOFs, the driver and control system need to be big and difficult to control. In some sense, a robotic hand with more DOFs is more dexterous. And the number of the DOFs directly determines the ability of a hand to bend around an object and adapt to its shape.

An important point on designing a robotic hand is how to decrease the volume, weight, and power cost. Therefore, designing a robotic hand with fewer drivers becomes an interesting direction. However, it becomes a challenge that how to design a robotic hand with feature of compact, low-cost and easy-operation but still keep the numbers of DOFs to support the ability of grasp adaptively different object. A hot researched way is to utilize fingers with underactuated driving mechanisms.

A finger with an underactuated driving mechanism has this ability with the additional feature that the finger can be controlled by a reduced number of actuators. A reduced number of actuators means that the finger can be built with cost oriented and user-friendly features. Currently, there are two main types of underactuated finger mechanisms, namely, tendon-actuated mechanisms and linkage-based mechanisms as outlined in [17].

Tendon-actuated mechanisms can be easily built with compact architectures but they can exert only small grasping forces. The other disadvantage of tendon systems is that they are strongly affected by friction and elasticity [18]. Some patents are issued on development of robotic hand with tendons mechanisms in [19] and [20]. A tendons mechanism for an underactuated prosthetic hand is reported in [21] and [22]. An optimal design problem is discussed as related to the diameter and location of the pulleys in [23]. A two degree-of-freedom/one-actuator finger is presented with isotropic force characteristics in [24]. Here, force-isotropic means that the intensity of the forces exerted at the center of each phalanx, and on the grasped object, are identical. A design of an underactuated finger mechanism in TBM hand is presented in [25]. Some preliminary results on a cosmetic prosthetic hand with tendon driven underactuated mechanism and compliant joints are also presented in [26].

For applications where large grasping forces are requested, linkage-based mechanisms are more convenient. The US Patent [27] and its improved version [28] were published in 1998 and 2003. But the mechanism proposed in [27] and

[28] can hardly be built in a compact design, especially when the mechanism is demanded to be embedded within the finger body with a human-size during the whole motion of grasp operation. The thumb of TH-1 hand of Tsinghua University has a prismatic passive joint that is driven by a spring [29]. An improved version for TH-2 hand is reported in [30]. Principles for underactuated mechanism and flexible mechanisms with 1 DOF finger mechanism are carefully discussed in [17] and they have been considered in developing the solution that is presented in [31]. A pneumatically driven underactuated hand is reported in [32].

Some literatures are presented on the aspect of design, grasp stability, contact force characteristics, grasp characteristics, etc., of the underactuated finger. Zollo et al.[33] proposed a biomechatronic approach to the design of an anthropomorphic artificial hand able to mimic the natural motion of the human fingers. In [34], architectures of two-DOF underactuated fingers are proposed and a simulation tool is designed to analyze their behaviors. Then, the relationship between the input torque of the finger actuator and the contact forces on the phalanges are discussed in [35]. A procedure is presented in [36] to analyze the grasp stability of two-phalanx underactuated fingers. Luo et al. [37] studied the grasp characteristics of an underactuated hand. A five-fingered underactuated prosthetic hand controlled by surface electromyographic signals is presented in [38].

In this chapter, principle, kinematics, statics, and design methods of an underactuated finger mechanism for robotic hands are discussed first. Then, a simulation procedure and simulation results are introduced. Finally, some conclusions are outlined.

## 6.2 Principle of an Underactuated Finger Mechanism

At LARM, Laboratory of Robotics and Mechatronics in Cassino Italy, a specific research activity has been devoted to designing low-cost easy-operation grasping devices with 1-DOF anthropomorphic fingers since the early 1990s, as indicated in [17, 39, 40] and [41]. Figure 6.1 shows a prototype of LARM Hand Version 4, with three 1-DOF fingers [42]. The size of this prototype is 1.2 times bigger than an average human hand of Italian population. The physical size of the phalanx bodies is: height—2 cm; length—6 cm, 3.71 cm, 3.5 cm for phalanx 1,2,3 respectively. Its main features are low-cost design and easy-operation. Each finger is basically composed of two four-bar linkage mechanisms. An optimal design method for finger mechanisms of anthropomorphic fingers with 1 DOF has been reported in [39]. In order to improve the capability of grasping objects with different sizes and shapes, solutions with underactuated mechanisms or flexible mechanisms are carefully discussed in [17]. A new version of LARM Finger will be built with an underactuated mechanism as in following design. Thus, LARM Hand will possess the ability to grasp a large family of objects as a human finger does.

**Fig. 6.1** A prototype of LARM hand 4 with finger mechanisms within finger bodies

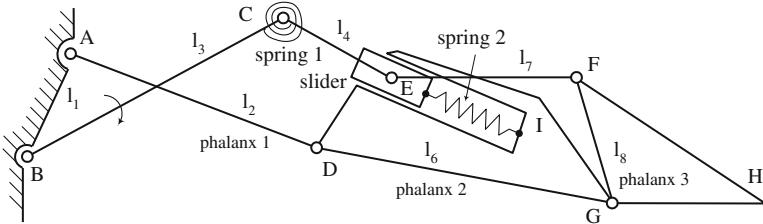


For a humanoid finger, the finger mechanism must be very compact to preserve the overall size of the finger body that will be built with human size. A solution has been selected for a finger mechanism with springs only from those solutions advised in [17]. As shown in Fig. 6.2, the specific new kinematic design has been proposed in [31] with a torsional spring at a revolute joint and a sliding joint is used with a linear spring to model a flexible link.

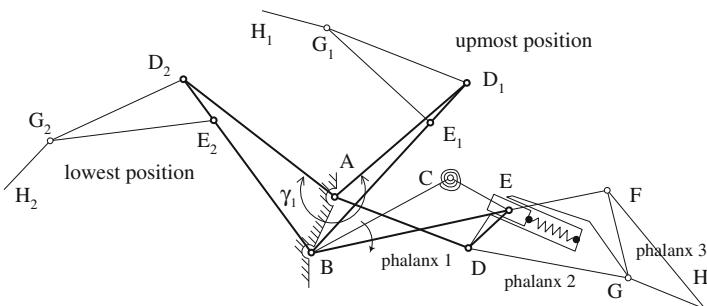
As shown in Fig. 6.2, the introduced mechanism has seven links, one slider and two springs. Link  $l_1$  is the frame. Links  $l_2$ ,  $l_6$ , and  $l_8$  are the three phalanxes. Revolute joints are at  $A, B, C, D, E, F, G$  points, and there is a sliding joint within the phalanx 2. A torsional spring (spring 1) is located between links  $l_3$  and  $l_4$  as surrounding the joint at point  $C$ . Another spring (spring 2) is located along the sliding joint. The end points of the linear spring are fixed on the slider and link  $l_6$ . The only one motor is installed to have the shaft at the joint in point  $B$  and it drives directly link  $l_3$ .

Since the finger is light weight and moves slowly in the mechanism plane, static considerations can be of main concerns. Obviously, the springs will keep their initial length without any deformations when phalanxes are free. Namely, point  $C$  acts like a fixed point and point  $E$  is virtually fixed on link  $l_6$ , because the slider will have no move respect to link  $l_6$ . In this case, the introduced finger mechanism acts virtually as a cross four bar linkage  $ABED$  as shown in Fig. 6.3.

Because the whole mechanism has to be embedded within a finger body with a big aspect ratio, the frame being link  $l_1$  can be the shortest link or the opposite to shortest link of the four-bar linkage  $ABED$ . Four bar linkage  $ABED$  can be double-crank or double-rocker mechanism. When link  $l_2$  is a crank, phalanx 1 being link  $l_2$  can rotate circle around point  $B$ . When link  $l_2$  is a rocker, phalanx 1 being link  $l_2$  will reach its limiting position when the four bar linkage  $ABED$  shapes a triangle. With a proper design, phalanx 1 can still have a wide angular stroke  $\gamma_1$  as shown in Fig. 6.3.

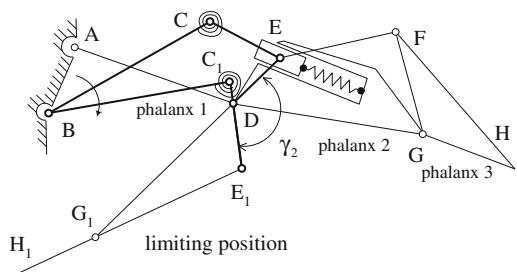


**Fig. 6.2** Kinematic scheme of an underactuated finger mechanism



**Fig. 6.3** The angular stroke of the phalanx 1

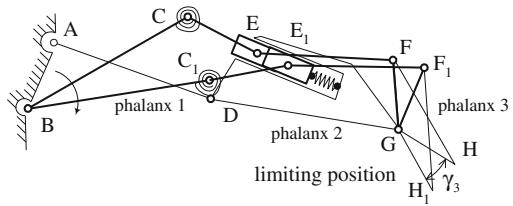
**Fig. 6.4** The forward angular stroke of the phalanx 2 relative to phalanx 1



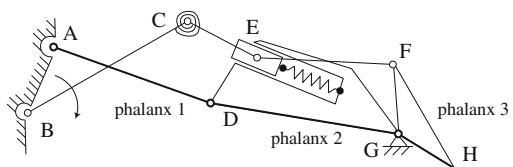
Once a phalanx contacts with an object tightly or reaches its limiting position, it can not move forward again, namely the phalanx is stopped. Referring to Fig. 6.2, in case that phalanx 1 is stopped while phalanxes 2 and 3 are still free, any movement of link l<sub>3</sub> will still actuate phalanx 2 through link l<sub>4</sub>. Phalanx 1 is stopped means that point D is virtually fixed on the base. Spring 1 will start to deform with the push of motor. While point E is still virtually fixed on link l<sub>6</sub> because spring 2 keeps still. Thus, the phalanx 2 is virtually driven by the four bar linkage BCED as shown in Fig. 6.4.

Obviously, with a proper design of link length of the four-bar linkage BCED, phalanx 2 (here being the link between point D and E) can even be a crank that can

**Fig. 6.5** The forward angular stroke of the phalanx 3 relative to phalanx 2



**Fig. 6.6** The condition that the mechanism stops to move forward

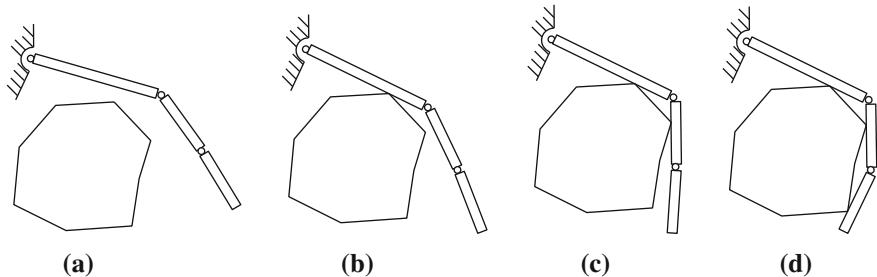


rotate circle around point  $D$ . However, the four-bar linkage  $BCED$  may meet a dead-point when it shapes a triangle. Namely, the driver being link  $l_3$  cannot push forward phalanx 2 again, it reaches its limiting position as shown in Fig. 6.4. After phalanx 1 is stopped, phalanx 2 can rotate forward with a forward angular stroke  $\gamma_2$  as shown in Fig. 6.4. The distance between point  $B$  and  $D$  is changing when phalanx 1 is turning. Thus, the exact value of  $\gamma_2$  will change a little too. However, an estimate value of  $\gamma_2$  can be calculated at an appointed configuration, for example at the limiting position of phalanx 1.

Referring to Figs. 6.2 and 6.5, even if phalanx 2 is stopped, any motion of link  $l_3$  can continuously actuate the slider. Then the motion of slider will actuate phalanx 3 through link  $l_7$ . As shown in Fig. 6.5, obviously when points  $B$ ,  $C$ , and  $E$  locate at one line, the slider cannot move forward again. Then phalanx 3 cannot turn forward again, namely it reaches the limiting position. As shown in Fig. 6.5, with a proper design (a short link  $l_8$ ), short move interval of the slider can generate a big forward angular stroke  $\gamma_3$  of phalanx 3 relative to phalanx 2. An estimate value of  $\gamma_3$  can be at the limiting position of phalanx 2.

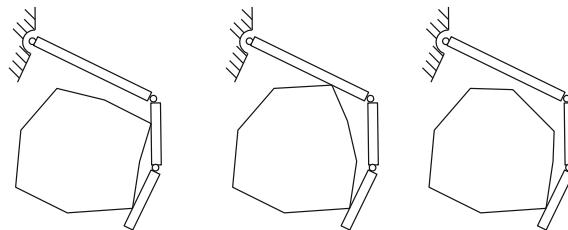
As shown in Fig. 6.6, once phalanx 3 is stopped, the position of point  $G$  is virtually fixed on the frame. This means that position of the triangle  $ADG$  is fixed. Thus phalanx 1 (link along with  $AD$ ) and phalanx 2 (link along with  $DG$ ) cannot move again, whether they have touched an object or no. And the other part of the mechanism can obviously not move too. Thus, the introduced finger mechanism stops to move forward only if phalanx 3 cannot move forward again.

The above analysis shows the underactuation principle of underactuated finger mechanism through the example of the introduced mechanism. Namely, any movement of link  $l_3$  will actuate phalanxes 1, 2, and 3 to move forward before they touched the object or reached a mechanical limiting position as shown in Fig. 6.3. When phalanx 1 cannot move again, any movement of link  $l_3$  will continuously actuate phalanx 2 and 3 to move forward as shown in Fig. 6.4. When phalanx 1 and 2 cannot move again, any movement of link  $l_3$  will continuously actuate phalanx 3 to move forward as shown in Fig. 6.5. Only if phalanx 3 cannot move



**Fig. 6.7** A typical sequence of grasping an object: **a** first phase—approaching motion; **b** second phase—first phalanx in contact; **c** third phase—second phalanx in contact; **d** final configuration—third phalanx in contact

**Fig. 6.8** Final grasping configurations with less contact points



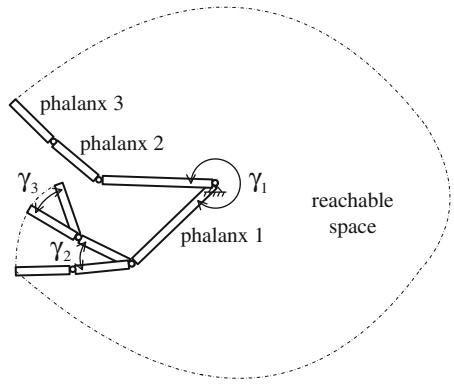
forward again, the finger stops to move forward as shown in Fig. 6.6. Thus, the introduced mechanism can generate a proper underactuation operation although only one motor is installed at the joint in point  $B$  and it drives directly link  $l_3$ .

The typical close sequence can be shown in Fig. 6.7. Namely, phalanxes 1, 2, and 3 will touch the object one after the other while the underactuated mechanism will adjust its configuration to the object size and shape toward a final finger configuration with all the phalanxes in contact with the object. A sequence of this kind of grasping can be described in three phases: the finger moves freely; phalanx 1 reaches contact with the object and phalanxes 2, 3 are still moving free; phalanx 2 reaches contact with the object and phalanx 3 continues to move toward the object.

Due to different shape or position of object, sometimes a distal phalanx would contact with the object before proximal one. Thus, second phase or third phase would be jumped from the grasping sequence. However, the grasping operation can also be achieved while the only difference is the final configuration will possess less contact points as shown in Fig. 6.8.

The area that the finger can sweep with its phalanxes can be indicated with dash-dot line as shown in Fig. 6.9. This area can be identified as the reachable space of the finger, while the range of the reachable space is decided by the parameters  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$  (it is assumed that these angular strokes can be achieved by the solid mechanism). Obviously, a finger with the introduced mechanism can possibly grasp any object get into the reachable space. In fact, the object can be wrapped by the finger independently of its shape and size, when the whole object

**Fig. 6.9** Reachable space of a finger with the introduced mechanism



is inside the reachable space. Thus, the reachable space and values of  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$  can be used to denote the adaptive capability to grasp object with different size and shape. The reachable space and values of  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$  can be calculated through a simulation procedure that is based on formulation for kinematic analysis of the mechanism.

According to the above analysis, although the introduced mechanism has only one active DOF, it can operate by underactuated action to adaptively grasp objects with different sizes and shapes by adapting its configuration.

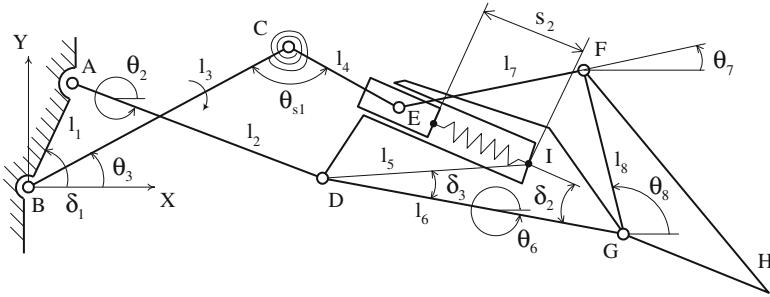
### 6.3 Kinematics for Finger Operation

In order to model the operation of the introduced finger mechanism, a kinematic analysis is thought necessary to deduce suitable formulations. A coordinate system has been fixed at the point  $B$  on the frame link  $l_1$  and kinematical parameters are identified as shown in Fig. 6.10.

Let  $l_i$  refer also to the length of link  $l_i$ ,  $i = 1, 2, \dots, 8$ . In particular,  $l_5$  is the length between points  $D$  and  $I$ . Let point  $I$  be the reference point for the sliding motion of the slider.  $s_2$  is the distance from point  $E$  to point  $I$ . Let  $\theta_i$  be the angle from X axis to the link  $l_i$ ,  $i = 2, 3, 6, 7, 8$ . The parameter  $\delta_1$  is the angle from X axis to link  $l_1$ .  $\delta_2$  is the angle of the sliding joint direction with respect to the edge  $DG$  of link  $l_6$  (namely phalanx 2).  $\delta_3$  is the angle  $IDG$ .  $\theta_{s1}$  is the angle of spring 1, namely angle between link  $l_3$  and  $l_4$ . The closure equations of link polygons  $ABCEID$  and  $DIEFG$  can be expressed respectively as

$$l_1 e^{i\delta_1} + l_2 e^{i\theta_2} + l_5 e^{i(\theta_6 + \delta_3)} = l_3 e^{i\theta_3} + l_4 e^{i(\theta_{s1} + \theta_3 - \pi)} + s_2 e^{i(\theta_6 - \delta_2)} \quad (6.1)$$

$$l_5 e^{i(\theta_6 + \delta_3)} + l_7 e^{i\theta_7} = s_2 e^{i(\theta_6 - \delta_2)} + l_6 e^{i\theta_6} + l_8 e^{i\theta_8} \quad (6.2)$$



**Fig. 6.10** Kinematical parameters of the introduced finger mechanism

The operation of the finger mechanism can be simulated once the kinematic variables for the links can be solved as functions of the input angle  $\theta_3$  of the actuated link. The kinematic variables are  $\theta_2, \theta_3, \theta_{s1}, \theta_6, \theta_7, \theta_8$  and  $s_2$ , when  $l_1, l_2, \dots, l_8, \delta_1, \delta_2$  and  $\delta_3$  are given structural parameters. When phalanxes are free, the springs will keep their initial length without any deformations, namely  $\theta_{s1}, s_2$  are known as their design initial value. Thus when angle of driver link  $\theta_3$  is known, the operation variables  $\theta_2$  and  $\theta_6$  can be solved by using Eq. (6.1) as

$$\theta_6 = 2 \arctan \frac{B_6 - \sqrt{A_6^2 + B_6^2 - C_6^2}}{A_6 - C_6} \quad (6.3)$$

$$\theta_2 = \arctan \frac{-t_2 + s_2 \sin(\theta_6 - \delta_2) - l_5 \sin(\theta_6 + \delta_3)}{-t_1 + s_2 \cos(\theta_6 - \delta_2) - l_5 \cos(\theta_6 + \delta_3)} \quad (6.4)$$

where  $t_1 = l_1 \cos \delta_1 - l_3 \cos \theta_3 - l_4 \cos(\theta_{s1} + \theta_3 - \pi)$ ,  
 $t_2 = l_1 \sin \delta_1 - l_3 \sin \theta_3 - l_4 \sin(\theta_{s1} + \theta_3 - \pi)$ ,  $A_6 = -2t_1s_2 \cos \delta_2 + 2t_1l_5 \cos \delta_3 + 2t_2s_2 \sin \delta_2 + 2t_2l_5 \sin \delta_3$ ,  $B_6 = -2t_1s_2 \sin \delta_2 - 2t_1l_5 \sin \delta_3 - 2t_2s_2 \cos \delta_2 + 2t_2l_5 \cos \delta_3$ ,  $C_6 = t_1^2 + t_2^2 + s_2^2 + l_5^2 - l_2^2 - 2s_2l_5 \cos(\delta_2 + \delta_3)$ .

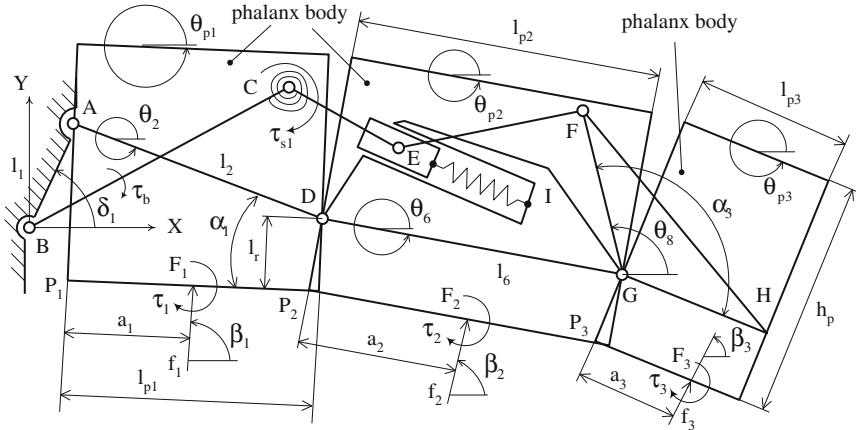
Successively, the angles  $\theta_7$  and  $\theta_8$  can be solved by using Eq. (6.2) as

$$\theta_8 = 2 \arctan \frac{B_8 + \sqrt{A_8^2 + B_8^2 - C_8^2}}{A_8 - C_8} \quad (6.5)$$

$$\theta_7 = \arctan \frac{t_4 + l_8 \sin \theta_8}{t_3 + l_8 \cos \theta_8} \quad (6.6)$$

where  $t_3 = -l_5 \cos(\theta_6 + \delta_3) + s_2 \cos(\theta_6 - \delta_2) + l_6 \cos \theta_6$ ,  $t_4 = -l_5 \sin(\theta_6 + \delta_3) + s_2 \sin(\theta_6 - \delta_2) + l_6 \sin \theta_6$ ,  $A_8 = 2t_3l_8$ ,  $B_8 = 2t_4l_8$ ,  $C_8 = t_3^2 + t_4^2 + l_8^2 - l_7^2$ .

But the computation of the kinematic variables  $\theta_{s1}$  and  $s_2$  can be solved only by taking into account also the spring behavior through a suitable static analysis when the finger contacts with an object.



**Fig. 6.11** A model for static formulation with operation parameters

## 6.4 Statics for Grasping Actions

A static analysis is necessary to link the configuration of the finger mechanism to the spring conditions. As reported in [43], a real grasp of human finger can have many contact points between an object and one phalanx. But it is possible to consider a minimum number of contact points between an object and a finger. Thus, a static planar grasp can be obtained with three contact points, one at each phalanx. In Fig. 6.11, one contact for each phalanx has been assumed at the contact point  $F_i$ , that is located at the bottom of phalanx  $i$  ( $i = 1, 2, 3$ ). The contact force and torque are indicated as  $f_i$  and  $\tau_i$  ( $i = 1, 2, 3$ ). Let vector of contact forces be  $f_e = [\tau_1, \tau_2, \tau_3, f_{1x}, f_{1y}, \dots, f_{3x}, f_{3y}]^T$ , where  $f_{ix}, f_{iy}$  are the  $X$  and  $Y$  components of  $f_i$ . Let generalized driving force vector be  $\tau = [\tau_b, \tau_{s1}, \tau_{s2}]^T$ , where  $\tau_b$  is the driving torque of motor, while  $\tau_{s1}$  and  $f_{s2}$  are the acting torque of spring 1 and acting force of spring 2, respectively.

According to the principle of virtual work, the relationship between external force and driving force can be written as

$$\tau = J^T f_e \quad (6.7)$$

where  $J$  is the Jacobian matrix of the introduced mechanism.

The Jacobian  $J$  can be formulated by using orientation of the phalanxes and coordinates of the contact points ( $F_i$ ,  $i = 1, 2, 3$ ) in the finger. Let each phalanx be shaped as a rectangle body as shown in Fig. 6.11. Distance  $h_p$  is the height of phalanxes.  $l_{pi}$  is the length of phalanx  $i$  ( $i = 1, 2, 3$ ).  $l_r$  is the distance from the fixed position of joint  $D$  to phalanx's edge and similarly for the other phalanxes.  $\theta_{pi}$  is the orientation angle of phalanx  $i$  ( $i = 1, 2, 3$ ) with respect to X axis.  $\alpha_1$  is the angle between link  $l_2$  and bottom of phalanx 1. Edge  $DG$  is designed as parallel to edge

of phalanx 2.  $\alpha_3$  is the angle between  $FG$  and edge of phalanx 3. Thus, according to the geometry in Fig. 6.11, the orientation angles of phalanxes can be calculated as function of the finger kinematic variables as

$$\begin{aligned}\theta_{p1} &= \theta_2 + \alpha_1 \\ \theta_{p2} &= \theta_6 \\ \theta_{p3} &= \theta_8 - \alpha_3\end{aligned}\quad (6.8)$$

As shown in Fig. 6.11, point  $P_i$  is the left lower point of phalanx body  $i$  ( $i = 1, 2, 3$ ). Distance from point  $P_i$  to point  $F_i$  is given by the parameter  $a_i$  ( $i = 1, 2, 3$ ). Let  $[x_i, y_i]^T$  be the coordinates of  $F_i$  ( $i = 1, 2, 3$ ). By referring to the geometry in Fig. 6.11, position coordinates of phalanxes can be calculated as

$$\begin{aligned}x_1 &= l_1 \cos \delta_1 + (l_1 \sin \delta_1 + l_r) \cos(\theta_{p1} - 0.5\pi) + a_1 \cos \theta_{p1} \\ y_1 &= l_1 \sin \delta_1 + (l_1 \sin \delta_1 + l_r) \sin(\theta_{p1} - 0.5\pi) + a_1 \sin \theta_{p1} \\ x_2 &= l_1 \cos \delta_1 + l_2 \cos \theta_2 + l_r \cos(\theta_{p2} - 0.5\pi) + a_2 \cos \theta_{p2} \\ y_2 &= l_1 \sin \delta_1 + l_2 \sin \theta_2 + l_r \sin(\theta_{p2} - 0.5\pi) + a_2 \sin \theta_{p2} \\ x_3 &= l_1 \cos \delta_1 + l_2 \cos \theta_2 + l_6 \cos \theta_6 + l_r \cos(\theta_{p3} - 0.5\pi) + a_3 \cos \theta_{p3} \\ y_3 &= l_1 \sin \delta_1 + l_2 \sin \theta_2 + l_6 \sin \theta_6 + l_r \sin(\theta_{p3} - 0.5\pi) + a_3 \sin \theta_{p3}\end{aligned}\quad (6.9)$$

Thus, the  $9 \times 3$  Jacobian matrix  $J$  can be formulated by using its definition given in Eq. (6.7). Let  $J_{ij}$  refer to the Jacobian's entry at  $i$ th row and  $j$ th column. According to Eq. (6.7), the upper three rows of  $J$  can be formulated by using kinematic formulation in Sect. 6.3 in the expressions

$$\begin{aligned}J_{11} &= \frac{\partial \theta_2}{\partial \theta_3}, J_{12} = \frac{\partial \theta_2}{\partial \theta_{s1}}, J_{13} = \frac{\partial \theta_2}{\partial s_2}, J_{21} = \frac{\partial \theta_6}{\partial \theta_3}, J_{22} = \frac{\partial \theta_6}{\partial \theta_{s1}}, J_{23} = \frac{\partial \theta_6}{\partial s_2}, J_{31} \\ &= \frac{\partial \theta_8}{\partial \theta_3}, J_{32} = \frac{\partial \theta_8}{\partial \theta_{s1}}, J_{33} = \frac{\partial \theta_8}{\partial s_2}\end{aligned}\quad (6.10)$$

By using Eq. (6.9), the remaining entries of  $J$  can be formulated as

$$J_{41} = w_1 J_{11}, J_{42} = w_1 J_{12}, J_{43} = w_1 J_{13} \quad (6.11)$$

$$J_{51} = w_2 J_{11}, J_{52} = w_2 J_{12}, J_{53} = w_2 J_{13} \quad (6.12)$$

$$J_{61} = w_3 J_{11} + w_4 J_{21}, J_{62} = w_3 J_{12} + w_4 J_{22}, J_{63} = w_3 J_{13} + w_4 J_{23} \quad (6.13)$$

$$J_{71} = w_5 J_{11} + w_6 J_{21}, J_{72} = w_5 J_{12} + w_6 J_{22}, J_{73} = w_5 J_{13} + w_6 J_{23} \quad (6.14)$$

$$\begin{aligned}J_{81} &= w_3 J_{11} + w_7 J_{21} + w_8 J_{31}, J_{82} = w_3 J_{12} + w_7 J_{22} + w_8 J_{32}, \\ J_{83} &= w_3 J_{13} + w_7 J_{23} + w_8 J_{33}\end{aligned}\quad (6.15)$$

$$\begin{aligned}J_{91} &= w_5 J_{11} + w_9 J_{21} + w_{10} J_{31}, J_{92} = w_5 J_{12} + w_9 J_{22} + w_{10} J_{32}, \\ J_{93} &= w_5 J_{13} + w_9 J_{23} + w_{10} J_{33}\end{aligned}\quad (6.16)$$

$$\begin{aligned}
w_1 &= -(l_1 \sin \delta_1 + l_r) \sin(\theta_{p1} - \frac{\pi}{2}) - a_1 \sin \theta_{p1}, \\
w_2 &= (l_1 \sin \delta_1 + l_r) \cos(\theta_{p1} - \frac{\pi}{2}) + a_1 \cos \theta_{p1}, w_3 = -l_2 \sin \theta_2, \\
w_4 &= -l_r \sin(\theta_{p2} - \frac{\pi}{2}) - a_2 \sin \theta_{p2}, w_5 = l_2 \cos \theta_2, w_6 = l_r \cos(\theta_{p2} - \frac{\pi}{2}) + a_2 \cos \theta_{p2}, \\
w_7 &= -l_6 \sin \theta_6, \quad w_8 = -l_r \sin(\theta_{p3} - \frac{\pi}{2}) - a_3 \sin \theta_{p3}, \quad w_9 = l_6 \cos \theta_6, \\
w_{10} &= l_r \cos(\theta_{p3} - \frac{\pi}{2}) + a_3 \cos \theta_{p3}.
\end{aligned}$$

Thus, the value of  $J$  at a certain configuration can be calculated by using Eqs. (6.10–6.16).

When analyzing a grasp by a finger, only positive grasp pressure can be considered as feasible. In addition, neglecting contact torque and friction force gives a conservative condition for a finger grasp. This means that a contact force can be assumed as perpendicular to the phalanx contact surface. Let  $\beta_i$  be the direction angle of  $f_i$ , i.e., angle from axis X to  $f_i$ , then it holds that

$$\beta_i = \theta_{pi} + \frac{1}{2}\pi, \quad i = 1, 2, 3 \quad (6.17)$$

Let  $f_i$  be the magnitude of  $f_i$ . Consequently, X and Y components of the contact force can be calculated as

$$\begin{aligned}
f_{xi} &= f_i \cos \beta_i = f_i \cos(\theta_{pi} + \frac{1}{2}\pi), \quad i = 1, 2, 3 \\
f_{yi} &= f_i \sin \beta_i = f_i \sin(\theta_{pi} + \frac{1}{2}\pi)
\end{aligned} \quad (6.18)$$

Let  $k_1$  and  $k_2$  be the coefficients of spring 1 and 2, respectively. Thus, the acting forces of the springs can be expressed as

$$\tau_{s1} = k_1(\theta_{s1} - \theta_{s10}) \quad (6.19)$$

$$f_{s2} = k_2(s_2 - s_{20}) \quad (6.20)$$

where  $\theta_{s10}$  and  $s_{20}$  are the initial size values for springs 1 and 2, respectively.

Therefore, the contact force vector is given as  $f_c = [0, 0, 0, f_1 \cos \beta_1, f_1 \sin \beta_1, \dots, f_3 \cos \beta_3, f_3 \sin \beta_3]^T$  and Eq. (6.7) can be simplified as

$$\begin{bmatrix} \tau_b \\ \tau_{s1} \\ f_{s2} \end{bmatrix} = \begin{bmatrix} \tau_b \\ k_1(\theta_{s1} - \theta_{s10}) \\ k_2(s_2 - s_{20}) \end{bmatrix} = V \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \quad (6.21)$$

where

$$V = \begin{bmatrix} J_{41} \cos \beta_1 + J_{51} \sin \beta_1 & J_{61} \cos \beta_2 + J_{71} \sin \beta_2 & J_{81} \cos \beta_3 + J_{91} \sin \beta_3 \\ J_{42} \cos \beta_1 + J_{52} \sin \beta_1 & J_{62} \cos \beta_2 + J_{72} \sin \beta_2 & J_{82} \cos \beta_3 + J_{92} \sin \beta_3 \\ J_{43} \cos \beta_1 + J_{53} \sin \beta_1 & J_{63} \cos \beta_2 + J_{73} \sin \beta_2 & J_{83} \cos \beta_3 + J_{93} \sin \beta_3 \end{bmatrix} \quad (6.22)$$

In Eqs. (6.1), (6.2), and (6.21), there are totally seven equations and 11 variables  $\theta_2, \theta_3, \theta_{s1}, \theta_6, \theta_7, \theta_8, s_2, f_1, f_2, f_3$  and  $\tau_b$ . Thus, when four of those variables are known, the others can be solved by using the combination of Eqs. (6.1), (6.2), and (6.21). For example, when phalanxes 1, 2, and 3 are all free,  $f_1, f_2$ , and  $f_3$  are

known as being null. Once the value of  $\theta_3$  is given too, since it is the input of the finger configuration, then  $\theta_2$ ,  $\theta_{s1}$ ,  $\theta_6$ ,  $\theta_7$ ,  $\theta_8$ ,  $s_2$ , and  $\tau_b$  can be solved from Eqs. (6.1), (6.2), and (6.21). However, when the finger has contacted with the object, the shape and size of the object and the relative position between the finger and the object need to be considered for solving Eqs. (6.1), (6.2), and (6.21).

In order to analyze the static of the finger mechanism at the grasp configuration, the situation that the object is fixed on the frame is considered.

At a certain grasp configuration,  $\tau_{s1}$  and  $f_{s2}$  can be calculated by using Eqs. (6.19) and (6.20) because  $\theta_{s1}$  and  $s_2$  are known. Then  $f_i$  ( $i = 1, 2, 3$ ) can be calculated by using Eq. (6.21) according to a given value of  $\tau_b$ . Generally, a phalanx can push the object, but cannot pull the object. This forms a constraint for contact force.

## 6.5 Design Method

### 6.5.1 Size Design

In order to design a humanoid finger mechanism, an optimal design method has been developed as reported in [39]. The method can design outline and basic structure of the finger mechanism through fulfilling basically requirements on human-like motion and grasp, compact size and lightweight. However, a new design task consists of designing the additional links and sliding joint for underactuated moving ability, for example length  $l_3$ ,  $l_4$ ,  $l_7$  of links 3, 4, 7 and stroke  $s_2$  of the sliding joint of the introduced mechanism as shown in Fig. 6.2. Links 3 and 4 can be designed by considering the four-bar linkage *BCED* in Fig. 6.4. Stroke of sliding joint and Link 7 can be designed by considering four-bar mechanism *BCEI* and *IEFG* as shown in Figs. 6.5 and 6.6, respectively. The design requirements are:

- to keep the mechanism within the phalanx' body during its operation;
- driving angle of four bar mechanism considered is greater than  $40^\circ$ .

In particular, the size of the links and the motion range of the springs are determined through a trial-error analysis up to ensure for limit cases of a given family of objects that all the phalanges are in contact with the object and the finger mechanism is kept inside the finger body. After a suitable set of parameters is designed, a design solution can be conveniently checked through simulation as shown in Sect. 6.5. A reference simulation calculates a typical grasp that the finger performs to grasp an object as shown in Fig. 6.7.

As shown in Fig. 6.7, phalanges 1, 2, and 3 will touch the object one after the other. A grasping sequence can be described in three phases: 1st phase—the finger moves freely; 2nd phase—phalanx 1 touches the object and phalanges 2, 3 are still free; 3rd phase—phalanx 2 touches the object and phalanx 3 continues to move

toward the object. In the schemes of Fig. 6.7, the object and phalanxes are assumed as rigid bodies, and the object is fixed with respect to the base of the finger. The phalanxes will stop as soon as they touch the object. At each phase, the finger operates actually according to an acting mechanism as shown in Figs. 6.3, 6.4, and 6.5. Therefore, links lengths and ranges of spring strokes can be adjusted by using the acting mechanisms and their fairly simple kinematics in a step-by-step analysis that will give suitable solution with the prescribed requirements of phalanx-object contacts and finger mechanism in finger body.

### 6.5.2 Design of the Coefficients of Springs

The design method to compute the spring coefficients is outlined in the following based on static analysis.

According to Eq. (6.21), the relation between contact force and driving torque and spring coefficients can be obtained as

$$\begin{bmatrix} \tau_b \\ k_1 \\ k_2 \end{bmatrix} = U \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \quad (6.23)$$

where

$$U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\theta_{s1} - \theta_{s10}} & 0 \\ 0 & 0 & \frac{1}{s_2 - s_{20}} \end{bmatrix} V \quad (6.24)$$

Thus, the spring coefficients  $k_1$  and  $k_2$  can be computed as function of a grasp configurations that can be determined through a kinematic simulation. Then, coefficients of the two springs can be evaluated from Eq. (6.23) according to a set of expected or prescribed contact forces at a given configuration.

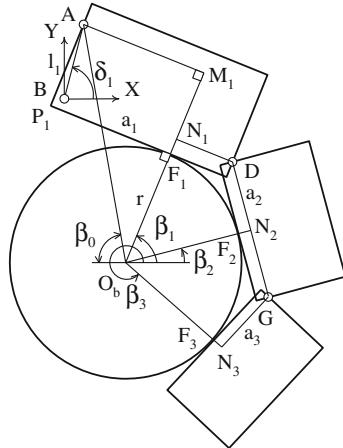
Once the spring coefficients  $k_1$  and  $k_2$  are determined, the spring design can be checked by computing and verifying the contact forces at the final finger configuration for the given driving torque  $\tau_b$ .

## 6.6 Simulation Procedure of an Underactuated Finger

### 6.6.1 A Simulation Procedure for a Typical Grasp Sequence

A typical grasp test for a robotic hand consists in grasping a cylinder. Phalanxes 1, 2, and 3 will touch the object one after the other through the sequence as shown in Fig. 6.7. According to the geometry relationship between the finger and the

**Fig. 6.12** A model for the finger grasping a cylinder



cylinder object as shown in Fig. 6.12, some kinematic variables can be computed when the phalanxes have contact with the object one by one.

In Fig. 6.12,  $O_b$  and  $r$  indicate the center and the radius of the cylinder object, respectively.  $\beta_0$  is the angle between X axis and line  $AO_b$ .  $N_1$  is the point on the line  $O_bF_1$  that makes line  $DN_1$  parallel to the edge of phalanx 1.  $N_2$  is the point on the line  $O_bF_2$  that makes line  $DN_2$  parallel to the edge of phalanx 2.  $N_3$  is the point on the line  $O_bF_3$  that makes line  $GN_3$  parallel to the edge of phalanx 3.  $M_1$  is the point on the line  $O_bF_1$  that makes line  $AM_1$  parallel to the edge of phalanx 1. Thus, the length of  $F_1M_1$  is equal to the length of  $AP_1$ , namely  $l_1 \sin \delta_1 + l_r$ . The coordinates  $(x_0, y_0)$  refer to the position of the center of the cylinder with respect to XY frame. Thus, when phalanx 1 contacts with the cylinder object, the direction angle of  $f_1$  and the position of the contact point  $F_1$  can be computed through

$$\begin{aligned} \beta_1 &= \pi - \beta_0 - \angle AO_b M_1 \\ &= \pi - \arctan \frac{l_1 \sin \delta_1 - y_0}{x_0 - l_1 \cos \delta_1} - \arccos \frac{r + l_1 \sin \delta_1 + l_r}{\sqrt{(x_0 - l_1 \cos \delta_1)^2 + (l_1 \sin \delta_1 - y_0)^2}} \end{aligned} \quad (6.25)$$

$$a_1 = \sqrt{(x_0 - l_1 \cos \delta_1)^2 + (l_1 \sin \delta_1 - y_0)^2 - (r + l_1 \sin \delta_1 + l_r)^2} \quad (6.26)$$

where  $\angle AO_b M_1$  indicates the angle between segments  $AO_b$  and  $O_b M_1$ . Then, according to Eqs. (6.17) and (6.8),  $\theta_2$  can be computed as

$$\theta_2 = \beta_1 - \frac{1}{2}\pi - \alpha_1 \quad (6.27)$$

When phalanxes 1 and 2 come to contact with the cylinder object, the direction angle of  $f_2$  and the position of the contact point  $F_2$  can be computed through

$$\begin{aligned}\beta_2 &= \beta_1 - \angle N_1 O_b D - \angle N_2 O_b D \\ &= \beta_1 - \arctan \frac{l_{p1} - a_1}{r + l_r} - \arccos \frac{r + l_r}{\sqrt{(l_{p1} - a_1)^2 + (r + l_r)^2}}\end{aligned}\quad (6.28)$$

$$a_2 = l_{p1} - a_1 \quad (6.29)$$

Then, according to Eqs. (6.17) and (6.8),  $\theta_6$  can be computed as

$$\theta_6 = \beta_2 - \frac{1}{2}\pi \quad (6.30)$$

When phalanxes 1, 2, and 3 are all in contact with the cylinder object, the direction angle of  $f_3$  and the position of the contact point  $F_3$  can also be computed through

$$\begin{aligned}\beta_3 &= \beta_2 - \angle N_2 O_b G - \angle N_3 O_b G \\ &= \beta_2 - \arctan \frac{l_{p2} - a_2}{r + l_r} - \arccos \frac{r + l_r}{\sqrt{(l_{p2} - a_2)^2 + (r + l_r)^2}}\end{aligned}\quad (6.31)$$

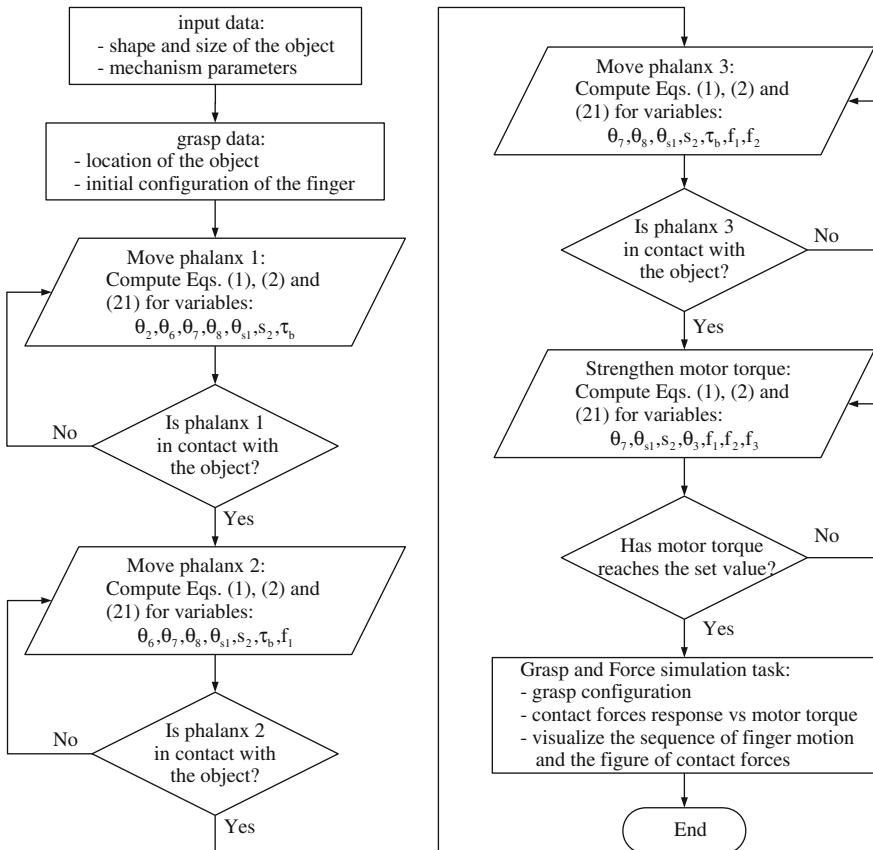
$$a_3 = l_{p2} - a_2 \quad (6.32)$$

Finally, according to Eqs. (6.17) and (6.8),  $\theta_8$  can be computed as

$$\theta_8 = \beta_3 - \frac{1}{2}\pi + \alpha_3 \quad (6.33)$$

Therefore, when all the phalanxes move freely,  $f_1, f_2$ , and  $f_3$  are known with null values. When phalanx 1 reaches contact with the object,  $f_1$  becomes unknown but  $\theta_2$  can be computed by using Eq. (6.27). When phalanx 2 reaches contact with the object,  $f_1$  and  $f_2$  become unknowns but  $\theta_2$  and  $\theta_6$  can be computed by using Eqs. (6.27) and (6.30), respectively. When phalanxes 1, 2, and 3 are all in contact with the object,  $f_1, f_2$ , and  $f_3$  are unknowns but  $\theta_2, \theta_6$ , and  $\theta_8$  can be computed by using Eqs. (6.27, 6.30), and (6.33), respectively. Thus, once the motor input angle  $\theta_3$  or torque  $\tau_b$  is given, the kinematics of the finger can be solved by using Eqs. (6.1), (6.2), and (6.21). Generally when the motor input angle  $\theta_3$  is given, a position control is used. When the motor input torque  $\tau_b$  is given, a force control is used. According to Eq. (6.21), when the value of  $f_1, f_2$ , and  $f_3$  are 0,  $\tau_b$  equals 0 too. Thus, when all phalanxes move freely, only the motor input angle  $\theta_3$  can be given. On the other hand, when all phalanxes reach contact with the object, namely at the final grasp configuration, the value of  $\theta_3$  is fixed and only the motor input torque  $\tau_b$  can be given.

Thus, a numerical simulation for the finger mechanism can be performed accordingly through a numerical procedure that is outlined in the flowchart in Fig. 6.13 by using the introduced formulations in Eqs. (6.1), (6.2), and (6.21). In the introduced simulation procedure, an initial value is given for the motor input angle  $\theta_3$  at the initial configuration at which all the phalanxes are free. Then, the



**Fig. 6.13** A flowchart for the introduced simulation procedure

motor input angle  $\theta_3$  is decreased continuously until when all the phalanxes reach contact with the object because of a sequence of phalanx motions. Once the final grasp configuration is reached, the simulation switches to force simulation. Namely, the motor input torque  $\tau_b$  is given increasingly, so that its value will give the contact forces  $f_1$ ,  $f_2$ , and  $f_3$  for a static equilibrium of the grasp. The outlined simulation procedure can be used to check grasping capability of a finger and its operation as well as to determine suitable sizes of the finger mechanism and phalanx bodies. In fact, by iteratively looking at conditions of the finger motion and grasp contact, a design solution can be determined by adjusting the link sizes, the spring elements, and the location of the contact points toward a sequence of finger configurations and toward a final firm grasp that is considered satisfactory for the given task.

### 6.6.2 A Simulation Procedure for Reachable Space

The outlined procedure in the flowchart in Fig. 6.13 can be also used to compute the reachable space of the finger mechanism as a measure of the self-adaptive capability in grasping objects with different size and shape. The procedure can be outlined as in the following. Let  $\theta_{s1}, s_2$  keep their original values and the motor input angle  $\theta_3$  is a given value, then the values of  $\theta_2, \theta_6, \theta_7, \theta_8$  can be solved by using Eqs. (6.1) and (6.2). Therefore, increasing  $\theta_3$  gradually from 0 to 360°, the reachable space can be calculated as the area between the upmost and lowest limiting positions of phalanx 1 and the angular stroke of phalanx 1.

When  $\theta_2$  keeps its value for the lowest position of phalanx 1,  $s_2$  keeps original value while  $\theta_3$  is a given value, then  $\theta_{s1}, \theta_6, \theta_7, \theta_8$  can be solved again by using Eqs. (6.1) and (6.2). Therefore, increasing  $\theta_3$  gradually from its value to the value for the lowest position of phalanx 1, the reachable space can be calculated as the area from the lowest limiting positions of phalanx 1 to the limiting position of phalanx 2 with forward angular stroke of phalanx 2.

Finally, when  $\theta_2$  and  $\theta_6$  keep their values for the limiting position of phalanx 2 while  $\theta_3$  is a given value, then  $\theta_{s1}, s_2, \theta_7, \theta_8$  can be solved again by using Eqs. (6.1) and (6.2). Therefore, increasing  $\theta_3$  gradually from its value to the value for the limiting position of phalanx 2, the reachable space can be calculated as the area from the limiting positions of phalanx 2 to the limiting position of phalanx 3 with forward angular stroke of phalanx 3.

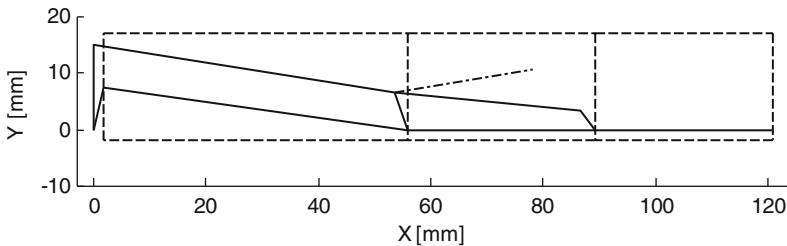
## 6.7 Simulations

A series of numerical simulations have been computed with the introduced formulation through a suitable code in Matlab®. In order to develop a finger with average sizes of Chinese populations, the size of phalanx body has been prescribed as in Table 6.1. The two joints in  $D$  and  $G$  can be located on the extremity of the phalanx plate with a clearance of  $l_r = 2$  mm from the bottom. With the initial prescribed data in Table 6.1 after several simulations, a suitable solution has been obtained as shown in Fig. 6.14 with mechanism sizes in Table 6.2. The reachable space of this solution can be plot as shown in Fig. 6.15. The angular strokes can be calculated as  $\gamma_1 = 283.2^\circ$ ,  $\gamma_2 = 148.8^\circ$ , and  $\gamma_3 = 135.8^\circ$ . The springs' coefficients have been determined with values  $k_1 = 0.37$  Nm/rad and  $k_2 = 3000$  N/m.

Initial value of the input angle  $\theta_3$  of the motor has been assumed as  $\theta_3(0) = 16.2^\circ$  and initial angles of  $\theta_2, \theta_6, \theta_7, \theta_8$  have been computed as  $-8.0, 0, -5.9, 130.0^\circ$ , respectively. The input angle  $\theta_3$  has been scanned with an initial step of  $0.80^\circ$ , and then with a step of  $1.0^\circ$ . The initial values of spring parameters have been assumed by  $\theta_{s10} = 81.2^\circ$ ,  $s_{20} = 25.0$  mm, respectively. First, the computed simulations refer to the case in which the finger grasps a cylinder object with the radius of  $r = 30$  mm and at the position  $(16.3, -46.3)$  of its center with respect to

**Table 6.1** Initial prescribed sizes of phalanx bodies for a Chinese-like finger

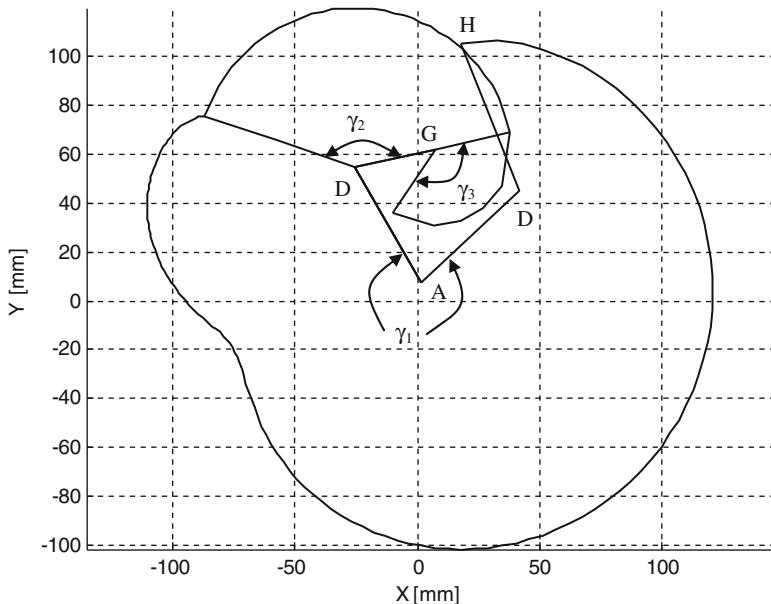
Phalanx i	1	2	3
$l_{pi}$ (mm)	54.0	33.4	31.5
$h_p$ (mm)	19.0	19.0	19.0

**Fig. 6.14** A sketch of the introduced finger mechanism with Chinese size (Continuous line indicates the mechanism; dashed lines indicate the phalanx bodies; dash-dot line indicates the sliding joint.)**Table 6.2** Structure parameters for the designed underactuated finger mechanism

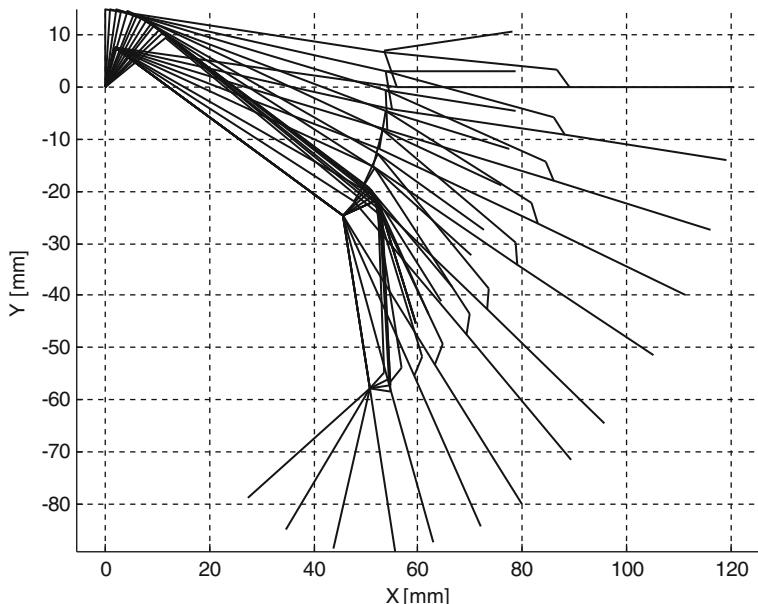
i	1	2	3	4	5	6	7	8
$l_i$ (mm)	7.8	54.5	15.0	54.1	24.9	33.4	33.3	4.0
$\alpha_i$ ( $^{\circ}$ )	8.0	—	130.0	—	—	—	—	—
$\delta_i$ ( $^{\circ}$ )	76.7	-8.4	25.4	—	—	—	—	—

XY frame. Obviously, the object can be absolutely wrapped by the finger because it is inside the reachable space.

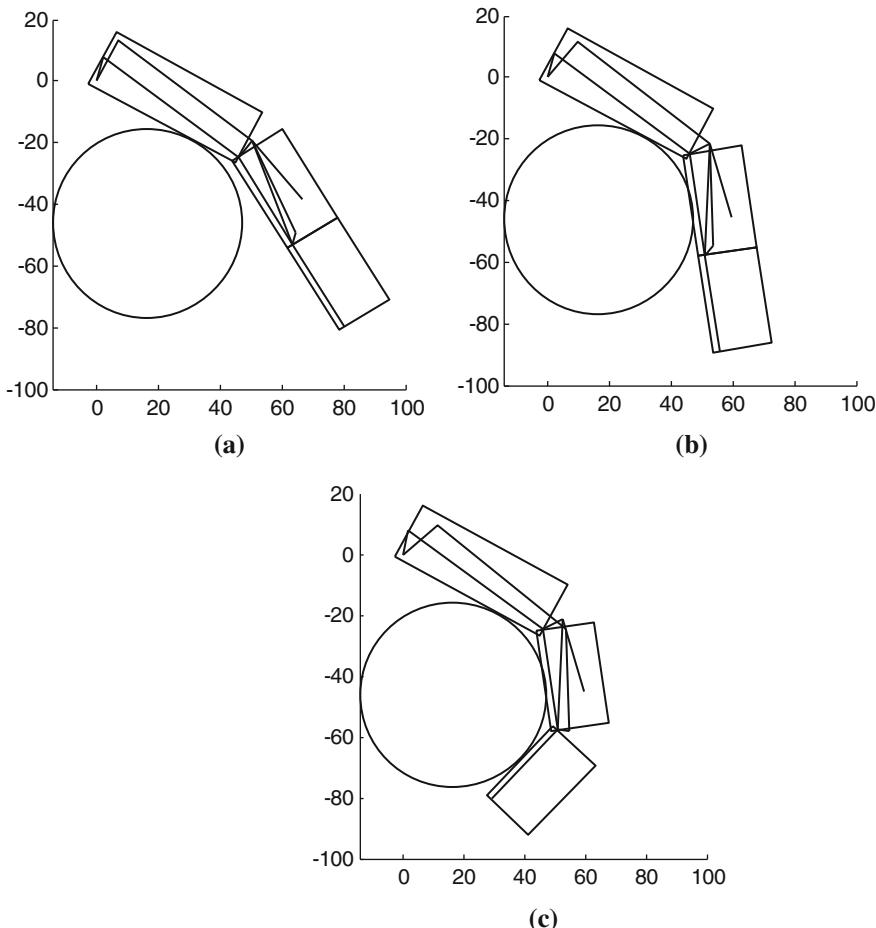
The simulation results are shown in Figs. 6.16, 6.17, 6.18, 6.19, and 6.20. Figure 6.16 shows the grasping sequence of final design for the finger underactuated mechanism while Fig. 6.17 shows a grasping sequence of the finger with Chinese size, respectively. In addition, plot of phalanx angles of the introduced underactuated finger mechanism with Chinese size are shown in Fig. 6.18 as referring to the motion of the phalanxes. Figure 6.19 shows the computed grasp forces by the phalanxes 1 and 2 on the object and the motor driving torque before phalanx 3 contacts with the object. The computed grasp forces by the phalanxes 1, 2, and 3 on the object at the final grasp configuration are shown in Fig. 6.20 as function of the motor driving torque that has been used to analyze the final grasp equilibrium. As shown in Fig. 6.18, phalanxes 1, 2, and 3 rotate smoothly to the final angles of 28.5, 81.2, and 138.2°, respectively, when the motor turns of 50.0° ( $\theta_3$  from 90 to 40°). As shown in Fig. 6.19, the phalanxes 1 and 2 start suitable grasp forces on the object during the grasp sequence. As shown in Fig. 6.20, the grasp forces show as function to the motor driving torque at the final wrap configuration. For example, when  $\tau_b$  is 0.45 Nm, the grasp forces have been computed as acting on the object with values  $f_1 = 6.1$  N,  $f_2 = 1.1$  N and  $f_3 = 2.7$  N. When



**Fig. 6.15** Reachable space of the introduced finger mechanism with Chinese size



**Fig. 6.16** A simulation result of grasping a cylinder object by the designed Chinese-like underactuated finger mechanism

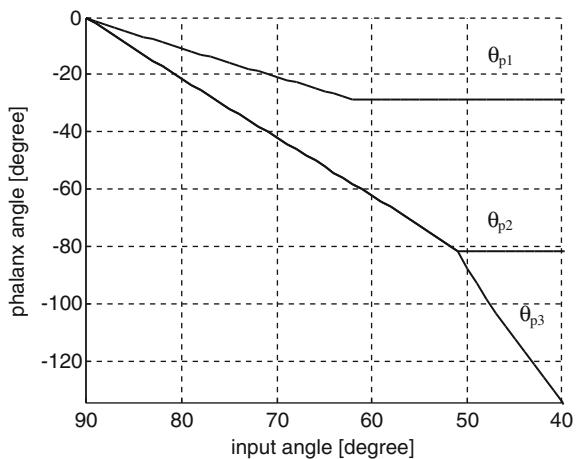


**Fig. 6.17** A grasp sequence of the designed finger with the cylinder object, Fig. 6.16: **a** phalanx 1 reaches the object at  $\theta_3 = 62.0^\circ$ ; **b** phalanx 2 reaches the object at  $\theta_3 = 51.0^\circ$ ; **c** phalanx 3 reaches the object at  $\theta_3 = 40.0^\circ$

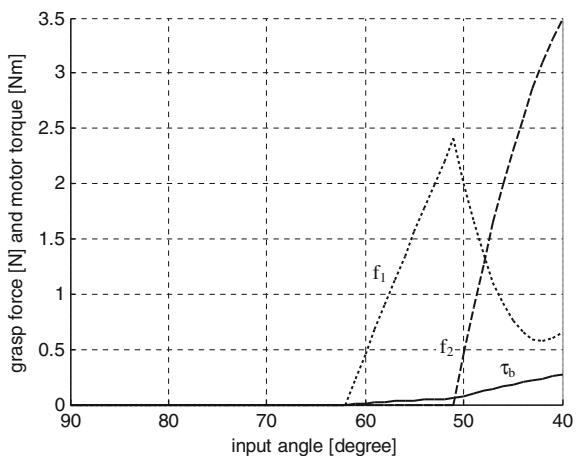
$\tau_b$  is 0.54 Nm, the grasp force  $f_2$  becomes zero, namely the phalanx 2 is going to lose the object. If driving torque  $\tau_b$  is strengthened more, a negative grasp force  $f_2$  is demanded to keep the current grasp configuration. In practice, the phalanx 2 will deviate from the object to form a new equilibrium configuration.

In order to check and show the versatility and feasibility of the introduced mechanism architecture and computed design solution, the introduced simulation procedure has also been applied to cylindrical objects with different sizes ( $r = 25$  mm and 40 mm, respectively) but at the same position. As shown in Fig. 6.21, the size of an object can modify the finger configuration and the grasp contact point within a wide range of size of grasp operations.

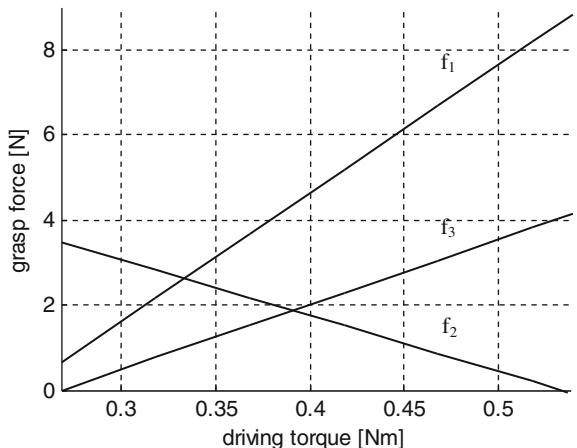
**Fig. 6.18** Phalanx angles of the underactuated finger mechanism during the grasp simulation of Figs. 6.16 and 6.17

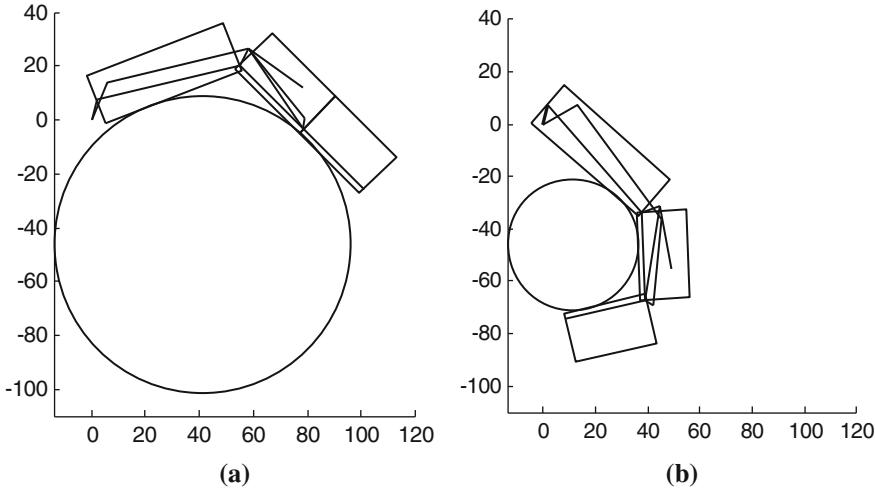


**Fig. 6.19** Motor driving torque and grasp forces by phalanxes 1 and 2 before the phalanx 3 reaches contact with the object, during the grasp simulation of Figs. 6.16 and 6.17



**Fig. 6.20** Grasp forces by the phalanxes 1, 2 and 3 as function of the motor driving torque at the final grasp configuration as shown in Fig. 6.17c





**Fig. 6.21** The different grasp configurations for the cylinder object with different sizes: **a** the radius of the object is  $r = 25$  mm; **b** the radius of the object is  $r = 55$  mm

In addition, among those simulation results, the distances among the mechanism links and borders of the phalanx bodies have been checked by computing that successfully the mechanism is kept inside the phalanx bodies with more than 1.8 mm clearance during the analyzed grasping sequence.

The results of the final simulation in Figs. 6.16, 6.17, 6.18, 6.19, 6.20, and 6.21 show the feasibility of the final size of the underactuated finger mechanism in Table 6.2 both in terms of motion/force properties and mechanical design. The phalanxes act properly through a suitable sequence mimicking a human-like grasp preserving the features of low-cost design and ease-operation. Low-cost design is ensured by the linkage structure and compact layout that is given by the fact that all the mechanism links remaining always inside the finger body during the grasp operation. The easy-operation feature has been preserved by having confirmed one actuator only and by suitable smooth sequence of the phalanx actions.

## 6.8 Conclusions

In this chapter, from the view of developing practical robotic hands, a new underactuated finger mechanism for robotic hands is mainly introduced. Kinematics analysis, static analysis, and design method are discussed for the introduced mechanism as related to design issues and achieved results. A simulation procedure for design purposes as based on the kinematics and statics analysis through trial-error iterations is also discussed. A reachable space is identified and characterized in terms of the range and capability of finger operation to adaptively grasp objects with

various shapes and sizes. A mechanical design is verified through a simulation procedure in which a finger grasps a cylinder object. It is shown that a proper design of the introduced mechanism can be successfully embedded into a finger body with human size. The finger shows a humanoid grasp behavior with a proper efficiency and object-adaptation through a 1 DOF actuated operation.

## References

1. Mason MT, Salisbury JK (1985) Robots hands and the mechanics of manipulation, MIT Press, Cambridge
2. Butterfass J, Grebenstein M, Liu H, Hirzinger G (2001) DLR-hand II: next generation of a dexterous robot hand. In: IEEE international conference on robotics and automation, Seoul, pp 109–114
3. Zhang Y, Han Z, Zhan H, Shang X, Wang T, Guo W (2001) Design and control of the BUAA four-fingered hand. In: Proceedings of the 2001 IEEE International Conference on Robotics and Automation, Seoul, pp 2517–2522
4. Gosselin CM, Mountambault S, Gosselin CJ (1993) Manus colobi: preliminary results on the design of a mechanical hand for industrial applications. In: 19th ASME design automation conference, Albuquerque, vol 65, issue no 1, pp 585–592
5. Townsend WT (2000) The barrett hand grasper—programmable flexible parts handling and assembly. Ind Robot: An Int J :181–188
6. Barrett Hand. Available from: <http://www.barretttechnology.com>
7. Fukaya N, Toyama S, Asfour T, Dillmann R (2000) Design of the TUAT/Karlsruhe humanoid hand. In: IEEE/RSJ international conference on intelligent robots and systems, Takamatsu, pp 1754–1759
8. Raparelli T, Mattiazzo G, Mauro S, Velardocchia M (2000) Design and development of a pneumatic anthropomorphic hand. J Robot Syst 17(1):1–15
9. Dechev N, Cleghorn WL, Nauman S (1999) Multiple finger, passive adaptive grasp prosthetic hand. Mech Mach Theory 36:1157–1173
10. Suares R, Grosch P (2004) Dexterous robotic hand MA-I software and hardware architecture. Intelligent manipulation and grasping, Genova, pp 91–96
11. Martin E, Desbiens AL, Laliberte' T, Gosselin C (2004) SARAH hand used for space operation on STVF robot. Intelligent manipulation and grasping, Genova, pp 279–284
12. SARAH Hand. Available from. [http://www.robot.gmc.ulaval.ca/recherche/theme04\\_a.html](http://www.robot.gmc.ulaval.ca/recherche/theme04_a.html)
13. Edsinger-Gonzales A (2004) Design of a compliant and force sensing hand for a humanoid robot. Intelligent manipulation and grasping, Genova, pp 291–295
14. Roccella S, Carrozza MC, Cappiello G, Dario P, Cabibihan JJ, Zecca M, Hiwa H, Itoh K, Matsumoto M, Takanishi A (2004) Design, fabrication and preliminary results of a novel anthropomorphic hand for humanoid robotics: RCH-1. In: IEEE/RSJ international conference on intelligent robots and systems, Sendai, pp 266–271
15. ROBOSOFT Hand. Available from. <http://www.robosoft.fr/SHEET/05Grippers/1002ROBO-SOFT3FINGER/3Finger.html>
16. Qian X, Zhang Q (1996) The driver method and application of a driver mechanism of knuckle joint. China Patent No. 1136988A
17. Ceccarelli M, Tavolieri C, Lu Z (2006) Design considerations for underactuated grasp with a one D.O.F. anthropomorphic finger mechanism. In: Proceedings of 2006 IEEE/RSJ international conference on intelligent robots and systems (IROS 2006), Oct 9–15, Beijing, pp 1611–1616
18. Ceccarelli M (2004) Fundamentals of mechanics of robotic manipulation. Kluwer/Springer, Dordrecht

19. Mullen JF (1972) US Patent 3694021, Sept 26, 1972
20. Crisman JD, Kanojia C, Zeid I (1996) US Patent 5570920, Nov 5, 1996
21. Massa B, Roccella S, Carrozza MC, Dario P (2002) Design and development of an underactuated prosthetic hand. In: Proceedings of the 2002 IEEE international conference on robotics and automation, Washington, May 2002, pp 3374–3379
22. Carrozza MC, Suppo C, Sebastiani F, Massa B, Vecchi F, Lazzarini R, Cutkosky MR, Dario P (2004) The SPRING hand: development of a self-adaptive prosthesis for restoring natural grasping. *Auton Robot* 16:125–141
23. Cabas R, Cabas LM, Balaguer C (2006) Optimized design of the underactuated robotic hand. In: Proceedings of the 2006 IEEE international conference on robotics and automation (ICRA 2006), Orlando, May 2006, pp 982–987
24. Krut S (2005) A force-isotropic underactuated finger. In: Proceedings of the 2005 IEEE international conference on robotics and automation (ICRA 2005), Barcelona, April 2005, pp 2314–2319
25. Dechev N, Cleghorn WL, Naumann S (2001) Multiple finger, passive adaptive grasp prosthetic hand. *Mech Mach Theory* 36:1157–1173
26. Carrozza MC, Cappiello G, Stellin G, Zaccone F, Vecchi F, Micera S, Dario P (2005) A cosmetic prosthetic hand with tendon driven under-actuated mechanism and compliant joints: ongoing research and preliminary results. In: Proceedings of the 2005 IEEE international conference on robotics and automation (ICRA 2005), Barcelona, April 2005, pp 2661–2666
27. Gosselin C, Laliberte T (1998) US Patent 5762390, June 9, 1998
28. Laliberte T, Gosselin C (2003) Actuation system for highly underactuated gripping mechanism, US Patent 6505870, Jan 14, 2003
29. Zhang W, Chen Q, Sun Z (2003) Under-actuated humanoid robot hand with changeable grasping force. *J Tsinghua Univ (Sci & Tech)* 43(8):1143–1147
30. Zhang W, Chen Q, Sun Z, Zhao D (2004) Passive adaptive grasp multi-fingered humanoid robot hand with high under-actuated function. In: Proceedings of the 2004 IEEE international conference on robotics and automation, New Orleans, April 2004, pp 2216–2221
31. Wu L, Carbone G, Ceccarelli M (2009) Design an underactuated mechanism for a 1 active DOF finger operation, *Mech Mach Theory*, Accepted
32. Bégo V, Krut S, Dombre E, Durand C, Pierrot F (2007) Mechanical design of a new pneumatically driven underactuated hand. In: IEEE international conference on robotics and automation, Roma, April 10–14 2007, pp 927–933
33. Zollo L, Roccella S, Guglielmelli E, Carrozza MC, Dario P (2007) Biomechatronic design and control of an anthropomorphic artificial hand for prosthetic and robotic applications. *IEEE/ASME Trans Mechatron* 12(4):418–429
34. Laliberte T, Gosselin CM (1998) Simulation and design of underactuated mechanical hands. *Mech Mach Theory* 33:39–57
35. Birglen L, Gosselin CM (2004) Kinetostatic analysis of underactuated fingers. *IEEE Trans Robot Autom* 20(2):211–221
36. Birglen L, Gosselin CM (2006) Grasp-state plane analysis of two-phalanx underactuated fingers. *Mech Mach Theory* 41:807–822
37. Luo M, Mei T, Wang X, Yu Y (2004) Grasp characteristics of an underactuated robot hand. In: Proceedings of the 2004 IEEE international conference on robotics and automation, New Orleans, April 2004, pp 2236–2241
38. Zhao J, Jiang L, Shi S, Cai H, Liu H, Hirzinger G (2006) A five-fingered underactuated prosthetic hand system. In: Proceedings of the 2006 IEEE international conference on mechatronics and automation, Luoyang, June 25–28, 2006, pp 1453–1458
39. Nava Rodriguez NE, Carbone G, Ceccarelli M (2006) Optimal design of driving mechanism in a 1-DOF anthropomorphic finger. *Mech Mach Theory* 41:897–911
40. Civitillo R (2001) Design and experimental validation of an antropomorphic finger with one d.o.f., Master thesis, LARM, University of Cassino, Cassino, 2001. (in Italian)

41. Ceccarelli M, Carbone G, Ottaviano E, Nava Rodriguez NE (2004) An experimental validation of a three-fingered hand with 1 d.o.f. anthropomorphic fingers. In: International conference on intelligent manipulation and grasping IMG04, Genova, 2004, pp 285–290
42. LARM homepage. Available from. <http://webuser.unicas.it/weblarm/>
43. Park YC, Starr GP (1992) Grasp synthesis of polygonal objects using a three-fingered robot hand. *Int J Robot Res* 11(3):163–183

# Chapter 7

## Finger Orientation for Robotic Hands

Minzhou Luo

### 7.1 Introduction

Human hand has evolved to be a complex and adaptable dexterous tasks system. Due to its large number of degrees of freedom (DOFs), it can easily realize different orientations and it is able to reconfigure itself to different configurations very quickly, with the aim to grasp objects dexterously [1]. Furthermore, it can manipulate objects with arbitrary position and orientation.

There has been a growing interest in the development of multi-fingered robotic hands to mimic the performances and features of human hands. During the last several decades, different types of devices have been developed by researchers in institutes, universities, and corporations to achieve the grasp and operation of objects. These devices try to mimic the performances of human hands in order to have high flexibility and multi-purpose operation. Moreover, most of the available multi-fingered prototypes have a large number of DOFs, complex control, and high cost [2–7].

Nevertheless, those dexterous hands and mechanical grippers are still not able to fully reproduce all the grasping capabilities and cover all the grasp modes of human hand. This main technical problem can be supposed to be solved by considering the possibility of involving a device that can adjust finger postures to form versatile grasp configurations. In order to achieve this goal, it is important to design a robotic hand which can move and rotate its fingers.

In this chapter, problems of robotic hands are outlined firstly, based on the analysis of performances and features from human hand to existing robotic hands. Then, examples of several robotic hands which have orientation mechanisms are illustrated. Finally, conclusions are outlined with simple instructions.

---

M. Luo (✉)

Chinese Academy of Science, 350 Shushanhu Road, Hefei, Anhui, People Republic of China  
e-mail: lmz@iim.ac.cn

## 7.2 Attached Problems

In general, robotic hands can be classified into two kinds: the dexterous hands and the mechanical grippers.

Dexterous hands which usually have more than two fingers have been widely studied and reported, for example, in [8–13]. These types of devices have high flexibility in multi-purpose operation but usually involve high cost and complex control systems because of their high number of DOFs.

On the other hand, mechanical grippers usually have low cost and easy-operation features since they have low number of DOFs. Due to these two characteristics, mechanical grippers also have been widely studied. Some typical examples can be found in the literature [14–17]. However, low number of DOFs also brings drawbacks: these grippers are not so flexible like dexterous hands. Thus, they have very limited grasping and operation capability, which leads to very limited practical applications.

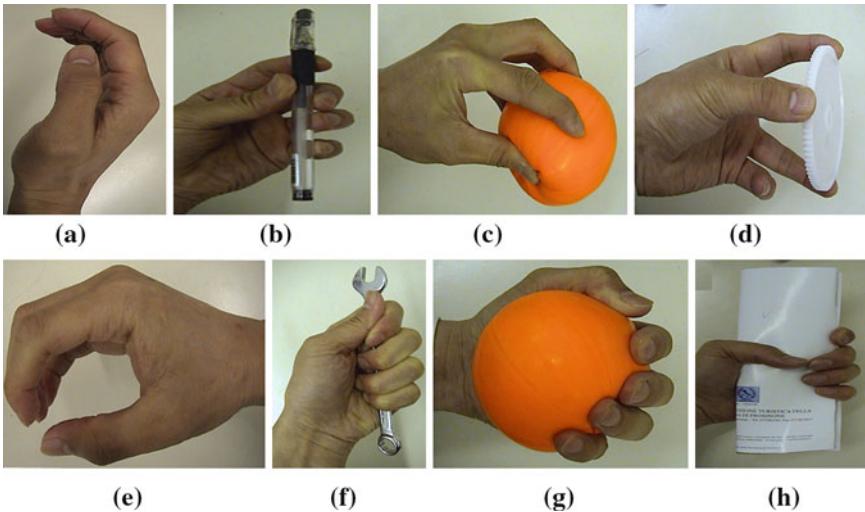
In order to make robotic hands have features of light weight, easy control system and low cost, in addition, with good grasping and operation capability, it is necessary to design a new kind of robotic hand which has more fingers than mechanical grippers (assuming that mechanical grippers just have two fingers) and less DOFs than dexterous hands. A suitable and effective way is to increase the number of fingers and pull in new mechanism to mechanical grippers to design the new robotic hand.

As the number of fingers increasing, the control system and mechanical design become more complex. Thus, the number of fingers should be as small as possible under the premise of functionality. In general, a robotic hand needs nine DOFs to move a target to any position and orientation [18]. And the best joint number of a finger is three. Thus, a robotic hand with three-joint fingers is the best solution when considering both economic and functional factors.

Besides the problem of finger number, the other key problem should also be solved, which is: how to make a robotic hand have more working modes with just three fingers? Since human hand has evolved to be a complex and adaptable dexterous tasks system, it should be a good way to find inspiration through studying the structure of a human hand.

### 7.2.1 Analysis of Grasp Mode in Human Hands

Human hand is a very good example of how proper grasp modes can be achieved in versatile tasks by using a combination of fingertips, joints, and phalanges. Grasp taxonomy of human hand has been given, for example in [19], which provides a classification according to the precision of the task and geometrical characteristics of the objects. Usually, there are two main kinds of grasp modes: pinching grasp and enveloping grasp as shown in Fig. 7.1a–h, respectively. Thin objects usually



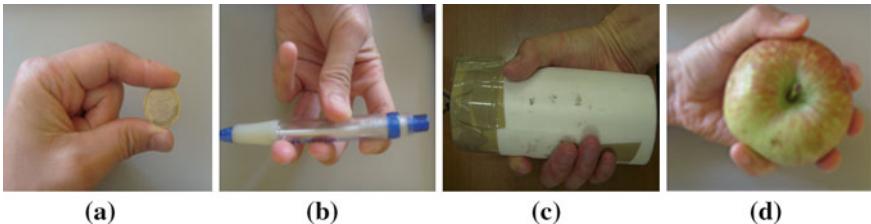
**Fig. 7.1** A grasp taxonomy of human hand: **a** pinching grasp, **b** grasp a pen, **c** grasp a sphere, **d** grasp circle, **e** enveloping grasp, **f** grasp a spanner, **g** grasp a sphere and **h** grasp a book

require dexterous operations. Thus, they should be grasped with pinching grasp mode. Voluminous or massive objects need to resist wrenches in any direction, so they should be grasped with enveloping mode.

In practical applications, human beings can choose proper grasp configuration and quantity of contact points in versatile grasps because of the versatility of fingers and palm. Grasp experiments have been carried out to experience the grasp taxonomy according to the grasp configuration and quantity of contact points for designing robotic hands. Figure 7.2 illustrates examples of proper grasp points and configurations in human grasp with different objects with the aim to identify feasible grasp modes also for a robotic hand by looking mainly at pinching and enveloping grasps.

A mapping between geometric features and grasp modes has been obtained from those tests by considering main characteristics of objects in terms of size and shape [16]. Sizes are given by width  $d$ , length  $l$ , and height  $h$  of the object volume. The finger maximal grasping capability is identified as  $r_o$ , in which  $r$  is the radius of a circular object. Shape of object can be oblong, circular, spherical, and cylindrical. Grasp taxonomy of human hand has been identified by means of grasp experiments in Fig. 7.2 and a grasp mapping can be defined as shown in Table 7.1.

As shown in Fig. 7.1, both pinching and enveloping grasps have two kinds of modes in other words. These two different grasp modes are called parallel-grasp and centripetal-grasp. The first one is called parallel-grasp mode because all fingers are lined up except the thumb. A parallel-grasp mode can be identified by a configuration in which the thumb and other fingers together grasp an object by acting in parallel to each other. This grasp mode includes two-finger parallel pinching, three-finger parallel pinching, and three-finger cylinder enveloping as



**Fig. 7.2** Experiments on human hand grasp: **a** pinching grasp of an iron coin, **b** pinching grasp of a pen, **c** enveloping grasp of a cylinder and **d** enveloping grasp of an apple

**Table 7.1** A mapping between grasp modes and object characteristics from the experiments in Fig. 7.2

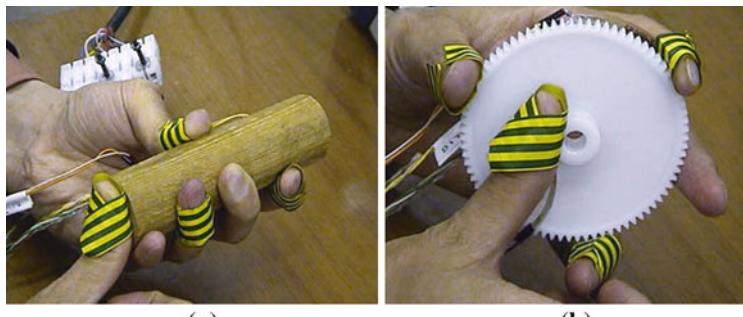
Size ranges of object	Shape of object	Type of object	Grasp mode
$h < r_o$	$d < r_o, l < 3 r_o$	Oblong	Iron cone
	$d < r_o, l > 3 r_o$	Oblong	Three-finger parallel pinching
	$r \leq r_o$	Circular	Plastic gear
$h > r_o$	$r > r_o, l < 3 r_o$	Spherical	Three-finger centripetal pinching
	$r > r_o, l > 3 r_o$	Cylindrical	Three-finger sphere enveloping
			Three-finger cylinder enveloping

shown in the examples in Fig. 7.2a–c, respectively. The other mode is called centripetal grasp mode since in the grasping configuration all the fingers including the thumb are located around the object surface. This grasp mode includes three-finger centripetal pinching and multi-finger sphere enveloping as shown in the examples in Figs. 7.1g and 7.2d, respectively.

In particular, a pinching grasp is characterized by pinching the object with a combination of thumb, index, and middle finger. Contact points are located at the fingertips of the fingers [16]. The thumb is opposite to the other fingers. In generally, the goal of a precision grasp can be recognized in grasping and manipulating an object with dexterous features. Thus, precision requires small grasp forces, full manipulability, and isotropy. Furthermore, location of contact points can be a critical issue since a limited number of contacts.

An enveloping grasp is characterized by having multiple contact points among grasped object, fingers, and hand palm [16]. It can maximize load capabilities and show a high stability, which is due to the large number of contact points that are distributed on the surface of grasped object. Heavy wrap configurations are the most powerful grasps, but they cannot be achieved with dexterous features.

Another experiment test has been carried out to illustrate the stability of these two basic grasp modes of human hand, although studying the needed force for moving and rotating fingers in robotic hands. In order to acquire accurate information of grasp force in each grasp mode, four contact force sensors have been attached respectively to each finger of a human hand. Three of them have been installed on the surface of the thumb (finger 1), the index finger (finger 2), and the little finger (finger 3), respectively. These fingers have been used to grasp an object

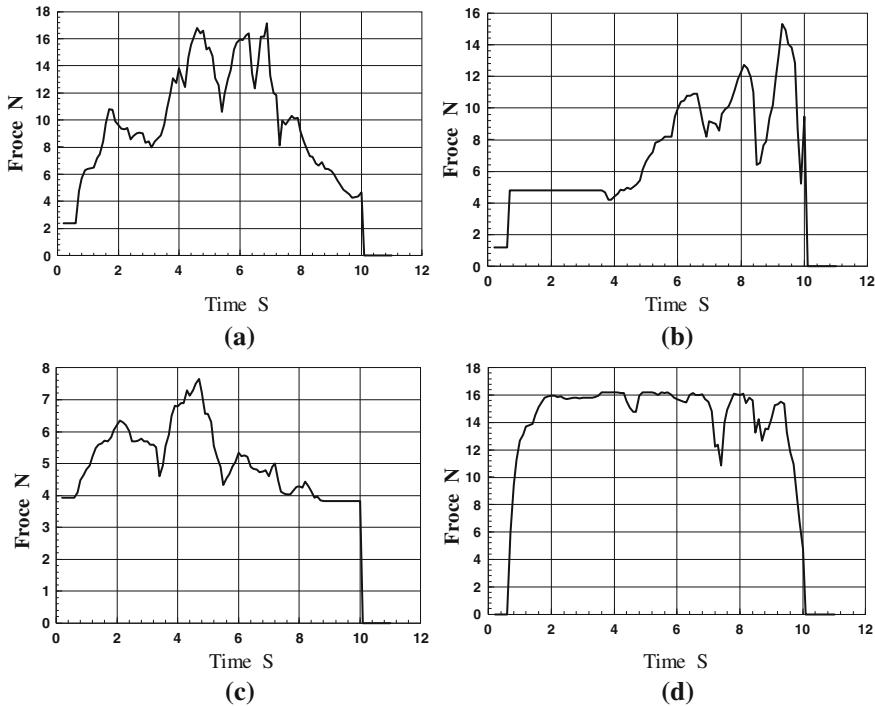


**Fig. 7.3** Experiments with human grasping: **a** three-finger parallel pinching grasp of a cylinder made of wood and **b** three-finger centripetal pinching grasp of plastic gear

with a three-finger parallel pinching grasp in Fig. 7.3a and with a three-finger centripetal pinching grasp in Fig. 7.3b. The thumb of other hand (finger 4) has been equipped with another force sensor and has been used to push the object as an external force in each grasp mode.

Figure 7.4 shows measured force results in an experiment test for grasping a cylindrical object in Fig. 7.3a. When given the external force by finger 4 as shown in Fig. 7.4d, the grasping forces measured from fingers 1, 2, 3 are showed in Fig. 7.4a–c, respectively. Fingers 1, 2, and 3 are distributed around the object boundary according to the parallel-grasp mode to reach force equilibrium. When the external force from the finger 4 is directed along the direction of the axis of the cylinder like in the case of Fig. 7.3a, the resultant force vectors that are exerted by the other three fingers cannot resist the external force (wrench). Thus, the object slips out from the hand. This experiment indicates that this grasp mode is not easy to reach stable grasp state, because some DOFs of the grasped object cannot be constrained. Figure 7.5 shows the force results of a relatively stable grasp mode like in the case of Fig. 7.3b. In the same way, when given the external force by finger 4 as shown in Fig. 7.5d, the grasping forces measured from fingers 1, 2, 3 are showed in Fig. 7.5a–c, respectively. Fingers 1, 2, and 3 are distributed around the boundary of a circular object. When the external force is directed along the radial directions of a circle-shaped object as shown in Fig. 7.3b, the three fingers can adjust their forces and direction to resist an external force and achieve equilibrium grasp quickly. The circular object cannot easily slip out from hand in this grasp mode and the grasp can be in an equilibrium state all the time.

These experiments verified that human hand can perform different grasp modes in grasping objects with different kinds of shapes and sizes. In particular, it is known that the three-finger centripetal-grasp mode is very efficient in robotic hands to resist an external wrench especially for sphere-shaped objects. It can ensure grasp stability and can be efficient to achieve force-closure grasp equilibrium by using minimum level of forces. Thus, it can be convenient to rearrange a grasp mode to a three-finger centripetal-grasp mode by moving and rotating one or more fingers.



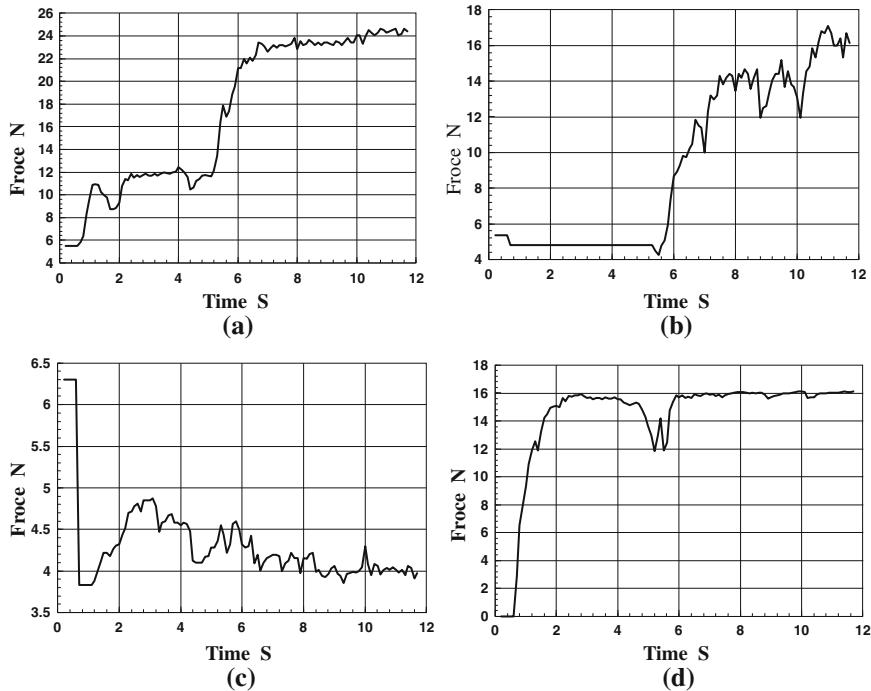
**Fig. 7.4** Measured grasping forces and external forces during the grasping by a human hand with three-finger parallel-grasp of a cylinder grasp mode in Fig. 7.3a: **a** force on finger 1, **b** force on finger 2, **c** force on finger 3 and **d** external wrench on finger 4

### 7.2.2 Orientation Activity in Grasping of Robotic Hands

With respect to human hands, most of robotic hands can grasp objects only with the three-finger parallel mode. They cannot change posture of the fingers to grasp a circle-shaped object with a uniform distribution of contact points to constrain the object since finger positions are fixed and only finger joints can rotate. When the object is grasped with a cylinder mode, the DOF along the axis of cylinder cannot be constrained. There are no contact points to resist an external wrench or force along the axis of cylinder.

On the other hand, some other existing robotic hands just can perform three-finger sphere enveloping grasp. In this grasp mode, the contact points are uniformly distributed around the boundary of sphere-shaped object. The robotic hand can resist an external wrench in all directions much better than in three-finger parallel grasp mode. In fact, this type of grasp mode is thought to be the most efficient and useful grasp mode for sphere or circle-shaped objects.

However, a robotic hand, which has only one grasp mode, whether three-finger parallel grasp mode or three-finger enveloping grasp mode, is not desirable in



**Fig. 7.5** Measured grasping forces and external forces during the grasping of the objects by a human hand with three-finger centripetal-grasp mode in Fig. 7.3b: **a** force on finger 1, **b** force on finger 2, **c** force on finger 3 and **d** external wrench on finger 4

practical applications. It should have at least those two grasp modes as mentioned above, or have as many effective grasp modes as possible.

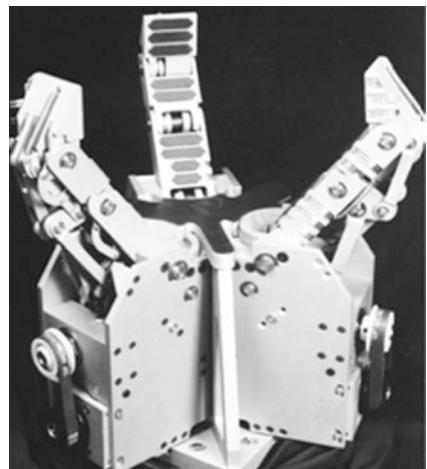
In order to solve this problem, it is valuable to look back at human hands. As can be seen in Figs. 7.1 and 7.2, grasp mode of human hand is realized through the corporation of not only fingers but also palm. In particular, thumb plays an important role in grasp modes. The most obvious feature in different grasp modes is that the thumb can rotate, which enables it to reach a suitable posture with respect to the other fingers to form variable grasp modes.

Thus, in the design of robotic hand, in order to make the hand have as many grasp modes as possible, it is necessary to bring in orientation mechanisms which can enable some of the fingers make rotation motions with respect to the others.

### 7.3 Prototypes Introduction

In fact, some researchers have realized the importance of bringing in orientation mechanisms into the structure of robotic hand. Here, several typical examples have been illustrated through aspects of hand structure, orientation mechanism, grasp

**Fig. 7.6** MARS hand  
(Reprinted with kind  
permission of Springer  
Science and Business Media)

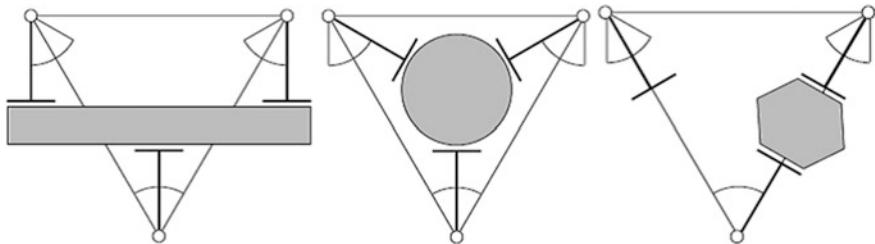


modes, experiments, and simulations. These examples are: Laval underactuated hand from Laval University, Institute of Advanced Manufacturing Technology (IAMT) Robotic hand, and Laboratory of Robotics and Mechatronics (LARM) robotic hand IV.

These robotic hands have several similar characteristics: first, all of them are underactuated robotic hands; second, each hand has three fingers; and third, each finger has three joints. In addition, they are low cost in manufacturing and easy operation in control due to few numbers of DOFs. Furthermore, they can perform several different kinds of working modes within the help of orientation mechanisms, which will be described in details.

### 7.3.1 *Laval Underactuated Hand*

Laval University has built several underactuated hands prototypes. The first underactuated hand built at Laval University is the Minimally Actuated Robotic System (MARS) hand [20], as shown in Fig. 7.6. This prototype was built as a test bed in order to experiment underactuation in fingers and study grasping strategies [21]. The surface of the fingers is flat since this is simple to manufacture and well suited for grasping. The tip of the fingers is not rounded, which is often seen in other robotic hands, in order to allow the grasping of small objects on a flat surface during pinch grasps. The fingers have a total length of 16.5 cm from the first joint to the tip and a width of 4 cm. The lengths of the phalanges (from base to tip) are 7, 5, and 4.5 cm, respectively. The distance from the centroid to the vertices of the equilateral triangle constituting the palm is 6 cm. Each of the three fingers is mounted on top of an actuation module. These modules are mounted on a main-frame and can be independently rotated by 60°, in order to obtain different grasp configurations, as illustrated in Fig. 7.7.



**Fig. 7.7** MARS finger orientations

The actuation module comprises an actuator (a DC brushless motor) driving the opening and closing motion of a finger. The transmission of the actuator to the finger is composed of a ballscrew in order to obtain large forces, and a timing belt in order to obtain compact modules and allow the modification of the transmission ratio. Each module also comprises a small gear motor to drive the orientation of the finger. The transmission is performed via a gear attached to the actuator and another attached to the mainframe.

Since each finger is driven by two actuators (one to drive the phalanges and other to change the orientation of the finger) the hand has a total of six actuators. Each finger has three grasping DOFs and one orientation DOF for a total of 12 DOFs for the hand. The total mass of the hand is approximately 9 kg. The parts are made of aluminum, the shafts and screws are made of steel, and the bushings are made of reinforced nylon.

Another underactuated robotic hand named Self-Adaptive Robotic Auxiliary Hand (SARAH), which has ten DOFs and two actuators, having the ability to couple the finger orientation with respect to the base is introduced as well [22].

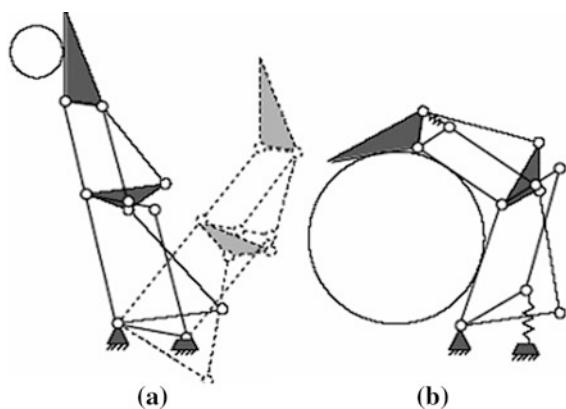
In order to further reduce the number of actuators and hence decrease the complexity of the controller required, the SARAH-class hands, have been built based on these two improvements. SARAH is an acronym for Self-Adaptive Robotic Auxiliary Hand, as shown in Fig. 7.8.

In addition to using underactuation in the fingers, introducing underactuation among them and coupling the orientation of the fingers leads to a robotic hand with ten DOFs and only two actuators. One actuator is used to drive the underactuation system which controls the opening/closing of all three fingers. The second actuator is used to drive the orientation of the fingers. Figure 7.9 illustrates a three-DOF underactuated finger mechanism [23]. It is composed of two parallelograms mounted in series. This mechanism is coupled to the phalanges of the finger but not to the other links of the shape adaptation mechanism (it is moving on a parallel plane). Two mechanical limits with springs at the top and bottom ends of the mechanism allow precision grasps to be performed and the adaptation to power grasps if necessary. Each of the three identical fingers of SARAH is mounted on an additional revolute joint whose axis is located on the vertex of an equilateral triangle and oriented normal to the plane of the triangle. With these additional revolute joints and the orientation mechanism shown in Fig. 7.10, the hands can be

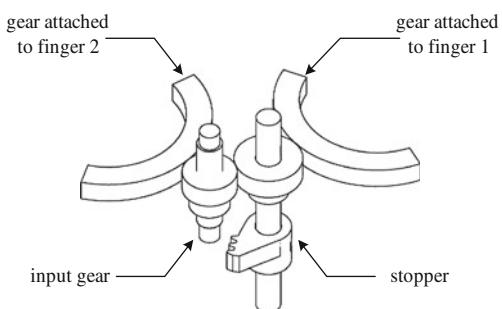
**Fig. 7.8** Self-adapting robotic auxiliary hand  
(Reprinted with kind  
permission of Springer  
Science and Business Media)

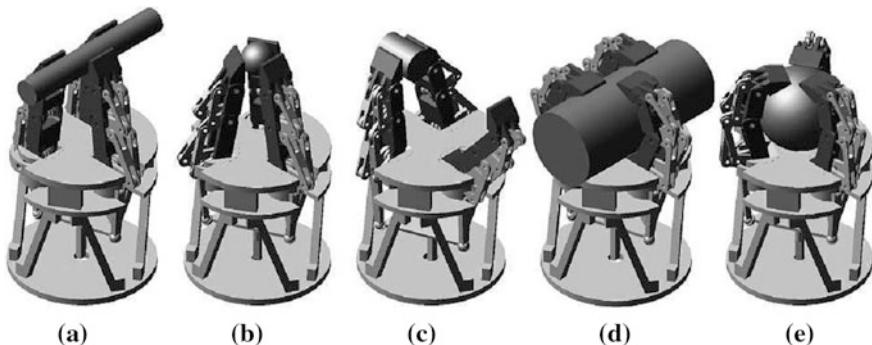


**Fig. 7.9** The parallel  
precision grasp mechanism  
(dark lines). **a** Parallel  
precision grasps and **b** power  
grasp



**Fig. 7.10** The orientation  
mechanism





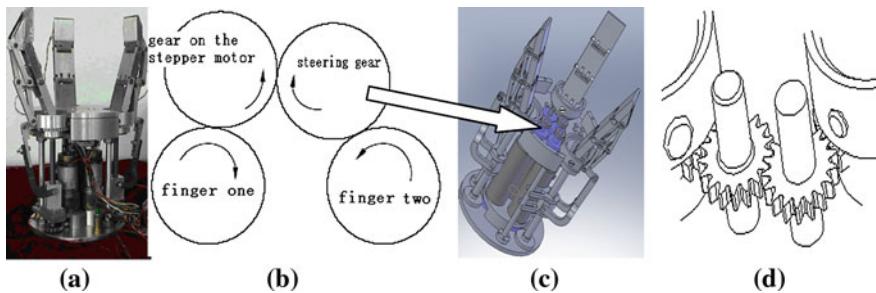
**Fig. 7.11** Typical grasp modes [21]: **a** precision cylindrical, **b** precision spherical, **c** precision planar, **d** power cylindrical and **e** power spherical

reconfigured by modifying the orientation of the fingers in order to adapt to the general geometry of the object to be grasped.

The main configurations of the fingers are obtained by orienting two of the fingers with a range of  $90^\circ$ , and the rotation of the two fingers is coupled by a geared mechanism, as shown in Fig. 7.10. The two sections of gears are attached to the two orientable fingers. In the planar configuration, where the third finger is not used, its closing motion is blocked at the hyper differential level by a stopper activated by the orientation mechanism. When appropriate, the teeth of the stopper will engage with one of the outputs of the differential. The purpose of the orientation transmission is to use one actuator to drive the two rotating fingers. This rotation is synchronized by the gearing system and the fingers rotate in opposite directions. The input gear, attached to the input shaft, directly drives the finger 1 gear. The second finger gear, driven via a free gear, is attached to a free shaft to obtain a rotation of the second finger in a direction opposite from that of the first finger.

The possible grasps are essentially the same as with the MARS hand. The main grasping configurations of the fingers available are thereupon: precision cylindrical, precision spherical, precision planar, power cylindrical, and power spherical as in Fig. 7.11a–e, respectively.

In the cylindrical configuration, two fingers point in the same direction while the third one points in the opposite direction and moves between the other two. This is the preferred configuration to grasp cylindrical/prismatic objects or objects where one dimension is significantly larger than those of the hand. In the spherical configuration, the three fingers are oriented toward the centroid of the triangle formed by the base joints of the fingers. This configuration is the preferred choice to perform enveloping grasps of objects with sizes close to the size of the hand or with no distinct particular axis. In the planar configuration, two fingers are directly facing each other and the third finger is not used. This configuration is used for pinch grasps, usually of objects with small dimensions.



**Fig. 7.12** IAMT I robotic hand and its orientation mechanism: **a** a prototype, **b** a scheme, **c** a 3D CAD mode and **d** a zoomed view of gear transmission

### 7.3.2 IAMT Robotic Hands

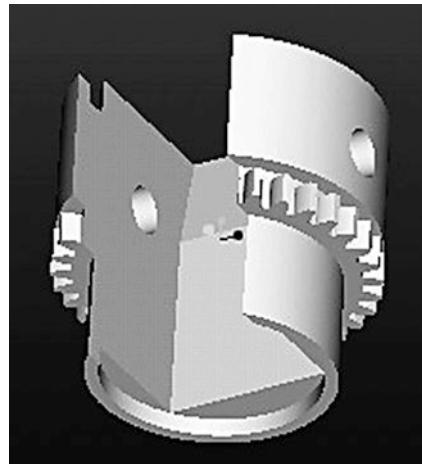
Having realized the importance of the orientation mechanism in robotics hands which conducts finger grasping or manipulate objects dexterously, IAMT has designed types of underactuated robotic hands with a novel orientation mechanism. Figure 7.12 illustrates a type of IAMT Robotic (denoted IAMT I Robotic) hand and its orientation mechanism. The hand has three-fingers with nine DOF and an orientation mechanism with 1 DOF, which can avoid the inharmonious action in SARAH and enhance the grasping stability [24].

The IAMT I Robotic hand has a total of four actuators, three of them are used to drive the opening/closing motion of all three fingers. The orientation mechanism is driven by one stepping motor, which is used to regulate the finger orientation and angle in order to form versatile grasping modes. A straight gear is fixed on the end of the stepping motor shaft, among one of the fingers and the stepping motor teeth, there is a tumbler gear to ensure that two fingers have relative movements, about two-third of the base root is directly processed out straight gear, and all these gears are installed in the same base plane as shown in Fig. 7.13. This is a universal mechanism which can be used in various types of robotic hands.

With the function of the orientation mechanism, the two fingers can rotate continuously around their frame axes by means of one stepping motor and transmission gears to achieve several grasp configurations. As for different objects and tasks, the grasp modes can be automatically switched between the pinching grasping and enveloping grasping depending on the elastic and mechanical limits, as shown in Fig. 7.14. Figure 7.15 reveals the versatile grasping modes of the IAMT I robotic hand.

The pinching grasping patterns include two-finger parallel pinching, three-finger parallel pinching, and three-finger centripetal pinching as shown in Fig. 7.14a–c, respectively. Enveloping grasping includes cylinder grasping and sphere grasping as shown in Fig. 7.14d and e, respectively. All these grasp modes essentially cover the grasp modes of human hand.

**Fig. 7.13** The 3D model of finger base



Experiments have been carried out through grasping objects with different kinds of shapes, dimensions, and materials, as shown in Fig. 7.15. Experimental results show that the IAMT I robotic hand has a good performance in doing versatile grasp tasks.

In order to further reduce the number of actuators, two levels differential mechanism is integrated in IAMT II, with the two stages differential mechanism, it can produce three outputs with one input, as illustrated in Fig. 7.16.

As showed in Fig. 7.17, the first stage differential gear train is composed of gears 1, 2, 3, 3' and planet frame  $H_1$ , from the input  $a$  to the output  $b$  and  $c$ , the motion is transmitted from the planet gear 3 and 3' to the two center gears 1 and 2. The second stage differential gear train is composed of gears 4, 5, 6, 6' and planet frame  $H_2$ ,  $c$  as the output in the first stage is denoted as the input in the second stage,  $d$  and  $e$  are the output.

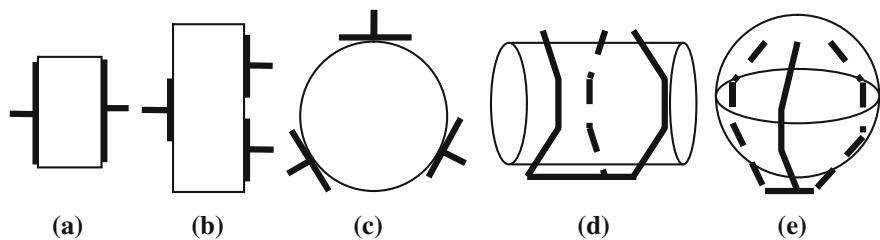
According to the formula of gear transmission ratio, transmission ratio  $i_{12}^H$  of the two center gears 1 and 2 in the first stage is:

$$i_{12}^H = \frac{\omega_1^H}{\omega_2^H} = \frac{\omega_1 - \omega_H}{\omega_2 - \omega_H} = \frac{n_1 - n_H}{n_2 - n_H} \quad (7.1)$$

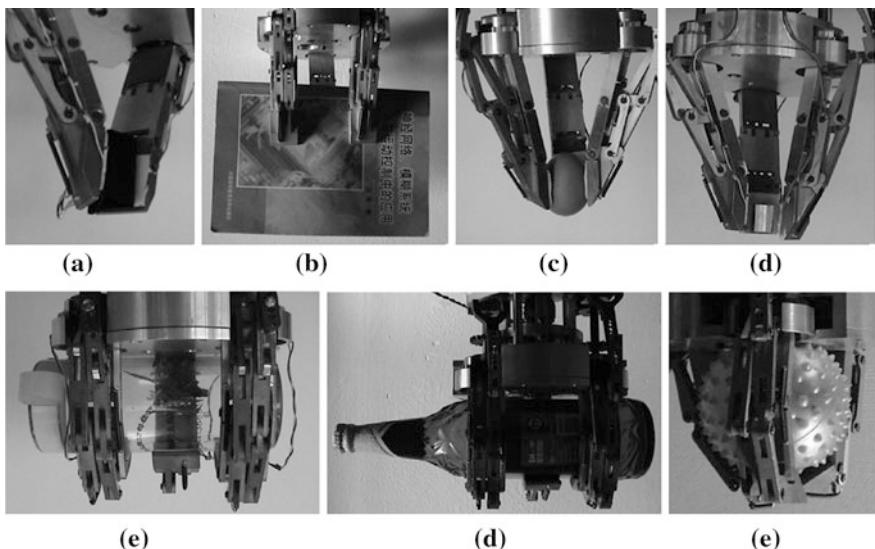
where,  $\omega_1, \omega_2, \omega_H$  are the angle velocities of gears 1, 2, and planet frame  $H_1$ ,  $n_1, n_2, n_H$  are the revolving speeds of gears 1, 2, and planet frame  $H_1$ .

Denote  $n_1/n_2 = k$ , according to the different values of  $k$ , it can get different values of  $i_{12}^H$ . Here in order to make three fingers have the same revolving speed, four bevel gears with the same size are choose to compose the first stage differential mechanism. The second stage differential mechanism is also composed of another four bevel gears with the same size.

The motions of the three differential output are transferred through the three gear pairs passed to the three screws, then the three screws which can change the rotation movement into a linear movement are used to drive the opening/closing



**Fig. 7.14** Versatile grasping patterns: **a** two-finger parallel pinch grasp, **b** three-finger parallel pinch grasp, **c** three-finger centripetal pinch grasp and **d** enveloping cylinder grasp; **e** enveloping sphere grasp

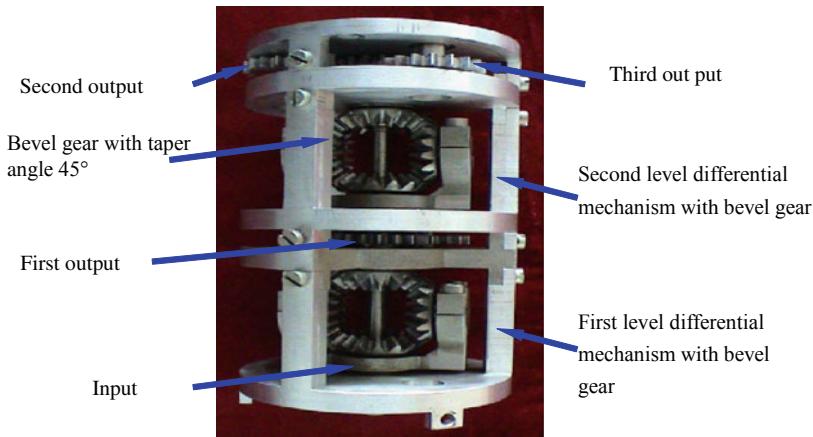


**Fig. 7.15** Experiments of versatile grasping tasks on objects: **a** a small cone, **b** a book, **c** an egg, **d** a small cylinder, **e** a cup, **f** a bottle of beer and **g** a rubber ball

motion of all three fingers. In order to make three fingers closed synchronization, the initial position of each finger should be the same.

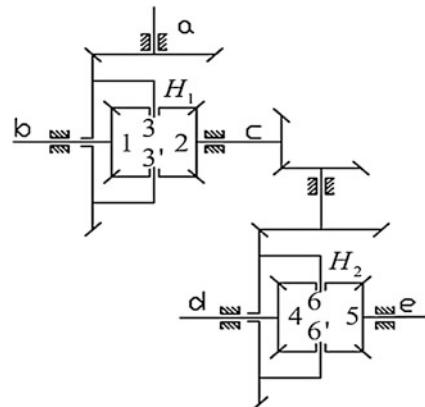
In the practical application, in order to save space, reduce weight, make the structure of the whole hand more compact, another principle of the differential transmission mechanism is given in Fig. 7.18. The input and output are arranged along the axial direction, among them *a* is the input, *b*, *d*, and *e* are the three outputs. It is obvious that such technology method requires high precision of processing and assembling, which leading to increase the difficulty.

This mechanism enables the robotic hand have ten DOFs with only two actuators, the prototype and grasping experiments are illustrated in Fig. 7.19.



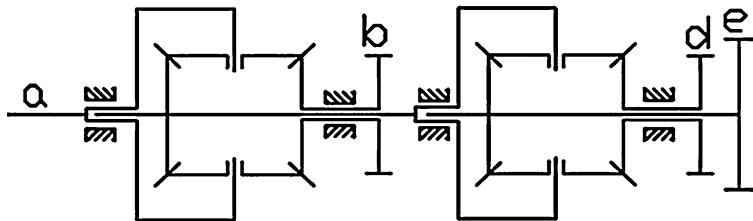
**Fig. 7.16** The two stages of differential mechanism with bevel gear

**Fig. 7.17** The principle of the differential transmission mechanism

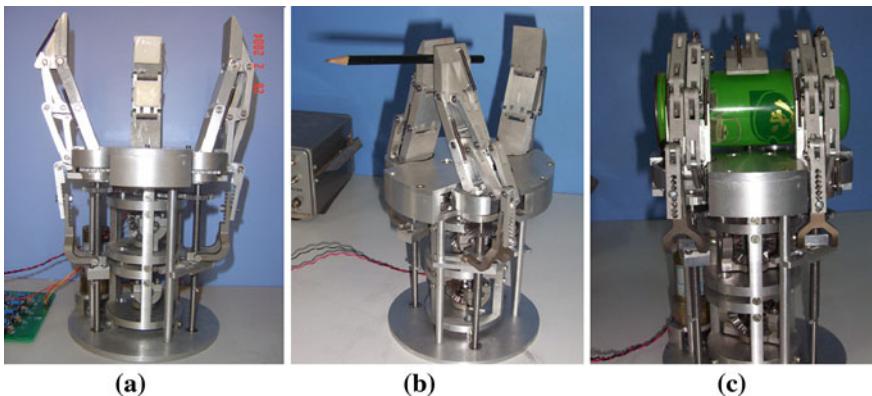


### 7.3.3 LARM Robotic Hands

Since late 1990s at LARM in Cassino, design and research activities have been carried out in order to design a multi-fingered robotic hand having low cost and easy-operation features, as outlined for example in [15, 17, 25, and 26]. Main design issues have been the development of a suitable driving mechanism and use of force sensors. In particular, a driving mechanism moves the three phalanxes of a finger with only one active DOF by mimicking a human-like cylindrical grasping. Moreover, the designed driving mechanism remains completely embedded in the finger body during the whole movement of the finger. The force sensors should be carefully selected and located in order to be implemented within low cost easy-operation force control architecture.



**Fig. 7.18** The final principle of the differential transmission mechanism



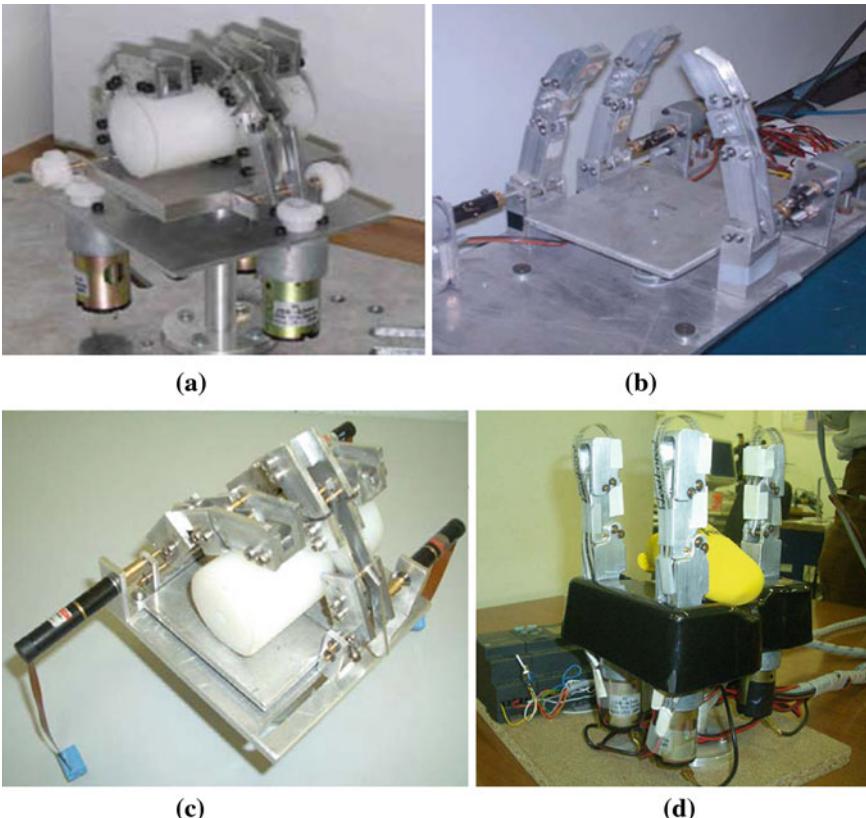
**Fig. 7.19** IAMT II robotic hands: **a** a prototype, **b** planar pinch grasp and **c** enveloping grasp

LARM hand is described by addressing its low cost easy-operation features since the design stage. Design evolution of LARM Hand is reported by describing peculiarities and differences among LARM Hand prototypes from version 1 to 4 (that is shown in Fig. 7.20). Advantages and drawbacks of LARM Hand design are described also with the aim of proposing a novel design for the driving mechanism with underactuation properties.

Since the four versions are similar in structure, here just take version IV as an illustrated example. The prototype LARM hand IV [27] is composed of three fingers with one DOF human-like design as shown in Fig. 7.20d. Each finger is an independent module that is constituted by a DC motor, a driving mechanism, and three human-like phalanges.

Grasp experiments have been carried out with LARM hand IV to test the grasp modes as in a general robotic hand, as shown in Fig. 7.21 [16]. The finger kinematics for a general robotic hand is described in Fig. 7.22. Joints 1, 2, and 3 are driven by linkages and actuated by one motor that is indicated by the torque  $\tau_1$  in Fig. 7.22. A driving moment is transferred from the first joint to fingertip frame in the  $xz$  plane of finger configuration.

Generally, the motion of a finger with its phalanges occurs in a plane, as shown in Fig. 7.22, and the actuating torque for each phalanx is still in that plane.

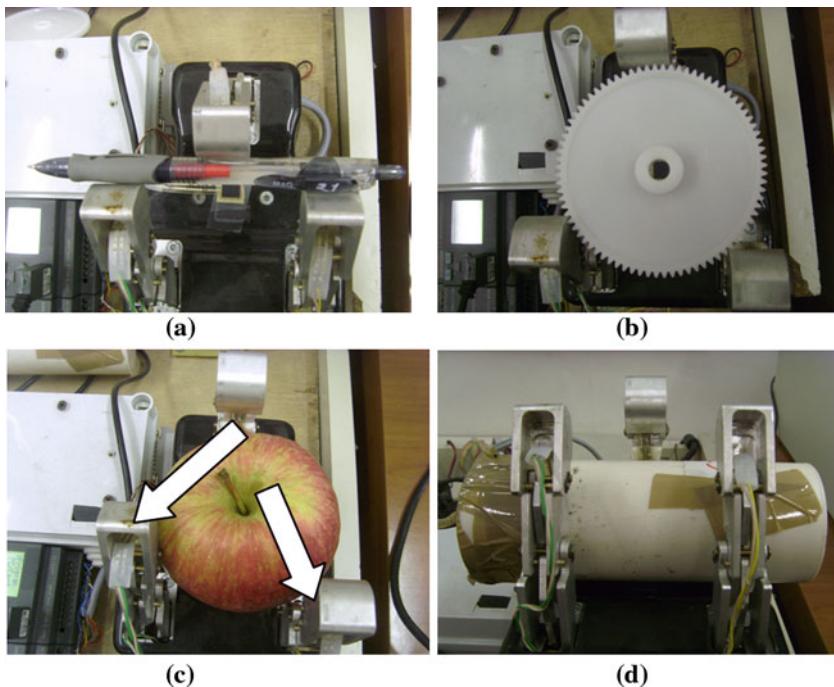


**Fig. 7.20** LARM hand serial: **a** version I, **b** version II, **c** version III and **d** version IV

Thus, with respective phalanx plane if a finger is rotated its plane of action, it could not be in the proper orientation to give a proper grasp action for a stable grasp. The three fingers can form a configuration of triangle grasp, as in the case of Fig. 7.23a, but the contacts are located at the corners of the phalanxes with a torque action about the finger axis. A proper grasp configuration will require the fingers to move and rotate in order to achieve a correct grasp configuration like the example in Fig. 7.23b. Such a finger configuration will not damage both the fingers and object.

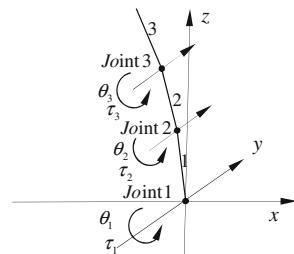
Figure 7.21c shows LARM robotic hand IV that grasps an apple with a three-finger parallel-grasp mode. During the grasp configuration, the fingers contact the apple through sharp corners of phalanxes which are indicated as white arrows in the figure.

The contact forces cannot be exerted correctly to the apple within the finger plane. Thus, this grasp mode can damage the apple and finger simultaneously, besides the fact that will not ensure a correct grasp.

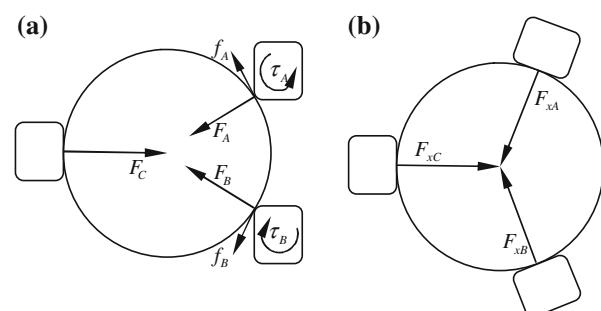


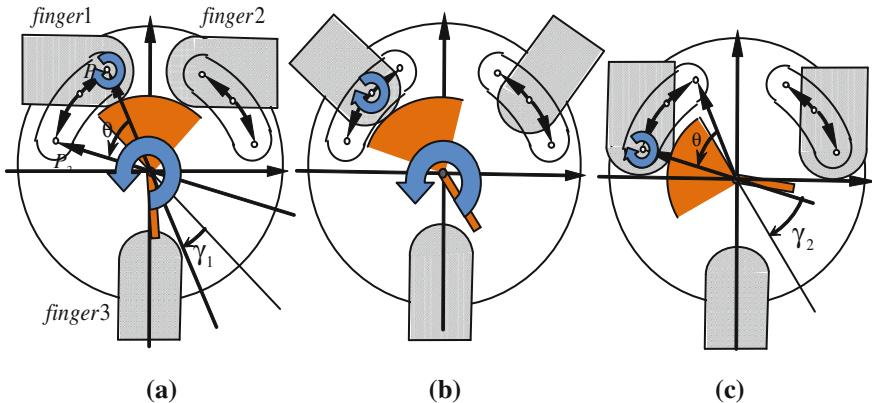
**Fig. 7.21** Laboratory experiments with LARM hand IV: **a** pinching grasp of a pen, **b** three-finger centripetal-grasp of a plastic gear, **c** enveloping grasp of an apple and **d** enveloping grasp of a cylinder

**Fig. 7.22** The finger kinematics of a general three-phalanx robotic finger



**Fig. 7.23** Schemes of stable grasp configurations and force distributions: **a** three-finger parallel-grasp and **b** three-finger centripetal-grasp





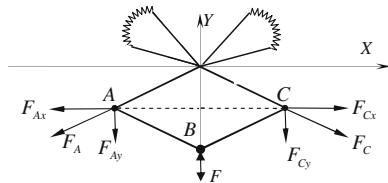
**Fig. 7.24** Finger configurations and positions of fan-shaped gears of the FRM: **a** two-finger parallel-grasp mode (starting configuration), **b** three-finger centripetal-grasp mode and **c** three-finger parallel-grasp mode (ending configuration)

A novel contribution should be considered in solving the above-mentioned problem in order to obtain reasonable grasp modes by using a unique mechanism within the existing LARM hand IV. The required mechanism can be characterized by a simultaneous motion for moving and rotating the fingers to obtain a three-finger centripetal-grasp mode with pinching and enveloping grasp capabilities. A symmetrical structure and one motor drive make this special mechanism design as a unique and novel solution for construction and assembly. A design peculiarity must be to obtain the system operating easily and cost low in order not to complicate the further design of fingers and hands. Such a mechanism will increase the number of grasp modes of a robotic hand. A general layout of a proposed mechanism is shown in Fig. 7.24.

The finger relocating mechanism (FRM) is assembled beneath the palm, and it will occupy a little space in the mechanical design of a robotic hand. The motor is installed beneath the palm as well. Through a slot the motor transmits the driving force to the four-bar linkage. The FRM is composed of a fan-shaped gear and a folding linkage. The fan-shaped gear in Fig. 7.24 is designed to cut down redundant plane of drive gear and to reduce the weight of mechanism as limited to the requested finger motion for the starting configuration shown in Fig. 7.24a and ending configuration shown in Fig. 7.24c. The work area of the gear section is more than angle  $\theta$ , because each fan-shaped gear needs an additional angle  $\gamma$  to ensure proper engaging of the two gears,  $\gamma_1$  is the additional angle of the fan-shaped gear in starting configuration and  $\gamma_2$  is the additional angle in ending configuration as shown in Fig. 7.24. Thus, the area rate  $p$  between the fan-shaped gear and a full gear can be computed as

$$p = (\theta + r_1 + r_2)/360 \quad (7.2)$$

**Fig. 7.25** A scheme for force analysis in the folding linkage



The starting configuration of the finger is shown in dashed line area in Fig. 7.24a. The finger is attached on the planetary gear. When the fan-shaped gear moves with anticlockwise motion, the finger will move and rotate clockwise. By using proper positions of the fan-shaped gear, finger positions can be arranged properly to achieve the main grasp modes of human hand.

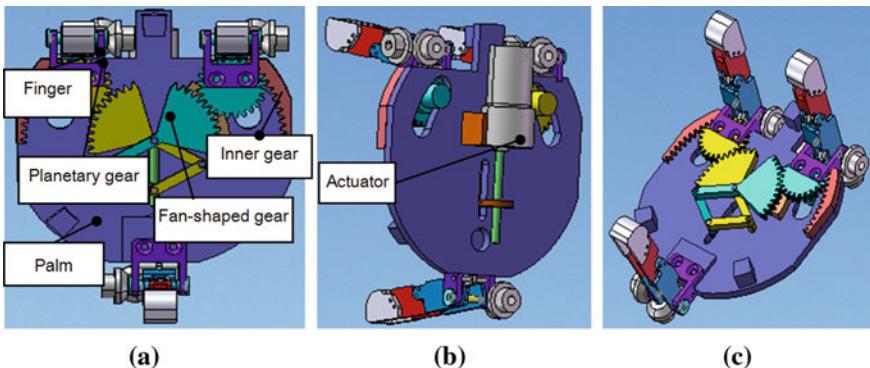
In Fig. 7.24, three grasp modes are shown as obtained by rotating the fan-shaped gears to suitable positions. In Fig. 7.24a, the grasp mode denotes the two-finger parallel-grasp mode. Robotic hand can grasp objects with two-finger parallel-grasp mode. In Fig. 7.24b, the grasp mode is a three-finger centripetal-grasp mode, and Fig. 7.24c shows the three-finger parallel-grasp mode. A folding linkage is used to drive the fan-shaped gears as a transmission unit that is connected to the actuators. The folding linkage for rotating and moving the fingers is a symmetrical structure design for symmetrical operation. This linkage design is also aimed to simplify the complexity of the system and to limit the actuation. This will be understood from a force analysis of the linkage action as referring to Fig. 7.25. Along the  $X$ -axis direction, all the joint forces are balanced from each other because of the symmetrical design.

Points A and C are symmetrical by  $Y$ -axis. Force  $F_A$  exists in point A and force  $F_C$  exists in point C, both  $F_A$  and  $F_C$  can be decomposed along  $X$ -axis and vertical axis ( $Y$ -axis), as illustrated in Fig. 7.25. At point B, driving force  $F$  only exists along  $Y$ -axis but does not exist along  $X$ -axis. According to the theory of force balance, the equation can be deduced along  $X$ -axis as

$$F_{Ax} = F_{Cx} \quad (7.3)$$

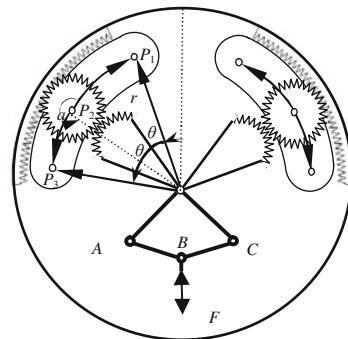
Likewise, the compound force along  $Y$ -axis is balanced from each other. Therefore, the FRM can act easily when the folding linkage reaches force balance. Although the proposed mechanism design is suitable for a robotic hand like the LARM hand IV, it can be applied to any gripper and robotic hand even with no constraints for the structure of the fingers.

Two inner gears are symmetrically fixed on the edge of the palm. Two small planetary gears can be driven from an initial position to a final position within a slot having circular shape. Two fingers are connected to the above-mentioned planetary gears. Then, the planetary gears are driven by a fan-shaped gear and the inner gears as shown in Fig. 7.26a. The fan-shaped gears can rotate about the center of the palm since they are connected to the links of the folding linkage. The fan-shaped gears are part of the link bodies. The folding linkage is a four-bar mechanism that is driven by one actuator acting at point B of Fig. 7.25. The



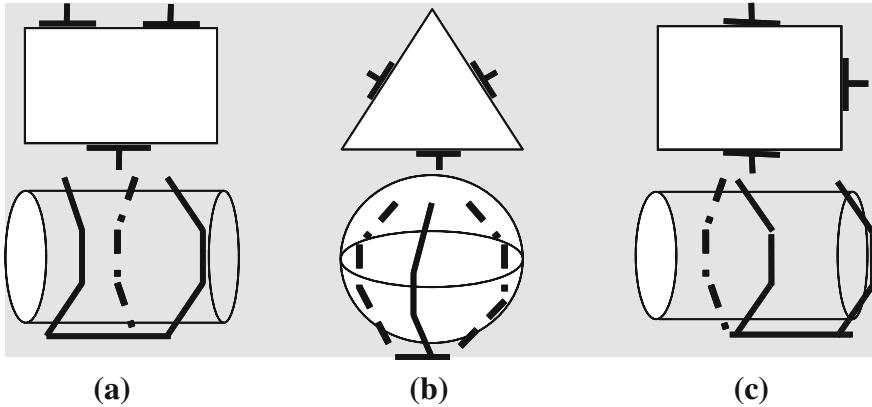
**Fig. 7.26** A 3D CAD model of the proposed FRM: **a** the mechanical design, **b** the actuation layout and **c** CAD view of the proposed design as applied to LARM hand IV

**Fig. 7.27** Simplified structure of rotating and moving mechanism



mechanical configuration of the actuator is shown in Fig. 7.26b. Therefore, this solution makes possible to actuate both fingers simultaneously and to move them symmetrically. Figure 7.26c shows the isometric view of the CAD design as applied to LARM hand IV. A proper installation in a robotic hand and a suitable design of the FRM requires a careful choice of the gear angle  $\theta$ , its initial value  $\theta_0$  and the rotation angle  $\alpha$  of each finger in Fig. 7.27. When fingers need to change the grasp mode, the action of the folding linkage makes the fan-shaped gear continuously change  $\theta$  and  $\alpha$  simultaneously.

At initial position with  $\theta = 0$ , the grasp mode is the three-parallel pinching grasp (plane) or cylinder (enveloping) grasp, just like in Fig. 7.28a. Then, with  $\alpha = 60^\circ$ , the grasp mode changes to a three-finger centripetal-grasp or sphere grasp, like in Fig. 7.28b. Finally with  $\alpha = 90^\circ$ , the grasp modes changes to two-finger parallel-grasp, like in Fig. 7.28c. For a satisfactory grasp, contact points must be distributed properly on the object boundary. In addition, force equilibrium should be guaranteed in a way to hold grasp stability as well. For these goals, a minimal requirement can be considered to be a force-closure condition. A grasp



**Fig. 7.28** Grasp configurations: **a** three-finger-parallel grasp, **b** three-finger-centripetal grasp and **c** two-finger parallel grasp

is considered to be stable if for any external wrench  $\omega$  (including contact force  $f_i$ , the arm of force  $R_i$  and torque  $\tau_i$ ) the fingers can exert a set of zero-pitch wrenches  $\omega_i$ , with  $i = 1, \dots, n$ . This can be expressed as

$$\omega + \sum_{i=1}^n \omega_i = 0 \quad (7.4)$$

in which

$$\omega_i = \begin{bmatrix} f_i \\ \tau_i \end{bmatrix} = \begin{bmatrix} f_i \\ R_i \times f_i \end{bmatrix} \quad (7.5)$$

Thus, a stable grasp must satisfy the following conditions:

- (1) Each contact force must satisfy the friction constraints within the coulomb friction cone  $C_i$ .
- (2) The grasp can achieve the force and torque equilibrium through the contact force redistribution when there are changes in external wrenches.

In this design, by using the FRM the main aim is to increase the stability of the three-finger centripetal-grasp mode. A stable grasp should be able to resist wrenches in any direction. A grasp stability measure has been proposed as in [28]. A stability index is defined as the amount of the smallest wrench (in any direction) that releases the grasp, when a force with unit strength is applied to each contact point. A stability measure can be computed by calculating the largest inscribing circle in the grasp wrench space.

At each contact point forces act within the friction cones as shown in Fig. 7.29. Each cone  $C_i$  is identified by two spanning vectors  $u_{1ci}$  and  $u_{2ci}$  as the areas that are shown in gray. In Fig. 7.29a, the cones  $C_2$  and  $C_3$  overlap each other. In the case of Fig. 7.29b, the three friction cones  $C_1$ ,  $C_2$ , and  $C_3$  are well differentiated. Thus, in

**Fig. 7.29** Schemes for stable grasp configurations and force distributions: **a** three-finger parallel-grasp and **b** three-finger centripetal-grasp

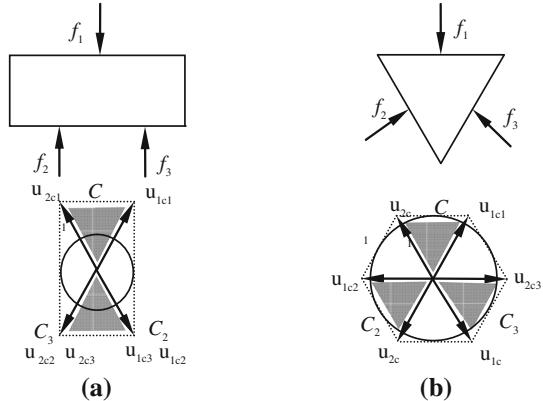


Fig. 7.29b, the radius of the largest inscribing circle in the grasp wrench space is much larger than that in the case of Fig. 7.29a. According to [28], the larger the inscribing circle is the more stable the mechanism will be. Therefore, the three-finger centripetal-grasp mode has better stability characteristics than the three-finger parallel-grasp mode.

Figure 7.30 is a simplified scheme of a gear transmission mechanism in the FRM. Here we mark the fan-shaped gear as gear 1, the planetary gear as gear 2 and the inner gear as gear 3.  $O_1$  is the rotating center of gear 1, and it is also the center of inside circle of gear 3;  $O_2$  is the mass center of gear 2;  $O_3$  is the contact point between gear 1 and gear 2,  $r_i$  ( $i = 1, 2, 3$ ) denotes the radius,  $n_i$  ( $i = 1, 2, 3$ ) denotes the angular velocity,  $z_i$  ( $i = 1, 2, 3$ ) denotes the teeth number,  $m_i$  ( $i = 1, 2, 3$ ) denotes the module of gears 1, 2, and 3, respectively. Here, we suppose  $H$  is the proposed linkage between  $O_1$  and  $O_2$  and  $n_H$  denotes its angular velocity.  $D$  is the instantaneous contact point between gear 2 and gear 3.  $O_2$  rotates about the point  $D$ , while  $O_3$  rotates about the point  $O_1$ . Points  $O_3$  and  $O_2$  have the same velocity  $V$  that is given by

$$V = n_H(r_1 + r_2) = n_2 r_2 \quad (7.6)$$

Since pitch circle diameter can be computed as  $r_i = 1/2m_i z_i$ , Eq. 7.6 can be rewritten as

$$n_2/n_H = (r_1 + r_2)/r_2 = (z_1 + z_2)/z_2 \quad (7.7)$$

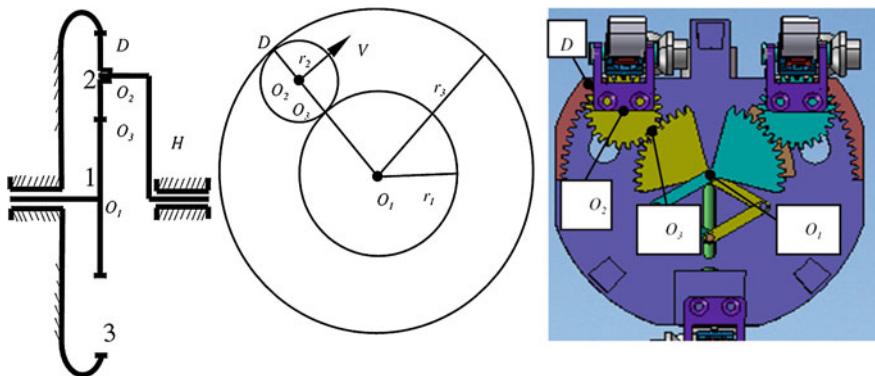
At the same time, the expression for the transmission ratio can be considered as

$$n_2/n_H = \alpha/\theta \quad (7.8)$$

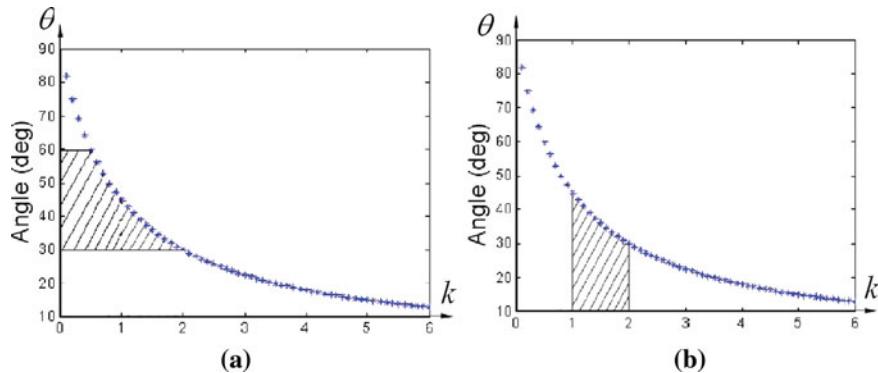
Thus, from the Eqs. 7.7 and 7.8, angle  $\theta$  can be given as

$$\theta = z_2 \alpha / (z_1 + z_2) \quad (7.9)$$

When  $k$  is the ratio of teeth number of gear 1 and gear 2 as  $k = z_1/z_2$ , Eq. 7.9 can be rewritten as



**Fig. 7.30** A scheme of the gear transmission mechanism



**Fig. 7.31** Feasible solutions for the gears design with Eq. 7.11 according to: **a**  $k$  limited range and **b**  $h$  limited range

$$\theta = \alpha/(k + 1) \quad (7.10)$$

With  $k$  constant, a fixed relationship is established between the translation angle  $\theta$  and the rotation angle  $\alpha$  in the finger motion.

According to the requirements of the proposed design, the angle  $\alpha$  can be conveniently fixed at  $90^\circ$ . Figure 7.31 shows plots for the expression

$$\theta = 90^\circ/(k + 1) \quad (7.11)$$

where feasible angles are indicated for practical implementation. In fact, not all the angles are suitable for the mechanism design, since a proper initial angle  $\theta_0$  is needed to avoid interference between the two fingers. Thus, in order to obtain three-finger centripetal-grasp mode, the angle of  $\theta + \theta_0$  must satisfy the

**Table 7.2** Feasible solutions for gear design in the FRM according to Fig. 7.31

$\alpha(^{\circ})$	$\theta(^{\circ})$	$\theta_0(^{\circ})$	$\theta + \theta_0(^{\circ})$	$k$
90	45	30	75	1.0
	36		66	1.5
	30		60	2.0

condition  $60^{\circ} \leq \theta + \theta_0 \leq 90^{\circ}$ . For a case with  $\theta_0 = 30^{\circ}$ , we can get  $30^{\circ} \leq \theta \leq 60^{\circ}$  as in Fig. 7.31a.

In addition, when considering the requirement of design size of engaging gears, the proportional coefficient  $k$  will be conveniently limited in a range  $1 \leq k \leq 2$ , as shown in Fig. 7.31b.

Numerical values of feasible solutions are listed in Table 7.2 according to the intersection of limited range for  $\theta$  and  $k$ . We can choose one of them as depending on practical needs.

In order to validate the correctness of the proposed design and its operation, simulations have been carried out by using the above-mentioned formulation and suitable configurations.

First, we give simulation of the angle and position of the rotating and moving mechanism, when the rotating angle is equal to  $90^{\circ}$ . We need to confirm whether when the finger rotates  $90^{\circ}$ , the finger moves for an angle  $\theta$  in the case (a)  $k = 1$ ;  $z_1 = z_2$  and (b)  $k = 2$ ;  $z_1 = 2z_2$ . The relationship between the translation and rotation of a finger can be computed by looking at the angles  $\theta$  and  $\alpha$ , respectively.

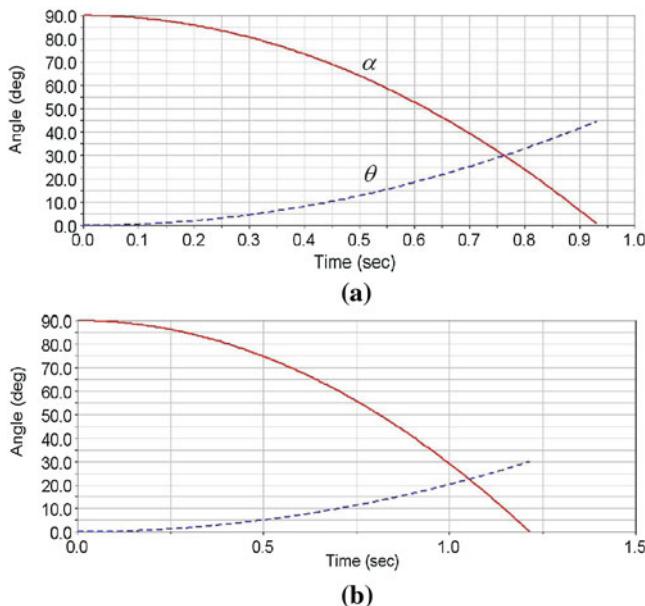
By using Eqs. 7.10 and 7.11, it is possible to compute output motion, as in the examples in Fig. 7.32 that refers to the case of  $90^{\circ}$  rotation of a finger. Practical cases are referred as referring to  $z_1 = z_2$  (Fig. 7.32a) and  $z_1 = 2z_2$  (Fig. 7.32 b). For the case in Fig. 7.32a, the finger rotation requires a translation with an angle of  $45^{\circ}$  and for the case of Fig. 7.32b an angle  $h$  of  $30^{\circ}$  is enough.

Figure 7.33 shows a model of the designed FRM in ADAMS environments, where main parts, contact points and contact forces are indicated.

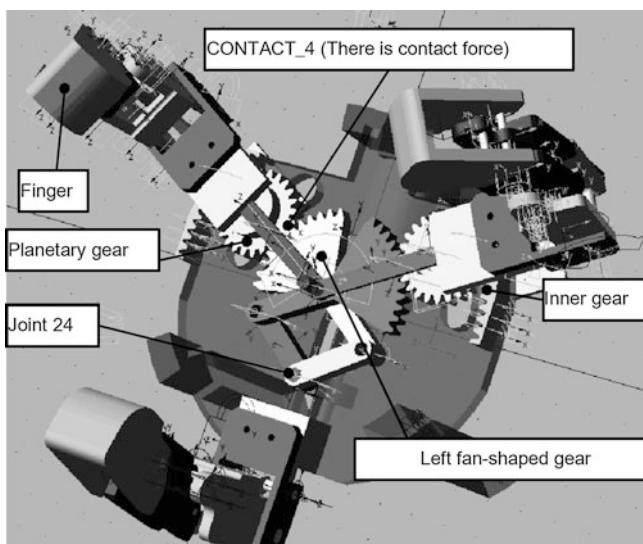
In Fig. 7.34, the three main grasp modes are presented by moving and rotating the finger posture for LARM hand IV. The computed simulation refers to a continuous motion of the FRM that gives a continuous relocation of fingers that can achieve the three main grasp models in Fig. 7.34.

The input action of the actuator is transmitted to the FRM at joint 24 in Fig. 7.33 (joint 24 is the same point as B shown in Fig. 7.27). The FRM moves from three-finger centripetal-grasp mode to three-finger parallel-grasp mode, and then to two-finger parallel-grasp mode. The corresponding simulation plots of angular velocity and angular acceleration velocity of the left fan-shaped gear are obtained as shown in Fig. 7.35. Because the driving torque at joint 24 is not constant, angular velocity and angular acceleration velocity of the left fan-shaped gear change from positive to negative and vice versa.

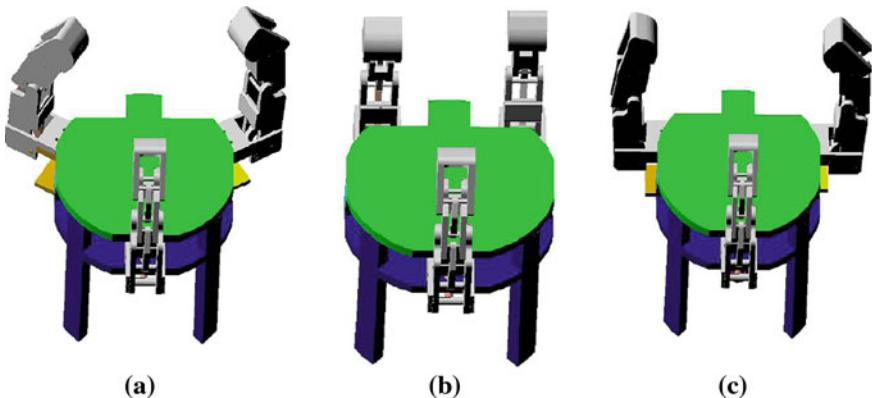
In Fig. 7.36, the driving torque at joint 24 in Fig. 7.33 is shown as computed for the simulation in Fig. 7.35. The driving torque at joint 24 pushes the folding linkage forward and backward. The oscillations in the plots are also due to the actions at the gears contacts.



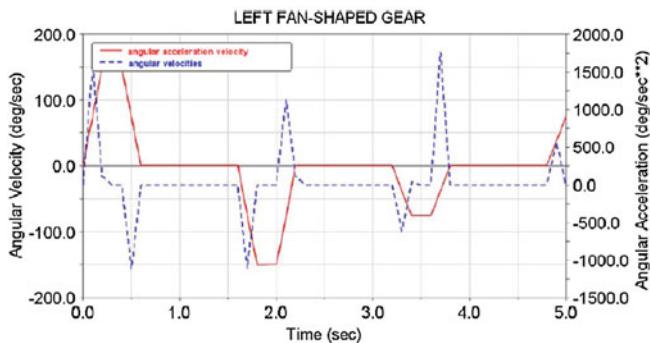
**Fig. 7.32** Computed relationship between  $\theta$  and  $\alpha$  for the cases: **a**  $k = 1$  and **b**  $k = 2$



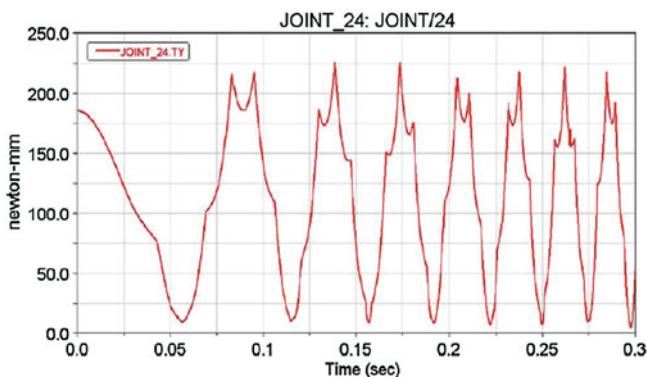
**Fig. 7.33** A model of the designed FRM in ADAMS environment



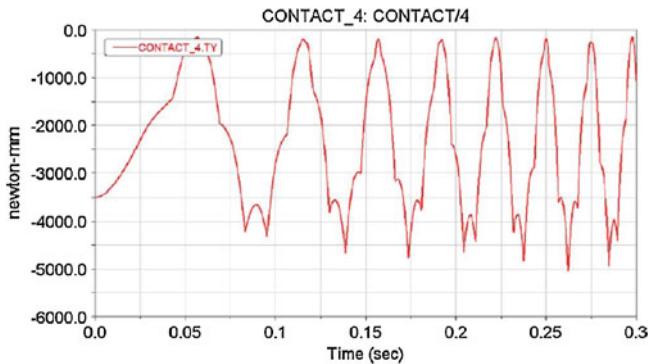
**Fig. 7.34** Simulated finger configurations as achieved by the designed FRM: **a** three-finger centripetal-grasp, **b** three-finger parallel-grasp and **c** two-finger parallel-grasp



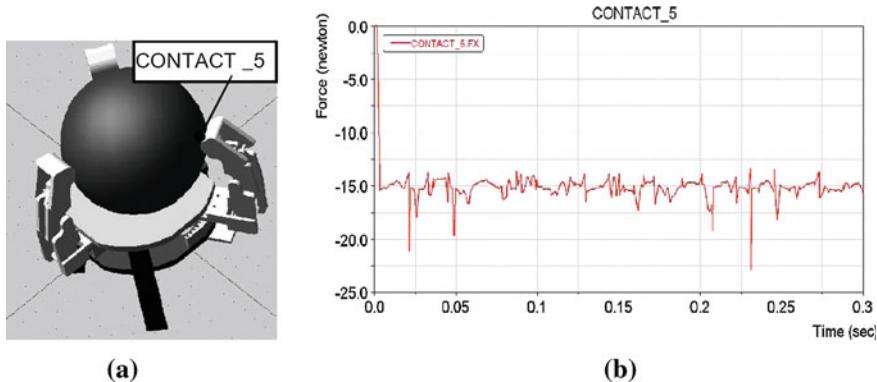
**Fig. 7.35** Simulation plots of angular velocity and angular acceleration velocity



**Fig. 7.36** Driving torque at the joint 24 in Fig. 7.33



**Fig. 7.37** Computed force at CONTACT\_4 in Fig. 7.33 between the fan-shaped gear and planetary gear



**Fig. 7.38** The contact force between the finger and the object with a three-finger centripetal-grasp mode: **a** the simulated configuration and **b** computed contact force

An important result from Fig. 7.36 is the maximum torque that can be identified as 187 N mm, since it is useful for choosing properly the actuator.

A characterization of the feasibility of the designed FRM in Fig. 7.33 has been computed in terms of contact force at teeth transmitting the force from the fan-shaped gear to the planetary gear with the finger. The computed output of the simulation is given in Fig. 7.37, in which it is possible to appreciate the feasible value and time history of the contact force that can be used to size properly the gear teeth. The three-finger centripetal-grasp mode is applied to grasp a sphere in Adams as shown in Fig. 7.38a, and contact points are set up between each finger phalanx and the sphere. One of the computed contact force is shown in Fig. 7.38b and it can reach dynamic equilibrium in a short time, as to indicate the convenience of changing the posture of the fingers. Because all the parts of the FRM and

the sphere are considered rigid bodies, there are also impulse forces at the contact point, which will not exist in practical applications.

As can be seen in the examples, robotic hands cannot achieve versatile grasps with different grasp modes if the fingers cannot move referring to the palm. The problem has been solved by adding mechanism for moving and rotating fingers to form more grasp modes.

## 7.4 Conclusion

In this section, an introduction to the state of high flexibility and multi-purpose operation of human hands and the limitations of current robotic hands is proposed.

Note that before performing a grasp, the human hand can adjust itself to a appropriate position by orienting fingers according to the feature of the object. Having realized the importance of bringing in orientation mechanisms into the structure of robotic hand. Here, several typical examples all with an orientation mechanism such as Laval Underactuated Hand, IAMT robotic hand, and LARM robotic hand IV have been illustrated through aspects of hand structure, orientation mechanism, grasp modes, experiments, and simulations.

## References

1. Taylor GL, Schwartz RJ (1955) The anatomy and mechanics of the human hand. *Artif Limbs* 2:22–35
2. Cutkosky MR (1989) On grasp choice, grasp model, and the design of hands for manufacturing tasks. *IEEE Trans Robot Autom* 5(3):269–279
3. Shimoga KB (1996) Robot grasp synthesis algorithms: a survey. *Int J Robot Res* 15(3): 230–266
4. Iberall T (1997) Human prehension and dexterous robot hands. *Int J Robot Res* 16(3): 285–299
5. Pons JL, Ceres R, Pfeiffer F (1999) Multi finger dexterous robotics hand and control: a review. *Robotica* 17:661–674
6. Bicchi A (2000) Hands for dexterous manipulation and robust grasping: a difficult road to-wards simplicity. *IEEE Trans Robot Autom* 16(6):652–662
7. Laschi C, Dario P, Carrozza MC et al (2000) Grasping and manipulation in humanoid robotics. In: First IEEE-RAS international conference on humanoid robots, Cambridge
8. Okada T (1986) Computer control of multijointed finger system for precise object handling. *Int Trends Manuf Technol Robot Grippers*, pp 391–417
9. Jacobsen SC, Wood JE, Knutti DF et al (1984) The UTAH/M.I.T. Dexterous hand: work in progress. *Int J Robot Res* 3(4):21–50
10. Salisbury KS, Vassura G (1983) Kinematics and force analysis of articulated mechanical hands. *J Mech Transm Actuation Des* 105:35–41
11. Caffaz A, Cannata G (1998) The design and development of the DIST-hand dextrous gripper. In: Proceedings of the IEEE international conference on robotics and automation, pp 2075–2080

12. Melchiorri C, Vassura G (1992) Mechanical and control features of the UB Hand Version II. In: IEEE-RSJ international conference on robots and systems IROS'92, Raleigh, pp 187–193
13. Butterfass J, Grebenstein M, Liu H, Hirzinger G (2001) DLR-hand II: next generation of dexterous robot hand. In: Proceedings of the IEEE Conference on robotics and automation, Seoul, pp 109–114
14. Gosselin CM, Birglen L (2004) Kinetostatic analysis of underactuated fingers. *IEEE Trans Robot Autom* 20(2):211–221
15. Ceccarelli M, Carbone G, Lanni C et al (2003) Experimental activity for designing a hand with One d.o.f. anthropomorphic fingers of human size. In: 12th international workshop on robotics in Alpe-Adria-Danube region RAAD'03, Cassino, 7–10 May 2003
16. Luo MZ, Carbone G, Ceccarelli M, Zhao XX (2010) Analysis and design for changing finger posture in a robotic hand. *Mech Mach Theory* 45:828–843. doi:[10.1016/j.mechmachtheory.2009.10.014](https://doi.org/10.1016/j.mechmachtheory.2009.10.014)
17. Carbone G, Iannone S, Ceccarelli M (2010) Regulation and control of LARM Hand III. *Robot Comput Integr Manuf* 26(2):202–211
18. Namiki A, Imai Y, Ishikawa M, Kaneko M (2003) Development of a high-speed Multi fingered hand system and its application to catching. *Proc IEEE/RSJ Int Conf Intell Robot Syst* 3(3):2666–2671
19. Cutkosky MR (1989) On grasp choice, grasp model and the design of hands for manufacturing tasks. *IEEE Trans Robot Autom* 5(3):269–279
20. Laliberte T, Gosselin CM (1998) Simulation and design of underactuated mechanical hands. *Mech Mach Theory* 33(1/2):39–57
21. Birglen L, Laliberte T, Gosselin CM (2008) Underactuated robotic hands, Springer tracts in advanced robotics, vol 40. Springer, Berlin, p 244
22. Rubinger B, Gregoris L, Gosselin C.M, Laliberte T (2001) Self-adapting robotic auxiliary hand (SARAH) for SPDM operations on the international space station. In: Proceedings of the sixth international symposium on artificial intelligence, robotics and automation in Space ISAIRAS. A New Space Odyssey, Montreal, 18–21 June 2001
23. Laliberté T, Birglen L, Gosselin MC (2002) Underactuation in robotic grasping hands. *Mach Intell Robot Control* 4(3):1–11
24. Luo MZ, Mei T, Wang XH, Yu Y (2004) Grasp characteristics of an underactuated robot hand. In: Proceedings of the 2004 IEEE international conference on robotics and automation, New Orleans, April 2004
25. Ceccarelli M, Carbone G, Ottaviano E, Rodriguez NEN (2004) An experimental validation of a three-fingered hand with one d.o.f. anthropomorphic fingers. In: International conference on intelligent manipulation and grasping IMG04, Genova, 1–2 July 2004
26. Ceccarelli M, Nava Rodriguez NE, Carbone G (2006) Optimal design of driving mechanism in a one-d.o.f. anthropomorphic finger. *Mech Mach Theory* 41:897–911
27. Nardelli A, Carbone G (2008) An experimental characterization of LARM hand IV. In: CD Proceedings of IFToMMFeIbIM international symposium on mechatronics and multibody systems MUSME08, San Juan, 2008, paper n.27
28. Ferrari C, Canny J (1992) Planning optimal grasps. In: Proceedings of the IEEE international conference on robotics and automation. Nice, France, pp 2290–2295

# **Chapter 8**

## **Parallel Wrists for Enhancing Grasping Performance**

**Massimo Callegari, Luca Carbonari, Giacomo Palmieri  
and Matteo-Claudio Palpacelli**

**Abstract** Good grasping and effective manipulation heavily depend on the performance of robotic wrists such as, e.g., the number of degrees of freedom, the kind of motion that is generated, the dexterity of the operations, the stiffness, and the size of the mechanical structure; such characteristics heavily affect kinematic and dynamic performance of the manipulation and can lead to a successful grasp or to an unexpected failure, if not taken into consideration since the early design steps. This chapter, after an introduction recalling the wrist structure of the industrial manipulators, focuses on parallel kinematics wrists, a rather new kind of mechanical architecture that has not found so far relevant industrial applications but shows very promising features, such as mechanical stiffness, high accuracy, lightweight construction, and so on. After presenting a powerful kinematical tool for the synthesis of parallel kinematics machines (SPM), which is based on Lie algebra, the design of a novel spherical wrist is discussed in details. A prototype machine, actuated by three brushless linear motors, has been built with the aim of obtaining good static and dynamic performance.

---

M. Callegari (✉) · L. Carbonari · M.-C. Palpacelli

Department of Industrial Engineering and Mathematical Sciences, Polytechnic University of Marche, Via Brecce Bianche, 60131 Ancona (AN), Italy  
e-mail: m.callegari@univpm.it

L. Carbonari  
e-mail: l.carbonari@univpm.it

M.-C. Palpacelli  
e-mail: m.palpacelli@univpm.it

G. Palmieri  
eCampus University, Via Isimbardi 10, 22060 Nоварate (CO), Italy  
e-mail: giacomo.palmieri@uniecampus.it

## 8.1 Introduction

Since its origins, robotics has been inspired by the features of human beings, both in terms of motor skills, related to the mechanical structure of human body, and in terms of control architecture and mental processing.

Looking at the human body from outside, a serial anthropomorphic structure can be recognized, typically adopted in industrial manipulators, which gives them specific properties including large workspace and high manipulability. On the contrary, by analyzing human joints in more detail, it is possible to identify parallel kinematic chains which result from the complex pattern created by tendons and bones.

Even the performance which is required to a grasping tool is usually compared to those of the human hand, that is able to grab and manipulate several kinds of objects, different in shape and dimension, and to keep them stable even in the presence of disturbing forces. Industrial devices, however, imitate only partially the hand behavior, preferring economy of scope and simplicity of realization to more complex manipulation capabilities (see for example the common 2-finger grippers which are only able to switch from the open to the close configuration). On the other hand, industrial manipulators are usually equipped by a 2- or 3-axes wrist that allow them to set the gripper in the best grasping conditions, overcoming its motion limitations.

The human hand is able to rotate through the three rotational degrees of freedom (DOFs) provided by the wrist and the forearm: the former is responsible for the hand radial/ulnar deviation and flexion/extension, while the latter for the hand pronation/supination rotation; several research studies on wrists and grippers have been proposed in the literature during past years [1–4], while a sample design of a humanoid robot wrist is provided by Schäfer and Dillmann [5].

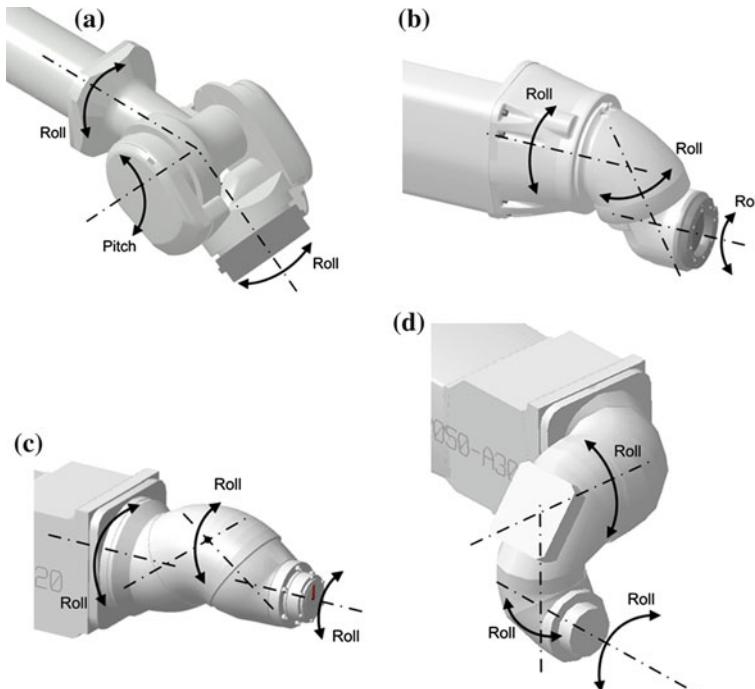
The dexterity of a 6-axis serial manipulator is mainly due to its wrist, which has to concentrate in a compact space three axes of rotation; the robotic arm, instead, is mainly responsible for the positioning of the wrist. For a 6-DOF fully parallel manipulator, such as the Gough-Stewart Platform [6], rotation and positioning are obtained by a proper coordination of robot actuators, since the moving platform is driven by six actuated legs which concurrently manage its 6-DOF: for this kind of spatial parallel manipulators, end-effector rotations and translations in space are highly coupled by the mechanical structure. Sometimes it is preferable to decouple the motion referring to spherical parallel wrists: in this case, a 3-axes platform rotates around a fixed point, which represents the wrist center of rotation [7].

## 8.2 Wrist Kinematics

The most successful architectures of industrial robot wrists belong to the families of Pitch-Yaw-Roll (PYR), Roll-Pitch-Roll (RPR) and 3-Roll wrists, according to the orientation of joint axes.

The kinematics of PYR wrists closely resembles the human hand, with the pitch and yaw axes mimicking the flexion/extension and the radial/ulnar deviation and the roll axis performing the rotation of the forearm. The difficulty in laying the three axes, so that they intersect at a common point prevents the attainment of a spherical motion; in this way, singular configurations are avoided but the design is more complex and highly coupled; as a matter of fact, nowadays, PYR wrists are not used anymore.

Figure 8.1 shows the most popular industrial wrists, based on the RPR and 3-Roll serial kinematics. The joint axes of Roll-Pitch-Roll wrists always meet at a common point, thus realizing a spherical motion: the possible occurrence of singular configurations at the center of their mobility range represents their major drawback, but their simple and modular architecture paved the way to the success of such design.



**Fig. 8.1** Typical industrial wrist architectures: Roll-Pitch-Roll (a) 3-Roll (b) modified 3-Roll (c) and lemma (d)

The 3-Roll solution, see Fig. 8.1b, is characterized by three rotations around consecutive roll axes: such architecture does not realize a spherical motion since the three axes do not intersect at a common point, but the possibility to obtain large rotations of the joints enhances the dexterity of the wrist. In Fig. 8.1c, two incident axes of rotation, which follow the first roll axis, provide only one DOF, because they are connected by a bevel gear transmission. Figure 8.1d shows the Lemma wrist, which is similar to the PYR architecture and is still used in some painting or coating applications.

The weight and the dynamic loads of the hand, including the grasped object, must be supported by the wrist in the most severe conditions; to this aim, stiffness and accuracy are required for the wrist in order to reduce possibilities of task failure and to achieve a more stable grasping operation.

A good design of robot grasping devices must aim at integrating actuators and transmissions within its mechanical structure, maintaining a modularity where each device is autonomous and responsible for carrying out its own functions: the robot hand for the grasping, the wrist for the orientation of the hand, and the robotic arm for the positioning of the wrist.

The robot wrist is therefore concerned with the hand rotation, undergoing the bending moment due to the load applied to the gripper: in order to provide the robot with the ability to manipulate different kinds of objects, in terms of weight and dimension, a parallel kinematics may be conveniently considered as an interesting alternative to a serial architecture for the wrist design, waiving part of the robot mobility in favor of compactness and a high power-to-mass ratio with a greater load capacity and stiffness.

### 8.3 Parallel Kinematics Wrists

Parallel Kinematics Machines (PKM's) offer several features that make them attractive for grasping operations: high mechanical stiffness, high accuracy, lightweight construction, and compact size (with reference to payload capacity); on the contrary, they generally have a limited workspace and a complex direct kinematics. Therefore, they can be profitably used in applications where the mentioned characteristics are required, for instance in biomedicine and surgery [8]. A great number of PKM's have been proposed as haptic devices in the field of human wrist or ankle rehabilitation, see, e.g., Takaiwa and Noritsugu [9]; Arata et al. [10] proposed a redundantly actuated 3-DOF parallel machine to be used as haptic device, while Unal and Patoglu [11] designed two exoskeletons based on parallel kinematics and Saltaren et al. [12] investigated the performance of 3-DOF spherical parallel manipulators in humanoid robotics.

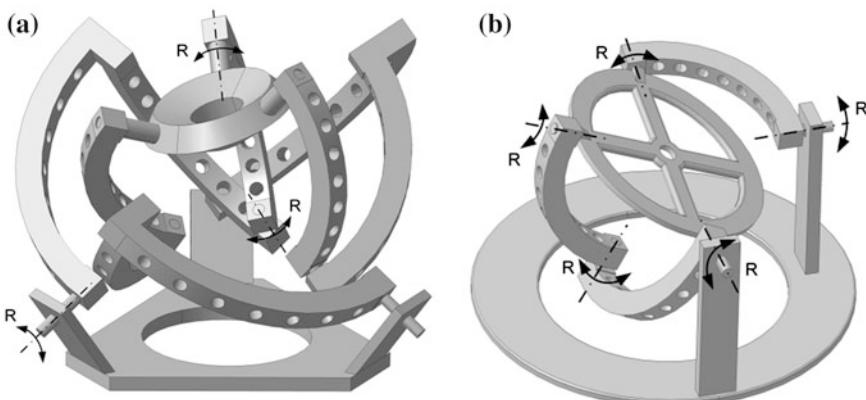
Spherical parallel machines (SPM's) are the class of parallel machines which are able to perform motions of pure rotations: such research field was first introduced by the work of Asada and Granito [13] but the most relevant results have been obtained by Gosselin and Angeles [14], that conceived the agile eye, the most

known mechanism of this family upon which many prototype machines have been designed since then. The agile eye is based on the 3-RRR overconstrained kinematics, with three identical legs composed by two links and characterized by a sequence of three revolute joints, as shown in Fig. 8.2a. Few other studies on the subject are available during the 1990s [15–17], while a renewed interest characterized the beginning of the new century and produced interesting results.

The use of synthesis methods based on screw theory, for instance, has been exploited by Kong and Gosselin [18, 19] that provide comprehensive listings of both overconstrained and nonoverconstrained SPM's; Hervé and Karouia, on the other hand, use the theory of Lie group of displacements to devise novel architectures, as the four main families in [20] or the 3-UPU, 3-RCC, 3-CCR, 3-CRC kinematics specifically treated in Karouia and Hervé [20–22]; Fang and Tsai [23] use the theory of reciprocal screws to present a systematic methodology for the structural synthesis of a class of 3-DOF rotational parallel manipulators (e.g., 3-RRS, 3-CRU, 3-CRC, 3-UPC). Further interesting, nonoverconstrained architectures are the 3-URC, 3-RUU, 3-RRS, and 3-RSR, studied by Di Gregorio [24–27], and the 3-URU by Huda and Takeda [28].

Parallel kinematics becomes a forced option for small size devices, where the scale factor requires the use of passive flexible joints and conjugate pairs are only exploited for the actuated ground joints. Specific results have been reached in this field by Yi et al. [29], Lusk and Howell [30, 31] Callegari et al. [32] and Palpacelli et al. [33].

Sometimes the wrist only needs 2 DOFs to perform the grasping, in which case different design solutions have been identified: Fontana et al. [34] presented a novel robotic wrist with high angulation capability, Carricato and Parenti-Castelli [35] proposed a fully decoupled 2-DOF parallel wrist. An example of a 2-DOF spherical device is shown in Fig. 8.2b, where a simplified version of the agile eye is sketched; in this case, the moving platform is not able to spin around its own axis.



**Fig. 8.2** Kinematics of the agile eye: original 3-RRR kinematics (a) and simplified 2-dof version (b)

## 8.4 Kinematic Synthesis of Spherical Parallel Machines

A useful aid for the kinematic synthesis of SPM's is provided by the contribution of Karouia and Hervé [21], whose work has been then reformulated by Kong and Gosselin [18, 19]. By following such approach, Callegari et al. [36] studied a new kinematics based on the 3-CPU structure, upon which a spherical wrist has been designed and eventually prototyped: the main synthesis steps of such machine are outlined in the following paragraphs.

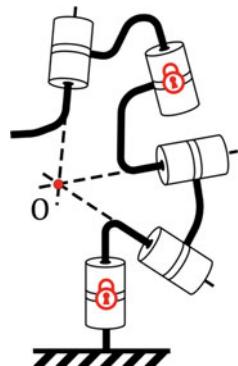
It is anticipated that the synthesis process has been driven by some heuristic rules. First of all, it was decided to focalize the study only on nonoverconstrained mechanisms: this choice avoids practical problems such as the strict dimensional and geometric tolerances needed by overconstrained machines during manufacturing and assembly phases. Moreover, the use of passive spherical pairs directly joining the platform to the base has been avoided and for economic reasons only modular solutions composed of three identical legs have been studied.

A simple mobility analysis shows that, in order to design a moving platform connected to the ground by identical legs and characterized by 3-DOFs, it must be composed by three limbs of connectivity five. In the general case, each leg is composed by four links connected to each other (and to mobile platform and fixed ground at the distal ends) by five lower joints (revolute R or prismatic P). To allow the end effector have 3-DOFs of pure rotation, 3 out of the 5 pairs in each limb must be revolute joints with concurrent axes: if such points of intersection are the same for all the three legs, the platform will only rotate around this common point O. In this way, each limb generates a 5-D manifold that must contain the 3-D group of spherical motions around the point O. The well-known wrist agile eye by Gosselin and Angeles is realized exactly in this way; nevertheless, in this case two revolute pairs in each limb are locked and the mechanism results overconstrained, as shown in Fig. 8.3.

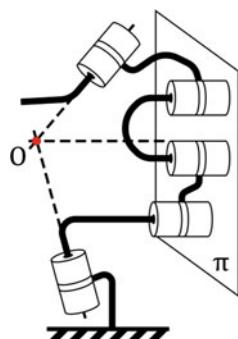
The spherical motion can be obtained also by using five revolute pairs  $R_1-R_5$  where the axes of the joints  $R_1$ ,  $R_3$ , and  $R_5$  still intersect at a common point while the axes of pairs  $R_2$  and  $R_4$  are parallel to the direction of  $R_3$ . In such a way, the three joints  $R_2$ ,  $R_3$ , and  $R_4$  will generate the 3-D subgroup of planar displacements  $G(\Pi)$ , i.e., the set of translations lying in  $\Pi$  and rotations around axes perpendicular to  $\Pi$ . The same subgroup  $G(\Pi)$  is generated also in case the axis of revolute joint  $R_3$  is still perpendicular to plane  $\Pi$  but does not cross the rotation center  $O$ , as shown in Fig. 8.4; therefore, also with this limb kinematics a spherical wrist can be obtained.

On the other hand, by following the same line of reasoning, the same subgroup of planar displacements  $G(\Pi)$  can be generated by substituting one or two revolute joints among the  $R_2$ ,  $R_3$ , and  $R_4$  set with prismatic pairs whose axes lie in the plane  $\Pi$ , thus obtaining limbs whose central joints are characterized by one of the sequences PRR, RPR, PPR, PRP, RRP, and RPP. Of course, two adjacent joints in limbs kinematics can be merged to yield simpler architectures with fewer links; for instance, two revolute joints with orthogonal axes can be superimposed to give a universal (U) joint, while the set of one revolute joint and one prismatic pair with the same axes are equivalent to a cylindrical (C) joint, as shown in Fig. 8.5.

**Fig. 8.3** Limb of connectivity five able to generate a spherical motion of the platform

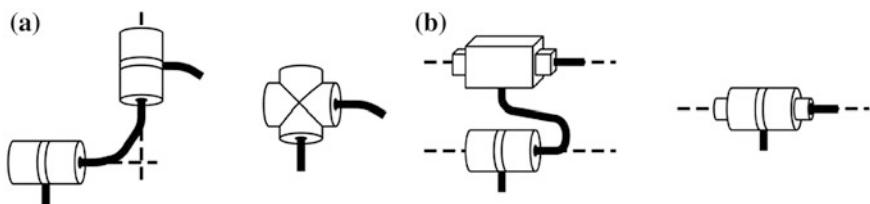


**Fig. 8.4** Limb with subgroup RRR generating the subgroup of planar displacements

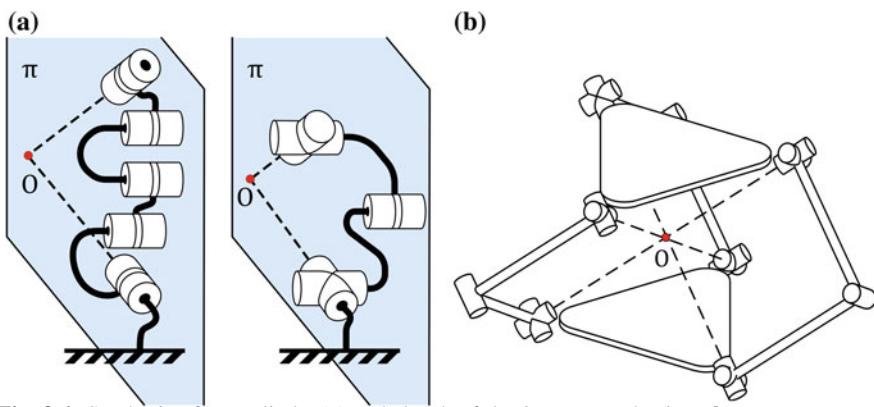


The kinematic chains described above prevent the  $i$ th limb's end from translating in the direction normal to the plane  $\Pi_i$ ,  $i = 1, 2, 3$ ; therefore, if three such chains are used for the limbs and the three normals to the planes  $\Pi_i$  are linearly independent, all the possible translations in space are locked and the mobile platform, attached to the three limbs, can only rotate around a fixed point.

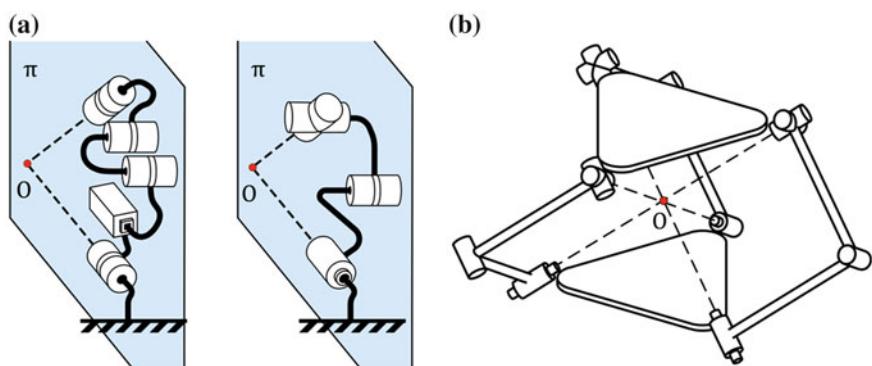
In this way, seven alternative design concepts have been considered, which are: 3-URU, 3-CRU, 3-URC, 3-UPU, 3-CPU, 3-UPC, 3-CRC. Figures 8.6, 8.7, 8.8, 8.9, 8.10 show the synthesis steps leading to the specific limb topology (a) and sketch a possible arrangement of the introduced joints (b). If the three limbs of each machine are arranged in the simplest possible setting (at least from a merely visual point of view), they will all lie within vertical planes: unfortunately in this case the three normals to limbs' planes are all parallel to the horizontal plane, and therefore result linearly dependent, allowing the platforms to translate along the vertical direction. Of course, there are many different settings of these normal axes in space that grant them to be linearly independent: Figs. 8.6b–8.10b show that limbs' planes have been tilted in such a way that they become mutually orthogonal in the initial configuration (or “home” position) of the wrist. In this manner, the kinematic relations of the related machines are greatly simplified and above all, even if this arrangement changes during operation of the machine, this configuration is the most far from the



**Fig. 8.5** Merge of two adjacent joints able to yield universal **a** or cylindrical **b** pairs



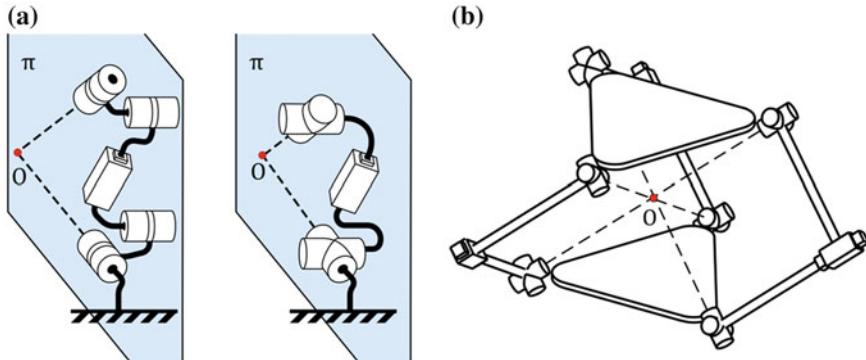
**Fig. 8.6** Synthesis of URU limbs **(a)** and sketch of the 3-URU mechanism **(b)**



**Fig. 8.7** Synthesis of CRU and URC limbs **(a)** and sketch of the 3-CRU mechanism **(b)**

singular setting previously outlined, therefore granting a better kinematic manipulability of the wrists.

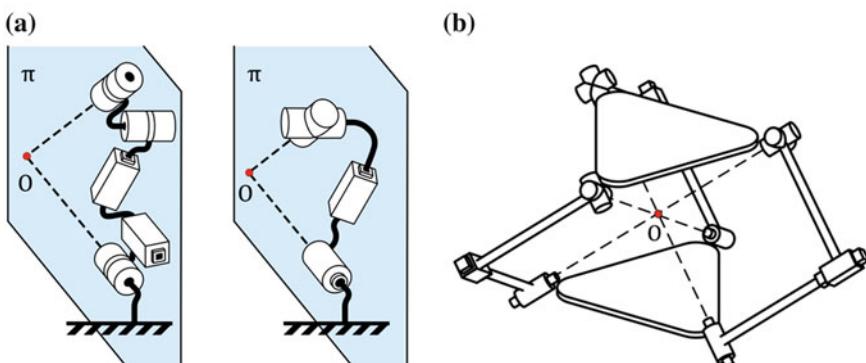
The kinematics of all these concepts has been investigated and in view of the design of a physical prototype, the 3-CPU architecture of Fig. 8.9 has been considered for further development; in making this choice it has been considered that:



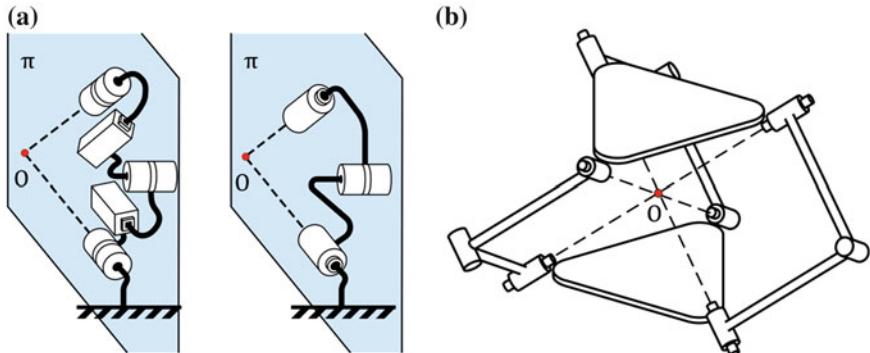
**Fig. 8.8** Synthesis of UPU limb (a) and sketch of the 3-UPU mechanism (b)

- if cylindrical pairs must be arranged on the mobile platform, its design will be cumbersome due to the stroke of the linear motion of the joints, unless a very limited workspace is accepted;
- if the intermediate pair is revolute, machine's kinematics is usually more complex than the case it is prismatic; see for instance, the kinematics of the 3-CRU [32] in comparison with the present 3-CPU case;
- the 3-UPU mechanism has been studied already by other researchers and, in case it is actuated by motors placed on the ground, it suffers from singularities in its home configuration, as shown for instance in [37, 38].

As a matter of fact, next sections will show many benefits of the 3-CPU architecture, such as the simplicity of the kinematic relations, the compactness of the mechanical design, the easiness of actuation, and finally the novelty of the concept. Before entering the study of the 3-CPU SPM, it is marginally noted that the same limbs' topology, with different joints arrangements, is able to provide several kinds of motions [39], included the pure translation [40]; moreover, the 3-



**Fig. 8.9** Synthesis of CPU and UPC limbs (a) and sketch of the 3-CPU mechanism (b)

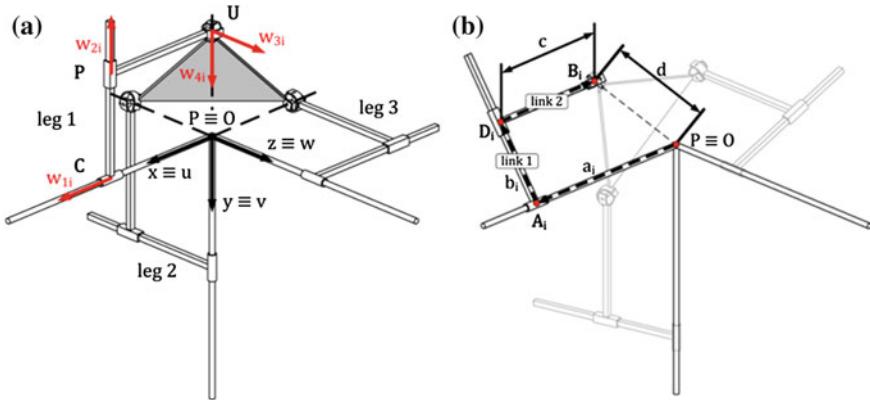


**Fig. 8.10** Synthesis of CRC limbs (a) and sketch of the 3-CRC mechanism (b)

CRU mechanism of Fig. 8.7, even if characterized by a far more complex kinematics, is extensively studied in [32] in view of the realization of a SPM for miniaturized assembly tasks.

## 8.5 Geometry and Reference Frames

Making reference to Fig. 8.11, the axes of cylindrical joints  $A_i$ ,  $i = 1, 2, 3$  intersect at point  $O$  (center of the motion) and are aligned to the axes  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{z}$  respectively of a (*fixed*) Cartesian frame centered in  $O$ . The first member of each link (1) is perpendicular to  $A_i$  and has a variable length  $b_i$  due to the presence of the prismatic joint  $D_i$ ; the second link (2) of the leg is set parallel the base cylindrical pair. The universal joint  $B_i$  is composed by two revolute pairs with orthogonal axes: one is perpendicular to leg's plane while the other intersects at a common point  $P$  with the corresponding joints of the other limbs; such directions, for the legs  $i = 1, 2, 3$  orderly, are aligned to the axes  $\mathbf{u}$ ,  $\mathbf{v}$ ,  $\mathbf{w}$  respectively of a (*mobile*) Cartesian frame, located in  $P$  and attached to the rotating platform. For a successful operation of the mechanism, such *manufacturing conditions* must be accompanied by a proper *mounting condition*: assembly should be performed in such a way that the two frames  $O(\mathbf{x}, \mathbf{y}, \mathbf{z})$  and  $P(\mathbf{u}, \mathbf{v}, \mathbf{w})$  come to coincide. If these conditions are verified, the points  $P$  and  $O$  remain coincident during the motion and the moving platform performs a spherical motion. In the initial configuration, the linear displacements  $a_i$  of the cylindrical joints are equal to the length  $c$  of the second link and the displacements  $b_i$  of the prismatic joints are equal to the constant length  $d$  for all the limbs. It is also evident that, for practical design considerations, SPM's based on the 3-CPU concept are efficiently actuated by driving the linear displacements of the cylindrical pairs coupling the limbs with the frame: therefore, in the following kinematic analysis it will be made reference to this case (i.e., joint variables  $a_i$ ,  $i = 1, 2, 3$  will be considered the actuation parameters).



**Fig. 8.11** Reference frames in home configuration (a) and geometry of a single limb (b)

Two auxiliary frames,  $O^*(x^*, y^*, z^*)$  and  $P^*(u^*, v^*, w^*)$ , are introduced, as shown in Fig. 8.12a, in order to better define the spatial configuration of the platform and to be able to easily visualize and understand the assigned tasks and the obtained results. The frame  $P^*(u^*, v^*, w^*)$  is attached to the mobile platform and is defined as follows:

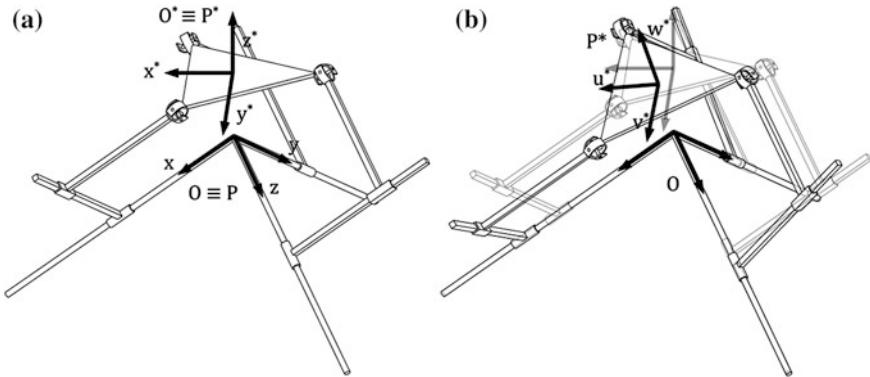
- the origin  $P^*$  is located at the center of the equilateral triangle formed by the points  $B_1$ ,  $B_2$ , and  $B_3$ ;
- the  $w^*$  axis is normal to the platform (outward direction);
- the  $u^*$  axis has the direction ( $P^*-B_1$ );
- the  $v^*$  axis is placed according to the right-hand rule.

The fixed frame  $O^*(x^*, y^*, z^*)$  is coincident with the mobile frame  $P^*(u^*, v^*, w^*)$  when the platform is in its initial configuration. Of course, since the frames are not placed at the center of the spherical motion, the two origins  $O^*$  and  $P^*$  will be coincident only in the home configuration, see Fig. 8.12b.

The orientation of the platform can be easily described by assigning the relative rotation between the mobile frame attached to the platform and the fixed frame solid with the ground. In the development of robot's kinematics, the two frames  $O(x, y, z)$  and  $P(u, v, w)$  will be used for the purpose and the rotation matrix  ${}_P R$  will be expressed by means of the following  $\{\alpha, \beta, \gamma\}$  Cardan angles set:

$${}_P \mathbf{R}(\alpha, \beta, \gamma) = \mathbf{R}_x(\alpha) \mathbf{R}_y(\beta) \mathbf{R}_z(\gamma) = \begin{bmatrix} c\beta c\gamma & -c\beta s\gamma & s\beta \\ s\alpha s\beta c\gamma + c\alpha s\gamma & -s\alpha s\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta \\ -c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha s\beta s\gamma + s\alpha c\gamma & c\alpha c\beta \end{bmatrix} \quad (8.1)$$

Path planning and wrist's control, on the other hand, shall be developed by assigning the orientation of  $P^*(u^*, v^*, w^*)$  frame with respect to  $O^*(x^*, y^*, z^*)$



**Fig. 8.12** Placement of auxiliary frames  $O^*(x^*, y^*, z^*)$  and  $P^*(u^*, v^*, w^*)$ : home pose (a) and perturbed setting (b)

frame, and the corresponding set  $\{\varphi_x, \varphi_y, \varphi_z\}$  of Cardan angles is defined in the same manner by:

$${}_{P^*}^{O^*} \mathbf{R}(\varphi_x, \varphi_y, \varphi_z) = \mathbf{R}_{x^*}(\varphi_x) \mathbf{R}_{y^*}(\varphi_y) \mathbf{R}_{z^*}(\varphi_z) \quad (8.2)$$

Since the orientation of the frame  $O^*$  with respect to the frame  $O$  is provided by the constant rotation matrix  ${}_O^R$ , the orientation of the mobile platform, if known in the  $O(x, y, z)$  space, can be easily mapped in the  $O^*(x^*, y^*, z^*)$  space by:

$${}_{P^*}^{O^*} \mathbf{R} = {}_{O^*}^R {}_P^T O {}_{O^*}^R \mathbf{R} \quad (8.3)$$

where the obvious identity  ${}_O^R = {}_P^R$  has been used.

## 8.6 Analysis of Mobility

It is now shown that, if the mentioned assembly and mounting conditions are satisfied, the mobile platform is characterized by motions of pure rotation, as anticipated in previous section; with reference to Fig. 8.11, such conditions may be geometrically expressed by:

1.  $\hat{\mathbf{w}}_{1i}$  and  $\hat{\mathbf{w}}_{4i}$  incident in  $P$ ;
2.  $\hat{\mathbf{w}}_{3i}$  perpendicular to the plane  $\langle \hat{\mathbf{w}}_{1i}, \hat{\mathbf{w}}_{4i} \rangle$  (i.e.  $\hat{\mathbf{w}}_{3i} \cdot \hat{\mathbf{w}}_{1i} = 0$  and  $\hat{\mathbf{w}}_{3i} \cdot \hat{\mathbf{w}}_{4i} = 0$ );
3.  $\hat{\mathbf{w}}_{2i}$  lying in plane  $\langle \hat{\mathbf{w}}_{1i}, \hat{\mathbf{w}}_{4i} \rangle$  (by taking into account (2), it leads to:  $\hat{\mathbf{w}}_{3i} = \hat{\mathbf{w}}_{1i} \times \hat{\mathbf{w}}_{2i}$ );
4.  $\hat{\mathbf{w}}_{2i}$  not parallel to  $\hat{\mathbf{w}}_{1i}$  therefore  $\hat{\mathbf{w}}_{1i} \times \hat{\mathbf{w}}_{2i} \neq \hat{\mathbf{0}}$  (for simplicity, the condition  $\hat{\mathbf{w}}_{1i} \cdot \hat{\mathbf{w}}_{2i} = 0$  has been taken).

If the point  $P$  is considered belonging to the  $i^{\text{th}}$  leg ( $i = 1, 2, 3$ ), its velocity can be written as follows:

$$\dot{\mathbf{P}}_i = \dot{\mathbf{P}}_{2i} + \dot{\mathbf{P}}_{ri} \quad (8.4)$$

where  $\dot{\mathbf{P}}_{2i}$  is the velocity of point  $P$  if considered attached to the 2nd link:

$$\dot{\mathbf{P}}_{2i} = \dot{\mathbf{B}} + \boldsymbol{\omega}_{2i} \times (\mathbf{P} - \mathbf{B}_i) = \dot{\mathbf{B}}_i + \boldsymbol{\omega}_{2i} \times d \hat{\mathbf{w}}_{4i} \quad (8.5)$$

and  $\dot{\mathbf{P}}_{ri}$  is the velocity of point  $P$  relative to a frame attached to link 2 and centered in point  $B_i$ :

$$\dot{\mathbf{P}}_{ri} = \dot{\theta}_{3i} \hat{\mathbf{w}}_{3i} \times (\mathbf{P} - \mathbf{B}_i) = \dot{\theta}_{3i} \hat{\mathbf{w}}_{3i} \times d \hat{\mathbf{w}}_{4i} \quad (8.6)$$

In Eq. (8.5),  $\boldsymbol{\omega}_{2i}$  is the angular velocity of link 2:

$$\boldsymbol{\omega}_{2i} = \dot{\theta}_{1i} \hat{\mathbf{w}}_{1i} \quad (8.7)$$

In the same way, with obvious meaning of the symbols, the vector describing the velocity of point  $B$  can be expressed as:

$$\dot{\mathbf{B}}_i = \dot{\mathbf{B}}_{1i} + \dot{\mathbf{B}}_{ri} \quad (8.8)$$

where:

$$\begin{aligned} \dot{\mathbf{B}}_{1i} &= \dot{a}_i \hat{\mathbf{w}}_{1i} + \boldsymbol{\omega}_{1i} \times (\mathbf{B}_i - \mathbf{A}_i) \\ &= \dot{a}_i \hat{\mathbf{w}}_{1i} + \dot{\theta}_{1i} \hat{\mathbf{w}}_{1i} \times (a_i \hat{\mathbf{w}}_{1i} - d \hat{\mathbf{w}}_{4i}) \\ &= \dot{a}_i \hat{\mathbf{w}}_{1i} - \dot{\theta}_{1i} \hat{\mathbf{w}}_{1i} \times d \hat{\mathbf{w}}_{4i} \end{aligned} \quad (8.9)$$

$$\dot{\mathbf{B}}_{ri} = \dot{b}_i \hat{\mathbf{w}}_{2i} \quad (8.10)$$

Equations (8.5)–(8.10) are substituted back in Eq. (8.4), it is found:

$$\dot{\mathbf{P}}_i = \dot{b}_i \hat{\mathbf{w}}_{2i} + \dot{a}_i \hat{\mathbf{w}}_{1i} + \dot{\theta}_{3i} \hat{\mathbf{w}}_{3i} \times d \hat{\mathbf{w}}_{4i} \quad (8.11)$$

By dot-multiplying Eq. (8.11) by  $\hat{\mathbf{w}}_{3i}$  and by taking into account the conditions (1–4), it is finally obtained:

$$\hat{\mathbf{w}}_{3i} \cdot \dot{\mathbf{P}}_i = 0 \quad (8.12)$$

that can be differentiated to yield:

$$\hat{\mathbf{w}}_{3i} \cdot \ddot{\mathbf{P}}_i + \dot{\hat{\mathbf{w}}}_{3i} \cdot \dot{\mathbf{P}}_i = 0 \quad (8.13)$$

Equations (8.12) and (8.13), written for the three legs, build up a system of six linear algebraic equations in six unknowns, the scalar components of  $\dot{\mathbf{P}}$  and  $\ddot{\mathbf{P}}$ . Such a system can be written in matrix form as follows:

$$\mathbf{L} \begin{bmatrix} \dot{\mathbf{P}} \\ \ddot{\mathbf{P}} \end{bmatrix} = \mathbf{0} \quad (8.14)$$

The  $6 \times 6$  matrix  $\mathbf{L}$  collects the coefficients of the linear system and can be partitioned as:

$$\mathbf{L} = \begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \dot{\mathbf{H}} & \mathbf{H} \end{bmatrix} \quad (8.15)$$

with:

$$\mathbf{H} = \begin{bmatrix} \widehat{\mathbf{W}}_{31}^T \\ \widehat{\mathbf{W}}_{32}^T \\ \widehat{\mathbf{W}}_{33}^T \end{bmatrix} \quad (8.16)$$

and  $\mathbf{0}$  being the null  $3 \times 3$  matrix.

If the matrix  $\mathbf{L}$  is not singular, the system in Eq. (8.14) only admits the trivial null solution:

$$\dot{\mathbf{P}} = \ddot{\mathbf{P}} = \mathbf{0} \quad (8.17)$$

which means that the point  $P$  does not move in space, i.e., the moving platform only rotates around  $P$ . On the other hand, it is possible to identify the singular configurations of the wrist by posing:

$$\det(\mathbf{L}) = [\det(\mathbf{H})]^2 = 0 \quad (8.18)$$

that leads to:

$$\det(\mathbf{H}) = \widehat{\mathbf{w}}_{31} \cdot \widehat{\mathbf{w}}_{32} \times \widehat{\mathbf{w}}_{33} = 0 \quad (8.19)$$

Equation (8.19) is satisfied only when the three unit vectors  $\widehat{\mathbf{w}}_{31}$ ,  $\widehat{\mathbf{w}}_{32}$  and  $\widehat{\mathbf{w}}_{33}$  are linearly dependent; therefore, the platform will incur in a *singularity configuration* if and only if the three vectors are linearly dependent. It might happen when:

- the planes containing the three legs are simultaneously perpendicular to the base plane;
- such planes are coincident with the base plane (configuration not reachable);
- at least two out of the three aforementioned planes are characterized by normal unit vectors which are parallel.

This justifies the previous choice of having the legs laid on mutual orthogonal planes, since such configuration results the most far from translation singularities.

## 8.7 Orientation Kinematics

Orientation kinematics is based on the definition of the relative rotation between mobile frame  $P(\mathbf{u}, \mathbf{v}, \mathbf{w})$  and fixed frame  $O(\mathbf{x}, \mathbf{y}, \mathbf{z})$ , with  $P \equiv O$ ; it can be solved by writing one loop-closure equation for each leg as follows:

$$(A_i - P) + (D_i - A_i) + (B_i - D_i) + (P - B_i) = \mathbf{0} \quad (8.20)$$

Equation (8.20) is conveniently expressed in a local frame  $O_i(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$ ,  $i = 1, 2, 3$  which is defined for each leg as follows: the  $\mathbf{x}_i$  axis is aligned with cylindrical joint's axis and the  $\mathbf{y}_i$  axis is chosen parallel to limb's first link, when it is laid in the initial configuration. It is obtained:

$$\begin{bmatrix} a_i \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -b_i c\theta_{1i} \\ -b_i s\theta_{1i} \end{bmatrix} + \begin{bmatrix} -c \\ 0 \\ 0 \end{bmatrix} + {}_O^i \mathbf{R}_P^O \mathbf{R}^P (P - B_i) = \mathbf{0} \quad (8.21)$$

In *inverse kinematics*, the values of  $\alpha$ ,  $\beta$ , and  $\gamma$  Cardan angles (or equivalently the elements  $r_{ij}$  of the rotation matrix  ${}_P^O \mathbf{R}$ ) are known and the joint variables  $a_i$  must be found; in this case, loop-closure Eq. (8.21) for  $i = 1, 2, 3$  represents three decoupled systems of nonlinear algebraic equations in the unknowns  $a_i$ ,  $\theta_{1i}$  and  $b_i$ , that can be solved to find the solutions:

$$\begin{cases} a_1 = c - d r_{12} \\ \theta_{11} = \text{atan2}(r_{32}, r_{22}) \\ b_1 = \frac{d r_{22}}{c \theta_{11}} \text{ or } b_1 = \frac{d r_{32}}{s \theta_{11}} \end{cases} \quad \begin{cases} a_2 = c - d r_{23} \\ \theta_{12} = \text{atan2}(r_{13}, r_{33}) \\ b_2 = \frac{d r_{33}}{c \theta_{12}} \text{ or } b_2 = \frac{d r_{13}}{s \theta_{12}} \end{cases} \quad \begin{cases} a_3 = c - d r_{31} \\ \theta_{13} = \text{atan2}(r_{21}, r_{11}) \\ b_3 = \frac{d r_{11}}{c \theta_{13}} \text{ or } b_3 = \frac{d r_{21}}{s \theta_{13}} \end{cases} \quad (8.22)$$

Equation (8.22) accounts for eight different solutions of the inverse kinematics problem, i.e., they are as many as the number of possible configurations of the three legs (two possible configurations per leg). It is noted, however, that only one value is found for the displacement of each actuated joint when the orientation of the platform is given.

The *direct kinematic* problem, on the other hand, assumes the knowledge of joint variables  $a_i$ ,  $i = 1, 2, 3$  and aims at finding all the admissible attitudes of the platform in the space. The analysis is performed by means of simple trigonometric manipulations; by substituting in Eq. (8.22) the expression of  $r_{ij}$  given in Eq. (8.1), it is obtained:

$$\begin{cases} r_{12} = -c\beta s\gamma = \frac{c-a_1}{d} \\ r_{23} = -s\alpha c\beta = \frac{c-a_2}{d} \\ r_{31} = -c\alpha s\beta c\gamma + s\alpha s\gamma = \frac{c-a_3}{d} \end{cases} \quad (8.23)$$

Direct kinematics is solved when the full rotation matrix  ${}_P^O \mathbf{R}$  has been computed, starting from the knowledge of the three components  $r_{12}$ ,  $r_{23}$ ,  $r_{31}$  given by Eq. (8.23). By taking into consideration that the columns of  ${}_P^O \mathbf{R}$  represent the unit

vectors of a Cartesian reference frame, the three relations of unitary modulus and the three relations of mutual orthogonality could be exploited in order to find the six unknown components, but the six arising nonlinear equations would be pretty difficult to solve in closed form. In the present case, it is simpler to parameterize the rotation matrix by means of the Cardan angles set given by Eq. (8.1) and to find the values of  $\alpha$ ,  $\beta$ , and  $\gamma$  by solving the nonlinear system in Eq. (8.23). It is noted that the parameterization (but not wrist kinematics) degenerates when  $\beta = \pm\pi/2$ .

By considering separately the singular cases, from the first two equations in Eq. (8.23) it is obtained:

$$s\alpha = \frac{r_{23}}{r_{12}} s\gamma \quad (8.24)$$

and by considering the first and the last equations in Eq. (8.23) it results:

$$\frac{r_{12}^2}{s^2\gamma} + \frac{r_{31}^2 + s^2\alpha s^2\gamma - 2r_{31}s\alpha s\gamma}{c^2\alpha c^2\gamma} = 1 \quad (8.25)$$

Equation (8.25) can be elaborated so as to obtain:

$$\begin{aligned} r_{12}^2(1 - s^2\gamma - s^2\alpha + s^2\alpha s^2\gamma) + s^2\gamma(r_{31}^2 + s^2\alpha s^2\gamma - 2r_{31}s\alpha s\gamma) \\ = s^2\gamma(1 - s^2\gamma - s^2\alpha + s^2\alpha s^2\gamma) \end{aligned} \quad (8.26)$$

and taking into consideration Eq. (8.24):

$$\lambda s^4\gamma + \mu s^2\gamma + v = 0 \quad (8.27)$$

with:

$$\begin{aligned} \lambda &= -2\frac{r_{31}r_{23}}{r_{12}} + \frac{r_{23}^2}{r_{12}^2} + r_{23}^2 + 1 > 0 \\ \mu &= r_{31}^2 - r_{23}^2 - r_{12}^2 - 1 \leq 0 \\ v &= r_{12}^2 \geq 0 \end{aligned} \quad (8.28)$$

It can be observed that, if the discriminant of Eq. (8.27) is positive, the signs of  $\lambda$ ,  $\mu$ , and  $v$  guarantee that the two solutions of Eq. (8.27) are positive real, and therefore four real values for  $\sin\gamma$  and eight values for the angle  $\gamma$  can be found. The values of the three parameters  $\lambda$ ,  $\mu$  and  $v$ , which cause negative values of the discriminant, correspond to reflection matrices (i.e., the determinant is equal to  $-1$ ): since this kind of matrices cannot be found using the parameterization of the Cardan angles, in this case the solutions of Eq. (8.27) become complex. Once  $\gamma$  is known, the value of  $\cos\beta$  can be obtained from the first equation in Eq. (8.23):

$$c\beta = -\frac{r_{12}}{s\gamma} \quad (8.29)$$

It is noted that, for the chosen parameterization, angle  $\beta$  varies in the range  $[-\pi/2, \pi/2]$ : since four of the eight previously obtained values of  $\gamma$  are certainly in quadrants I or IV, whereas the other four are in quadrants II or III, there are at the same time four positive values of  $\cos\beta$  (hence acceptable) and four negative values, which cannot be considered any further. Hence, up to now, eight acceptable solutions for the pair of angles  $(\gamma, \beta)$  have been found.

Finally, the last two equations in Eq. (8.23) are used to work out the values of  $\cos\alpha$  and  $\sin\alpha$ :

$$\begin{cases} s\alpha = -\frac{r_{23}}{c\beta} \\ c\alpha = -\frac{r_{31}c\beta + r_{23}s\gamma}{s\beta c\beta c\gamma} \end{cases} \quad (8.30)$$

Equation (8.30) provides just one value of  $\alpha$  for each pair  $(\gamma, \beta)$  previously determined. Hence the problem, and also the direct kinematics of the 3-CPU mechanism, admits at most eight solutions [16]. Of course, the problem is comprehensively solved only when all the singular cases are separately treated.

## 8.8 Differential Kinematics

By direct differentiation of the Eq. (8.22), the analytic Jacobian matrix  $\mathbf{J}_A$  is obtained:

$$\begin{bmatrix} \dot{a}_1 \\ \dot{a}_2 \\ \dot{a}_3 \end{bmatrix} = d \begin{bmatrix} 0 & -s\beta s\gamma & c\beta c\gamma \\ c\alpha c\beta & -s\alpha s\beta & 0 \\ -s\alpha s\beta c\gamma - c\alpha s\gamma & c\alpha c\beta c\gamma & -c\alpha s\beta s\gamma - s\alpha c\gamma \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \mathbf{J}_A \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \quad (8.31)$$

The vector of angular velocity  $\boldsymbol{\omega}$  of the platform is related to the derivatives of the Cardan angles by the relation:

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & s\beta \\ 0 & c\alpha & -s\alpha c\beta \\ 0 & s\alpha & c\alpha c\beta \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \mathbf{T} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \quad (8.32)$$

Using Eqs. (8.31) and (8.32), it is possible to work out the expression of the geometric Jacobian  $\mathbf{J}_G$  of the mobile platform:

$$\begin{bmatrix} \dot{a}_1 \\ \dot{a}_2 \\ \dot{a}_3 \end{bmatrix} = \mathbf{J}_A \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \mathbf{J}_A \mathbf{T}^{-1} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \mathbf{J}_G \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (8.33)$$

with

$$\mathbf{J}_G = d \begin{bmatrix} 0 & -c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha c\gamma - s\alpha s\beta s\gamma \\ c\alpha c\beta & 0 & -s\beta \\ -s\alpha s\beta c\gamma - c\alpha s\gamma & c\beta c\gamma & 0 \end{bmatrix} \quad (8.34)$$

It is noted that the geometric Jacobian  $\mathbf{J}_G$  is not a function of geometric parameters; therefore, machine's manipulability cannot be optimized by a proper selection of functional dimensions.

Turning to the analysis of singular poses, it is evident that limbs' structure does not allow for *inverse kinematics singularities*; on the other hand, *direct kinematics singularities* can be found by letting the determinant of the geometric Jacobian  $\mathbf{J}_G$  vanish:

$$\det(\mathbf{J}_G) = d^3 \left[ s^2 \beta - (c\alpha c\gamma - s\alpha s\beta s\gamma)^2 \right] \quad (8.35)$$

The zeros of this equation all lie on closed surfaces of the 3-D space  $\alpha$ ,  $\beta$ , and  $\gamma$ ; their intersections with the coordinate planes generate straight lines, as given by:

$$\begin{cases} \alpha = 0 \rightarrow \beta \pm \gamma = \pm \frac{\pi}{2} \\ \beta = 0 \rightarrow \alpha = \pm \frac{\pi}{2}, \gamma = \pm \frac{\pi}{2} \\ \gamma = 0 \rightarrow \pm \beta = \alpha \pm \frac{\pi}{2} \end{cases} \quad (8.36)$$

Figures 8.13a and 8.13b represent the singularity surfaces in the  $\alpha$ ,  $\beta$ ,  $\gamma$  and in the  $\varphi_x$ ,  $\varphi_y$ ,  $\varphi_z$  spaces, respectively.

Due to the complexity of the geometries represented in Fig. 8.13, it can be useful to plot the value of the determinant on several layers of the workspace, as shown in Fig. 8.14 for the two Cardan angles sets used in this work.

In Fig. 8.15a, moreover, the workspace volumes whose determinant assumes values in the range  $[-0.05, +0.05]$  have been taken out of the representation, while the color map still represents the local determinant value: it is now more appreciable the extent of singularity-free regions inside the workspace, Fig. 8.15b, where the planning of a motion could be performed; e.g., for the mechanism under design a sphere with a radius of about  $50^\circ$  can be internally inscribed.

Finally, the *translation singularities* found in Eq. (8.19) can be easily expressed as a function of articular coordinates  $\theta_{1i}$  by:

$$s\theta_{11}s\theta_{12}s\theta_{13} = c\theta_{11}c\theta_{12}c\theta_{13} \quad (8.37)$$

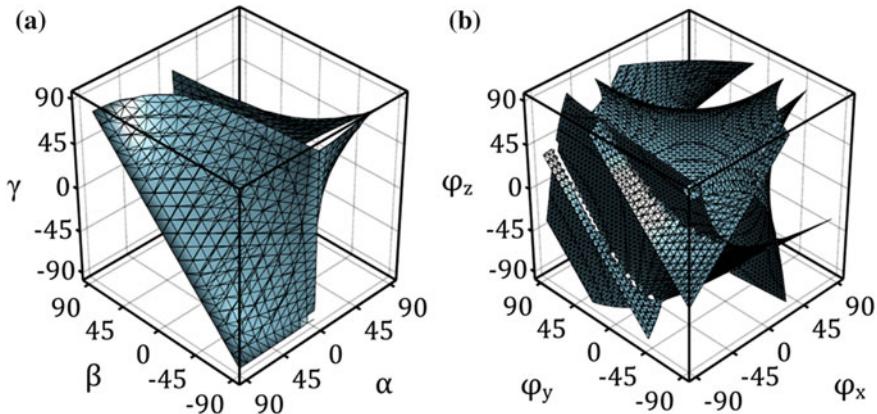
and taking into consideration inverse kinematics in Eq. (8.22) it is also obtained:

$$r_{32}r_{13}r_{21} = r_{11}r_{22}r_{33} \quad (8.38)$$

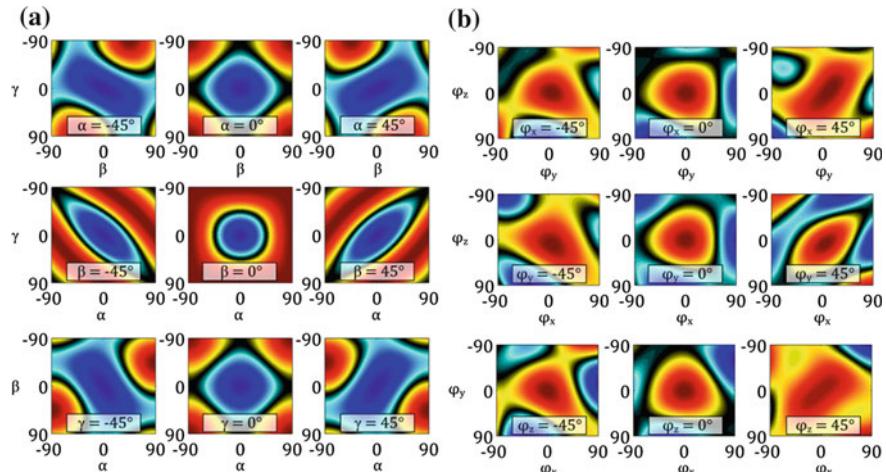
Equation (8.38) is a useful expression of translation singularities in task space, where the elements of the rotation matrix are used; the definition of the rotation matrix in Eq. (8.1) and few trigonometric manipulations yield an alternative expression in function of Cardan angles  $\alpha$ ,  $\beta$ , and  $\gamma$ :

$$s^2 \beta - (c\alpha c\gamma - s\alpha s\beta s\gamma)^2 = 0 \quad (8.39)$$

It is noted that Eq. (8.39) is satisfied for the same values of  $\alpha$ ,  $\beta$ ,  $\gamma$  that make Eq. (8.35) vanish; therefore, translation singularities coincide with direct kinematics singularities, i.e., no additional singular surfaces are present inside workspace.



**Fig. 8.13** Singularity surfaces of  $\mathbf{J}_G(\alpha, \beta, \gamma)$  (a) and  $\mathbf{J}_G^*(\varphi_x, \varphi_y, \varphi_z)$  (b)

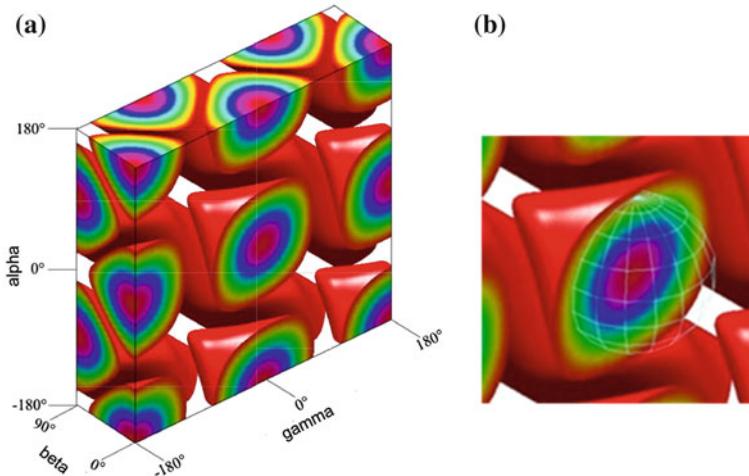


**Fig. 8.14** Color plot of the determinant of  $\mathbf{J}_G(\alpha, \beta, \gamma)$  (left) and  $\mathbf{J}_G^*(\varphi_x, \varphi_y, \varphi_z)$  (right) on several planes (black regions represent almost null values)

## 8.9 Dynamics

### 8.9.1 Analysis of Static Loads

The static analysis is useful in the first phases of machine design for the selection of machine's motors and for a first design of the links, with the related connecting bearings. The base relation is provided as usual by the well-known duality between kinematics and statics, which allows a straightforward assessment of the actuation efforts  $\tau$  needed to balance a moment  $\mathbf{n}_{pl}$  applied at the mobile platform:



**Fig. 8.15** Singularity surface in the  $\alpha$ ,  $\beta$ , and  $\gamma$  space: color map representing local determinant values (a) and close-up view of a connected singularity-free region (b)

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \mathbf{J}_G^{-T} \begin{bmatrix} n_{plx} \\ n_{ply} \\ n_{plz} \end{bmatrix} \quad (8.40)$$

It must be noted that the application of a force  $\mathbf{f}_{pl}$  at the center of the spherical motion does not require balancing forces by the actuators, but it is entirely born by frame bearings: the internal reactions at the bearings caused by the application of the mentioned external wrench have been evaluated as well and used during structural design.

### 8.9.2 Inverse Dynamics Model

Depending on the kind of tasks the wrist will be required to perform, it may not be sufficient to grant assigned static features, but wrist's dynamics should be accounted for as well. In this section, an inverse dynamics model of the 3-CPU mechanism is worked out by using the virtual work principle: it is assumed that frictional forces at the joints are negligible; therefore, the work produced by the constraint forces at the joints is zero and only active forces (including the gravitational effects) must be accounted in the developments.

In the derivation of the model, the notation is based on Fig. 8.11b and the second subscript  $i$  ( $i = 1, 2, 3$ ) indicates the  $i$ th limb while the first subscript  $j$  ( $j = 1, 2$ ) refers to the first or second link, respectively. Namely,  $m_{ji}$  and  $I_{ji}$  are the mass and (central) inertia tensor of the  $j$ th member of the  $i$ th limb;  $\omega_{ji}$  is its angular velocity and  $v_{ji}$  is the linear velocity of its center of mass;  $m_{pl}$ ,  $I_{pl}$ ,  $\omega_{pl}$ ,  $v_{pl}$  are the same quantities referred to the mobile platform. The total wrench of active

and inertial effects acting on the center of mass of  $j$ th member of the  $i$ th limb is written as:

$$\mathbf{F}_{ji} \doteq \begin{bmatrix} \mathbf{f}_{ji} \\ \mathbf{n}_{ji} \end{bmatrix} = \begin{bmatrix} m_{ji}\mathbf{g} - m_{ji}\dot{\mathbf{v}}_{ji} \\ -\mathbf{I}_{ji}\dot{\boldsymbol{\omega}}_{ji} - \boldsymbol{\omega}_{ji} \times (\mathbf{I}_{ji}\boldsymbol{\omega}_{ji}) \end{bmatrix} \quad (8.41)$$

In the same manner, the total wrench acting on the center of mass of the mobile platform is:

$$\mathbf{F}_{pl} \doteq \begin{bmatrix} \mathbf{f}_{pl} \\ \mathbf{n}_{pl} \end{bmatrix} = \begin{bmatrix} m_{pl}(\mathbf{g} - \dot{\mathbf{v}}_{pl}) + \mathbf{f}_e \\ -\mathbf{I}_{pl}\dot{\boldsymbol{\omega}}_{pl} - \boldsymbol{\omega}_{pl} \times (\mathbf{I}_{pl}\boldsymbol{\omega}_{pl}) + \mathbf{n}_e \end{bmatrix} \quad (8.42)$$

where  $\mathbf{f}_e$  and  $\mathbf{n}_e$  are the external force and moment applied to its center of mass; it is accidentally noted that the center of mass of the platform does not coincide with the fixed point O. If  $\boldsymbol{\tau}$  is the vector of the actuation forces and  $\mathbf{q}$  are the corresponding displacements, the principle of virtual work can be written for the present case:

$$(\delta\mathbf{q})^T \boldsymbol{\tau} + (\delta\mathbf{x}_{pl})^T \mathbf{F}_{pl} + \sum_{i=1}^3 \left( \sum_{j=1}^2 ((\delta\mathbf{x}_{ji})^T \mathbf{F}_{ji}) \right) = 0 \quad (8.43)$$

where the vector  $\mathbf{x}_{ji}$  gathers the position of the center of mass of  $j$ th member of the  $i$ th limb and the orientation of the same link and  $\mathbf{x}_{pl}$  expresses the position of the center of mass of the mobile platform and its orientation. It is noted that all the infinitesimal rotations appearing in Eq. (8.43) must be expressed as functions of the angular velocity of the respective link, e.g., for the platform:

$$\delta\mathbf{x}_{pl} = [v_{plx} \ v_{ply} \ v_{plz} \ \omega_{plx} \ \omega_{ply} \ \omega_{plz}]^T \delta t \quad (8.44)$$

Since all the virtual displacements in Eq. (8.44) must be compatible with the constraints, they are not independent but can rather be expressed as functions of an independent set of Lagrangian coordinates; if the Cardan angles  $\boldsymbol{\varphi} = [\alpha, \beta, \gamma]^T$  of the mobile platform are chosen for this purpose, the following relations hold between the introduced virtual displacements:

$$\delta\mathbf{q} = \mathbf{J}\delta\boldsymbol{\varphi} \quad \delta\mathbf{x}_{ji} = \mathbf{J}_{ji}\delta\boldsymbol{\varphi} \quad \delta\mathbf{x}_{pl} = \mathbf{J}_{pl}\delta\boldsymbol{\varphi} \quad (8.45)$$

where  $\mathbf{J}$ ,  $\mathbf{J}_{ji}$  and  $\mathbf{J}_{pl}$  are proper Jacobian matrices that can be found through the usual velocity analysis of the mechanism. Equation (8.43) can be written again as:

$$\delta\boldsymbol{\varphi}^T \left[ \mathbf{J}^T \boldsymbol{\tau} + \mathbf{J}_{pl}^T \mathbf{F}_{pl} + \sum_{i=1}^3 \left( \sum_{j=1}^2 \mathbf{J}_{ji}^T \mathbf{F}_{ji} \right) \right] = 0 \quad (8.46)$$

Since Eq. (8.46) is valid for any virtual displacement  $\delta\boldsymbol{\varphi}$  of the platform, in nonsingular configurations it is:

$$\boldsymbol{\tau} = -\mathbf{J}^{-T} \left( \mathbf{J}_{pl}^T \mathbf{F}_{pl} + \sum_{i=1}^3 \left( \sum_{j=1}^2 \mathbf{J}_{ji}^T \mathbf{F}_{ji} \right) \right) \quad (8.47)$$

Equation (8.47) completely describes manipulator's dynamics; all the elements in it have been worked out and the resulting model has been proofed by comparison with commercial packages' output.

### 8.9.3 Dynamic Analysis in the Task Space

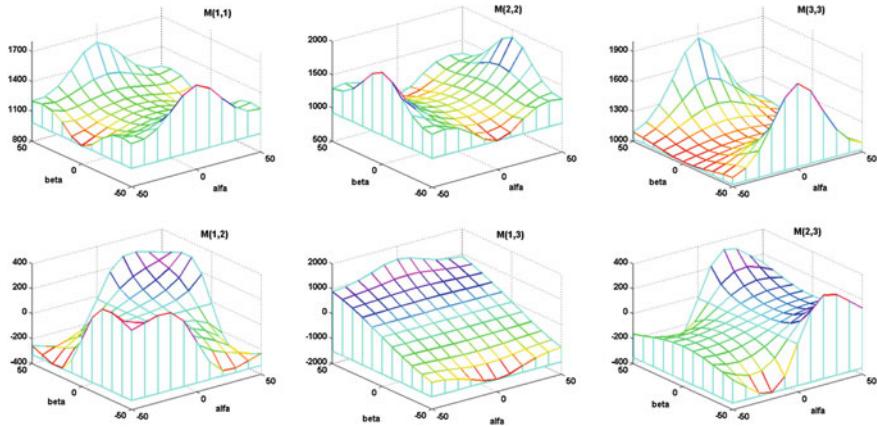
The dynamic expression in Eq. (8.47) is usefully reworked in order to explicit the dependency on a proper set of Lagrangian coordinates and its derivatives. In the case of parallel kinematics machines, the dynamic model results quite naturally written in the task space, due to the (usually) difficult expression of DKP; therefore in the present case, after some cumbersome manipulation, it is obtained:

$$\boldsymbol{\tau}_\varphi - \mathbf{J}_{pl}^T \mathbf{h} = \mathbf{M}_\varphi(\boldsymbol{\varphi}) \ddot{\boldsymbol{\varphi}} + \mathbf{C}_\varphi(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) \dot{\boldsymbol{\varphi}} + \mathbf{G}_\varphi(\boldsymbol{\varphi}) \quad (8.48)$$

with:  $\boldsymbol{\tau}_\varphi = \mathbf{J}^T \boldsymbol{\tau}$ , moments acting at the end effector and corresponding to actual forces  $\boldsymbol{\tau}$  at actuated joints;  $\mathbf{M}_\varphi(\boldsymbol{\varphi})$ , Cartesian mass matrix of the manipulator;  $\mathbf{C}_\varphi(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}})$ , vector of centrifugal and Coriolis terms;  $\mathbf{G}_\varphi(\boldsymbol{\varphi})$ , vector of gravity moments;  $\mathbf{h}$ , vector of external forces and moments acting at the center of mass of the mobile platform.

In view of the realization of possible control schemes based on the inversion of manipulator's dynamics, it is useful to study the variability of mass matrix throughout the workspace. In fact, a major simplification of the model would be yielded by neglecting the six nondiagonal terms of the mass matrix, whether actually allowed by their comparative magnitude; otherwise, all the elements in  $\mathbf{M}_\varphi$  and  $\mathbf{C}_\varphi$  could be considered constant. First simulation results show that, in this case, both simplifying assumptions could be taken into consideration, even if the validity of the reduced models weakens when the operating trajectories get closer to singularity surfaces, as expected.

Figure 8.16, for instance, shows the values of mass matrix' elements in different workspace configurations characterized by null roll angle, i.e.  $\gamma = 0$ : for robot's parameters it has been made reference to the virtual prototype, whose mass properties, presented in the following Table 8.1, are very similar to physical prototype ( $h$  is the total length of the lower part of the three limbs). In Fig. 8.17, the same plots have been normalized by dividing the matrix element by the (local) value of matrix determinant, to allow a relative comparison among elements that have very different magnitudes. It can be seen that near the isotropic point ( $\alpha = \beta = \gamma = 0$ ) the diagonal elements are dominant and matrix variability is limited, while off-diagonal elements show a stronger influence when getting closer to workspace boundaries; moreover, element  $M(3,1)$  is generally an order of magnitude greater than  $M(2,1)$  and  $M(2,3)$ .



**Fig. 8.16** Values of mass matrix' elements for null roll angle, i.e.  $\gamma = 0$  (note the different scales of the plots)

Such behavior gets even more evident if one moves away from the plane  $\gamma = 0$ . The plots have been traced for pitch and yaw angles varying between  $-50^\circ$  and  $+50^\circ$ , because the sphere of  $50^\circ$  radius in the Cardan angles space is completely free of singularities, as shown already in Fig. 8.15.

Other kinds of tests have been performed, aiming at identifying the relative contribute of various dynamic terms: for instance it seems that, even for high dynamics maneuvers, the contribute of gravity is never negligible, while Coriolis and centrifugal forces account for 10–16 % maximum; on the other hand, the mass and inertia of the mobile platform affect very slightly the overall dynamic behavior of the machine, possibly allowing for a major simplification of system's model.

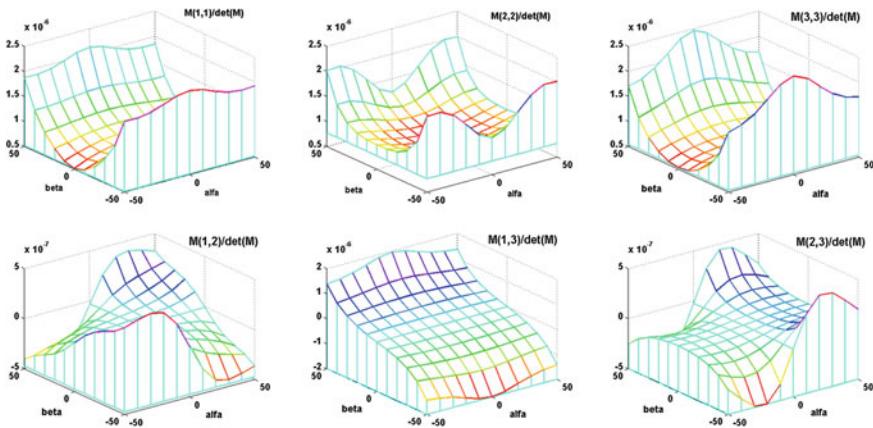
#### 8.9.4 Dynamic Manipulability

Machines' manipulability is usually studied only from a kinematic point of view, but high speed tasks may require to study also the dynamic behavior of the device when required to attain quick manipulation operations; on the other hand, this is particularly important when the device to be studied or designed is a manipulator's wrist, since the capability to develop high accelerations is usually one of the most important features.

By this aim, the dynamic manipulability ellipsoids introduced by Yoshikawa [41, 42] are a useful means to study the dynamic properties of a mechanism: they express graphically the capability of a given device to yield accelerations in all the directions stemming from one attitude of its workspace. As a matter of fact, many other measures of manipulability have been proposed by different researchers since that pioneering work, see e.g., [43], but very few applications dealt with orienting devices.

**Table 8.1** Main data used by the numerical model

Geometrical data			Mass data		
$c$	210	mm	<i>slider</i>	7.15	kg
$d$	490	mm	<i>link 1</i>	1.90	kg
$h$	280	mm	<i>link 2</i>	2.21	kg
$a_i \text{ min}$	319	mm	<i>platform</i>	11.73	kg
$a_i \text{ max}$	661	mm			
$b_i \text{ min}$	130	mm			
$b_i \text{ max}$	210	mm			

**Fig. 8.17** Plots of mass matrix' elements, normalized by determinant value, for null roll angle, i.e.  $\gamma = 0$  (note that all the scales of the graphs are multiplied by  $10^{-6}$  but  $M(1,2)$  and  $M(2,3)$  which are multiplied by  $10^{-7}$ )

Let us consider all the actuation forces  $\tau$  with unit norm:

$$\tau^T \tau = 1 \quad (8.49)$$

By manipulating Eq. (8.48) in order to work out  $\tau$ , it is obtained:

$$\tau = \mathbf{J}^{-T} \mathbf{M}_\varphi (\ddot{\boldsymbol{\phi}} + \ddot{\boldsymbol{\Phi}}_{bias}) \quad (8.50)$$

having defined:

$$\ddot{\boldsymbol{\Phi}}_{bias} = \mathbf{M}_\varphi^{-1} \left( \mathbf{C}_\varphi \dot{\boldsymbol{\phi}} + \mathbf{G}_\varphi + \mathbf{J}_{pl}^T \mathbf{h} \right) \quad (8.51)$$

A meaningful formulation of dynamic manipulability must be expressed as a direct function of the angular acceleration  $\dot{\boldsymbol{\omega}}$ ; therefore, the mapping between the rate of change of the Cardan angles  $\dot{\boldsymbol{\phi}}$  and the angular velocity  $\boldsymbol{\omega}$  must be made explicit:

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & s\beta \\ 0 & c\alpha & -s\alpha c\beta \\ 0 & s\alpha & s\alpha c\beta \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \rightarrow \boldsymbol{\omega} = \mathbf{E}(\boldsymbol{\phi})\dot{\boldsymbol{\phi}} \quad (8.52)$$

$$\dot{\boldsymbol{\omega}} = \mathbf{E}(\boldsymbol{\phi})\ddot{\boldsymbol{\phi}} + \dot{\boldsymbol{\omega}}_1(\boldsymbol{\phi}, \dot{\boldsymbol{\phi}}) \quad (8.53)$$

If  $\ddot{\boldsymbol{\phi}}$  is taken out of Eq. (8.53) and substituted into Eq. (8.50) it is then obtained:

$$\boldsymbol{\tau} = \mathbf{J}^{-T}\mathbf{M}_\varphi\mathbf{E}^{-1}(\dot{\boldsymbol{\omega}} + \dot{\boldsymbol{\omega}}_{bias}) \quad (8.54)$$

having defined:

$$\dot{\boldsymbol{\omega}}_{bias} = -\dot{\boldsymbol{\omega}}_1 + \mathbf{E}\mathbf{M}_\varphi^{-1}\mathbf{J}^T\mathbf{M}_\varphi^{-1}(\mathbf{C}_\varphi\dot{\boldsymbol{\phi}} + \mathbf{G}_\varphi + \mathbf{J}_{pl}^T\mathbf{h}) \quad (8.55)$$

The constraint expressed by Eq. (8.49) can be finally written in the following quadratic form:

$$\dot{\boldsymbol{\Omega}}^T \Gamma(\boldsymbol{\phi}) \dot{\boldsymbol{\Omega}} = 1 \quad (8.56)$$

with obvious meaning of the introduced terms:

$$\dot{\boldsymbol{\Omega}} = \dot{\boldsymbol{\omega}} + \dot{\boldsymbol{\omega}}_{bias} = \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_1 + \mathbf{E}\mathbf{M}_\varphi^{-1}\mathbf{J}^T\mathbf{M}_\varphi^{-1}(\mathbf{C}_\varphi\dot{\boldsymbol{\phi}} + \mathbf{G}_\varphi + \mathbf{J}_{pl}^T\mathbf{h}) \quad (8.57)$$

$$\Gamma(\boldsymbol{\phi}) = \mathbf{E}^{-T}\mathbf{M}_\varphi^T\mathbf{J}^{-1}\mathbf{J}^{-T}\mathbf{M}_\varphi\mathbf{E}^{-1} \quad (8.58)$$

The inspection of Eqs. 8.57 and 8.58 shows that gravity merely induces a translation of the dynamic manipulability ellipsoid while in general velocity has a complex, nonnegligible effect on manipulability. Making reference to the remarkable case of a fixed platform ( $\dot{\boldsymbol{\phi}} = \mathbf{0}$ ) with no external or gravity action applied ( $\mathbf{h} = \mathbf{G}_\varphi = \mathbf{0}$ ), Eq. (8.56) provides:

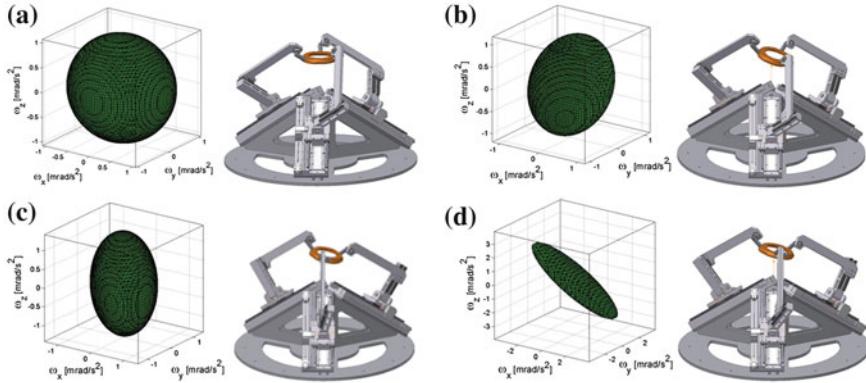
$$\dot{\boldsymbol{\omega}}^T \Gamma(\boldsymbol{\phi}) \dot{\boldsymbol{\omega}} = 1 \quad (8.59)$$

The quadratic form in Eq. (8.59) represents an ellipsoid in the Cartesian space of the angular accelerations: its eigenvalues express the square root of the maximum and minimum accelerations that can be developed with unit actuator forces while the eigenvectors represent the associated directions in the orientation space. Figure 8.18 represents graphically some dynamic manipulability ellipsoids of the robot in the poses sketched aside.

Two dynamic figures can be defined in order to sum up some relevant aspects of the dynamic manipulability analysis. The *measure of the dynamic manipulability*,  $w$ , defined as:

$$w = \sqrt{\det(\Gamma(\boldsymbol{\phi}))} \quad (8.60)$$

Results proportional to the volume of the manipulation ellipsoid, and therefore yields an overall information on the global manipulation capabilities, but fails



**Fig. 8.18** Dynamic manipulability ellipsoids at different poses ( $\alpha$ ,  $\beta$ ,  $\gamma$ ): (0°, 0°, 0°) **(a)** (20°, 20°, -5°) **(b)** (40°, 40°, 10°) **(c)** (54°, 53°, 10°) **(d)**

to capture the closeness to singular configurations or even the anisotropy of local dynamics. On the other hand, the *index of dynamic manipulability*,  $i$ , is independent from the volume of the ellipsoid and vanishes close to singular configurations:

$$i = \sqrt{\lambda_{\min}/\lambda_{\max}} \quad (8.61)$$

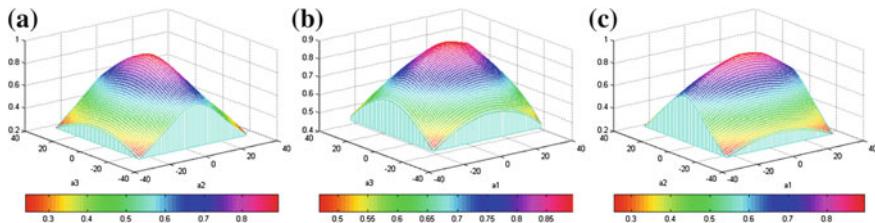
with  $\lambda_{\min}$ ,  $\lambda_{\max}$  minimum and maximum eigenvalues of the matrix  $\Gamma(\boldsymbol{\varphi})$ .

Usually, a dynamic optimization of the design aims at trying to maximize the global dynamic manipulability of the wrist while still guaranteeing a minimum threshold of the local features. Figure 8.19, for instance, shows sample plots of the index of dynamic manipulability as a function of actuators strokes  $a_i$  on the three coordinate planes for the final design of the wrist. For instance, the four configurations shown in Fig. 8.18, the indexes assume the values: 0.7755, 0.1374, 0.3571, and 0.0341 respectively, while it has been obtained a mean value of  $i_{ave} = 0.502$  over the central ±30° span of the workspace.

## 8.10 Prototype Design

The design of a first prototype has been developed, with the aim of obtaining high dynamic performance; as reference figures, the following requirements have been posed:

- orientation range (elevation and azimuth): 150°
- maximum angular velocity: 500°/s
- maximum angular acceleration: 5,000°/s<sup>2</sup>



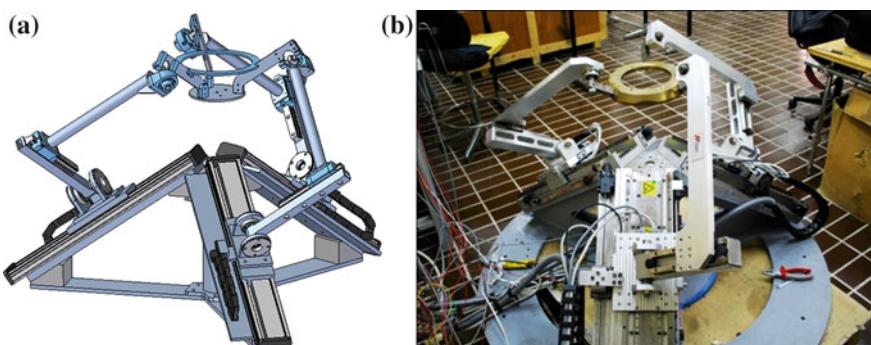
**Fig. 8.19** Plots of the index of dynamic manipulability as a function of actuators strokes on the three coordinate planes  $a_1 = 0$  (a)  $a_2 = 0$  (b)  $a_3 = 0$  (c)

- spatial resolution:  $0.01^\circ$
- overall dimensions of the machine: maximum volume of  $1\ m^3$ .

The particular form of the Jacobian matrix in Eq. (8.34) does not allow for a mechanical design based on the optimization of kinematic properties, since  $\mathbf{J}_G$  is not function of robot's geometry; therefore, heuristic considerations and several computer simulations allowed to refine the design.

Figure 8.20a sketches the design of a prototype meeting the posed requirements; in Fig. 8.20b a picture of the machine is presented. The limbs are made of *avional* (an aluminum-copper alloy) in order to join good mechanical properties with a lightweight construction. The mobile platform is made of bronze; therefore, allowing the precise machining in a single placement of the three journal bearings that have to meet orthogonally in a single point; in this way, it has been obtained a high stiffness together with precise geometrical alignments. It must be noted that such revolute joints are idle, since no rotation occurs at all if all the manufacturing and mounting conditions are correctly satisfied.

The actuation is based on three brushless linear motors Phase WVS 20.6.3, able to provide a maximum thrust of  $184\ N$  at the speed of  $6\ m/s$ , with a maximum acceleration of  $14.3\ g$  and is controlled by Nation Instrument hardware (Flex-motion/PXI architecture). Table 8.2 collects the most relevant data of prototype actuation.



**Fig. 8.20** CAD model of spherical wrist (a) and picture of a first laboratory prototype (b)

**Table 8.2** Main actuation data

Characteristics of the three linear motors			
$M_s$	2.95	kg	Stator mass
$K_t$	58	N/A	Torque constant
$I_n$	3	A	Nominal supply current
$T_n$	184	N	Nominal thrust
$v_n$	6	m/s	Nominal speed

## 8.11 Conclusions

The chapter has presented the mechanical features of robotic wrists, especially addressing the kinematics of mechatronic devices to be used in grasping and manipulation tasks. A powerful kinematical tool for the synthesis of such kind of machines, based on Lie algebra, has been presented and an innovative spherical parallel wrist developed at the Polytechnic University of Marche in Ancona has been described, revisiting all the main design steps, from kinematic synthesis up to physical prototyping.

The mechanical architecture of the machine is based on the 3-CPU kinematics, that allows simple actuation and is characterized by easy direct and inverse relations; the 3-CRU variant has been discarded due to a much more complex kinematics but it could be taken into consideration for the design of mini- or micromachines.

Position and rate kinematics have been worked out in closed form and all the singularity surfaces have been analyzed: it has been pointed out that the mechanism does not possess inverse kinematics singularities, while direct kinematics singularities and translation singularities lie on the same closed surface. The inner space, where motion paths can be safely planned, has been identified, and unfortunately it cannot be enlarged by kinematics optimization because machine's Jacobian does not depend on geometrical parameters.

The kinematic concept of the mechanism has been developed up to the stage of designing a spherical wrist to be used in cooperation with a translation machine for assembly tasks. The physical prototyping of the machine allowed to validate the good properties envisaged during the design phase, e.g., the high dynamics mainly provided by the three direct drive linear motors. On the other hand, few disadvantages of the concept itself have showed up: they are mainly due to the scarce accessibility of the center of the spherical motion, which is common to most parallel wrists, and to the difficult assembly, which requires a precise alignment of joints axes: this problem has been partially overcome by the manufacturing of specific fixtures that are characterized by very high accuracy and are used while assembling the machine.

In view of the strict cooperation of this wrist with a parallel translating machine for grasping and assembly tasks, specific position and interaction control schemes have been developed, but the related results have been presented already elsewhere by Carbonari et al. [44] and Bruzzone and Callegari [45].

## References

1. Charles SK, Hogan N (2010) Dynamics of wrist rotations. *J Biomech* 44:614–621. doi:[10.1016/j.jbiomech.2010.11.016](https://doi.org/10.1016/j.jbiomech.2010.11.016)
2. Pennestrì E (1991) On the kinematic analysis of geared robotic wrists. *Meccanica* 26: 155–160. doi:[10.1007/BF00429883](https://doi.org/10.1007/BF00429883)
3. Rosheim ME (1989) Robot wrist actuators. John Wiley & Sons Inc, New York
4. Saravanan R, Ramabalaji S, Godwin Raja Ebenezer N, Dharmaraja C (2009) Evolutionary multi criteria design optimization of robot grippers. *Appl Soft Comput* 9:159–172. doi:[10.1016/j.asoc.2008.04.001](https://doi.org/10.1016/j.asoc.2008.04.001)
5. Schäfer C, Dillmann R (2001) Kinematic design of a humanoid robot wrist. *J Robot Syst* 18:747–754. doi:[10.1002/rob.8113](https://doi.org/10.1002/rob.8113)
6. Fichter EF, Kerr DR, Rees-Jones J (2009) The Gough–Stewart platform parallel manipulator: a retrospective appreciation. *Proc Inst Mech Eng C: J Mech Eng Sci* 223:243–281. doi:[10.1243/09544062JMES1137](https://doi.org/10.1243/09544062JMES1137)
7. Di Gregorio R (2002) A new family of spherical parallel manipulators. *Robotica* 20:353–358. doi:[10.1017/S0263574702004174](https://doi.org/10.1017/S0263574702004174)
8. Bürger T, Laible U, Pritschow G (2001) Design and test of a safe numerical control for robotic surgery. *CIRP Ann Manuf Technol* 50:295–298. doi:[10.1016/S0007-8506\(07\)62125-8](https://doi.org/10.1016/S0007-8506(07)62125-8)
9. Takaiwa M, Noritsugu T (2005) Development of wrist rehabilitation equipment using pneumatic parallel manipulator. In: Proceedings of the IEEE international conference of robotics automation, Barcelona, Spain, pp 2302–2307. doi:[10.1109/ROBOT.2005.1570456](https://doi.org/10.1109/ROBOT.2005.1570456)
10. Arata J, Kondo H, Ikeda N, Fujimoto H (2011) Haptic device using a newly developed redundant parallel mechanism. *IEEE Trans Robotics* 27:201–214. doi:[10.1109/TRO.2010.2098272](https://doi.org/10.1109/TRO.2010.2098272)
11. Unal R, Patoglu V (2008) Optimal dimensional synthesis of force feedback lower arm exoskeletons. In: Proceedings of the IEEE/RAS-EMBS international conference on biomedical robotics and biomechatron, Scottsdale, USA, pp 329–334. doi:[10.1109/BIOROB.2008.4762871](https://doi.org/10.1109/BIOROB.2008.4762871)
12. Saltaren RJ, Sabater JM, Yime E, Azorin JM, Aracil R, Garcia N (2007) Performance evaluation of spherical parallel platforms for humanoid robots. *Robotica* 25:257–267. doi:[10.1017/S0263574706003043](https://doi.org/10.1017/S0263574706003043)
13. Asada H, Granito C (1985) Kinematic and static characterization of wrist joints and their optimal design. In: Proceedings of the IEEE international conference on robotics and automation, St Louis, USA, pp 244–250. doi:[10.1109/ROBOT.1985.1087324](https://doi.org/10.1109/ROBOT.1985.1087324)
14. Gosselin CM, Angeles J (1989) The optimum kinematic design of a spherical three-degree-of-freedom parallel manipulator. *ASME J Mech Trans and Automat Des* 111:202–207
15. Alizade RI, Tagiyev NR, Duffy J (1994) A forward and reverse displacement analysis of an in-parallel spherical manipulator. *Mech Mach Theory* 29:125–137. doi:[10.1016/0094-114X\(94\)90025-6](https://doi.org/10.1016/0094-114X(94)90025-6)
16. Innocenti C, Parenti-Castelli V (1993) Echelon form solution of direct kinematics for the general fully-parallel spherical wrist. *Mech Mach Theory* 28:553–561. doi:[10.1016/0094-114X\(93\)90035-T](https://doi.org/10.1016/0094-114X(93)90035-T)
17. Lee JJ, Chang S-L (1992) On the kinematics of the UPS wrist for real time control. In: Proceedings of the 22nd ASME biennial mechanisms conference and robotics, spatial mechanisms and mechanical systems, Scottsdale, USA, pp 305–312
18. Kong X, Gosselin CM (2004) Type synthesis of 3-DOF spherical parallel manipulators based on screw theory. *J Mech Des* 126:101–108. doi:[10.1115/1.1637655](https://doi.org/10.1115/1.1637655)
19. Kong X, Gosselin CM (2004) Type synthesis of three-degree-of-freedom spherical parallel manipulators. *Int J Robotics Res* 23:237–245. doi:[10.1177/0278364904041562](https://doi.org/10.1177/0278364904041562)
20. Karouia M, Hervé JM (2002) A family of novel orientational 3-DOF Parallel Robots. In: Proceedings of the 14<sup>th</sup> RoManSy, Udine, pp 359–368

21. Karouia M, Hervé JM (2000) A three-Dof tripod for generating spherical rotation. In: Lenarcic J, Stanisic MM (eds) Advances robot kinematics. Kluwer, Dordrecht
22. Karouia M, Hervé JM (2006) Non-overconstrained 3-DOF spherical parallel manipulators of type: 3-RCC, 3-CCR, 3-CRC. *Robotica* 24:85–94. doi:[10.1017/S0263574705001827](https://doi.org/10.1017/S0263574705001827)
23. Fang Y, Tsai L-W (2004) Structure synthesis of a class of 3-DOF rotational parallel manipulators. *IEEE Trans Robotics and Autom* 20:117–121. doi:[10.1109/TRA.2003.819597](https://doi.org/10.1109/TRA.2003.819597)
24. Di Gregorio R (2001) Kinematics of a new spherical parallel manipulator with three equal legs: the 3-URC wrist. *J. Robotic Syst* 18:213–219. doi:[10.1002/rob.1017](https://doi.org/10.1002/rob.1017)
25. Di Gregorio R (2001) A new parallel wrist using only revolute pairs: the 3-RUU wrist. *Robotica* 19:305–309. doi:[10.1017/S0263574700003192](https://doi.org/10.1017/S0263574700003192)
26. Di Gregorio R (2004) The 3-RRS wrist: a new, simple and non-overconstrained spherical parallel manipulator. *J Mech Des* 126:850–855. doi:[10.1115/1.1767819](https://doi.org/10.1115/1.1767819)
27. Di Gregorio R (2004) Kinematics of the 3-RSR wrist. *IEEE Trans Robotics* 20:750–754. doi:[10.1109/TRO.2004.829451](https://doi.org/10.1109/TRO.2004.829451)
28. Huda S, Takeda Y (2007) Kinematic analysis and synthesis of a 3-URU pure rotational parallel mechanism with respect to singularity and workspace. *J Adv Mech Des Syst Manuf* 1:81–92. doi:[10.1299/jamds.m.1.81](https://doi.org/10.1299/jamds.m.1.81)
29. Yi B-J, Chung GB, Na HY, Kim WK, Suh IH (2003) Design and experiment of a 3-DOF parallel micromechanism utilizing flexure hinges. *IEEE Trans Robotics* 19:604–612. doi:[10.1109/TRA.2003.814511](https://doi.org/10.1109/TRA.2003.814511)
30. Lusk CP, Howell LL (2008) Components, building blocks, and demonstrations of spherical mechanisms in microelectromechanical systems. *J Mech Des* 130(034503–1):4. doi:[10.1115/1.2829914](https://doi.org/10.1115/1.2829914)
31. Lusk CP, Howell LL (2008) Spherical bistable micromechanism. *J Mech Des* 130(045001–1):6. doi:[10.1115/1.2885079](https://doi.org/10.1115/1.2885079)
32. Callegari M, Cammarata A, Gabrielli A, Ruggiu M, Sinatra R (2009) Analysis and design of a spherical micromechanism with flexure hinges. *J Mech Des* 131:051003–051013. doi:[10.1115/1.3086796](https://doi.org/10.1115/1.3086796)
33. Palpacelli M-C, Palmieri G, Callegari M (2012) A redundantly actuated 2-dof mini pointing device. *ASME J Mechanisms and Robotics* 4(3):031012. ISSN: 1942-4302, eISSN: 1942-4310, doi:[10.1115/1.4006833](https://doi.org/10.1115/1.4006833)
34. Fontana M, Frisoli A, Salsedo F, Bergamasco M (2006) Kinematics of a new 2-DoF wrist with high angulation capability. Proc IEEE Int Conf Robot Autom, Orlando. doi:[10.1109/ROBOT.2006.1641924](https://doi.org/10.1109/ROBOT.2006.1641924)
35. Carricato M, Parenti-Castelli V (2004) A novel fully decoupled two-degrees-of-freedom parallel wrist. *Int J Robotics Res* 23:661–667. doi:[10.1177/0278364904044077](https://doi.org/10.1177/0278364904044077)
36. Callegari M, Marzetti P, Olivieri B (2004) Kinematics of a parallel mechanism for the generation of spherical motions. In: Lenarcic J, Galletti CU (eds) Advances in robot kinematics. Kluwer, Dordrecht
37. Zlatanov D, Bonev IA, Gosselin CM (2002) Constraint singularities of parallel mechanisms. In: Proceedings of the IEEE international conference on robotics and automation, Washington, USA, pp 496–502
38. Bonev IA, Zlatanov D (2001) The mystery of the singular SNU translational parallel robot. Available online at <http://www.parallelmechanisms.org/Reviews/Review004.html>
39. Carbonari L (2012) Extended analysis of the 3-CPU reconfigurable class of parallel robotic manipulators, Ph.D. thesis, Polytechnic University of Marche, Ancona
40. Callegari M, Palpacelli M-C (2008) Prototype design of a translating parallel robot. *Meccanica* 43:133–151. doi:[10.1007/s11012-008-9116-8](https://doi.org/10.1007/s11012-008-9116-8)
41. Yoshikawa T (1985) Dynamic manipulability of robot manipulators. *J Robotic Syst* 2: 113–124
42. Yoshikawa T (2000) Erratum to dynamic manipulability of robot manipulators. *J Robotic Syst* 17:449. doi:[10.1002/1097-4563\(200008\)17:8<449::AID-ROB5>3.0.CO;2-M](https://doi.org/10.1002/1097-4563(200008)17:8<449::AID-ROB5>3.0.CO;2-M)

43. Di Gregorio R, Parenti-Castelli V (2002) Dynamic performance indices for 3-DOF parallel manipulators. In: Lenarcic J, Thomas F (eds) *Advances in robot kinematics*. Kluwer, Dordrecht
44. Carbonari L, Bruzzone L, Callegari M (2011) Impedance control of a spherical parallel platform. *Int J Intell Mechatron Robot* 1:40–60. doi:[10.4018/ijimr.2011010103](https://doi.org/10.4018/ijimr.2011010103)
45. Bruzzone L, Callegari M (2010) Application of the rotation matrix natural invariants to impedance control of purely rotational parallel robots. *Adv Mech Eng*. doi:[10.1155/2010/284976](https://doi.org/10.1155/2010/284976)

## **Part III**

# **Control and Motion Planning**

# **Chapter 9**

## **A Survey on Different Control Techniques for Grasping**

**Balaguer Carlos, Gimenez Antonio, Jardón Alberto, Cabas Ramiro  
and Martínez de la Casa Santiago**

**Abstract** One of the main objectives of the robotics is to make the systems able to interact, modify, transmit, and receive information from the human environment. They reach the possibility to replace the human being in simple and basic functions with better results. In this way, robotic hands play a relevant role since they can perform different tasks, such as holding and manipulating, reaching visual communication, and obtaining direct contact with the environment, interacting and even modifying it. This chapter will focus on the different control techniques depending on the electromechanical design: linear and nonlinear control, and more advanced controls techniques. Finally, some solutions will be detailed through mechanical systems, and it will be explained how they can affect the development of new robotic hands, studying the case of underactuated architectures developed at roboticslab, which are implemented in a robots designed at RoboticsLab: ASIBOT (personal assistive robot) and TEO, the UC3M's second version of a full-sized humanoid.

---

B. Carlos (✉) · J. Alberto · M. de la Casa Santiago  
ROBOTICSLAB: Robotics Laboratory, University of Carlos III,  
C. Butarque, 15 28190 Leganes, Spain  
e-mail: balaguer@ing.uc3m.es

J. Alberto  
e-mail: ajardon@ing.uc3m.es

M. de la Casa Santiago  
e-mail: scasa@ing.uc3m.es

G. Antonio  
Automatic Control, Electronics and Robotics Research Group, University of Almería,  
Ctra. De la Playa, s/n 04120 Almería, Spain  
e-mail: agimfer@ual.es

C. Ramiro  
Arquimea Engineering, Avda. Gregorio Peces Barba, 1 28918 Leganés, Spain  
e-mail: rcabas@arquimea.com

## 9.1 Introduction

One of the main objectives of the robotics is to make the systems able to interact, modify, transmit, and receive information from the human environment [4]. They reach the possibility to replace the human being in simple and basic functions with better results. In this way, robotic hands play a relevant role because they can perform different tasks, such as holding and manipulating, reaching visual communication, and obtaining direct contact with the environment, interacting and even modifying it. When we refer to robotic hands, the first thing that comes up to our mind is a robotic system that tries to imitate, in the closest way, the skills and shapes that human hands have. There are many different robotic hands and each of them marks a determined improvement in a specific aspect, thanks to new technology used in its development. They have been designed using an innovative mechanic system and a different control technique adequate to the mechanic hand configuration.

These two concepts, as it will be detailed in this chapter, must be closely related because the efficient use of one of them depends on the improvement of the other. Based on this classification, it is possible to divide robotic hands into two types: the multi-actuated robotic hands, directed by technology and helped by special mechanisms, and the underactuated robotic hands, focused on complex and innovative mechanic systems. The second classification adds a new system concept that allows generating several and independent tasks fulfilled with only one actuator, referring to a new generation of robotic hands able to manipulate in a more dexterous way.

In this chapter, robotic hands will be analyzed from its technological, mechanical, and kinematic configuration aspects. First, there will be a brief explanation of robotic hands that have incorporated innovative mechanisms or new technology. Afterwards, the chapter will focus on the different control techniques depending on the electromechanical design: linear and nonlinear control, and more advanced controls techniques. Finally, some solutions will be detailed through mechanical systems, and it will be explained how they can affect the development of new robotic hands, studying the case of underactuated architectures developed at Roboticslab, which are implemented in their designed robots: ASIBOT (personal assistive robot), and TEO, the UC3M's second version of a full-sized humanoid.

## 9.2 Robotic Hands Classification

The control system on the grippers depends on the mechanic system in use. When using industrial grippers, the holding control is only focused on controlling the gripper's force applied over the object to be grabbed, normally similar to that of two fingers. In these cases, the main goal is to avoid the object's fall and its proper

hold. For these cases, the classic joint force control system is the most frequently used. This control has been implemented in the effort control system used by the joints of the handling robots [36].

Nowadays, the studies of the human hand and its imitation are one of the most researched areas in robotics. The reason why is because hands are the perfect universal tool and the majority of the objects and tools used every day are designed to adapt the human hand dexterity. The technological advances regarding actuators, sensors, and materials show that advance supporting the scientific evolution regarding robotic area are very important. However, we are far away from achieving a robotic handler which will be able to substitute the human hand.

Although in an industrial level, we are not only trying to develop robotic grippers for a specific application and a specific holding task, the scientific community is considering to use robotic hands in prosthesis for human beings. This challenge opens, in a scientific level, a wide range of problems that need to be sorted out before all this can become real.

Even that, during the last years, it has been developed a great number of robotic hands designed to research some specific behaviors of the human hand, using different technological advances, achieving really interesting improvements.

It can be said that the first works about the holding control were focused on industrial gripper used by the handler robots, and on the coordinate performance of a group of robots when moving loads together or with the help of a person [20, 29, 37]. Since the 1970s, with Okada's first works [28], it was developed a hand provided with multiple fingers, and a system of wires and pulleys for transmitting the movement to each one of the fingers.

The holding mechanisms used by the robot manipulator and humanoid robots can be divided into three well defined big groups: robotic grippers, underactuated robotic hands, and fully actuated robotic hands. These last ones can be also divided into following different criteria, such as their structural configuration, their aimed application in a behavior level, or their implemented technology. The main difference among these groups has to do with the relation between the degree of freedom (DOFs) they have, and the quantity of actuators composing their actuator system.

One of the main causes that can justify the existence of these three groups is the complexity of its control system, especially when the actuator system is formed by a considerable quantity of actuators; for an actuator system to be really efficient, the robotic system should have a sensorial system similar or more complex, and a structure that allows taking advantage of all the movement provided by the forenamed actuator system. All these elements need to occupy a specific space that is still far away from that of a human hand. For this reason, in a gripper for a robot manipulator, the problem regarding the weight and volume of the elements forming it is not as important as in case of the robotic hands.

A perfect combination among the actuator system, the sensorial system, the control system, and the structure supporting it can transform a robotic hand into a really versatile tool, because it can allow us to perform manipulating and handling tasks as well as a human hand can do. The problem is that the techniques studied in a theoretical level are not optimized or simplified enough for the used

technology to allow it or; simply, these techniques goes on or they become more complex and the technology goes always behind. It is the reason because we are looking for design and performing studies for achieving a simplification of the system without giving up any efficiency.

*Industrial Grippers.* The majority of the industrial grippers use common pneumatic cylinders as actuators. It is very difficult to control the position due to its high nonlineal equations describing its dynamic behavior; therefore, they are not really interested on solving the problem with the holding control. In this case, the research groups develop grippers with electric activation, and they apply the same techniques for force control used in the rest of the electrical activation in the robot manipulator.

*Multiactuated Robotic Hands.* These hands have the same number of actuators than DOFs. Due to the great number of actuators in it, they have to be placed close to the wrist, in a place no meaning an increase of the finger's inertia. These actuators are usually electric motors or pneumatic actuators (McKibben muscles). On the other hand, it is necessary to have transmission mechanisms for each actuator in each joint. Recently, it has been developed a group of robotic hands with reduced sized actuators that can be placed close to the joints, so the number of transmission elements can be reduced.

If the hand enjoys a lot of freedom degrees, and therefore it has a lot of actuators, the control strategy's tendency is to be more complex because the number of elements to be coordinated is bigger. However, the degree of skill and handling ability that can be reached is higher than that of the underactuated hands.

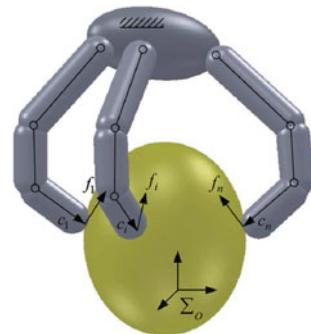
*Underactuated Robotics Hands.* The underactuation concept is known since many years ago and it is thought that, even if it can put into motion a robotic system using a minor quantity of actuators than the DOFs, it presents a great restriction in a functional level because the independent control system of each DOF is quite limited, and their control variables depend on the interaction between the system and the environment.

The performed advances regarding the underactuation subject are majority related to the design of new innovative mechanisms that allow the optimization of the system conveying a feeling of a great control of the actions. These mechanisms can be seen in the industry world in almost every complex mechanic system dotted with any type of actuator for generating movement.

### 9.2.1 Analysis and Synthesis of Grasping

Figure 9.1 shows a basic situation where a hand with  $n$  three-jointed fingers is grasping an object in 3-D space. Various analytical studies have been done extensively for this or some more general situations. Among basic concepts developed are grasp map (or grasp matrix), hand Jacobian, form and force closure grasps, and active and passive closures.

**Fig. 9.1** Multifingered hand grasping an object



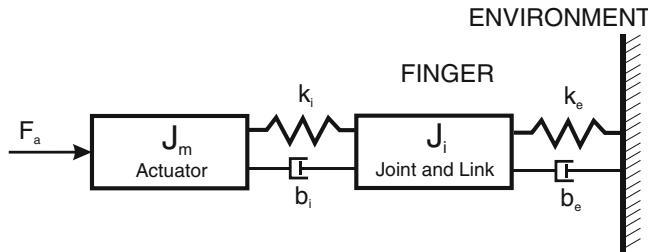
Consider the case where the fingertips and the object is making a point contact with friction, that is, finger  $i$  can exert a fingertip force  $f_{Ci} \in \mathbb{R}^3$  (3-D vector) on the object, but not moment, within the friction cone at the contact point  $c_i$ ). Then Grasp matrix [26] is the matrix  $\mathbf{G}$  transforming the contact forces at the fingertips  $\mathbf{f}_C = [\mathbf{f}_{C1}^T \mathbf{f}_{C2}^T \cdots \mathbf{f}_{Cn}^T]^T$  to the resultant force/moment vector  $t_B \in \mathbb{R}^6$  where  $\mathbf{t}_B = \mathbf{G}\mathbf{f}_C$ , and applying the principle of virtual work we obtain from the relation between the speed of the Grasp contact points and the object's gravity center  $\mathbf{v}_C = \mathbf{G}\mathbf{v}_B$ . If there is no slip between the object and the fingers' grasping points, we can also put in relation the speed of the object with the articular speed of the fingers through the expression  $\mathbf{J}\dot{\mathbf{q}} = \mathbf{G}^T\mathbf{v}_B$ , where  $\mathbf{J}$  corresponds to the Hand Jacobian matrix. We must point out that, in order to plan the way of grasping, it is normal to find solutions in the technical literature about artificial intelligence as in Yao et al. [39]; yet it is not the same for the study of the control in the actuators moving the joints of a robotic hand.

### 9.2.2 Mechanic Model of the Hands

The mechanic design and the control in an industrial gripper or in a robotic hand depend greatly on the kinematic chain of the mechanism and on the space available for hosting the actuators, mechanic systems, and sensors. We can find several solutions in the literature:

- Emplacement of the motors as close to the actuated joint as possible, reducing the number of transmission mechanisms to be used.
- Emplacement of the actuators in the palm or forearm; in this case, we have to use transmission mechanism for the actuators' movement in the joint place.

The dynamic models in a robotic hand, whose actuators are close to the joints, are similar to those in a traditional robot. Each finger is considered as another manipulator. If the actuators are placed far away from the joints, it is normal to find problems regarding to a nonlineal behavior in the mechanic systems such as the friction, the



**Fig. 9.2** Model of a robot joint with transmission flexibility

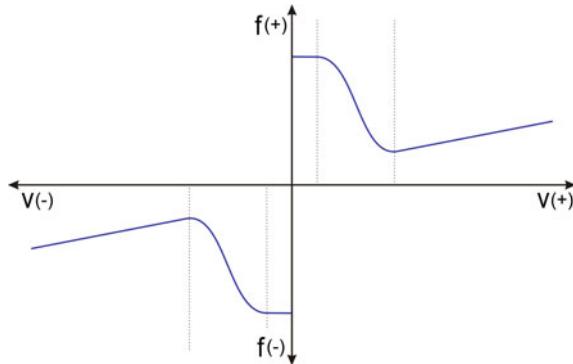
backslash, or the compliance of the transmission systems. An extra problem could be the emplacement of the sensors, which are usually force and position sensors in the actuator or in the joint, because sometimes, it does not reflect faithfully the contact forces produced when the finger in one hand gets in touch with the surrounding environment or the grasped object. Furthermore, this type of sensors can offer mistaken lectures due to small vibrations, disturbances, and so on.

If the actuator is very close to the joint that is about to move its mechanic model, it can be represented by the effort performed by the motor, its inertial mass, stiffness constant, and viscous friction. The finger can be represented by its inertial mass. We consider the presence of other stiffness constant and viscous friction between the final sensor present in the finger and the object to be handled. The dynamic model used for the finger of an actuator, and a DOF is the one represented in Fig. 9.2. Using this mechanic model is very easy to implement any classic control strategy (PID regulators, state space control, etc.) just if we can avoid the appearance of static frictions in its design and even if the hand has several fingers and joints in each one.

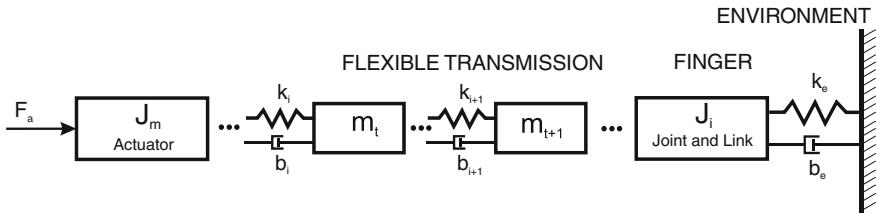
The majority of these mechanic systems need gears or other transmission elements in which the appearance of the friction phenomena is inevitable. In fact, the viscous lineal friction, represented in Fig. 9.3 by the coefficient  $b_i$ , is not a lineal function that depends on the speed, but the value highly nonlineal where other phenomena such as stiction and Coulomb friction appear. This behavior is reflected in the Stribeck curve, where we can appreciate that both are discontinuous at zero speed. These effects add nonlineal terms to the block in the transference of the mechanic model of the fingers in a gripper or robotic hand.

That nonlinearity can create limit cycles and a stability dependent of the entrance signal, an aspect to be taken into account during the mechanical design of the fingers as well as when designing its control architecture [31, 34].

If the motors are placed in positions far from the joints, the most frequently used transmission elements are the wires and pulleys, which add a greater nonlinearity to the finger's behavior. In this case, the wires can be modeled as in the appearance of a rigid solids chain, where each element has its own inertia, stiffness constant, and viscose. The model can be seen in Fig. 9.4 [24]. The effect produced by the wires in a control system increases the nonlinearity of the system, because the wires are flexible elements and their behavior is not that of the rigid solid ones. The bandwidth of



**Fig. 9.3** Stribeck curve



**Fig. 9.4** Model of a robot joint with tendon-based transmission and pulleys

the mechanic system is reduced so its instability can increase when performing an impedance control. Generally speaking, the actuators and sensors are physically placed in different points of the flexible structure, as it happens in case of fingers with tendons. There are unstable modes in a close trap system [10].

### 9.3 Implemented Controls in Robotic Hands

In this section, we introduce the control methods more frequently used in the robotic hands, multi and underactuated that authors have found among the different researching groups working in this topic. There will not be all the currently working ones; however, the authors think that there are the most representative control strategies. The majority of the hands studied by the authors have this type of control or some small variation from the ones represented below. As we can see, the controls are related to the points analyzed in the previous section: mechanic models, friction problems, and multi or underactuated mechanisms.

### 9.3.1 Simple Classic Control

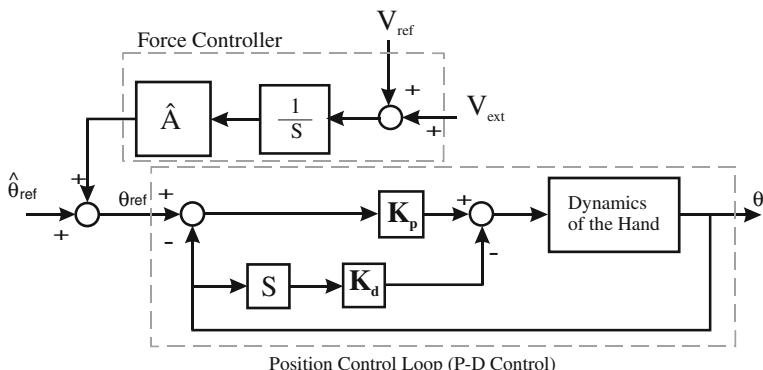
One of the first hands using the classic control was the Utah/MIT hand. Each of his four fingers has four rotational DOFs [27, 33]. Position control is derived from a linear model of the mechanism's dynamics using a pole placement controller. Other example of hand and first implementation is the Gifu HAND II [17].

The underactuated hands usually have simple controls, because the goal is to create hands that occupied little space and have a low cost, so it is necessary to implement a simpler control. An example is the one developed by Zollo [42]. A PD control with compensation of the gravitational and strain energy has been implemented. The gravitational energy term compensates the position of each one of the phalanges of the hand regarding the function of its position, because it is a very light hand. Its mechanic system has several pre-charged spring in the joints whose strain energy needs to be compensate when sending the reference signal to the actuators in the hand. Its control law can be seen in the Eq. (9.1).

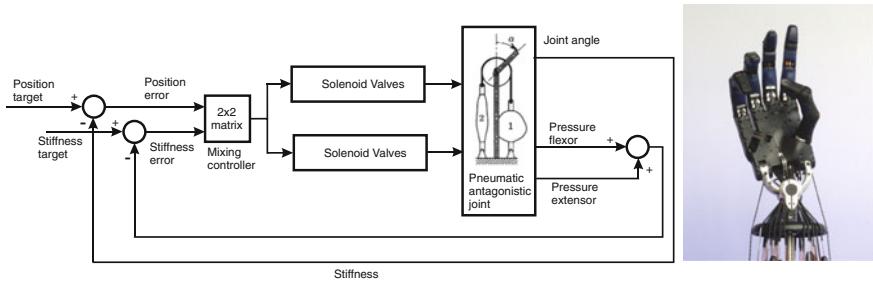
$$\tau = K_P(q_d - q_{act}) + K_D(\dot{q}_d - \dot{q}_{act}) + g(q_{act}) + \tau_e \quad (9.1)$$

Other authors propose a specific math model for the robotic arm and hand. So Bae et al. [3] proposed an only kinematic chain for a robotic arm of three DOFs and two fingers of two DOFs for controlling the object grasping in a plane. In this case, we propose a simple proportional regulator.

Gunji et al. [13] propose a control position complementary to a force control for controlling the grasping and sliding that can appear when handling an object. The hand is underactuated by three AC connected servomotors, Harmonic-Drive, and it has three fingers, each one with three DOFs. Figure 9.5 presents the control scheme proposed for each finger. There is a PD regulator controlling the position of each finger. The reference signal that should be followed by the motors is a combination between the signal consigned by the user and the force sensor reading in the extreme of the phalanges of the hand's fingers.



**Fig. 9.5** Classical control for Gunji underactuated hand



**Fig. 9.6** Pneumatic control of *shadow hand*. ©Shadow Robot Company 2008

### 9.3.2 Pneumatic Actuators Control

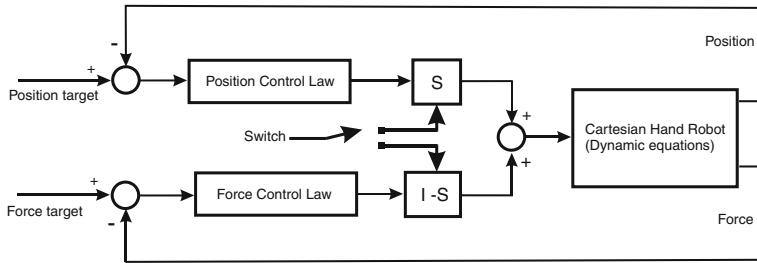
Although the majority of the hands developed use electric motors, some research teams have preferred to use pneumatic actuators based on the McKibben muscles [14]. It is the case of the hand developed by Shadow Robot Company [32]. The control of the hand has a high nonlinear behavior. Röthling et al. [30] propose a mixed control of position and elasticity in each of the antagonist actuators in the hand. These two physic magnitudes keep a correlation in the dynamic behavior of the hand, where the control signal reaching the solenoids is the opening time of each one of them (Fig. 9.6).

$$\begin{pmatrix} t_{\text{flex}} \\ t_{\text{ext}} \end{pmatrix} = \begin{pmatrix} K_\theta & K_S \\ -K_\theta & K_S \end{pmatrix} \begin{pmatrix} \Delta\theta \\ \Delta S \end{pmatrix} \quad (9.2)$$

The control signals  $t_{\text{flex}}$  and  $t_{\text{ext}}$  are the time period when the valves of the flexor and extensor muscle keep open. This system presents a nonlinear area, so it is necessary to use a deadband of  $0.6^\circ$  for the position and 0.2 bar for the pressure.

### 9.3.3 Hybrid Position/Force Control

This method proposes a controller where the system can choose between two control laws: a position one, and a force one, depending on the situation in which the robotic hand stands. Usually, if the fingers of the hand are moving freely without holding any object, we will use a position controller; however, if the hand is holding an object or it is in contact with the environment, the robot system will use the force controller. This type of control was firstly proposed during the 1970s by Craig and Raibert [9]. Many robotic hands have a control law based on this type of control and present different variations depending on the particular characteristics of each one of them (Fig. 9.7).



**Fig. 9.7** Hybrid position/force control

We take this into account in order to choose a specific control signal or other different matrix switchers. Their sum gives the unit. An example of that matrix can be seen in the following Eq. (9.3)

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad S' = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad S + S' = I \quad (9.3)$$

The second version of the hand, developed by the University of Gifu, has implemented a Hybrid position/force control [18, 19]. The hand uses encoders for measuring the position of its 16 joints, and therefore closes the loop in the control of position. The hand has five force-sensor sensors of six axes for controlling the force.

Yoshikawa [40, 41] proposes a hybrid position/force control with the following control law, a derivative and proportional action for the control in position and an integral action for the control in force. Also, they use this control in one hand with fingers made of a soft and flexible tissue with good results (soft-fingered hand).

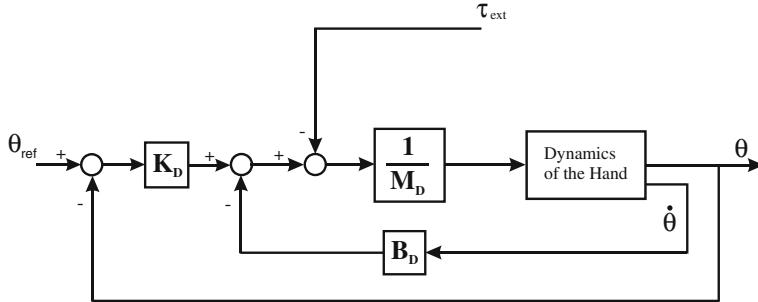
$$\begin{aligned} u_O &= \ddot{r}_{Od} - K_V(\dot{r}_O - \dot{r}_{Od}) - K_P(r_O - r_{Od}) \\ u_f &= \tilde{f}_d - K_I \int (f_d - \tilde{f}_d) dt \end{aligned} \quad (9.4)$$

where  $K_V$ ,  $K_P$ , and  $K_I$  are the gain matrix and  $r_{Od}$  and  $f_d$  are the desired trajectories of position and force. The signal  $\tilde{f}_d$  is the estimated or measured force.

Mouri et al. [25] propose a hybrid control for a robotic hand that offers massages on the human skin. The regulators used are: a classic position control and a force control through force impedance.

### 9.3.4 Impedance Control

This strategy allows controlling the relationship among the hand, the object, and the environment where the mechanic system moves around (impedance/admittance). The force executed by the hand's fingers over the handled object depends



**Fig. 9.8** Impedance control

on its position and its impedance. This expression in the Cartesian space, making reference to an electric simile, is  $f = Z(x_d - x_c)$ , where  $f$  is the force executed,  $x_d$  the desired position,  $x_c$  the real position, and  $Z$  the mechanic impedance of the hand containing the inertia, the viscous friction, and its stiffness constant.

The method is based on analyzing the error performed between the real and the desired position in a fingertip; the data can be put in contrast with the contact force through an impedance/admittance with adjustable parameters. This means, the measure of the contact force performed by the robot's extremity is indirectly measured. Therefore, we have

$$\tau_e = M_D \ddot{r}_0 + D_D (\dot{r}_0 - \dot{r}_{0r}) + K_D (r_0 - r_{0r}) \quad (9.5)$$

where  $M_D$ ,  $D_D$ , y  $K_D$ , are the desired matrix of inertia, damping coefficient, and stiffness, respectively, all of them defined symmetric and positive.  $r_{0r}$  is the reference position, the one to be reached by the claws' extremes or the fingers when they are not grasping any object, and  $r_0$  is the position of the claws or fingers when in touch with an object.

The specifications in the design of the matrix  $M_D$ ,  $D_D$ , y  $K_D$  follow the next criteria; typically, it is specified by the high values on  $K_D$  in those directions where being accurate is a requirement, and low values when values in the contact force against the object are required. Great  $D_D$  values are used if a high dissipation of energy is needed. Last, the  $M_D$  values allow a soft behavior of the manipulator when facing the contact force. The result force needed for reaching the impedance property required appears in Eq. (9.6)

$$\tau_0 = (M_0 - M_D) \ddot{r}_0 + D_D (\dot{r}_0 - \dot{r}_{0r}) + K_D (r_0 - r_{0r}) + h_0 + g_0 \quad (9.6)$$

The control algorithm can be seen in Fig. 9.8.

This control algorithm has been implemented in the DLR Hand, developed by DLR, German Aerospace Research, [5, 6, 22] where the force control of the hand's fingers is implemented through a impedance control in the Cartesian space of the robot. The second development of this hand was collaborating with the Harbin Institute of Technology (HIT), emerging into the (DLR-HIT-HAND). In this hand,

it has been also implemented an impedance control in a five finger hand, taking into account the flexibility in the joints [8]. Based on this technology, the same team has developed the Schunk Anthropomorphic Hand, [23, 38]. Another well-known hand is the Barrett hand, [35]. Both hands can provide position control and joint torque based impedance control for their fingers.

Others researches perform an impedance control taking only into account the stiffness of the impedance in the mechanic system, as in the case of the NASA Robonaut-2 hand, [1]. Their stiffness control commands a torque proportional to the joint error. Given a desired vector of joint values,  $r_0$ , the vector of desired joint torques follows as Eq. (9.7), where  $K_D$  is the diagonal stiffness matrix and  $r_{0r}$  is the sensed joint positions

$$\tau_e = K_D(r_0 - r_{0r}) \quad (9.7)$$

Comparing the two controls, the hybrid one has a better behavior than the impedance one, regarding the accuracy in the force and position. However, the hybrid control needs to change the control mode between position and force whenever the state of the contact of the hand with the environment, or the object, changes from contact mode to noncontact. That means, the hand moves from moving each finger freely to grab an object and move it, which is when the contact forces appear. A small delay in the change from position mode to force mode can provoke a great overshoot in the answer when in position mode.

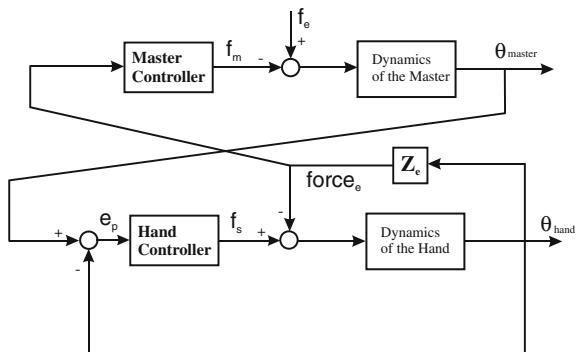
The impedance control does not have this problem, and its stability margin is bigger. Also, the impedance control has the advantage that the stiffness and the damping appear clearer in the system and can be calculated easily for some specific tasks performed by the user. On the other hand, in order to control the position and the force of the object and the environment, it is necessary to specify the impedance of the desired equilibrium position carefully.

These two models require a great computational load if we want to implement them rigorously. Both models need a detailed and precise knowledge about hand's and object dynamics. This is the reason why these two control methods have not been used fully. However, they have established the bases for developing more practical and simpler control algorithms.

## 9.4 Hand Control in Teleoperated Systems

The controls previously mentioned, as it has been already told, are difficult to be implemented, as well as, to introduce their control algorithms into a hardware which could be able to answer back fast enough to the needs of the system. Therefore, many of the current applications request the help of vision techniques that improve the behavior of the hands analyzed here. We will provide more information in the following chapter. Others authors request the help of the teleoperation for controlling the hand grasp. In this case, the user makes use of a model, replica of the hand

**Fig. 9.9** Force-position bilateral control scheme



grasping the object, and he implements the classic techniques used in teleoperated robots. This is the case in works by Kubo et al. [21].

The scheme of the force-position control, Fig. 9.9, is a modern version of the position–position control used in teleoperated robots [11]. In that figure, we can appreciate two control systems linked: in the low part, there is the slave (robotic hand that handle the object) and in the upper part, the master, which is the replica used by the user. The user receives in the master the information about the force executed by the robotic hand over the handled object. At the same time, the master sends the position reference to each one of the actuators that move the fingers in the robotic hand. The system has two regulators: the master usually has proportional regulator, and the slave has proportional-derivative one, because it improves the system’s answer getting better the overshoot produced by the contact with the object to be handled by the robotic hand. The integral action is not used because it can cause the instability of the system. We should remember that the user is the one closing the loop, because he has visual information about the movements performed by the fingers.

Kubo et al. [21] propose a grasping control with this scheme in which the master’s DOF number is smaller than the slave’s one. In order to achieve this, they propose the use of an observer of the disturbances presented in the robotic hand and the decoupling of the information received about the position and force using Discrete Fourier Transform. The DLR-HIT-Hand uses a bilateral control of the force-position. They have developed an exoskeleton as the master in order to control the movements in the robotic hand that achieve a bilateral force-position control [23, 15].

## 9.5 Application in the UC3M

As it has been previously mentioned, the RoboticsLab has developed the ASIBOT robot [12, 16], Portable Robot for Disabled Assistance. For this robot, the team developed an underactuated hand, RL1 [7]. This robot works with disabled people in domestic environment, and we thought it is convenient that the robot had a hand

similar to a human hand, that was able to handle the same elements used by a human being in a kitchen: dishes, spoons, glasses, and so on. That is the moment when emerges the possibility to design a new robotic hand to be perfectly able to be integrated to the arm and to have more functionality. However, it was required to be created under several parameters already established by the development. Those parameters were:

- *Integration*. The integration of the RL1 was supposed to be complete. This implied the RL1 to be created inside the arm but not damaging the functionality of it.
- *Weight*. The new design should diminish the inertia moment to the extreme of the arm and to improve the end load. The weight saved at the RL1 should be won in loading.
- *Size*. The size of the RL1 was restricted to the interior dimension of the anchor.
- *Functionality*. It was required to hold daily objects by an ordinary user in their personal environments and workplaces.
- *Driving system*. The driving system, as the control system, should be hosted inside the anchor, indirectly restricting the design in strength and number of actuators it should have.

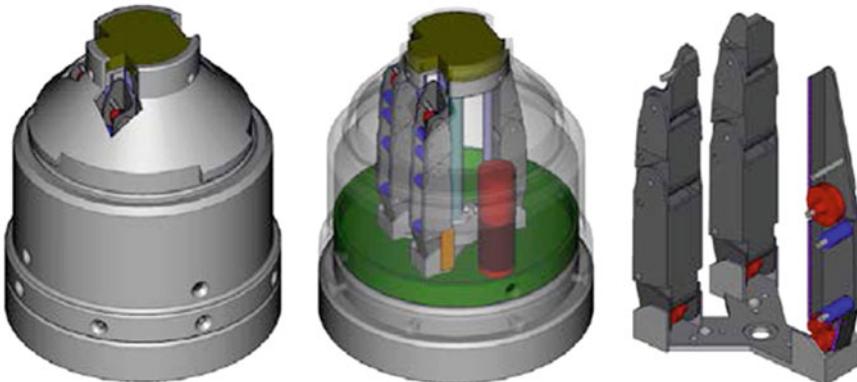
Once these pre-established requirements were defined, the general configuration of the RL1 hand was detailed as:

- Number of fingers
- Palm configuration
- Number of DOF and actuated DOF
- Kind of actuator
- Kind of Force Transmission
- Involved mechanisms.

Finally, it was concluded that the RL1 Hand should be formed by one thumb with two DOFs plus two fingers opposite to it, with three DOFs each. They would be fixed over a plane and static palm. Apart from the palm, all the RL1 Hand structure should enter and leave the ASIBOT anchor (80 mm) when needed by the user.

The final design of the RL1 Hand was divided into two subsystems: the finger design and the driving system design. The driving of all the joints of each finger is done by tendons and pulleys, and driven by a main pulley that is part of the driving system. In Fig. 9.10 is shown the location of the primary and secondary pulleys that make the pair to each finger phalange and also the tendon trajectories through it.

Most of the robotic hands that exist nowadays are mounted on the extremities of a robotic arm. It is said that they are available at any time as it is only necessary to move the hand to the necessary place to approach the tasks. They are in an operative state without the need of a previous movement through an actuator system. For the RL1 Hand, this situation is different as it is situated inside an anchor (Fig. 9.10). It requires not only moving the hand to the necessary location but also a previous step to put the hand in an operative state. For that reason, the



**Fig. 9.10** Location of the RL1 hand inside the docking mechanism of the ASIBOT robot

RL1 Hand should pass through an intermediate step that allows it to leave the anchor in order to be operative afterwards.

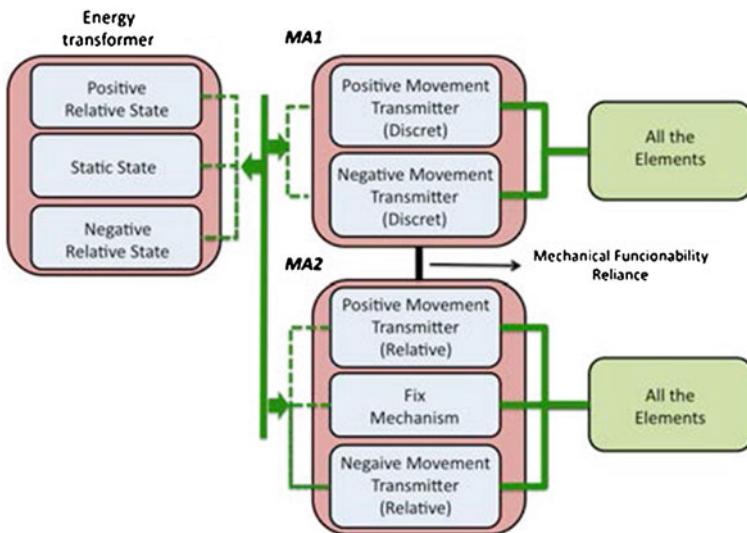
The actuator system should provide all the necessary states the RL1 Hand might need to achieve holding tasks and, apart from that, it should develop all the movements to pass from the different states: standby to operative. A specific design for the RL1 Hand of an actuator system is shown in Fig. 9.11.

The driving system of the RL1 Hand is formed by an energy transformer that provides three states to the two mechanisms named MA1 and MA2 (as showed in Fig. 9.11). Both mechanisms give movement to all the robotic system elements. Only one of them works at a time, thanks to a specific mechanism. This allows the energy transformer to send its movement selectively to each M<sub>A</sub>x depending on the state of the Hand.

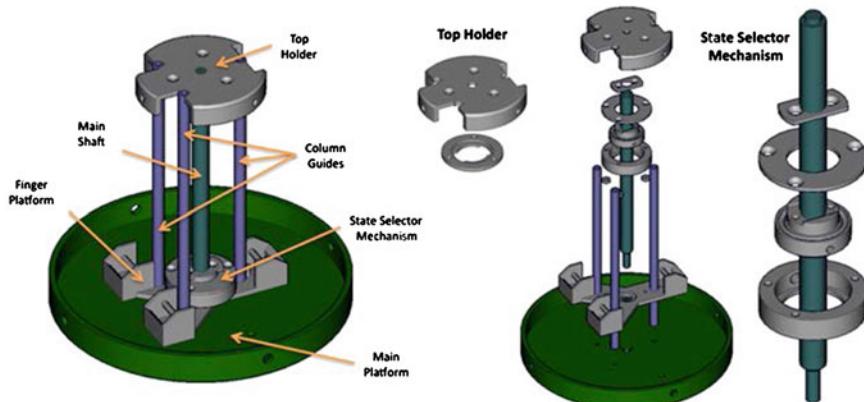
The extraction system is shown in Fig. 9.12. In it, it is possible to see, apart from all the elements that form it, a mobile platform named Main Platform, which moves in only one direction through the Guiding Columns. The central Main Axis that transmits the movement of the energy transformer and the state selector mechanism, which is attached to the Main Platform, makes this Platform to move up and down positioning the fingers outside and inside the anchor.

The states selector mechanism, thanks to a mechanical fusible hosted inside it, passes from a spindle which runs over a central Slotted Axis to a pulley that hosts the tendons of the three fingers and transmits the moment generated by the energy transformer to each joint, every time the main platform finds a limit at the Top Bracket.

The fingers and the thumb are hosted in a cradle. The main function of these cradles is to allow the fingers to remain in a parallel and straight position inside the anchor. As it is seen in Fig. 9.12, the cradle pulley helps the tendon to pass in an inverted way. For that reason, the joint that corresponds to the cradle rotates in a contrary way to the rest of the joints. Another objective of the cradle is to help the driving system of the RL1 to reach all the positions and tasks needed by rotating



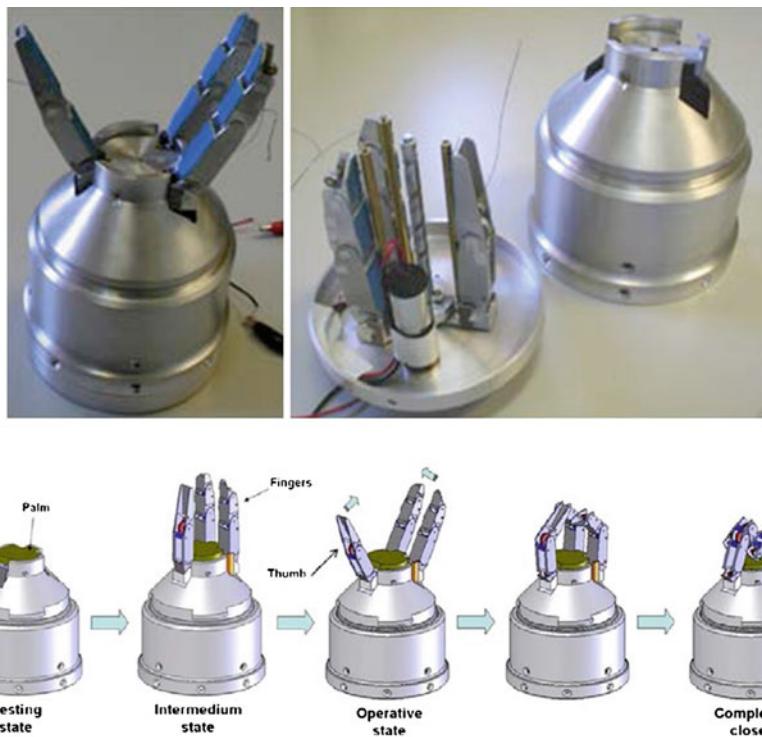
**Fig. 9.11** RL1 hand's driving system



**Fig. 9.12** Drive system integrated inside the RL1 hand's structure

the energy transformer in just one way. This allows the hand to pass from a standby state to an operative one, ready to hold an object through different states transforming mechanisms.

It is possible to see an already developed RL1 Hand at Fig. 9.13. This group of mechanisms makes the RL1 Hand driving system to be formed by a unique Multi-State Actuator that allows the Hand to pass through different positions, since the resting states inside the anchor until the fingers and thumb driving position, necessary to hold an object. It is also possible to see those different positions.



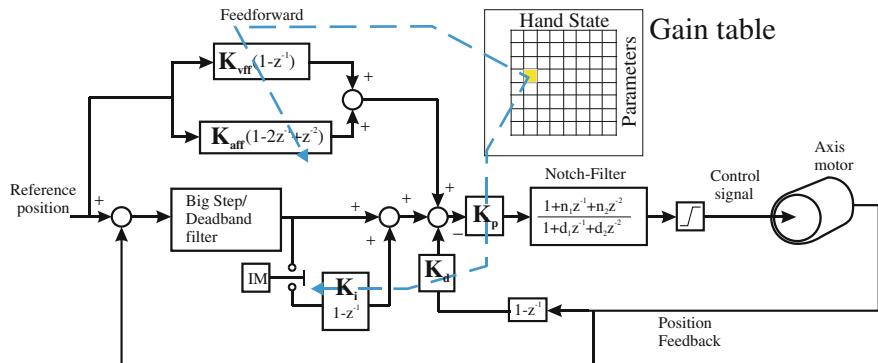
**Fig. 9.13** RL1 hand and its states

As we have seen before, the RL1 hand presents different states and the behavior of the transmission mechanisms, the actuator, and the charge change depending on their state. In each one, the dynamic behavior presented in the hand is well-known, and the load to be moved has always a similar inertia. The system has only position sensors for the actuator of the RL1 hand. For this reason, we have decided to use an adaptive control with gain planning [2].

The regulator has a proportional action ( $K_p$ ) in the direct chain, a derivative gain ( $K_d$ ) in the feedback signal of the speed, to make the exit signal softer and an integral action ( $K_i$ ) to reduce error in a stationary situation. Thanks to the three of them, it is possible to improve the dynamic conditions in the motors' movement and they are able to reduce the error in a permanent regime.

The integral gain can be activated or deactivated in order to avoid the wind-up effect which can provoke saturation in the control signal. Finally, a notch filter is in charge of eliminating the system's resonance frequency, without changing the gain of the control signal. The deadband filter takes into account the nonlinear effects produced by the saturation of the electronic equipment and a deadzone, similar to the ones provoked by the gaps present in the reducers.

There are two feedforward signals, one for compensating the dynamic delays than can cause high monitoring errors ( $K_{aff}$ ), and another one for compensating



**Fig. 9.14** The end-effector for the spherical universal harvesting robot (courtesy of the authors)

the static or sliding friction ( $K_{vff}$ ). Finally, the output control signal has a second-order filter (Notch Filter) in order to minimize the noise that can be present in the signal of some frequencies.

If the reference signals are pure steps, the robot's mechanic is subjugated to strong accelerations and dynamic efforts that could end damaging it. Therefore, it is important to use a profile that can reduce the abrupt changes in the acceleration. The combination of the regulator's function and the reference signal when defining the speed profile will make possible to improve the dynamic behavior of each joint (Fig. 9.14).

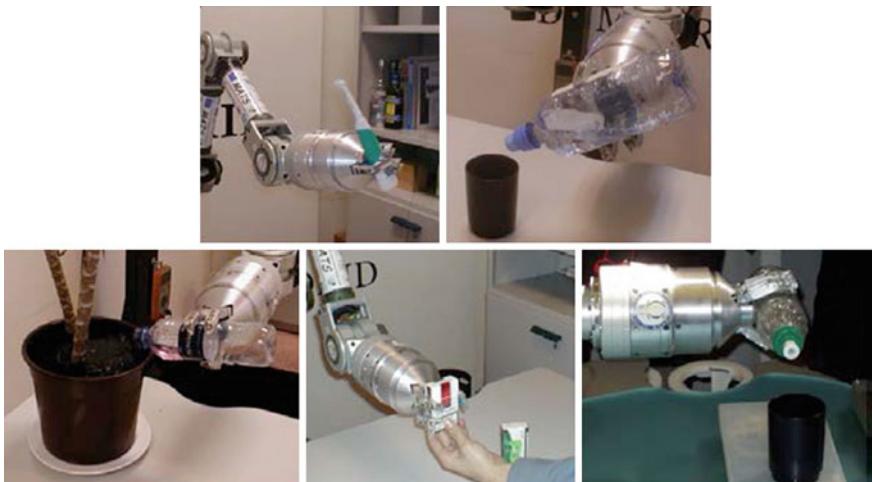
The control algorithm is not too complex, and the number of tasks to be carried out is minimal because there is only a motor. This algorithm is implemented in the onboard computer of the ASIBOT robot. In Fig. 9.15, we can see the different grasping movements of the RL1 hand when holding different domestic objects with different shapes. The results produced by the control algorithm are very suitable.

The other application example is the design of the underactuated hand for TEO humanoid. This is the second version of a full-sized humanoid robot totally designed and manufactured at UC3M by the researchers of RoboticsLab.

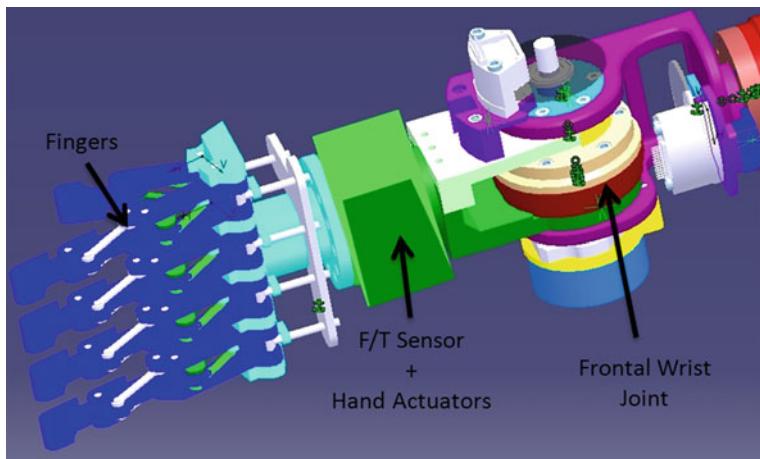
The design of the anthropomorphic hand for TEO robot has been influenced for the idea of maintaining the same concept that in the other parts of the robot: to emulate as much as possible the human form factor. Nevertheless, the premises of ease of use and cost-effective design has been important during its development. Due to this, TEO hand has been designed with five fingers with three phalanges which are actuated by only two motors.

The robotic hand is attached to the robot by means of a mechanical structure that integrates a force/torque sensor. The hand actuators are integrated within this structure of union. This set of systems constitutes the wrist of the robot (see Fig. 9.16).

As introduced before, TEO robot hand is underactuated to reduce the volume of sensors and actuators needed to integrate on board and also this will reduce



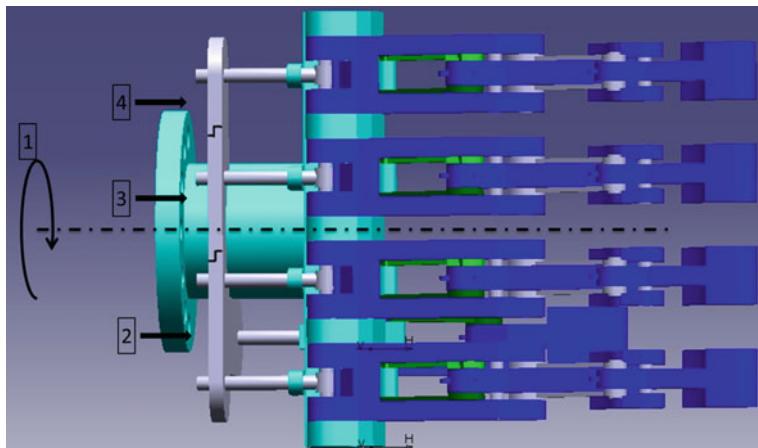
**Fig. 9.15** RL1 hand with different types of grasp tasks



**Fig. 9.16** TEO's hand coupled with the force/torque sensor at wrist

fabrication costs. The movement to perform a gripping action is produced by means of two motors. The first one is a rotary motor situated in the wrist (1) and is used to locate a linear actuator in the appropriate position. This second actuator pushes down the mechanism connected to the moving finger. In Fig. 9.17, it can be observed the positions to move thumb, the forefinger and the middle finger (2); the ring finger (3) and the little finger.

TEO hand has been conceived to grip mainly cylindrical and spherical object. This issue is imposed by its nature of underactuated hand. Each phalanx is moved



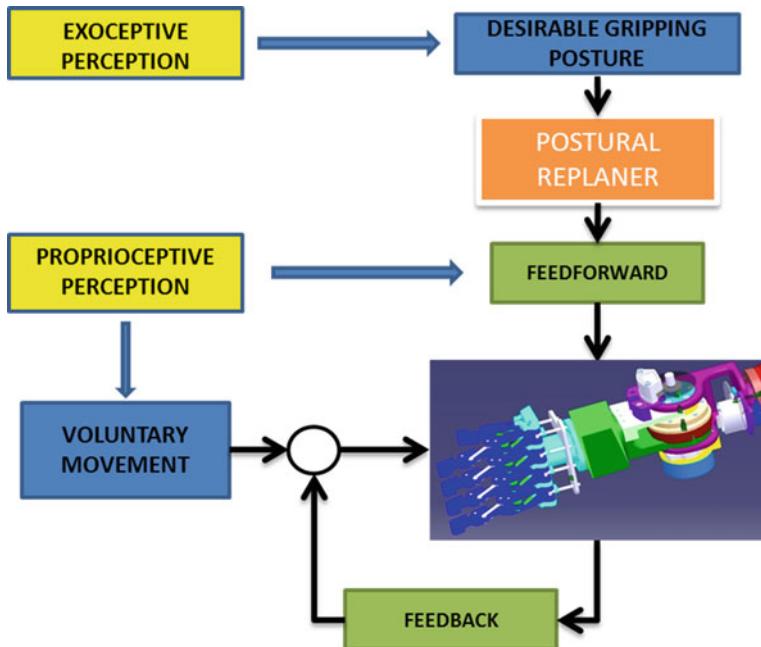
**Fig. 9.17** TEO's hand actuator positions to connect the drive motor

by a double mechanism. The clenching movement is produced by a connecting rod integrated in each phalanx. When the mechanism to move fingers is pushed down the finger moves to the target position and the mechanical system keep locked on when the object is gripped. To extend the fingers, it is necessary to unlock the retaining system with the help of the linear actuator. Once the finger is unlocked, the finger is extended with the help of elastic tendons.

TEO hand movement will be planned with the help of three different systems, some of them still under development. The first one is a vision system to plan the arm and hand movement to the target position where the object is located. The second one is the force/torque sensor used to measure the payload during the gripping action. The last one will be tactile sensors attached to the phalanx to measure the grip force. The grip control system is integrated within a feedforward postural control system inspired in human processes.

At low level, the controller of the hand will work in an hybrid Position/Force scheme close similar to the previously presented for ASIBOT's RL1 Hand, (Fig. 9.17). The local controller will receive desired position and force references from central host computer from the CANOpen network. Those references will be calculated remotely, by the central computer, taking into account the inputs at the wrist sensor (Fig. 9.18).

The proprioceptive perception corresponds to the state of the robotic hand and the exoceptive perception is the information obtained from the Force/Torque and the tactile sensors. Both perceptions are fed into a feedforward controller to anticipate the gripping posture to prevent crashes, object fallings, and so on.



**Fig. 9.18** TEO's hand control is integrated in the full body control schema

## 9.6 Conclusions

In this chapter, we have classified the different grasping elements into three groups: grippers, underactuated hands, and multi-actuated hands. We have talked about how each one of their design affect from a control point of view. Later on, we have analyzed the mechanic systems that are present in them, how they affect them from a control point of view: emplacement of the actuators, elements for transmitting force and movement, appearance of undesired lineal effects, and so on.

Following that, we have presented the different control strategies more frequently used in robotic hand that we have been able to find in writings. As it can be seen, the most frequently used are those controlling the information about the force produced during the grasping, as in the case of the hybrid and impedance controls. The great majority of the authors use these two types of strategies including different regulators. It is also clear that there are few researchers who make use of neuronal nets or fuzzy control.

The forenamed control, sometimes, is not enough for obtaining a proper performance of the system. For that reason, a lot of the current applications tend to complement themselves with servo control visual techniques as it will explain in the following chapters.

Finally, some solutions will be detailed through mechanical systems and how they can affect the development of new robotic hands, studying the case of

underactuated architectures developed at Roboticslab, which are implemented in one robot designed at Robotics Lab: ASIBOT (personal assistive robot), and TEO, the UC3M's second version of a full-sized humanoid.

**Acknowledgments** This work has been funded by the following projects: Robotic Cognitive Assistant for Persons with Special Needs (ARCADIA), DPI2011-00022/001, and Design and development of variable impedance actuators for Service Robots (DAVARBOT), DPI2011-22513, supported by the Spanish Ministry of Science and Innovation. Also thanks to the Madrid's Regional Government to support this work under the project: COMANDER (2011/00095/001).

## References

1. Abdallah M, Platt R, Wampler CW, Hargrave B (2010) Applied joint-space torque and stiffness control of tendon-driven fingers. In: IEEE-RAS international conference on humanoid robots, pp 74–79
2. Astrom A, Hägglund T (2006) Advanced PID control. ISA, New York
3. Bae JH, Arimoto S, Ozawa R, Sekimoto M, Yoshida M (2006) Proceedings of the 2006 IEEE international conference on robotics and automation, pp 2131–2136
4. Bicchi A (2000) Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity. *IEEE Trans Robot Autom* 16(6):652–662
5. Borst Ch, Fischer M, Haidacher S, Liu H, Hirzinger G (2003) DLR Hand 11: experiments and experiences with an anthropomorphic hand. In: Proceedings of the 2003 IEEE international conference on robotics and automation, pp 702–707
6. Butterfass J, Grebenstein M, Liu H, Hirzinger G (2001) DLR-Hand II: next generation of a dexterous robot hand. In: Proceedings of the 2001 IEEE international conference on robotics and automation, pp 109–114
7. Cabás R, Balaguer C (2005) Design and development of a light weight embodied robotic hand activated with only one actuator. In: IEEE international conference on intelligent robots and systems
8. Chen C, Lii NY, Minghe J, Shaowei F, Hong L (2010) Cartesian impedance control on five-finger dexterous robot hand DLR-HITII with flexible joint. In: Proceedings third international conference on intelligent robotics and applications (ICIRA 2010). Springer, Berlin, Germany Part pt.1, pp 1–12
9. Craig J, Raibert M (1979) A systematic method for hybrid position/force control of a manipulator. In: IEEE computer software applications conference
10. Eppingen SD, Seering WP (1992) Three dynamic problems in robot force control. *IEEE Trans Robot Autom* 8(6):751–758
11. Ferre M, Barrio J, Melchiori C, Bogado JM, Castedo PL, Ibarra JM (2007) Experimental results on bilateral control using and industrial telemanipulator. In: Ferre M et al (eds) Advances in telerobotics, STAR vol 31. Springer, Berlin pp 177–190
12. Giménez A, Jardón A, Correal R, Cabas R, Balaguer C (2004) Service robot applications for elderly people care in home environments. International workshop on advances in service robotics
13. Gunji D, Mizoguchi Y, Teshigwara S, Ming A, Namiki A, Ishikawaand M, Shimiko M (2008) Grasping force control of multi-fingered robot hand based on slip detection using tactile sensor. In: Proceedings, IEEE/RSJ international conference on intelligent robots and systems, pp 2605–2610
14. Hildebrandt A, Sawodny O, Neumann R, Hartman A (2002) A flatness based design for tracking control of pneumatic muscle actuators. In: Proceedings of ICARV, vol 3, pp 1156–1161

15. Hongen F, Fang H, Xie Z, Liu H (2009) An exoskeleton master hand for controlling DLR/HIT hand. In: Proceedings of the 2009 IEEE/RSJ international conference on intelligent robots and systems, pp 3703–3708
16. Jardon A, Gimenez, A, Correal R, Cabas R, Martínez S, Balaguer C (2006) A portable light-weight climbing robot for personal assistance applications. *Ind Robot: Int J* 33(4):303–307
17. Kawasaki H, Komatsu T, Uchiyama K, Kurimoto T (1999) Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II. In: Proceedings of the IEEE international conference on systems, man, and cybernetics (IEEE SMC'99), pp 782–787
18. Kawasaki H, Shimomura H, Shimizu Y (2001) Educational-industrial complex development of an anthropomorphic robot hand. *Adv Robot* 15(3):357–63
19. Kawasaki H, Komatsu T, Uchiyama K (2002) Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II. *IEEE/ASME Trans Mechatron* 7(3):296–303
20. Khatib O (1987) A unified approach for motion and force control of robot manipulators: the operational space formulation. *IEEE J Robot Autom* 3(1):43–53
21. Kubo R, Shimono T, Ohnishi K (2009) Flexible controller design of bilateral grasping systems based on a multilateral control scheme. *IEEE Trans Ind Electron* 56(1):62–68
22. Liu H, Hirzinger G (1999) Joint torque based Cartesian impedance control for the DLR hand. In: Proceedings of the IEEE/ASME international conference on advanced intelligent mechatronics, pp 695–700
23. Liu H, Meusel P, Hirzinger G, Jin M, Liu Y, Xie Z (2008) The modular multisensory DLR-HIT-hand: hardware and software architecture. *IEEE/ASME Trans Mechatron* 13(4): 461–469
24. Melchiori C, Kaneko H (2008) Robots hands. Chapter 15 of handbook of robotics. Springer, West Lafayette, pp 345–360
25. Mourik K, Terashima K, Minyong P, Kitagawa H, Miyoshi T (2007) Identification and hybrid impedance control of human skin muscle by multi-fingered robot hand. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, pp 2895–2900
26. Murray RM, Li Z, Sastry SS (1994) A mathematical introduction to robotic manipulation. CRC Press, Boca Raton
27. Narasimhan S, Siegel D, Hollerbach J, Biggers K, Gerpheide G (1986) Implementation of control methodologies on the computational architecture for the Utah/MIT hand. In: Proceedings of the IEEE international conference on robotics and automation, pp 1884–1889
28. Okada T (1979) Object-handling system for manual industry. *IEEE Trans Syst Man Cybern* 9(2):79–89
29. Puente EA, Balaguer C, Barrientos A (1991) Force-torque sensor-based strategy for precise assembly using a SCARA robot. *Robot Auton Syst* 8(8):203–212
30. Röthling F, Haschke R, Steil J, Ritter H (2007) Platform portable anthropomorphic grasping with the bielefeld 20-DOF shadow and 9-DOF TUM hand. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems
31. Starr GP (1988) Cartesian stiffness control of the JPL/Stanford/Salisbury hand robotics and automation. In: Proceedings of 1988 IEEE international conference on robotics and automation
32. Shadow Robot Company (2003) Design of a dexterous hand for advanced CLAWAR applications
33. Speeter TH (1990) Modal state position controller for the UtahMIT dexterous hand. *IEEE Trans Robot Autom* 6(2):274–277
34. Townsend WT, Salisbury JK (1989) Mechanical bandwidth as a guideline to high-performance manipulator design. In: Proceedings of the 1989 IEEE international conference on robotics and automation
35. Townsend W (2000) The BarretHand grasper—programmable flexible part handling and assembly. *Ind Robot: Int J* 27(3):181–188
36. Villani L, Schutter J (2008) Force control. Chapter 7 of handbook of robotics. Springer, West Lafayette, pp. 161–185

37. Whitney DE (1977) Force feedback control of manipulator fine motions. *ASME J Dyn Syst Meas Control* 99:91–97
38. Xue Z, Zoellner M, Dillmann R (2008) Automatic optimal grasp planning based on found contact points. In: Proceedings of the 2008 IEEE international conference on advanced intelligent mechatronics
39. Yao S, Zhan Q, Ceccarelli M, Carbone G, Lu Z (2009) Proceedings of the 2009 IEEE international conference on mechatronics and automation
40. Yoshikawa T, Sugie T, Tanaka M (1988) Dynamic hybrid position/force control of robot manipulators—controller design and experiment. *IEEE J Robot Autom* 4(6):699–705
41. Yoshikawa T (2010) Multifingered robot hands: control for grasping and manipulation. *Ann Rev Control* 34:199–208
42. Zollo L, Roccella S, Guglielmelli E, Carroza MC, Dario P (2007) Biomechatronic design and control of an anthropomorphic artificial hand for prosthetic and robotic applications. *IEEE/ASME Trans Mechatron* 12(4):418–429

# **Chapter 10**

## **Hardware Control System of Robotic Hands**

**Zhan Qiang**

**Abstract** Control system is a very important component of robotic hands, of which the functions and applications are dominated by it, although very few papers discussed it. A control system can be divided into hardware system and software system and both of them combine to realize the control functions of a robotic hand. With the development of microelectronics technology during the past 30 years, the hardware control system of robotic hands has gained significant progress, which greatly promotes the development of robotic hands. This chapter will first give a brief review of the development of the hardware control system of robotic hands from 1980s to present, and then four different hardware control systems of robotic hands designed by our lab are introduced with detailed descriptions so as to provide more information on the design of hardware control system of robotic hands.

### **10.1 Development Review of the Hardware Control System of Robotic Hands**

For a robotic hand, the hardware control system is an entity composed of electric and electronic devices, such as microprocessors, drive circuits, and communication chips, with which the robotic hand can realize expected motions and functions. Of course, without software the hardware can do nothing, so the related software is an indispensable partner of the hardware for the control of any robotic hand.

---

Z. Qiang (✉)

Laboratory of Complex Mechanism and Intelligent control (CMIC), Robotics Institute,  
Beijing University of Aeronautics and Astronautics, No.37, Xueyuan Road, Haidian,  
Beijing, China  
e-mail: qzhan@buaa.edu.cn

Compared with other research aspects of robotic hands, such as mechanism design and control methods, hardware control system of robotic hands is less touched in papers, because it usually represents “easy” job. Whenever it appears in a paper, it seems so insignificant that it usually covers a very small patch. However, contrast to the phenomenon, hardware control system is so important that it dominates the functions and applications of robotic hands. For example, during 1980s the hardware control system of robotic hands was very big, slow, and expensive and only few institutes could own it and do experiments on robotic hands, whereas during these 10 years the hardware control system of robotic hands becomes so small, fast, and cheap that many institutes can do experiments on self-designed robotic hands and some prosthetic robotic hands like “i-Limb” can help those hand amputees [9].

In this section, we will look back on the development of the hardware control system of robotic hands from 1980s to present. In order to make the introduction clear, the period of more than 30 years is artificially divided into two stages: from 1980s to 1990s and from 2000s to present. Relatively speaking, the development of the hardware control system of robotic hands in the first stage is much slower than that in the second stage.

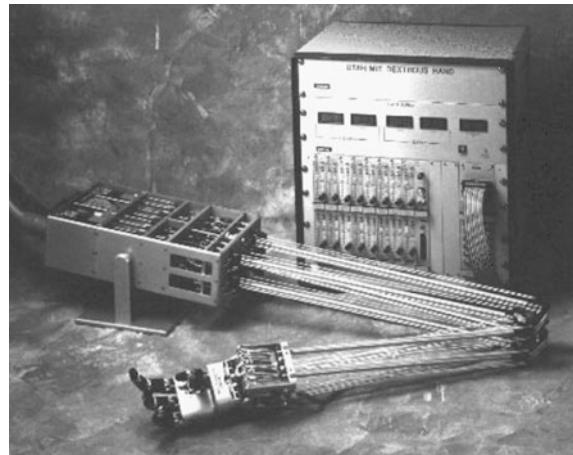
### ***10.1.1 Hardware Control Systems of Robotic Hands During 1980s–1990s***

Because the hardware control system of robotic hands is an assembly of electronic and electric devices and components, its development is also a reflection of the development of electronics and electrics technology, especially the electronics technology. Encouraged by science fiction presentations depicting high performance robots and the promising markets of anthropomorphic robots, several robotic hands to mimic the human hand’s structure and transmission mechanism were designed in 1980s, such as Utah-MIT hand [14], Stanford/JPL hand [18], Mark-I hand [19], among which the Utah-MIT hand is such a typical one that much research based on it was conducted by many different institutes.

#### **10.1.1.1 Hardware Control System of Utah-MIT hand**

Utah-MIT hand has 4 fingers, 16 joints actuated by two antagonistic tendons each, 32 tendon tension sensors, and 16 joints position encoders. Electropneumatic actuators are selected to realize the motions of all the joints through tendons. At that time, the development of those electropneumatic actuators was such a complex task that it costs about 3 years [14, 15]. Because of the big volume of those electropneumatic actuators, they were not embedded in the hand body but packaged as a long rectangular box. Figure 10.1 shows an Utah-MIT hand with its big actuator pack (the one with cables connecting with the hand) and control box.

**Fig. 10.1** Utah-MIT hand and its control box (Courtesy to the University of Utah)

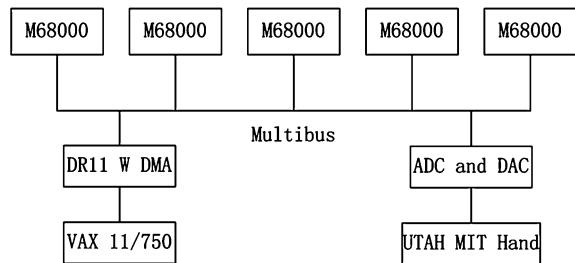


In order to meet the requirement of 400 Hz servo rate, the control system needs to read 19,200 sensor values and send out 12,800 actuator values per second simultaneously. Due to no single processor could be adequate for the control task at that time, a multiprocessor architecture was adopted from both cost and programmability considerations. The hardware control system of Utah-MIT hand includes a host and a multiprocessor real-time controller. During 1980s, two different hardware control systems of Utah-MIT hand adopting two different control architectures, Muse and CONDOR, were implemented. The block diagram of the hardware control system of Muse is shown in Fig. 10.2 [22], of which the host is a Vax 11/750 and the real-time controller is made up of five Motorola 68000 (1 MIPS, no floating-point unit) based single-board computers linked by an Intel Multibus. The connection between the Vax and the real-time controller is a DMA link.

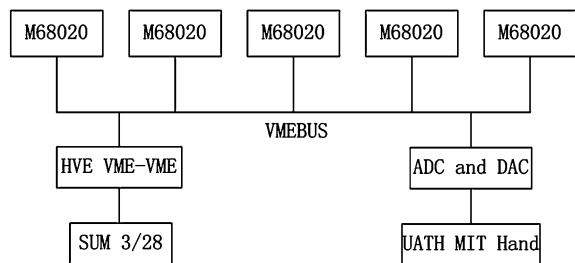
The block diagram of the hardware control system of CONDOR is shown in Fig. 10.3 [22]. The host is a Sun-3 system and the real-time controller utilizes five Motorola 68020 (2.5 MIPS) based single-board computers with Motorola 6881 floating-point unit (160 KFLOPS). The host and the controller are connected by VME bus. The implementation of the CONDOR system is behind that of the Muse system about 2 years, but because of the much faster processors coupled with floating-point engine the CONDOR control system provides a much faster environment for the Utah-MIT hand. Furthermore, the Sun-3 host provides a much better user environment installed with industry standard window system [21, 22].

Due to the big volume, the actuators and the hardware control system of Utah-MIT hand were put outside of the hand body, so the space inside the hand body can be made full use of to place transmission mechanism and sensors. Of course, most contemporary people would feel that the CONDOR control system of the Utah-MIT hand is too big and cumbersome, although it was such an excellent control system and almost the best choice for the hand at that time.

**Fig. 10.2** Hardware block diagram of Muse (From Narasimhan et al. [22])



**Fig. 10.3** Hardware block diagram of CONDOR (From Narasimhan et al. [22])

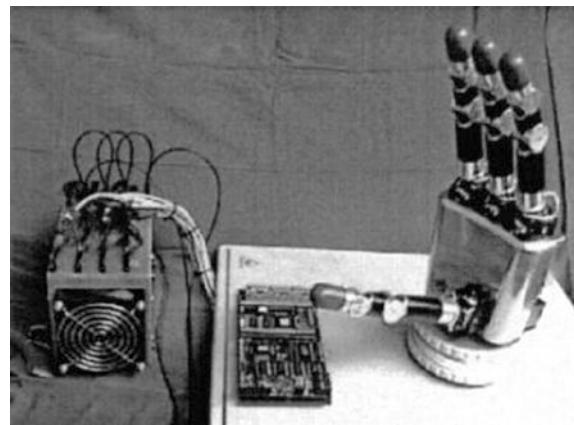


### 10.1.1.2 Hardware Control System of DLR Hand I

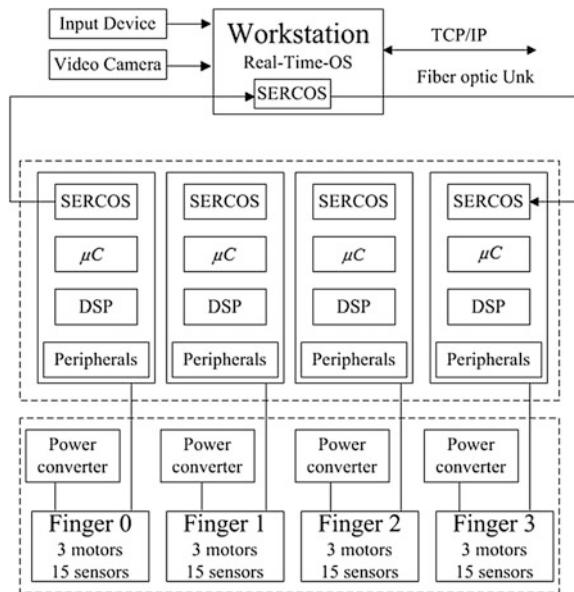
With the development of microelectronics technology, much smaller and much faster microprocessors appeared in 1990s, and the hardware control system of robotic hands also becomes much smaller and more powerful. In 1990s, the DLR Hand I of Institute of Robotics and Mechatronics of Germany, shown in Fig. 10.4 [1], is a very famous star in the robot field. The most significant difference between DLR Hand I and Utah-MIT hand is that the actuators of DLR Hand I were put inside the hand body rather the outside in Utah-MIT hand. DLR Hand I is a semianthropomorphic robotic hand that has four fingers with three DOF (Degree of Freedom) each. Each finger joint is actuated by a specially designed linear actuator that consists of a brushless DC motor and a planetary roller spindle drive. Each finger body of DLR Hand I is equipped with about 28 sensors [1].

The control architecture of DLR Hand I is shown in Fig. 10.5 [1]. A workstation running real-time operating system works as the host and four DSP control boards with TMS320C31 (30 MIPS, 60 MFLOPS) work as the lower controller to deal with the control of four fingers separately. The host workstation and the local finger controllers communicate via SERIAL Real time COmmunication System (SERCOS) by fiber optic link. SERCOS is powerful enough to exchange all sensor information and control signal within 1 ms for all the four fingers. The lower controller of four fingers is assembled as a small box ( $120 \times 110 \times 220$  mm) with a total mass of 1600 g, shown in Fig. 10.4. The lower controller needs to deal with the control of 12 brushless DC motors, the information capturing of 100 sensors (35 sensors per finger) [1].

**Fig. 10.4** DLR hand I and its control box (From Butterfass et al. [1])



**Fig. 10.5** Architecture of DLR hand I (From Butterfass et al. [1])



### 10.1.1.3 Comparisons of the Microprocessors of Utah-MIT hand and DLR Hand I

Table 10.1 gives some comparisons of the microprocessors used by the hardware controllers of Utah-MIT hand and DLR hand I [1, 20–22]. From Motorola 68000 to Motorola 68020 and TMS320C31, of which the processing capability becomes more and more powerful, so less microprocessors can be used in the hardware controller of those robotic hands but with much faster processing speed.

**Table 10.1** Comparison of microprocessors used in Utah-MIT hand and DLR hand I

Type	Speed (MIPS)	Float point speed (FLOPS)	Size (mm)
Motorola 68000	1	0	17.45 × 17.45 (64 pins)
Motorola 68020	2.5	160 K	34.16 × 34.16 (114 pins)
TI TMS320C31	30	60M	28.25 × 28.25 (132 pins)

### 10.1.2 Hardware Control Systems of Robotic Hands During 2000s–Present

After 2000, more powerful DSP, ARM, and FPGA chip processors have been widely applied to industry and media. Accordingly, the control hardware of robotic hands also changes a lot and a main trend or big difference from previous is that the hardware controller is gradually entering into the hand body (palm or finger) and becomes part of the hand, such as DLR Hand II shown in Fig. 10.6 [12].

#### 10.1.2.1 Hardware Control System of DLR Hand II

DLR Hand II has four identical fingers with four joints and three degrees of freedom each, and there is an additional degree of freedom in the palm to allow the hand to adjust for either stable grasping or fine manipulation. The high integration of the actuation system, the sensors, and the communication electronics in the hand, the precise control based on lots of sensors and the anthropomorphic dexterous grasping and manipulation make it be recognized as one of the best robotic hands in the world [17].

The typical characteristics of the control system of DLR Hand II are distributed, modular, complex and expensive. The last two characteristics come from its specially designed motors, lots of sensors (hall sensors, potentiometers, strain gauges, force/torque sensors, and temperature sensors), special sensor installation process, multilevel software, and the expensive versa module eurocard (VME) bus board [17]. Instead of the digital signal processors used in DLR Hand I, DLR Hand

**Fig. 10.6** DLR hand II  
(From Haidacher et al. [12])



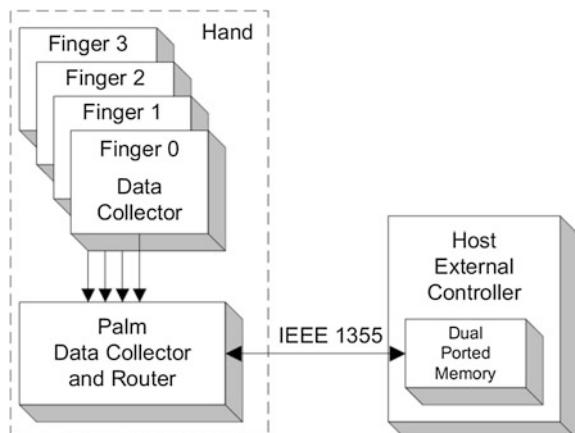
II uses two industrial PowerPC VME bus boards to process all the higher level data. Both boards run the commercial real-time operating system VxWorks and communicate with external computers by LAN connection and serial interface. FPGAs are used to build up a network to collect and transmit sensors data from each finger to a higher level digital signal processing controller such as Hand Data Collector. The use of FPGAs in the hand brings much flexibility of sensor data collecting and including new data source.

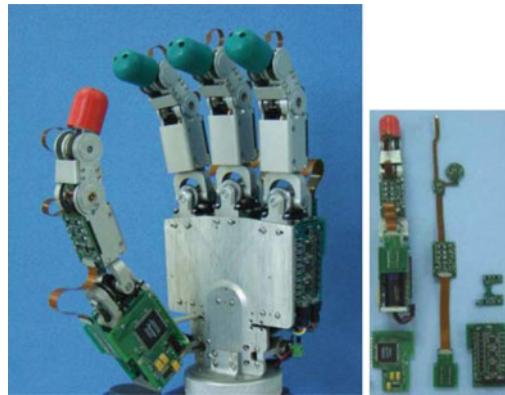
Generally, the hardware control system of DLR Hand II consists of a host external controller and four lower controllers embedded in each finger body. The lower controller is composed of several small hard printed circuit boards and several flexible printed circuit boards so as to pass through joints with one degree of freedom. The host external controller communicates with each finger controller via an integrated serial communication system (IEEE 1355) by embedding a communication controller in the base unit of each finger. The serial communication hardware of DLR hand II is shown in Fig. 10.7 [12]. The communication controller is responsible for the collecting and distributing of all information, processing some reasonable signal and distributing control data to motor actuators. The embedded controller design of DLR Hand II achieves such a big advancement on the cabling that reducing the external cables from 400 (DLR Hand I) to 12 [2, 3, 12].

#### 10.1.2.2 Hardware Control System of DLR-HIT Hand

DLR-HIT Hand, shown in Fig. 10.8 [17], was jointly developed by Harbin Institute of Technology (HIT) and DLR based on the experience of DLR Hand II. DLR-HIT Hand has similar structure with DLR Hand II: four identical fingers and total 13 DOFs with an extra DOF for the palm, but DLR-HIT Hand is smaller than DLR Hand II and commercially available devices and electronic components are as far as possible used in order to decrease its manufacturing cost.

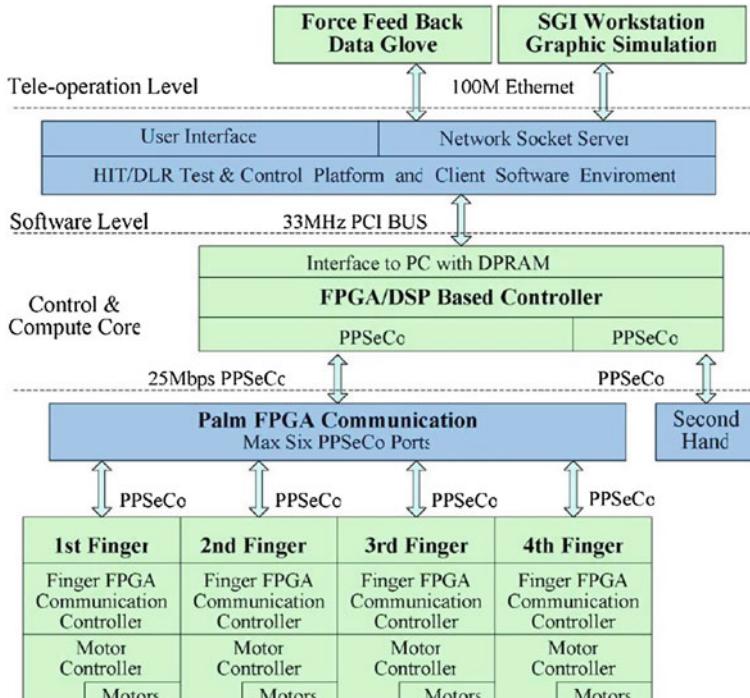
**Fig. 10.7** Communication hardware of DLR hand II  
(From Haidacher et al. [12])





**Fig. 10.8** DLR-HIT hand and one finger module (From Liu et al. [17])

As shown in Fig. 10.9 [17], the hardware control system of DLR-HIT hand includes a PC with a PCI-based DSP/FPGA board as a master, a palm FPGA board, and four-finger FPGA-BLDC boards as a slave. The master and the slave communicate through a self-designed point-to-point high speed serial



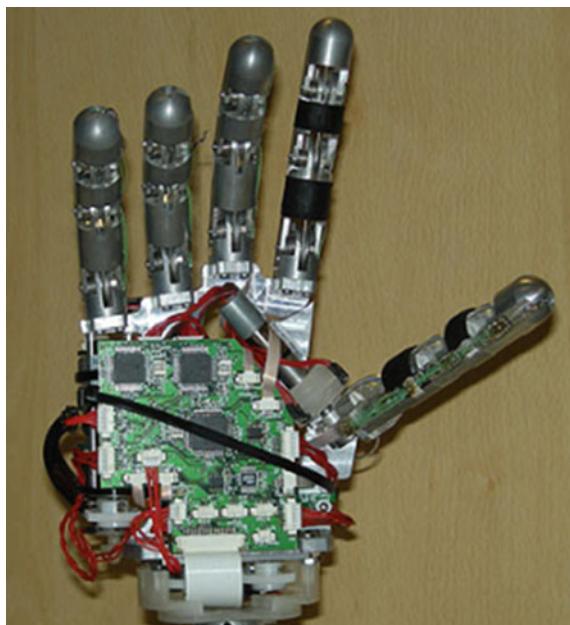
**Fig. 10.9** Control architecture of DLR-HIT hand (From Liu et al. [17])

communication (PPSeCo) system with 25 M baud rate. With a TI DSP TMS320C6713 (225 M clock and maximum 1350 M MFLOPS) the DSP/FPGA PCI board can realize complex control methods and multichannel communication with the slave controllers or sensors. Like a router the palm FPGA board performs data transmission between the finger FPGA and the DSP/FPGA PCI board. Each finger has a module (shown in Fig. 10.8, right) that integrates the fingers FPGA, the BLDC motor control board and the sensor board together. The finger FPGA board is used to convert sensor data and communicate with the palm FPGA. The BLDC motor control board is used to drive three motors in the finger and its small size ( $35 \times 65$  mm) makes it easy to be embedded in the finger body. Flexible printed circuit board (PCB) is designed to reduce the connectors for sensors and pass through joints of each finger as well as link sensors and FPGA. The mechatronics design philosophy of the DLR-HIT Hand effectively minimizes the hand dimension and cable numbers (only five cables including power supply) [17, 23].

#### 10.1.2.3 Hardware Control System of SmartHand

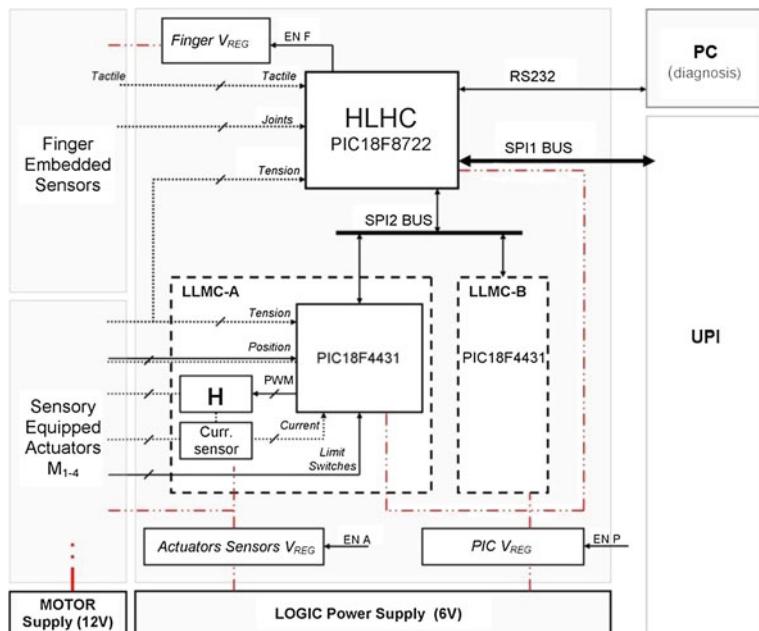
The prosthetic SmartHand, shown in Fig. 10.10 [5], symbolizes a new trend that robotic hands can be used to help those disabled, so its control system should also consider the biointeraction with the user. With the similar size of a human hand SmartHand has five fingers actuated by four motors and 40 embedded sensors for automatic control and biofeedback [5].

**Fig. 10.10** SmartHand  
(From Cipriani et al. [5])



Shown in Fig. 10.11 [8], the control architecture of SmartHand has a high-level hand controller HLHC based on the Microchip microcontroller PIC 18F8722 and two low-level motor controllers LLMC based on PIC18F4431. The high-level controller communicates with two low-level controllers through a fast SPI bus and communicates with external PC or user-prostheses interfaces (UPI) through RS232 or a fast SPI bus. The high-level controller is mainly used to gather sensors information, manage power and hand's status. The two low-level controllers are mainly used to realize sensing and control. The hardware control system of SmartHand is flexible enough to support the real-time control of the four active axes, real-time identification of external commands, computing of control loops and delivering of sensory biofeedback. Different from other control systems of robotic hands, the control algorithms of SmartHand can be programmed in the hand controller rather than a PC. In other words, for SmartHand a PC is only an interface for a user to access the hand not a high-level controller [5–8].

At present, the hardware control system of robotic hands has become so small and fast that a prosthetic hand can be made with similar size of a human hand and the grasp control methods can be executed in real time, but compared with mechanism, hardware control system is still a bigger obstacle that prevents a robotic hand from being a more compact, light, and dexterous anthropomorphic hand. For most designers and researchers, the hardware control system of robotic hands is hoped to be even much smaller, even much lighter, and with even fewer



**Fig. 10.11** Control architecture of SmartHand (From Cipriani et al. [5])

cables so as to achieve more compact integration of control system and mechanism, if the hope can come true, the development of robotic hands will come to a new stage and more and more robotic hand products will enter into people's daily life.

## 10.2 Hardware Control Systems of Robotic Hands Designed by CMIC Lab

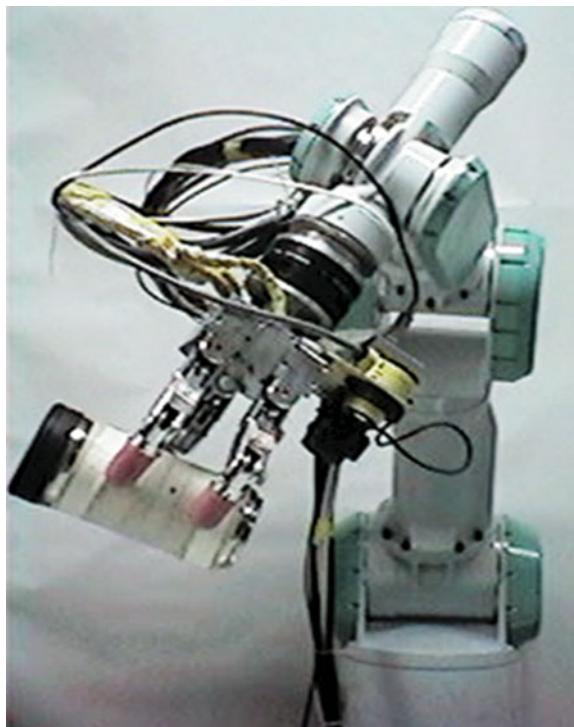
Our Robotics Institute started the research on robotic hands from 1980s [10, 11], during the past time about six different robotic hands were designed and we also experienced the development of microelectronics technology and the hardware control system of robotic hands. In this section, four hardware control systems of robotic hands designed by CMIC lab during the near 10 years will be introduced in detail so as to make up the incomplete description of the above hardware control systems and provide more detailed information on the design of hardware control system of robotic hands.

When one wants to design a hardware control system for a robotic hand, such following simple procedures can be referred to. First, according to the control requirements of a robotic hand, the control architecture should be designed. Then, the number and kind of actuators, sensors, external communication ports, etc., should be determined. Last, the real-time control board should be designed by choosing appropriate DSP, ARM, or FGPA chips as well as other peripherals.

### 10.2.1 A DSP Based Hardware Control System of BH-3 Hand

BH-3, shown in Fig. 10.12, is a nine DOF, three-fingered robotic hand first designed by Prof. Zhang Qixian. Its nine DC motors are embedded in the palm of the hand and steel string is used to transmit the motion from each motor to each finger joint. At the end of each finger, there is a six DOF force/torque sensor [24]. The hardware control system of BH-3 hand is composed of a personal computer (the host) and a single TI DSP control board (the real-time controller) and RS-232 is used to realize communication between them. The host computer has a 450 MHz Intel processor and a Borland C++ programming environment where the control methods for the robotic hand are programmed and executed. The real-time controller based on a TMS320F240 DSP deals with the control of nine DC servo motors, nine encoders, six joint angle sensors, and three force/torque sensors [13]. With the hardware control system, the BH-3 hand can realize position control, force control, and hybrid control. Figure 10.12 shows BH-3 hand mounted on a PA-10 robot arm grasps a stainless steel cup based on the forces and torques of three finger tips.

**Fig. 10.12** BH-3 hand mounted on a PA-10 arm



With the TI DSP control board a control box ( $440 \times 180 \times 300$  mm) for the control of BH-3 hand was implemented, shown in Fig. 10.13. The control box is about 3 kg and there are about 40 cables between the control box and the robotic hand, so before each experiment we should make careful check of each electric connector and during each experiment we should also prevent those cables from being tangled with other objects. At that time, the cost, the volume, and the cables of the hardware control system were very big obstacles that prevent the extensive research and application of the BH-3 hand.

**Fig. 10.13** BH-3 hand control box with the DSP board



### 10.2.2 A DSP Based Hardware Control System of LARM Hand

A DSP based hardware control system was designed for the control of a three DOF, three-fingered LARM hand III, shown in Fig. 10.14. LARM hand III is composed of three 1 DOF human-like fingers transmitted by links. The actuation system consists of three DC motors with a planetary reduction gear box on each axis. Piezoresistive force sensors installed on the robotic hands monitor the grasping force. Piezoresistive effect consists in a resistance variation of a suitable material when a contact force is applied on it [16]. This kind of sensors has been chosen because of its small size and low cost. Therefore, it is necessary to use a proper conditioning circuit with the aim to obtain a tension output from a resistance variation. Furthermore, the encoder of each motor is also used to monitor the position of three fingers.

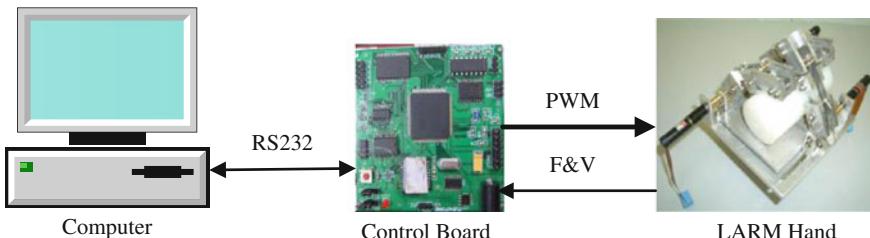
The design requirements of the hardware control system of LARM hand III are summarized as following:

- (i) Easy operation for users;
- (ii) Able to provide the control information such as force value, motor speed, and joint angles;
- (iii) Debug tools for tuning some control parameters.

To address the above requirements, the following solutions have been implemented in the design:

- (i) Provide some operation commands for the high-level control panel: Grasp by Position, Grasp by Force, Grasp by Step, Release and Emergency Stop;
- (ii) Send back current force values and joint angles information to the high-level control panel or monitor controller;
- (iii) Provide a debug interface that can set the control parameters and plot the performance curves.

According to the above considerations, a control scheme based on a DSP controller is designed, shown in Fig. 10.14. That control scheme is composed of two parts that communicate with each other through RS232 bus. The first part is a



**Fig. 10.14** Control architecture of LARM hand III

high-level control panel (shown in Fig. 10.15) programmed with Microsoft Visual Studio.net 2003, through which an operator can control or interacts with the LARM Hand. Another part is a low-level DSP controller that controls the motion of three motors via PWM pulse.

The control panel is divided into six sections with different functions. In the ‘Debug’ section, three buttons are used to control the motion of the LARM hand. For example, ‘Grasp by Step’ button can be used to control the hand to grasp an object step by step and can also be used to test the motion of three fingers; ‘Release’ button can be used to release the hand to the ‘zero’ or ‘open’ position; ‘Emergency’ button can be used to stop the motion of the hand when its motion is out of control or in other ‘emergency’ conditions.

In the ‘Force Control’ section, three edit boxes, ‘Finger1 Force’, ‘Finger2 Force’, and ‘Finger3 Force’, can be used to input ‘threshold’ force values on each finger and force-based grasp control can be realized according to the preset force values. ‘Set Force’ button is used to send the force values to the low-level controller through RS-232 bus. ‘Grasp-force’ button is used to control the hand to grasp an object based on the force control method that will be realized in the low-level controller. When inputting force values, it should be noted that the range of the input force value is from 0.5 to 100 N, which is also the detection range of the chosen force sensor.

In the ‘Position Control’ section, four edit boxes can be used to input parameters, three of which, ‘Finger1 Position’, ‘Finger2 Position’ and ‘Finger3 Position’, are expected angle positions of the first joints of three fingers, and the last one, ‘Reduction Ratio’, is the reduction ratio of the gear box attached on each motor. Here, it is supposed that each motor has same gear box and same reduction ratio.

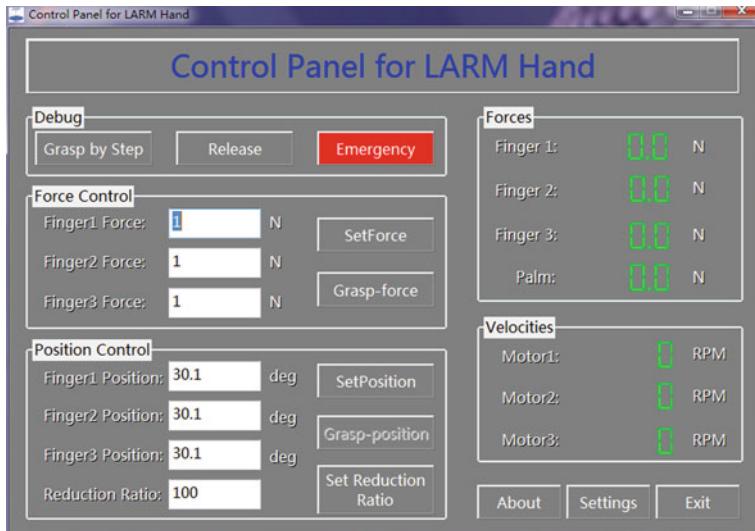
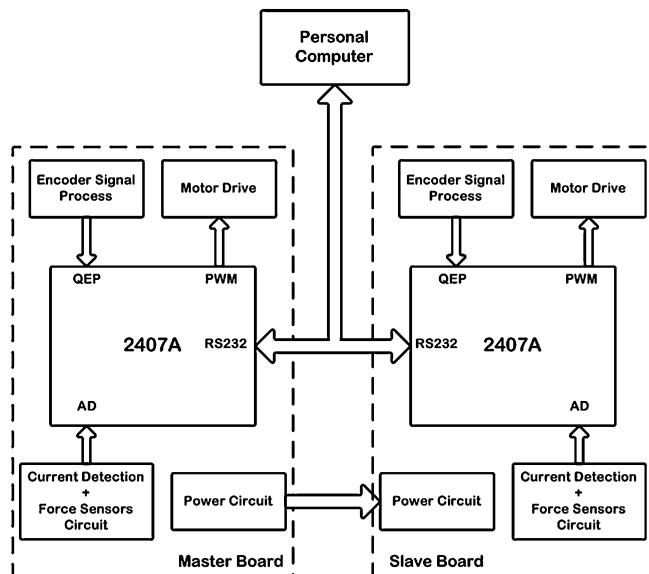


Fig. 10.15 Control panel of the LARM hand

‘Set Position’ button is used to send the position values to the low-level controller through RS-232 bus. ‘Set Reduc-Ratio’ button is used to send the input reduction value to the low-level controller through RS-232 bus. ‘Grasp-position’ button is used to control the hand to grasp an object based on the position control method that will be realized in the low-level controller. When inputting position values, it should be noted that the range of the input position value is from  $0^\circ$  to  $360^\circ$ .

In the ‘Forces’ section, the force values detected by four force sensors on three fingers and the palm will be displayed according to the information transmitted from the low-level controller by RS-232 bus. In the ‘Velocities’ section, velocities of three motors to drive three fingers will be displayed and the information is also transmitted from the low-level controller by RS-232 bus. Through these two sections, we can easily and directly watch the change of the contact forces and the velocities of three motors. Furthermore, the transmitted information of forces and velocities from the low-level controller will be saved to a file, so we can use this panel to control a hand to grasp an object and record the forces and velocities.

The DSP controller based on motor control chip TMS320LF2407A mainly consists of the power circuit, the encoder-signal processing circuit, the motor drive circuit, the force sensor’s signal processing circuit, and the RS-232 interface circuit. The hardware architecture diagram of the DSP controller is shown in Fig. 10.16 and two DSPs are used to compose a master–slave system, of which the two control boards are shown in Fig. 10.17, the master board at the left and the slave board at the right. The master board contains two motor drive circuits and performs the motion control of two fingers. The slave board contains one motor



**Fig. 10.16** Architecture diagram of the DSP controller



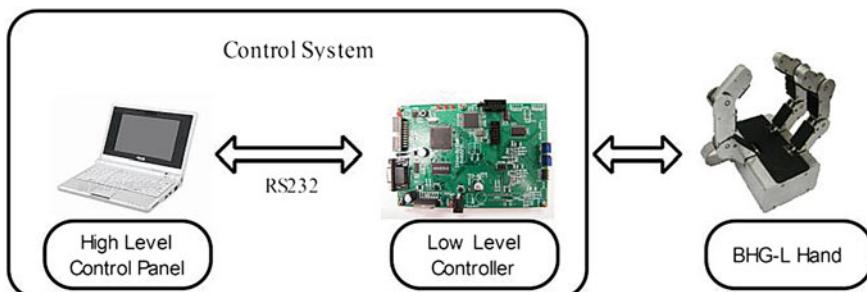
**Fig. 10.17** DSP control boards

drive circuit and performs the motion control of the third finger. Those two control boards can communicate with a PC through RS-232 bus. With the DSP controller, PID based position control and PD based force control are realized [4].

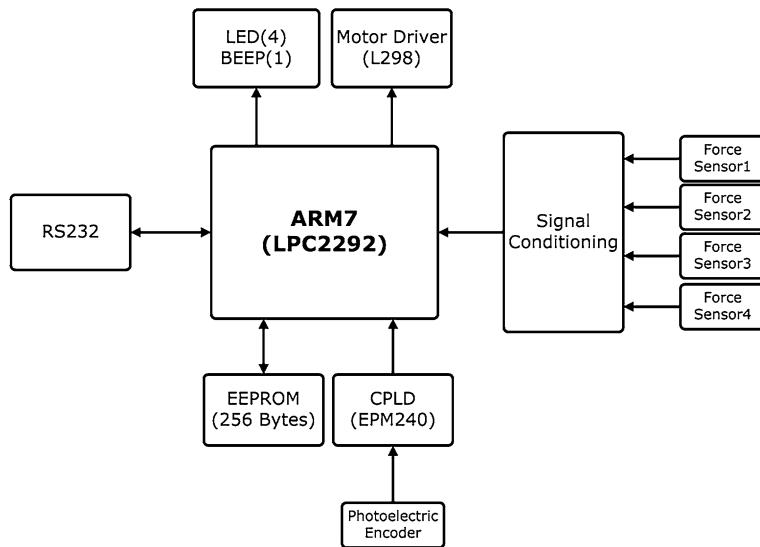
### 10.2.3 An ARM Based Hardware Control System of BHG-L Hand

BHG-L hand is a three-finger, one DOF robotic hand with link transmission to realize the synchronous motion of three phalanges of each finger. At the second phalange of each finger, there is a pressure sensor to detect the contact force between the finger and the grasped object. The actuated DC motor is embedded in the palm and the motion and torque are transmitted from the motor to the three fingers by tooth belts and a gear unit.

Similar to the control architecture of LARM hand, the control system of BHG-L hand shown in Fig. 10.18 is composed of two parts communicating with each other through RS-232 bus. One part is a high-level software control panel that can run at a PC, by which an operator can control and interact with the BHG-L hand;



**Fig. 10.18** Control scheme for BHG-L hand

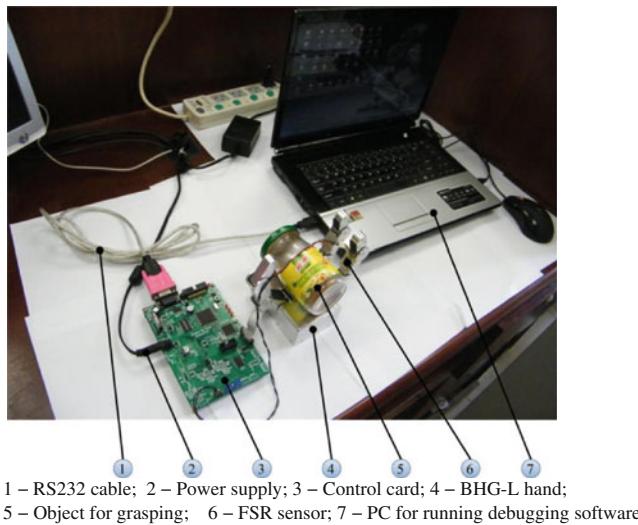


**Fig. 10.19** Hardware architecture diagram of low-level controller

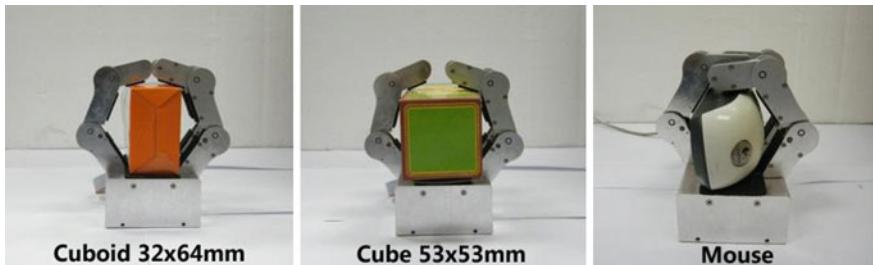
another part is a low-level control card that controls the motion of the motors via PWM pulse according to the control algorithm realized in it.

The hardware architecture of the low-level controller designed for the BHG-L hand is shown in Fig. 10.19 and the physical hardware is shown in Fig. 10.20. The hardware components for the basic close-loop control include a powerful ARM7 chip LPC2292 from NXP, a small CPLD chip EPM240T100C5 from ALTERA, a photoelectric encoder installed on the drive motor from MAXON, and the force sensing resistors (FSR) from INTERLINK. As shown in Fig. 10.19, the embedded architecture is built around the ARM7 (LPC2292) microcontroller. This computing unit provides many advantages for industrial control applications because it has abundant interfaces and robust performance. Indeed, it has a CPU clock of 60 MHz, a flash memory of 256 KB, and SDRAM of 16 KB that has been expanded to 512 KB in this control board for running  $\mu$ C/OS-II. However, the quadrature-encoder pulse (QEP) circuit has not been integrated in the LPC2292 chip. To improve the accuracy of position measurement for position control, a fourfold frequency multiplication circuit has been implemented in the CPLD (EPM240T100C5). The FSR is a polymer thick film device which exhibits a decrease in resistance with an increase in the force applied to the active surface. Those FSR sensors are installed on the second phalanges of all fingers for force measurement.

In the ARM control card, fuzzy logic based position control and a modified direct adaptive force tracking control method are designed. Figure 10.20 shows the setup of the experiment system and Fig. 10.21 shows the experiments that with the hardware controller the BHG-L hand grasps objects with different size and different shape so as to test its adaptability and some control methods.



**Fig. 10.20** Setup of the experiment system of BHG-L hand



**Fig. 10.21** Objects grasping experiments of BHG-L hand

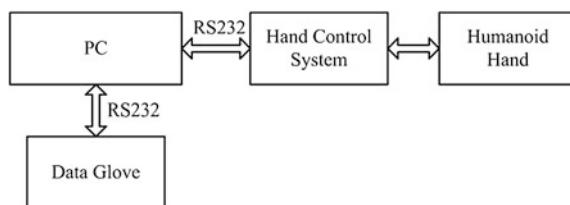
#### 10.2.4 An ARM Based Hardware Control System of CMIC-5 Hand

A five-fingered humanoid hand CMIC-5 implemented by our lab in 2011 is shown in Fig. 10.22 and it is about 1.2 times size of a normal human hand. CMIC-5 hand has five DC motors, of which four motors are embedded in four fingers except the little finger which is coupled with the ring finger, and the fifth motor is embedded in the palm to rotate the thumb. In each finger, steel string and worm gear are used to transmit motions from each motor to each joint.

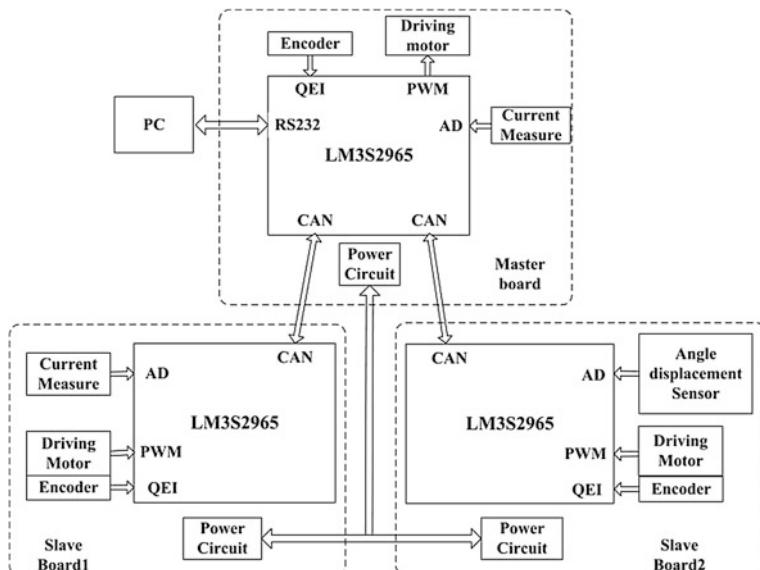
The control architecture of CMIC-5 hand is shown in Fig. 10.23. A PC is used as a host to gather motion instructions from external and send motion instructions



**Fig. 10.22** Experimental setup of CMIC-5 hand



**Fig. 10.23** Control scheme for CMIC-5 hand



**Fig. 10.24** Diagram of the control system of CMIC-5 hand



**Fig. 10.25** Grasping experiments of CMIC-5 hand

to the hand control system. At present, a data glove is employed to produce the motion instructions for the control system of the humanoid hand. Through RS-232 bus, the position information of each finger obtained from data glove is acquired and then is resent to the control system of the humanoid hand. The control system of the humanoid hand will complete the position control loop.

As shown in Fig. 10.24, the control system of CMIC-5 hand consists of three same boards. One is a master board for receiving motion instructions from the PC, and the other two boards are slave boards for motion control. Each control board can drive two DC motors and communicates with other boards through CAN bus. The main control chip is selected as TILM3S2965 which provides powerful motor control functions. To control the DC motors, an H-bridge chip LMD18200 is chosen in the control system. The PID based position control and armature current of DC motor based force control are realized in the hand control boards. Figure 10.25 shows a user control the CMIC-5 hand to grasp different objects through a data glove.

### 10.3 Conclusion

As an assembly of electronic and electric components, the hardware control system of robotic hands has gained significant progress during the past 30 years, which greatly promotes the development and application of robotic hands. Nowadays, the hardware control system of robotic hands has become so small that it can be embedded in the hand body and so fast that it can execute complex control methods in real time. In the future with the occurrence of much smaller and much faster microprocessors and other novel microprocessors, such as bioprocessors and optical processors, the hardware control system of robotic hands will make even more significant progress so as to promote the development and application of robotic hands to a new stage.

## References

1. Butterfass J, Hirzinger G, Knoch S, Liu H (1998) DLR's multisensory articulated hand part I: hard and software architecture. In: Proceedings of the IEEE international conference on robotics and automation, vol 3, pp 2081–2086
2. Butterfass J, Grebenstein M, Liu H, Hirzinger G (2001) DLR-Hand II: next generation of a dexterous robot hand. In: Proceedings of the international conference on robotics and automation. Seoul, Korea, pp 109–114
3. Butterfass J, Fischer M, Grebenstein M, Haidacher S, Hirzinger G (2004) Design and experiences with DLR hand II. In: Proceedings of the tenth international symposium on robotics with applications. Seville, Spain
4. Cai Y, Zhan Q, Ceccarelli M, Carbone G, Yao S, Lu Z (2010) Design and simulation of a DSP controller for a LARM hand. In: Proceedings of the 2nd international Asia conference on informatics in control, automation and robotics, vol 1. Wuhan, pp 361–364
5. Cipriani C, Controzzi M, Vecchi F, Carrozza MC (2008) Embedded hardware architecture based on microcontrollers for the action and perception of a transradial prosthesis. In: Proceedings of the 2nd Biennial IEEE/RAS-EMBS international conference on biomedical robotics and biomechatronics. Scottsdale, AZ, USA, pp 848–853
6. Cipriani C, Controzzi M, Carrozza MC (2009) Progress towards the development of the SmartHand transradial prosthesis. In: Proceedings of the IEEE international conference on rehabilitation robotics. Kyoto, Japan, pp 682–687
7. Cipriani C, Controzzi M, Carrozza MC (2010) Objectives, criteria and methods for the design of the SmartHand transradial prosthesis. *Robotica* 28:919–927
8. Cipriani C, Controzzi M, Carrozza MC (2011) The SmartHand transradial prosthesis. *J NeuroEng Rehabil* 8(29)
9. Connolly C (2008) Prosthetic hands from touch bionics. *Ind Robot* 35(4):290–293
10. Guo G, Zhang Q (1987) Structure study of a new multi-functional hand-mechanism. *Robot* 1(6):26–31
11. Guo G, Zhang Q (1989) Analysis and optimization design of a three D.O.F. multi-articulated finger-mechanism. *Robot* 3(2):51–57
12. Haidacher S, Butterfass J, Fischer M, Grebenstein M, Joehl K (2003) DLR hand II: hard- and software architecture for information processing. In: Proceedings of the 2003 IEEE international conference on robotics and automation. Taipei, Taiwan, pp 684–689
13. Han Z (2003) Research on data processing of force/torque sensors and control for dexterous hand. Ph.D. dissertation, Beihang University, pp 85–89
14. Jacobsen SC, Wood JE, Knutti DF, Biggers KB, Iversen EK (1984) The UTA/M.I.T. dexterous hand: work in progress. *Int J Robot Res* 4(3):21–50
15. Jacobsen SC, Iversen IK, Knutti D, Johnson RT, Biggers KB (1986) Design of the UTA-MIT dexterous hand. In: Proceedings of IEEE international conference on robotics and automation. USA, pp 1520–1532
16. Konteck C (2001) Specification Sheet for Standard Lusense Sensors of PS3 Family
17. Liu H, Meusel P, Hirzinger G, Jin M, Liu Y, Xie Z (2008) The modular multisensory DLR-HIT-Hand: hardware and software architecture. *IEEE/ASME Trans Mechatron* 13(4):461–469
18. Loucks C, Johnson V, Boissiere P et al (1987) Modeling and control of the stanford/JPL hand. In: IEEE international conference on robotics and automation, pp 573–578
19. Maeda Y, Tachi S, Fujikawa A (1989) Development of an anthropomorphic hand (Mark-I). In: Proceedings of international symposium on industrial robots. Tokyo, pp 537–544
20. Motorola Inc (1992) Motorola M68000 Family Programmers Reference Manual
21. Narasimhan S, Siegel DM, Hollerbach JM, Biggers KB, Gerpheide GE (1986) Implementation of control methodologies on the computational architecture for the Utah/MIT hand. In: Proceedings of the IEEE conference on robotics and automation, vol 3. San Francisco, pp 1884–1889

22. Narasimhan S, Siegel DM, Hollerbach JM (1988) Condor: a revised architecture for controlling the Utah-MIT hand. In: Proceedings of the IEEE international conference on robotics and automation, vol 1, pp 446–449
23. Wei R, Gao XH, Jin MH, Liu YW, Liu H, Seitz N, Ggruber R, Hirzinger G (2005) FPGA based hardware architecture for HIT/DLR hand. In: Proceedings of IEEE/RSJ international conference on intelligent robots System. Edmonton, Canada, pp 3233–3238
24. Zhan Q, Zhang Q (2001) A dual-arm plaform for inner space station operation. In: Proceedings of the international conference on bio-robotics and teleportation. Beijing, pp 347–350

# Chapter 11

## Visual Servo Control of Robot Grasping

Theodor Borangiu

**Abstract** This chapter describes the basic principles and methods for visual servo control of robot manipulators in grasping tasks, proposing implementing solutions with multitasking robot-vision controllers. *Guidance vision* is introduced as an advanced motion control method, which provides flexibility when integrating robots in manufacturing cells with unstructured environment and in line quality inspection. Two important system architectures are analyzed: *dynamic look-and-move* systems (open loop robot-vision architectures), and *direct visual servoing* systems (closed-loop robot-vision architectures). The generic task analyzed in this chapter is the visual tracking of material flows with fixed and mobile cameras for robot grasping from stationary and moving scenes. Novel contributions are included with respect to modeling the grasping style, robot guidance from mobile, arm-mounted cameras, and authorizing collision-free object grasping based on real-time fingerprint evaluation.

### 11.1 Introduction

Tasks in visual servoing consist in controlling the motion to manipulate the robot environment using vision, as opposed to just observing the environment, like in active vision and structure from motion. A robot motion, guided by vision to grasp an object or to interact with it, will be planned from visual data created at run time and mapped into the operational space to provide the transformation  $\mathbf{x}(t) = \mathbf{x}_n^0(t)$ ,  $\mathbf{x} \in \mathbf{R}^r, t \in \mathbf{R}^+$  which is necessary for the end effector to access the object or interact with it in a desired manner.

---

T. Borangiu (✉)

Centre of Research in Robotics CIMR, University Politehnica of Bucharest,  
313, Spl. Independentei, District 6 060042 Bucharest, Romania  
e-mail: theodor.borangiu@cimr.pub.ro

Visual servoing of robots uses structural features extracted from images as *form* and *contour image features* for object recognition and locating or collision avoidance. *Context features* may be added to the data to simplify object search at run time. The form and contour image features refer to the projection of a *body* or *hole* physical feature of an object (e.g., the part to be grasped, the gripper's fingers, or the robot tool) onto the camera image plane. Typical image features are: *edges* and *corners* for contours, respectively the *shape*, *center of mass*, *orientation* of bodies, and holes or *contrived patterns* to form descriptors. Image features must be unambiguously located in different views of the robot scene by different virtual cameras [1]. Visual servoing systems typically use one of the following camera configurations:

- *Stationary* (fixed outside the robot workspace): their location is time-invariant relative to the robot base frame  $(x_0, y_0, z_0)$ ;
- *Mobile* (arm-mounted or hand-held): their location relative to  $(x_0, y_0, z_0)$  is time variant as the robot link on which the camera is mounted moves relative to the world frame.

This chapter discusses, for both cases, visual servo control laws of *feature tracking for object grasping* using Guidance Vision of the Robot (GVR) to pick objects at rest from stationary scenes or “on-the-fly” from conveyor belts.

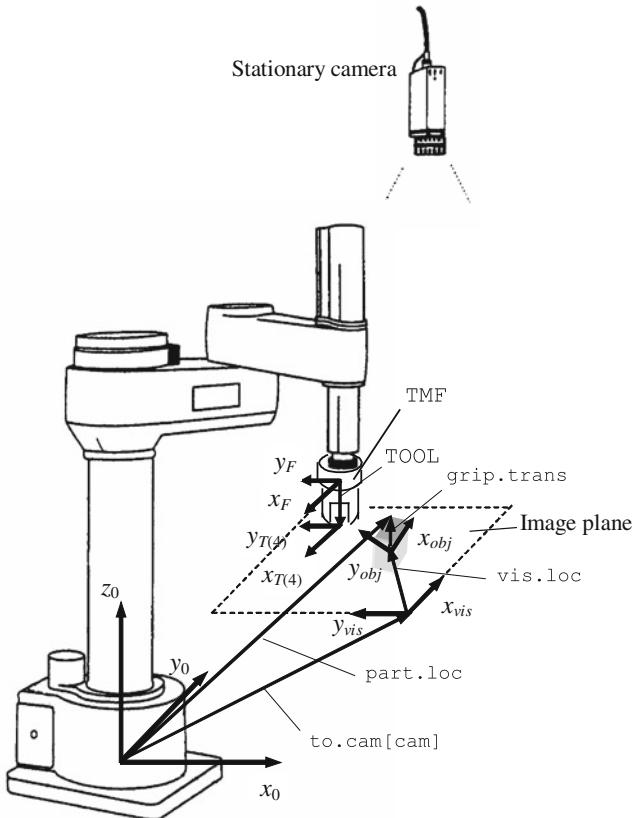
A systemic view of the robot motion planning and tracking for robot grasping stationary and moving parts is presented in this chapter. This view allows the task-oriented management of a scene's foreground and provides part qualifying, robustness to variation in robot workspace lighting and adaptation to unstructured material flows to be accessed by the robot. For arm-mounted cameras, there will be discussed a solution to reuse at run time the camera-robot model for multiple picture-taking points.

Random scene foregrounds, as the conveyor belt, may need to be faced in robotic tasks. Depending on the parts shape and their dimension along  $z^+$ , grasping models  $Gs\_m$  are off-line trained for object classes. If there is the slightest uncertainty about the risk of collision between the gripper and parts in the scene—touching or very close one to the other—then *extended grasping models*  $\mathcal{EG}_m = \{Gs\_m, \mathcal{FGP}\_m\}$  must be created by adding the gripper's *fingerprint model*  $\mathcal{FGP}\_m$  to authorize part access only after *clear grip tests* at run time.

## 11.2 Frames, Image Features, and Visual Servoing Taxonomies

### 11.2.1 *Frames, Image Features and Camera Configuration*

Robotic tasks are typically described with respect to more than one coordinate frames. The fixed or moving locations of these frames can be linked via relative



**Fig. 11.1** Robot and vision frames, and relative coordinate transformations linking them

transformations, or poses. A robot motion, guided by vision to grasp an object or to interact with it, will be planned from visual data created at run time and mapped into the operational space to provide the pose  $\mathbf{x}(t) = \mathbf{x}_n^0(t)$ ,  $\mathbf{x} \in \mathbf{R}^r, t \in \mathbf{R}^+$  such that the end-effector accesses the object or interacts with it in a desired manner, imposed by the application task [2]. An analysis of the components of this transformation and of the coordinate frames they interconnect results from Fig. 11.1 which shows a SCARA robot manipulator and a stationary, down looking camera.

The visually planned and tracked motion will guide the robot toward the grasping location of an object of interest identified and located in the image plane ( $x_{vis}, y_{vis}$ ). There are four components contributing to the computation of the end-effector's destination transformation  $\mathbf{x}_n^0 = part.loc$  which is relative to the base frame  $(x_0, y_0, z_0)$  of the robot fixed base:

1. The position and orientation offset  $\mathbf{x}_{\text{vis}}^0 = \text{to.cam}[\text{cam}]$  of the image coordinate frame  $(x_{\text{vis}}, y_{\text{vis}})$  with respect to the world frame  $(x_0, y_0, z_0)$  of the robot. This coordinate transformation is created off-line by an interactive *camera–robot calibration* program for all virtual cameras *cam* in use.
2. The pose  $\mathbf{x}_{\text{obj}}^{\text{vis}} = \text{vis.loc}$  of the object-attached frame  $(x_{\text{obj}}, y_{\text{obj}})$  relative to the vision frame  $(x_{\text{vis}}, y_{\text{vis}})$ . This coordinate transformation is computed at run time by the vision processor and describes, in most cases, the location of the frame placed in the center of mass of the currently identified object, with an orientation expressed by the angle  $\angle(\text{MIA}, x_{\text{vis}})$  between the minimum inertia axis (MIA) of the object's 2D image and the axis  $x_{\text{vis}}$ .
3. The pose  $\mathbf{x}_T^{\text{obj}} = \mathbf{x}_n^{\text{obj}} = \text{grip.trans}$  of the frame  $(x_T, y_T, z_T)$  attached to the end effector with respect to the object's locating frame  $(x_{\text{obj}}, y_{\text{obj}})$ . This relative transformation is always considered when visually controlling object grasping, to recuperate the third dimension (the depth of view) which was lost during image acquisition by the 2D projection. For the SCARA robot in Fig. 11.1,  $n = 4$  and  $(x_T, y_T, z_T) = (x_4, y_4, z_4)$  in the end-tip point T.
4. The pose  $\mathbf{x}_T^F = \mathbf{x}_n^F = \text{TOOL}$  of the end-effector frame  $(x_T, y_T, z_T)$  with respect to the coordinate frame  $(x_F, y_F, z_F)$  which is attached to the tool mounting flange TMF of the robot manipulator.

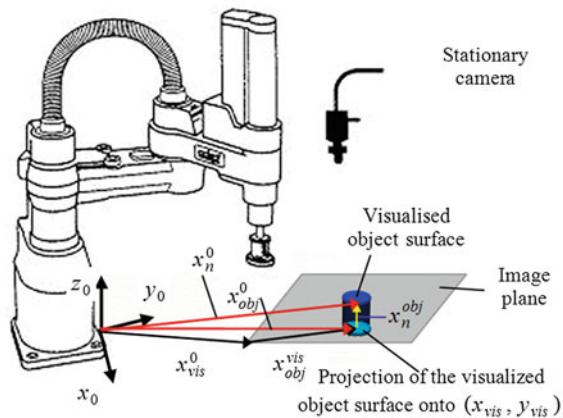
*Image feature parameters* represent real-valued quantities that are computed from one or several image features. Typical feature parameters used in visual servo control are: the image plane coordinates  $u, v$  of visualized object points  $\mathbf{P}^{\text{vis}}$ ; the distance between two image points,  $\text{dist}(\mathbf{P}^{\text{vis}}, \mathbf{Q}^{\text{vis}})$ , and the orientation of the line connecting these two points,  $\angle(\mathbf{P}^{\text{vis}} \mathbf{Q}^{\text{vis}}, x_{\text{vis}})$ ; the parameters of the gripper's fingerprints projection on the image plane: shape, area, location, and so on.

Hence, a real-valued mathematical or logical expression having arguments one or more image feature parameters may be used to dynamically update the description of an object (part to be tracked/grasped) or robot tool. Image feature parameters can be stored in numerical form  $f_i$  (possibly bounded); they map the *object space* into the *feature parameter space*  $\mathcal{F}$ , by means of feature parameter vectors  $\mathbf{f} = [f_1 \ f_2 \ \dots \ f_k]^T \in \mathcal{F} \subseteq \mathbf{R}^k$ . Any feature parameter  $f_i$  takes values in  $\mathcal{F}_i$ . The set of all  $k$  chosen feature parameters defines a function which maps the object space into the Cartesian product  $\mathcal{F}_1 \times \mathcal{F}_2 \times \dots \times \mathcal{F}_k$ . Thus, a certain object to be visually tracked and grasped, or a gripper (tool) to be visually positioned will appear as a point in the  $k$ -dimensional feature space.

The mapping from the location of the end effector to the set of corresponding image feature parameters, computed according to the projective geometry of the camera will be denoted by  $\mathbf{F}: \mathcal{T} \rightarrow \mathcal{F}$ . Considering  $\mathcal{F} \subseteq \mathbf{R}^2$  the feature parameter space of  $u, v$  coordinates of the projection of some point  $\mathbf{P}^{\text{vis}}$  onto the image plane  $(x_{\text{vis}}, y_{\text{vis}})$ , then, assuming perspective projection,  $\mathbf{f} = [u \ v]^T$ .

Visual servo systems typically use one of two camera configurations: stationary (fixed outside the robot workspace), respectively *mobile* (*arm-mounted* or *hand-held*). For any of the two cases, one should make the difference between

**Fig. 11.2** Stationary camera configuration and related camera–robot relative transformations  $x_{obj}^0, x_n^0$  respectively for feature tracking- and feature tracking for object grasping tasks



two classes of robot-vision tasks, which lead to distinct visual servo control laws:

- *Feature tracking* for Automated Visual Inspection (AVI) applications: the parameters of image feature(s) on a part are evaluated, but the part is not extracted from the fixed scene or conveyor belt.
- *Feature tracking for object grasping* in Robotic tasks with Guidance Vision (RV), when the robot motion is visually guided to pick objects at rest or “on-the-fly” from conveyor belts [3–5].

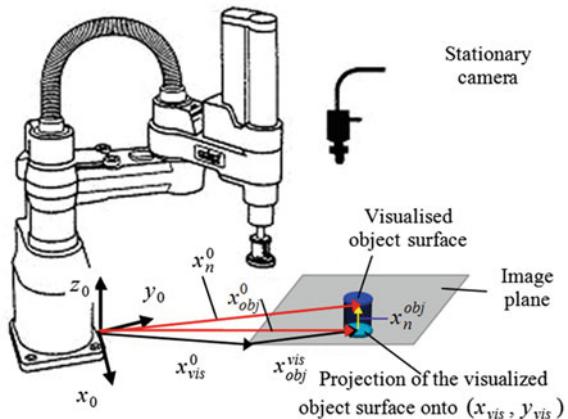
Figure 11.2 shows the stationary camera configuration and the camera–robot transformations.

The pair “physical camera–virtual camera (a data set describing the task-driven context in which the physical camera’s information will be interpreted)” is related to the base coordinate system of the robot by the time-invariant pose  $x_{vis}^0$  evaluated a single time during an interactive off-line camera–robot calibration session, and to the object in the scene by  $x_{obj}^{vis}$ .

The camera image of the object  $x_{obj}^{vis}$  is independent of the robot motion (unless the target is the end effector itself, described for example by image feature of the gripper’s fingerprints projected onto the image plane). The pose  $x_{obj}^{vis}$  is variable in time for parts traveling on conveyor belts; the computation of this information is done at run time, and involves the *search*, *recognition*, and *locating* of the image features(s) on the object of interest [6, 7].

For object grasping, the image features must unambiguously describe the entire object for its successful identification and locating at run time. In addition, the pose  $x_n^{obj}$  of the gripper, relative to the frame attached to the object in its current location, is required.

**Fig. 11.3** Mobile camera configuration and related camera–robot relative transformations  $x_{obj}^0, x_n^0$  respectively for feature tracking and feature tracking for object grasping tasks



For a *stationary camera*, the relationship between these poses is:

$$\begin{aligned} \mathbf{x}_{obj}^0 &= \mathbf{x}_{vis}^0 : \mathbf{x}_{obj}^0, \text{ for (a) feature tracking} \\ \mathbf{x}_n^0 &= \mathbf{x}_{vis}^0 : \mathbf{x}_{obj}^0 : \mathbf{x}_n^{obj}, \text{ for (b) feature tracking for object grasping} \end{aligned} \quad (11.1)$$

Figure 11.3 shows the mobile camera configuration (*arm-mounted*) and the related camera–robot transformations.

In this case, there is a learned, constant coordinate transformation  $\mathbf{x}_{vis}^s$  between the pose of the camera and the pose of the segment  $s$  of the manipulator, on which the camera is mounted:

$$s = \begin{cases} 3, & \text{for arm - mounted camera (usually on link 3 - the arm)} \\ n, & \text{for hand - held camera (mounted on link } n \text{ - the gripper)} \end{cases}$$

The relative pose  $\mathbf{x}_{vis}^s$  is a single time computed by camera–robot calibration.

In order to reuse the calibration transformation at any picture-taking point where the camera is moved and snaps images, the pose  $\mathbf{x}_s^0$  of the link  $s$ -attached frame relative to the world frame must be backwards computed at run time from the encoder data in  $\mathbf{x}_n^0$  at the picture-taking points, by removing any current displacement of the terminal joints  $i, s \leq i \leq n$  [8].

For a mobile camera mounted on segment  $s$  of the robot manipulator, which locates image features of objects  $\mathbf{x}_{obj}^s$  relative to  $(x_{vis}, y_{vis})$ , the two feature-tracking problems are expressed by the next pose composition equations:

$$\begin{aligned} \mathbf{x}_{obj}^0 &= \mathbf{x}_s^0 : \mathbf{x}_{vis}^s : \mathbf{x}_{obj}^s, \text{ for (a) feature tracking} \\ \mathbf{x}_n^0 &= \mathbf{x}_s^0 : \mathbf{x}_{vis}^s : \mathbf{x}_{obj}^s : \mathbf{x}_n^{obj}, \text{ for (b) feature tracking for object grasping} \end{aligned} \quad (11.2)$$

For either choice of the camera configuration, prior to executing a visual servo task of type "feature tracking for object inspecting" or "feature tracking for object grasping":

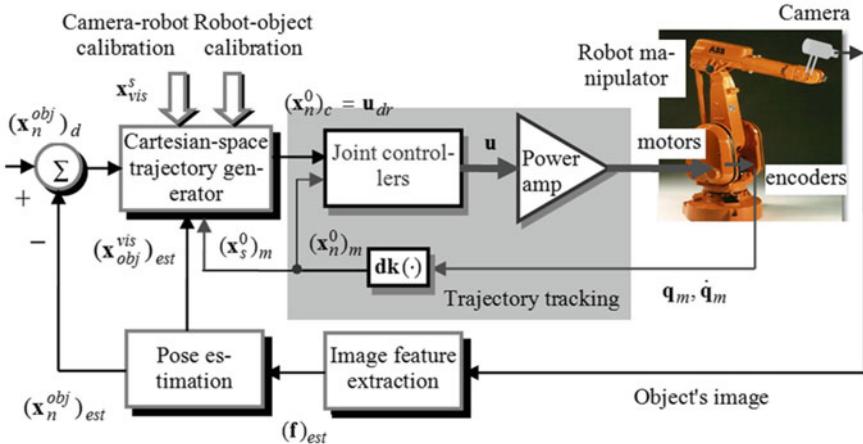
- Camera calibration must be performed to determine the *intrinsic camera parameters* such as focal length, pixel pitch, a.o.; in actual robot-vision programming environments (such as V+) these parameters are encapsulated in coefficients which express, respectively, the ratio between one pixel and one millimetre, and the ratio between the pixel's width and height;
- Camera-Robot Calibration must be performed in order to determine the *extrinsic camera parameters* which are encapsulated in the fixed camera's pose: either  $\mathbf{x}_{\text{vis}}^0$  relative to the world frame  $(x_0, y_0, z_0)$  for a stationary camera, or  $\mathbf{x}_{\text{vis}}^s$  relative to the frame  $(x_s, y_s, z_s)$  attached to the segment on which the mobile camera is mounted.

The feature-based description of objects (gripper, parts) allows planning and updating the motion of a robot (visual servo) at run time, either for *automated visual inspection* by tracking image features and computing the values of their parameters, or for *robot guidance* by tracking and locating image features the values of which are further used to compute the object grasping data [9].

### **11.2.2 Visual Servoing Architectures**

The taxonomy of visual servoing architectures is defined in a first stage, by classifying robot-vision systems according to the *type of structure* of the closed-loop motion controller (A or B) and to its *type of control law* (1 or 2):

- A. *Hierarchical motion control structure*, with the vision processor providing set points as reference input to the robot's joint-level controller—thus using joint data feedback to internally stabilize the robot. This structure corresponds to the *interlaced look-and-move* control schemes, where motion tracking and image processing are *pipelined* (Fig. 11.4).
- B. *Direct motion control structure*, with the visual motion controller directly computing the joint-level inputs. This corresponds to *direct visual servo* schemes, the trajectory controller being entirely eliminated and replaced by a visual servo controller that directly computes joint inputs, thus using vision alone to stabilize the robot.
1. *Position-based control structures* use error signals defined in task space coordinates. Features are extracted from images and their associated parameters are computed and used in conjunction with a geometric model of the visualized object and of the known camera–robot ensemble, to estimate first the pose of the object with respect to the camera. Feedback is then computed by reducing errors in estimated pose space, and applied to the task-space trajectory generator.



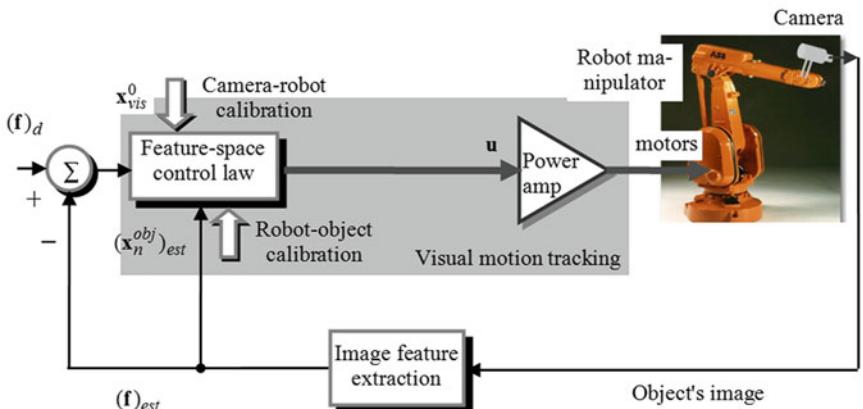
**Fig. 11.4** Position-based look-and-move servoing scheme for object tracking (system structure A1)

2. *Image-based control structures* use error signals defined directly in terms of image features extracted from binary or grayscale object's images.

It can be observed that, whereas the global robotic system operates in an *open loop structure at motion control level*, it is subject to a *closed-loop control at the global task level* [8, 10, 11].

In image-based servoing, control values are computed at run time directly from image features.

This significantly reduces image interpretation and eliminates errors caused by sensor modeling and camera calibration. However, this technique induces an



**Fig. 11.5** Image-based direct visual servoing architecture for object tracking (system structure B2)

important challenge to motion controller design, since the manipulator plant is nonlinear and strongly coupled [12] (Fig. 11.5).

Nearly, all implemented robot-vision systems, including all industrial systems adopt, from the four above presented architectures, the interlaced look-and-move control schemes A1 or A2 due to three basic reasons:

- *Image processing* (acquisition and enhancement, segmentation, feature extraction, and object recognition and locating) is relatively time consuming leading to low sampling rates of about 8–16 ms, which makes direct motion control of the end effector an extremely challenging problem, in view of its nonlinear, coupled dynamics. Interlaced look-and-move schemes A1 and A2 use internal measured joint feedback  $\mathbf{q}_m, \dot{\mathbf{q}}_m$  at a high sampling rate of 1 ms to stabilize the robot by DK feedback, and to offer the vision processor an “idealised” axis dynamics.
- *Human-robot interfaces* accept nowadays joint or Cartesian velocity specification or incremental position commands, allowing thus portability and simplification of the visual servo software.
- A decoupling of the kinematic singularities of the manipulator from the visual controller is provided by the look-and-move scheme, allowing the robot to be integrated into the closed-loop architecture at task level as an ideal Cartesian motion device [13–16].

In most cases, the control algorithm is expressed in one of the following terms:

- *Moving the camera* (hand-held or arm-mounted) to a pose specified in terms of a number of image features extracted from the segmented image of the object to be grasped.
- *Moving the end effector* to a pose specified in terms of a number of feature images once extracted—when the object was located by a fixed camera from its segmented image.

In both cases, the position of the end-effector relative to the world frame  $\mathbf{x}_n^0$  is determined at run time indirectly (using the composition of relative poses) by its a priori known kinematic relationship:  $\mathbf{x}_{\text{vis}}^0$ —for stationary camera or  $\mathbf{x}_{\text{vis}}^s$ —for mobile camera (camera–robot calibration), and by its learned and updated grasping transformation  $\mathbf{x}_n^{\text{obj}}$  (robot-object calibration). However, errors in the camera–robot calibration relationships  $\mathbf{x}_{\text{vis}}^0, \mathbf{x}_{\text{vis}}^s$  lead to positioning errors which cannot be observed and compensated by the vision part, unless control points on both the end effector and the target object can be visually observed.

Thus, the distinction will be made between *endpoint open-loop* (EOL) systems that only observe the target object, and *endpoint closed-loop* (ECL) systems that observe both the target object and the robot end-effector. As compared to EOL structures, ECL systems are more accurate, but imply more demanding vision problems and induce field of view constraints on image processing [17].

*Position-based look-and-move control* will be further discussed, according to the system structure A1 represented in Fig. 11.4. Features are extracted from the image and used to estimate the pose  $\hat{\mathbf{x}}_{\text{obj}}^{\text{vis}} = (\mathbf{x}_{\text{obj}}^{\text{vis}})_{\text{est}}$  of the target (object, point)

relative to the camera. Using these values, an error between the current estimated and the desired pose of the robot  $(\mathbf{x}_{\text{obj}}^{\text{vis}})_d$ , is defined in the task space  $\mathcal{T}$ .

In this way, position-based control neatly separates the control actions, i.e., the computation of the feedback signal  $(\mathbf{x}_s^0)_m = \mathbf{dk}(\mathbf{q}_m, s)$ ,  $n \leq s \leq n$  using the direct kinematics model  $\mathbf{dk}(\cdot)$  of the robot manipulator, from the estimation problem involved in computing position or pose  $\hat{\mathbf{x}}_{\text{obj}}^{\text{vis}}$  from visual data  $(\mathbf{f})_{\text{est}}$ .

*Definition 1.* A *visual positioning task* is expressed by an error function  $\mathbf{E}: \mathcal{T} \rightarrow \mathbf{R}^m$ . This function is referred to as *virtual kinematic error function* VKE. A positioning task is fulfilled when the end effector has been moved in pose  $\mathbf{x}_n = \mathbf{x}_n^0$  if  $\mathbf{E}(\mathbf{x}_n) = \mathbf{0}$ .

Once a suitable VKE function was defined and its parameters are instantiated from visual data, a compensator can be designed that reduces the value of the VKE function to zero. This compensator computes at every sampling time instant the necessary end-effector position  $(\mathbf{x}_n)_c$  that is sent as dynamic reference  $\mathbf{u}_{dr}$  to the joint-space (or operational-space) motion tracking controller. Since the VKE functions are defined usually in the Cartesian space, it is common sense to develop the compensator's control law through geometric insight.

### 11.2.3 Robot Motion Control Based on Pose-Feature Extraction

In industrial applications of position-based dynamic look-and-move control structures, the robot–vision system works in most cases with off-line learned objects which can be visually recognized and located at run time [18]. It becomes thus possible:

- To recover the object's pose,  $\hat{\mathbf{x}}_{\text{obj}}$ , relative to the base frame of the robot, from the direct estimate  $\hat{\mathbf{x}}_{\text{obj}}^{\text{vis}}$  of the object's pose in the vision frame and by composing it with the camera–robot calibration estimate  $\hat{\mathbf{x}}_{\text{vis}}$ ;
- To define stationing points  $\mathbf{S}^{\text{obj}}$  on the object's image, relative to a suitable object-attached frame  $(x_{\text{obj}}, y_{\text{obj}})$ .

The problem of visual feature tracking for object grasping control will be further presented for the two well-known camera configurations.

#### 11.2.3.1 Stationary, Down Looking Camera

Assuming a random part presentation in the robot workstation, the object's pose relative to a (unique) camera frame  $\hat{\mathbf{x}}_{\text{obj\_1}}^{\text{vis}}$ , will be estimated at run time, in a *first stage* in terms of the following image feature parameters:

- $x_C, y_C$ : coordinates of the center of mass C of the 2D projection of the object's visualized surface onto the image plane  $(x_{\text{vis}}, y_{\text{vis}})$ ;
- $\text{orient} = \angle(\text{MIA}, x_{\text{vis}})$ : orientation angle of the object.

The object-attached frame  $(x_{\text{obj}\_1}, y_{\text{obj}\_1})$  will have the origin in  $\mathbf{C}$  and the abscissa  $x_{\text{obj}\_1} \equiv \text{MIA}$  (Fig. 11.6).

To grasp objects of a certain type always in the same way, irrespective of their location in the robot scene, the desired (unique) pose of the gripper,  $\mathbf{x}_{n^*}^{\text{obj}}$ , relative to the object-attached frame must be a priori learned.

Let us denote by  $\mathbf{G}$  the projection of the end-tip point  $\mathbf{T}$ , the origin of the gripper-attached frame  $(x_n, y_n, z_n)$ , onto the image plane:  $\mathbf{G} = \text{proj}|_{(x_{\text{vis}}, y_{\text{vis}})}\{\mathbf{T}\}$ .

For a desired grasping style,  $\mathbf{G}^{\text{obj}\_1}$  is a stationing point in the object's coordinates  $(x_{\text{obj}\_1}, y_{\text{obj}\_1})$ , irrespective of the current position and orientation of the object. Its coordinates are:

$$x_G = d_{CG} * \cos(\alpha) \quad y_G = d_{CG} * \sin(\alpha) \quad (11.3)$$

where  $d_{CG} = \text{dist}(\mathbf{C}, \mathbf{G})$ , and  $\alpha = \angle(\text{dir}(\mathbf{C}, \mathbf{G}), \text{MIA})$  measured CCW from MIA to the direction CG, i.e.  $\text{dir}(\mathbf{C}, \mathbf{G})$ .

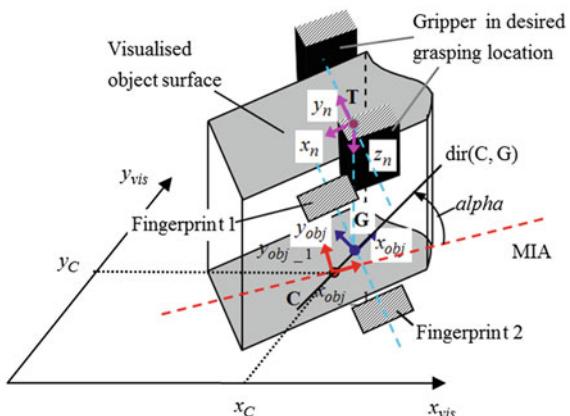
In a second stage, the object-attached frame will be shifted to origin  $\mathbf{G}$ , by a translation of distance  $d_{CG}$  along  $\text{dir}(\text{MIA})$  followed by a rotation of angle  $\alpha$  about the normal in  $\mathbf{C}$  to the image plane, as represented in Fig. 11.6.

Given an object pose,  $\mathbf{x}_{\text{obj}}^{\text{vis}}$ , estimated visually at run time, and assuming that the object was recognized as a member of that class for which a relative grasping pose  $\mathbf{x}_{n^*}^{\text{obj}}$  was a priori learned using a stationary camera calibrated to the robot base frame by  $\mathbf{x}_{\text{vis}}$ , then the positioning error can be defined by the VKE function

$$\mathbf{E}(\mathbf{x}_n; \tilde{\mathbf{x}}_{n^*}^{\text{obj}}, \hat{\mathbf{x}}_{\text{obj}}^{\text{vis}}, \hat{\mathbf{x}}_{\text{vis}}) = \mathbf{x}_{n^*}^n = \hat{\mathbf{x}}_0^n : \hat{\mathbf{x}}_{\text{vis}} : \hat{\mathbf{x}}_{\text{obj}}^{\text{vis}} : \tilde{\mathbf{x}}_{n^*}^{\text{obj}}, \quad (11.4)$$

where  $\tilde{\mathbf{x}}_{n^*}^{\text{obj}} = \begin{cases} \mathbf{x}_{n^*}^{\text{obj}} & \text{a priori known from learning, particular "grasping style"} \\ \hat{\mathbf{x}}_{n^*}^{\text{obj}} & \text{visually updated at run time, general "grasping style"} \end{cases}$

**Fig. 11.6** The 2-stage definition of the object-attached frame



With an EOL system,  $\hat{\mathbf{x}}_0^n = \text{inverse}(\hat{\mathbf{x}}_n^0)$  will be dynamically updated by the trajectory generator to bring to zero the positioning error  $\mathbf{x}_{n^*}^n$ . This can be simply done applying for an IK-based Resolved Motion Rate Control algorithm.

The closed-loop servo control uses the visually estimated pose of the object,  $\hat{\mathbf{x}}_{\text{obj}}^{\text{vis}}$ , the estimated camera–robot calibration pose,  $\hat{\mathbf{x}}_{\text{vis}}$  and assumes that reduced-error direct kinematics ( $\hat{\mathbf{x}}_n^0$ )—and inverse kinematics ( $\hat{\mathbf{x}}_0^n$ ) models are available.

As for the imposed grasping pose, for a priori unknown object location in the scene, some components in  $\hat{\mathbf{x}}_{n^*}^{\text{obj}}$  must be estimated at run time whenever the “grasping style” is *general*, i.e., such that  $\mathbf{G} \neq \mathbf{C}$  and  $\mathbf{G}$  does not lie on MIA.

With an ECL system capable to observe both the object ( $\hat{\mathbf{x}}_{\text{obj}}^{\text{vis}}$ ) and the end effector ( $\hat{\mathbf{x}}_n^{\text{vis}}$ ) and to estimate their poses, the error Eq. (4) can be written as:

$$\mathbf{x}_{n^*}^n = \hat{\mathbf{x}}_{\text{vis}}^n : \hat{\mathbf{x}}_0^{\text{vis}} : \hat{\mathbf{x}}_{\text{vis}}^0 : \hat{\mathbf{x}}_{\text{obj}}^{\text{vis}} : \tilde{\mathbf{x}}_{n^*}^{\text{obj}} = \hat{\mathbf{x}}_{\text{vis}}^n : \hat{\mathbf{x}}_{\text{obj}}^{\text{vis}} : \tilde{\mathbf{x}}_{n^*}^{\text{obj}} \quad (11.5)$$

One can notice that for an ECL system the uncertainties in both the robot’s kinematics models and in the camera–robot calibration model do not affect the positioning accuracy of the global system, since the corresponding terms dropped out of the error Eq. (11.5) [19, 20].

### 11.2.3.2 Mobile, Arm-Mounted Camera

Consider now an EOL system capable to observe, recognize, and locate objects with a mobile camera mounted on segment  $s$ ,  $3 \leq s \leq n$  of the robot manipulator. The camera was a priori calibrated relative to segment  $s$  of the manipulator by the estimate  $\hat{\mathbf{x}}_{\text{vis}}^s$ , recognizes and locates objects on the image plane at run time by furnishing their pose estimates,  $\hat{\mathbf{x}}_{\text{obj}}^{\text{vis}}$ .

This system is also capable to evaluate at run time the pose of the frame attached to link number  $s$ ,  $\hat{\mathbf{x}}_s^0$ , by a *backward* computational scheme starting from  $\hat{\mathbf{x}}_n^0$  (measured by the joint encoders at the current sampling time instance) and going back through the robot kinematic model. Then, the positioning error for object grasping can be defined by the VKE function:

$$\mathbf{E}(\mathbf{x}_n; \tilde{\mathbf{x}}_{n^*}^{\text{obj}}, \hat{\mathbf{x}}_{\text{obj}}^{\text{vis}}, \hat{\mathbf{x}}_{\text{vis}}^s, \hat{\mathbf{x}}_s^0) = \mathbf{x}_{n^*}^n = \hat{\mathbf{x}}_0^n : \hat{\mathbf{x}}_s^0 : \hat{\mathbf{x}}_{\text{vis}}^s : \hat{\mathbf{x}}_{\text{obj}}^{\text{vis}} : \tilde{\mathbf{x}}_{n^*}^{\text{obj}} \quad (11.6)$$

and the control signal  $\hat{\mathbf{x}}_0^n = \text{inverse}(\hat{\mathbf{x}}_n^0)$  is output to the motion tracking controller in terms of an IK-based RMRC law.

If the visual servo control problem is switched to an ECL architecture, which is also capable to observe the end effector and estimate at run time its pose  $\hat{\mathbf{x}}_n^{\text{vis}} = \text{inverse}(\hat{\mathbf{x}}_{\text{vis}}^n)$ , then the error Eq. (11.6) becomes:

$$\begin{aligned} \mathbf{x}_{n^*}^n &= \hat{\mathbf{x}}_s^n : \hat{\mathbf{x}}_s^0 : \hat{\mathbf{x}}_s^0 : \hat{\mathbf{x}}_{\text{vis}}^s : \hat{\mathbf{x}}_{\text{obj}}^{\text{vis}} : \tilde{\mathbf{x}}_{n^*}^{\text{obj}} \\ &= \hat{\mathbf{x}}_{\text{vis}}^n : \hat{\mathbf{x}}_s^{\text{vis}} : \hat{\mathbf{x}}_{\text{vis}}^s : \hat{\mathbf{x}}_{\text{obj}}^{\text{vis}} : \tilde{\mathbf{x}}_{n^*}^{\text{obj}} = \hat{\mathbf{x}}_{\text{vis}}^n : \hat{\mathbf{x}}_{\text{obj}}^{\text{vis}} : \tilde{\mathbf{x}}_{n^*}^{\text{obj}} \end{aligned} \quad (11.7)$$

As in the previous case of the stationary camera, the positioning accuracy depends only on the quality of the vision estimates of the object and gripper poses.

Despite the evident increase in positioning accuracy, ECL configurations are difficult to implement and more expensive than EOL ones.

### 11.2.4 Estimation of Point and Pose Image Feature Parameters

To speed up the computation of the part pose from image features, actual image features may be used as feedback signal for controlling the manipulator, instead of the part's interpreted location described by the mass center and min. inertia axis.

*Definition 2.* Consider an object  $O$  observable by the stationary camera. Then, the parameters of the following pose-features are calculable [21, 22]:

- *Center of mass*  $\mathbf{C}_o(x_C, y_C)$ , its coordinates are:

$$x_C(O) = m_{10}(O)/m_{00}(O) \quad x_C(O) = m_{01}(O)/m_{00}(O), \quad (11.8)$$

- where  $m_{pq}(O)$  are the moments of  $(p + q)$  order of object  $O$ :

$$m_{pq}(O) = \sum_{x=0}^{N_x} \sum_{y=0}^{N_y} x^p y^q f_o(x, y), \quad (11.9)$$

- $x, y$  are respectively the coordinates along  $x_{\text{vis}}, y_{\text{vis}}$  of the analyzed pixel, belonging to  $O$ ,  $N_x, N_y$  are respectively the max. values for  $x, y$  and

$$f_o(x, y) = \begin{cases} 0, & \text{pixel not belonging to object } O \\ 1, & \text{pixel belonging to object } O \end{cases}. \quad (11.10)$$

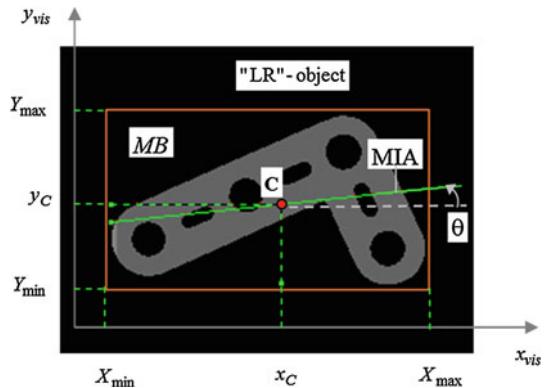
- *Principal orientation axis*  $A_o$ , its angle  $\theta = \angle(A_o, x_{\text{vis}})$ —positive if measured CCW from the vision abscissa  $x_{\text{vis}}$  of the vision frame—is

$$\theta = \begin{cases} \tan^{-1} \frac{\mu_{02} - \mu_{20} \pm \sqrt{(\mu_{02} - \mu_{20})^2 - 4\mu_{11}^2}}{2\mu_{11}}, & A_O \equiv \text{the minimal inertia axis(MIA)} \\ \angle(\mathbf{C}_o \mathbf{F}, x_{\text{vis}}), & A_O \equiv \text{the line from } \mathbf{C}_o \text{ to the furthest} \\ & \text{point on the outer contour} \\ \angle(\mathbf{C}_o \mathbf{N}, x_{\text{vis}}), & A_O \equiv \text{the line from } \mathbf{C}_o \text{ to the nearest} \\ & \text{point on the outer contour} \end{cases} \quad (11.11)$$

- where  $\mu_{pq}$  are the centered moments of  $(p + q)$  order of object  $O$ :

$$\mu_{pq}(O) = \sum_{x=0}^{N_x} \sum_{y=0}^{N_y} (x - x_C)^p (y - y_C)^q f_o(x, y), \quad (11.12)$$

**Fig. 11.7** Pose-image features from the binary image of an “LR”-type object



- Minimum rectangle box  $MB(O)$ , location of O), its parameters are:

$X_{\text{Min}}, X_{\text{Max}} = \text{left, respectively right limit of } MB$ , measured along  $x_{\text{vis}}$   
 $Y_{\text{Min}}, Y_{\text{Max}} = \text{down, respectively up limit of } MB$ , measured along  $y_{\text{vis}}$

(11.13)

Figure 11.7 represents these pose-image features for an “LR”-type object.

Feature selection for control is concerned with which features best relay information about the relative pose of the stationary object with respect to the moving camera. The selection process must not only consider criteria for *image recognition*—such as feature robustness, completeness and uniqueness, but also *control* criteria, such as controllability and sensitivity. The  $(x, y)$  position of a feature such as a *corner* or a *circle* in the image will be referred to as a *feature point*. The selection of reliable image features, providing a good visual control, depends on a combination of weighted image recognition and control criteria. *Image recognition criteria* used in the feature selection process should include:

- rare (or unique) features;
- feature set robustness (or likelihood to be seen);
- computational inexpensive features;
- feature set completeness.

### 11.3 Robot-Object Calibration for Collision-Free Grasping

Robot-object calibration provides the values of the functions which “transform” a shape recognized in the *image scene* into an object-related grasping description, part of the *robot scene*. The evaluation of this set of *functions for scene objects* will

be presented in this section, in the hypothesis that the camera's acquisition axis is perpendicular to the plane of the image scene.

In the frame of conceptual description of image feedback processing for robot control, the set of functions for scene objects is placed at the basic *learning* level of object grasping [23–25]. The set of functions for scene objects result by extending the training sessions which create the object models, and operate on the same class-representing objects to be grasped according to desired robot scene parameters (or “grasping styles”).

### 11.3.1 Using Analytical Methods for Robot-Object Calibration

To calculate the set of robot scene parameters for object grasping, off-line training schemes use interactively pose- and point-image features (center of mass and orientation in the object's image) and *robot points* learned in Cartesian space.

The set of robot scene parameters which allow task-imposed object grasping apply usually for those classes of a priori modeled parts which are recorded in the knowledge database of the robot-vision system.

Robot-object training schemes are based on analytical methods which associate pose-image features expressed in the vision frame,  $\mathbf{x}_{\text{obj}}^{\text{vis}}$ , to the Cartesian desired poses of the gripper relative to the object,  $\mathbf{x}_n^{\text{obj}}$  ( $n$  is the number of DOF), using the previously learned coordinate transformation between the image frame  $(x_{\text{vis}}, y_{\text{vis}})$  and the robot base frame  $(x_0, y_0, z_0)$ .

Training robot scene parameters for grasping is a repetitive process performed, for each class of objects of interest, by placing the class-representing part in the field of view of the camera for as many times as the number of relative gripper-object grasping locations that must be learned for that class.

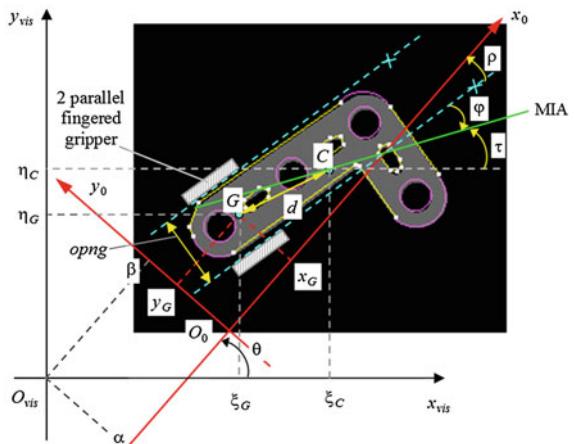
The following four robot scene parameters completely specify the relative gripper-object grasping pose:

- the grasping “height”,  $ht$ ;
- the gripper's opening,  $opng$ ;
- the  $xy$  offset,  $d$ , of the grasping point  $G = \text{proj}|_{(x_{\text{vis}}, y_{\text{vis}})}\{T\}$  relative to the center of mass  $C$  of the object's image ( $T$ -origin of  $(x_n, y_n, z_n)$ );
- the roll angle of the gripper,  $\varphi$ , relative to an orientation axis of the object (in the sequel the minimum inertia axis (MIA) of the object's image).

These four grasping parameters ( $ht, opng, d, \varphi$ ) are particular for each class of objects and “grasping style” and will be therefore learned and stored in related class records. Three of these parameters:  $opng, d$  and  $\varphi$  are indicated in Fig. 11.8, which shows the projection of the gripper's fingers in the object-relative grasping location, onto the image plane.

Two orthogonal frames are considered:  $(x_{\text{vis}}O_{\text{vis}}y_{\text{vis}})$  representing the vision frame which defines the image plane, and  $(x_0O_0y_0)$  representing the base frame of

**Fig. 11.8** Robot scene parameters  $opng, d, \phi$  for a projected grasping location



the robot. It is assumed that the robot is table-top mounted, and a zone of the table is visualized by the stationary camera, hence there is no Z offset between the two above frames ( $Z|_{\text{to.cam[cam]}} = 0$  for the virtual camera cam in use). The mathematical relationships, describing the coordinate transformations for some point  $i$  from the vision frame  $(x_{\text{vis}}, y_{\text{vis}})$  to the robot frame  $(x_0, y_0, z_0)$ , are:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} \xi_i \\ \eta_i \end{bmatrix} + \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \quad (11.14)$$

The five parameters  $(\alpha, \beta, \theta, a, b)$  which establish this transformation have been a priori estimated in the *camera–robot calibration* sequence, for a predefined association between the physical camera in use and one specified virtual camera, cam, and correspond to the values:

- $\alpha = X|_{\text{to.cam[cam]}}$ , the *x\_offset* between the image frame and the robot frame;
- $\beta = Y|_{\text{to.cam[cam]}}$ , the *y\_offset* between the image frame and the robot frame;
- $\theta = \text{rz}|_{\text{to.cam[cam]}}$ , the *roll\_offset* between the image frame and the robot frame;
- $a = \text{pix.to.mm}$ , the ratio between the pixel's dimension along the *X* axis and one millimeter;
- $b = \text{pix.to.mm/XY\_scale}$ , the ratio between the pixel's dimension along the *Y* axis and one millimeter (the value of *XY\_scale*, produced by the camera–robot calibration program, indicates how “square” is the pixel).

The object in the image plane has a center of mass  $C$  of known coordinates  $\xi_C, \eta_C$  and one reference axis (MIA) turned with respect to the abscissa  $x_{\text{vis}}$  of the vision frame with a known angle  $\tau$ . The values of  $\xi_C, \eta_C$  and  $\tau$  are computed at run time by the vision processor after successfully locating the object in the image frame  $(x_{\text{vis}}, y_{\text{vis}})$ .

Let  $G$  be the projection of gripper's controlled end point  $T$  onto the image plane; its coordinates expressed in the robot frame  $(x_0 O_0 y_0)$  are  $x_G, y_G$ . Using the

inverse coordinates transformation (robot to camera), it is possible to calculate  $\xi_C$  and  $\eta_C$ —the coordinates of  $G$  in the vision frame:

$$\begin{bmatrix} \xi_G \\ \eta_G \end{bmatrix} = \begin{bmatrix} 1/a & 0 \\ 0 & 1/b \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \xi_i \\ \eta_i \end{bmatrix} + \begin{bmatrix} x_C - \alpha \\ x_C - \beta \end{bmatrix} \quad (11.15)$$

The angles  $\rho$  and  $\phi$  measure the roll rotation of the gripper relative to the  $x_0$  axis of the robot's reference system, respectively relative to the object's MIA.

The following algebraic relation holds between these two angles:

$$\phi = \rho + \theta - \tau \quad (11.16)$$

It is to be noted that the angle  $\phi$  is a specific constant for one class of objects and associated grasping style, and will be thus stored in the database of this class. The angle  $\theta$  is also a specific constant for the calibration of the robot with respect to the association between the physical camera and one predefined virtual camera. Consequently, the result of Eq. (11.16) can be interpreted in two different ways:

- $\varphi$  is the computed value *during the robot-object calibration session*; it is obtained from: a priori known  $\theta, \tau$  obtained by locating the class object in the image plane and  $\rho$  measured by the joint encoders in the Cartesian desired grasping location [26];
- $\rho$  is the computed value *at run time*, to orient the gripper with the desired angle  $\varphi$  relative to the object's MIA—currently located in the image plane (with angle  $\tau$ ) by help of the a priori known  $\theta$ .

It is also possible to calculate the distance  $d$  between the object's center of mass and the projection of the origin of the gripper frame  $(x_n, y_n, z_n)$  onto the image plane:

$$d^2 = (\xi_G - \xi_C)^2 + (\eta_G - \eta_C)^2 \quad (11.17)$$

Taking into account the angle formed by the direction  $CG$  with the minimal inertia axis of the object, it is:

$$\cos(\rho + \tau) = (\xi_G - \xi_C)/d, \quad \sin(\rho + \tau) = (\eta_G - \eta_C)/d, \quad (11.18)$$

and, by trigonometry development:

$$\begin{aligned} \xi_G - \xi_C &= d \cos \varphi \cos \tau - d \sin \varphi \sin \tau \\ \eta_G - \eta_C &= d \cos \varphi * \sin \tau - d \sin \varphi * \cos \tau \end{aligned} \quad (11.19)$$

If the system composed of the two equations (11.19) is solved in the two unknowns  $d \cos \varphi$  and  $d \sin \varphi$ , then the values of  $\Gamma_1, \Gamma_2$  are obtained, completely describing the location of the gripper point  $G$  with respect to the center of mass  $C$  of the object—which is located in the image plane by  $\xi_C, \eta_C$  and  $\tau$ .

$$\begin{aligned}\Gamma_1 = d \cos \varphi &= (\xi_G - \xi_C) \cos \tau + (\eta_G - \eta_C) \sin \tau \\ \Gamma_2 = d \sin \varphi &= (\xi_G - \xi_C) \sin \tau + (\eta_G - \eta_C) \cos \tau\end{aligned}\quad (11.20)$$

Thus, the values of  $\Gamma_1, \Gamma_2, \varphi$  completely identify the position and orientation of the robot's gripper relative to the object, located with  $\xi_C, \eta_C$  and  $\tau$  in the image plane at run time. For this reason,  $\Gamma_1, \Gamma_2$  and  $\varphi$  will be stored after robot-object calibration in the database of each class of objects of interest.

For an object visually recognized and located at run time by its center of mass  $C(x_C, y_C)$  and  $MIA(\tau)$  relative to the vision frame  $(x_{vis}, y_{vis})$ , the location of the gripper in the grasping position—expressed in the robot frame  $(x_0, y_0, z_0)$ —can be obtained using the equations:

$$\begin{aligned}\rho &= \varphi + \tau \theta \\ \begin{bmatrix} \xi_G \\ \eta_G \end{bmatrix} &= \begin{bmatrix} \xi_C \\ \eta_C \end{bmatrix} \begin{bmatrix} \cos \tau & -\sin \tau \\ \sin \tau & \cos \tau \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix} \\ \begin{bmatrix} x_G \\ y_G \end{bmatrix} &= \begin{bmatrix} a \cos \theta & b \sin \theta \\ a \sin \theta & b \cos \theta \end{bmatrix} \begin{bmatrix} \xi_G \\ \eta_G \end{bmatrix} + \begin{bmatrix} \alpha \\ \beta \end{bmatrix}\end{aligned}\quad (11.21)$$

The values of  $x_G, y_G$  and  $\rho$  resulting from Eq. (11.21) and the learned values of  $ht$  and  $opng$ , together with the real-time object locating algorithm producing  $x_C, y_C$  and  $\tau$  from image feature extraction, provide at run time the Cartesian coordinates of the gripper location which will allow the robot to grasp the currently recognized object according to its learned scene parameters.

Analytical robot-object calibration methods like the above presented one are interactively implemented during off-line training sessions, which alternatively use the image processing resources—usually by sequences of monitor commands—and the robot manipulator itself (its set of joint encoders and the embedded direct kinematics model)—as a Cartesian pose-measuring device [27].

A standard training session executing the functions for scene objects provides the set of robot scene parameters for object grasping,  $d, \varphi, ht, opng$ , in a sequence of steps:

1. The pattern database containing the a priori trained models for the classes of objects to be visually recognized and located, grasped and manipulated at run time by the robot, is downloaded from disk.
2. Objects representing the classes of interest will be placed in the field of view of the camera, one sample for each class. Then, for each object visible in the scene, the robot will be moved in manual mode until the desired grasping location will be reached. These Cartesian robot points will be saved without modifying the gripper's opening (the object must be held in the gripper, during all this time). For each robot point related to object grasping, the particular hand opening is added.
3. The object is released from the gripper, the robot arm is moved into a safe location outside the camera's field of view, and where it does not obstruct the image of the previously grasped object.

4. An image acquisition is performed for all the class objects in the camera's field of view.
5. The last grasped and released object is retrieved from the vision queue containing all class objects; it is recognized and located in the image frame. For each located object, the name of the class to which it belongs must be specified, as well as the ID number of its latest trained grasping point. Multiple grasping poses for one class of objects are permitted; a run time search can be initiated after locating an object of interest, to visually identify and authorize only collision-free grasping poses [28, 29].
6. The robot scene parameters  $\varphi, \Gamma_1, \Gamma_2, ht, opng$  for grasping the currently retrieved object are obtained from the following computing sequences:
  - $\theta$  is read from the camera-to-robot calibration transformation, `to.cam[cam]`;
  - $ht, \varphi, opng$ , and  $\xi_G, \eta_G$  are extracted from the last recorded robot point, by relating it to the vision frame;
  - $\xi_C, \eta_C$  and  $\tau$  result from visual object locating (embedded in the vision software);
  - $\varphi, d$  and  $\Gamma_1, \Gamma_2$  are computed respectively from Eqs. (11.16), (11.17), and (11.20).

Upon user request, the robot-object calibration parameters may be displayed for each class object. They are finally stored on disk for future usage with the trained object classes. It is preferable to *learn off-line* robot grasping poses relative to the object-attached frames, rather than *automatically compute* them *at run time* when executing the application program [30, 31], because this significantly increases the processing load and slow down the “look-and-move” pipelining.

### **11.3.2 Learning Robot Scene Parameters from Image Data**

Feature tracking for object grasping requires the a priori learning of robot scene parameters for those classes of objects which will be picked and manipulated by the robot at run time. These parameters depend on several elements:

- The shape of the object recognized in the image scene.
- The relative location of the gripper's fingers relative to the object, irrespective of the object's placement in the image scene—the *grasping style model*  $Gs\_m$ .
- The a priori knowledge about the risk of collision between the gripper and the image scene foreground, i.e., the estimation of the “scene context” at run time:
  - invariant or random appearance of part types in the image scene;
  - constrained (in position and/or orientation) or randomly located parts in the image frame  $(x_{vis}, y_{vis})$ ;

- risk of parts overlapping, touching, or close positioned one with respect to the others in the image scene.

The parameters of a grasping transformation  $\mathbf{x}_n^{\text{obj}}$  for a given class of objects are hence the components of the grasping style model, off-line trained. They map at run time the shape of an object recognized and located in the image scene into an object-related grasping description, part of the robot scene.

As shown in Fig. 11.1, there are four coordinate transformations to be composed at run time to provide the gripper's reference point  $\text{part.loc} = \mathbf{x}_n^0$  expressed in the robot base frame coordinates  $x_0, y_0, z_0$ : (1)  $\text{to.cam}[\text{cam}] = \mathbf{x}_{\text{vis}}^0$ , the calibration transformation; (2)  $\text{vis.loc} = \mathbf{x}_{\text{obj}}^{\text{vis}}$ , the object pose computed at run time from object features extracted from image data: the coordinates  $x_C, y_C$  of the object's center of mass C, and the angle  $\angle(\text{MIA}, x_{\text{vis}})$  between the vision plane abscissa and the object's minimum inertia axis; (3)  $\text{grip.trans} = \mathbf{x}_n^{\text{obj}}$ , the grasping transformation a priori learned for the recognized object O and stored as  $\mathcal{G}_s\_m(O)$  in the memory; (4) TOOL, the known gripper pose relative to the TMF.

Usually there are two ways to express the  $\mathcal{G}_s\_m(O)$  model, for an object O and learned grasping style, function of the object-related image data [C,MIA (O)]:

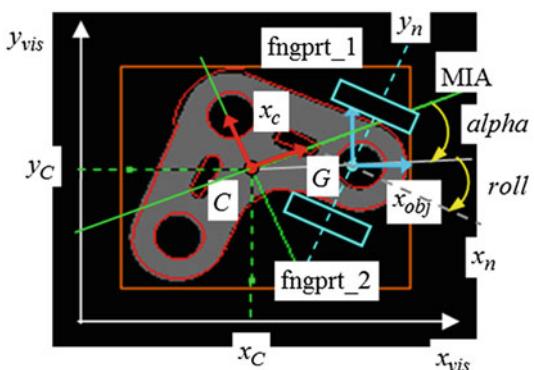
1.  $\mathcal{G}_s\_m = \{d.cg, alpha, z\_off, rz\_off\}$ , where (see Fig. 11.9):

- $d.cg = \text{dist}(C, G)$ ,  $alpha = \angle(\text{MIA}, \overline{CG})$ , with  $G = \text{proj}|_{(x_{\text{vis}}, y_{\text{vis}})}\{\mathbf{T}\}$ , T is the end-tip point of the robot and  $d.cg \neq 0, alpha \neq 0$  for general grasping;
- $z\_off = \text{dist}(T, G) \neq 0, rz\_off = \angle(\overline{CG}, x_n)$  account for the object's grasp height and imposed gripper roll angle.

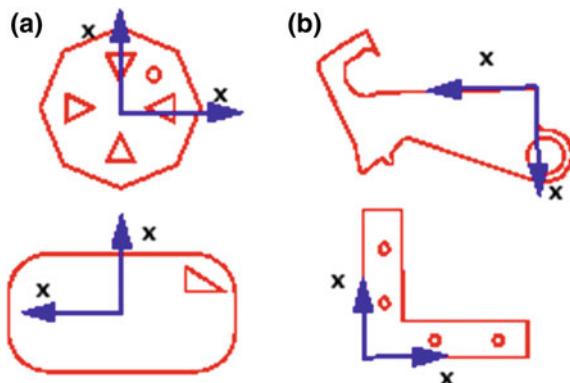
2.  $\mathcal{G}_s\_m = \{x\_off, y\_off, rz\_off, z\_off\}$ , where:

- $x\_off, y\_off$  are the position offsets of the user-defined object coordinate system  $(x_{\text{obj}}, y_{\text{obj}})$  which will be automatically attached by vision at run time to a recognized instance of O- relative to the object's center of mass;

**Fig. 11.9** Object-attached frames for grasp style  
 $\mathcal{G}_s\_m = \{d.cg, alpha, z\_off, rz\_off\}$



**Fig. 11.10** Default placement of the object coordinate system (a) user-defined placement of the object frame depending on object features and robot tasks (b). Reference Adept Technology Inc.



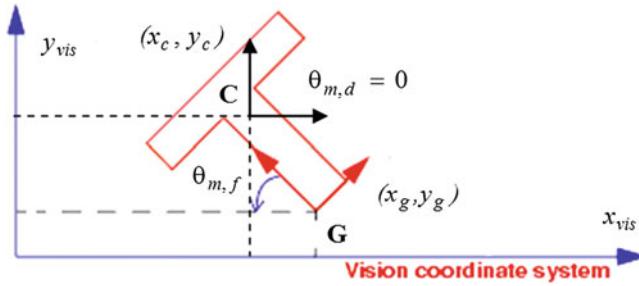
- $r_{z\_off}$  is the roll offset angle  $\angle(x_{vis}, x_{obj}) = \angle(x_{vis}, x_n)$  of the user-defined object frame relative to  $x_{vis}$ , which coincides with the roll orientation of the gripper in the object grasping location;
- $z\_off$  is the value offsetting the gripper along  $z^+$  from the image plane; if the object coordinate system's orientation is trained with  $r_{z\_off}$  such that  $x_{obj} \parallel x_n$  for a desired grasping style, and if  $x\_off, y\_off$  and  $r_{z\_off}$  are automatically applied at run time to the recognized instance of class object  $O$ , located from image data [C, MIA], it results that  $z\_off$  is the only component (always nonzero) of the grasp transformation  $grip.trans$  to be composed at run time with the camera-robot calibration transformation and the current object pose (returned by vision) for effective robot access to the part to be grasped.

The position and orientation of the object (attached) coordinate system can be thus specified by the user anywhere within the limits of the image during the training of the  $Gs\_m$  model. Its origin  $G(x_g, y_g)$  may be offset relative to the centre of mass  $C(x_c, y_c)$  anywhere, at training time—not necessarily on the object's contour or body (Fig. 11.10).

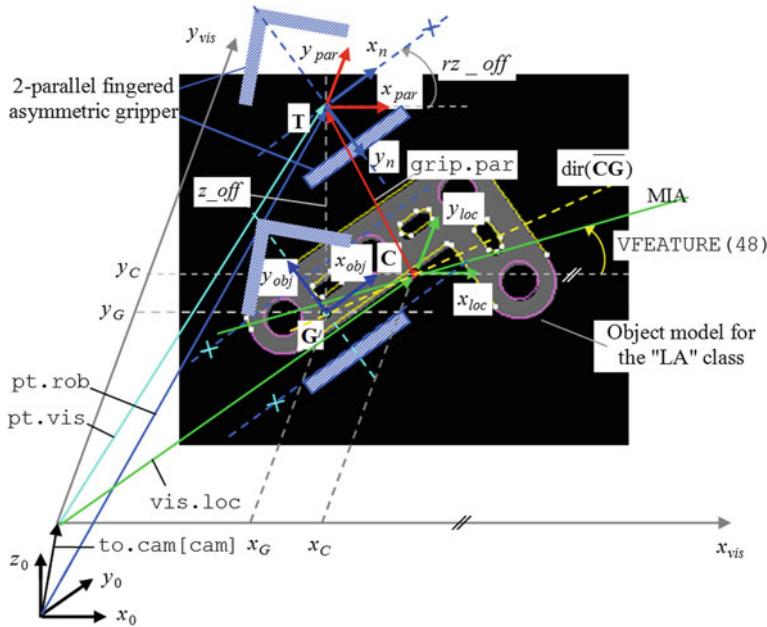
Figure 11.11 shows how changing at training time the *default* location of the object's coordinate system from  $(x_{m,d}, y_{m,d}) = (x_c, y_c), \theta_{m,d} = 0$  to a feature-related one  $(x_{m,f}, y_{m,f}) = (x_g, y_g), \theta_{m,f}$ , is performed using the  $Gs\_m$  model parameters:  $x_{m,f} = x_{m,d} + x\_off$ ,  $y_{m,f} = y_{m,d} + y\_off$ ,  $\theta_{m,f} = \angle(x_{vis}, x_n)$ .

At run time, during object search (recognition and locating in  $(x_{vis}, y_{vis})$ ), the position  $x_v, y_v$  and orientation  $\theta_v$  of an object in the input image will correspond respectively to the position and orientation of the object frame—shifted by vision at run time from [C, MIA] with the a priori trained values  $x\_off, y\_off, r_{z\_off}$ .

Learning the robot-object scene parameters  $x\_off, y\_off, r_{z\_off}$  (the  $x, y, roll$  offsets of the object frame relative to its center of mass and MIA) and  $z\_off$  (the  $z$  offset along  $z^+$  of the grasping transformation) of the  $Gs\_m$  model can be done in an interactive session which uses the robot itself as a measuring device (encoder data and features extracted from image data) and the user-defined numeric



**Fig. 11.11** Changing the object coordinate system when training the  $\mathcal{G}_s\text{-}_m$  model (example the ObjectFinder grasping model from Adept Technology Inc.)



**Fig. 11.12** Learning in  $V^+$  the location offsets  $x\_off$ ,  $y\_off$ ,  $rz\_off$  of the object-attached frame and the component  $rz\_off$  of the grasping transformation on the “LA” model

computation (Fig. 11.12). An object was modeled with the name “LA”; the learning procedure is exemplified for the  $V^+$  robot programming environment:

1. Placing the model in the camera’s field of view. Because the object class “LA” is not yet modeled, the unique object in the image plane is a *blob*.
2. Using the teach box, the robot is manually moved with open gripper in the desired grasping location of the blob.

3. DO CLOSEI  
DO OPENI  
DO CLOSEI; The gripper is actuated to eliminate part dragging.
4. HERE pt.rob: The current robot location is stored in pt.rob, which expresses the position and orientation offsets of the gripper-attached frame ( $x_4, y_4, z_4$ ) with respect to the robot base frame ( $x_0, y_0, z_0$ ).
5. The gripper is opened, and the robot is moved outside the camera's field of view from the system's terminal (a PC) in an a priori learned point, safe:

```
DO OPENI
DO DEPARTS 100
DO MOVE safe
```

6. Computing the location of the gripper's frame ( $x_4, y_4, z_4$ ) relative to the vision frame ( $x_{\text{vis}}, y_{\text{vis}}$ ). The camera with the ID cam is used; the result is assigned to the transformation variable pt.vis:

```
DO SET to.cam[cam]:pt.vis=pt.rob
```

7. Accessing the components of pt.vis:

```
DO DECOMPOSE arr.pt []=pt.vis
```

8. Visualizing the blob and locating it in the image plane, without having moved it in the scene meanwhile:

```
VPICTURE (1, -1) -1,1
```

DO VLOCATE (1, 2) ``?``, vis.loc Because a blob was detected and located, its vision location returned in the variable vis.loc has the  $x, y$  coordinates of the blob's center of mass  $\mathbf{C}(x_C, y_C)$  and a null orientation,  $rz = 0$ , i.e.  $\angle(x_{\text{loc}}, x_{\text{vis}}) = 0$ .

9. Accessing the components of vis.loc:

```
DO DECOMPOSE arr.c []=vis.loc
```

10. Computing the  $xy$  offsets of the user-defined model reference frame relative to the default reference frame of the model, based on its center of mass:

```
DO x_off=arr.pt[0]-arr.c[0]
DO y_off=arr.pt[1]-arr.c[1]
```

11. Computing the  $z^+$  offset as nonzero parameter of the grasp transformation to be added at run time for recognized model instances:

```
DO z_off=arr.pt[2]
```

12. Computation of the *roll* offset angle  $rz\_off$  of the user-defined model reference frame. A relative transformation grip.par is defined for the actual location of the blob, expressing the offsets of the frame ( $x_{\text{par}}, y_{\text{par}}$ ) attached in the

gripper's terminal point **T**, and having its axes parallel to those of the vision frame  $(x_{\text{vis}}, y_{\text{vis}})$ , i.e.  $x_{\text{par}} \parallel x_{\text{vis}}, y_{\text{par}} \parallel y_{\text{vis}}$ :

```
DO SET grip.par=TRANS(x_off,y_off,z_off,0,0,0)
```

The roll offset  $rz\_off$  is evaluated from:

```
DO SET to.cam[cam]:vis.loc:grip.par:rot_rz=pt.rob
DO DECOMPOSE arr.roll[] = rot_rz
DO rz_off=arr.roll[5]
```

It can be observed from the graphics in Fig. 11.12 that  $x_{\text{off}} = x_G - x_C$ ,  $y_{\text{off}} = y_G - y_C$  and  $rz_{\text{off}} = \angle(x_4, x_{\text{vis}})$ ; they will be input as “xy – roll” offsets in the mode 5 VTRAIN.FINDER training operation of V<sup>+</sup>, and stored. At run time, the system will use them to automatically shift and rotate the instance-attached frame from the default model frame (in C and with zero rotation).

As for the  $z^+$  offset value  $z_{\text{off}}$ , it allows defining the grasping transformation to be applied at run time for accessing any recognized and located “LA” instance:

```
DO SET grip.la=TRANS(0,0,z_off,0,180,0)
```

Between the moments of time the four  $G_s_m$  parameters were learned and the effective training of the new reference frame of the model, via a model training software operation (here the V<sup>+</sup> multi-operation VTRAIN.FINDER in mode 5 was referred), the model should be not displaced (rotated or shifted in the visualized scene). This is due to the fact that the instance reference frame will be defined by the robot-vision system at run time by automatically updating the learned xy – roll offset values and taking into account the model's orientation at training time (in the case of V<sup>+</sup> it is: VFEATURE(48) =  $\angle(MIA_{\text{model}}, x_{\text{vis}})$ ).

## 11.4 Guidance Vision for Robot Grasping Tasks

This last section puts together the previously introduced principles and schemes for the visual servo control of robot grasping, and presents the way in which the trained robot-object scene parameters are used at run time with the pose-image feature data to control the structured search, tracking, and picking of objects [68].

Real-time mechanisms are discussed for two camera configurations: stationary (or *fixed-mount*) and mobile (or *arm-mounted*). The guidance vision principle will be exemplified for SCARA-type robot manipulators and the V+ environment.

### 11.4.1 Guiding Robot Grasping from Stationary Cameras

A *fixed-mount* (or *stationary*) camera is any camera that acquires images at a fixed location in the robot workspace. Before a camera can be used for inspection or guidance vision, it must be calibrated and the calibration data transferred into the vision system. One extrinsic parameter of the camera-to robot calibration relates through an *invariant* transformation (denoted here to.cam[cam]), the fixed vision frame ( $x_{\text{vis}}, y_{\text{vis}}$ ) to the world coordinate system ( $x_0, y_0, z_0$ ) in the robot's base.

Once the robot-object scene parameters *trained* respectively in the recognition and grasping models  $\mathcal{F}_{s\_m}, \mathcal{G}_{s\_m}$  and the object search mode *planned*, the visual instance *search* and robot *motion control* for grasping is implemented according to a real-time scheme exploiting the structuring facilities of the robot programming languages. For a fixed scene and stationary camera, the software operation for recognizing model instances will be executed a single time, considering the search planning associated with the camera previously configured for object recognition and locating (i.e., VFINDER in V+ robot-vision programs).

The real-time instance search and locating scheme for visually servoing the robot in pick and manipulate tasks contains the following steps:

1. Various setups, counter initializing including those for the objects of interest (to be grasped and handled by the robot).
2. Acquisition of a single image, with VPICTURE in mode 2 (acquisition without processing), which provides the grayscale input image data for the entire scene.
3. Recognizing all model instances, and detecting all blobs in the scene with the VFINDER operation, according to the ObjectFinder functionality.
4. DO (begin of the search loop for objects of interest)
5. Extracting with VLOCATE the pose-image feature data for the currently located instance in the vision queue, in the desired order. The extraction order may be: (a) *any* (the gripper may be at most brought always from the same direction to access objects) or (b) in a *particular type order* (when for each VLOCATE the name of the current type of object to be accessed must be specified). The transformation variable `loc_var` will be assigned the position and orientation of the currently located model instance. Several levels of DO loops may be nested in the top level one, if the *find-particular* mode is imposed by the robotic task. For example, upon locating and extracting each one of  $n$  assembly supports, in any or particular order, the current assembly support object  $k$ ,  $1 \leq k \leq n$  needs respectively  $k_1, \dots, k_{nk}$  components— $nk$  types of other objects to be presented in the particular order  $Ordre(k)$  for mounting. Such a task imposes at least three nested loop levels.
6. After each VLOCATE (object locating) operation, its validity will be checked (in V+ with the function VFEATURE(1)), and the result used to condition either the inspection or picking and handling of the current object in the queue of segmented objects (object images extracted from the background).

7. If VFEATURE(1) ==TRUE:
8. The access in the program is permitted to valid VFEATURE data describing the model instance. For example, in V+ the position and orientation of the instance-attached frame are respectively given by:  

```
VFEATURE(2)=DX(loc_var)
VFEATURE(3)=DY(loc_var)
VFEATURE(7)=RZ(loc_var)
```
9. The model instance currently located is further inspected, grasped and manipulated, or processed with an end-effector tool.
10. END No action will be carried upon the current, not located object.
11. UNTIL ... terminating condition for the robotic task.
12. Display of messages, counter values, and so on.

The following V+ program commands the robot to grasp from the scene objects named “LA”, for which a recognition and locating model ,  $\mathcal{F}_{s\_m}, \mathcal{G}_{s\_m}$ , with  $\mathcal{G}_{s\_m} = \{x_{\text{off}}.\text{la}, y_{\text{off}}.\text{la}, rz_{\text{off}}.\text{la}, z_{\text{off}}.\text{la}\}$  were a priori created. Parts must be extracted from the scene always from left to right, and palletized in a 2D stack along  $x_0^+, y_0^+$ , with lin\_la lines parallel to  $x_0^+$  and col\_la columns parallel to  $y_0^+$ ; the offsets in the stack along  $x_0^+, y_0^+$  are respectively dx\_la and dy\_la, and the point pal\_la was a priori learned for the first stack’s location:

```
.PROGRAM extract_LA()
;... Various settings...
VPICTURE(cam.virt,-1,virt.fr+phy.fr,virt.fr+phy.fr) 2
VFINDER(cam,1,-1,-1) ibr_rect
VWAIT;wait for operation to complete
total=VQUEUE(''LA'',cam)
n_la=0; number of located objects
FOR i=1 TO total
  VLOCATE(cam,2,3) ''LA''
  IF VFEATURE(1) THEN
    SET grip=TRANS(0,0,z_off.la,0,180,0)
    col=n_la MOD col_la
    lin=INT(n_la/col_la)
    SET dest=SHIFT(pal_la BY col*dx_la,lin*dy_la,0)
    CALL pick_place()
    n_la=n_la+1
  END
END
.END
.PROGRAM pick_place()
;... Set up speed, parameter hand.time, ...
SET part.loc=to.cam[cam]:vis.obj:grip
APPRO part.loc, dec_z;approach current pick location
```

```

BREAK
SPEED reduced_value
MOVES part.loc
CLOSEI
DEPARTS dec_z
APPRO dest, dec_z; approach current place location
BREAK
SPEED reduced_value
MOVES dest
OPENI
DEPARTS dec_z
RETURN
.END

```

### 11.4.2 Guiding Robot Grasping from Mobile Cameras

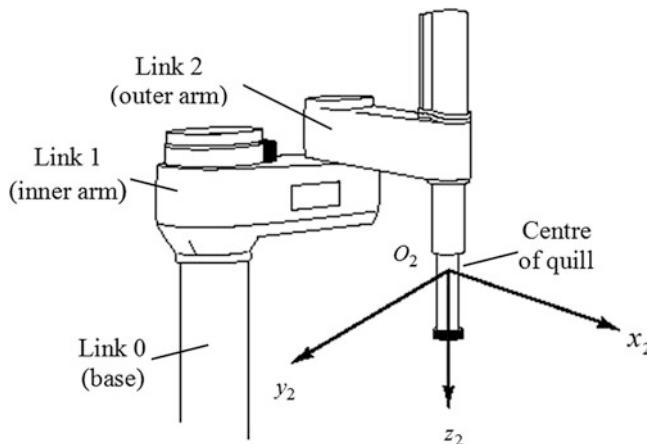
This section describes how to track and locate object from mobile cameras in one of the most common mountings: cameras mounted on the second link of SCARA, XY, XYZ, or XYZ-theta type robots. The methods and implementing solutions introduced in this section can be easily extended to define guided vision application to robotic tasks with the camera mounted on any segment of a manipulator [8, 32].

#### 11.4.2.1 Reusing the Camera-Robot Model for Multiple Picture-Taking Points

When a robot-mounted camera is used in guidance vision, the camera-to-robot calibration transformation defines the location of the vision coordinate system,  $(x_{\text{vis}}, y_{\text{vis}})$ , relative to the coordinate frame  $(x_s, y_s, z_s)$ ,  $2 \leq s \leq n$  of the segment  $s$  of the robot manipulator on which the camera is mounted. The case of cameras mounted on the outer link of SCARA robots will be discussed. Figure 11.13 shows the coordinate frame  $(x_2, y_2, z_2)$ , attached to link 2, which is the *outer* arm segment of the SCARA manipulator.

The coordinate system for link 2 is defined as follows:

- The  $z_2^+$  axis is normal to the base plane  $(x_0, y_0)$  and points down, i.e.,  $z_2^+$  is aligned with the axis of motion (sliding) of the 3<sup>rd</sup> joint ( $s + 1 = 3$ )
- The origin  $O_2$  is located where axis  $z_2$  intersects the common normal to the axes  $z_1, z_2$ , i.e.,  $O_2$  is fixed relative to the outer link, at the center of the quill, at the height of the quill flange when joint 3 is at its zero position.
- The  $x_2^+$  axis is normal both to  $z_1$  and  $z_2$  pointing away from joint 2, as if it is an extension of the outer link.



**Fig. 11.13** The coordinate frame  $(x_2, y_2, z_2)$  attached to link 2 of a SCARA robot

- The  $y_2^+$  axis is assigned as  $y_2 = (z_2 \times x_2) / \|z_2 \times x_2\|$  (right-handed frame).

Also, the world reference frame is the same as the frame of the base (link 0) of the SCARA robot. Its origin is on the axis of joint 1 at the level of the robot base.

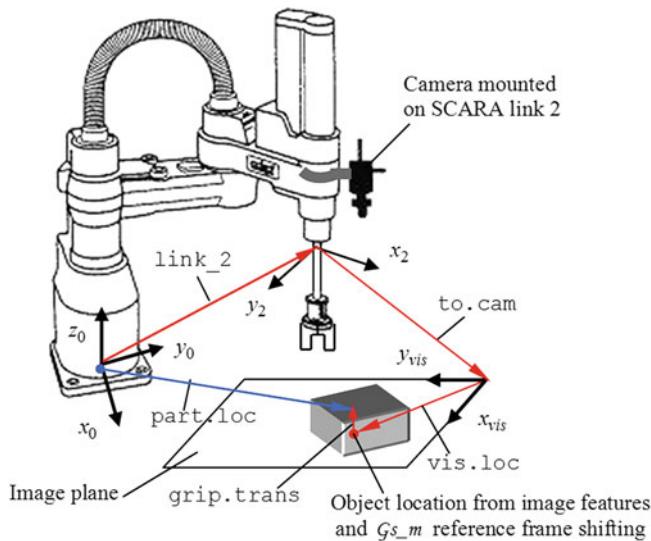
The camera being mounted on link 2 the outer segment of the SCARA robot, it can be moved around in both X and Y directions parallel to the work surface. It is assumed that the camera is mounted such that it is pointing vertically downward, and it cannot make contact with the robot outer link or equipment attached. When mounting the camera on link 2 of the robot arm, make sure the robot can change configuration (LEFTY/RIGHTY) without damaging the camera.

The camera-to-robot calibration transformation, learned off line, represents the transformation relating the vision frame  $(x_{\text{vis}}, y_{\text{vis}})$  to the coordinate system  $(x_2, y_2, z_2)$  attached to the robot link the camera is mounted on. In order to use location information from the vision system, one must know the relationship between the link holding the camera and the robot world coordinate system.

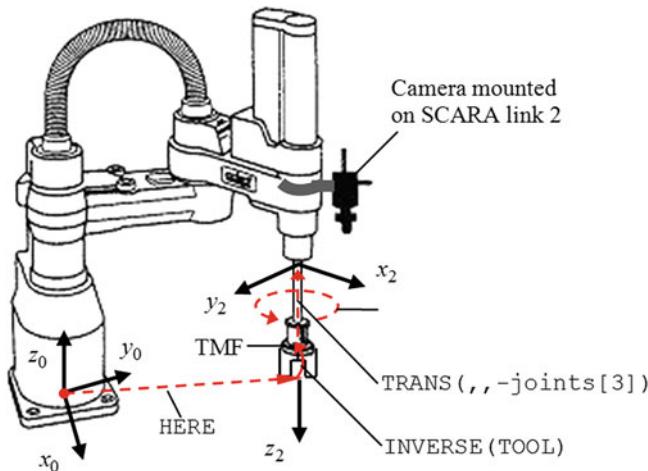
Since the camera-to-robot calibration transformation was created based on the link 2 coordinates frame, any time this transformation to.cam[cam] is used at run time, it must be applied to the “link 2” coordinate frame. Thus, to reuse the one time-learned camera-to-robot calibration transformation for any picture-taking point, the current value of the pose link\_2 must be computed *backwards* from the robot’s end-effector location when the image is acquired [4].

Each time a picture is taken at a new location, a real-time computation, based on the operations above, will have to be executed (see Fig. 11.14):

1. The function HERE returns a transformation value representing the current location of the robot tool tip point. This location is evaluated at run time by reading the instantaneous value of the joint encoders in the picture-taking point, after letting the camera settle for some short time.



**Fig. 11.14** The scheme for calculating the link\_2 relative transformation backwards from the current end-effector location (four-axis SCARA vision transformation)



**Fig. 11.15** Components of the vision location for object tracking with arm-mounted camera

2. INVERSE(TOOL) removes any TOOL transformation, installed for the end effector in use, from the resulting transformation in the picture-taking point.
3. RZ(-joints[4]) removes any rotation of joint 4 from the resulting transformation.
4. TRANS(,, -joints[3]) removes any quill extension from the resulting transformation.

Operations 3 and 4 need the current 3rd and 4th robot joint positions in the picture-taking point. There are two solutions to this problem:

- A. Create a precision point at the picture-taking point and access its 3rd and 4th components:

```
HERE #pict_point
DECOMPOSE joints[1]=#pict_loc
;The current quill extension and roll angle are the
; 3rd and 4th elements respectively in the array
; ''joints[]''.
```

- B. Directly record the current robot joint positions in the picture-taking point. To this purpose, the next program instruction can be used:

```
JHERE,,j3,j4
;The values of j3,j4 are respectively the quill
;extension and roll angle in the picture point.
```

The complete computation of the link\_2 transformation in the picture-taking point, based on the backwards removing procedure, will be executed at run time with the following command:

```
SET link_2=HERE:INVERSE(TOOL):RZ(-j4):TRANS(,,-j3)
```

#### 11.4.2.2 Tracking Parts for Grasping with Arm-Mounted Cameras

Figure 11.15 shows all components of the vision location that must be joined together at run time to guide the robot for object tracking and grasping, from any picture-taking point image feature data.

link\_2 is calculated as above presented, using the backwards procedure which differs for 4-axis and 5-axis SCARA robots;

to.cam[cam] is once created during camera-to-robot calibration and reused to relate the vision coordinate system to the robot base frame;

vis.loc—returned by the vision operation VLOCATE for blob or model instance locating—is added to create a relative transformation representing the location of the found object in the vision coordinate system ( $x_{vis}, y_{vis}$ ); This location is evaluated by the vision system in terms of pose-image features indicating the position and orientation of the object-attached reference frame ( $x_{obj}, y_{obj}$ );

grip.trans is added to create a transformation that offsets the gripper only along  $z^+$  if the found *model instance* is to be grasped at a location offset and/or rotated from the location returned by the vision operation.

### 11.4.3 Authorizing Collision-Free Grasping By Fingerprint Modelling

Random robot scene foregrounds, as the conveyor belt, may need to be faced in robotic tasks. Depending on the parts shape and on their dimension along  $z^+$ , grasping models  $Gs\_m$  are off-line trained for object models object. However, if there is the slightest uncertainty about the risk of collision between the gripper and parts on the belt-touching or close one relative to the other-, *extended* grasping models  $\mathcal{EG}\_m = \{Gs\_m, \mathcal{FGP}\_m\}$  must be created by adding the gripper's *fingerprint model*  $\mathcal{FGP}\_m$  to effectively authorize part access at run time only after *clear grip tests*.

*Definition.*  $\mathcal{M}\mathcal{FGP}\_m(\mathcal{G}, O) = \{\mathcal{FGP}\_m_1(\mathcal{G}, O), \dots, \mathcal{FGP}\_m_k(\mathcal{G}, O)\}$  is defined as *multiple fingerprint model* for a  $p$ -fingered gripper  $\mathcal{G}$  and a class of objects  $O$ , describing the *shape*, *location*, and *interpretation* of  $k$  sets of  $p$  projections of the gripper's fingerprints onto the image plane  $x_{vis}, y_{vis}$  for the corresponding  $k$  grasping styles  $Gs\_m_i, i = 1, \dots, k$  of  $O$ -class instances. A  $\mathcal{FGP}\_m_i(\mathcal{G}, O)$  model has the following parameter structure:

- $finger\_shape(\mathcal{G}) = number, shape_i, size_i, i = 1, \dots, p$  expresses the shape of the gripper in terms of its number  $p$  of fingers, the shape and dimensions of each finger. *Rectangular*-shaped fingers are considered; their size is given by “width” and “height”;
- $fingers\_location(\mathcal{G}, O) = \{x_{ci}(O), y_{ci}(O), r_{zi}(O)\}, i = 1, \dots, p$ , indicates the relative location of each finger with respect to the object’s *center of mass* and *minimum inertia axis* (MIA). At training time, this description is created for the object’s model, and its updating will be performed at run time by the vision system for any recognized instance of the prototype.
- $finger\_viewing(\mathcal{G}, pose\_context_i, i = 1, \dots, p)$ , indicates how “invisible” fingers are to be treated; fingers are “invisible” if they are outside the field of view.
- $grip = 1, \dots, k$  are the  $k$  gripper-object  $Gs\_m(\mathcal{G}, O)$  distinct grasping models a priori trained, as possible alternatives to face at run time foreground context situations.

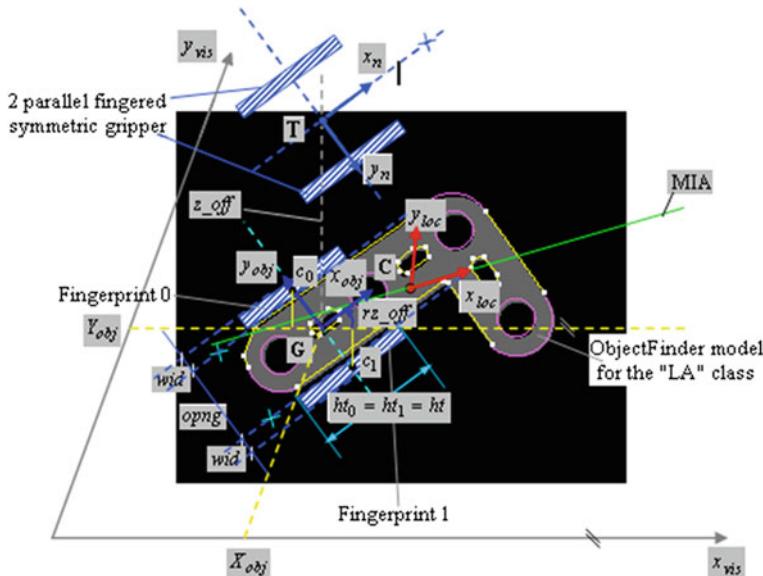
A *collision-free grasping transformation*  $\mathcal{CF}(Gs\_m_i, O)$  will be selected at run time from one of the  $k$  *grip* parameters, after checking that all pixels belonging to  $\mathcal{FGP}\_m_i$  (the projection of the gripper’s fingerprints onto the image plane  $x_{vis}, y_{vis}$ , in the  $O$ -grasping location) cover only background-coloured pixels. To provide a secure, collision-free access to objects, the following *robot-vision sequence* must be executed:

1. *Training*  $k$  sets of parameters of the multiple fingerprints model  $\mathcal{M}\mathcal{FGP}\_m(\mathcal{G}, O)$  for  $\mathcal{G}$  and object class  $O$ , relative to the  $k$  learned grasping styles  $Gs\_m_i(\mathcal{G}, O), i = 1, \dots, k$

2. *Installing* the multiple fingerprint model  $\mathcal{MFGP\_m}(\mathcal{G}, O)$  defining the shape, position and interpretation (viewing) of the robot gripper for clear-grip tests, by including the model parameters in a database available at run time. This must be done at the start of application programs prior to any image acquisition and object locating.
3. *Automatically performing the clear-grip test* whenever a prototype is recognized and located at run time, and grips  $\mathcal{FGP\_m}_i, i = 1, \dots, k$  have been a priori defined for it.
4. On line *call of the grasping parameters* trained in the  $\mathcal{Gs\_m}_i(\mathcal{G}, O)$  model, which corresponds to the first grip  $\mathcal{FGP\_m}_i$  found to be clear.

The first step in this robot-vision sequence prepares off-line the data allowing to position at run time two Windows Region of Interest (WROI) around the current object, invariant to its visually computed location, corresponding to the two gripper fingerprints. The data refer to the size, position and orientation of the gripper's fingerprints, and is based on:

- the *number and dimensions of the gripper's fingers*: 2-parallel fingered grippers are considered, each one having a rectangular shape of  $wd_g \times ht_g$ ;
- the *grasp location of the fingers relative to the model* of objects of interest.



**Fig. 11.16** Learning the fingerprint model  $\mathcal{FGP\_m}$  location data  $c_{0x}, c_{0y}, r_{0z}, c_{1x}, c_{1y}, r_{1z}$  relative to the ‘‘LA’’ finder model for a parallel symmetric gripper with 2 rectangular-shaped fingers of size  $wd_0, ht_0, wd_1, ht_1$ . The current grip is for the grasp model  $\mathcal{Gs\_m} = \{x\_off, y\_off, rz\_off, z\_off\}$

This last information is obtained by *learning* any grasping transformation for a class of objects (e.g. “LA”), and is described by help of Fig. 11.16.

In general, grips are tested in the order of their numbering. Once a grip is found to be clear, further testing is halted, because only one clear grip is required. The clear-grip algorithm checks each of the rectangles defining the grip until *one* is found to be *not clear*, or *all* are found to be *clear*.

For ObjectFinder models, the creation of the fingerprint model is very simple, but its installation and execution of the clear-grip test are not integrated in vision macro operations. Thus, the clear-grip test will be performed by calling in the application program a routine anytime a model instance was recognized and successfully located. This routine is based on a statistics analysis of the number of background pixels covered by each one of the gripper’s fingerprints in the image plane.

Figure 11.16 shows an “LA”-type object, trained and planned with the dual Object Finder model  $\mathcal{DFG}_m = (\mathcal{Fp}_m(O), \mathcal{Gs}_m(O))$ , for the recognition and grasping style definition of an instance  $O$ .

Two frames are used:

- $(x_{\text{loc}}, y_{\text{loc}})$  is the default reference frame attached automatically by the vision system to the ObjectFinder model in its center of mass  $\mathbf{C}(x_C, y_C)$  and aligned with its minimum inertia axis,  $x_{\text{loc}} \equiv \text{MIA}$ , at training time.
- $(x_{\text{obj}}, y_{\text{obj}})$  is the user-specified object coordinate system, which is shifted with  $x_{\text{off}} = x_G - x_C, y_{\text{off}} = y_G - y_C$  mm and turned with  $rz_{\text{off}} = \angle(x_{\text{obj}}, x_{\text{vis}})$  deg, where  $x_{\text{obj}} \parallel x_4(\text{SCARA})$ , and  $\mathbf{G}$  is the projection of the gripper’s end point  $\mathbf{T}$  on the image plane, in the grip position.

Once trained and installed the dual ObjectFinder model, any time an instance of the model “LA” is recognized and successfully VLOCATEd, the vision system returns the X, Y, and RZ data of the instance’s coordinate system. The data are available as the first, second, and sixth components of the transformation  $\text{vis}.\text{obj}$  assigned by VLOCATE, being expressed relative to  $(x_{\text{vis}}, y_{\text{vis}})$ .

The fingerprint model data  $c_{0x}, c_{0y}, rz_0, c_{1x}, c_{1y}, rz_1$  will be learned relative to the instance’s location  $X_{\text{obj}}, Y_{\text{obj}}, RZ_{\text{obj}}$ , expressed in the vision frame  $(x_{\text{vis}}, y_{\text{vis}})$ .

As can be seen from Fig. 11.16

$$\begin{aligned} c_{0x} &= X_{\text{obj}} - (opng / 2 + wid / 2) * \sin(rz_{\text{off}}) \\ rz_0 &= RZ_{\text{obj}} \\ c_{1x} &= X_{\text{obj}} - (opng / 2 + wid / 2) * \sin(rz_{\text{off}}) \\ c_{1y} &= Y_{\text{obj}} + (opng / 2 + wid / 2) * \cos(rz_{\text{off}}) \\ rz_1 &= RZ_{\text{obj}} \end{aligned} \quad (11.22)$$

where

$$X_{\text{obj}} = x_G - x_C, Y_{\text{obj}} = y_G - y_C, RZ_{\text{obj}} = rz_{\text{off}}|_{\mathcal{Gs}_m(\{\text{LA}\})}.$$

Assuming that the fingerprint rectangles have identical size  $wid, ht$ , the clear-grip routine, to be called and executed at run time, is based on generating two rectangular Windows Region of Interest WROI (e.g. in V+ with the operation `VWINDOW1`), having the exact shape, size, position, and orientation of the gripper fingerprints.

These two WROIs emulate in this way the projections of the two fingerprints on the image plane. Statistics will be gathered from the two windows in what concerns the number of background pixels existing in each window, i.e., background pixels covered by each fingerprint projection.

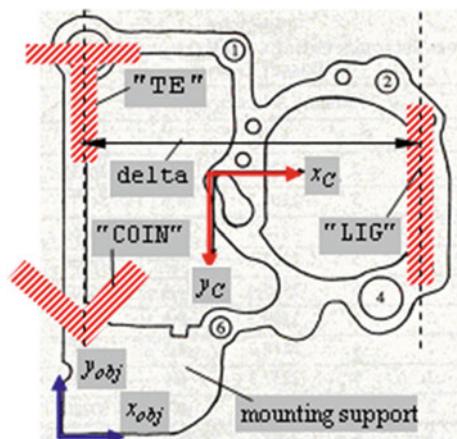
By comparing these two numbers with the fingerprint areas expressed in raw pixels, one detects whether the foreseen grip is clear (when the differences of areas are less than a residual value) or not clear (when a significant number of window pixels have foreground colors).

In opposition to the case of prototype instances, one cannot define multiple fingerprint models composed by different grips for one ObjectFinder model, because the unique object reference frame defined at training time is automatically attached to recognized instances.

## 11.5 Experimental Results and Conclusions

Experiments have been carried out on a development platform using a Cobra 600TT SCARA robot with compact Adept controller integrating an AVI vision board. The collision-free object access strategy was implemented for assembling MECCANO parts of 3.5 mm height on carburettor flanges (Fig. 11.17).

**Fig. 11.17** Assembling “COIN”, “TE” and “LIG” isolating components on a carburettor flange



**Table 11.1**  $G_{s,m}$  parameters learned for “COIN”, “TE”, and “LIG” models of assembly components

$G_{s,m}$ parameters	x_offset (mm)	y_offset (mm)	rz_offset (°)	z_offset (mm)
Class models				
“COIN”	-11.60	-6.75	-77.84	2.67
“TE”	-19.07	-11.02	-32.89	1.81
“LIG”	-11.38	-6.38	-77.06	2.26

Table 11.1 indicates the offset values learned in an interactive training session for the models of the classes of thin MECCANO parts on the carburettor flange.

The parts, randomly arriving on a conveyor belt, are picked “on-the-fly” by the 2-fingered gripper of the robot, after checking the fingerprint models  $\mathcal{FGP}_m$  against the foreground for parts closely placed or touching between them.

## References

1. Feddema J, Lee CSG, Mitchell OR (1991) Weighted selection of image features for resolved rate visual feedback control. *IEEE Trans Robot Autom* 7(February):31–47
2. Borangiu TH (2002) Advanced robot motion control. Romanian Academy Press & AGIR Press, Bucharest
3. Allen PK, Timcenko A, Yoshimi B, Michelman P (1993) Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. Robot. Automat.* 9(2):152–165
4. Borangiu TH, Ionescu F, Manu M (2003) Visual servoing in robot motion control. In: Proceedings of 7th multi-conference on systemics, cybernetics and informatics SCI’03, Orlando, July 27–30, pp 987–992
5. Borangiu TH, Kopacek P (eds) (2004) Proceedings volume from the IFAC workshop intelligent assembly and disassembly-IAD’03, Bucharest, Oct 9–11, 2003, Elsevier Science, Oxford
6. Arimoto S, Naniva T, Parra-Vega V, Whitcomb LL (1995) A class of Quasi-neutral potentials for robot servo-loops and its role in adaptive and learning controls, intelligent automation and soft computing, 1, 1, AutoSoft Press, 85–98
7. Battilotti S, Lanari L (1996) Tracking with disturbance attenuation for rigid robots. In: Proceedings of IEEE international conference on robot automation, Minneapolis, pp 1570–1583, April 1996
8. Borangiu TH, Dupas M (2001) Robot–vision. *Mise en œuvre en V+*. Romanian Academy Press & AGIR Press, Bucharest
9. Borangiu TH (2001) Reactive control architecture for mobile robots. In: Proceedings of the 2nd national workshop on mobile robots, Craiova, pp 47–52 (2001)
10. Borangiu TH, Ecaterina Oltean V, Manu M (2000) Multi-processor design of nonlinear robust motion control for rigid robots, *Lecture Notes in Computer Science No. 1798*, Springer, Berlin, pp 224–238
11. Borangiu TH, Ivanescu NA (2003) Visual servo architectures for rigid robots. In: Proceedingts of the 12th international workshop on robotics in Alpe-Adria-Danube Region RAAD’03, Cassino, May 7–10, pp 221–226
12. Doulgeri Z, Fahantidis N, Konstandinidis A (1998) On the decoupling of position and force controllers in constrained robotic tasks. *J Robot Syst* 15(6):373–340

13. Corke P (2001) Robotics toolbox release 6, [ftp.mathworks.com](ftp://mathworks.com), MathWorks FTP Server, April 2001
14. Freeman R, Kokotovic P (1996) Lyapunov design. In: Levine WS (ed) *The control handbook*. CRC Press, New York, pp 932–940
15. Haralick RM, Shapiro LG (1993) Computer and robot vision. Boston, Addison-Wesley
16. Hoffbeck JP, Langrebe DA (1996) Covariance matrix estimation and classification with limited training data. *IEEE Trans Pattern Anal Mach Intell* 8(11):431–442
17. Bejczy AK, Kim WS, Venerna SC (1990) The phantom robot: predictive displays for tele-operation with time delay. In: Proceedings of IEEE international conference on robotics and automation, pp 546–551, 1990
18. Jang BK, Chin RT (1992) One-pass parallel thinning: analysis, properties and quantitative evaluation. *IEEE Trans Pattern Anal Mach Intell* 14:11
19. Mc Carragher B, Horland G, Sikka P, Aigner P, Austin D (1997) Hybrid dynamic modelling and control of constrained manipulator systems. *IEEE Robot Autom Mag* 4(2):27–44
20. Nelson BJ, Papanikolopoulos NP, Khosla PK (June 1996) Robotic visual servoing and robotic assembly tasks. *IEEE Robot Autom Mag* 23:97–102
21. Hosoda K, Asada M (1994) Versatile visual servoing without knowledge of true Jacobean. In: Proceedings of IEEE/RSJ international conference on intelligent robots and systems IROS'94, New York, pp 186–191, Sept 1994
22. řtefănoiu D, Borangiu TH, Ionescu F (2003) Generating planar workspace boundaries by radius parsimonious increase, preprints of the IFAC interantional workshop intelligent assembly and disassembly IAD'03, Bucharest, Oct 9–11, 90–95 (2003)
23. Borangiu TH, Manu M (1999) Robot-object calibration techniques. In: Proceedings of international conference of IFAC/IEEE control systems and computer science CSCS'99, Bucharest, pp 218–223
24. Borangiu TH, Ivanescu N, Brotac S (2002) An analytical method for visual robot –object calibration. In: Proceedings of the 7th international workshop robotics in Alpe-Adria-Danube Region RAAD'98, Balatonfüred, pp 149–154
25. Hossu A, Borangiu TH, Croicu AL (1995) Robotvisionpro machine vision software for industrial training and applications. Version 2.2, Cat. #100062, Amsterdam, Tel Aviv, New Jersey, Eshed Robotec (1982), 1995
26. Mamitsvalov AG (1998) n-dimensional moment invariants and conceptual mathematical theory of recognition n-dimensional solids, *IEEE Trans Pattern Anal Mach Intell* 20:8
27. Moscheni F, Bhattacharjee S, Kunt M (1998) Spatiotemporal segmentation based on region merging, *IEEE Trans Pattern Anal Mach Intell* 20:8
28. Nguyen T, Oommen BJ (1997) Moment preserving piecewise linear approximations of signals and images. *IEEE Trans Pattern Anal Mach Intell* 19:1
29. Park J, Bien Z (1995) Design of an advanced machine vision system for industrial inspection. *Intell Autom Soft Comput* 1(2):209–219
30. Pearce A, Caelli T, Bischof WF (1994) Rulegraphs for graph matching in pattern recognition. *Pattern Recogn* 27:1231–1248
31. Qi L, Tufto DW (1997) Principal feature classification. *IEEE Trans Neural Netw* 8(1): 167–172
32. Nethery JF, Spong MW (1994) Robotica: a mathematica package for robot analysis. *IEEE Robot Autom Mag* 1(1):13–20

# Chapter 12

## Path Planning for Grasping Tasks

João Carlos Mendes Carvalho and Sezimária F. Pereira Saramago

**Abstract** In this chapter a novel algorithm is formulated and implemented for optimum path planning of parallel manipulators. A multiobjective optimization problem has been formulated for an efficient numerical solution procedure through kinematic and dynamic features of manipulator operation. Computational economy has been obtained by properly using a genetic algorithm to search an optimal solution for path spline functions. Numerical characteristics of the numerical solving procedure have been outlined through a numerical example applied to CaPaMan, Cassino Parallel Manipulator, both for path planning and for design purposes.

### 12.1 Introduction

In the past two decades parallel architectures have been extensively studied because they show advantages such as higher stiffness and accuracy positioning with respect to serial architectures, and they can operate at high velocities and accelerations as mentioned for example in [25]. These characteristics suggest their use in many applications, like for example in assembly and disassembly processes, packing, tool and object handling, milling machines, and motion simulation. Thus, several new parallel mechanisms have been conceived, designed, and built together with a development of theoretical and practical investigations like those that are presented in [6, 9, 11, 16, 22, 23, 26–28, 42–44].

---

J. C. M. Carvalho (✉) · S. F. P. Saramago  
Federal University of Uberlândia, Av. Joao Naves de Avila, Uberlandia-MG,  
2160-38408100, Brazil  
e-mail: jcmendes@mecanica.ufu.br

Simultaneously, with the study of new robotic structures, end effectors have been the subject of several researches since these devices are the elements of interaction between the robot and the environment, picking, handling, inspecting, or working objects. As the greater the sophistication of the robot application, more important its end effector.

In precise operations, robot hands imitating the human hand enable high efficiency and flexibility. However, the structure of such hand is very complicated because the human hand has many degrees of freedom (d.o.f.). One alternative consists in using industrial grippers where the style of jaws determines the required force and torque for the gripper application.

The development of gripping devices, in general, considers the static analysis of forces and torques that can be applied to an object through grasp contacts in order to keep the object firmly in the gripper.

But a critical factor in determining the required gripper force and torque is the resulting force due to the weight of the manipulating part and from acceleration.

Then, the trajectory should be as smooth as possible; i.e., abrupt changes in position, velocity, and acceleration should be avoided in order to prevent vibrations and ensuring that parts do not drop from grippers.

The goal of robot systems is to produce goods at as low cost as possible. The minimum-cost trajectory planning in the two-staged realization of manipulators control (i.e., planning first and tracking next) is, thus, an important effort to accomplish that goal. A large number of robotic applications involve repetitive processes. This technological characteristic justifies offline trajectory planning. In many industrial applications the movements of robot manipulators are planned manually in ad hoc manner so that usually they do not perform tasks with optimal paths. Generally, this method for path planning cannot give the best paths or/and the maximum permissible speeds at all points along programmed paths.

Basically, the trajectory planning problem consists in finding a relationship between two elements belonging to different domains: time and space [2] and that can be studied based on the desired motion to do a specific task. There are several objectives of the trajectory planning that can be cited: to avoid undesirable effects such as vibrations and wear of mechanical components, to prevent collisions avoiding damages on the mechanical structure, designing the system, choosing and sizing the actuators and transmission devices, reducing errors during the motion execution, optimizing the performances of the system, to ensure the reliability on grasping tasks, and others.

The trajectory can be done by point-to-point motion and by a continuous path motion. In the point-to-point motion, the robot end effector moves from an initial pose (or an initial configuration) to a final pose (or final configuration). In this motion the velocities and accelerations at the ends are zero and the described trajectory between points is not important. Then, it can be applied when no obstacles exist and are commonly used in *pick-and-place* operations for material handling and spot welding.

In the continuous path motion, the robot end effector traces a continuous trajectory along a curve such as those used on laser cutting and sealant

applications. In general, the trajectory is defined by a set of via points which a curve can be obtained by interpolation or approximation methods. The interpolation methods consist in obtaining a polynomial curve that passes through the points while on the approximation the curve passes near the points from a predefined distance.

Pick-and-place motion and continuous path motion can be planned in joint space or in task space (operational space usually defined in Cartesian space). However, if the trajectory planning is done in the task space the inverse kinematics must be solved, solving problems such as singularities and multiple solutions and, if the planning is done in the joint space should be given special attention to the end effector behavior because it cannot describe exactly the desired trajectory.

Several techniques are available for planning the desired movement, each of them with peculiar characteristics such as initial and final positions, velocities, accelerations, jerk, duration, and so on [1, 2, 7, 36, 45, 46].

When repetitive processes are prescribed, it is possible to develop a methodology to move a robot manipulator along a specified optimum path. This can be achieved through a formulation of a suitable optimization problem. Considerable research activity has been carried out in order to obtain optimal paths with serial robots and the corresponding literature is very rich. For example, Lin et al. [24] have proposed a procedure to determine a cubic polynomial joint trajectory through an algorithm for minimizing the traveling time subject to physical constraints on joint velocities, accelerations, and jerks. Shin and Mckay [41] have presented a solution to the problem of minimizing the power consumption of moving a serial robotic manipulator along a specified end effector path subject to input torque/force constraints, by taking into account the dynamics of the manipulator. Other optimum path planning methods can be found like for example in [7, 8, 21, 29, 31, 37–40, 47, 48].

A clear tendency exists in looking at models in nature to represent processes that can be called “intelligent.” There is strong evidence that natural processes related to human beings are very well developed and they can be adapted to the engineering world by bringing surprising results in a lot of applications. In the family of the so-called evolutionary computation methods, which include a growing number of paradigms and methods, G.A. are considered to be the most promising. G.A. are computational search methods that are based on the mechanisms of natural evolution and genetics [15]. In G.A. a population of possible solutions for a problem evolves in agreement with probabilistic operators that can be formulated as biological metaphors, so that, in general, individuals represent better solutions as the evolutionary process continues.

In this work, a general formulation has been proposed for optimum path planning for manipulators by taking into account mechanical energy of the actuators and total traveling time for a formulation of a multiobjective function. Feasible trajectories have been defined by using spline functions that can be obtained through offline computation. The nature of the proposed optimization problem is such that in general, several local minima exist and can be observed throughout a numerical procedure. A G.A. is capable of finding the global

minimum amongst many local minima, particularly in the case where traditional techniques fail. For this reason, a G.A. has been used for solving the numerical optimization problem for the optimal path of manipulators containing contradictory optimality criteria.

In the trajectory modeling, the uniform cubic B-splines that allow control of the degree of continuity at the joints between adjacent segments are used [18]. This fact is important because smooth transition is required in most robotic applications. To optimize a manipulator operation, the mechanical energy can be considered as one of the most significant aspect, since the energy formulation considers both the dynamics and kinematics characteristics of the manipulator. To maximize the operation speed, it is necessary to minimize the traveling time. Thus, these characteristics are considered together to build a multiobjective function. These objectives are in conflict with each other, mainly in the applications where the manipulator should work at high velocities. A general analysis code has been developed by involving a dynamical model and a trajectory planning of a robot.

The proposed procedure has been applied to CaPaMan, Cassino Parallel Manipulator, as a practical example, also in order to further improve its dynamic performance by reducing its power consumption through a minimization of the needed actuator energy and total traveling time in trajectory motion. CaPaMan is a parallel manipulator, having three d.o.f., which has been conceived at LARM: Laboratory of Robotics and Mechatronics at Cassino, Italy.

## 12.2 A Formulation for Optimal Path Planning

The path planning task for a manipulator with  $n$  d.o.f. can be described using  $m$  knots in the trajectory of each  $k$ th joint of a manipulator. The prescribed task can be given by the initial and final points  $P_0$  and  $P_m$  of the trajectory. The movement of the manipulator can be obtained by the simultaneous motion of the  $n$  joints in order to perform the prescribed task.

Many real-world engineering design or decision making problems involve the simultaneous optimization of multiple conflicting objectives. A multiobjective optimization problem is defined as such when the goal is to simultaneously minimize or maximize several functions with the same objective, with one function often conflicting with another [12]. In such cases, one must look for an optimized vector defined in  $n$ -dimensional Euclidean space,  $E^n$ , of project variables that offer the best solution for a vector of objective functions defined in the  $k$ -dimensional Euclidean space,  $E^k$ , of objective functions [32]. Multiobjective optimization problems appear in a variety of scientific applications, and several researchers have focused on developing methods to solve them.

Such problems may be subject to restrictions and all related functions may be nonlinear. Several objectives are available and can be considered to solve these complex optimization problems. In such cases, the same values of design variables

are unlikely to result in the best optimal values for all the objectives. Hence, some trade-offs between the objectives are needed to ensure a satisfactory design. Because system efficiency indices may differ (and be mutually contradictory), it is reasonable to use the multiobjective approach to optimize the overall efficiency. This can be done mathematically correctly only if a principle of optimality is used.

We have used a Pareto-optimality principle whose essence is as follows. The solution to the multiobjective optimization problem is considered Pareto-optimal if no other solution better satisfies all the objectives simultaneously [30]. In other words, there may be other solutions that better satisfy one or several objectives, but they must be less satisfactory than the Pareto-optimal solution in satisfying the remaining objectives. In that case, the result of the multiobjective optimization problem is finding a full set of Pareto-optimal solutions. As a rule, it is impossible to find a full infinite set of Pareto-optimal solutions for particular real-life problems. Therefore, the aim of a multiobjective engineering problem is to determine a finite subset of criteria-distinguishable Pareto-optimal solutions. Some classical methods are based on scaling the functions, with the vector objective function transformed into a scalar function, while others treat objective functions as additional restrictions.

In a multicriteria optimization, one deals with a design variable vector  $x$ , which satisfies all the constraints and makes the scalar performance index as small as possible. This index is calculated by taking into account each component of an objective function vector  $f(x)$ . An important feature of such multicriteria optimization problem is that the optimizer has to deal with conflicting objectives. A possible approach to this problem is the so-called compromise programming. This approach does not provide unique solution to the problem but a set of solutions named as Pareto-optima set [12]. Weighting objectives is one of the most usual and simple alternative approaches for multiobjective optimization problems. In this case, the objective function  $f$  can be determined by the linear combination of the  $r$  criteria  $f_1, \dots, f_r$ , together with the corresponding weighting factors  $w_1, \dots, w_r$  in the form

$$f(x) = \sum_{i=1}^r r_i w_i f_i(x), \text{ considering } \sum_{i=1}^r w_i = 1 \quad (12.1)$$

$r_i$  are constant multipliers,  $w_i \geq 0$  are the weighting coefficients that represent the relative importance of each criterion. Usually, weighting factors are assumed with the conditions  $0 \leq w_i \leq 1$ . Then, it is possible to generate Pareto-optima set for the original problem by varying the weights  $w_i$  in the objective function.

Objective weighting is obviously the most usual substitute model for vector optimization problems. The trouble here is attaching weighting coefficients to each of the objectives. The weighting coefficients do not necessarily correspond directly to the relative importance of the objectives or allow trade-offs between the objectives to be expressed. For the numerical methods for seeking the optimum of (12.1) so that  $w_i$  can reflect closely the importance of objectives, all the functions should be expressed in units of approximately the same numerical values.

An optimality criterion concerning with energy aspects of the path motion can be conveniently expressed in terms of the work that is needed by the actuators. In particular, the work by the actuators is needed for increasing the kinetic energy of the system in a first phase from a rest condition to actuators states at which each actuator is running at maximum velocity. In a second phase, bringing the system back to a rest condition, the kinetic energy will be decreased to zero through the actions of actuators and brakes. The potential energy of the system will contribute to size the necessary work by the actuators and friction effects in the joints can be assumed as negligible as compared to the actions of actuators and brakes. Therefore, in the robotic applications the potential energy change can be considered negligible if it is compared with the kinetic energy that is due to a desired fast manipulator motion.

Thus, we have considered it convenient to use the work  $W_{\text{act}}$  done by the actuators in the first phase of the path motion as an optimality criterion for optimal path generation as given by the expression

$$W_{\text{act}} = \sum_{k=1}^n \left[ \int_0^{t_k} \tau_k \dot{\alpha}_k \, dt \right] \quad (12.2)$$

in which  $n$  is the number of d.o.f. of the manipulator;  $\tau_k$  is the  $k$ th actuator torque;  $\alpha_k$  dot is the  $k$ th shaft angular velocity of the actuator; and  $t_k$  is the time coordinate value delimiting the first phase of path motion with increasing speed of the  $k$ th actuator.

Therefore, trying to minimize the ratio  $W_{\text{act}}/W_{\text{act}0}$  with  $W_{\text{act}0}$  as a prescribed value, has the aim to size at the minimum level the design dimensions and operation actions of the actuators in generating a path between two given extreme positions. The prescribed value  $W_{\text{act}0}$  has been chosen as referring to the power of a commercial actuator that has been considered suitable for the robot operation.

Indeed, in general once the actuator work is minimized, energy aspects of the system operation will be optimized consequently.

A scalar objective function can be proposed in order to consider minimum actuator work  $W_{\text{act}}$  and minimum traveling time  $Tt$  simultaneously in the form

$$\min f = w_1 \frac{W_{\text{act}}}{W_{\text{act}0}} + w_2 \frac{Tt}{Tt_0} \quad (12.3)$$

subject to

$$\alpha_k^l \leq \alpha_k(t) \leq \alpha_k^u \quad (k = 1, \dots, n) \quad (12.4)$$

$$Tt^l \leq Tt \leq Tt^u \quad (12.5)$$

where  $l$  and  $u$  stand for lower and upper values of the variables within their feasible ranges, respectively;  $\alpha_k$  is the  $k$ th joint variable;  $n$  is the number of d.o.f. of the manipulator;  $t$  is the time variable in the interval  $[0, Tt]$  for the path between  $P_0$

and  $P_m$ .  $Tt$  is the total traveling time at the end point  $P_m$  when  $t = 0$  is assumed at the initial point  $P_0$ ;  $Tt_0$  is total traveling time for the initial guess solution.

It is worth noting that the proposed multiobjective function in Eq. (12.3) is composed of two competitive terms. In general, the first term  $w_1 (W_{\text{act}}/W_{\text{act}0})$  decreases when the total traveling time  $Tt$  increases, while the second term  $w_2 (Tt/Tt_0)$  does the opposite. Therefore, the optimization process does not converge to the minimum permitted motion time  $Tt^l$ . The proposed formulation in Eqs. (12.2–12.5) requires the computation and consideration of the manipulator kinematics and dynamics. The motion kinematics will be addressed in the next section by using suitable algorithms for path planning and the torque computation has been formulated in Sect. 12.5.

A G.A. is capable of finding the global minimum among many local minima, particularly in the case where traditional techniques fail. For this reason, a G.A. has been used for solving the numerical optimization problem for the optimal path of manipulators containing contradictory optimality criteria. A brief summary of this algorithm is presented in Sect. 12.4.

## 12.3 A Formulation for Trajectory Modeling

In order to determine joint trajectories, one can use the given initial and final points  $P_0$  and  $P_m$  in the Cartesian coordinates. These given points can be transformed into joint coordinates by solving the inverse kinematics. Then, cubic polynomial can be chosen to describe joint trajectories since Proportional Derivative control law is generally used for manipulator actuators in industrial robots. B-splines are often used as interpolating functions to represent a trajectory of mechanical systems. An important characteristic is that they allow to control the degree of continuity between two adjacent segments. This fact is important because smooth transition is required for path planning in many applications such as in robotics. Another important characteristic of the cubic B-splines is that they satisfy the convex hull property, which allows the refinement of a trajectory.

B-Spline is, in reality, a curve fitting by approximation because the generated curve does not pass on the control points (or control parameters). If it is necessary to pass the curve through a set of known points, an interpolation can be done by obtaining new control points from the known data points [10, 13, 34, 35]. Let  $U = \{t_0, \dots, t_m\}$  be a nondecreasing sequence of real numbers i.e.,  $t_i < t_{i+1}$ ,  $i = 0, 1, \dots, (m - 1)$ , where  $t_i$  are called *knots*,  $U$  is the *knot vector* and, the  $i$ th B-spline basis function of  $d$ -degree (order  $d + 1$ ) is denoted by  $B_{i,d}(t)$ .

Thus, referring to the  $n$  d.o.f of the manipulator, each trajectory function  $\alpha_k(t)$  can be modeled by a uniform cubic B-Spline in the form

$$\alpha_k(t) = \sum_{i=0}^m p_i^k B_{i,d}^k(t) \quad (m \geq n, k = 1, \dots, n) \quad (12.6)$$

where  $p_i^k$  (with  $i = 0, \dots, m$ ) are the  $m + 1$  control parameters corresponding to the trajectory function  $\alpha_k(t)$ ;  $B_{i,d}$  are the functions that can be defined by using the Cox de Boor recurrence formulas [14], with  $d = 3$  for cubic spline, in the form

$$B_{i,1}(t) = \begin{cases} 1 & \text{if } t_i \leq t \leq t_{i+1} \\ 0 & \text{if } t_i > t \text{ or } t > t_{i+1} \end{cases} \quad (12.7)$$

$$B_{i,d}(t) = \frac{t - t_i}{t_{i+d-1} - t_i} B_{i,d-1}(t) + \frac{t_{i+d} - t}{t_{i+d} - t_{i+1}} B_{i+1,d-1}(t) \quad \forall t$$

Since  $\alpha_k(t)$  is a cubic, its  $j$ th derivatives with respect to  $t$  can be straightforward computed as

$$\frac{d^j \alpha_k(t)}{dt^j} = \sum_{i=0}^m p_i^k \frac{d^j B_{i,d}^k(t)}{dt^j} \quad (12.8)$$

Thus, the first and second derivatives related to the time are given by:

$$\dot{\alpha}_i(t) = \sum_{k=0}^m p_k^i \dot{B}_{k,d}^i(t) \text{ and } \ddot{\alpha}_i(t) = \sum_{k=0}^m p_k^i \ddot{B}_{k,d}^i(t) \quad (12.9)$$

The initial cubic B-spline control points can be obtained by guess fitting a cubic polynomial between the initial and final points of each trajectory. It is worth noting that the time interval is divided into  $n_t$  steps. The efficiency of the process depends on the number of steps adopted to discretize the time, but a large number of  $n_t$  requires a greater computational effort. Note that the optimization design variables for path planning are the  $m$  control parameters  $p_k^i$  of each joint trajectory together with the total time  $Tt$  of path traveling. Therefore, considering a manipulator with  $n$  input actuators, the total number of design variables for path planning is  $Nv = (n \times m) + 1$ .

## 12.4 Genetic Algorithms

G.A. is an iterative technique based on random search with adaptations of the search minimum in coordinate directions. It can be useful to overpass small variations in the objective function by using a probabilistic criterion in order to avoid stopping the procedure at a local minimum. G.A. is a subset of evolutionary algorithms that model biological processes to optimize highly complex cost functions. This algorithm allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the “fitness”, i.e., that minimizes the cost function.

G.A. is a random technique that can jump out of a local minimum. The method was developed by Holland [19] during the 1960s and 1970s and finally popularized by one of his students, Goldberg [15], who successfully solved a difficult problem for his dissertation, involving the control of a gas pipeline transmission.

The technique's main characteristics are as follows: (a) G.A.s operate on a population of points and, unlike conventional methods, they do not focus all the search effort on only one point; (b) They operate in a space of coded solutions rather than directly in the search space; (c) They do not require derivation, unimodality, or any other function knowledge to operate; (d) They need only the value of the objective function for each individual of the population, and (e) They use probabilistic rather than deterministic transitions.

Thus, similar to the evolution process in the search for the most adapted solution along successive generations, the optimization procedure improves these solutions until the optimal one is identified. A simple G.A. performs basically three operations: selection, crossing, and mutation.

Let the vector  $x = (x_1, x_2, \dots, x_{Nv})$  with  $Nv$  design variables. A population is composed of a set of  $Np$  points in the feasible region (called chromosomes or individuals). In the binary genetic algorithm, the encoding of a chromosome that has  $Nv$  variables, each variable represented by  $m_s$  bits, is shown below

$$\text{chromosome} = \left[ \underbrace{\overbrace{11010}^{x_1}}_{\text{gene}_1} \quad \underbrace{\overbrace{01100}^{x_2}}_{\text{gene}_2} \quad \dots \quad \underbrace{\overbrace{01010}^{x_i}}_{\text{gene}_i} \quad \dots \quad \underbrace{\overbrace{00101}^{x_{Nv}}}_{\text{gene}_{Nv}} \right]$$

Thus, each gene symbolizes a variable of the objective function. The gene is represented by binary string of length  $m_s$  for encoding the  $i$ th element of the objective variable of the vector  $x$ . The size of the genes may be different because it depends on the side constraints.

The initial population is usually generated in a random manner or through some heuristic process. The decoding can be done according to Eq. (12.10):

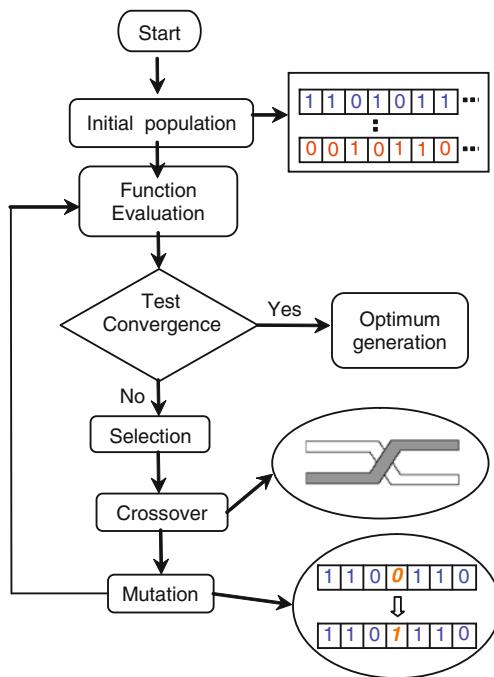
$$x_i = x_i^l + \frac{x_i^u - x_i^l}{2^{m_s} - 1} \sum_{j=0}^{m_s-1} b_{ij} 2^j, \quad \text{for } i = 1, \dots, Nv \text{ variables} \quad (12.10)$$

where  $x_i^l$  and  $x_i^u$  represent the lower and upper values of each objective variable. Note that, by using this equation, the real numbers are assigned to variables within the feasible region. The main characteristics of this technique are presented in the flowchart in Fig. 12.1.

As in natural genetics, evolution goes hand-in-hand with variety; hence, it is important for individuals to display different degrees of adaptation to their environment. This means that the initial population covers the search space in the best possible way.

In the *Selection* operation, a temporary population of  $Np$  individuals is generated considering the proportional probability of each individual with respect to its relative adaptability in the population. The individuals presenting low adaptability will be more prone to disappear [17]. There exist a great number of selection operators. In the roulette wheel selection, using the fitness value  $F_i$  of all strings, the probability of selecting the  $i$ th string is

**Fig. 12.1** A flowchart for procedures in genetic algorithms



$$p_i = F_i \left/ \sum_{j=1}^{Np} F_j \right. \quad (12.11)$$

The *Crossover* operator works by selecting two individuals who will exchange genetic material. It is also a random process which occurs with a probability established by the user. As in nature, *Mutation* is a rarely occurring event whose purpose is to ensure that important genetic material is not irrevocably lost. The process ends when the maximum number of generations is reached, or, according to the stagnation concept, when an improvement in the population is observed after several serial iterations. This methodology is potentially very useful for solving optimization problems.

In the optimization process, G.A. was applied through the program Genetic Algorithms Optimization on Toolbox (GAOT) developed for Houck et al. [20].

## 12.5 CaPaMan Prototype

The proposed procedure has been applied to CaPaMan, as a practical example, also in order to further improve its dynamic performance by reducing its power consumption through a minimization of the needed actuator energy and total

**Table 12.1** Sizes and motion parameters of the built prototype of CaPaMan, Figs. 12.2 and 12.3

$a_k = c_k$ (mm)	$b_k = d_k$ (mm)	$H_k$ (mm)	$r_p = r_f$ (mm)	$\alpha_k$ (°)	$s_k$ (mm)
200	80	100	109.5	45; 135	-50; 50

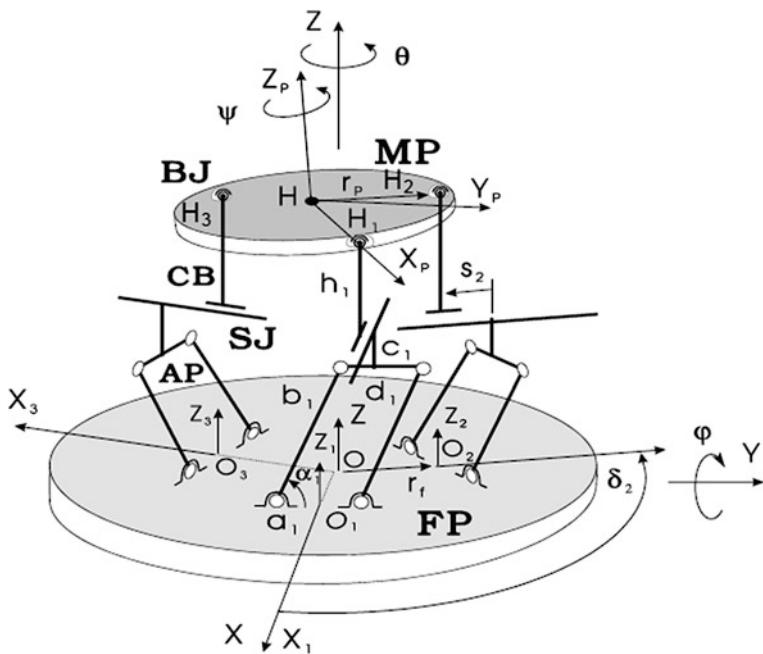
traveling time in trajectory motion. CaPaMan architecture has been conceived at LARM Cassino, where a prototype has been built for experimental activity since 1996 [6], Figs. 12.2 and 12.3.

A schematic representation of the CaPaMan manipulator is shown in Fig. 12.2, where the fixed platform is FP and the moving platform is MP. MP is connected to FP through three identical leg mechanisms and is driven by the corresponding articulation points  $H_1$ ,  $H_2$ , and  $H_3$ . An articulated parallelogram AP, a prismatic joint SJ, and a connecting bar CB compose each leg mechanism. AP's coupler carries the SJ and CB transmits the motion from AP to MP through SJ; CB is connected to the MP by a spherical joint BJ, which is installed on MP. CB may translate along the prismatic guide of SJ keeping its vertical posture and BJ allows MP to rotate in the space. Each plane, which contains AP, is rotated of  $\pi/3$  with respect to the neighbor one. Figure 12.3 shows a built prototype whose main design parameters are listed in Table 12.1.

Particularly, design parameters of the  $k$ th leg are identified through:  $a_k$ , which is the length of the frame link;  $b_k$ , which is the length of the input crank;  $c_k$ , which is the length of the coupler link;  $d_k$ , which is the length of the follower crank; and  $h_k$ , which is the length of the connecting bar. The kinematic variables are:  $\alpha_k$ , which is the input crank angle;  $s_k$ , which is the stroke of the prismatic joint. Finally, the size of MP and FP are given by  $r_p$  and  $r_f$ , respectively,  $H$  is the center point of MP,  $O$  is the center point of FP,  $H_k$  is the center point of the  $k$ th BJ, and  $O_k$  is the middle point of the frame link  $a_k$ , Fig. 12.2. The motion of MP with respect to FP can be described by considering a world frame  $O\text{-}XYZ$ , which is fixed to FP, and a moving frame  $H\text{-}X_pY_pZ_p$ , which is fixed to MP.

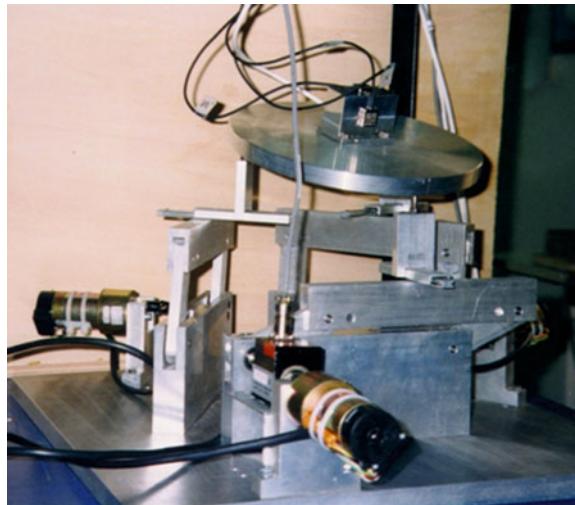
The symmetry characteristics of CaPaMan architecture have been useful to formulate analytical dynamic equations to compute the input torques, which are necessary for a given motion trajectory of the movable platform as reported in [3–5]. Assumptions have been made in order to simplify the equations such as the effects of link elasticity and viscous damping of the joints have been neglected; links are assumed to be rigid bodies and the joints are frictionless and have no clearance. In addition, only the inertial effects of the mobile platform have been considered, since the legs of parallel architectures are lighter than the movable plate. Successively, the inertial effects of the legs and prismatic joint have been superposed.

By neglecting the friction on prismatic and spherical joints, the only forces that are applied to the rods CB by the mobile platform are those, which are contained in the plane of the articulated parallelogram, i.e.,  $F_{ky}$  and  $F_{kz}$  as shown in Fig. 12.4.



**Fig. 12.2** Kinematic chain and design parameters of CaPaMan (Cassino Parallel Manipulator)

**Fig. 12.3** A prototype of CaPaMan with accelerometers and a dynamic torsionmeter at LARM in Cassino



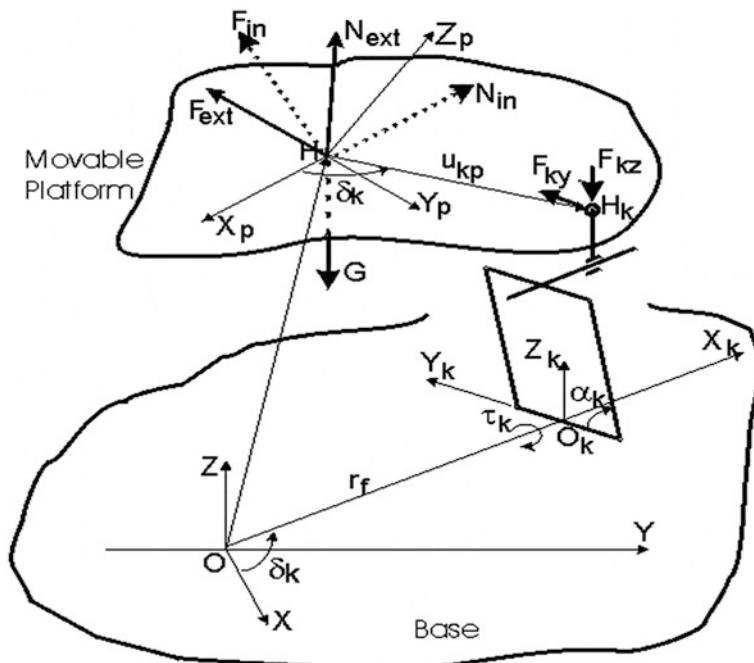
The  $F_{kx}$  component determines the sliding of the prismatic joint along  $s_k$  coordinate. Thus, the components of the resultant force  $F$  and torque  $N$  applied to the mobile platform can be computed as

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \\ N_x \\ N_y \\ N_z \end{Bmatrix} = \begin{Bmatrix} -\frac{\sqrt{3}}{2}F_{2y} + \frac{\sqrt{3}}{2}F_{3y} \\ F_{1y} - \frac{1}{2}F_{2y} - \frac{1}{2}F_{3y} \\ F_{1z} + F_{2z} + F_{3z} \\ -u_{1z}F_{1y} + \frac{1}{2}u_{2z}F_{2y} + \frac{1}{2}u_{3z}F_{3y} + u_{1y}F_{1z} + u_{2y}F_{2z} + u_{3y}F_{3z} \\ -\frac{\sqrt{3}}{2}u_{2z}F_{2y} + \frac{\sqrt{3}}{2}u_{3z}F_{3y} - u_{1x}F_{1z} - u_{2x}F_{2z} - u_{3x}F_{3z} \\ \frac{1}{2}(\sqrt{3}u_{2y} - u_{2x})F_{2y} - \frac{1}{2}(\sqrt{3}u_{3y} + u_{3x})F_{3y} + u_{1x}F_{1y} \end{Bmatrix} \quad (12.12)$$

with

$$\begin{Bmatrix} u_{kx} \\ u_{ky} \\ u_{kz} \end{Bmatrix} = r_p R \begin{Bmatrix} \cos \delta_k \\ \sin \delta_k \\ 0 \end{Bmatrix}; \quad (k = 1, 2, 3) \quad (12.13)$$

when the values  $\delta_1 = 0$ ;  $\delta_2 = 2\pi/3$ ;  $\delta_3 = 4\pi/3$  are considered.



**Fig. 12.4** Forces acting on the mobile platform of CaPaMan

Equations (12.12) and (12.13) can be solved in closed form to obtain an explicit expression for forces  $F_{ky}$  and  $F_{kz}$ . Referring to Fig. 12.4 and once the reaction forces in the spherical joints  $H_k$  are computed, the torque  $\tau_{Pk}$  ( $k = 1, 2, 3$ ) on the input crank shaft of each articulated parallelogram can be obtained by considering only the inertial effects of the movable platform in the form

$$\tau_{Pk} = \frac{F_{kz} b_k}{2} \frac{\sin 2\alpha_k}{\sin \alpha_k} - F_{ky} b_k \left( \frac{h_k}{c_k \tan \alpha_k} + 1 \right) \left( 1 - \frac{h_k}{h_k \cos \alpha_k + c_k \sin \alpha_k} \right) \sin \alpha_k \quad (12.14)$$

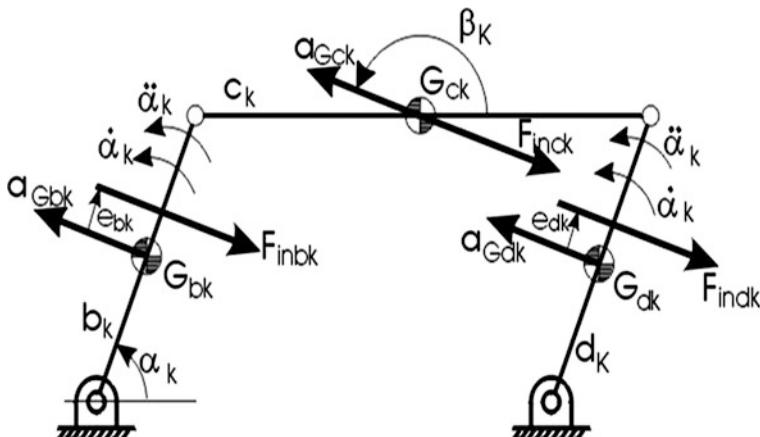
where  $b_k$ ,  $c_k$ , and  $h_k$  are the geometrical dimensions shown in Fig. 12.2;  $F_{ky}$  and  $F_{kz}$  are the reaction forces in the spherical joints  $H_k$ .

The contribution of the legs to the inverse dynamics of the CaPaMan can be determined by a kinematic analysis of the articulated parallelograms. The centers of mass of the links can be identified as shown in Fig. 12.5.

A kinetostatic analysis and superposition principle can be used for computing separately the inertial and gravitational effects of the links  $b_k$ ,  $c_k$ , and  $d_k$  and then they can be combined to give the torque generated by the gravity effect  $\tau_{Mk}$  as

$$\begin{aligned} \tau_{Mk} = & 2l_{bk} F_{inbk} \sin(\alpha_k - \beta_k + \pi) + F_{23k} b \sin(\alpha_k + \pi - \gamma_k) \\ & + b \left[ m_{bk} \cos \alpha_k + \frac{m_{ck}}{2} \frac{\sin 2 \alpha_k}{\sin \alpha_k} \right] g \end{aligned} \quad (12.15)$$

The angle  $\beta_k$  in Eq. (12.15) defines the direction of the acceleration of the center of mass of the  $k$ th link with respect to the horizontal axis as shown in Fig. 12.5. It is assumed to be positive counterclockwise. Similarly, the angle  $\gamma_k$  in Eq. (12.15) defines the direction of the reaction force vector acting on the ground



**Fig. 12.5** Inertia forces arising on each articulated parallelogram of CaPaMan legs

pivot of link  $d_k$  and it is also assumed to be positive counterclockwise. The terms  $l_{bk}$ ,  $F_{23k}$ , and  $\gamma_k$  in Eq. (12.15) can be written as

$$\begin{aligned} l_{bk} &= \frac{b}{2} + \frac{I_{Gbk}\ddot{\alpha}_k}{F_{inbk}} \frac{1}{\sin(\alpha_k - \beta_k + \pi)} \\ F_{23k} &= \sqrt{\left\{ F_{inck} \left[ \cos(\beta_k + \pi) + \frac{\sin(\pi - \beta_k)}{2\tan\alpha_k} \right] \right\}^2 + \left[ \frac{F_{inck} \sin(\beta_k + \pi)}{2} \right]^2} \quad (12.16) \\ \gamma_k &= \operatorname{tg}^{-1} \left\{ \left[ F_{inck} \left( \cos(\beta_k + \pi) + \frac{\sin(\pi - \beta_k)}{2 \tan \alpha_k} \right) \right] / \left[ \frac{F_{inck} \sin(\beta_k + \pi)}{2} \right] \right\} \end{aligned}$$

where  $I_{Gbk}$  is the inertia matrix with respect to link center of mass  $G_{bk}$ ;  $F_{inbk}$ ,  $F_{inck}$ , and  $F_{indk}$  are the inertia forces of each link of the articulated parallelograms. Details of derivations of the terms in Eq. (12.16) are reported in [3, 4].

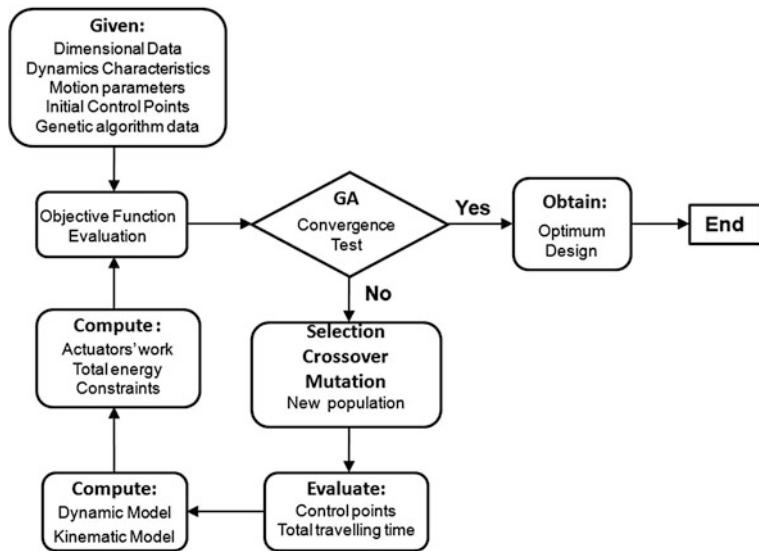
The superposition principle can be used again in order to obtain the total torque  $\tau_k$  on the input shaft of the articulated parallelogram as the sum of the torques computed in Eqs. (12.14) and (12.15) in the form

$$\tau_k = \tau_{Pk} + \tau_{Mk} \quad (12.17)$$

## 12.6 A Numerical Example of Optimal Path Planning

The Matlab code GAOT [20] has been used to find an optimum path through G.A. A general analysis code has been developed by involving a dynamical model and a trajectory planning of the robot. Then, it has been implemented into the proposed solving procedure according to the flowchart in Fig. 12.6. In particular, a dimensional data, dynamic characteristics and motion parameters have been used for computing the kinematics and dynamics of the system. The dynamic model has been used for computing the mechanical energy that is needed for the motion. The proposed optimization procedure requires inputs in the form of control points, G.A. data, and computed mechanical energy. If the total energy and the traveling time are not optimal under given constraints, the procedure searches for a different path. If the total energy and traveling time are optimal, then the procedure stops and gives as output the identified optimal path.

A numerical example referring to CaPaMan is reported by assuming a starting position given by  $\alpha_1 = 60^\circ$ ,  $\alpha_2 = 50^\circ$ ,  $\alpha_3 = 80^\circ$ , and a final position given by  $\alpha_1 = 90^\circ$ ,  $\alpha_2 = 120^\circ$ ,  $\alpha_3 = 100^\circ$ . Dimensional data of CaPaMan are reported in Table 12.1. The robot is considered as starting at rest and coming to a full stop at the end of the trajectory. Thus,  $\dot{\alpha}_k(0) = \dot{\alpha}_k(T_f) = 0$  for all the joint trajectories. Singularities of CaPaMan have been investigated in [33]. In particular, CaPaMan has only one singular configuration within its working range and it is given by the three input shafts at  $90^\circ$  position (vertical configuration). This configuration has been avoided in the path planning by adding a proper inequality constraint. Moreover, additional



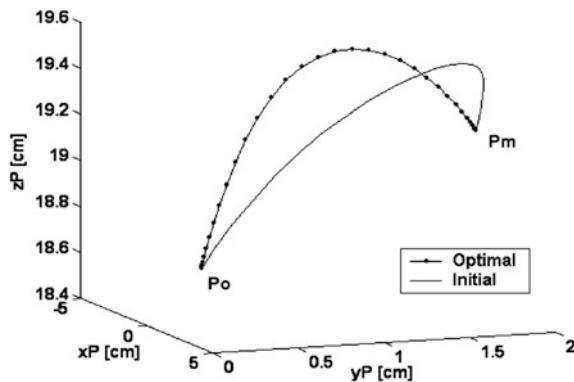
**Fig. 12.6** A flowchart of the proposed numerical procedure for optimum path planning of CaPaMan parallel manipulator

**Table 12.2** Optimum values as obtained from the optimization process with  $w_1 = 0.2$  and  $w_2 = 0.8$

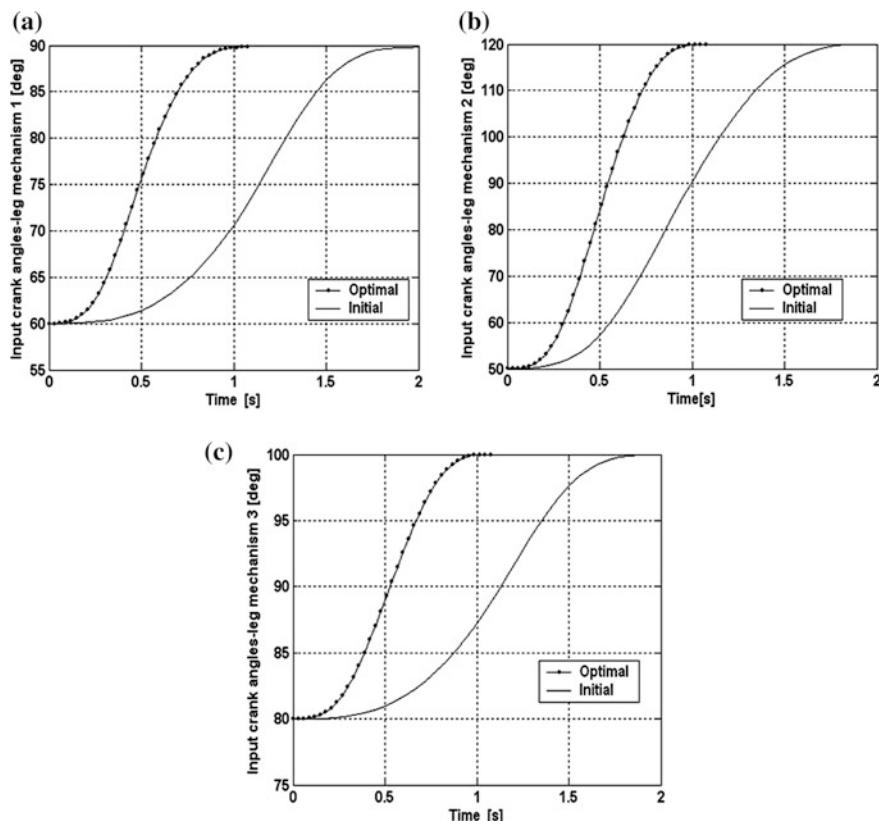
	Objective function $f$	Work by actuators $W_{act}$ (J)	Traveling time $Tt$ (s)
Initial guess value	1.00	4.11	2.0
Optimum value	0.63	3.90	1.1
Performance improvement	37 %	5.1 %	45 %

inequality constraints have been implemented for limiting the motion of each input crank of CaPaMan to its feasible working range that is from 45 to 135°, Table 12.2.

A suitable number of control points are necessary to obtain a proper B-spline. The higher the number of control points, the more complex the profile that can be obtained. But, any increase in the number of control points produces also an increase in the computational costs. The number of control points has been chosen as equal to eight for the reported case of study while the time interval has been divided into  $n_t = 32$  steps. It is worth noting that the number of control points and steps within the time interval have been chosen after simulation trials as a good compromise between accuracy and computational efficiency. The number of the design variable is  $Nv = 25$ , considering the number of actuators for CaPaMan  $n = 3$  and  $m = 8$  is the chosen number of control points. In addition, the following parameters have been used for the G.A. as suggested by the authors' experience:  $Np = 200$  individuals in the population; 300 generations; binary strings for



**Fig. 12.7** Numerical results for the case of Table 12.2 for  $w_1 = 0.2$  and  $w_2 = 0.8$  in the form of a 3D plot of the position of the center of the movable plate of CaPaMan as function of time

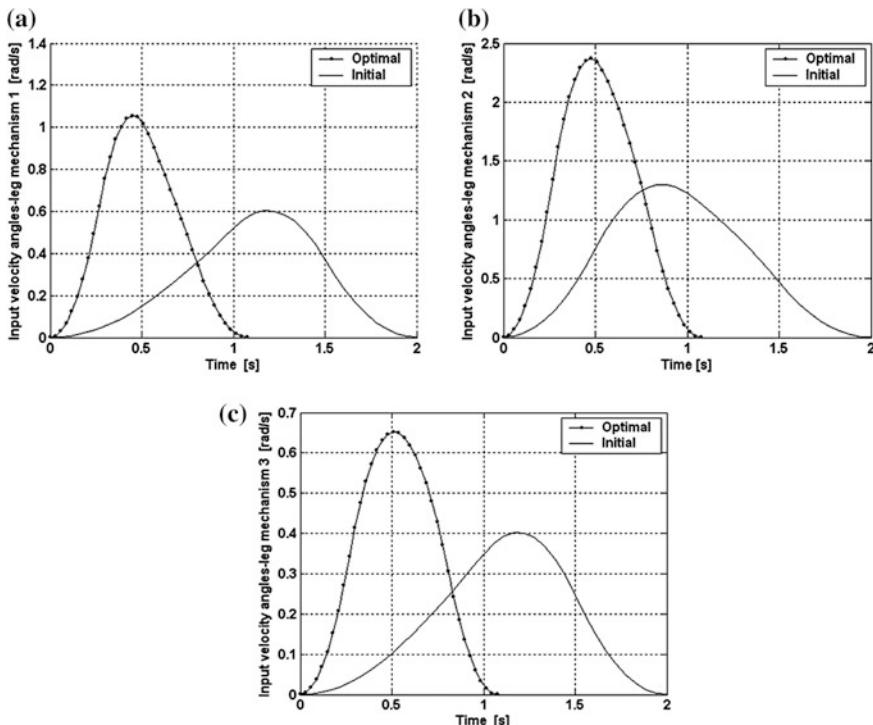


**Fig. 12.8** Time evolution of the input crank angles for initial guess (continuous line) and optimal solution (dotted line) for the numerical example with CaPaMan in Fig. 12.7: **a** leg mechanism 1; **b** leg mechanism 2; **c** leg mechanism 3

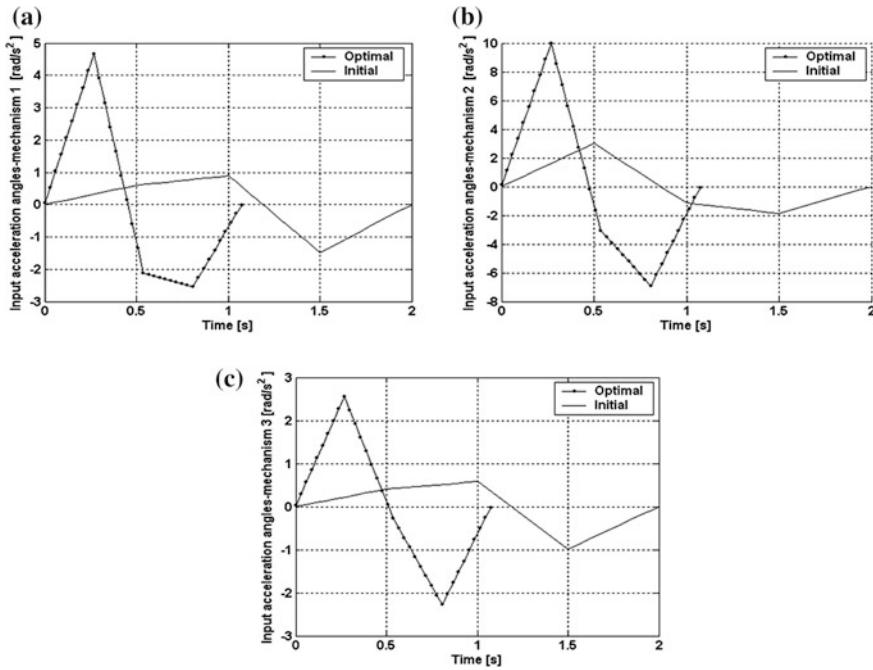
representation of individuals; roulette wheel selection with elitism. Crossover and mutation probabilities have been chosen as equal to 0.60 and 0.02, respectively. The above-mentioned values have been chosen in order to achieve a good compromise between accuracy of results and computational costs.

A numerical case of study is reported as referred to CaPaMan, Fig. 12.2 and Table 12.1, with the above-mentioned path data. The initial guess and optimal values are compared in Table 12.2 that shows a significant improvement of the performance index for the optimal trajectory in Fig. 12.7. This result is obtained using GAOT code in a Pentium 4 Computer with a computational time of less than 10 min. Moreover, for this application the weighting coefficients have been assumed as  $w_1 = 0.2$  and  $w_2 = 0.8$  in order to give more emphasis to the minimization of the total traveling time as required in several practical applications.

Figure 12.7 shows a 3D plot of the initial guess and optimal path of the center of the movable plate of CaPaMan. In particular, the guess path between the given points  $P_0$  and  $P_m$  has been chosen as a cubic spline with three intermediate knots that are arranged in such a way so as to set the initial and final accelerations to



**Fig. 12.9** Time evolution of the input crank angular velocities for initial guess (*continuous line*) and optimal solution (*dotted line*) for the numerical example with CaPaMan in Fig. 12.7: **a** leg mechanism 1; **b** leg mechanism 2; **c** leg mechanism 3

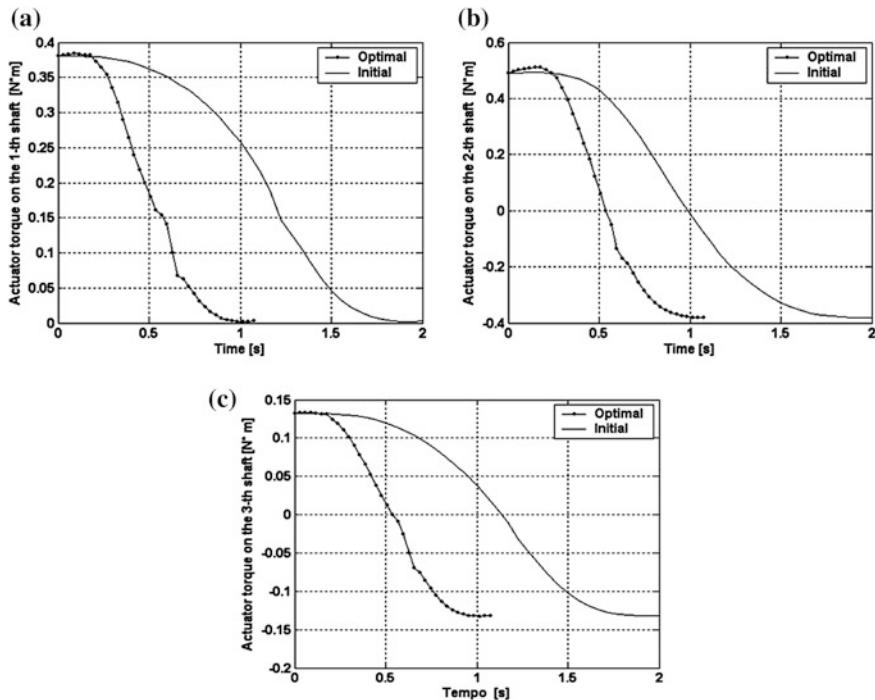


**Fig. 12.10** Time evolution of the input crank angular accelerations for initial guess (*continuous line*) and optimal solution (*dotted line*) for the numerical example with CaPaMan in Fig. 12.7: **a** leg mechanism 1; **b** leg mechanism 2; **c** leg mechanism 3

zero. One can observe that the computed optimal trajectory is smooth and fulfills the given constraints for the initial and final points of the path.

Figure 12.8 shows the initial guess and optimal trajectories of the input crank angles. It can be observed that although the total traveling time has been strongly reduced, the optimal curves are smooth.

Figure 12.9 shows the initial guess and optimum value of angular velocity for the input cranks. It is worth noting that the maximum values of crank angles' velocity in the optimum case are almost twice the maximum values of velocity in the initial guess. Thus, using the optimal solution, the desired motion of CaPaMan could be achieved almost in half the time with respect to the initial solution. Namely, the final position is reached in 1.1 s in the optimum solution and in 2 s in the initial guess case, as also reported in Table 12.2. Moreover, the maximum values of angular velocity for the input cranks in the optimum solution are 2.3 rad/s at the most (see Fig. 12.9b) and this value is still within the feasible working ranges for the actuators of CaPaMan. Similar considerations can be deduced for acceleration plots in Fig. 12.10. In fact, the maximum value of acceleration for the optimum solution is limited to 10 rad/s<sup>2</sup> as shown in Fig. 12.10b. Moreover, the crank angle accelerations for the optimum solution in Fig. 12.10 also have a feasible smooth profile. In addition, both velocities and accelerations do not show



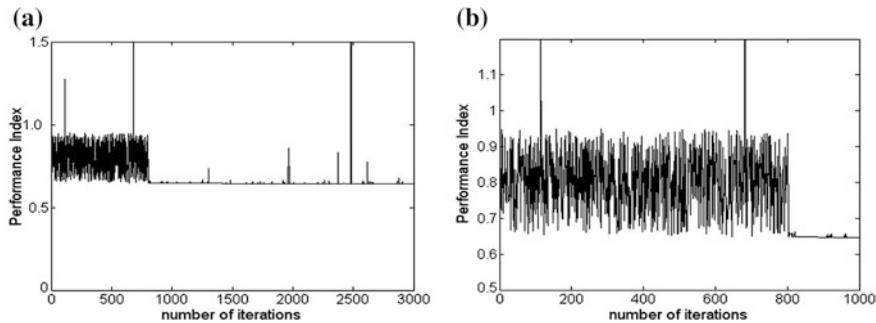
**Fig. 12.11** Time evolution of the actuator torques on input cranks for initial guess (continuous line) and optimal solution (dotted line) for the numerical example with CaPaMan in Fig. 12.7: **a** leg mechanism 1; **b** leg mechanism 2; **c** leg mechanism 3

high frequency oscillations that can be dangerous for durability of actuator operation.

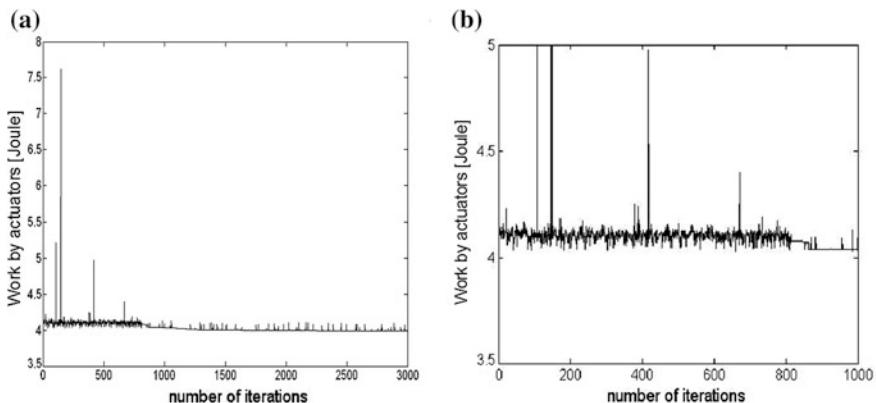
Figure 12.11 shows a comparison of initial guess and optimal actuator torque on the input shafts. It can be observed in Fig. 12.11 that the optimal torque has a smooth shape, oscillations of torque are limited, and a feasible maximum value of the torque is obtained as equal to 0.5 Nm (see Fig. 12.11b). Moreover, the maximum values of the torque which are required for the optimum solution do not increase significantly with respect to initial guess.

Figure 12.12 shows the evolution of the objective function  $f$  as function of the number of iterations. In particular, the objective function  $f$  converges to a value of 0.63 from an initial value of 1.00 after less than 800 iterations with a performance improvement of 37 % as reported in Table 12.2.

Moreover, Fig. 12.13 shows the evolution of the energy as function of the number of iterations. In particular, the needed energy  $E$  decreases from 4.11 to 3.90 Nm/s<sup>2</sup> with performance improvement of 5.1 % as also reported in Table 12.2. Thus, an improvement of all the components of the objective function  $f$  has been achieved as summarized in Table 12.2.



**Fig. 12.12** Evolution of the objective function for the case of Fig. 12.7 as function of the number of iterations: **a** the full plot; **b** a zoomed view from 0 to 1000 iterations



**Fig. 12.13** Evolution of the total manipulation energy for the case of Fig. 12.7 as function of the number of iterations: **a** the full plot; **b** a zoomed view from 0 to 1000 iterations

It is important to recall that a random optimization has been used for the proposed case of study. The initial population has been considerably evolved and after several generations the population has converged to an optimal solution. However, in the case of conflicting objectives, usually the set of optimal solutions contains more than one solution. Thus, the optimum result represents a compromise solution among the functions that contribute to the multiobjective function given by Eq. (12.9). Of course, in case the user changes the weighting coefficients, different results will be obtained.

## 12.7 Conclusions

A general optimum path planning procedure has been proposed as applied specifically to parallel manipulators by using suitable formulation for kinematics and dynamics of path motion. In particular, optimality criteria have been identified in work by actuators and traveling time of the manipulator trajectory. A numerical case of study is illustrated by referring to the parallel manipulator named as CaPaMan. The results of the optimum procedure show the engineering feasibility of the proposed formulation in order to improve the dynamics performance of a parallel manipulator and to reduce power consumption in a path planning application.

## References

1. Angeles J (1997) Fundamentals of robotic mechanical systems: theory, methods, and algorithms. Springer, New York
2. Biagiotti L, Melchiorri C (2008) Trajectory planning for automatic machines and robots. Springer, Berlin
3. Carvalho JCM, Ceccarelli M (1999) A dynamic analysis for cassino parallel manipulator. In: Proceedings of 10th IFToMM world congress, Oulu, vol 3, pp 1202–1207
4. Carvalho JCM, Ceccarelli M (2001a) The inverse dynamics of cassino parallel manipulator. In: Proceedings of 2nd workshop on computational kinematics—CK 2001, Seoul, pp 301–308
5. Carvalho JCM, Ceccarelli M (2001) A closed form formulation for the inverse dynamics of cassino parallel manipulator. *J Multibody Syst Dyn* 5(2):185–210
6. Ceccarelli M (1997) A new 3 dof spatial parallel mechanism. *Mech Mach Theory* 32(8):895–902
7. Chen YC (1991) Solving robot trajectory planning problems with uniform Cubic B-Splines. *Optim Control Appl Methods* 12:247–262
8. Choi YK, Park JH, Kim HS, Kim JH (2000) Optimal trajectory planning and sliding mode control for robots using evolution strategy. *Int J Robotica* 18(4):423–428
9. Clavel R (1988) DELTA: a fast robot with parallel geometry. In: Proceedings of 18th international symposium on industrial robots, Lausanne, Swiss, pp 91–100
10. De Boor C (2000) A practical guide to spline. Springer, New York
11. Di Gregorio R (2002) A new family of spherical parallel manipulators. *Int J Robotica* 20(4):353–358
12. Eschenauer H, Koski J, Oszyczka A (1990) Multicriteria Design Optimization. Springer, Berlin
13. Ferreira WRB, Carvalho JCM (2011) Robot trajectory planning using cubic B-spline in joint space. In: Proceedings of the international symposium on multibody systems and mechatronics—MuSMe2011, Universitat Politecnica de Valencia, Spain
14. Foley JD, Van-Dam A, Feiner SK, Hughes JF (1990) Computer graphics: Principles and practice, 2nd edn. Addison-Wesley, Massachusetts
15. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning reading. Addison-Wesley, Massachusetts
16. Gosselin C, Angeles J (1988) The optimum kinematic design of a planar three-degree-of-freedom parallel manipulator. *ASME J Mech Transm Autom Des* 110:35–41
17. Haupt RL, Haupt SE (1998) Practical genetic algorithm. Wiley, New York, pp 25–48
18. Hearn D, Baker MP (1994) Computer graphics. Prentice Hall, New Jersey p 07632

19. Holland JH (1975) Adaptation in natural and artificial systems. MIT Press, Illinois Genetic Algorithm Laboratory. IlliGAL, University of Illinois at Urbana-Champaign, pp 11–147
20. Houck CR, Joinez JA, Kay MG (1995) A genetic algorithms for function optimization: a Matlab Implementation. In: NCSU-IE technical report, University of North Caroline, USA
21. Johnson CG, Marsh D (1999) Modelling robot manipulators with multivariate B-splines. *Int J Robotica* 17(3):239–247
22. Karouia M, Hervé JM (2006) Non-overconstrained 3-dof spherical parallel manipulators of type: 3-RCC, 3-CCR, 3-CRC. *Int J Robotica* 24(1):85–94
23. Kosinska A, Galicki M, Kedzior K (2003) Design of parameters of parallel manipulators for a specified workspace. *Int J Robotica* 21(5):575–579
24. Lin CS, Chang PR, Luh JYS (1983) Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *IEEE Trans Automat Contr* 28:1066–1073
25. Merlet JP (2005) Parallel robots. Springer, Dordrecht
26. Merlet JP, Gosselin C (1991) Nouvelle architecture pour manipulateur parallèle à six degrés de liberté. *Mech Mach Theory* 26(1):77–90
27. Miller K (1995) Experimental verification of modelling of delta robot dynamics by direct application of hamilton's principle. In: Proceedings of IEEE international conference on robotics and automation—ICRA'95, Nagoya, pp 532–537
28. Parenti-Castelli V, Di Gregorio R, Bubani F (1999) Workspace and optimal design of a pure translation parallel manipulator. In: Proceedings of XIV national congress on applied mechanics—AIMETA '99, Como, paper no. 17
29. Park FC, Kim J, Bobrow JE (1999) Algorithms for dynamics—based robot motion optimization. In: Proceedings of 10th world congress on the theory of machines and mechanics—IFToMM, Oulu, pp. 1216–1221
30. Oliveira LS, Saramago SFP (2010). Multiobjective optimization techniques applied to engineering problems. *J Brazilian Soc Mech Sci Eng* XXXII:94–104
31. Oliveira PJ, Saramago SFP, Carvalho JCM, Carbone G, Ceccarelli M (2007) An optimum path planning for cassino parallel manipulator by using inverse dynamics. *Robotica (Cambridge)* 26:229–239
32. Oszczka A, (1981) An approach to multicriterion optimization for structural design. In: Proceedings of international symposium on optimum structural design, University of Arizona
33. Ottaviano E, Gosselin CM, Ceccarelli M (2001) Singularity analysis of CaPaMan: a three-degree of freedom spatial parallel manipulator. In: IEEE International Conference on Robotics and Automation ICRA2001, Seoul, pp 1295–1300
34. Piegl L, Tiller W (1997) The nurbs book. Springer, New York
35. Rogers DF (2001) An introduction to nurbs. Morgan Kaufmann Publishers, San Diego
36. Santos RR, Saramago SFP, Steffen V Jr (2008) Robot path planning in a constrained workspace by using optimal control techniques. *Multibody Sys Dyn* 19:159–177
37. Santos RR., Saramago SFP, Steffen V Jr (2010) Optimal task placement of a serial robot manipulator for manipulability and mechanical power optimization. *Intell Inf Manag* 2:512–525
38. Saramago SFP, Steffen V Jr (2001) Trajectory modeling of robots manipulators in the presence of obstacles. *Kluwer Acad J Optim Theory Appl* 110(1):17–34
39. Saramago SFP, Ceccarelli M (2002) An optimum robot path planning with payload constraints. *Int J Robotica* 20:395–404
40. Saramago SFP, Ceccarelli M (2004) Effect of some numerical parameters on a path planning of robots taking into account actuating energy. *Mech Mach Theory* 39(3):247–270
41. Shin KG, Mckay ND (1986) A dynamic programming approach to trajectory planning of robotic manipulators. *IEEE Trans Automat Contr* 31(6):491–500
42. Stewart D (1965) A platform with six degrees of freedom. *Proc Inst Mech Eng Lond*, 180:371–386
43. Tsai LW, Stamper R (1996) A parallel manipulator with only translational degrees of freedom. In: Proceedings of the ASME DETC, Irvine, Paper DETC1996/MECH-1152
44. Wang J, Gosselin CM (1997) Kinematic analysis and singularity representation of spatial five-degree-of-freedom parallel mechanisms. *J Robotics Syst* 14(2):851–869

45. Wolovich WA (1985) Robotics: basic analysis and design. Holt, Rinehart and Winston, New York
46. Yang X, Chen Z (2005). A new high precision fitting approach for NURBS tool paths generation. In: Proceedings of ASME international design engineering technical conference and computers and information in engineering, pp 255–262
47. Zeghloud S, Blanchard B, Pamanes JA (1994) Optimization of kinematics performances of manipulators under specified task conditions. In: Proceedings of 10th CISM-IFTOMM symposium—Romansy, vol 10, Gdansk, pp 247–252
48. Zhao J, Bai SX (1999) Load distribution and joint trajectory planning of coordinated manipulation for two redundant robots. *Mech Mach Theory* 34:1155–1170

# Chapter 13

## Grasp and Motion Planning for Humanoid Robots

Markus Przybylski, Nikolaus Vahrenkamp, Tamim Asfour  
and Rüdiger Dillmann

**Abstract** The capability of humanoid robots to grasp objects is a key competence for their successful application in human-centered environments. We present an approach for grasping daily objects consisting of offline and online phases for grasp and collision-free motion planning. The proposed method generates object-related sets of feasible grasping configurations in an offline phase that are being used for online planning of grasping motions on a humanoid robot. Generating force-closure (FC) grasps on 3D objects is considered to be a hard problem, since many parameters, such as hand kinematics, object geometry, material properties, and forces have to be taken into account, making the space of possible candidate grasps too large to search exhaustively. We believe that the key to find stable grasps in an efficient manner is to use a special representation of the object geometry that can be easily analyzed. In this chapter, we present a novel grasp planning method that evaluates local symmetry properties of objects to generate only candidate grasps that are likely to be of good quality. We achieve this by computing the medial axis which represents symmetry properties of 3D objects by inscribing spheres of maximum diameter into the original shape. Our grasp planner performs offline analysis of the object's medial axis and generates geometrically meaningful candidate grasps. These are then tested for FC in order to create sets of feasible grasps. The resulting grasp sets are used during the online phase for planning collision-free grasping motions with the IK-RRT approach. In contrast to classical motion planning algorithms related to Rapidly exploring Random Trees (RRT), the IK-RRT planner does not rely on one specific goal configuration, but it implicitly uses a goal region in configuration space that is implied by a set of potential grasping configurations in workspace. By using efficient IK-solvers to

---

M. Przybylski (✉) · N. Vahrenkamp · T. Asfour · R. Dillmann  
Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT),  
Adenauerring 2, 76131 Karlsruhe, Germany  
e-mail: Markus.przybylski@kit.edu

sample potential goal configurations during planning, the IK-RRT planner is able to efficiently compute collision-free grasping motions for high-dimensional planning problems. Further, an extension to bimanual grasping problems is discussed and evaluations on the humanoid robot ARMAR-III [3] are performed.

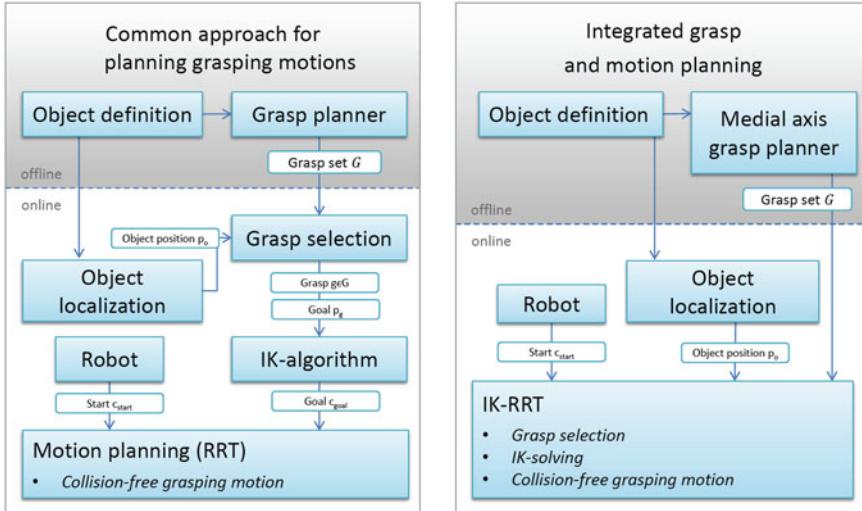
### 13.1 Introduction

The increasingly aging society will benefit from intelligent domestic robots that are able to assist human beings in their homes. The ability to grasp objects is crucial to many supporting activities a service robot might perform, such as serving a drink, tidying up or giving water to the flowers, for example. Human beings perform grasps intuitively on almost any kind of object. In contrast, grasping is a challenging problem for robots. Modern robot hands have many DoF, making grasp planning a high-dimensional problem. On top of that, the object's shape and material properties have to be taken into account which adds another layer of complexity to the planning problem. These issues lead to a search space which is too large to be searched exhaustively. Therefore, a grasp planning algorithm should exploit all possible clues to reduce the search space and to restrict force-closure testing to a carefully chosen set of geometrically meaningful candidate grasps. In this chapter, we show that we can use the medial axis to generate promising candidate grasps based on the symmetry properties of the object.

Grasping becomes even more difficult for robots when considering bimanual tasks. When performing everyday manipulation tasks, such as putting plates in a cabinet or loading a dishwasher, humans often re-grasp the objects they manipulate. Having two arms allows people to reach for an object with one arm and place it with the other, effectively increasing the reachable space without moving in the workspace. If humanoid robots are to exploit their two-armed capabilities, they must possess computationally efficient algorithms for grasping, re-grasping, and dual-arm tasks.

Because humanoid robots are meant to operate in cluttered domestic environments, planning algorithms are needed to generate collision-free trajectories. However, planning a reaching or a re-grasping motion requires choosing a feasible grasping pose with respect to an object to manipulate and finding a configuration of the robot's joints which places the robot's end effectors at this pose. Thus, the planning algorithm must decide which of the feasible grasping poses should be selected and determine the robot's joint configuration for that pose. In the case of re-grasping, a suitable object position which allows grasping by the second hand must also be calculated.

A classical step-wise planning approach is depicted in Fig. 13.1 on the left. In an offline phase a grasp planner is used to pre-compute a set of potential grasping configurations. In the online phase, this set of grasps  $G$  is employed to select a feasible grasp  $g \in G$  with respect to the localized object position. The target pose  $p \in SE(3)$  is calculated by applying the object-related grasping pose  $g$



**Fig. 13.1** Planning grasping motions with a classical step-wise approach (*left*) and with the IK-RRT approach (*right*)

to the object's pose in workspace. By solving the Inverse Kinematics (IK) problem, a goal configuration  $c_{goal}$  can be computed, which is passed to the motion planning algorithm. Several problems may arise with such step-wise approaches. The selection of the grasp  $g \in G$  has to be done without knowing if the  $g$  is reachable, i.e. a collision-free IK solution exists. Further, there is usually no guarantee that a collision-free grasping motion exists, that brings the end effector from the current position  $c_{start}$  to  $c_{Goal}$ . Determining if such a solution exists is computational as hard as solving the motion planning problem. Hence, heuristics must be used to guess that the grasp selection and the IK solution results in a setup that can be solved by the motion planning approach. As shown in Fig. 13.1 on the right, the IK-RRT approach avoids the discussed problems since the problem of selecting a feasible grasp and of solving the IK are implicitly handled.

## 13.2 Related Work

Many approaches for grasp planning have been developed in the past. Grasp synthesis on the contact level concentrates primarily on finding a predefined number of contact points without considering hand geometry [32]. Some work on automatic grasp synthesis focusses especially on object manipulation tasks [35, 40]. Shimoga [40] presents a survey on measures for dexterity, equilibrium, stability, dynamic behavior, and algorithms to synthesize grasps with these properties. Li et al. [31] recorded grasps for basic objects using motion capturing and used this information

to perform shape matching between the inner surface of the hand and novel objects. The resulting candidate grasps were clustered and pruned depending on the task. Since simulators such as GraspIt! [34], OpenRAVE [16] and Simox [42], and frameworks like ROS [37] have become available it is possible to simulate candidate grasps with robot hand models on object models, where hand kinematics, hand and object geometries as well as physical and material properties and environmental obstacles can be taken into account. In the recent past, many researchers developed grasp planning methods based on these simulation environments. Berenson et al. [7, 9] developed a grasp scoring function that considers not only grasp stability but also takes environmental obstacles and kinematic reachability into account. In Vahrenkamp et al. [45] an integrated grasp and motion planning algorithm is presented where the task of finding a suitable grasping pose is combined with searching collision-free grasping motions. Ciocarlie et al. [13] introduced the concept of eigengrasps which allows for grasp planning in a low-dimensional subspace of the actual hand configuration space. Goldfeder et al. [21] used the eigengrasp planner to build a grasp database containing several hands, a multitude of objects and the associated grasps. They used Zernike descriptors to exploit shape similarity between object models to synthesize grasps for objects by searching for geometrically similar objects in their database. They extended this approach to novel objects [20], where partial 3D data of an object are matched and aligned to known objects in the database to find suitable grasps.

A number of simulator-based approaches to grasp planning rely on shape approximation of 3D object models. The basic idea underlying these approaches is that many objects can be decomposed into component parts that can be represented by simplified geometric shapes. Then rules are defined to generate candidate grasps on these components which allows for pruning of the search space of possible hand configurations. This concept is also known as grasping by parts. The first method in this context was presented by Miller et al. [33] who used boxes, spheres, cylinders, and cones to approximate the shape of the object. However, the user has to perform the decomposition of the object into these primitives manually. Goldfeder et al. [22] presented a method that automatically approximates an object's geometry by a tree of superquadrics and generates candidate grasps on those. Huebner et al. [26] developed an algorithm that decomposes objects into a set of minimum volume bounding boxes. While these approaches significantly reduce the complexity of grasp planning, this comes at a price. Many grasps a human would intuitively use might not be found due to poor object geometry approximation. Especially box decomposition yields only a relatively small number of candidate grasps for an object. This might be too restrictive for a real robot, as obstacles in the environment or kinematic constraints of the robot might turn many grasps infeasible. We believe for these reasons that it is desirable to carefully evaluate geometric information of the object to produce only high-potential candidate grasps, so costly validation of candidate grasps by collision-checking and testing for FC can be reduced to a minimum, and therefore a large number of good grasps can be efficiently generated. We achieve this goal by representing the object's shape by the medial axis, which allows for efficient

generation of geometrically and kinematically reasonable candidate grasps that have a high likelihood to be stable.

Planning collision-free motions for robots with many DoF is known to be a P-Space hard problem [38]. Hence, complete algorithms will suffer from low performance mainly caused by the task of building a representation of  $C_{\text{Free}}$ , the part of the configuration space (C-Space) whose configurations do not result in work space collisions. Instead of generating a representation of  $C_{\text{Free}}$ , probabilistic algorithms may be used to implicitly cover the free space in order to avoid a time consuming computation of  $C_{\text{Free}}$ . Approaches based on rapidly Exploring random trees RRT are widely used in the context of planning grasping and reaching motions for humanoid robots. The basic concept of RRTs and theoretical background can be found in [29] or [30].

Planning grasping motions with pre-defined sets of grasping poses is discussed in Drumwright et al. [8, 17, 46]. These approaches use offline calculated target poses for which IK-solutions are searched during the planning process. The approaches suffer from several limitations, such as steepest descent methods that may get stuck in local minima [17, 46] or the disregard of the orientation [46]. The approach presented here is strongly related to the methods presented in [8] and [43, 44]. The workspace goal regions (WGR) planner, presented in [8], does not rely on explicit grasping poses, but so-called WGR, continuous regions in work-space, are used to define the target for planning. Similar to the work in [10], where target regions are described as functions, such definitions allow to easily setup target regions for symmetric objects. Planning dual-arm motions is addressed in Vahrenkamp et al. [43, 44] where collision-free motions for two end effectors are planned with RRT-based algorithms for bimanual grasping or re-grasping actions. In Gienger et al. [19], object specific task maps are used to simultaneously plan collision-free reaching and grasping motions. The proposed planner uses analytic gradients to optimize the motion costs and the choice of the grasp on the manifold of valid grasps.

### 13.3 Medial Axis Grasp Planning

Grasp planning is the process of finding such hand poses and hand configurations of a given robot hand model with respect to a given object model that result in stable grasps. The complexity of grasp planning arises from the multitude of parameters involved. Modern robotic hands have many DoF that have to be dealt with during grasp planning. Objects to be grasped have many different sizes and shapes. Both robot hands and objects have material properties that have to be taken into account as well. This makes grasp planning a high-dimensional problem which cannot be solved using brute-force search methods. Instead, a grasp planner can benefit from symmetries of an object in order to restrict the search to promising areas of the search space where many potentially stable candidate grasps can be found.

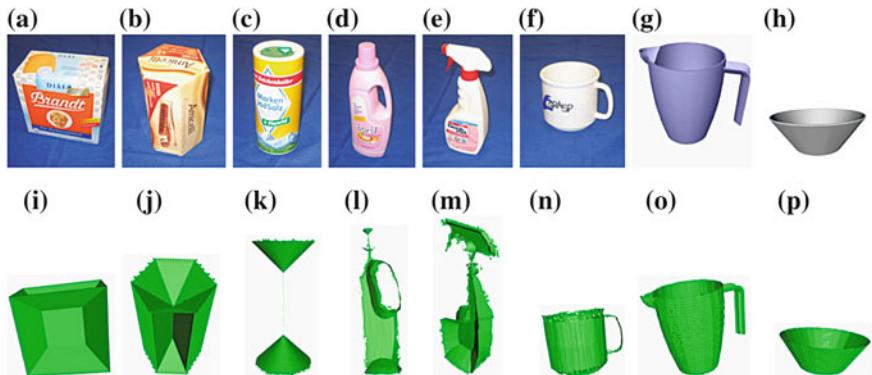
This section is based on our research described in [36]. It is organized as follows. We begin with a brief definition of the medial axis in Sect. 13.3.1. Section 13.3.2 describes necessary preconditions for the computation of the medial axis (see Sect. 13.3.3). Section 13.3.4 describes how the symmetry information contained in the medial axis can be analyzed to generate candidate grasps (see Sect. 13.3.5) which are then evaluated in simulation experiments in Sect. 13.3.6.

### 13.3.1 Medial Axis

Three-dimensional shapes can be approximated by inscribing balls of maximal diameter, i.e. balls that touch the shape boundary at least at two different points. The union of these balls' centers is called the medial axis. Together with the radii of these maximally inscribed balls it is referred to as the medial axis transform. The medial axis transform is a complete shape descriptor, i.e. it contains all necessary information to reconstruct the original object's shape. The medial axis was originally introduced by Blum [11] as a means for biological shape recognition and its computation and applications have been a fruitful area of research [4]. As it provides a compact representation of shapes, their features, and connectivity it has been applied in numerous domains, including CAD model simplification [41], tool-path creation in CAM [24], routing in sensor networks [12], and feature extraction in geometric design [25]. Yet we are not aware of any previous attempt to use the medial axis in grasp planning. In the following, we show that the medial axis contains high-level symmetry information of an object that can be easily exploited to produce big numbers of candidate grasps that are very likely to be stable. This makes it possible to significantly reduce costly validation of candidate grasps. Figure 13.2 shows the objects we use for our experiments and their respective medial axes.

### 13.3.2 Sampling the Object Surface

Before we are able to compute the medial axis, we need to sample points on the object surface. We adopt and modify the method from [7] for this purpose. We calculate an axis-aligned bounding box for the object, sample this bounding box uniformly, and calculate vectors from the sampling points on the bounding box perpendicular to the surfaces of the bounding box. We then use these vectors to perform ray collision on the object. The points where the rays intersect with the object are a good sampling of the object surface. For objects with openings like cups we additionally generate rays perpendicular to an imaginary cylinder surface inside the object to obtain sampling points on the inner surface of the object. We are aware that more sophisticated sampling techniques might be necessary for more complex objects, but as the focus of this work is not on sampling, we leave the evaluation of more advanced sampling methods to future work.



**Fig. 13.2** Our test objects and their medial axes

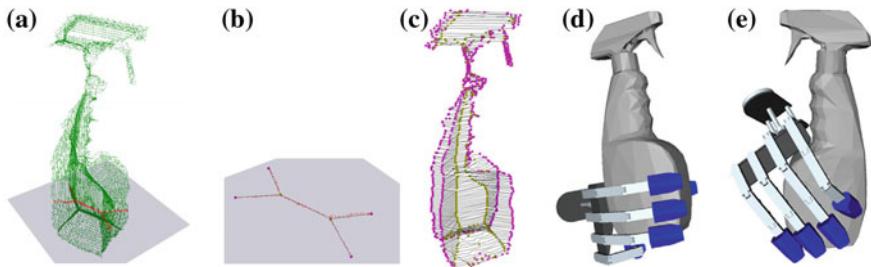
### 13.3.3 Computing the Medial Axis

In this step, we compute the medial axis of the object based on the surface sampling points we generated. The robust computation of the medial axis is quite difficult and still subject to research [4]. Therefore, we use the Tight Cocone software [15] for this purpose. Tight Cocone computes an approximation of the medial axis as a set of points together with a triangulation. Figure 13.3a shows an approximation of a spray bottle's medial axis, where the green dots indicate the inscribed spheres' centers. In the following, we refer to the set of points that approximates the medial axis as  $M$ .

### 13.3.4 Analysis of the Medial Axis

In this step, we analyze the object's medial axis representation we generated in the previous step in order to find familiar structures which can be exploited for generating candidate grasps. Before we continue with the details of our algorithm, we introduce the following terminology which will be used in the remainder of this chapter:

- $p$  denotes a plane that contains a projection of a subset of the medial axis points.
- $M(p)$  denotes the medial axis points projected into a plane  $p$ .
- The medial axis points in a plane  $p$  may be grouped into clusters  $c_i$ .  $M(c)$  denotes the medial axis points associated to a cluster  $c$ . For planes  $p_i$  that contain only one cluster,  $M(c)$  and  $M(p)$  may be used as synonyms.
- $H_{\text{conv}}(X)$  denotes the convex hull of a set of points  $X$ .
- $B(H_{\text{conv}}(X)) \subset X$  denotes the subset of points  $x_i \in X$  that lie at the boundary of  $H_{\text{conv}}(X)$ .
- $\text{vol}(H_{\text{conv}}(X))$  denotes the volume of  $H_{\text{conv}}(X)$ .



**Fig. 13.3** Important steps of our algorithm: medial axis points of a spray bottle with one slice highlighted in red (a), minimum spanning tree of one slice (b), an object as a collection of minimum spanning trees (c), example grasp resulting from one minimum spanning tree (d), example grasp with tilted approach direction by taking into account several slices (e)

- $d_{\min}(X, \text{CoG}(X))$  denotes the minimum distance of all points  $x_i \in X$  to the center of gravity (CoG) of a cluster  $c$ .

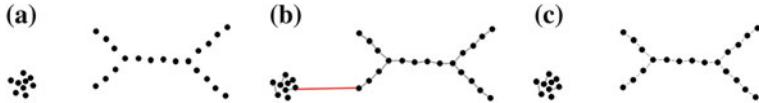
In the proposed method we will also use methods from graph theory. In this context we will use the following terms:

- $\deg(v)$  is the degree of a vertex  $v \in V$  of a graph  $G = (V, E, w)$  with a set of vertices  $V$ , a set of edges  $E$ , and a weight function  $w$ .
- A *branching vertex* is a vertex  $v \in V$  with  $\deg(v) \geq 3$ .
- $\text{MST}(G)$  is the minimum spanning tree of a graph  $G$ .  $\text{MST}(c)$  denotes the minimum spanning tree of the induced subgraph  $G_0 = (V_0, E_0, w_0)$  of  $G$  where  $V_0$  contains all medial axis points of the cluster  $c$ .

### 13.3.4.1 Subdividing the Medial Axis Point Cloud

As a complete medial axis is rather difficult to interpret, we subdivide it into slices and analyze each slice of the medial axis individually. Therefore, we define a set of equidistant planes  $p_i$  that are parallel to a virtual supporting surface beneath the object. The distance between two adjacent planes is a constant which has to be chosen by the user. We obtain slices  $M(p_i)$  of  $M$  by projecting each point  $x_i \in M$  into its nearest neighbor plane  $p$ . A projection plane  $p_i$  and the corresponding slice  $M(p_i)$  of the medial axis are illustrated in Fig. 13.3a. Figure 13.4a shows a different example of a slice of the medial axis.

We interpret the medial axis data in a projection plane  $p$  as a weighted complete graph  $G = (V, E, w)$ , where the set of vertices  $V$  is given by the medial axis points  $M(p)$  and the weight function  $w$  is defined by the pairwise Euclidean distances of the points of the medial axis in the projection plane  $p$ . In order to determine each point's nearest neighbors we compute the minimum spanning tree  $G_{\text{MST}} = \text{MST}(G)$  of this graph and obtain a tree structure of the medial axis data in each projection plane



**Fig. 13.4** Processing a slice of the medial axis: points of a single slice (a) MST connecting all points with their respective nearest neighbors (b) and two clusters obtained by pruning an edge with a distance weight  $w > d_{\text{cut}}$  from the MST(c)

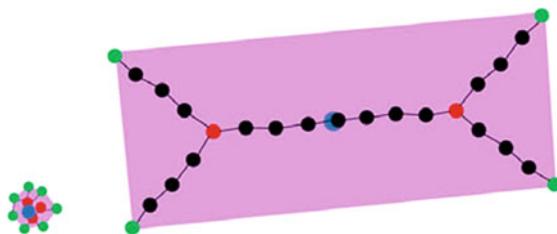
(see Fig. 13.4b). If the medial axis of the object branches projection planes  $\pi_i$  exist where the points of the medial axis form clusters  $c_i$ . We detect these clusters by pruning all edges of the minimum spanning tree that have a distance weight  $w$  exceeding a certain cutoff threshold  $d_{\text{cut}}$  (see Fig. 13.4c). Then each cluster  $c$  has its own  $\text{MST}(c)$  which is further analyzed.

#### 13.3.4.2 Extracting Structural Information from the Medial Axis

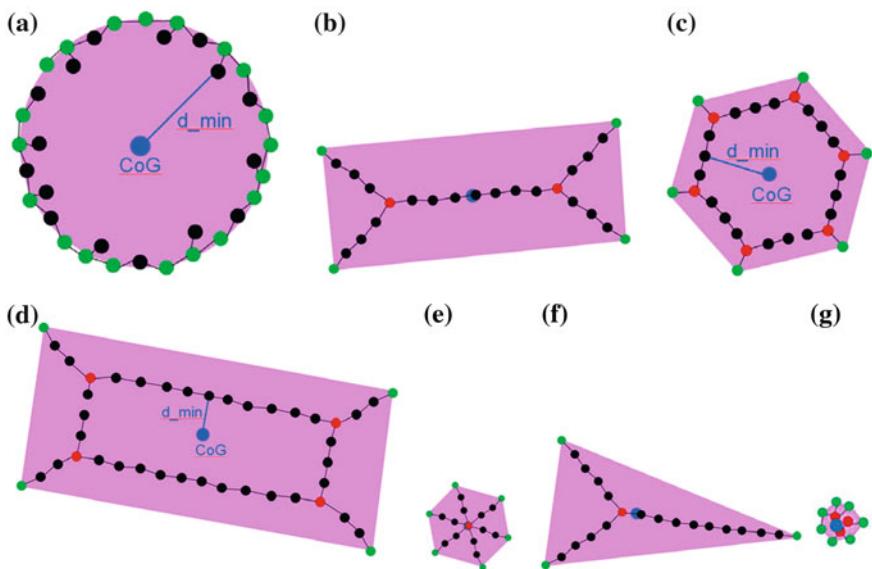
For each cluster  $c$  we identify the structure of its medial axis points  $M(c)$  by evaluating its minimum spanning tree  $\text{MST}(c)$  and the measures defined above (for an illustration see Fig. 13.5). We introduce the following structures:

- *Circle*. We classify a cluster  $c$  as a circle if  $d_{\min}(M(c), \text{CoG}(c))$  exceeds a threshold  $d_{\text{thresh}}$  and there are more than  $k_1$  points at the boundary of its convex hull, i.e. if  $|B(H_{\text{conv}}(M(c)))| > k_1$  holds true, where  $k_1$  is a constant (see Fig. 13.6a).
- *Star with ring*. In contrast, we classify a cluster as a *star with ring* if  $d_{\min}(M(c), \text{CoG}(c)) > d_{\text{thresh}}$  but  $|B(H_{\text{conv}}(M(c)))| < k_2$ , where  $k_2$  is a constant (see Fig. 13.6c, d).
- *Tree*. We distinguish two types of trees. If  $\text{MST}(c)$  contains exactly one branching vertex we call  $c$  a cluster with a *star* (see Fig. 13.6e, f). If  $\text{MST}(c)$  contains exactly two branching vertices we call  $c$  a cluster with a *preference direction* (see Fig. 13.6b).
- *Symmetry axis element*. If  $|B(H_{\text{conv}}(M(c)))| < k_2$  and the volume of the convex hull is very small, i.e.  $\text{vol}(H_{\text{conv}}(M(c))) < k_3$ , we classify the structure as a symmetry axis element (see Fig. 13.6g).

We give a short motivation and some example values for the constants introduced above. For this work, we choose  $k_1 = 40$ ,  $k_2 = 10$ ,  $k_3 = 0.0001$ ,  $d_{\text{thresh}} = d_{\text{cut}} = 5\rho$ , where  $\rho$  is the sampling resolution for sampling the object's surface in the first step of our algorithm.  $d_{\text{thresh}}$  and  $d_{\text{cut}}$  depends on  $\rho$  because the density of the medial axis points roughly equals the surface sampling resolution  $\rho$ . The choice of  $k_2$  is motivated by the idea that the structures with plane symmetries we want to exploit for grasp planning typically have a limited number of vertices at the boundary of their convex hull. Structures with more than 10 such vertices rather indicate the lack of such plane symmetries but instead suggest the presence of a symmetry axis. Or differently put: if you keep adding vertices to a convex



**Fig. 13.5** Two clusters in a slice of the medial axis with their MSTs and several features highlighted in color. Pink convex hull  $H_{\text{conv}}(c)$  of a cluster  $c$ . Green vertices at boundary of the convex hull  $B(H_{\text{conv}}(c))$ . Red branching vertices of the MST. Blue cluster's center of gravity (CoG)



**Fig. 13.6** Structures in slices of the medial axis Ring (a) preference direction (b) stars with ring (c,d) stars (e,f) symmetry axis element (g)

polygon you will eventually end up with a circle. Theoretically, it is possible to choose  $k_1 = k_2$ , which we do not test in our experiments. Instead, we empirically choose  $k_1$  with  $k_1 > k_2$  in order to have a stronger discrimination between circles and stars with ring. The reason for introducing  $k_3$  is noise and discretization errors in the medial axis approximation. In this theory, the presence of a symmetry axis should be indicated by the presence of single isolated medial axis points in a slice. In practice, however, there will be a set of medial axis points clustered in a very small area.

Figures 13.3b and 13.6b show minimum spanning trees  $\text{MST}(c)$  with a preference direction. Figure 13.3c depicts an object as a collection of  $\text{MST}(c_i)$  with highlighted branching and end vertices. Using the structural information we

extracted from the medial axis we are able to generate promising candidate grasps in the next step.

### 13.3.5 Generating Candidate Grasps

In this section, we present a set of heuristics describing how promising candidate grasps can be generated based on the structural information extracted from the medial axis in the previous step. Before we describe these heuristics, we explain which parameters we use to describe a candidate grasp. In [7], the authors defined a set of parameters describing a candidate grasp:

- An approach direction  $P_d$  of the hand.
- A 3D target point  $P_t$  on the surface of the object the hand is approaching.
- A roll angle  $P_r$  of the hand about the approach direction.
- A vector of joint values  $P_p$  defining the preshape of the hand.

For generating candidate grasps, the method described in [7] samples the object's surface by casting rays from a bounding box of the object. Each point where a ray intersects with the object is used as a target point  $P_t$  and the surface normal at  $P_t$  is used as an approach direction  $P_d$ . The palm normal of the hand is orientated toward  $P_d$  and moves toward the object until collision occurs. Then the hand is closed in order to wrap the fingers around the object. For each approach direction, a set of different roll angles  $P_r$  and preshapes  $P_p$  is tested. We adopt the grasp parameters from [7] but we use a different policy to generate them. Instead of sampling the object's surface, we carefully choose approach directions  $P_d$  and target points  $P_t$  based on the structural information we extracted from the medial axis. In addition to that, we are able to choose such roll angles  $P_r$  and preshapes  $P_p$  that have a high probability to result in a stable grasp when the hand collides with the object and the fingers wrap around it.

In this context, we would like to emphasize the importance of the hand's roll angle and preshape. One of the most basic classes of grasps for human hands are grasps where the thumb as the first virtual finger opposes the other fingers which form a second virtual finger. In our opinion, it is therefore a promising strategy to preferably generate geometrically meaningful candidate grasps where these two virtual fingers are likely to contact the object at opposing sides. Accordingly, if not stated otherwise, we suggest to use a parallel preshape for most of the candidate grasps we generate in the following and adjust the roll angle of the hand with respect to the symmetry information from the medial axis. We would like to state that favoring candidate grasps with two opposing virtual fingers does not impose any restriction on the kinematics of the robotic hand to be used. To the extent of our knowledge, all currently available robotic hands are capable of moving the thumb to a position opposed to the other fingers.

In the following, we present various heuristics to generate candidate grasps depending on the structures we detected in slices and clusters of the medial axis.

The general approach is always to look at several adjacent planes  $p_i$  and to derive promising approach directions and roll angles for the hand from the information in these  $p_i$ .

- *Clusters with a preference direction.* If in a plane  $p$  we have a cluster  $c$  with a preference direction, there are exactly two branching vertices  $v_1, v_2$ . The vectors  $a_1 = v_2 - v_1$  and  $a_2 = v_1 - v_2$  are very promising approach directions  $P_d$ . For  $a_1$  we use  $v_1$  and for  $a_2$  we use  $v_2$  as the target point. An example of a resulting grasp is shown in Fig. 13.3d. If there are neighbor planes to  $p$  that contain also clusters with preference direction we can use the coordinates of their branching vertices to calculate tilted approach directions  $a_{1,\text{tilted}}$  and  $a_{2,\text{tilted}}$  to make our hand better conform to the object's surface (see Fig. 13.3e). We also use this information to choose roll angles  $P_r$  that make it probable that the fingers will be able to wrap around the object.
- *Clusters with a star structure.* If a cluster contains a star structure, we proceed in a similar way but we generate an approach direction  $P_d$  from every spike of the star to its branching vertex. There are two exceptions to this rule. For stars with three spikes we check if the star is symmetric with regard to one of its spikes. In this case, we treat the star as a structure with a preference direction. The second exception is for stars with four spikes. If every two opposing spikes in such a star define an axis of symmetry the profile of the object in this plane or cluster is likely to be a square for which we use the angle bisecting vectors of the spikes as approach directions.
- *Clusters with a symmetry axis element.* Clusters with a symmetry axis element structure indicate that the profile of the object in this cluster is circular. If we find a cluster of this type we search in adjacent planes for more clusters with a symmetry axis element structure and use this information to generate approach directions perpendicular to the axis defined by the symmetry axis element structures.
- *Clusters with a circle structure.* For clusters with a circle structure, we search for clusters with a circle structure in adjacent planes and generate approach directions perpendicular to the symmetry axis defined by the mean coordinates of the  $M(c_i)$  in these planes.
- *Treating two clusters together.* In general, candidate grasps can be individually generated for each cluster of a plane. Yet if there are exactly two clusters that are located close to each other the process of generating individual candidate grasps for each cluster will probably produce many unsuccessful candidate grasps. Therefore, we treat both clusters together. If both clusters are small in terms of the volume of their convex hull  $\text{vol}(H_{\text{conv}}(M(c_i)))$  we use the clusters' centers of gravity (COG) as target points and the vectors  $b_1$  and  $-b_1$  connecting the COGs as approach directions. If one cluster is significantly smaller than the other we treat the bigger cluster independently and generate for the smaller cluster only an approach direction that is directed from the COG of the smaller cluster toward the COG of the bigger cluster.

- *Candidate grasps at the top and the bottom of objects.* In order to generate candidate grasps for the top and the bottom of an object, we distinguish a number of different cases that take into account the structure of the medial axis in planes near the top and the bottom of the object. If there are planes with a star with ring structure, followed by planes with star, as for the prismatic chocolates box, or preference direction structure, as for the bread box object, we generate candidate grasps with roll angles  $P_r$  aligned with the preference direction or the spikes of the star. If there are circles or symmetry axis element structures, we generate candidate grasps in such a way that the approach direction coincides with the symmetry axis and use various different roll angles. For these candidate grasps it also makes sense to use a spherical hand preshape if this is supported by the hand model. If circle structures are present we also check by ray collision if the object has an opening. For open objects we generate candidate grasps at the rim of the object with roll angles of the hand that make it probable that at the final position of the hand the rim is between the thumb and the other fingers.
- *Size of the object.* In our attempt to minimize costly testing of candidate grasps we also consider the size of the object with respect to the hand. We use ray collision to roughly estimate the width of the object geometry perpendicular to our approach direction and compare this value to the diameter of the biggest ball our hand can stably grasp. If the structure is too wide for the hand we do not have to generate a candidate grasp as it will fail.

### 13.3.6 Experiments and Results

In this section, we present some experiments for our grasp planning method. We perform experiments on the set of typical household objects depicted in Fig. 13.2. Except for the spray (Fig. 13.2e) and the detergent (Fig. 13.2d) and the cup (Fig. 13.2f), where the object model was generated using a 3D laser scanner [6, 28] all object models are handmodeled. We use the hand model of our humanoid robot ARMAR-III [2] to evaluate our algorithm in OpenRAVE. Each candidate grasp is evaluated by moving the hand with its palm facing the object along the approach direction until collision is detected. Then the fingers of the hand close until all fingers have contact with the object or the joint limits are reached and no finger link can move any more. Then we evaluate grasp quality by computing the commonly used worst-case epsilon measure for FC, as described by Canny and Ferrari [18]. For the force-closure computations we use a friction coefficient of 0.5.

For each candidate grasp we test two different alternatives. The first alternative is the one described above. For the second alternative we move the hand after contact with the object a small distance  $d_{\text{back}}$  away from the object along the approach direction, before we close the fingers. This way, we are able to test candidate grasps where only the finger links have contact with the object, but not the palm. For this kind of grasps we empirically choose  $d_{\text{back}} = 2.5$  cm for all objects, except for the pitcher's handle. This choice of  $d_{\text{back}}$  prevents palm contact,



**Fig. 13.7** Some representative force-closure grasps generated by our algorithm

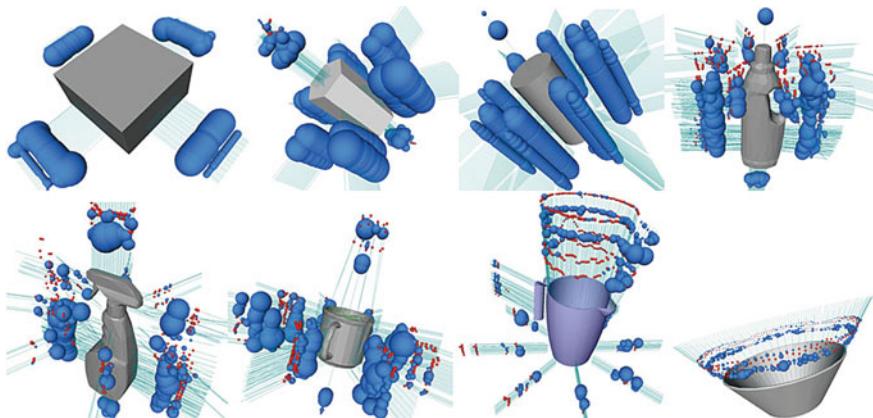
but makes it still probable that more than one link per finger will contact the object. In the special case of the pitcher's handle, we empirically choose  $d_{\text{back},1} = 7 \text{ cm}$  and  $d_{\text{back},2} = 8 \text{ cm}$  in order to enable contacts at the fingertips only, so we can evaluate candidate grasps at the handle. The variables  $d_{\text{back}}$ ,  $d_{\text{back},1}$  and  $d_{\text{back},2}$  have to be chosen empirically, as they depend both on the size and kinematic structure of the hand, as well as on the thickness of the object. Some representative force-closure grasps produced by our algorithm are depicted in Fig. 13.7. We note that in our opinion the grasps produced by our method look quite intuitive for a human being.

Figure 13.8 shows a visualization of the force-closure quality of all candidate grasps generated by our method. The rays indicate the approach directions of the hand toward the object. Each sphere indicates the final wrist position of a candidate grasp. Each blue sphere represents a force-closure grasp. For these grasps the diameter of the sphere is proportional to the force-closure score of the respective grasp, i.e. the biggest spheres represent the grasps with the highest force-closure rating. Red spheres represent grasps that do not fulfill the force-closure criterion.

To demonstrate the advantages of our method, we also compare it to Berenson's grasp planning algorithm [7] that uses only minimal knowledge of the object's geometry. As described above, Berenson's algorithm uniformly samples the object surface and uses the surface normals at the sampled surface points directly as approach directions for the hand. For Berenson's method as well as for our own, we choose a surface sampling resolution  $\rho_1$  of one ray per cm. In case of the bowl, we choose a sampling resolution  $\rho_2$  of one ray per two cm, as the object is very big. Above that, we evaluate grasps with eight different roll angles of the hand around the approach direction, i.e. with roll angles  $\alpha_i$  where

$$a_i = 0.125\pi k, \quad k \in [0, 7]. \quad (13.1)$$

We also test  $d_{\text{back},1} = 0 \text{ cm}$  and  $d_{\text{back},2} = 2.5 \text{ cm}$  for all candidate grasps as also described above for our proposed method. The results of the comparison are given in Table 13.1. We note that while Berenson's method produces far more candidate grasps, our algorithm has a significantly higher likelihood that the



**Fig. 13.8** Force-closure quality of the tested candidate grasps. *Rays* indicate the approach directions of the hand toward the object. *Blue spheres* indicate the wrist positions of the hand where grasps with a positive force-closure rating were found. The diameter of a *blue sphere* is proportional to the force-closure score of the respective grasp. *Red spheres* indicate grasps where the force-closure rating was zero

**Table 13.1** Candidate grasps tested and percentage of force-closure (FC) grasps found

Objects	Our method		Method from [7]	
	Candidates	FC (%)	Candidates	FC (%)
Bread box	632	86.2	13,440	15.5
Prismatic box	1,344	90.7	8,512	36.0
Salt can	2,144	96.9	7,904	45.7
Detergent	1,996	65.9	12,672	26.2
Spray	1,304	55.1	11,200	21.2
Cup	1,428	59.5	6,688	37.0
Pitcher	1,124	47.0	15,504	25.9
Salad bowl	504	68.5	13,648	4.5

generated grasps are force-closure. The differences in efficiency are especially striking for objects like the bread box (Fig. 13.2a) and the salad bowl (Fig. 13.2h) which are too big to be grasped from arbitrary directions or with arbitrary roll angles of the hand. For most other objects in our experiments the proposed method still outperforms Berenson's method by a factor of two or more in terms of the fraction of force-closure grasps among the generated candidate grasps. This is especially interesting because it shows that our method significantly reduces the computation time overhead for collision detection and force-closure testing.

## 13.4 Planning Collision-Free Grasping Motions

In the previous section, an efficient approach was presented for building sets of potential grasps for an object. These grasp sets are stored in a database, which can be accessed by the robot while performing grasping tasks online. Since humanoid robots operate in cluttered environments collision-free grasping motions have to be planned. Therefore, we propose the IK-RRT algorithms that can be used for single arm as well as for bimanual tasks. Since the planner depends on an efficient IK solver, we show how reachability analysis together with analytic IK-solvers can be used to solve complex IK-problems for arbitrary kinematic chains.

### 13.4.1 Reachability Analysis

The use of reachability information can speed up randomized IK-solvers as shown in Sect. 13.4.2. The reachability data is represented by a grid of voxels in 6D pose space. Each voxel holds information about the probability that an IK query can be answered successfully [4, 17, 23]. It can be used to quickly decide if a target pose is too far away from the reachable configurations and therefore if a (costly) IK solver call is likely to return a solution.

When considering a discretized workspace representation, the reachability of a voxel  $v$  cannot be expressed as a binary value, since there might be poses  $p_r \in v$  that are reachable while other poses  $p_0 \in v$  are not. Hence, the reachability of a voxel can just give a hint (or a probability) that a pose inside that voxel is reachable. When considering the reachability of a voxel as a probability that a pose is reachable, the discretized reachability data can be interpreted as a frequency distribution known from descriptive statistics and therefore we use the term reachability distribution in this work to name the discretized reachability data that is represented by entries of 6D workspace voxels.

Reachability data can be determined by solving a large number of IK requests and counting the number of successful queries for each voxel. Another way of generating a representation of the reachable workspace is to randomly sample the joint values while using the forward kinematics to determine the pose of the end effector and thus the corresponding 6D voxel [5]. An analytic approach of generating a representation of the reachability is presented by Kee and Karwowski [28].

For further speedup we propose a gradient descent approach which can be used to optimize the search for a graspable object pose. If an object pose was found, where the corresponding reachability entry lies above a threshold, we apply a search for a local maximum. This is done by checking the neighbor voxels of the reachability distribution. If there is a voxel with a higher reachability entry and the new pose is collision free, the object 6D position is moved toward this voxel by the extent of the corresponding dimensions of a voxel. The new position then lies inside the voxel with the higher reachability entry. This is repeated until there are

no neighbors with higher entries which means the position is at a local maximum of the discretized reachability distribution.

To avoid losing the probabilistic completeness by applying the discretized reachability distributions and the gradient descent approach, these extensions to the original algorithm are only used with some probability during the search loop. Thus, the theoretical behavior of the IK-solvers remains untouched while the performance can be considerably increased.

### 13.4.2 Inverse Kinematics

Finding a robot configuration that places the end effector at a given pose is known as the (IK) problem. Though analytical solution of IK is possible for some manipulators which have not more than six DoF [14], the IK solver for the humanoid robot ARMAR-III [2] has to consider two seven DoF arms and a three DoF hip.

The IK-solvers we present in this work use a combination of gradient descent in pre-computed reachability distributions and random-sampling of free parameters ([Sects. 1.4.2.1](#) and [1.4.2.2](#)) and we show how to apply our approach to one or two arm queries with fixed or varying object poses. In the case of a varying object pose, the search for a collision-free and graspable object pose is part of the IK task and the result consists of a robot configuration and a 6D object pose. This IK approach is extremely efficient, requiring only a few milliseconds to solve a query as opposed to iterative IK algorithms (e.g. Sciavicco et al. [39]) which could be trapped by joint limits. Further details can be found in [43, 44].

Please note that the presented IK-solvers do not consider any stability constraints. Stability, as well as other constraints, can be incorporated during IK sampling, where configurations that do not fulfill the constraints are discarded (see [Algorithm 1](#)).

#### 13.4.2.1 Single Arm IK-Solver

To reach and grasp a fixed object with one hand, the IK problem has to be solved. In the case of ARMAR-III, the operational space can be increased by additionally considering the three hip joints of the robot, which leads to a kinematic chain with 10 DoF. Our approach to solving the IK problem uses a combination of gradient descent in reachability distribution and random sampling of free parameters.

Typically, an arm of a humanoid robot consists of six to eight DoF and is part of a more complex kinematic structure. If an analytical method exists for solving the IK problem for six DoF of an arm, a randomized algorithm can be constructed which randomly samples the preceding joints (such as the hip) and uses the analytical IK solver for determining the final arm configuration. This probabilistic approach increases the operational space of the robot arm and is suitable for randomized planning algorithms.

For ARMAR-III we use a specialized analytic approach for solving the seven DoF IK problem for one arm where all possible elbow positions are computed and, depending on the parameterization, the best one is chosen [1]. If there are multiple solutions, the behavior can be adjusted. Either the one with the lowest accumulated joint movement or a random solution out of the set of possible results is selected. In addition to this IK solving, it is desirable to consider the joints of the robot's hip since the reachable space increases significantly when using additional DoF. In this case the three hip joints of ARMAR-III are randomly sampled until an IK query is successfully answered.

If a configuration was found which brings the end effector to the desired pose, the IK solution has to be checked against self-collisions and collisions with obstacles in order to avoid invalid configurations. If the collision checker reports a collision, the solution is rejected and the search is continued.

The approach is probabilistically complete, which means if time goes to infinity, the probability of finding a solution will go to unity if one exists. To avoid endless runtimes, the search for an IK solution is stopped after a specific number of tries and it is assumed that there is no valid result.

The most convenient kinematic chain for reaching or grasping an object with ARMAR-III consists of the three hip joints followed by seven arm joints. This 10 DoF kinematic chain leads to a large reachable space and thus enables the robot to perform grasping and manipulation operations without moving the robot's mobile platform.

Since the reachability representation is linked to the shoulder, it moves when setting the three hip joints randomly in the search loop of the probabilistic IK solver. For this reason, the target pose, which is given in the global coordinate system, is transformed to the shoulder coordinate system and the corresponding voxel of the resulting pose is determined. The analytical IK solver is only called if the entry of this voxel is greater than zero (or a given threshold). When considering a fixed pose  $p \in SE(3)$ , the IK solver of Algorithm 1 can be used to calculate a solution  $c_{\text{result}}$  covering the free parameters ( $K_{\text{free}}$ ) and the joints that can be handled by the analytic IK solver ( $K_{\text{IK}}$ ).

---

**Algorithm 1:** *SolveIkHybrid( $K_{\text{free}}$ ,  $K_{\text{IK}}$ ,  $p_{\text{target}}$ )*


---

```

1 while ( $\text{!TimeOut}()$ ) do
2    $c_{\text{free}} \leftarrow \text{SampleFreeParameters}(K_{\text{free}});$ 
3    $\text{SetRobotConfiguration}(K_{\text{free}}, c_{\text{free}});$ 
4   if ( $\text{Reachability}(K_{\text{IK}}, p_{\text{target}}) > 0$ ) then
5      $c_{\text{IK}} \leftarrow \text{CallIKSolver}(K_{\text{IK}}, p_{\text{target}});$ 
6      $c_{\text{result}} \leftarrow \{c_{\text{free}}, c_{\text{IK}}\};$ 
7     if ( $\text{!Collision}(c_{\text{result}})$ ) then
8       return  $c_{\text{result}};$ 
9   end
10 end
11 return NULL;

```

---

**Table 13.2** Performance of the 10 DoF IK-solvers

	Without obstacle		With obstacle	
	Average runtime (ms)	# IK calls	Average runtime (ms)	# IK calls
Without reach. distribution	1,404	101.9	2,880	217.3
With reach. distribution	60	6.1	144	13.6

## Evaluation

To measure the performance of the 10 DoF IK solver, a wok with 15 associated grasping poses is set to a random pose in front of the robot. In this example,  $K_{\text{free}}$  covers three hip joints and  $K_{\text{IK}}$  covers seven arm joints of ARMAR-III. The *GraspIK*-solvers with and without support of reachability analysis are called in order to find a valid configuration for bringing the end effector to one of the 15 grasping poses. An example result of the IK solver in a partly blocked scene is shown in Fig. 13.10 (left). The results of Table 13.2 are determined by computing the averages of 100 IK queries with randomly generated object poses.<sup>1</sup> The average runtime and the number of calls of the analytical Seven DoF IK solver are given for setups with/without reachability distributions and in scenes with/without obstacles (further details can be found in Vahrenkamp et al. [44, 45]). It turns out that the use of the reachability information speeds up the IK solver enormously and it allows the use of these approaches in real-world applications.

---

**Algorithm 2:** *GraspIK*( $K_{\text{free}}$ ,  $K_{\text{IK}}$ ,  $G$ ,  $p_o$ )

---

```

1  $P \leftarrow \emptyset;$ 
2  $K \leftarrow K_{\text{free}} \cup K_{\text{IK}};$ 
3 foreach ( $g \in G$ ) do
4    $p_{\text{grasp}} \leftarrow g \cdot p_{\text{object}};$ 
5   if ( $\text{Reachability}(K, p_{\text{grasp}}) > 0$ ) then
6      $P \leftarrow P \cup \{p_{\text{grasp}}\};$ 
7 end
8 if ( $P == \emptyset$ ) then
9   return NULL;
10 while ( $\text{TimeOut}()$ ) do
11    $p_{\text{grasp}} \leftarrow \text{SampleRandomGraspingPose}(P);$ 
12    $c_{\text{Result}} \leftarrow \text{SolveIkHybrid}(K_{\text{free}}, K_{\text{IK}}, p_{\text{grasp}});$ 
13   if ( $c_{\text{Result}}$ ) then
14     return  $c_{\text{result}};$ 
15 end
16 return NULL;
```

---

<sup>1</sup> These performance evaluations have been carried out on a DualCore system with 2.0 GHz.

### 13.4.2.2 Dual-Arm IK Solver

If the robot should re-grasp or hand-off an object, the search for a valid regrasping configuration includes a collision-free object pose and a valid and collision-free IK-solution for both arms. In case of ARMAR-III this leads to an IK problem, where the combination of the 6D object pose, three hip joints, and seven DoF for each arm results in a 23-dimensional solution vector.

As shown in Algorithm 3, the *RegraspIK* approach is related to a kinematic chain  $K_{\text{free}}$  covering the free joints (the hip joints for ARMAR-III), two kinematic chains  $K_{\text{IK}}$  that cover the arm joints and two sets  $G_{\text{left}}$  and  $G_{\text{right}}$  of potential grasps.

To find a reachable object pose in the task space of the robot, the 6D pose of the object and the configuration of the three hip joints can be sampled randomly until a call of the IK solver is successful for one of the poses. Therefore, the Cartesian position of the object is limited to the extent of the reachable space and the rotation component does not have any restrictions.

---

**Algorithm 3:** *RegraspIK*( $K_{\text{free}}, K_{\text{left}}, K_{\text{right}}, G_{\text{left}}, G_{\text{right}}$ )

---

```

1 while ( $\text{!TimeOut}()$ ) do
2    $p_o \leftarrow \text{SampleRandomObjectPose}();$ 
3    $(c, p'_o, g_{\text{left}}, g_{\text{right}}) \leftarrow \text{ReachableGrasps}(K_{\text{free}}, K_{\text{left}}, K_{\text{right}}, p_o, G_{\text{left}}, G_{\text{right}});$ 
4   if ( $c$ ) then
5     return ( $c, p'_o, g_{\text{left}}, g_{\text{right}}$ );
6 end
7 return NULL;

```

---

The IK solver for finding a suitable regrasping configuration is shown in Algorithm 4. The initial object pose  $p_o$  is optimized by searching the local maximum of the reachability, as exemplarily shown in Fig. 13.9. The *DualArmIK* algorithm is similar to the single arm case (see Algorithm 2), but it considers two grasp sets for both arms.

---

**Algorithm 4:** *ReachableGrasps*( $K_{\text{free}}, K_{\text{left}}, K_{\text{right}}, p_o, G_{\text{left}}, G_{\text{right}}$ )

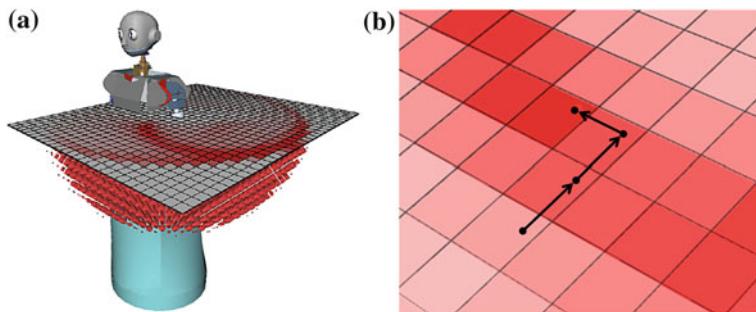
---

```

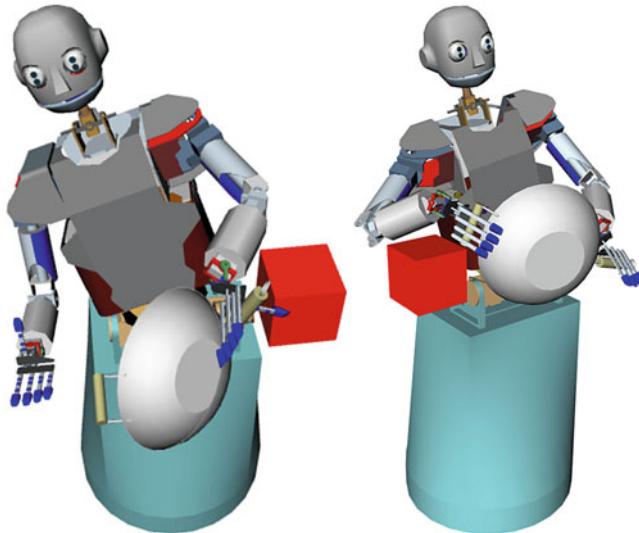
1 while ( $\text{!TimeOut}()$ ) do
2    $g_{\text{left}} \leftarrow \text{RandomGrasp}(G_{\text{left}});$ 
3    $g_{\text{right}} \leftarrow \text{RandomGrasp}(G_{\text{right}});$ 
4   if ( $\text{Reachability}(K_{\text{left}}, g_{\text{left}} \cdot p_o) > t_{EV}$   $\&$ 
       $\text{Reachability}(K_{\text{right}}, g_{\text{right}} \cdot p_o) > t_{EV}$ ) then
5      $p'_o \leftarrow \text{OptimizeGraspingPose}(K_{\text{left}}, K_{\text{right}}, p_o, g_{\text{left}}, g_{\text{right}});$ 
6      $c \leftarrow \text{DualArmIk}(K_{\text{free}}, K_{\text{left}}, K_{\text{right}}, g_{\text{left}} \cdot p'_o, g_{\text{right}} \cdot p'_o);$ 
7     if ( $c$ ) then
8       return ( $c, p'_o, g_{\text{left}}, g_{\text{right}}$ );
9   end
10 end
11 return NULL;

```

---



**Fig. 13.9** **a** A 2D view of the reachability distribution of ARMAR-III. **b** The 2D projection of a gradient descent optimization. The color intensity is proportional to the probability that a pose inside the voxel is reachable



**Fig. 13.10** *Left* example results of the 10 DoF single arm with an obstacle. *Right* the 17 DoF dual-arm IK-solvers. The dual-arm IK algorithm provides a feasible joint configuration and a collision-free object pose

### Evaluation

The average run times of the dual-arm IK-solvers are shown in Table 13.3. Again, the IK solver is queried 100 times in a scene without any obstacles (first two columns) and in a scene with an obstacle (last two columns). The first row shows the solution in case the search for a feasible object pose for grasping is included (23-dimensional problem) and the second row shows the results when the object is already grasped with the left hand and only a configuration for both arms and the

**Table 13.3** Performance of the dual-arm IK-solvers

	Without obstacle		With obstacle	
	Avg runtime (ms)	# IK calls	Avg runtime (ms)	# IK calls
Flexible grasp selection	47	3.3	161	6.5
Object grasped with left hand	162	3.2	220	4.3

hip (17 dimensional) is searched. Here the object is linked to the left hand and less possible grasping combinations for both hands are available for the IK solver and thus the runtime increases. An example result of the dual-arm IK solver is shown in Fig. 13.10 (right).

### 13.4.3 The IK-RRT Planner

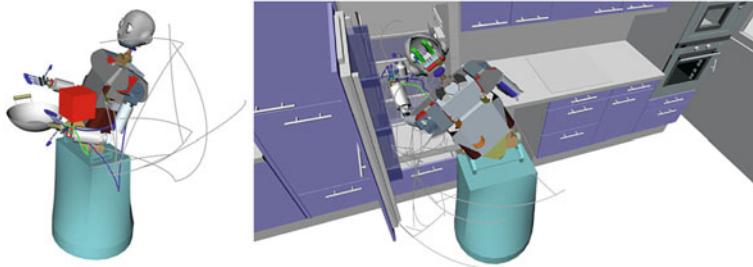
The proposed planning algorithms combine the search for collision-free motions with the search for solutions of the IK problem in one planning scheme [43, 44]. The planners are initialized with a set of grasping poses and feasible target configurations are calculated along with an object pose. The computation of feasible target configurations is done during the planning process and thus the planning is not limited to an incomplete set of targets.

Since sets of potential grasps are used, it is possible to calculate an IK solution for each pose of each grasp candidate in the database and to use this set of configurations as targets for the planning process. This will lead to a planning scheme where the search for solutions is limited to the pre-calculated IK solutions. Since, in general, there are infinite numbers of solutions for the IK problem, the planner could fail although there is a valid motion for an IK solution which was not considered. Furthermore, it can be time-consuming to calculate the IK solutions for every grasping pose in advance. If the feasible grasps are densely sampled, the pre-calculation has to be done for a large number of poses. These problems can be avoided, if the search for valid IK solutions is included in the planning process.

For efficient planning, an IK solver is used in order to generate goal positions during the planning process. The planner uses as input a set of feasible grasping poses, which, combined with the pose of the object, defines a set of target poses. These poses are used as input for the IK solver.

The IK-RRT algorithm works as follows:

- Initialization: the forward part of the Bi-RRT algorithm is initialized with a start configuration, the backward tree is empty until an IK solution is found.
- The planning loop grows the two trees and tries to connect them via an intermediate configuration.



**Fig. 13.11** The results of the IK-RRT planner in a scene with an obstacle (*left figure*) and in a cluttered scene (*right figure*). The solution is marked *blue*; the optimized solution is shown in *green*

- With some probability, a random grasp out of the set of feasible grasps is chosen and a call to the randomized IK solver is performed. When a feasible IK configuration  $q_{IK}$  is found, it is added to the backward tree and the new node is marked as a solution node.

Since the IK search is probabilistically complete for the set of grasps and the RRT-Connect algorithm is known to be probabilistically complete [29], the IK-RRT approach is probabilistically complete. In Fig. 13.11 results of the IK-RRT approach are shown. The original and the optimized solution path are depicted in blue and green.

---

**Algorithm 5:** IK-RRT( $q_{start}, p_{obj}, gc$ )

---

```

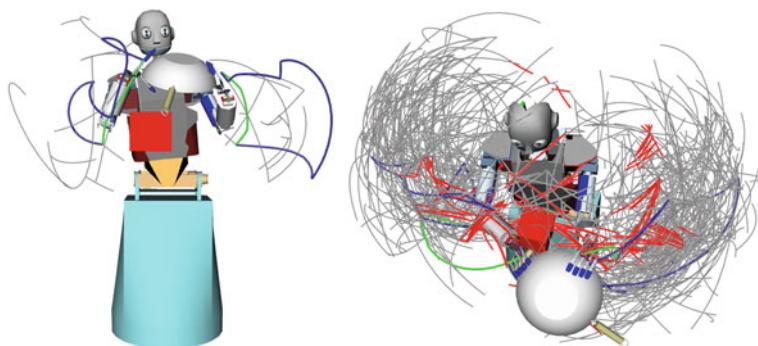
1 RRT1.AddConfiguration( $q_{start}$ );
2 RRT2.Clear();
3 while (!TimeOut()) do
4   if (#IKSolutions == 0 || rand() <  $p_{IK}$ ) then
5      $grasp \leftarrow GetRandomGrasp(gc);$ 
6      $p_{target} \leftarrow ComputeTargetPose(p_{obj}, grasp);$ 
7      $q_{IK} \leftarrow ComputeIK(p_{target});$ 
8     if (!Collision( $q_{IK}$ )) then
9       RRT2.AddConfiguration( $q_{IK}$ );
10    else
11       $q_r \leftarrow GetRandomConfiguration();$ 
12      if (RRT1.Connect( $q_r$ ) & RRT2.Connect( $q_r$ )) then
13         $Solution \leftarrow BuildSolutionPath(q_r);$ 
14        return PrunePath( $Solution$ );
15      end
16    end
17 end

```

---

### 13.4.4 Bimanual IK-RRT

To plan a re-grasping motion with two arms, two problems have to be solved. First, the configuration for handing off the object from one hand to the other hand must be determined. This configuration must bring the object, which is grasped with one hand, to a position where the other hand can apply a feasible grasp. This search also includes choosing which grasp should be applied with the second hand. The configuration is only valid if there are no collisions between the arms, the environment, the object, and the robot. Second, there must exist a collision-free trajectory which brings the arm with the grasped object and the other arm to the re-grasping position. With the IK methods presented in Sect. 13.4.2.1 it is possible to generate feasible configurations for bimanual re-grasping tasks. The search for these configurations can be included in an RRT-based planner as described in Sect. 1.4.3. The dual-arm IK solver is used to generate IK solutions during the planning process. These IK solutions include a valid pose of the object with the corresponding joint configuration of the hip and both arms for grasping the object with both hands. The Algorithm 5 has to be adapted slightly to include the dual-arm IK solver. Instead of a predefined object pose, the object is attached to the kinematic structure of one arm and the IK solver operates on the set of feasible grasps. The resulting Bimanual IK-RRT planner can be used for computing collision-free re-grasping trajectories in cluttered environments. In Fig. 13.12 (left) the results of the Bimanual IK-RRT are shown for an exemplary scene. For comparison the results of the J<sup>+</sup>-RRT (see [43, 44]) are shown on the right. It can be seen, that the resulting search tree of the IK-RRT approach is significantly smaller than the corresponding search tree of the J<sup>+</sup>-RRT method. Hence, the planning time is lower and the IK-RRT performs better.



**Fig. 13.12** *Left* the bimanual IK-RRT is used to find a re-grasping motion. The wok is grasped with the *left hand* and the collision-free solution trajectory results in the re-grasping configuration. The solution is marked *blue*, the optimized solution is shown in *green*. *Right* the re-grasping motion is planned with the dual-arm J<sup>+</sup>-RRT, presented in Vahrenkamp et al. [44, 45]. The *red* parts are generated by the extend to goal part of the algorithm

### 13.4.5 Results

Several test scenarios have been evaluated in order to show the usability and the good performance of the IK-RRT approach. Therefore, single arm and bimanual grasping tasks were investigated and the results are presented in the following sections.

#### 13.4.5.1 Single Arm Reaching

In Table 13.4 the performance of the IK-RRT planner is evaluated in test scenarios without an obstacle and in a scene with a fixed obstacle (see Fig. 1.11(left)). The average values of 100 test runs are shown.

#### 13.4.5.2 Bimanual Re-Grasping

The result of the bimanual re-grasping planners is shown in Table 13.5. The test setup is similar to the single arm tests (a setup without and with a fixed obstacle is depicted in Fig. 13.12).

#### 13.4.5.3 Bimanual Motion Planning in a Kitchen Scenario

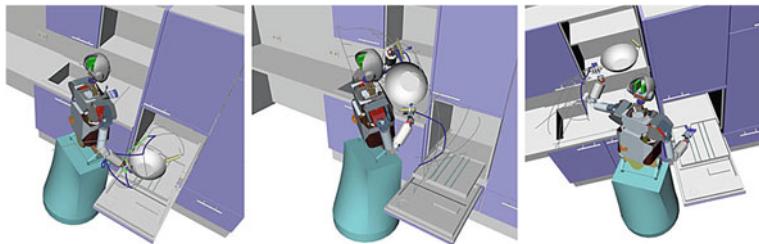
To evaluate the performance and capabilities of the developed algorithms in real world scenarios, a manipulation task in a kitchen environment is studied (Table 13.6). A wok should be grasped with the right hand of the robot, a re-grasping motion has to be planned and finally the object has to be placed in a cabinet. The planning framework should be able to generate collision-free joint trajectories in reasonable time. For this example, the task of solving the IK problem and the collision-free motion planning are considered separately. This leads to a planner which loses the ability of being probabilistically complete, since the planning is limited to one set of IK solutions and if this IK solution is not reachable the planning will fail. The experiments showed that the situation where an IK solution is not reachable by a collision-free motion was never observed and thus this theoretical disadvantage does not affect the applicability of this manipulation planning approach in this experiment. Theoretically it is possible to build a planner which is probabilistically complete. This can be done for this kind of manipulation planning problem, by searching IK solutions in parallel and for every solution an instance of the planning algorithm is started. If time goes to infinity, all possible IK solutions will be discovered and if a valid solution exists the planner will find it.(Fig. 13.13).

**Table 13.4** Performance of the single arm approaches

	Without obstacle average runtime (ms)	With obstacle average runtime (ms)
IK-RRT	140	480

**Table 13.5** Performance of the bimanual re-grasping planner

	Without obstacle average runtime (ms)	With obstacle average runtime (ms)
Bimanual IK-RRT	278	469



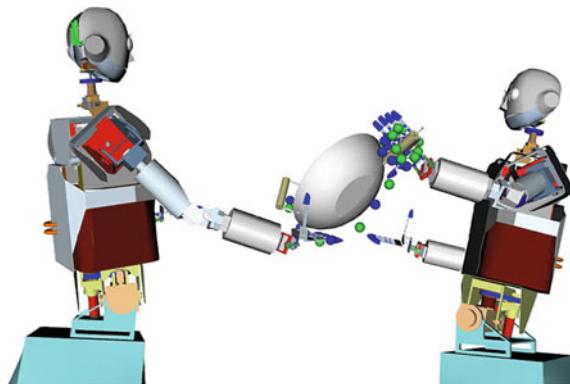
**Fig. 13.13** The results of the three planning tasks. In the *left* image the wok is grasped with the right hand, then the re-grasping procedure is executed and finally the object is placed in the cabinet

#### 13.4.5.4 Hand-Over Between Two Robots

The proposed algorithms can be used to generate collision-free re-grasping motions for two robots. Instead of considering two arms of one robot, two arms of two different robot systems can be used as input for the planning algorithms. A result of such a re-grasping motion can be seen in Fig. 13.14. The performance of the two arm hand-over planning algorithms is similar to the one robot case of Sect. 13.4.5. From the algorithmic point of view the only difference between the one robot and the two robot problems are the additional hip joints of the second robot (see [44, 45]).

#### 13.4.5.5 Experiment on ARMAR-III

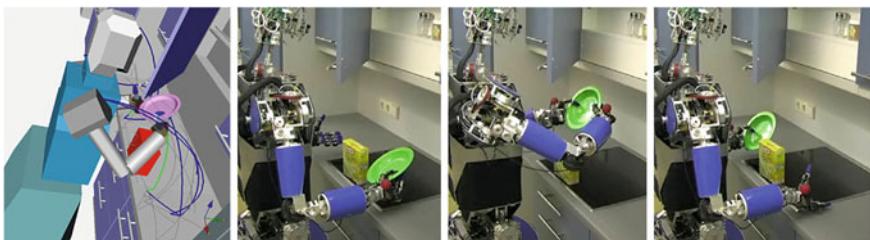
In this experiment ARMAR-III is operating in a kitchen environment where the partly opened cabinet and a box are limiting the operational workspace of the robot. A planner for bimanual re-grasping (see Sect. 13.4.4) is used to find a hand-over configuration and to plan a collision-free handoff motion for both arms. The



**Fig. 13.14** A hand-over configuration for two robots

**Table 13.6** Performance of the kitchen experiment

	IK solving (ms)	Motion planning (ms)
Grasp	19.6	345
Re-grasping	760.7	4,702
Place	22.6	1,263
Complete	802.9	6,310



**Fig. 13.15** The humanoid robot ARMAR-III is re-grasping a plate in the kitchen. The *left* image shows the planned trajectory and the other pictures were taken during execution on ARMAR-III

resulting trajectory moves both hands and the plate that is already grasped with the right hand, to the hand-over position and after re-grasping the arms are moved to a standard pose. This real world experiment shows how the bimanual re-grasping planners enable the humanoid robot ARMAR-III to hand-over objects from one hand to the other in the presence of obstacles (Fig. 13.15).

## 13.5 Conclusion

In this chapter, we presented a combined approach for grasp and motion planning.

The grasp planning algorithm is based on the medial axis of 3D objects. The only requirement of this algorithm to be met by a robotic hand is the capability to oppose the thumb to the other fingers which is fulfilled by all hand models we know. We performed experiments on a set of household objects that show that our method effectively reduces the computational overhead for costly collision detection and force-closure testing by generating and evaluating only candidate grasps that—from a geometric point of view—have a high probability to be stable. In contrast to related work that uses boxes or superquadrics to approximate an object’s geometry, the medial axis representation of an object does not sacrifice potentially promising candidate grasps to poor geometry approximation. Instead, we showed that the medial axis provides valuable structural and symmetry information that can be easily used to generate big quantities of force-closure grasps in an efficient manner. As we derived the heuristics for candidate grasp generation directly from geometric object properties our method produces many grasps that, in our opinion, seem quite intuitive to a human.

Further, we presented an IK-based strategy for planning reaching, grasping, and re-grasping motions including single and dual-arm tasks. The search for a suitable and collision-free configuration for grasping or re-grasping an object is included in the IK-RRT planning algorithm and thus the planner covers the search for suitable target configurations implicitly. Since the IK-RRT approach depends on an efficient IK solver which is able to sample the set of all potential solutions, we showed how such IK-solvers can be constructed for challenging tasks, such as re-grasping. The redundancy of the kinematic structure is covered by sampling-based techniques and by employing analytic IK-solvers for sub-chains of the robot it was possible to construct efficient IK algorithms. Additionally we showed how the analysis of the reachability can increase the efficiency of such sampling-based IK-solvers. Several advantages of the developed IK-approaches were presented: included collision-detection, implicit joint-limit avoidance, and no local minima problem that arise in most of Jacobian-based IK-solvers. The presented techniques do not rely on a specific robot and thus it can be used to develop sampling-based IK-solvers for arbitrary kinematic chains of a robot.

Based on these IK methods the developed IK-RRT planning concept combines the search for IK solutions by planning a collision-free motion. Additionally a feasible grasp from a set of potential grasping configurations is implicitly selected and hence, no grasp selection or IK solving has to be done in advance. The presented planners can be used to efficiently plan reaching and re-grasping tasks without defining explicit target configurations. This leads to planning algorithms which can be applied to humanoid robots and which do not require explicit goal configurations. Several planning setups were investigated and the performance of the proposed algorithms was evaluated in simulations and real world experiments.

In this work, we showed how efficient grasp and motion planning in the context of humanoid robotics can be achieved. Therefore, a novel grasp planner was presented which allows the efficient generation of suitable grasps. Further, reachability analysis is performed in an offline step. The precomputed information from both, the grasp planner and the reachability analysis, can be exploited to increase the efficiency of online motion planning. We showed how complex grasping motions can be planned considering single handed and bimanual tasks in cluttered environments.

The results of this work can be used to plan reaching and grasping motions. To execute such trajectories on a real robot system one has to deal with inaccuracies caused by noisy sensor data and inexact world modeling. Hence, closed-loop control incorporating vision and haptic feedback could be used to ensure the correct execution of planned motions. In [46] and [44] we showed how Visual-Servoing approaches can be used for reliable execution of single handed and bimanual grasping motions. Through the Visual-Servoing-based approach, the planned motions are slightly adapted in order to reflect the expected hand-object relation as well as possible during execution.

## References

1. Asfour T, Dillmann R (2003) Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem. In: IEEE/RSJ international conference on intelligent robots and systems (IROS)
2. Asfour T, Regenstein K, Azad P, Schröder J, Bierbaum A, Vahrenkamp N, Dillmann R, (2006) Armchair-III An integrated humanoid platform for sensory-motor control. In: IEEE-RAS international conference on humanoid robots (humanoids 2006), pp 169–175
3. Asfour T, Azad P, Vahrenkamp N, Regenstein K, Bierbaum A, Welke K, Schröder J, Dillmann R (2008) Toward humanoid manipulation in human-centred environments. *Robo Auton Syst* 56(1):54–65
4. Attali D, Boissonnat JD, Edelsbrunner H (2009) Stability and computation of medial axes: a state-of-the-art report. In: Möller T, Hamann B, Russel B (eds) Mathematical foundations of scientific visualization, computer graphics, and massive data exploration. Springer, Berlin, pp 109–125
5. Badler NI, Phillips CB, Webber BL (1993) Simulating humans: computer graphics animation and control. Oxford University Press, New York
6. Becher R, Steinhäus P, Zollner R, Dillmann R (2006) Design and implementation of an interactive object modelling system. In: International symposium on robotics
7. Berenson D, Diankov R, Nishiwaki K, Kagami S, Kuffner J (2007) Grasp planning in complex scenes. In: 7th IEEE-RAS international conference on humanoid robots, 2007, pp 42–48. doi:[10.1109/ICHR.2007.4813847](https://doi.org/10.1109/ICHR.2007.4813847)
8. Berenson D, Srinivasa S, Ferguson D, Collet A, Kuffner J (2009) Manipulation planning with workspace goal regions. In: IEEE international conference on robotics and automation (ICRA’2009), Kobe, Japan
9. Berenson D, Srinivasa SS (2008) Grasp synthesis in cluttered environments for dexterous hands. In: 8th IEEE-RAS international conference on humanoid robots, 2008, humanoids 2008, pp 189–196. doi:[10.1109/ICHR.2008.4755944](https://doi.org/10.1109/ICHR.2008.4755944)

10. Bertram D, Kuffner J, Dillmann R, Asfour T (2006) An integrated approach to inverse kinematics and path planning for redundant manipulators. In: ICRA, pp 1874–1879
11. Blum H (1967) Models for the perception of speech and visual form. A transformation for extracting new descriptors of shape, pp 362–380. MIT Press, Cambridge
12. Bruck J, Gao J, Jiang A (2007) Map: medial axis based geometric routing in sensor networks. *Wirel Netw* 13(6):835–853
13. Ciocarlie M, Goldfeder C, Allen P (2007) Dimensionality reduction for hand-independent dexterous robotic grasping. In: IEEE/RSJ international conference on intelligent robots and systems, 2007, IROS 2007, pp 3270–3275. doi:[10.1109/IROS.2007.4399227](https://doi.org/10.1109/IROS.2007.4399227)
14. Craig JJ (1989) Introduction to robotics. Addison-wesley, Reading
15. Dey T, Goswami S (2003) Tight cocone: a water-tight surface reconstructor. In: Proceedings of the eighth ACM symposium on solid modeling and applications, pp 127–134. ACM, New York
16. Diankov R, Ratliff N, Ferguson D, Srinivasa S, Kuffner J (2008) Bispace planning: concurrent multi-space exploration. In: Robotics: science and systems
17. Drumwright E, Ng-Thow-Hing V (2006) Toward interactive reaching in static environments for humanoid robots. In: International conference on intelligent robots and systems, 2006 IEEE/RSJ, pp 846–851
18. Ferrari C, Canny J (1992) Planning optimal grasps. In: Proceedings IEEE international conference on robotics and automation, vol 3, pp 2290–2295. doi:[10.1109/ROBOT.1992.219918](https://doi.org/10.1109/ROBOT.1992.219918)
19. Gienger M, Toussaint M, Jetchev N, Bendig A, Goerick C (2008) Optimization of fluent approach and grasp motions. In: 8th IEEE-RAS international conference on humanoid robots, humanoids 2008, pp 111–117. doi:[10.1109/ICHR.2008.4755940](https://doi.org/10.1109/ICHR.2008.4755940)
20. Goldfeder C, Ciocarlie M, Dang H, Allen PK (2009) The columbia grasp database. In: IEEE international conference on robotics and automation, 2009. ICRA'09, pp 1710–1716. doi:[10.1109/ROBOT.2009.5152709](https://doi.org/10.1109/ROBOT.2009.5152709)
21. Goldfeder C, Ciocarlie M, Peretzman J, Dang H, Allen PK (2009) Data-driven grasping with partial sensor data. In: International conference on intelligent robots and systems, IROS 2009. IEEE/RSJ, pp 1278–1283. doi:[10.1109/IROS.2009.5354078](https://doi.org/10.1109/IROS.2009.5354078)
22. Goldfeder C, Lackner C, Pelosof R, Allen PK (2007) Grasp planning via decomposition trees. In: IEEE international conference on robotics and automation, 2007, pp 4679–4684. doi:[10.1109/ROBOT.2007.364200](https://doi.org/10.1109/ROBOT.2007.364200)
23. Guilamo L, Kuffner J, Nishiwaki K, Kagami S (2005) Efficient prioritized inverse kinematic solutions for redundant manipulators. In: Proceedings of the IEEE international conference on intelligent robots and systems (IROS), pp 3921–3926
24. Held M (2001) Vroni: an engineering approach to the reliable and efficient computation of voronoi diagrams of points and line segments. *Comput Geom Theory Appl* 18:95–123
25. Hisada M, Belyaev AG, Kunii TL (2002) A skeleton-based approach for detection of perceptually salient features on polygonal surfaces. *Comput Graph Forum* 21:689–700
26. Huebner K, Ruthotto S, Krasic D (2008) Minimum volume bounding box decomposition for shape approximation in robot grasping. In: IEEE international conference on robotics and automation, ICRA 2008, pp 1628–1633. doi:[10.1109/ROBOT.2008.4543434](https://doi.org/10.1109/ROBOT.2008.4543434)
27. Kasper A, Becher R, Steinhaus P, Dillmann R (2007) Developing and analyzing intuitive modes for interactive object modeling. In: International conference on multimodal interfaces (2007)
28. Kee D, Karwowski W (2002) Analytically derived three-dimensional reach volumes based on multijoint movements. *Hum Factors J Hum Factors Ergonomics Soc* 44(15):530–544
29. Kuffner J, LaValle S (2000) RRT-Connect: an efficient approach to single-query path planning. In: IEEE international conference on robotics and automation (ICRA'2000), San Francisco, pp 995–1001
30. LaValle SM (2006) Planning algorithms. Cambridge University Press, Cambridge, <http://planning.cs.uiuc.edu/>
31. Li Y, Fu JL, Pollard NS (2007) Data-driven grasp synthesis using shape matching and task-based pruning. *IEEE Trans Vis Comput Graph* 13(4):732–747. doi:[10.1109/TVCG.2007.1033](https://doi.org/10.1109/TVCG.2007.1033)

32. Liu YH, Lam M, Ding D (2004) A complete and efficient algorithm for searching 3-d form-closure grasps in discrete domain. *IEEE Trans Rob* 20(5):805–816
33. Miller A, Knoop S, Christensen HI, Allen PK (2003) Automatic grasp planning using shape primitives. In: Proceedings IEEE international conference on robotics and automation, 2003, ICRA'03, vol 2, pp 1824–1829
34. Miller AT, Allen PK (2004) Graspit! a versatile simulator for robotic grasping. *IEEE Robot Autom Mag* 11(4):110–122. doi:[10.1109/MRA.2004.1371616](https://doi.org/10.1109/MRA.2004.1371616)
35. Morales A, Chinellato E, Fagg AH, del Pobil AP (2004) Using experience for assessing grasp reliability. *Int J Hum Robot* 1(4):671–691
36. Przybylski M, Asfour T, Dillmann R (2010) Unions of balls for shape approximation in robot grasping. In: Proceedings of the IEEE international conference on intelligent robots and systems (IROS)
37. Quigley M, Gerkey B, Conley K, Faust J, Foote F, Leibs J, Berger E, Wheeler R, Ng A (2009) Ros: an open-source robot operating system. In: ICRA. Kobe, Japan
38. Reif JH (1979) Complexity of the mover's problem and generalizations. In: SFCS'79: proceedings of the 20th annual symposium on foundations of computer science (SFCS 1979), pp 421–427. IEEE Computer Society, Washington
39. Sciavicco L, Siciliano B (2000) Modeling and control of robot manipulators, 2nd edn. Springer, London, pp 96–100
40. Shimoga KB (1996) Robot grasp synthesis algorithms: a survey. *Int J Hum Robot* 15(3):230–266
41. Thakur A, Banerjee AG, Gupta SK (2009) A survey of cad model simplification techniques for physics-based simulation applications. *Comput Aided Des* 41(2):65–80
42. Vahrenkamp N, Simox. URL <http://www.sourceforge.net/projects/simox>
43. Vahrenkamp N, Berenson D, Asfour T, Kuffner J, Dillmann R (2009) Humanoid motion planning for dual-arm manipulation and re-grasping tasks. In: Proceedings of the IEEE international conference on intelligent robots and systems (IROS), St. Louis, USA, pp 2464–2470
44. Vahrenkamp N, Böge C, Welke K, Asfour T, Walter J, Dillmann R (2009) Visual servoing for dual arm motions on a humanoid robot. In: Proceedings of IEEE-RAS international conference on humanoid robots (humanoids), pp 208–214
45. Vahrenkamp N, Do M, Asfour T, Dillmann R (2010) Integrated grasp and motion planning. In: ICRA. Anchorage, USA
46. Weghe MV, Ferguson D, Srinivasa S (2007) Randomized path planning for redundant manipulators without inverse kinematics. In: IEEE-RAS international conference on humanoid robots

## **Part IV**

# **Applications**

# **Chapter 14**

## **Hardware for Industrial Gripping**

### **at SCHUNK GmbH & Co. KG**

**Christopher Parlitz**

**Abstract** The process of handling component parts in production is often underrated as technically simple or even trivial. Also from a production point of view, it is clear that the handled workpiece does not increase in value during the handling process. Nevertheless, handling is the key of automation allowing machines to operate on workpieces automatically, i.e., making changes to the workpiece and therefore enhancing its value. Humans, having one of the most flexible gripping technologies, combined with superior sensors and data processing, tend to underestimate such handling tasks. The automated gripping of an object largely depends on the characteristics of the workpiece, e.g. its shape, weight, and stability. The focus of the following chapter will be on actual gripping. Solutions using suction or magnetism are not discussed. An overview of the necessary components of successful gripping will be given, followed by a brief history of gripping at SCHUNK. A range of different gripper types and future functionality will be presented and some unusual examples of applications involving gripping shown. The last part describes robotic hands with several degrees of freedom and several fingers.

#### **14.1 Terms and Fundamentals of Industrial Robot Handling**

The process of handling component parts in production is often underrated as technically simple or even trivial. Also from a production point of view, it is clear that the handled workpiece does not increase in value during the handling process.

---

C. Parlitz (✉)  
SCHUNK GmbH & Co. KG, Lauffen/Neckar, Germany  
e-mail: Christopher.parlitz@de.schunk.com

Nevertheless, handling is the key of automation allowing machines to operate on workpieces automatically, i.e., making changes to the workpiece and therefore enhancing its value. Humans, having one of the most flexible gripping technologies, combined with superior sensors and data processing, tend to underestimate such handling tasks. The automated gripping of an object largely depends on the characteristics of the workpiece, e.g. its shape, weight, and stability. The focus of the following chapter will be on actual gripping. Solutions using suction or magnetism are not discussed. An overview of the necessary components of successful gripping will be given, followed by a brief history of gripping at SCHUNK. A range of different gripper types and future functionality will be presented and some unusual examples of applications involving gripping shown. The last part describes robotic hands with several degrees of freedom and several fingers.

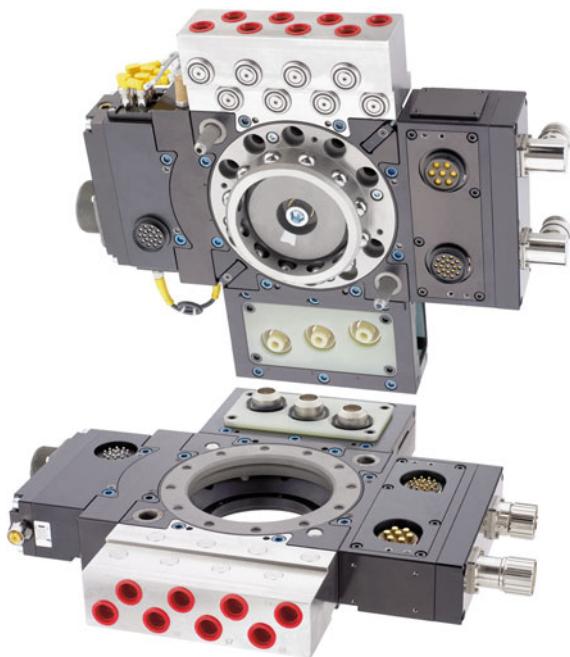
### ***14.1.1 Components of Industrial Handling***

Global competition and new distribution channels have shown that any one thing is usually available at all times. At the same time, in almost every product category, the diversity of variants is growing. The customized end product is becoming more standard, and at the same time the idle time which can be accepted by the customer between the placing of an order and delivery shortens. “Order today—delivery tomorrow.” Such are the expectations of the customers. This means flexible production is necessary to be able to react to these volatile markets. To reduce production cost the trend is toward electrical automation, replacing pneumatic, and hydraulic devices.

In order to fulfill these expectations, flexible handling inside and outside the processing line plays a crucial role. When thinking of industrial handling, the following components need to be considered [1]:

- *Handling or moving device.* This is the actuated kinematic chain which enables a workpiece to be moved. This might just be a simple rotation around one axis or linear displacement, but can also be a complex industrial robot with several degrees of freedom (DOF). According to a study [2] by the International Federation of Robotics (IFR), the total accumulated sales of industrial robots since their first appearance in the 1960s are more than 2,200,000 units, while the number of actual operational robots is lower. Due to withdrawal from service and scrapping, this figure is estimated somewhere in the range of 1,021,000–1,300,000. The first robot was the Unimate by Joe Engelberger and George Devol. It had its debut at General Motors in 1961. Up to this day, the automotive industry has been and will continue to be a driving force of robot development. While no precise figures are known, it can be assumed that the majority of these robots do some kind of handling or other tasks like inspection or testing.
- *Flange.* While mechanical interfaces are often standardized (e.g., “ISO 9409-1:2004 Manipulating industrial robots—Mechanical interfaces—Part 1: Plates”)

**Fig. 14.1** Pneumatic tool changer with locking system consisting of change head and change adapter



adapters are usually needed to combine the moving device with the actual gripper. In more complex handling scenarios the flange might be equipped with a *tool changer*.

- **Tool Changer.** Modern automation systems have to adapt quickly to the changing requirements of weight and shape. In order to complete these tasks, today's gripper quick-change systems are already very popular. Usually, a change system consists of a change head and change adapter (see Fig. 14.1). The head, mounted onto the robot, couples up the adapter mounted onto the tool or gripper. In industrial automation, usually a pneumatically driven locking piston ensures that the connection is secure. After coupling, pneumatic and electric feed-through need to automatically supply the robot tool. Additional robot tools are exchanged in a so-called storage rack. They ensure that the grippers with energy supply and sensor system can be exchanged quickly and automatically. A self-retaining locking system ensures that robot and effector remain coupled in case of power failure. Modern quick-change solutions go one step further, offering highly efficient valve adapter plates for quick-change systems, which comprise up to 14 microvalves and a complete valve terminal. They make handling processes more efficient. Air feed-through and the locking of quick-change systems can be switched from the valves. The adapter plates with integrated high-performance valves are assembled between the robot flange and the changing head. In conventional solutions, the compressed air in the air supply is lost. If valve adapter plates are used, just the piston area of the sealed actuators and lines within the

changing systems are filled with air. This drastically reduces the required amount of air and shortens cycle time.

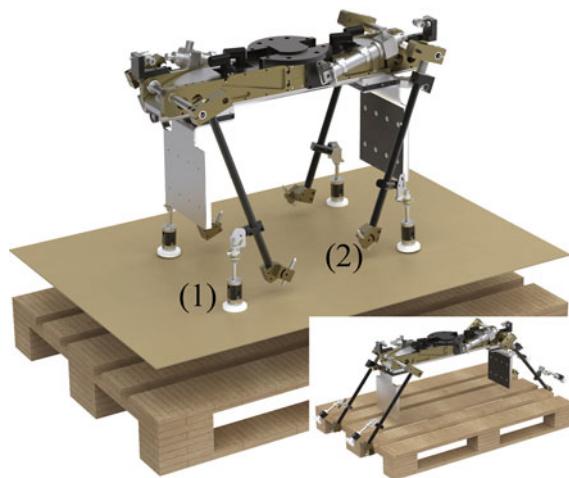
- *Gripper module.* The gripper module combines the actual gripper drive and gripper kinematics. On the one hand the kinematics transforms the movement of the actuator (e.g., pneumatic piston) to the actual movement of the gripper fingers. The drive, on the other hand transforms power (e.g. air pressure, electricity) into a rotary or translatory movement.
- *Gripper jaws.* The gripper fingers are the interface between the gripper and the workpiece. The fingers either induce the appropriate force or shape to fit the workpiece allowing the handling device to move the part. The fingers are the actual operating elements of the gripper. Good contact between workpiece and gripper fingers is essential for safe gripping with minimal force. Therefore, it is sensible to maximize the contact area between the workpiece and the fingers. Additionally, special coating can enhance friction and reduce the force required. There are two common possibilities to grasp an object: form fit and force fit [3]. Pure form-fit gripping means that the active forces and mass moments of inertia are vertically directed to the contact surfaces between workpiece and gripper fingers during the entire process. Form fit is the best choice as the gripping force can be kept low, and therefore have limited influence on the workpiece. Nevertheless, the direction of movement needs to be chosen carefully to make sure that the gripping type does not change during workpiece handling. Otherwise the defined gripping force may not be sufficient. Pure force-fit gripping means that the gripping force of the gripper fingers is induced by friction force only. Thus, gripping force is clearly dependent on the adhesive friction coefficient which may change under real working conditions, such as in dirty environments. Apart from these holding methods, there are also combinations of force and form fit. Other options are clamping by suction (pneumatic force fit), clamping of objects by magnetic or electrostatic force, as well as the use of an adhesive medium. Obviously, a combination of suction or magnetism as well as clamping might be integrated into one gripper system (Fig. 14.2).

Often a so-called gripper is therefore actually a complete gripper system consisting of several subsystems. These systems, next to the ones mentioned, could be sensors systems for quality assurance or process stability, control systems responsible for a complete array of grippers, tolerance compensation, and so on.

### 14.1.2 Safe Gripping

Losing one's grip on a glass usually does not do a great deal of harm. Losing a workpiece during handling may lead to major injury or financial loss. Liquid metals might be spilt or high payloads might turn into a kind of projectile. In industrial automation workpiece, loss mainly occurs during an emergency stop or power failure. In case of an emergency stop the moving device suddenly brakes

**Fig. 14.2** All-in One Pallet Gripper LEG with additional vacuum handling with vacuum suction cups (1) for handling of intermediate layers and gripper arm (2) for pallet handling tasks

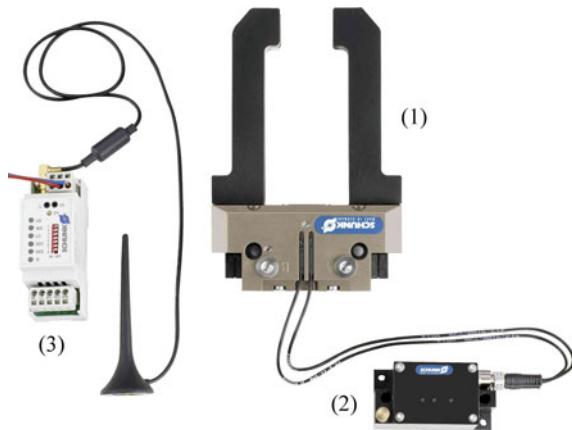


causing forces of inertia on the workpiece. These forces are not usually taken into account in the regular gripping force calculation. Therefore, the ideal gripping point of a workpiece is its center of gravity, minimizing torques during movement. If the energy supply for the gripper is reduced or completely cut off, the workpiece is most probably lost, if the respective provisions for gripping force maintenance have not been made beforehand. Several safeguarding devices against power loss are established. These are, e.g., spring elements which press the piston of a pneumatic gripper into the closing positions or safety valves maintaining the air pressure preventing air leakage from the gripper cylinder.

### 14.1.3 Gripper Sensors

In industrial automation, one can differentiate between tactile sensors requiring a force or torque action directly onto the measuring device, or nontactile sensing working at a distance between sensor and object monitored. The nontactile sensors include visual sensors like cameras. These sensors work with light as measuring medium. Other principles are resistance change, ultrasound, induction, capacity, magnetism, or other effects. Since the gripper is the only part of an automated system touching a workpiece sensors requiring tactile information need to be integrated here. Direct contact with the workpiece is established which easily permits tactile measuring, e.g., when accurate information on gripping forces is required. Besides differentiating between tactile and nontactile sensing, sensors can further be classified into switching sensors and measuring sensors. Switching sensors are often used for monitoring the respective gripper status, e.g., monitoring the state gripper open or closed (Fig. 14.3). Simple sensors like those based on magnetism or inductive proximity switches are sufficient for checking whether a

**Fig. 14.3** Wireless sensor system—consisting of (1) Gripper (2) Transmitter with two position sensors and (3) Receiver for installation in the control cabinet



workpiece has been picked up. Measuring sensors are normally used for measuring characteristics of a workpiece, for positioning the gripper or controlling the gripping force. In the past, the focus of attention was on the mechanics of the gripping systems, but now the sensor system has gained importance. Well-adjusted gripping forces and a safe feedback system for handling of compliant and sensitive parts are very important. For such applications, mechatronic grippers with their numerous options offer the optimum precondition. Even pneumatically actuated grippers combined with sensor modules can achieve amazing results. The PGN plus multi-tooth guided universal gripper shows what is already possible today. The program comprises more than inductive proximity switches, which are already standard for most pneumatic grippers. Electronic magnetic switches, for example, can be completely integrated into the gripper, and monitored from inside of the gripper at the piston position. The latest generation of magnetic switches is programmable and can even recognize two switching points instead of one. In order to increase process stability, the grippers have programmable switch-off points. For applications in machines, in confined applications or in adverse environments, where cables would endanger the process stability cable-free radio-controlled sensor systems are available. And even in extreme environmental conditions such as high temperatures, explosion hazard, aggressive fluids, or extreme quantities of hot chips or coolant, there is always a standardized module available: the fluid monitoring system works completely without any electric sensors or additional cables. Via a pneumatic measuring line and an AND-valve it transfers the information “open”, “gripped” or “closed” to the control unit.

Even for inspection tasks, universal grippers can be equipped: flexible positioning sensors can detect up to five pre-defined position areas of the jaws—ideal for sorting various components or to carry out an automatic quality control. More efficient are the high resolution analog sensors, which transform the universal gripper into a precision measuring station. With an extremely high resolution of up to 0.003 mm, the gripper measures the ongoing process of every individual

gripped component. Via the PLC, any number of switching points can be defined and therefore any number of components or tolerance ranges can be differentiated.

## 14.2 History of Automated Handling at SCHUNK

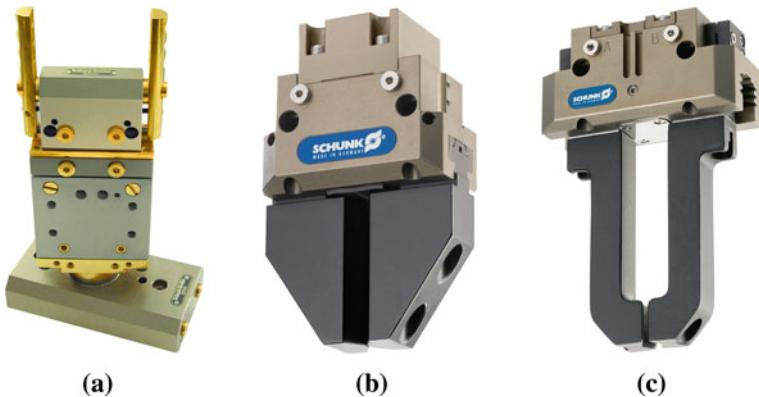
The history of grippers and of industrial robots is closely linked. But the actual history of artificial hands is much older. Replicas of hands were first used in the domain of prosthesis. A famous example is the metal prosthesis of Goetz von Berlichingen (1480–1562), also called the “Knight with the Iron Hand”. It had five fingers which could be bent passively, fixed and released by the press of a button. The hand weighed more than 1.5 kg but was a remarkable achievement for the time. Today’s prostheses are of course much more advanced, electronically driven and controlled by bio-electrical impulses when muscles contract in the wearer’s amputated stump. For a prosthetic hand the focus is much more on aesthetics, weight and broad range of functionality, while the supply of power is limited, so self-locking mechanisms are preferred. Today’s prosthetic hands have integrated drive technologies and weigh less than 500 g.

No artificial hand has so far come close to the human hand as the “tool of tools” (Aristotle: 3. Book “On the Soul” 350 B.C.) which was even considered the “extension of the human brain” by Kant. At the beginning of human evolution, however, the human hand was not equipped with the fine motoric skills which allow us to work with keyboards, writing equipment, or other tools. It took millions of years to train such refined movements. Gripping tools are undergoing a similar evolutionary process. One can only hope that it is not going to take so long.

The first gripper modules were developed at SCHUNK nearly 30 years ago [4]. This standard gripper PPG (Fig. 14.4a) offered by SCHUNK in 1983 had a gripping force of 650 N at a 6 mm stroke and 1450 N at a 3 mm stroke (6 bar pressure). The next milestone was set by the PGN gripper (Fig. 14.4b), which was built on the same technical principle as the PPG, but had more than three times higher force to weight ratio. Later generations of this gripper series like the PGN plus (Fig. 14.4c) further increased this ratio while also improving service life. Additionally, a multi-tooth guidance system spreads the load. The guiding is no longer done by the classical T-slot but by a multiple prismatic guidance arranged in parallel.

The electronic gripper era at SCHUNK started in 2003, when “amtec robotics” was incorporated into the SCHUNK Group and innovative mechatronic components for robotic actuation completed the product range for automation components.

Today’s grippers have gained considerably in functionality, quality, and sensor integration.



**Fig. 14.4** History of SCHUNK Grippers—PPG (a), PGN (b), PGN plus (c)

## 14.3 Current Gripper Design

### 14.3.1 *Types of Grippers*

Industrial robots work quickly, reliably and never stop. The achieved output and quality significantly depend on the individual periphery, particularly of the gripping systems. In the hunt for the optimum gripper, engineers and users have to consider a large variety of influencing factors. Table 14.1 gives an overview of typical relevant factors that need to be considered when choosing a gripper.

In addition to efficiency and process reliability, sustainability of the gripping systems plays an increasing role.

Besides the proven pneumatically driven gripping modules, more and more electrically driven grippers have been recently established. Both types of drive have great features. A pneumatic gripper has high compact performance, which means in a restricted installation space, relatively high gripping forces are achieved. They are robust, attractively priced, and can be easily commissioned and maintained.

Despite these many advantages, a kind of paradigm shift has recently been noticed. Mechatronic grippers are all-rounders, well-equipped, and energy efficient. And despite a higher initial investment, they are also attractive from an economic point of view. In addition to gripping, they allow the control on the gripping position, stroke, closing speed, acceleration, or force. In the future, more and more mechatronic modules will be controlled and remotely maintained from the Internet or a data line. Moreover, the very functional and visually effective design promotes trust in function, and safety of man and machine. The fast progress of electric servo drives assures that the number of mechatronic components will become increasingly attractive in price. Furthermore, the number of users with mechatronic know-how is increasing, and thus the acceptance of mechatronic solutions.

**Table 14.1** Relevant factors for gripper selection

Relevant Factors for Gripper Selection	
Usage:	Recurring or nonrecurring (i. e., more than one installation) Technical improvement over existing technology New design Cost reduction
Ambient conditions	Temperature Humidity Degree of pollution
Workpiece	Type of workpiece Weight Material at contact point (Steel, Aluminium, Rubber, and so on.) Clamping marks allowed? Static friction coefficient Gripper diameter Distance of center of mass to gripper top edge
Gripper	Parallel, angular, or centric gripper Electric or pneumatic Electric 24VDC, 48VDC, or 230 VAC Field bus type Ingress protection IP Orientation (Hanging, lying, and standing)
Parameters	Field Bus Baud rate Module-Address (ID) Referencing
Jaws	Form fit (prism jaw) or Force fit Prism angle, prism width Minimal stroke Mass of jaw Material of jaw at contact point Lateral set-off of workpiece center of gravity
Forces	Acceleration of gripper in x, y, and z Additional process force on gripper

One example of an electric gripper with sensors is the gripper PG070 (Fig. 14.5). The gripper includes a electronically commuted permanent magnet synchronous (servo) motor and a spindle drive with magnetic brake. An incremental encoder determines the position and angle velocity. A linear guidance in combination with the ball revolution spindle permits an especially accurate and sensitive gripping force control.

Legal requirements can play a role in selecting the drive system as an example in South Korea shows. Here, mechatronic solutions are mandatory for reasons of conservation. In Europe, mechatronic components offer advantages when looking at the new machine directives for machine and system manufacturers in the framework of safety management. In the short and long term, it can be assumed that pneumatic and mechatronic drives will establish themselves in the field of industrial automation and will be chosen according to their advantages. Whether

**Fig. 14.5** Servo-electric  
2-finger parallel gripper PG  
70



pneumatic or electric gripper, for opening and closing the fingers and the arrangement of fingers the following options have been established in industry [5]:

- *Two finger parallel gripper* (Fig. 14.6a). Two base jaws move in a lateral, synchronous gripping movement. For a pneumatic gripper this movement is usually induced by a piston. An electrical gripper uses a gear mechanism that transmits torque into a linear displacement.
- *Three finger centric gripper* (Fig. 14.6b). Three base jaws are orientated in a 120° angle on a circular gripper body and are ideal for grasping cylindrical objects. The gripper can be used, so that the workpiece is self-centering when grasping.
- *Two finger angular gripper* (Fig. 14.6c). The two-finger angular gripper is driven by a piston with spring return. The base jaws are opened at an angle in a synchronized fashion by the bearing-mounted lever mechanism. The reset is done by compression spring.
- *Three finger angular gripper* (Fig. 14.6d). This gripper has a similar mechanism to the two-finger angular gripper. Each base jaw has a piston opening the base jaws and resetting them with a spring.
- *Two finger radial gripper* (Fig. 14.6e). These grippers are also called 180° angular grippers and have the advantage of avoiding additional gripper motions. Since each jaw rotates away by 90°, the fingers are completely removed from the work area. A motion to retract the entire gripper can be omitted.



**Fig. 14.6** **a** Two-finger parallel gripper (e.g. MEG). **b** Three finger centric gripper (e.g. PZN plus). **c** Two finger angular gripper (e.g. LGW). **d** Three finger angular gripper (e.g. SGW). **e** Two finger radial gripper (e.g. PRG). **f** Four finger concentric gripper (e.g. PZV). **g** Long stroke gripper (e.g. PZH)

- *Four finger concentric gripper* (Fig. 14.6f). Four-finger grippers are ideal for precise picking, orientation, and subsequent joining of square materials. Often the base jaws are driven by a wedge hook activated by a piston.

So-called *special long stroke grippers* (Fig. 14.6g) offer higher variability. Instead of changing the jaws or complete gripper again and again, they can handle components of various sizes alternately. Thus, the PZH plus long stroke gripper with high compact performance from SCHUNK, for example, achieves such a long stroke due to the belt drive, making a costly exchange of top jaws, or gripper no longer necessary. Its patented multi-tooth guidance and housing made of high-tensile aluminum make it performance intensive. Compared with similar grippers, it applies a gripping force which is about twice as high. In order to achieve a certain stroke and force, smaller sizes are sufficient. This saves room and lowers energy consumption, and investment costs. The large center bore allows the installation of a camera, feed through of material, or gripping of overlong workpieces.

The potential of modern lightweight solutions is tremendous: existing robots can be upgraded for handling heavier parts. For new plants, smaller dimension robots can be used, cycle times are shortened, lighter multi-gripper systems can be developed, and the energy efficiency of robots and handling systems is increased.

The following two examples show what is possible today.

#### *Example 1: Modular heavy load gripper.*

At a weight of just 350 kg, the SLG servo-driven heavy load gripper (Fig. 14.7) from SCHUNK disposes of a 100 mm stroke per finger and a gripping force of 23,000 Newtons. It reliably handles components of up to 700 kg safely. The gripper's rigidity and its jaws have been optimized, and in case of high dynamic payloads, no inadmissible forces occur, and excess material is removed. The gripping systems can be driven by various electro-motors. The drive is actuated via a centrally located trapezoidal thread, which evenly and precisely transmits the force onto the roller-guided base jaws. For operation, the gripper is integrated into the robot's control unit as a seventh axis.

#### *Example 2: Variable Long Stroke Grippers*

The support structure of the LEG-C (Fig. 14.8) long-stroke gripper is made of carbon fiber composite material. At a weight of 10 kg, it disposes of a gripping force of up to 4,000 N, and a variable stroke of up to 600 mm. If a shorter travel path for the fingers is needed, the stroke can be scaled as required. Cycle times are minimized and the consumption of compressed air is reduced. For a continuous lightweight design concept, the aluminum fingers are also designed in a light-weight version. They move on a recirculating ball carriage. Via amount and distance of the guiding carriages, the gripper can be perfectly adjusted to the individual payload. For particularly high gripping forces or extremely long fingers, the distance between the carriages is increased. Even fingers with a length of more than 1,000 mm can be used. The almost scope-free belt drive and the synchronized

**Fig. 14.7** Heavy load gripper—the SLG handles parts up to 700 kg



**Fig. 14.8** A gripper made of carbon fiber with very long stroke (LEG-C)



fingers assure that the gripper works precisely at a variable stroke, and if required large and small components can be alternately handled.

### 14.3.2 Future Gripper Functionality

While dexterous manipulation from multi-fingered hands is covered below, there is a range of functionality which can be added on the software side with, e.g., the existing kinematics of a two fingered parallel gripper. Sophisticated gripping processes in medical technology, biotechnology, food processing, measurement and testing technology, and in man-to-machine cooperation demand a great level of flexibility and sensitivity. To achieve this, new technologies need to be developed. With the WSG 50 SCHUNK (Fig. 14.9) has integrated electronics into the base jaws, e.g., for

**Fig. 14.9** Servo-electric 2-finger parallel gripper with integrated web server, highly precise gripping force control and long stroke (WG 50)



measuring forces via sensors, which are integrated into the fingers. Also a removable MicroSD card is used for saving programs and data. Storing parameters and documents and retrieving gripper settings after changing modules are possible without restrictions. The integrated web server allows a configuration via Ethernet TCP/IP as well as remote diagnosis/maintenance. Updates are possible via the Internet. Additionally, parameterization and diagnosis are possible with all conventional web browsers, making proprietary software on service laptops unnecessary. Another highlight is the integrated scripting language for programming the gripper's behavior. Here, a customized gripper program can be developed influencing the gripper's behavior based on sensor input or external I/O.

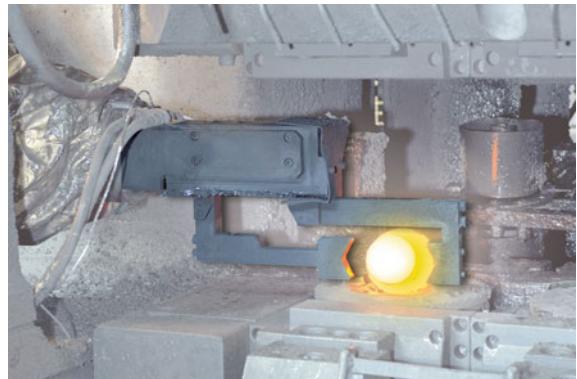
## 14.4 Applications

The examples given below are taken from SCHUNK industry solutions and further details can be found in [6].

### 14.4.1 Foundry Automation

In foundries robots are used for loading and unloading die casting machines and the connected dies. PZN plus 100 series centric grippers are used for gripping round parts (Fig. 14.10). Numerous parallel grippers in the PGN plus 125 series are sometimes combined into double grippers, so that larger parts can be gripped at several points at the same time. A PGN plus 300 that can deal with workpiece weights of up to 30 kg is also used for handling entire transmission casings. All of the grippers are equipped with optional dust protection equipment. They are also equipped with Viton seals that can survive temperatures of up to 130° without damage and can therefore be used in extremely hot environments. Sealed air is also

**Fig. 14.10** Taking up red-hot cast-iron parts and depositing them in cooling container



applied to some of the gripper modules to prevent the penetration of dust and slag, increasing the service life of the grippers.

#### 14.4.2 Handling of Salami Sausages

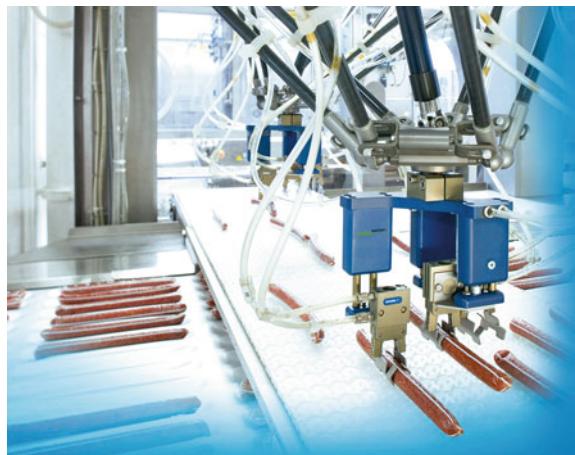
Salami sausages are conveyed on a feeding belt to the packaging station. The sausages are to be removed from the feeding belt and placed on the prepared packaging foil (Fig. 14.11). A robot control unit determines the gripping position of the sausages using a camera system. Grippers with stroke units are moved down, one after the other, and grip the sausages. Afterward, three sausages are deposited simultaneously and parallel on the packaging foil. With this setup the following was achieved:

- Due to the angular gripping principle, the sausage positions are centered.
- Sausages do not have to be pre-sorted or put into magazines.
- Highly reliable performance, even at 10 g.
- Suitable design for food, thanks to H1-compatible greases and food-compatible materials.
- Due to the lightweight design and very fast closing times (20 ms), a throughput of 600 mini-sausages per minute is possible.
- Also conceivable for other packaging applications, e.g., inserting 1.5 V batteries in blister packages.

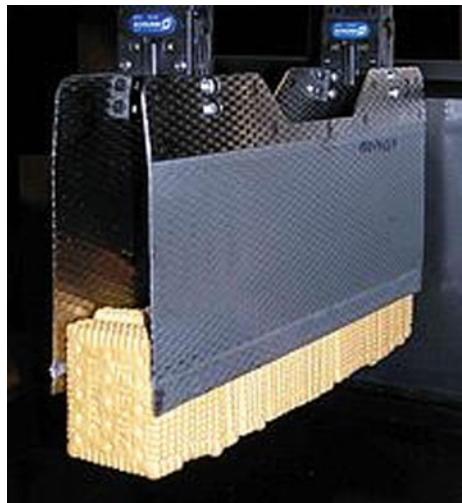
#### 14.4.3 Handling Baked Goods (cookies)

For the handling of butter cookies, a batch of 51 cookies is to be accepted and deposited in a package (Fig. 14.12). This was solved by equipping two small part angular grippers with special stainless steel gripper plates. The gripper plates have exchangeable food-compatible silicone inserts on the cookie side. This has the following benefits:

**Fig. 14.11** Handling of sausages on a conveyor belt for packaging



**Fig. 14.12** Handling baked goods (cookies)



- Damage-free cookie gripping
- Handling of a complete batch of cookies in one operation
- Hygienic and more reliable than suction technology
- Parts in contact with the product are food compatible (top jaws made of stainless steel and with silicone inserts).

#### **14.4.4 Handling of Crankshafts**

A rotary gripping module is to be used for simultaneous loading and unloading of a grinding machine with crankshafts in one step (Fig. 14.13). The conceived

**Fig. 14.13** A rotary gripping module with two grippers for simultaneous loading and unloading of a grinding machine with crankshafts in one step



functional module is attached to a movable portal. In order to ensure the necessary work safety, collision and overload protection is integrated. If there is a collision, an emergency stop signal is put out. Two swivel units are combined with each other. The first one swivels in the x, y plane by 180°, the other turns the workpiece by 30° in the x-z plane. Two dust-tight 2-finger parallel grippers are fastened to the swivel unit via an adapter plate. Both the swivel units as well as the grippers have Viton seals and are operated with an air purge (0.5 bar) to protect them from penetrating coolant. In addition, the grippers each have two torque sensors (supports) and have a workpiece presence check using compressed air. The solution is a maintenance-friendly complete functional module for use in contaminated surroundings.

## 14.5 Dexterous Manipulation and Robotic Hands

Industrial demands on gripping systems are economically met by grippers with a very limited (i.e., usually one) DOF leading to simple geometric grasps, and thus to a restricted range of objects grasped. This means that in most cases for each object grasped different fingers and possibly even a different gripper is required. A major advantage of this simple technology is that a robust exterior, long life under hard use, water- or dust-proof requirements can be fulfilled easily in industry. Recently, more complex automation processes handling an increased range of object geometries lead to an increasing number of components and gripping actions. As a consequence, the handling system becomes more expensive and the speed of the gripping process decreases. While suction grippers might be a solution for some applications handling a wide range of different geometries, they do not fulfill the requirements of precise handling for assembly tasks.

To grip a wide range of objects with one gripper, multi-finger hands are required.

Up to now restraints on a commercial launch of multifunctional gripping systems partly could be found in the comparatively low payload, the lack of reliability, the inadequate force, or the high price. The common aim is still to overcome the restrictions in gripping variety in order to cover as many gripping problems as possible in a wide field of applications at an economically justifiable price.

Currently, there are two main basic design methods (integrated and modular) for dexterous hands. The modular approach requires all actuators for finger movements to be integrated into the hand and fingers themselves. On the one hand this means they can be combined with any manipulators or handling devices. On the other hand, they are usually noticeably larger than human hands. Integrated designs put actuators into the manipulator carrying the hand, i.e., are taken out of the hand and moved into the “forearm”. Power is either transmitted via cables, rodding, or pneumatically. These hands are often comparable to the form and size of human hands. The usage of these hands requires the appropriate arm kinematics, though.

Both types of designs, modular and integrated, are sometimes underactuated. The flexibility of the hand is achieved by self-adapting finger kinematics which is either passive or is controlled via cables or roddings from the actuators. The DOF of the fingers is larger than the number of actuators used. The fingers adapt flexibly with coupled elements to the object’s actual shape. Underactuated hands have the disadvantage that controlling the hand especially in in-hand manipulation cannot be done at a low level.

Most robotic hands so far are either pure research projects or are only made commercially available to the research market.

SCHUNK has developed a multifunctional robot gripper in bilateral co-operation with the Institute for Process Control and Robotics (IPR), University of Karlsruhe [7]. The SCHUNK SDH 2 (Fig. 14.14) is the only 3-fingered fully actuated hand



**Fig. 14.14** Electrically-operated 3-finger gripper hand SDH

designed for industrial applications and has been commercially available for a significant amount of time. The design is a combination of performance and power density integrated into a professional design. A modular system with electrically driven joints and defined interfaces as motion modules has been worked out permitting extensive adaptability. The motion is driven by brush-type motors and gear boxes developed together with Harmonic Drive. Part of the waterproof concept is that all electrical connection cables are integrated. Due to an absolute encoder in each joint, the angle position is known directly after mounting and permanently. All modules are equipped with electrical connectors. The joint torques are 4.8 Nm for the proximal and 2.1 Nm for the distal joint of a finger and can be compared to an average woman's grip. The joint elements are combined by lightweight profiles variable in length. Tactile sensor arrays are positioned on every interacting surface of the gripper.

Main components of the joints are Harmonic Drive units. These are used for several reasons: A simple three-element construction combined with the unique harmonic drive principle puts extremely high reduction ratio capabilities into a very compact and lightweight package. Backlash is held at a minimum. Harmonic Drive gears are available with excellent positioning accuracy of less than 1 min of arc and repeatability within a few seconds of arc. They are reversible and can be used for speed reduction as well as for speed increase. Harmonic Drive gears exhibit very high torsional stiffness over the whole torque range, as well as almost linear hysteresis behavior.

The SDH has three identical fingers, each with two independently actuated rotary joints. A seventh independent actuator rotates two of the fingers in a coupled, symmetric fashion about the base (palm) to achieve the different grasping configurations (Fig. 14.15).

In total, the SDH has seven independent actuators controlling seven DOF through eight moving joints. The SDH motions are shown by the red arcs on the joints shown in Fig. 14.16.

The top right photo shows a tactile array on a proximal phalanx.

The main grasping surface of each link is equipped with an integrated tactile sensor array arranged in a  $6 \times 14$  rectangular grid with a contact force resolution



**Fig. 14.15** Grasping Configurations of SDH—Centric configuration (a), cylinder configuration (b), precise configuration (c)



**Fig. 14.16** Details of SDH components

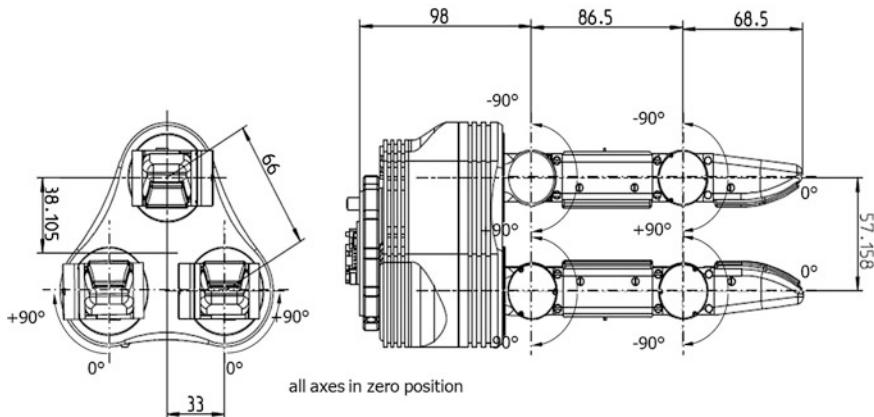
of 0.5 N. They are covered by a silicone layer to prevent damage and increase contact friction.

Two of the modular joints with stiff, high torque, harmonic drives (black cylinders) are shown beside a 10 cent coin. The connection plate for quickly mounting and dismounting the hand from an arm is shown bottom right. The lower left portion of the figure shows the drive mechanism that rotates the bases of the two fingers. This pivoting motion is coupled, so that the two fingers move together in symmetric fashion as illustrated by arcs 1a and 1b. The upper left portion of the figure shows the two links of each finger which are highly integrated with the tactile sensors and modular joint drives.

The basic attributes of the existing SDH and are shown in Table 14.2.

**Table 14.2** Basic attributes of SDH

Attribute	Value
Size	40% larger than human hand, weight 1.95 kg
Power	24VDC, Max Current 5A, Approx 120 W
Load	Finger Tip = 13 N (2 × human hand), Distal/Proximal torque (2.1/1.4 Nm)
Electrical interface	PC based, CAN Bus, TCP/IP or RS-232
Control interface	API interface, C++ or python

**Fig. 14.17** CAD Drawing of SDH

The SDH is shown in Fig. 14.17 in plan and overhead views to highlight the kinematic and geometric parameters most important to grasping. Note that in these views the fingers are rotated to curl in parallel planes. The three vertical dimensions (in mm) in the elevation view and the two dimensions (in mm) in the plan view can be adjusted freely at minimal cost. It is also easy to change the minimum and maximum joint angles and the shapes of the finger profiles.

The manufacturing cost of the current SDH product is raised by high-precision servo components and the six tactile sensor arrays. This will decrease with demand. While most buyers of the SDH are in R&D and are working on software developed for advanced gripping (e.g. [8]), the design was for industrial applications. Time will show which applications of today's research will be seen in the automation of tomorrow.

## References

1. Wolf A, Steinmann R (2004) Greifer in Bewegung. Springer, Berlin
2. IFR Statistical Department (2010) World robotics 2010 industrial robots. IFR Statistical Department, Frankfurt

3. Hesse S, Monkman GJ, Steinmann R, Schunk H (2004) Robotergreifer. Carl Hanser Verlag, Munich
4. Wolf A, Steinmann R, Schunk H (2005) Grippers in motion. Springer, Berlin
5. SCHUNK Online Catalogue (2011) [http://www.schunk.com/schunk/schunk\\_websites/products/products\\_level\\_1\\_overview\\_typ4.html?product\\_level\\_1=244&product\\_level\\_2=0&product\\_level\\_3=0&country=INT&lngCode=EN&lngCode2=EN](http://www.schunk.com/schunk/schunk_websites/products/products_level_1_overview_typ4.html?product_level_1=244&product_level_2=0&product_level_3=0&country=INT&lngCode=EN&lngCode2=EN). Accessed Dec 2011
6. SCHUNK Industry Solutions (2011) Field proven applications. [http://www.schunk.com/schunk/schunk\\_websites/webcms/EN/1290079/381589/381592?country=INT&lngCode=EN&lngCode2=EN](http://www.schunk.com/schunk/schunk_websites/webcms/EN/1290079/381589/381592?country=INT&lngCode=EN&lngCode2=EN). Accessed Dec 2011
7. Haag M (2006) Multifunctional gripping systems also fulfill rough industrial demands. In: Proceeding of the joint conference on robotics (ISR 2006—ROBOTIK 2006), Munich
8. Haase T (2011) Greifplanung und Greifskills für reaktives Greifen, University of Karlsruhe

# **Chapter 15**

## **Grasping in Agriculture: State-of-the-Art and Main Characteristics**

**F. Rodríguez, J. C. Moreno, J. A. Sánchez and M. Berenguel**

**Abstract** The automation of agricultural tasks has been widely developed in recent years due to labor requirements and its high cost for growers. The majority of agricultural tasks is of the “pick and place” type, which has led to robotization’s coming onto the scene as a solution for growers. In turn, they have witnessed how robots have begun to work on their farms or in their greenhouses. A foundation on the agricultural tasks is shown, as well as major developments to produce them. For this, the tasks, where they are used, are explained and the main organoleptic characteristics are set out as a basis to consider for future development. Within such developments, this work is focused on the grasping end-effector used in agriculture. A summary of the most relevant grippers in agricultural robotics is shown. In order to classify them, the Cutkosky taxonomy is used. Some representative grippers are taken as examples and explained in more detail.

### **15.1 Introduction**

Agriculture has probably been one of the least technically developed fields due to varying types of problems. On one hand, there are socioeconomic constraints, such as the fragmentation of farms, the traditional capital investment, low added-value of products, the seasonality of crops, the usually cheaper labor in relative terms, and a limited knowledge. Other factors of a technical nature must be added to

---

F. Rodríguez (✉) · J. C. Moreno · J. A. Sánchez · M. Berenguel  
Automatic Control, Electronics and Robotics Research Group, University of Almería,  
Agrifood Campus of International Excellence (ceiA3), Ctra. de la Playa,  
s/n—E-04120, La Cañada de San Urbano (Almería), Spain  
e-mail: frrodrig@ual.es

these such as lack of uniformity, in both the products and crop environments, with a very mountainous terrain, and adverse weather variables.

Nowadays, agriculture faces major demands for quality, product variety, and strong competition. Such demands force farmers to improve the production process, which usually leads to a rise in the harvest's final cost. One of the most important costs farmers have is labor; in many cases, it is expensive because of the specialization. All of these expenses increase the risk assumed by farmers and lead to a reduction in profits.

Automation, as a first step, and currently the improvement in the structures and growth systems, genetic engineering, and the utilization of computers, sensors, and actuators, allow better results in production to be achieved. As a second step, robotics has become an important tool for the development in coming years. In agriculture, many tasks are tedious (i.e. pruning), repetitive (i.e. harvesting), or even dangerous for human health (i.e. spraying). Some of the tasks are of the “pick and place” type, easily transferable to a robotic system. Robots offer the promise of reduced costs, increased safety, greater yields, reduced use of chemicals, and increased operational flexibility, including night-time operations.

In this sense, there are several international R&D programs to introduce technologies like robotics into agricultural fields where automation companies, research centers, and universities participate. The most positive aspect of these technological programs is that they have produced a lot of intermediate developments that have already been incorporated into different agricultural tasks, so their usefulness has been demonstrated. This fact is supported by studies performed by different agencies and institutions like the report “Technology Strategy Report for Creating a Robot Society in the 21st Century” by the Japan Robot Association (JARA) in 2001 [20]. It indicates that agriculture will be an important field for the application of robotics, especially in Europe, based on four factors: ability to develop new unique products (which are then copied by other countries), ability to export the developments made, ability to create a home market, and ability to create new markets.

On the other hand, the development of different features like autonomous navigation, robustness, and sensor processing will lead to several new outdoor applications which will open or change new markets in agriculture. The change of robotics into intelligent flexible systems with vast sensorial and auctorial capacities will lead to high-profit markets in the classical sectors of flexible production, manufacturing, logistics, etc., but also new profitable sectors like agriculture will be able to establish [10]. Papers on fruits and vegetables harvesting robots have been reported since more than 20 years ago, but they have not been practically used so far, because a precise, robust, and high-speed operation and a low cost is required for the robot [27, 30].

Nowadays, there is no mature technology in robot harvesting for fruit and vegetable. The problems mainly are low recognition ratios in real conditions, where crops vary enormously in terms of color, size, and shape, and high damage rate of fruits and vegetables during handling. Furthermore, the robot has to operate under adverse climatic conditions, such as high relative humidity, high

temperature and changing light conditions [60, 63, 68]. Recently, the US-Robotics Roadmap indicates that the agricultural robot has now matured to the point where it is ready to be transitioned for further commercialization and field deployment within the next few years [5]. All this has led to a significant increase in the number of robots developed around crops, as the World Robotics 2010 Executive Summary of the IFR Statistical Department indicates [18].

The agricultural tasks where robots are present are diverse, but could be clustered into soil preparation, seeding, transplanting, grafting, precision fertirrigation, pruning, deleaving, plague recognition, harvesting, pulverization, crop removal, and post-harvesting tasks. This chapter will focus on robots that work during the crop growth, because post-harvesting robotization is similar to an industrial operation. Robot application in agricultural production tasks has been researched and developed since the 1980s by many researchers who have found, in agricultural robotics, an important source of encouragement [21, 26, 29]. The nature of these analyzed tasks requires the use of mobile robots combined with manipulator robots with different end-effectors (spraying nozzles, irrigation droppers, vacuums, harvesting cutters, suction pads, etc.). Each one needs a particular design depending on the application, because it is necessary to manipulate fragile material, like plants and fruit.

The crop's variety, soils and soilless culture, crop closeness due to increasingly intensive agriculture like outdoor agriculture as protected agriculture, different protection structures and materials, climate, position acquired for fruits and leafs, etc. All of these conditions make the design process difficult as well as the work of robots and, obviously, their own end-effectors. Many authors have researched how to use robots in agriculture and the design of end-effectors has been widely worked on and tried out during crop growth.

End-effectors may consist of a gripper or a tool. Grippers are subsystems of handling mechanisms which provide temporary contact with the object to be grasped [43]. Although the term "gripper" is also used in cases of no actual grasping (suction cup, magnets, etc.), in this chapter it is only considered impactive mechanical grippers whereby prehension is achieved by impactive forces, which normally posses between two and four fingers moving synchronously in most cases. This is also justified because this book is devoted to this kind of grippers. Therefore, although there are developments in agriculture of other kinds of grippers (astrictive, contigutive, and ingressive), they will not be discussed in this review.

The main objective of this chapter is to demonstrate the necessity of the use of the impactive mechanical grippers in the robotization of agricultural tasks and, perform a state of art of the main developments of grippers with fingers that are used to handle different materials. This chapter is presented as follows. The next section describes the main characteristics of agricultural tasks and their robotization to understand the necessity of using different types of grippers. They will manipulate vegetal material as plants and fruits, whose main characteristics are indispensable for the gripper design process so, in Sect. 15.3, the basic physics and mechanics properties of these materials are shown related with the agriculture

tasks (transplanting, grafting, pruning, and harvesting) that handle them. Following, a broad State-of-the-Art of agricultural impactive gripper is shown describing in tables their main characteristics sorted and based on the taxonomy proposed by Cutkosky [7], depending on the grasping mode of the grippers: robots with precision grasping and robots with power prehensile grasping. Furthermore, in this section, for each agricultural grasping sorting, a description of a representative agricultural robot has been attached, with the aim of giving more information about the gripper's design and the tasks carried out, with some results of their test in field. The mechanical features, sensors, control systems, and specific electronics for each type of task to be performed are described. Finally, some conclusions are drawn from economic and technological points of view.

## 15.2 Robotization of Agricultural Task

The main phases of the agricultural cycle are soil preparation, planting, production, and harvesting. In each one, some tasks must be performed, some of them being open to robotics.

### 15.2.1 *Soil Preparation*

This phase refers to the mechanical manipulation of soil that alters its structure and strength, in order to provide and maintain optimum soil conditions for germination, growth, and development of plants, thus showing their productive capacity. Tillage is an integral part of the production process, which aims to establish the optimal conditions for the establishment and growth of crops. It has traditional reasons: remove weeds and provide a suitable environment; air the topsoil, incorporate fertilizer or organic matter, increase water holding capacity, and prepare for the risk of flooding. It is performed, with tillers and tractors, once or several times a year depending on the crop. During the soil preparation of outdoor crops, farm machinery exists to conduct this work using tractors with their tackles. These machines must perform multiple operations, react to unexpected disturbances and make decisions at high speeds (sometimes up to 3 m/s) in a dynamic and changing environment as the setting. The first phase of the robot application was the introduction of teleoperation and automatic driving of these systems; many trials exist in Europe, America, and Asia. The second phase is the development of fully autonomous tractors, which connect the various tools to perform tasks, existing commercial systems of important companies like the Orchard tractor developed by John Deere in cooperation with Autonomous Solutions Inc. [66].

A wide review of autonomous agricultural vehicles and robotic field operations can be found in Edan et al. [9], Kondo et al. [26]. The design of this machinery must consider the problems related to working in field environments as the robot

will move dust, lighting changes, and it is subject to abrasive surfaces, extreme weather, and high vibrations, so the sensory system must be suitable for such tasks. Furthermore, the space to move in these environments is usually very small, so it is necessary to develop economic systems that provide location accuracy better than the current standard GPS, like a sensor combination (sensor fusion). On the other hand, in large areas of land more robots can work, so the coordination between them must be optimal, or in closed environments such as greenhouses where there is a danger if used by combustion engines that generate harmful gasses.

### ***15.2.2 Planting***

In outdoor conditions, the planting process consists of dropping the seeds or placing them into the soil. In the case of intensive crops, this process of planting has automated machines that perform this task. It is common to use the services of nurseries that sow the seeds in a given substrate in cavity trays and when the seedlings reach a certain state, they are transplanted to the final place inside the greenhouse. Like in the previous section, there are robotic tractors and agricultural machinery modified to accomplish this task using tractors with appropriate rigging, always referring to outdoor crops based on most of the studies [9, 26]. During design, it must be considered that the end elements have to work well in air, so it must penetrate into the substrate of the seed tray, taking the seedling and transporting it without being damaged.

Other tasks in the planting process are:

- Vegetative multiplication: a type of asexual reproduction employed by plants wherein new independent individuals emerge from the vegetative parts of plants.
- Grafting: a technique whereby tissues from one plant are inserted into those of another so that the two sets of vascular tissues may join together. Salinity, or the pathogens organism present in the soil, cause damage in the roots that reduce the growth and the final production. The use of patterns—plants which are under the soil, chosen for their roots and their resistance to the soil stresses—is a very common practice in agriculture. The patterns have a better adaptation to the subsoil environment and, furthermore, are resistant to the pathogens. On the contrary, the variety—the aerial part of the graft—is chosen for its production, fruit quality, or other agronomical criteria.
- Transplanting process: a technique involving moving a plant from one location to another. The seedlings are transported from the cavity trays to the soil or soilless culture bag, or even a pot. It is very common in intensive agriculture, increasing uniformity, and reducing the planting area. Due to the price of the seeds, and before being taken to the final place, the seedlings spent the first stages of their life in nurseries. In most greenhouses, this task is performed manually, which is very stressful as the holes are dug with a heavy iron bar,

resulting in hand injuries. Planting via this process is therefore very harmful to human health.

There have been several developments to robotize these tasks [29], however, these tasks are still performed manually. Significant manpower is necessary to carry out these tasks and it is a major cost for growers. As they require high precision and work with very delicate material like seeds or new plants, the main innovations should occur in the gripper to carry out the work of cutting and joining of different parts or transporting the seedlings, even coupling with pressure sensors to prevent damage to the plant.

### ***15.2.3 Production***

This phase includes the period between the transplant or the appearance of the first true leaves, and the last harvesting in the case of annual crops. It is composed of tasks such as pesticide spraying, pruning (selective removal of parts of a plant), precision fertirrigation, plague recognition, weed removal, harvesting, and crop removal. Harvesting is the task that requires most inputs and resources and is discussed in the next section. In trees or long-term plants, this period includes from the end of the final harvesting of the last production cycle until the final harvesting of the current cycle (normally referred to as annual periods). During this period, the greatest quantity of resources are consumed, labor and expenses are spent with the aim to obtain the best profit from the production. Because of the duration and quantity of resources used, many researchers have worked on the development of robots to carry out the different tasks included in this phase. This labor requires mobile robots to be able to move among the plants or trees along the corridors intended for them, being careful with the plants in areas where mobility is reduced by the effect of soil and the crops.

Furthermore, the design of the end-effectors is a challenge for the researchers due to the wide variety of material to manipulate (leaves, fruits, stems, etc.) and the different tasks to be done. Some robot developments are worth highlighting [41, 58] for spraying, [3] for a more precise and efficient use of water and fertilizer in irrigation systems, [49] for deleafing plants or [67] for weed control. In Kondo and Ting [29], Edan et al. 2009 [9], Kondo et al. [26] a wide review of robotic developments for these tasks can be found.

### ***15.2.4 Harvesting***

This is the processing of extensive grain crops, corn, and others in which plants are usually cut into the bottom blade-type tackles then collected and packaged in a process without interruption. These operations are conducted primarily by the

different mechanisms that integrate the machine. Specifically, the robotization of this field consists of developing systems to become a standard commercial harvester or tractor in an autonomous vehicle. A project highlight is the Demeter developed by Carnegie Mellon University and New Holland machinery company [51].

Piece harvesting (vegetables and fruits on plants or trees) is an interesting field because of the full task automation has not been solved yet. Furthermore, the economic importance of this task should be highlighted, considering that approximately 11 million hectares of fresh fruit exist in the world, for which the costs of manual harvesting represent around 57 % of the total cost [16].

Technical advances in recent years have brought about considering the challenge of a total harvest robot. The difficulties of achieving this are obvious because the products are very complex and variable, as well as the variability of environmental conditions and seasonality, and socioeconomic factors. When designing a harvesting robot, the morphology to work with irregular volumes must be considered, as well as two key factors:

- Plants and trees can be found in a large area of land, so mobility should be provided to the robot. Therefore, robots are usually harvester hybrid type, i.e., manipulator arms loaded onto platforms or mobile robots.
- In the operation of picking fruit, the robot must pick the fruit and separate it from the plant or tree, so the design of the end-effector is fundamental.

Many designs exist for the robotization of the harvesting process. Some examples of fruits harvested with robotic end-effectors include: dates [1], strawberries [62], tomatoes [38], cherries [64], apples [23], citrus fruits [6], grapes [44], watermelon [55], cabbage [46], radish [11], peppers [25], mushrooms [52], and cucumbers [68]. In Kondo and Ting [29], Li et al. [32, 33], Kondo et al. [26], a wide review of harvesting robots can be found.

In conclusion, agriculture is given special attention with regard to the incorporation of advanced technologies, given the increasing demands of production, diversity, and quality of the products, all with the growing problem of scarcity and high cost of labor.

### 15.3 Vegetable Material Handling

Agricultural robots normally work with two kinds of vegetable material: plants and fruit. The main characteristics of each one are indispensable for the gripper design. These materials are characterized by their properties in the Table 15.1. Although others can be found, like chemical properties (cell water content, lignin content or maturity), the above mentioned directly influence the mechanical characteristics. Furthermore, other characteristics of vegetable material are the optical properties (color, reflectance, transmittance, etc.), these are important for the fruit detection and location, both tasks being fundamental parts of the sensorial. This system is

Table 15.1 Main characteristics of the vegetable products

Properties	Transplanting grafting		Pruning		Harvesting	
	Seedlings	Plants and plant parts	Vegetable and tree fruits	Whole plants		
Mechanics	Handling resistance Cutting resistance Break resistance	Size Shape Weight Placing	Size Shape Weight Placing	Sphericity (roundness) Texture	Elasticity Deformation resistance Compressibility Skin resistance Firmness	Handling resistance Cutting resistance Elasticity Impact resistance Firmness
Basic physics	Size	Size Shape Weight Placing	Size Shape Weight Placing	Symmetry	Size Shape Weight Placing	Handling resistance Cutting resistance Impact resistance Frictional resistance Cutting resistance Viscosity

not part of the gripper, so in this chapter is not described. Both property types (chemical and optical) are not determinants for the gripper design although they must be taken into account in the design of the agricultural robots.

The agricultural material is as wide that each variety, fruit, etc., presenting its own properties. As melons and tomatoes are different sizes, so the harvesting gripper has to be distinct. Cucumbers and zucchinis have a similar size but different mechanical resistance, etc. The main physical and mechanical properties of plants and fruits that must be taken into account during the gripper's design stage. Furthermore, it is important for the design of the gripper to have knowledge of the necessary tasks, for example if the gripper must carry them out and not only hold the vegetable material.

In grafting, pattern and variety are joined through a grafting technique, which is a precise and repetitive task where workers use sharp cutting tools (that may cause hand injuries). Grafts are mainly used in hi-tech agriculture and in fruit trees, because of the high cost of doing so. The process is carried out over two clearly defined stages; in the first stage, the robot will work with the pattern and in the second with the variety. Both processes are very similar, the plants are taken carefully by the gripper and then they are cut [47, 69]. Both cuttings must have symmetrical shapes. Finally, the varieties have to be pasted in the last stage of the grafting process. The handling and cutting resistance is basic for this process, so the seedling stems are very fragile due to water-content and the lignin absence in their cells. The Gripper–cutter–joiner end-effector must be designed carefully in order to carry out this task adequately.

Transplanting is not an easy process to perform [65] as the seedlings are very delicate and could break, given the low resistance to breakage because of the high cell water content. Besides, it is important to take into account that in many cases it is necessary to maintain seedling rootballs, formed by roots and substrate which could be damaged (roots) or dismantled (substrate) due to the mechanical transplanting action. The appropriate task fulfillment is basic for the good development of the seedlings with or without rootballs. The robots must locate the seedling in a cavity tray and put it into the substrate where it is cultivated. This last operation is even more complicated as the plantlets and the rootballs could be damaged, which affects the crop growth and the uniformity expected in these kinds of crops.

The task with the greatest number of developments in recent years is harvesting. Fruit or plant picking is a complex process due to plenty of different shapes, mechanical characteristics, and weight and sizes that can be found in the crops. All of these properties make it impossible to design an end-effector for all fruits; hence such patterns have to be more specific for each task. Gripper designs mainly depend on the shape, weight, and size of the products which are harvested [29]. Although others properties and their relationship with the fruits and vegetable characteristic are also important, as the compressibility which is closely related to the locule number, the symmetrical internal structure and the loading position, increasing it when the locule number decreases, when the internal structure is symmetric and when the loading slope is lower [37], skin resistance and loss firmness which was tried out through impact testing [2], impact resistance that is increased where the

lobule wall begins [34, 35], some mechanical properties trials as the frictional resistance tests, for example in Kabas and Ozmerzi [22] the static and dynamic coefficients of friction and the deformation resistance curves for cherry tomato varieties are shown; the elasticity and viscosity are tested in Li et al. [36].

## 15.4 Classification of Grippers

In this section, a summary of the most relevant grippers in agricultural robotics is shown. In order to classify them, the taxonomy proposed by Cutkosky [7] is used. In the following sections, two tables are shown summarizing the characteristics of the most important grasping robots found in the academic literature. Both tables include seven fields:

- Target product, showing the type of manipulated objects (flowers, bedding, fruits, plants...).
- Number and type of fingers, and finger mechanism.
- Staying function shows if the end-effector has an element to hold the products.
- Cutting shows if the end-effector has an element to cut the target product.
- Specific type of grasping from Cutkosky [7].
- Reference, containing academic references.

In agricultural robotics, grasping robots can be classified in two large groups depending on the grasping mode of the grippers: robots with precision grasping and robots with power prehensile grasping.

### 15.4.1 Precision Grippers

In precision grasping, the object is held with the tips of the finger and thumb. Here, two types of precision grasping are analyzed: prismatic (for long objects, using the thumb opposed to the rest of the fingers, picking the objects like two virtual fingers), and circular (for objects with radial symmetry, using all the fingers in similar way, picking the object like three virtual fingers) [7]. Based on this sorting, the Table 15.2 shows a summary with the most important characteristics of agricultural grippers in academic literature.

Two representative robots have been chosen as examples of precision grippers. The cucumber harvester of Van Henten et al. [68] is selected as an example of prismatic precision grasping. This robot (Fig. 15.1) consists of an autonomous vehicle that uses the heating pipes mounted on the ground of the greenhouse as a rail, a manipulator (7DOF Mitsubishi RV-E2) mounted on a linear slide for the positioning of the end-effector during the harvesting operation, two camera vision systems, and an end-effector (Fig. 15.1).

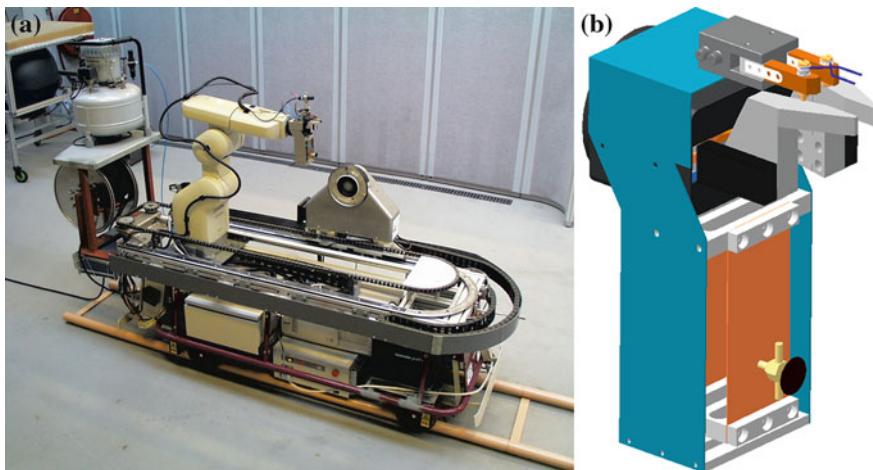
**Table 15.2** Summary of gripper characteristics of robots in academic literature for precision grasping

Tasks/target product	Number and type of fingers	Finger mechanism	Staying function	Cutting	Grasping classification	Reference
Apple	2 parallel fingers	Rigid fingers	No	No	Prismatic	[4]
Asparagus	2 parallel fingers	Water-filled fingertips	No	No	Prismatic	[42]
Asparagus	2 angular fingers	Rigid fingers	No	Cutting knife	Prismatic	[19]
Bedding plants	2 fingers	Rigid prismatic fingers actuated by an air cylinder	No	No	Prismatic	[65]
Bedding plants	2 fingers	2 Rigid fingers	No	No	Prismatic	[31]
Bedding plants	2 parallel fingers	Rigid fingers	A pushing device	No	Prismatic	[44]
Bunch of grapes	2 fingers enabled to penetrate into the soil by air cylinders	Rigid shovel-shaped fingers describing a triangle when closed	No	No	Prismatic	[54]
Cherry	2 angular fingers	Rigid fingers equipped with soft rubber components	Suction	Back-and-forth mechanism	Prismatic	[64]
Cucumber	2 parallel fingers	Rigid elements attached to typical rigid fingers	Suction	Thermal cutting device	Prismatic	[68]
Date General fruits	2 angular fingers 3 articulated fingers	Rigid curved fingers 2 mobile within a circular track around the palm central and one fixed articulated fingers: a configurable grasping mode	No No	No No	Prismatic Configurable	[11] [61]
Melon	2 types with 2 fingers	2 rigid fingers V-V type notch edge and 2 fingers for a basked gripper	No	No	Prismatic	[8]
Phalaenopsis	2 parallel fingers	Stainless steel forceps attached to typical rigid fingers	No	No	Prismatic	[17]
Saffron flowers	2 elements, a movable cam and a fixed striker	Rigid elements with an special geometry	No	Cam-striker mechanism	Prismatic	[53]

(continued)

**Table 15.2** (continued)

Tasks/target product	Number and type of fingers	Finger mechanism	Staying function	Cutting	Grasping classification	Reference
Silkworm	2 angular fingers	2 DoF fingers with sector-shaped fingertips made of silicone rubber and with air No pressure as power source	No	No	Prismatic	[48]
Spherical fruit	2 parallel fingers	Upper finger with a groove with "V" shape front part	Suction	Laser	Circular	[39]
Spherical fruit	2 parallel fingers	Rigid fingers	Suction	Laser	Circular	[40]
Strawberry	2 parallel fingers	Rigid fingers	No	Mechanical cutter	Prismatic	[27]
Strawberry	2 parallel fingers	Rigid fingers	Suction	One finger is shaped like a cutter	Prismatic	[15]
Tomato	2 parallel fingers	Rigid fingers	Suction	No	Circular	[45]
Tomato cluster	2 fingers, one movable and one fixed	Rigid "U" shaped fingers	A pushing device	Cutter attached to upper finger	Prismatic	[28]



**Fig. 15.1** A cucumber harvesting robot (courtesy of the authors). **a** Cucumber harvesting robot, **b** cucumber harvesting gripper

The end-effector contains a gripper with two fingers and a suction cup to grasp the fruit (cucumbers), and a thermal cutting device to separate the fruit from the plant. The gripper is a modified version of the Mitsubishi gripper 1E-HM01. The robot was tested in an experimental greenhouse, with an average speed of 45 s per cucumber, and with a success rate of 80 %. In essence, the failures were due to inaccuracies in the position estimation.

The operation is as follows: with a 3D vision system the cucumber is located; after that using the fingers of the gripper, the robot grips the stalk of the cucumber; after that the fruit is cut and with the suction cup the cucumber (detected by a computer vision system) is immobilized during the transportation stage. On average, the robot needs 45 s to pick one cucumber. The harvesting robot has been used in a greenhouse with a success rate of 80 %. It is important to note that thermal cutting creates an important improvement versus the traditional knife-based cutting. This is due to the fact that when the same knife is used to cut the stalk of the fruit over and over again, there is a risk of virus transmission from one plant to the other, so a cleaning mechanism must be defined to remove any organic residual after a cutting operation. This approach is not practical for a robotics application. Furthermore, with thermal cutting the wounds of both the fruit and the plant are closed during the cutting process.

In Liu et al. [40], a spherical universal fruit harvesting robot is demonstrated, chosen as an example of circular precision grasping. The gripper of this robot (Fig. 15.2) is universal for spherical fruit such as tomatoes, apples, and citrus. This element connects with the wrist of the industrial manipulator Yaskawa Motoman SV3X. The end-effector includes a gripper with two fingers (to grip the fruit), a laser cutting device (to cut the peduncle), and a vacuum suction pad device (to separate the pieces of fruit within the same cluster). Once the fruit is picked by the



**Fig. 15.2** The end-effector for the spherical universal harvesting robot (courtesy of the authors)

gripper, the peduncle was cut by the laser, driven by an actuating mechanism composed of focusing lens (connected with the laser by a fiber), DC motor, and a bearing system (it will drive the focusing lens to rotate to cut stems).

#### **15.4.2 Power Prehensile Grippers**

Power grippers are distinguished by large areas of contact between the grasped object and the surface of the fingers and palm and by little or no ability to impart motions with the fingers [7]. The agricultural power grippers require clamping of the vegetable material, so all of them belong to the prehensile type. Similar to the precision gripper, this gripper can be classified into prismatic (for long objects, picking them with wrap symmetry) and circular (for compact objects, picking them with radial symmetry). In both cases, the fingers surround parts of the objects. The Table 15.3 summarizes the characteristics of the most relevant agricultural grippers in the academic literature for power prehensile grasping.

Two types of power prehensile grasping are considered as example of power grippers. A robot for heavy fruits handling proposed by Sakai et al. [56] whose

**Table 15.3** Summary of gripper characteristics of harvesting robots in academic literature for power prehensile grasping

Tasks/target product	Number and type of fingers	Finger mechanism	Staying function	Cutting function	Grasping classification	Reference
Apple	2 angular fingers	Spoon-shaped fingers	No	No	Circular sphere	[13]
Apple	4 angular fingers	4 rigid fingers padded with sponge	No	No	Circular sphere	[23]
Apple	2 angular fingers	Spoon-shaped fingers	No	Electric cutting device	Circular sphere	[70]
Cabbage	4 angular fingers	4 shovel-shaped rigid fingers, actuated by 2 motors	No	Blades attached to the fingertips of two fingers	Circular sphere	[46]
Eggplants	6 angular fingers	2 L-shaped rigid fingers and 5 rubber actuators	Suction	Scissors	Prismatic	[14]
Food-products with varying shapes	2 parallel fingers one movable and one fixed	Polyurethane pouches filled with magnetorheological fluid attached to rigid fingers	No	No	Prismatic	[50]
Fruits with varying shapes	3 angular fingers	Rubber fingers	No	Scissors	Circular sphere	[12, 59]
General fruits	3 articulated fingers	2 mobile within a circular track around the palm central and one fixed articulated fingers: a configurable grasping mode	No	No	Configurable	[61]
Kiwi	2 angular fingers	Rigid curved fingers	No	No	Circular sphere	[60]
Raddish	2 angular fingers	Bucket-like fingers	No	Linear blade attached to the fingertips	Circular sphere	[11]

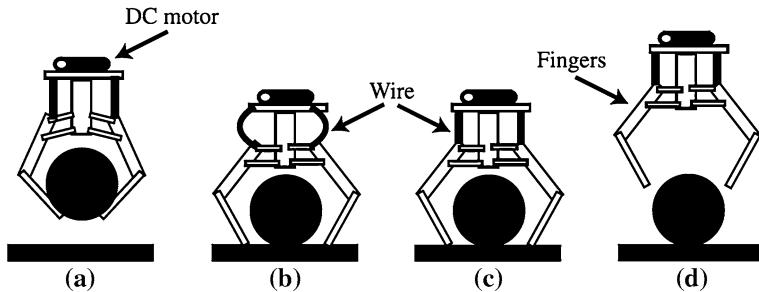
(continued)

**Table 15.3** (continued)

Tasks/target product	Number and type of fingers	Finger mechanism	Staying function	Cutting	Grasping classification	Reference
Slugs	3 angular fingers	3 rigid fingers with wiper blades, underneath each of the three wiper blades is a plate which produces passive alignment with the contours of the ground	No	No	Circular sphere	[24]
Tomato	4 angular fingers	Underactuated and articulated fingers made of square and rectangular Acetyl Butyl Styrene	Suction	No	Circular sphere	[38]
Watermelon	4 angular fingers	Rigid curved fingers	No	No	Circular sphere	[56, 57]



**Fig. 15.3** The end-effector for the robot (courtesy of the authors)

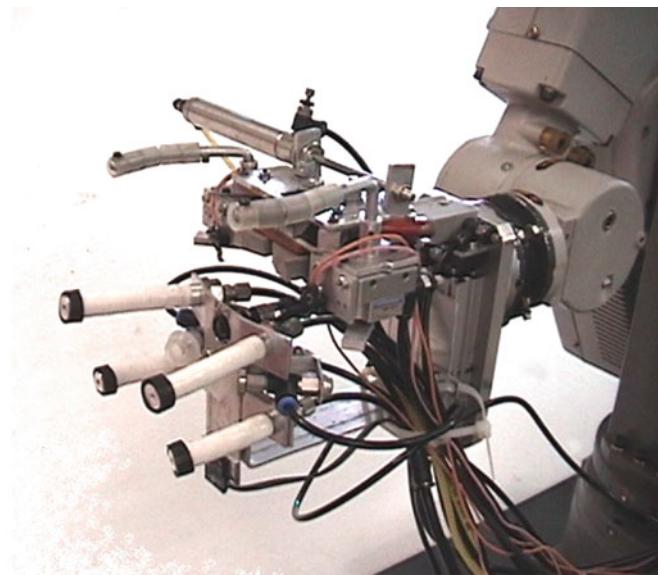


**Fig. 15.4** Motion plan for placing using reaction forces for the robot (courtesy of the authors)

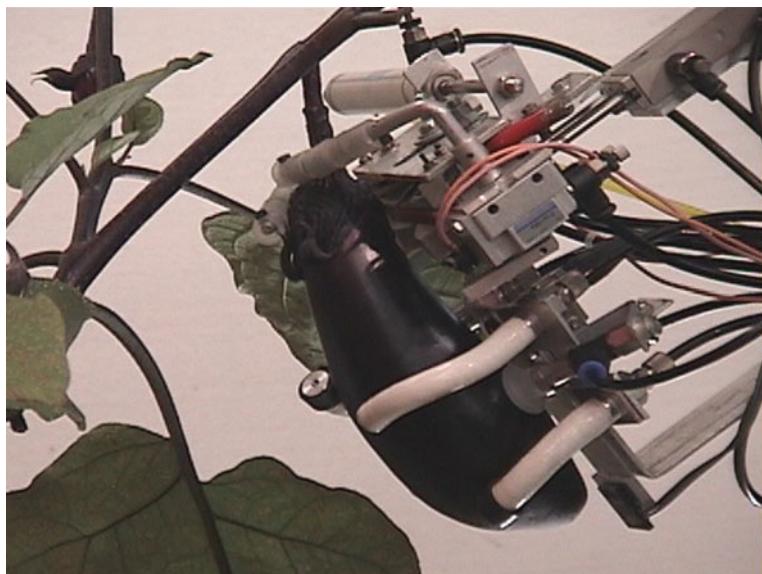
end-effector consists of four 1-DOF fingers with no actuators (Fig. 15.3). The picking uses passive force. A wire mechanism is implemented to place the heavy fruit using less power (Fig. 15.4). First, due to gravity the fingers remain closed (Fig. 15.4a).

When the gripper makes contact with the ground the wire tension is zero because of the reaction forces, and the fingers open (Fig. 15.4b). After that, using a low-power DC motor the wires are reeled (Fig. 15.4c) so that when the gripper rises there is no contact with the fruit (Fig. 15.4d). This gripper has been tested performing watermelon harvesting experiments in open field, grasping watermelons with a mass between 3 and 10 kg. The percentage of successful operations to the total trials was 86.7 %.

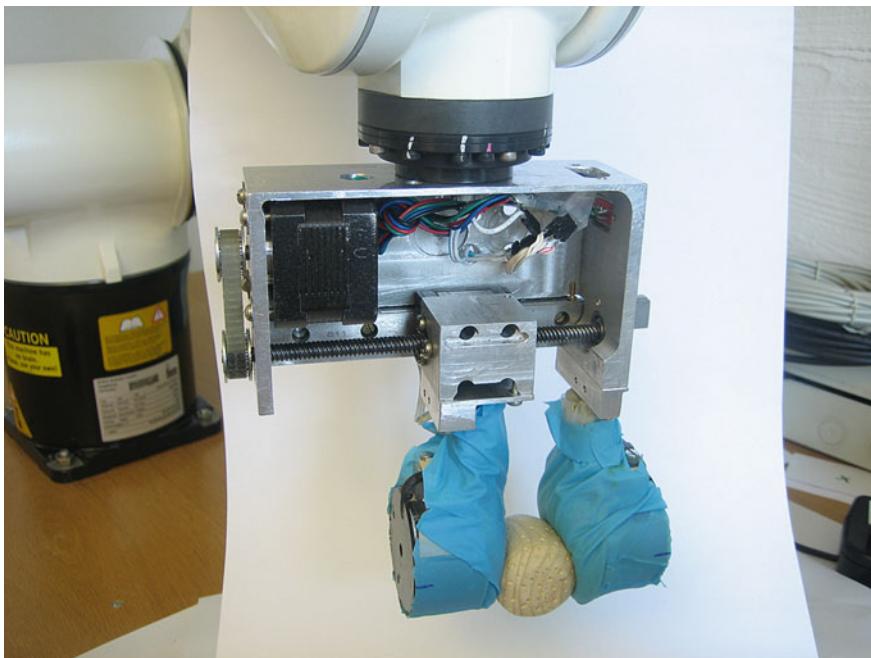
In Hayashi et al. [14], an eggplant harvesting robotic system is presented. This robot is composed of a vision system, an industrial manipulator arm with five DOF, and an end-effector (Figs. 15.5 and 15.6). The CCD camera is attached to the center of the end-effector and a photoelectric sensor is attached to the bottom. The



**Fig. 15.5** The end-effector for the harvesting system (courtesy of the authors and Japan Agricultural Research Quarterly, JARQ)



**Fig. 15.6** The gripper for the harvesting system picking an eggplant (courtesy of the authors and Japan Agricultural Research Quarterly, JARQ)



**Fig. 15.7** The MR fluid gripper holding a model strawberry (courtesy of the authors)

end-effector consists of a gripper (with four rubber actuators), a size-judging mechanism (with two suction pads), and a peduncle cutting mechanism (with scissors). Two additional fingers are used as a guideline for the fruit-picking operation, specifically to catch the peduncle. All the mechanisms are actuated with 0.4 MPa of compressed air. A laboratory experiment was done with a successful harvesting rate of 62.5 % and with an average speed of 64 s per eggplant.

A case to highlight is the gripper for delicate fruits handling developed in Pettersson et al. [50]. The robot gripper (Fig. 15.7), mounted on a six-axis KUKA manipulator, utilizes the effects of a magnetorheological (MR) fluid (suspensions of micron-sized polarizable particles, oil, or water with iron particles). In presence of a magnetic field, the fluid exhibits yield strength. In the gripping process, two polyurethane pouches filled with this fluid are molded around the product's contours. Activating an electromagnet in the gripper, a large increase in the MR fluids yield stress (strength), confines the fruit in the mold produced by the fingers (consisting of two electromagnets and two polyurethane pouches). The gripper has two parallel fingers, one movable and one fixed. The gripper uses a stepper motor with a linear bearing and a ball screw, providing a step resolution of 0.031 mm/step and a maximum grip separation of 79 mm.

## 15.5 Conclusion

This chapter gives a wide review of robotization technology applied in agriculture. From an economic and working point-of-view, the main advantages of the robotization of agricultural tasks are: manpower replacement for the lack in the numbers of effectives; worker replacement in hazardous tasks on health such as pesticide spraying; performing repetitive and tedious tasks such as picking fruit or plant tutoring; allowing tasks to be carried out in the evening hours which leads to a long time and cost saving, for example during harvesting; the need for precision in some agricultural tasks such as those related to biotechnology, in particular the plants propagation from plant tissue; improving efficiency and quality of some of the tasks and uniformity in the conduct of holes for transplanting; environmental risk reduction as reducing the amount of plant protection product (PPP) that is emitted into the air; cost savings (not due to labor) because it decreases the amount of fuel and products used in certain areas; and improving the quality of products such as the use of fewer pesticides.

Although nowadays the state of robotics is very advanced and its application is clear within the agricultural field, in practice it is not easy to develop robots with high performance levels to carry out agricultural tasks because of the environment where the robots must work, the objects which they must manipulate and their locations are variable. For this reason, the use of robotics in agriculture is limited compared to its use in industrial sector. In fact, a commercial system does not exist, so most have been developed in universities and other research centers, although some companies offer robotic systems for agricultural tasks related with autonomous navigation, but never for vegetable material manipulation. Even so, it is undoubted that agriculture presents challenges and thrilling opportunities to develop robotic systems.

From a technological point-of-view, agricultural robot developments are composed of an autonomous or guided mobile platform (not covered in this chapter) and a manipulator (from one to eight or more arms). In fact, the developments are focused on the sensorial systems and on the end-effectors, so they usually use commercial robotic arms. Related with the end-effectors, the grippers are widely used due to the nature of the vegetable material and the performance tasks. This chapter shows wide grippers' State-of-the-Art which can draw the following conclusions:

- In the gripper designs, it is essential to consider the physical and mechanical properties of the vegetable material.
- Performance, task efficiency, and final cost must be considered in the gripper design process.
- There are no universal clamps, it is necessary to design one for each product or type (shape, size and weight) to manipulate.
- The harvesting process is where the greatest number of developments exists. In these end-effectors, the inclusion of fruit subjection elements and a stem or peduncle cutting system is typical.

- It would be convenient for the grippers to have fingers that adapt to the shape of the vegetal material, wrapping better, and avoiding damage.
- In this study, it can be deduced that the precision grippers can be designed using only two fingers to perform tasks. On the other hand, power grippers typically need at least four fingers, with some developments with two and three fingers having been found.
- Some designs of agricultural grippers include many actuators and sensors which hinder the control and increase the cost. So, in general the complexity of these subsystems must be reduced.
- The Cutkosky sorting is useful to classify, analyze, and understand the different agricultural grippers; it is even helpful to design them based on how they must wrap the vegetable material.
- Most proposals come from Asian countries such as Japan and Korea followed by European developments.

Although a large quantity of agricultural robot developments exist due to the agricultural sector is especially important in these places, it is necessary to continue working hard to improve their performance, specially the system to locate the fruit or vegetable, in order to introduce such robotic systems into the commercial market. In the following outlook, the main problems for the agricultural-robots establishment are presented:

- The main technical problems in agricultural robotics are: the fruits and vegetables recognition and location (in general with changing light conditions and with a great variety of different characteristics depending on the fruit or vegetable), the robot operation speed (in general much lower than the human operator's speed), and the storage ways (in order to automate all the stages for the harvesting process this aspect must be considered). Actually, the grippers are not a key problem; only one consideration must be taken into account, different structures must be used depending on the product to grasp.
- Farms, greenhouses, or crops adaption for the use of robots, not only taking into account the robot movement, otherwise growth regulation (pruning, defoliation, etc.) is an important tool to get that, for example, the fruits are more visible and easily to be recognized and localized. Most of the farms are not adapted to use the robots and the crops are not regulated and, for instant, leaves hide the fruits (tomato plants) or the fruits growth in difficult localizations (orange trees). The adaptation is a big important investment that not many growers are willing to make.
- From an economic point-of-view, the use of robots in farms is not clear due to the narrow profit of many agricultural areas hinder the acquisition of this kind of systems. There are few studies on this issue and some of them conclude that the payback time of a harvesting robot would be about 5 years. For this reason, it is essential use the programmable and flexibility capabilities of a robot to use it in different tasks to make it profitable before. In this process, the grippers have the main role.

**Acknowledgments** This work has been funded by the—Controlcrop—Project, PIO-TEP-6174, from the Counseling of Economy, Innovation and Science, Government of Andalusia (Spain). The authors are grateful for the invaluable contributions of the Experimental Station of the Cajamar Fundation and the Agrifood Campus of International Excellence (ceiA3).

## References

1. Aljanobi AA, Al-hamed SA, Al-Suhaimi SA (2010) A setup of mobile robotic unit for fruit harvesting. In: Proceedings of the 19th international workshop on robotics in Alpe-Adria-Danube region—RAAD 2010, Budapest, Hungary, 23–25 June 2010
2. Arazuri S, Jaren C, Arana JI, Perez de Ciriza JJ (2007) Influence of mechanical harvest on the physical properties of processing tomato (*Lycopersicon esculentum* Mill.). *J Food Eng* 80 (1):190–198
3. Belforte G, Deboli R, Gay P, Piccarolo P, Ricauda-Aimonimo D (2006) Robot design and testing for greenhouse applications. *Biosyst Eng* 95(3):309–321
4. Bulanon DM, Kataoka T, Okamoto H, Hata S (2004) Determining the 3-D locator of the apple fruit during harvest. In: Proceedings of the automation technology for off-road equipment, Kyoto, Japan, 7–8 Oct 2004
5. CCC/CRA Roadmapping for Robotics (2009). A roadmap for US robotics, from internet to robotics, 94 pp
6. Ceres R, Pons JL, Jiménez AR, Martín JM, Calderón L (1998) Design and implementation of an aided fruit-harvesting robot (Agribot). *Ind Robot Int* 25(5):337–346
7. Cutkosky MR (1989) On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Trans Robotics Autom* 5(3):269–279
8. Edan K, Haghghi R, Stroshine M, Cardenas-Weber M (2006) Robot gripper analysis: finite element modeling and optimization. *Appl Eng Agric* 8(4):563–570
9. Edan Y, Han S, Kondo N (2009) Automation in agriculture. In: Shimon Nof (ed) Chapter 63 of handbook of automation. Springer, West Lafayette, pp 1095–1128
10. EURON (2004) EURON research roadmap. European robotic research network, 137 pp
11. Foglia M, Reina G (2006) Agricultural robot for radicchio harvesting. *J Field Robot* 23 (6/7):363–377
12. Fujiura T, Ura M, Kawamura N, Namikawa K (1990) Fruit harvesting robot for orchard. *J Jpn Soc Agric Mach* 52(1):35–42
13. Guo J, Dean Z, Ji W, Xia W (2010) Design and control of the open apple-picking-robot manipulator. In: Proceedings of the 3rd IEEE international conference on computer science and information technology (ICCSIT), vol 2, pp 5–8
14. Hayashi S, Ganno K, Ishii Y, Tanaka K (2002) Robotic harvesting system for eggplants. *Jpn Agric Res Q* 36(3):163–168
15. Hayashi S, Shigematsu K, Yamamoto S, Kobayashi K, Kohno Y, Kamata J, Kurita M (2010) Evaluation of a strawberry-harvesting robot in a field test. *Biosyst Eng* 105:160–171
16. Hortwatch (2002) Post harvest—a time to review the orchard performance. Product of AgFirst. <http://www.hortwatch.com/library/post-harvest.html>
17. Huang Y-J, Lee F-F (2010) An automatic machine vision-guided grasping system for *Phalaenopsis* tissue culture plantlets. *Comput Electron Agric* 70:42–51
18. IFR Statistical Department (2010) Executive summary of the study world robotics 2010, 8 pp
19. Irie N, Taguchi N, Horie T, Ishimatsu T (2009) Asparagus harvesting robot coordinated with 3-d vision sensor. In: Proceedings of the IEEE international conference on industrial technology, Gippsland, VIC, pp 1–6
20. Japan Robot Association (2001) Summary report on technology strategy for creating a robot society in the 21st century. Japan Robot Association, Japan 28 pp

21. Jian S, Tiezhong Z, Liming X (2006) Actuality and prospect of picking robot for fruits and vegetables. *Trans Chin Soc Agric Mach* 37(5):158–162
22. Kabas O, Ozmerzi AZIZ (2007) Determining the mechanical properties of cherrymato varieties for handling. *J Texture Stud* 39(3):199–209
23. Kassay L (1992) Hungarian robotic apple harvester. Paper ASAE No. 92-7042 St. Joseph, MI 49085. ASAE annual meeting papers 1992, pp 1–14
24. Kelly I, Melhuish C (2001) SlugBot: a robot predator. proceedings of the European conference on artificial life (ECAL). In: Lecture notes in artificial intelligence (LNAI), vol 2159, pp 519–528
25. Kitamura S, Oka K (2005) Recognition and cutting system of sweet pepper for picking robot in greenhouse horticulture. In: Proceedings of the IEEE international conference on mechatronics and automation, vol 4, 1807–1812
26. Kondo N, Monta M, Noguchi, N (2011) Agricultural robots: mechanisms and practices. Trans Pacific Press, Melbourne, p 348
27. Kondo N, Ninomiya K, Hayashi S, Ota T, Kubota T (2005) A new challenge of robot for harvesting strawberry grown on table top culture. ASAE annual international meeting, Tampa, Florida
28. Kondo N, Taniwaki S, Tanihara K, Yata K, Monta M, Kurita M, Tsutumi M (2007) An end-effector and manipulator control for tomato cluster harvesting robot. In: 2007 ASABE annual international meeting, Minneapolis, Minnesota
29. Kondo N, Ting KC (1998) Robotics for bioproduction systems. ASAE Publications, St. Joseph, p 325
30. Kondo N, Yamamoto K, Yata K, Kurita M, (2008) A machine vision for tomato cluster harvesting robot. In: ASABE annual international meeting, Providence, Rhode Island
31. Kutz LJ, Miles GE, Hammer PA, Krutz GW (1987) Robotic transplanting of bedding plants. *Trans ASAE* 30(3):586–590
32. Li Q, Hu T, Wu Ch, Hu X, Ying Y (2008) Review of end-effectors in fruit and vegetables harvesting robot. *Trans Chin Soc Agric Mach* 39(3):175–179
33. Li ZG, Liu JZ, Li PP (2008) Analysis of workspace and kinematics for a tomato harvesting robot. In: Proceedings of international conference on intelligent computation technology and automation, Changsha, China
34. Li ZG, Liu JZ, Li PP (2009) Study on the collision mechanical properties of tomatoes gripped by harvesting robot. *Afr J Biotechnol* 8(24):7000–7007
35. Li ZG, Liu JZ, Li PP (2010) Relationship between mechanical property and damage of tomato during robot harvesting. *Trans CSAE* 26(5):112–116
36. Li ZG, Li PP, Liu JZ (2010) Effect of tomato internal structure on its mechanical properties and degree of mechanical damage. *Afr J Biotechnol* 9(12):1816–1826
37. Li ZG, Li PP, Liu JZ (2011) Physical and mechanical properties of tomato fruits as related to robot's harvesting. *J Food Eng* 103(2):170–178
38. Ling PP, Ehsani R, Ting KC, Chi Y-T, Ramalingam N, Klingman MH, Draper C (2004) Sensing and end-effector for a robotic tomato harvester. In: 2004 ASAE/CSAE Annual International Meeting, Ottawa, Ontario, Canada
39. Liu J, Li P, Li Z (2007) A multi-sensory end-effector for spherical fruit harvesting robot. In: Proceedings of the IEEE international conference on automation and logistics, Jinan, China
40. Liu J, Li Z, Li P, Mao H (2008) Design of a laser stem-cutting device for harvesting robot. In: Proceedings of the IEEE international conference on automation and logistics, Qingdao, China, pp. 2370–2374
41. Mandow A, Gómez-de-Gabriel JM, Martínez JL, Muñoz VF, Ollero A, García-Cerezo A (1996) The autonomous mobile robot AURORA for greenhouse operation. *IEEE Robot Autom* 3(4):18–28
42. Mattiazzo G, Mauro S, Raparelli T, Velardocchia M (1995) A fuzzy controlled pneumatic gripper for asparagus harvesting. *Control Eng Pract* 3(11):1563–1570
43. Monkman GJ, Hesse S, SteinmannR, Schunk H (2007) Robot grippers. Wiley-VCH, Weinheim, p 463

44. Monta M, Kondo N, Shibano Y (1995) Agricultural robot in grape production system. In: Proceedings of the IEEE international conference on robotics and automation, pp 2504–2509
45. Monta M, Kondo N, Ting KC (1998) End-effectors for tomato harvesting robot. *Artif Intell Rev* 12:11–25
46. Murakami N, Otsuka K, Inoue K, Sugimoto M (1995) Robotic cabbage harvester. In: Proceedings of international symposium on automation and robotics in bioproduction and processing, vol 2. Kobe, Japan, pp 25–32
47. Nishiura Y, Murase H, Honami N, Taira T (1995) Development of plug-in grafting robotic system, 1995. In: IEEE international conference on robotics and automation, vol 3. Chengdu, China, pp 2510–2517
48. Ohura M, Takanobu H, Ishihara H, Aizawa T, Watanabe Y, Miura H (2005) Development of a silkworm handling robot. ASAE Annual International Meeting, Tampa, Florida
49. Ota T, Bontsema J, Hayashi S, Kubota K, Van Henten EJ, Van Os EA, Ajiki K (2007) Development of a cucumber leaf picking device for greenhouse production. *Biosyst Eng* 98(4):381–390
50. Pettersson A, Davis S, Gray JO, Dodd T, Ohlsson T (2010) Design of a magnetorheological robot gripper for handling of delicate food products with varying shapes. *J Food Eng* 98:332–338
51. Pilarski T, Happold M, Pangels P, Ollis M, Fitzpatrick K, Stentz A (2002) The demeter system for automated harvesting. *Auton Robot* 13:9–20
52. Reed JN, Miles SJ, Butler J, Baldwin M, Noble R (2001) Automation and emerging technologies: Automatic mushroom harvester development. *J Agric Eng Res* 78(1):15–23
53. Ruggiu M, Manuello Bertetto A (2006) A mechanical device for harvesting *crocus sativus* (saffron) flowers. *Appl Eng Agric* 22(4):491–498
54. Ryu KH, Kim G, Han JS (2001) Development of a robotic transplanter for bedding plants. *J Agric Eng Res* 78(2):141–146
55. Sakai S, Iida M, Umeda M (2002) Heavy material handling manipulator for agricultural robot. In: Proceedings of the IEEE international conference on robotics and automation. Washington DC, USA, pp 1062–1068
56. Sakai S, Iida M, Osuka K, Umeda M (2008) Design and control of a heavy material handling manipulator for agricultural robots. *Auton Robot* 25:189–204
57. Sakai S, Osuka K, Maekawa T, Umeda M (2005) Active vision of a heavy material handling agricultural robot using robust control, a case study for initial cost problem. In: 2005 IEEE/RSJ international conference on intelligent robots and systems, pp 1707–1713
58. Sanchez-Hermosilla J, Rodriguez F, Gonzalez R, Guzman JL, Berenguel M (2010) A mechatronics description of an autonomous mobile robot for agricultural tasks in greenhouses. *Mobile Robots Navigation*, Intech Publisher
59. Sarig Y (1993) Robotics of fruit harvesting: a state-of-the-art review. *J Agric Eng Res* 54:265–280
60. Scarfe AJ, Flemmer RC, Bakker HH, Flemmer CL (2009) Development of an autonomous kiwifruit picking robot. In: Proceedings of the 4th international conference on autonomous robots and agents, Wellington, New Zealand
61. Shanjun L, Shuangji Y, Yanlin Z, Liang M, Zhen L (2011) Grasp planning analysis and strategy modeling for underactuated multi-fingered robot hand using in fruits grasping. *Afr J Agric Res* 6(9):2086–2098
62. Shiigi T, Kondo N, Kurita M, Ninomiya K, Kamata J, Hayashi T, Kobayashi K, Shigematsu K, Kohno Y (2008) Strawberry harvesting robot for fruits grown on vegetable top culture. In: ASABE Paper No. 084046, ASABE Annual International Meeting, 2008, Providence, Rhode Island, USA
63. Tang X, Zhang T, Liu L, Xiao D, Chen Y (2009) A new robot system for harvesting cucumber. In: ASABE Paper No. 096463, ASABE annual international meeting, 2009, Reno, Nevada
64. Tanigaki K, Fujiura T, Akase A, Imagawa J (2008) Cherry-harvesting robot. *Comput Electron Agric* 63:65–72

65. Ting KC, Giacomelli GA, Shen SJ, Kabala WP (1990) Robot workcell for transplanting of seedlings part II—end-effector development. *Trans ASAE* 33(3):1013–1017
66. Torrie M, Cripps D, Swensen J (2002) Joint architecture for unmanned groundsystems (JAUGS) applied to autonomous agricultural vehicles. In: Proceedings of the ASAE conference, Chicago, Illinois, pp 1–12
67. Van Henten EJ, Van Tuyl BAJ, Hoogakker GJ, Van Der Weerd MJ, Hemming J, Kornet JG, Bontsema J (2004) An autonomous robot for de-leafing cucumber plants grown in a high-wire cultivation system. In: Proceedings of ISHS international conference on sustainable greenhouse systems, vol 691. Leuven, Belgium, pp 877–884
68. Van Henten EJ, Hemming J, Van Tuyl BAJ, Kornet JG, Meuleman J, Bontsema J, Van Os EA (2002) An autonomous robot for harvesting cucumbers in greenhouses. *Auton Robot* 13:241–258
69. Yamada H, Buno S, Koga H, Uchida K, Ueyama M, Anbe Y, Mori H (1995) Development of a grafting robot. In: Proceedings of international symposium on automation and robotics in bioproduction and processing, vol 3. Kobe, Japan, pp 71–78
70. De-An Zhao, Ji W, Zhang Y, Chen Y (2011) Design and control of an apple harvesting robot. *Biosyst Eng* 110(2):112–122

# Chapter 16

## Microgrippers and their Influence on High Precision Assembly Processes

A. Raatz and R. J. Ellwood

**Abstract** Microgrippers have their roots in macrogrippers, with many of the first microgrippers being scaled down versions of their larger counterparts. Although similar, with the smaller handled parts comes a different set of requirements. After presenting an overview of these requirements, different microgrippers are presented. Of these, a more detailed discussion of mechanical, vacuum, electrostatic, capillary, and freezing microgripping technologies is presented. In the last portion of this chapter, three different microgrippers will be used for the same microassembly task. Within these examples, it is shown how gripper design plays a large role in obtaining the required assembly tolerances. In one example, the gripper design is modified to improve the available information about the assembly scene, allowing an improvement in the resulting assembly uncertainty. In a further example, hot melt adhesives are used with both passive and active microgripping solutions. Through these examples, it is shown how gripper design is an important part in improving the assembly uncertainty within microassembly.

### 16.1 Introduction

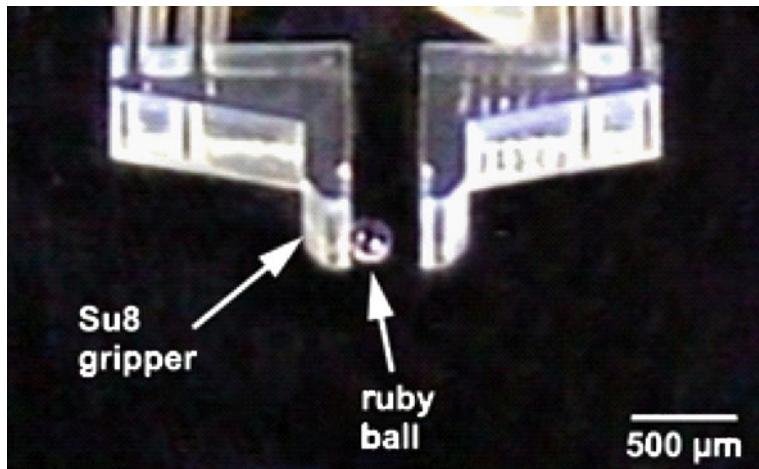
Products from air bags to cell phones achieve their functionality through the conglomeration of a diverse range of smaller components [1]. As the miniaturization trend pushes the devices to be smaller and smaller, technology advances are

---

A. Raatz (✉) · R. J. Ellwood

IWF: Institute of Machine Tools and Production Engineering, Technische Universität Braunschweig, Langer Kamp 19b 38106 Braunschweig, Germany  
e-mail: a.raatz@tu-bs.de

R. J. Ellwood  
e-mail: j.ellwood@tu-bs.de



**Fig. 16.1** A ruby ball sticking to the jaw of a microgripper [5]

requiring that these devices be able to do more. On the component side, the advances in semiconductors have led to an increase in functionality, a decrease in size, and all this with a decrease in price. For the resulting assembly task, this means that these smaller components need to be placed with greater accuracy within a smaller area. Although most of these parts fall into the category of surface mount devices (SMD), the ones that do not are the source of problems for microgrippers.

Within the assembly process, the SMD devices are to a certain degree standardized and offer a flat region on the top of the part which allows them to be handled with vacuum grippers [2]. As one diverges from this standardization, the parts take on a wide variety of shapes, come in many different sizes and can be made from a diverse range of materials. This spectrum can be seen in such things as microballs [3], microlenses, and in electronic components.

Additional problems arise as these parts get smaller and smaller, as the weight of the part is not longer more than the electrostatic, Van der Waals, and/or capillary forces. As illustrated in Fig. 16.1, this can cause parts to appear to stick to the gripper, which has resulted in these forces being referred to as stiction forces. In most cases, these forces are naturally occurring and are thus unpredictable. In the case of the ruby ball sticking to one of the jaws of a mechanical gripper, it is not possible to say how these forces will affect the ball when the jaws open. Assuming that the part can still be assembled, this will result in an increase in the assembly uncertainty. Within microassembly, this problem is well known and has been extensively described [4].

One way to overcome the effects resulting from the naturally occurring electrostatic forces is through the chosen assembly strategy. The three main assembly strategies are self-assembly, serial (sequential) assembly, and parallel (simultaneous) assembly [6]. It is also possible to join any of these to create a hybrid

assembly strategy. The best of all the assembly strategies is arguably self-assembly. Here, the parts find the correct position through things such as mechanical stops or through the minimization of capillary forces. During reflow soldering processes, it is possible that smaller parts will float in a pool of melted solder. In such a position, the floating parts will seek a minimal energy position. Dependent on the shape, it is possible to design the assembly so that the minimal energy configuration is the desired position [7]. This method has seen limited implementation as it is highly dependent on the part geometries and the used joining technology.

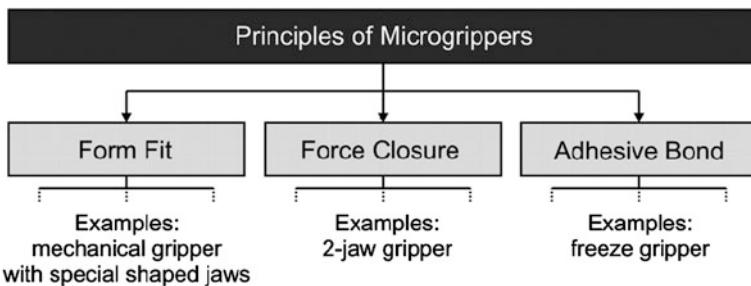
The most common assembly strategy is serial assembly, which requires each part to be handled separately. One of the main advantages of this strategy is that each part is individually justified to the substrate. As a consequence, the cycle times are relatively high when compared to other strategies. In comparison, a parallel strategy enables multiple parts to be assembled simultaneously. Such a strategy can be realized using parts which are not fully separated. An example is silicon wafer based parts which are not fully diced. Here, it is possible to handle all the parts on the wafer simultaneously. Following the joining process in which the wafer is handled, it is possible to separate the parts from one another. In addition to being able to place multiple parts at one time, this method increases the size of the handled part, making the parts easier to handle and preventing stiction forces from becoming a problem. Another parallel concept is a gripper which can handle more than one part at a time [3]. Although parallel strategies can be economical, many of the hard to handle parts as well as the product designs themselves prevent the industrial application of parallel assembly.

With the implementation of a serial strategy, it is common that the gripper is designed around a specific gripping task and part. With the diversity of parts which are handled, it is clear that there are many criteria which need to be considered when designing such grippers. The realization of such a gripper started with macrogripper considerations, and will inherently share some of the same design requirements. This includes such things as:

- Opening and closing actions have to happen in a timely manner.
- Robust/acceptable cycles to failure.
- A flexible or modular system is often favored.

There is a clear overlap between the requirement sets for macro and micro-grippers. With the smaller sizes of the microparts, different effects take on a dominant role, with some being unique. These include such things as:

- Fragile parts which often have fragile functional structures.
- Limited surfaces which are suitable for gripping.
- When applicable, fiducials on the part need to be visible.
- The gripping system cannot be a source of contamination as many of the microassembly tasks take place in a clean room environment.
- Due to the small weight of the objects in comparison with the surface forces, it is common for parts to “stick” to the gripper.



**Fig. 16.2** The three different categories of microgrippers with examples

- The gripper size should be proportional to the part.

With each part comes its own unique set of requirements, which has lead to many different microgrippers. Research has shown that the following concepts can be used to create a microgripper with similar concepts being grouped together:

- Mechanical.
- Vacuum.
- Electrostatic.
- Capillary.
- Freeze, cryogenic, or ice.
- Push–pull [8, 9].
- Magnetic.
- Bernoulli [10].
- Wave or Ultrasonic [11, 12].

The applicability of each gripper for a specific part is dependent on the requirement set for the given task. In most cases, there are grippers which are not applicable. For example, an electrostatic-sensitive device (ESD) cannot be handled by an electrostatic gripper. It is also clear that a vacuum gripper is not capable of gripping a porous part. Within the microgripper design process, it is important to consider the applicability of the grippers at the start of the design process.

Many different ways of organizing these grippers have been proposed [13–15]. These include the number of objects which can be gripped, access method, or the gripping principle [13]. Within the scope of this paper, the gripping principles form fit, force closure, and adhesive bond are used to classify microgrippers. These groups have been visualized in Fig. 16.2.

Microgrippers built on the basis of the form fit principle take advantage of a unique aspect of a part to allow it to be handled. An example of such a gripper is used to handle microballs [3]. Here, the gripper has triangle shaped depressions in which the microballs can take on a defined position, while the microballs are held in place using a vacuum. Although there are form fit elements found in other grippers, grippers based solely on this principle are very part specific. With restricted flexibility, the direct use of this method has seen limited implementation.

The most popular of the microgrippers fall into the force closure category. The simplest of these would be a gripper similar to tweezers. Here, two jaws are used to hold a part in place. Although the main concept of macro- and microgrippers is the same, there are many differences in these grippers, which are discussed in the following section.

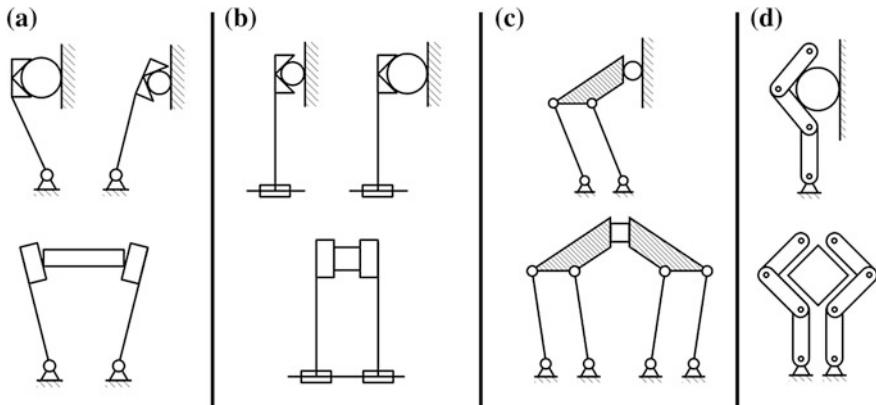
The last category takes advantage of molecular adhesion forces between the gripped particle and the gripper and fall into the category of adhesive bond grippers. Such grippers have found little application in the macro-world as the gripped object is often too heavy for the available gripping forces and can leave residue on the handled part [13]. This becomes less of a problem with microparts, as the gravitational forces reduce with part size. Viable microgrippers which fall into this category include capillary grippers and cryogenic grippers.

These three categories are used as a general classification for the different types of grippers. It is often the case that a gripper will take aspects from multiple categories, creating a hybrid design. In the following section, a review of the two main microgripping methods, mechanical grippers and vacuum grippers are presented. As these two concepts are the most prevalent microgrippers, they will be discussed in detail. In addition to these methods, microgrippers based on electrostatic, capillary, and cryogenic principles are presented. Although research is driving these concepts in the right direction, they have yet to reach widespread industrial application. Research has also been conducted on push-pull, magnetic, Bernoulli, and ultrasonic grippers. Although interesting, these grippers are beyond the scope of this paper. With an understanding of these grippers, a further look at their implementation within microassembly is explored. Here, one robot with different grippers is used to show how gripper design can have a large effect on the resulting assembly uncertainty within the microworld.

## 16.2 Mechanical Microgrippers

Very similar to macropart assembly, one of the most common tools for handling microparts is a mechanical gripper. Such a gripper is comparable to a pair of tweezers, where two jaws are used to clamp an object in place. As the macrogrippers become smaller and smaller, the resulting gripper and actuators are often disproportionately large in comparison to the task. This has led to the two main areas of research within this field, one being the kinematics of the gripper and the second being the actuation. Further research has been conducted with the goal of incorporating a force sensor into the gripper and will be briefly discussed. After discussing the kinematics and actuation of microgrippers, different examples of mechanical microgrippers are presented.

The manner in which a micropart is gripped is crucial to insuring that the part is properly held. During the design phase, the shape of the part in addition to the path the gripper jaw traverses, the gripper force, gripper jaw range, and the method of force transmission need to be considered [13]. Looking at the path the gripper jaw



**Fig. 16.3** Four ways of achieving a mechanical gripper: **a** rotational motion, **b** parallel motion, **c** parallel gripping resulting from linkages, **d** flexible gripper [14, 15]

traverses, there are four main concepts. As illustrated in Fig. 16.3 these path concepts are: a rotational motion, a parallel motion, a circular translational, and a flexible gripper. For each of these methods, there are single jaw methods (top) and double jaw methods (bottom) [14, 15]. The single jaw method is a simplified method in that only one of the jaws is moved and the object is pushed against a stationary surface. In comparison, the double jaw method where both of the jaws are actuated. Depending on the object to be gripped, a form fit element can be added to allow a centering effect. Here, the jaws have been modified to center round parts, such as ball bearings and is illustrated in the top pictures in Fig. 16.3.

Of these, the rotational motion is one of the easiest to obtain and is the typical path traversed by tweezers. Tweezers represent a special case in that both jaws rotate about a single fixed point. In comparison, many microgrippers rotate about two different points at the base of the tool. The locations of these points dictate how well a gripper can pick a specific part. As depicted in Fig. 16.3a a bottom, if these points are too close or too far from one another, the part does not sit flush to the gripper jaws. This can result in force peaks at the corner of the part, a force distribution which is asymmetrical, and if the angle is too large, can even force the part out of the grippers [15]. Grippers who use a parallel jaw motion are thus favored over rotational (Fig. 16.3a, b). A direct parallel motion (Fig. 16.3b) can be achieved with a direct linear actuation concept. Due to the poor miniaturizations potential as well as high costs of the actuators and guides, it has seen limited implementation. Through the integration of different linkage elements, such as a combination of four bar linkages, the parallel jaw motion can be obtained without linear guides or actuators (Fig. 16.3c). This is achieved through the integration of rotational elements which can allow for the substitution of conventional hinges with flexure hinges. Flexure hinges simplify the resulting construction as they allow for a monolithic design and can have an integrated restoring force. Last is the group of grippers which are inspired by the human hand, and fall into the

category of flexible grippers (Fig. 16.3d). Due to the complexity and size of these grippers, their implementation within microassembly has been limited.

With an idea of the desired motion and jaw path, additional information such as the allowable gripping force and jaw range enable the design process to begin. These design parameters are used within a kinematic synthesis to find acceptable gripper geometries. There are many different kinematic elements which can be used, with different kinematic considerations being discussed in [13–15].

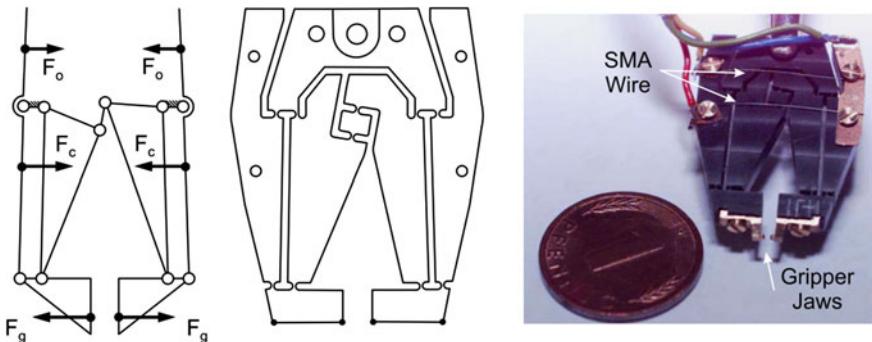
In comparison to the grippers, desired motions, and forces within microgrippers, the typical forms of actuation are often too large. Although some of the actuation forms such as pneumatic actuation carry over from macrogrippers, there are other innovative concepts stemming from smart materials which enable smaller designs. These methods are briefly discussed and are followed by real world applications.

Pneumatic actuation can be a favorable choice of driving microgrippers and often take the form of an air driven piston. This simple design is robust against changes in temperature, vibration, and dirt. Within microgripping, it allows a compact design within the gripper, as the source of the compressed air is often separate from the handling system.

There are many actuators which fall under the heading of smart materials, and of these shape memory alloy (SMA), piezoactuators, and thermal actuators are common within microgrippers. A smart material is a material which reacts in a controlled fashion to an external stimulus. SMA materials have a relatively large deformation which can be recovered when exposed to a change in stress or temperature [13]. Although it is possible to make them in a wide range of shapes, a simple wire design can be implemented, as seen in Fig. 16.4. These designs have the advantage that they are small and easy to control. On the down side, cycle times are influenced by the time an element takes to cool. Piezoelectric ceramics are another common smart material that changes size when a voltage is applied across it. These controllable displacements offer comparably large forces with the downside of small displacements. When these small displacements are considered within the kinematic synthesis, they are easy to overcome.

Another possible method to actuate microgrippers is thermal expansion actuation. Here, a material with a high coefficient of thermal expansion is either heated or cooled to cause motion. There are two main ways in which these elements are used as actuators. One is the direct implementation of the metal, and the second is through a bimetal. In Ref. [15], a metal with a high coefficient of thermal expansion is used to directly actuate a gripper. Similar to piezoactuators, kinematic considerations are used to obtain larger deformations. In comparison, a bimetal uses two metals with different thermal properties. As the metals are heated, the different expansion rates cause the metal or actuator to deflect toward the metal with the lower thermal expansion properties.

Hand and hand with the method of actuation is a method to measure the amount of force being applied to the grasped part. To see the gripping force during the handling of parts, Ref. [15] presented the integration of strain gauges into the jaws



**Fig. 16.4** Mechanical gripper proposed by Pittschellis, showing opening, closing, and gripping forces ( $F_o$ ,  $F_c$ ,  $F_g$ ) [13]

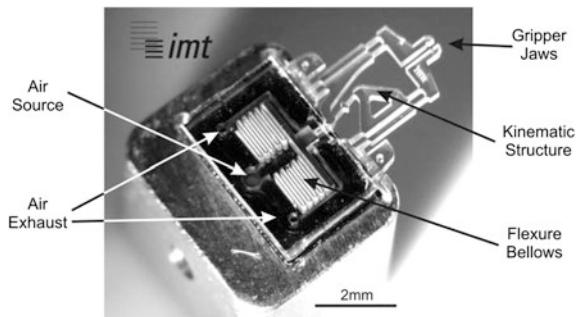
of the gripper. The results show that a force signal of  $1.1 \pm 0.4$  mV/mN can be obtained.

With all the parts put together, the mechanical gripper is one of the most flexible microgripper. Although there are many advantages to this kind of gripper, there are also clear downsides. One of the main issues with mechanical micro-grippers arises from the small height of the microparts. Drastically limiting the gripping area to the jaw contact area, this can lead to force peaks at the area of gripping. As these grippers are larger than the parts being grasped, there is often interference between the bottom of the gripper and the surface that the part is on. In order to overcome this, it is common to place the gripper at an angle to the part. The angle of the bottom plane of the gripper to the surface plane is referred to as the contact angle.

In addition to the contact angle, it is also important to look at the realization of such complex grippers. If one tries to shrink a macrogripper for such applications, many problems in the realization will arise. Not only are there many small parts, many of which would be very expensive if they could even be produced, one would then have to put them all together. Critical elements to realize are the joints, especially when one tries to integrate ball bearings. Fortunately, there is a very simple answer to this, flexure hinges. Flexure hinges are compliant mechanisms which can be realized by intentionally making a part thinner in one region. This thinner region allows for compliance, a region in which the part is intended to bend. The integration of flexure hinges allow for monolithic designs which are comparatively easy to produce. Examples of such grippers are illustrated in Figs. 16.4 and 16.5.

As illustrated in Fig. 16.1, the electrostatic forces found between microparts and the grippers can lead to the handled part sticking to the gripper. In order to limit these effects, it is possible to reduce the contact electrification. This can be achieved by minimizing gripping forces, creating the gripper jaw out of a hard and rough material, and by reducing the capillary forces through dry working conditions [4, 5].

**Fig. 16.5** A pneumatic microgripper made from SU-8 [15]



### 16.2.1 Mechanical Grippers in Practice

There are many ways that the presented components can be put together to create mechanical grippers. The majority of these grippers are the subject of research, with very few actually being commercially available. As it is not possible to mention all of them, an example for each of the main actuator principles, SMA, thermal, piezoelectric, and pneumatic are presented.

The first design which is considered uses two parallelograms which allow the jaws to move parallel to one another, as illustrated in Fig. 16.4. It is additionally shown how an opposing direction four bar linkage can be used to actuate both jaws simultaneously, allowing it to be driven with one actuator. As the SMA actuators have slow cycle times, two actuators working antagonistically are used to overcome this drawback. The locations of these actuators are visualized with  $F_O$  for the opening force and  $F_C$  for the closing force. Along with the kinematic diagram, Fig. 16.4 shows the resulting gripper [13].

Taking advantage of thermal actuation, the MEMS Precision Instruments Company has created a compliant structure which is driven by thermal expansion. Referred to as microtweezers, these grippers are etched from a single crystal silicon wafer with the smallest gripper being 4.5 mm long, 1 mm wide, and a thickness between 25 and 100  $\mu\text{m}$  [16]. With a wide variety of tip variations, they offer a stroke of 35  $\mu\text{m}$  [17].

Piezoelectronics can be used in a variety of different ways. At PiezoSystem Jena, a piezoactuator is used to drive the “grippy III”. With a 300  $\mu\text{m}$  opening, this gripper is designed for handling fiber optics. The gripping force is variable, and can be altered by changing an integrated spring. The S-805-00 gripper has a length of 54.5 mm, a width of 38 mm, and has a height of 7 mm [18].

Within research, Hoxhold [15] presented a piston driven pneumatic microgripper. The presented gripper is primarily made out of SU-8, an epoxy-based negative photoresist, with a detailed account of this gripper provided [15, 19, 20]. As can be seen in Fig. 16.5, the resulting hybrid structure is approximately 4  $\times$  7 mm. The labels in Fig. 16.5 are for the opening action, when the air source is compressed air. If this source is change to a vacuum, air will enter through the exhaust holes and the gripper will close. With a vacuum pressure of  $-0.8$  bar, a

force of 32 mN can be achieved. By switching to a pressure of 1.25 bar, it is possible to open the gripper to a width of 520  $\mu\text{m}$ .

This brief insight into the practical application represents only a fraction of the grippers which have been realized. The wide selection of design characteristics, from kinematic to actuation and grippers created for individual tasks, leads to an equally wide spectrum of mechanical grippers.

### 16.3 Vacuum Microgrippers

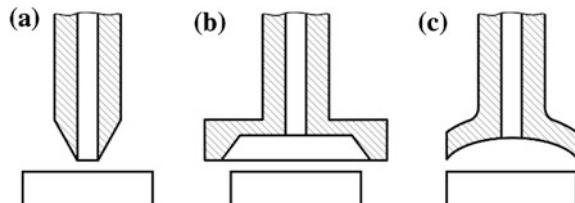
One of the few standardized parts within microassembly fall into the category of surface mount devices (SMD). These electronic components get their name from the fact that they are assembled by being placed on a surface such as circuit boards. Due to the volume of part which needs to be assembled on modern circuits, a standardization of these parts has taken place [2]. Among many of the standardized parameters, a flat surface on the top of the part is often left free. This allows the appropriate vacuum gripper to pick up the part while maintaining the flexibility of the used gripper. Although it is common to use a chip-shooter within larger SMD automation, a single chip handling process is considered here. With such a scenario, the robot first moves the gripper just above the part. A vacuum is then turned on and the part is sucked onto the bottom of the gripper. The robot then justifies the handled part to the substrate, places it, and turns off the vacuum to release the part. If the part sticks to the gripper, it is possible to invert the vacuum source and use air pressure to blow the part off the gripper. It is important to mention that this additional step can lead to an unwanted displacement of the part.

The widespread use of this method within SMD assembly has helped vacuum grippers reach a level of industrial maturity. There are many different designs for vacuum grippers from needles, die collets, and suction cups as illustrated in Fig. 16.6 [14]. Each has the same trait in that the vacuum is used in a force closure method to hold parts to the gripper. The resulting force exerted on the part can be controlled through the regulation of the vacuum. It is also possible to minimize the forces between the gripper and the part by using the squeeze film effect, which is important for delicate parts.

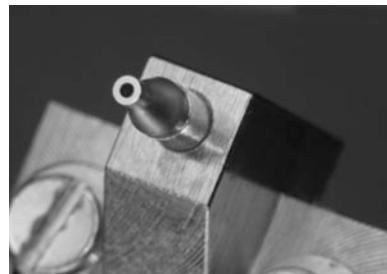
When a vacuum gripper is being designed, it can take on just about any shape, be made from a wide variety of materials, and come in a wide variety of sizes. These concepts share the basic design criterion that air leaks must be minimized and require a planar surface. Here the size of the needle has a lower limit which is material dependent and there is an optimal gripper size for a given part [21]. For plastic and rubber this is approximately 1 mm. For metallic and ceramic grippers, this is approximately 200  $\mu\text{m}$ . It was shown in [21] that borosilicate glass capillaries can be used to create needles with tip diameters between 25 and 50  $\mu\text{m}$ .

There are many features which make vacuum grippers very attractive. With a simple design it is possible to make a robust and compact microgripper. As most robots have compressed air, the vacuum is readily available. With vacuum as the

**Fig. 16.6** **a** Needle, **b** Die collet, and **c** Suction cup vacuum grippers [14]



**Fig. 16.7** A needle based vacuum microgripper



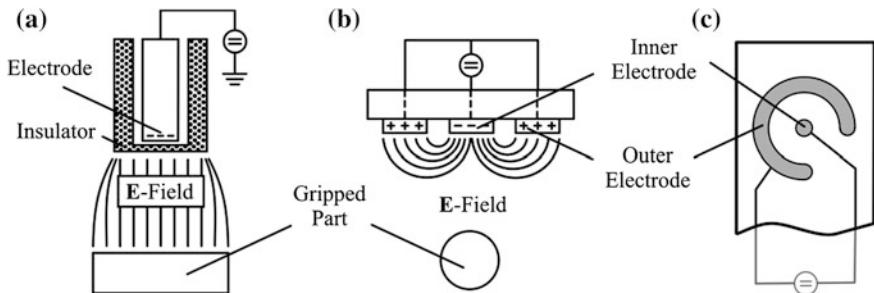
driving force, it is also easy to control such systems. When compared to mechanical grippers, vacuum grippers can be much smaller in size. This comes at a cost, as there is a minimal part size which can be successfully gripped. There are many different answers to what the smallest size may be and vary from  $200\text{ }\mu\text{m}$  [22],  $80\text{ }\mu\text{m}$  [21], to  $10\text{ }\mu\text{m}$  [1].

### 16.3.1 Practical Application of Vacuum Microgrippers

A needle microgripper is illustrated Fig. 16.7, and shows the simplicity with which such a robust gripper can be constructed. With the small size of the handled parts, the main variations of vacuum based microgrippers are within the size of the contact area and the vacuum opening of the resulting gripper. If the handled parts allow for a larger contact area, the stability of the part with the gripper during handling can be increased. Examples which show an extended gripper surface are shown in the section “Microgrippers and Process Influences on Microassembly”.

## 16.4 Electrostatic Microgrippers

One of the reoccurring problems within microassembly is that as the parts get smaller, their weight is no longer large enough to overcome the stiction forces acting on them. In the case of mechanical and vacuum microgrippers, this leads to the handled part sticking to the gripper. This can be turned from a problem to a



**Fig. 16.8** Electrostatic gripper concepts: **a** unipolar, **b** bipolar, and **c** a realized bipolar concept [5]

solution and is the basis of electrostatic microgrippers. Here, electrostatic forces are introduced or created, enabling the parts to be held during the assembly process. Although it is possible to use electrostatic actuators within mechanical grippers, this section looks at microgrippers that use electrostatic adhesion forces to attract and hold the part to the gripper.

Building on the original problem that the electrostatic forces between the part and the gripper are unpredictable, it is possible to induce a charge on the part [5]. Taking out the uncertainty about the existence of a charge, this method is able to pick up parts, but the problem of accurately releasing them still remains. A solution to this has been the integration of an active element into the gripper. This element is designed to generate an electric field, which can be eliminated to release the part [5, 23].

Within these electrostatic grippers, the part being picked has a large effect on the requirements of the gripper and the resulting holding force. The mobility of the electronic charges within a part plays a large role, with the parts being divided into conductors and insulators. When a conductor is handled, a homogeneous field can be used to handle the part [5] and can be created with a unipolar design. In contrast, a bipolar design will allow the creation of an inhomogeneous field which can be used to pick parts which are insulators.

In Fig. 16.8, both the unipolar and bipolar electrode concepts are shown. The main difference between these two designs can be seen in how the grippers are grounded. As the names implies, both the ground and source are located within a bipolar gripper. An advantage of the bipolar design shown in Fig. 16.8, is that it will also have a centering effect on the gripped parts.

Extending the work of Oh [23], Wrege discussed the realization of bipolar grippers to handle parts which are electronic insulators [5]. Here, gold was sputter deposited onto a Pyrex substrate. Wrege shows how the gripping force that a bipolar gripper exerts on the handled part is relative to the part distance from the gripper. With an operating voltage of 400 V, it was shown that a 400  $\mu\text{m}$  Ruby ball could be handled [24]. It is also shown that the reliability of electrostatic gripping is dependent on the humidity. In an example shown in [5], when the

relative humidity increases above 65 % Rh3 and using an operating voltage of 1.2 kV, parts larger than 400  $\mu\text{m}$  could no longer be gripped.

In comparison with mechanical and vacuum grippers, electrostatic grippers tend to have a lower contact pressure. Additionally, these grippers have comparably weak holding forces in the lateral direction. One of the main disadvantages of electrostatic grippers is the large operating voltages. In addition to an increased operating risk to the user, there is also a risk to the handled parts. This is primarily true for ESD, which can easily be damaged by such a gripper.

## 16.5 Capillary Microgrippers

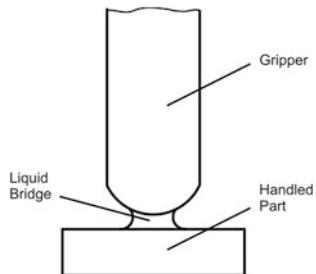
Research is pushing the boundaries of conventional microgrippers, and exploring new opportunities. With this in mind, microgrippers based on capillary and freezing effects have been the topic of research, but have yet to reach widespread industrial maturity. Although they have not reached this pinnacle, they are still viewed as viable options.

The ability of a fluid to appear to climb up thin tubes can be attributed to capillarity. Here, the intermolecular attractive forces between the fluid and the tube allow the fluid to seemingly defy gravity. At work here is a combination of the surface tension and the forces of adhesion between the liquid and the tube. Extending this to the microworld, it is possible to place a drop of liquid between the gripper and the part to be handled. When the surface tension forces which arise between the gripper and the part are large enough, it is possible to pick up the part [25].

Here, some of the critical design features are not only the chosen liquid, but the size, shape, and material of the gripper as well as the size, shape, and material of the handled part [26]. In Ref. [27], the capillary force is shown to be dependent on the distance between the handled part and gripper, the volume of the liquid, and the radius of curvature of the gripper. It is also discussed how the force increases as the radius of the gripper increases, and as the radius decreases so do the resulting gripping force. As the gripping force is dependent on the radius of the gripper, Biganzoli created a gripper which can actively alter this radius [27]. For the picking tasks, the radius could be increased and for the place task, decreased. A generalized capillary gripper is illustrated in Fig. 16.9. In addition to picking the parts, this radius also has a centering effect. Similar to self-assembling parts, the part seeks an orientation with minimal energy.

In Ref. [26], Lambert extensively shows the realization of a capillary gripper for the handling of watch ball bearings which have diameters of 300 and 500  $\mu\text{m}$ . The robustness of this gripper is shown in that it is able to hold the desired parts while the gripper is accelerated at 10 G, and can then be successfully placed. After being released, the unavoidable residual liquid (here water) which is left on the handled part evaporates within a few seconds.

**Fig. 16.9** Simplified capillary gripper



This microgripper shows that such capillary grippers are applicable to micro-assembly. Although this gripper works for the given case, it is very specific to the ball bearings for which it was designed. The complexity and introduction of a liquid onto the part do limit the industrial implementation of this gripper.

## 16.6 Freezing Microgrippers

One of the more unlikely types of microgrippers takes advantage of a phase change to create the desired gripping force. Similar to capillary grippers, an additional medium is introduced between the gripper and the part. The gripper then cools this medium below its freezing point, joining the part with the gripper. This effect can be reversed, in that heat is added to the system melting this medium and allowing the part to be released.

There has been extensive research done on the application of this concept [28–30]. The keys to these grippers are the introduction of a liquid to either the part or the gripper and the realization of the phase change. Due to the gripper being cooled, it is common to introduce the liquid to the part before the gripping process has started. In the above-mentioned research, it has been shown that a heat pump, such as a Peltier element can be used to cool the gripper. A heat pump has the advantage that it can be reversed, which in turn reverses the phase change and releases the part. In Ref. [30], they also propose the introduction of a Joule heater into the tip of the gripper. This heat source allows heat to be quickly introduced into the gripper tip allowing for faster cycle times. Due to increased tip temperatures, the evaporation time of the liquid is also accelerated. The addition of this heating element allowed for a gripping time of 1.4 s to be achieved. During these tests, this gripper has a reliability of 90 % and used a droplet volume of 0.1  $\mu\text{l}$ . Unlike vacuum and mechanical grippers, this cycle time is dependent on the part size, the thermal conductivity of the part, and the size of the liquid drop.

Although the complexity of phase change grippers cannot be overlooked, they do offer some clear advantages for handling microparts. Such grippers do not damage the gripping surface, can be used with porous parts, and provide comparably high holding forces. These holding forces are also very large in the lateral

direction and are independent of the shape and material of the part. In Ref. [28], it was stated that cryogenic grippers are 20–100 times stronger than vacuum grippers.

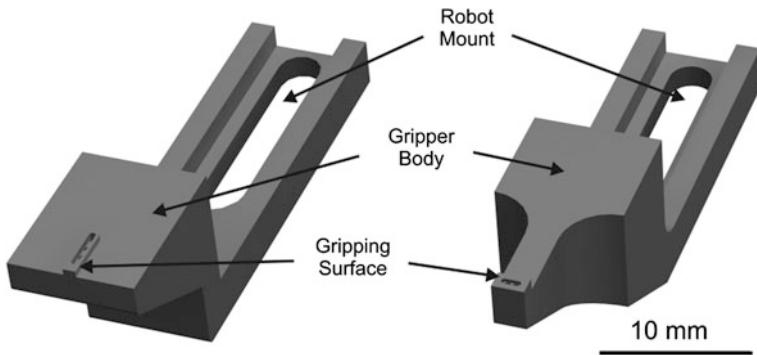
All of these advantages come at a price. Such grippers are often large in comparison with the other presented grippers. Phase change grippers also require the introduction of a fluid on the part before it is handled, creating an additional step in the handling process. As mentioned in Ref. [30], it is possible to leave impurities on the part from the chosen medium. If the handled parts are small enough, it is possible for capillary forces to arise [25].

## 16.7 The Influence of Microgrippers within Microassembly

It has been shown that a precise robot and a suitable microgripper do not inherently result in acceptable assembly uncertainties [31]. Within the microworld, the smallest effect can have a large impact on the resulting assembly performance. Within this section, three different microgrippers are used to set similar microparts on the same substrate. In order to allow the gripping and assembly concepts to be compared, the assembly robot and optical sensor are kept constant. To position the parts, the hybrid parallel robot with five degrees of freedom, the micabo<sup>f2</sup>, was used [31]. Developed at the Institute for Machine Tools and Production Technology (IWF) at the TU Braunschweig, it has been shown that this robot is capable of a repeatability of 0.6 µm. This repeatability was determined using the ISO standard 9283.

In order to gain information about the assembly scene, the integration of the camera directly above the gripper is favored. With the chosen assembly robot, this requires that the camera be placed within the head of the robot. To further increase the information about the assembly scene, a 3D vision system has been chosen. Developed at the Institute of Production Metrology (IPROM) at the TU-Braunschweig, this unique sensor takes advantage of photogrammetry and a unique prism and mirror design to achieve a sensor small enough to fit into the head of the robot. The repeatability of the position measurement with this sensor is about 0.5 µm in the horizontal plane and 0.8 µm in the vertical plane [32]. This sensor uses white circle fiducials to calculate the position of the parts.

The last link is how the robot determines where to place the micropart. There are two main strategies which can be realized, an absolute and a relative strategy. The difference between these strategies is that the relative strategy can see both the handled part and substrate at the same time while an absolute method relies on two separate measurements. When calculating the positioning vector, the absolute strategy needs to put both measurements together. As the relative strategy is able to see both parts at the same time, the positioning vector can be calculated with one measurement. Another strength in the relative strategy lies after the first correction. As both parts are still visible within the vision system, iterative fine corrections can be made based on information from the last iteration. In the



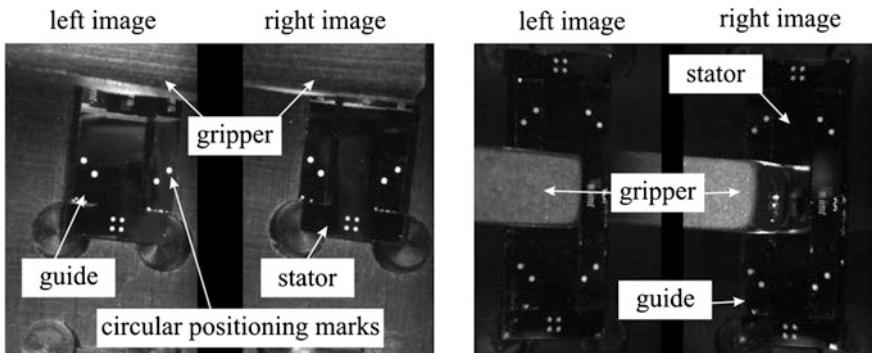
**Fig. 16.10** Two vacuum grippers for microassembly

following examples, the parts are iteratively corrected until the relative positioning vector is less than  $0.8 \mu\text{m}$ . In the following examples, silicon wafer based microparts are assembled. These flat parts are based on those of a linear microactuator which was the demonstrative product of the CRC 516 [33]. In the first two examples, the parts are joined using cyanacrylat and in the third a hot melt adhesive is used.

The first two grippers which are considered have a very similar design and are illustrated in Fig. 16.10. Some of the similarities include the long oval hole located at the back to fix the gripper to the robot. On the back side of these grippers is the vacuum supply, which is achieved with a standard M5 pneumatic connection. The main difference between the grippers is the part orientation to the gripper. With the left gripper, long and narrow parts are gripped so that the parts are parallel to the gripper, and within this paper referred to as the parallel design. Illustrated on the right of this figure is the perpendicular design, where the narrow parts can be gripped perpendicular to the gripper. This improvement of part orientation has come at the cost of a reduction in gripper surface area.

This slight change in gripper design allows for a better view of the assembly scene. The resulting sensor information can be seen in Fig. 16.11, with the parallel gripper on the left and the perpendicular design on the right. It is important to recall that a 3D camera system uses photogrammetry and thus there are two images for each gripper. In the original scene, four points on the substrate (stator) can be seen and two on the handled part (guide). In comparison to the gripper which holds the parts perpendicular, this is increased to four on the handled part and eight on the substrate. As these points are far from one another, there is a lever effect which allows for a better measurement [31, 34].

The experimental parts were set on the substrate using both designs. Using the parallel design, it was shown that parts could be assembled with an assembly uncertainty of  $36 \mu\text{m}$  [34]. As the repeatability of the robot and the tolerances of the measurement system are substantially less than this, insight into how the assembly process can be improved is sought. One of the primary observations was that only one side of the handled part was being observed. In addition, a little



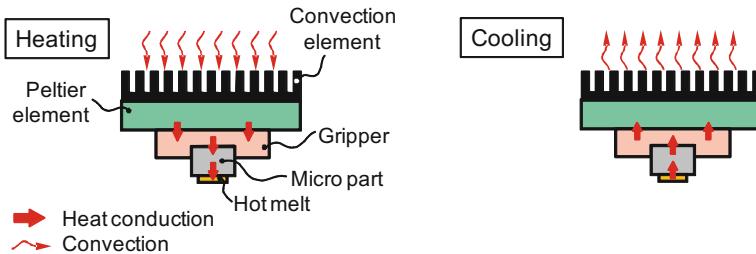
**Fig. 16.11** View from the integrated vision sensor of assembly scene with the parallel design (left) and the perpendicular design (right)

angular difference on the visible side of the part can lead to a large position difference on the other, not visible end of the part. Although the gripper was able to properly hold the part, it also blocked the sensor from seeing the other half of the part. If however, the parts are gripped differently, it is possible to see both sides of the part. With the redesigned gripper, parts are gripped in the middle and the guides are perpendicular to the gripper. The same parts were again set using the redesigned, perpendicular gripper. Due to limitations, only ten parts could be set using the redesigned, and it was measured that these parts had an assembly uncertainty of less than 10 µm [35].

Although the original gripper was able to hold the microparts, and thus met most of the challenges placed on it by the assembly task, its design limited the obtainable assembly uncertainty. Within microassembly, every last effect needs to be considered. The small change in part orientation resulted in a large improvement in the assembly uncertainty, but there was still a substantial difference between this and the capabilities of the assembly system.

To further improve on this, the joining process was taken into consideration. Upon inspection, it was viewed that moisture-curing, heat-curing, and radiation-curing glues often require setting times longer than 10 s [36]. If the glue is only partially set when the part is released, it is possible that the part will swim, tip, or move. In order to overcome this, a physical bonding adhesive such as a hot melt was seen as an alternative.

The integration of a hot melt adhesive within an assembly process offers shorter setting times, which in turn reduces the chances of a faulty placement. The integration of a hot melt adhesive is no trivial task though, as it cannot simply be applied between the parts. Such glues need to be heated to a specific temperature and are often printed onto one of the parts [37]. Two ways of heating up such adhesives were considered. It is possible to hold the part over a heater and then place the part, and the other option would be to create an active gripper which directly heats up the part. Both concepts have clear advantages and disadvantages and have been compared in [38, 39]. It is shown that for the small demonstration



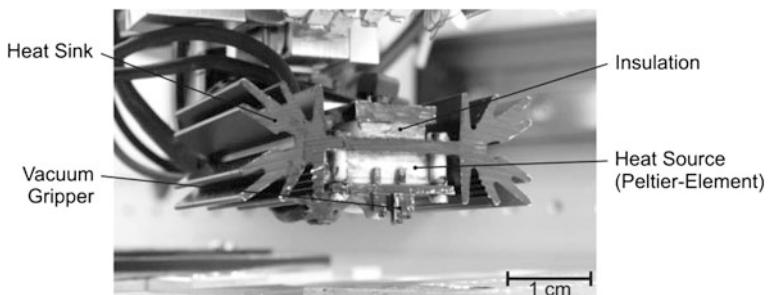
**Fig. 16.12** The heating and cooling phases of an active gripper [32]

parts, the active design can be optimized for better process times and results in a better assembly uncertainty. With these clear advantages, the realization of such an active design is briefly presented, with the active concept illustrated in Fig. 16.12.

What sets such an active gripper apart from other vacuum grippers is the integration of additional functionality, here the ability to heat up parts and the hot melt adhesive. Similar to the freezing grippers, a heat pump taking advantage of the Peltier effect is implemented. Again the Peltier element is advantageous in that when the voltage is reversed, it will go from heating to cooling (Fig. 16.12). The resulting active gripper for hot melt adhesives can be seen in Fig. 16.13. In order to improve the performance of this gripper, an additional heat sink has been integrated. At the cost of increasing the size of the gripper, this heat sink drastically increases the available surface area of the gripper.

Microparts similar to those set with the two previous microgrippers were again assembled. One of the main differences in these microparts is that they have been printed with hot melt adhesive [37]. This gripper also allowed the parts to be held perpendicular to the gripper. As presented in Ref. [39], the assembly uncertainty of  $7.57 \mu\text{m}$  was obtained with this new gripper.

As all three grippers handled similar parts and were assembled using the same robot with integrated 3D vision system, a comparison of the grippers is possible. It was first shown that the part orientation within assembly scene can impact the assembly uncertainty. This first correction allowed an improvement of the



**Fig. 16.13** Vacuum gripper with integrated heat pump, designed for hot melt adhesives [36]

assembly uncertainty from 36  $\mu\text{m}$  to less than 10  $\mu\text{m}$ . Extending this concept is an active gripper which takes advantage of hot melt adhesives to further reduce the assembly uncertainty to 7.57  $\mu\text{m}$ . Although this is still a large step in the right direction, there is still a large difference between this optimized assembly uncertainty and the robot's position accuracy of 0.8  $\mu\text{m}$ .

## 16.8 Conclusion

The desire to make hybrid assemblies smaller while increasing their functionality presents many technological challenges. With the decreasing size of the electrical and mechanical components come smaller and smaller assembly tolerances. Although the precision of the assembly robots is better than these tolerances, the link between the part and robot, as well as the joining technology play a large role in the resulting assembly uncertainty. Within this chapter, an overview of different microgrippers is presented. This is followed by the use of three different microgrippers which are used within an assembly task. The resulting assembly uncertainty for each gripper is then determined, showing the applicability of the design considerations.

The overview of microgrippers starts with the design requirements and design considerations for microgripping. This leads to a brief introduction into the diverse spectrum of microgripping concepts. Of these the two most industrial prevalent, mechanical and vacuum microgrippers, are discussed in detail. For mechanical microgrippers, it is shown that kinematic considerations enable the jaws to traverse a desired path. In addition to different methods of actuating these grippers, several industrially available grippers are discussed. The basis of the vacuum microgrippers is less complex, and a brief discussion of the nozzle and the contact surface considerations are presented. Following this, grippers based on capillary, freezing, and electrostatic principles are presented. Although grippers based on these concepts have seen limited industrial application, they do have potential for the future.

The combination of a precise robot, a functional microgripper, and a good machine vision system does not inherently result in an acceptable assembly uncertainty. In order to compare the influence of microgrippers design, a single demonstrative assembly task is taken into consideration. Three different microgrippers are then used with the same robot, measurement system, assembly task, and vacuum based gripping concept to complete a task. The iterative design process of these microgrippers is presented along with the resulting assembly uncertainty. It is first shown that the gripper can be improved through a design which takes the vision system into consideration. Further improvement can be made using an active gripper design, coupled with the joining technology. Through these improvements, it was shown that the assembly uncertainty of 36  $\mu\text{m}$  could be reduced to 7.57  $\mu\text{m}$ . It is clear that the assembly uncertainty is approaching the robot's positioning accuracy of 0.8  $\mu\text{m}$ . In order to overcome the remaining difference, future research will continue to look at the assembly and joining

processes. Such processes must offer the required precision, accuracy, flexibility, and be easily integrated into existing systems. This will in turn continue to push the boundaries and requirements of microgrippers.

## References

1. Hesselbach J (2002) MikroPRO: Untersuchung zum internationalen Stand der Mikroproduktionstechnik. Vulkan-Verl, Essen
2. Wormuth D, Zapf J (2001) Surface mount technology basics. Siemens DEMATIC AG, Munich
3. Brecher C, Weinzierl M (2008) Parallel precision alignment of multiple micro components. *Microsyst Technol* 14(12):1847–1853
4. Fearing RS (1995) Survey of sticking effects for micro parts handling. In: IEEE/RSJ international conference on intelligent robots and systems, vol 2, p. 2212
5. Wrege J (2007) Elektrostatisch unterstützte Handhabungstechniken in der Mikromontage. Vulkan-Verlag, Essen
6. Pokar G (2004) Untersuchung zum Einsatz von ebenen Parallelrobotern in der Mikromontage. Vulkan-Verlag, Essen
7. Srinivasan U, Liepmann D, Howe R (2001) Microstructure to substrate self-assembly using capillary forces. *J Microelectromech Syst* 10(1):17–24
8. Haliyo DS, Dionnet F, Regnier S (2006) Controlled rolling of micro objects for autonomous manipulation. *J Micromechatron* 3(2):75–101
9. Xie H, Régnier S (2009) Three-dimensional automated micromanipulation using a nanotip gripper with multi-feedback. *J Micromech Microeng* 19(7):75009
10. Grutzeck H, Kiesewetter L (2002) Downscaling of grippers for micro assembly. *Microsyst Technol* 8(1):27–31
11. Reinhart G, Hoeppner J (2000) Non-contact handling using high-intensity ultrasonics. *CIRP Ann Manuf Technol* 49(1):5–8
12. Reinhart G, Kirchmeier T (2011) Fuzzy logic based ultrasonic gripper design for handling small parts. In: 2011 IEEE international symposium on assembly and manufacturing (ISAM): IEEE, pp 1–6
13. Pittschellis R (1998) Mechanische Miniaturgreifer mit Formgedächtnisantrieb. VDI-Verlag, Düsseldorf
14. Monkman GJ, Hesse S, Steinmann R, Schunk H (2007) Robot grippers. Wiley-VCH, Weinheim
15. Hoxhold B (2010) Mikrogreifer und aktive Mikromontagehilfsmittel mit integrierten Antrieben. Shaker, Aachen
16. MEMS Precision Instruments (2011) Micro tweezer specifications. <http://www.memspi.com/>. Accessed Oct 2011
17. Agnus J, Nectoux P, Chaillet N (2005) Overview of microgrippers and design of a micromanipulation station based on a MMOC microgripper. In: 2005 international symposium on computational intelligence in robotics and automation: IEEE, pp 117–123
18. Piezosystem Jena (2011) Piezogripper grippy III fibergrripper: data sheet
19. Seidemann V, Bütfisch S, Büttgenbach S (2002) Fabrication and investigation of in-plane compliant SU8 structures for MEMS and their application to micro valves and micro grippers. *Sens Actuators A Phys* 97–98(1–2):457–461
20. Hoxhold B, Büttgenbach S (2008) Pneumatic actuators for micro grippers and active assembling devices. In: Borgmann H (ed) Conference proceedings/actuator 08, Bremen, Germany. HVG, Division Messe Bremen, Bremen, pp 766–769

21. Zesch W, Brunner M, Weber A (1997) Vacuum tool for handling microobjects with a nanorobot. In: Proceedings of international conference on robotics and automation: IEEE, pp 1761–1766
22. Bütfisch S (2003) Entwicklung von Greifern für die automatisierte Montage hybrider Mikrosysteme. Shaker, Aachen
23. Oh H-S (1998) Elektrostatische Greifer für die Mikromontage. VDI-Verlag, Düsseldorf
24. Hesselbach J, Wrege J, Raatz A (2007) Micro handling devices supported by electrostatic forces. CIRP Ann Manuf Technol 56(1):45–48
25. Mastrangelo C, Hsu C (1993) Mechanical stability and adhesion of microstructures under capillary forces. I. Basic theory. J Microelectromech Syst 2(1):33–43
26. Lambert P, Delchambre A (2005) Design rules for a capillary gripper in microassembly. In: Assembly and task planning: from nano to macro assembly and manufacturing, 2005 (ISATP 2005), pp 67–73
27. Biganzoli F, Fassi I, Pagano C (2005) Development of a gripping system based on capillary force. In (ISATP 2005). The 6th IEEE international symposium on assembly and task planning: from nano to macro assembly and manufacturing, 2005: IEEE, pp 36–40
28. Stephan J, Seliger G (1999) Handling with ice—the cryo-gripper, a new approach. Assembl Autom 19(4):332–337
29. Kochan A (1997) European project develops “ice” gripper for micro-sized components. Assembl Autom 17(2):114–115
30. Lang D, Tichem M, Warner F (2007) An Industrial prototype of a liquid solidification based micro-gripping system. In: 2007 IEEE international symposium on assembly and manufacturing: IEEE, pp 227–232
31. Schöttler K, Raatz A, Hesseibach J (2008) Precision assembly of active microsystems with a size-adapted assembly system. In: Ratchev S, Koolemeijer S (eds) IFIP—international federation for information processing. Springer, Boston, pp 199–206
32. Keck C, Berndt M, Tutsch R (2011) Stereophotogrammetry in microassembly. In: Büttgenbach S, Burisch A, Hesselbach J (eds) Microtechnology and MEMS. Springer, Berlin, pp 309–326
33. Büttgenbach S, Burisch A, Hesselbach J (eds) (2011) Microtechnology and MEMS. Springer, Berlin
34. Schöttler K (2008) Planung und Untersuchung automatisierter Mikromontageprozesse unter besonderer Berücksichtigung der Einflussgrößen. Vulkan-Verlag, Essen
35. Ellwood RJ, Raatz A, Hesselbach J (2010) Vision and force sensing to decrease assembly uncertainty. In: Ratchev S (ed) IFIP advances in information and communication technology. Springer, Berlin, pp 123–130
36. Rathmann S, Ellwood J, Raatz A, Hesselbach J (2011) Design of a microassembly process based on hot melt adhesives. In: Büttgenbach S, Burisch A, Hesselbach J (eds) Microtechnology and MEMS. Springer, Berlin, pp 345–364
37. Hemken G, Böhm S, Dilger K (2011) Use of hot melt adhesives for the assembly of microsystems. In: Büttgenbach S, Burisch A, Hesselbach J (eds) Microtechnology and MEMS. Springer, Berlin, pp 327–343
38. Rathmann S, Raatz A, Hesselbach J (2010) Active gripper for hot melt joining of micro components. In: Ratchev S (ed) IFIP advances in information and communication technology. Springer, Berlin, pp 191–198
39. Rathmann S (2012) Prozessfuehrung Automatisierter Klebprozesse in der Mikromontage (Preprint). Vulkan-Verlag, Essen

# Chapter 17

## A New Way of Grasping: PARAGRIP—The Fusion of Gripper and Robot

**Tom Mannheim, Martin Riedel, Mathias Hüsing  
and Burkhard Corves**

**Abstract** In this chapter a novel flexible and versatile handling concept, called PARAGRIP (Parallel Gripping), will be presented. This concept is based on a reconfigurable architecture with a modular and alterable layout. The robot system is able to handle objects with six degrees of freedom (DOF) by forming a parallel kinematic structure including several robotic arms and integrating the object as a movable platform. As many kinematic parameters, like the grasp and base points of the arms as well as the arm combination can be chosen freely, the handling system offers a fast and easy possibility to adapt the system to the requirements of the task. This adaptation can proceed before or even during manipulation. The latter is realized by regrasping, if more than three arms are available in the layout. This chapter deals with the questions, which reconfiguration possibilities are offered by this handling system and how a configuration set can be planned automatically.

### 17.1 Introduction

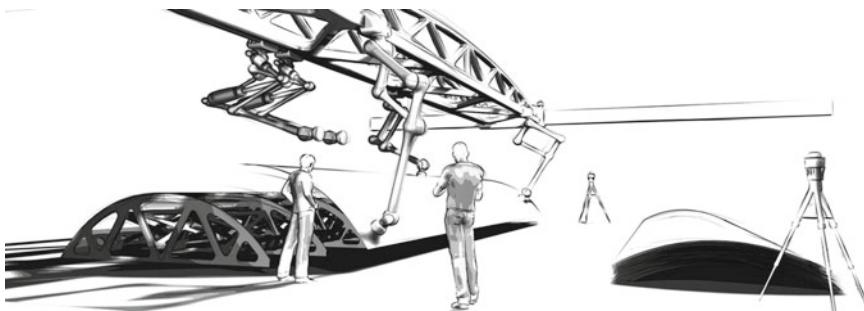
In recent years, manufacturing industry is governed by fundamental changes with regard to the conditions, like progressive globalization and rapid technological development as well as changes in the resources situation [1, 2]. Classical demands

---

T. Mannheim · M. Riedel · M. Hüsing · B. Corves (✉)

IGM: Department of Mechanism Theory and Dynamics of Machine,  
RWTH Aachen University, Eifelschornsteinstraße 18 52062 Aachen, Germany  
e-mail: mbox@igm.rwth-aachen.de

T. Mannheim  
e-mail: mannheim@igm.rwth-aachen.de



**Fig. 17.1** Design concept of the robot system PARAGRIP

on handling systems are currently undergoing changes. In the past, higher load capacity, greater precision, and higher speeds were demanded. However, priorities are increasingly shifting toward customized production and flexible solutions to component-dependent problems. Often currently available handling systems cannot fulfill the increased demands of complex and frequently changed assembly tasks [3].

A classic example of challenging manipulation tasks is the field of important applications dealing with the assembly of large components, such as in airplanes, ships, and wind turbines in small series. In order to automatically assemble those products, a very versatile and flexible handling concept is needed. Even with a large and customized gripper a common industrial robot is not an appropriate approach to fulfill the demands. The gripper has to either be customized or replaced depending on the parameters of the object, which often means uneconomic effort and inefficiency in small series. A better approach is to use co-operating robots to ensure that the object can be handled independent of size and shape [4]. A major disadvantage is the high overall number of actuators, e.g., 18 actuators for 3 robots to perform 6-DOF object motion. This causes complex control architecture and leads to high costs. Furthermore, rearranging industrial robots requires a long set up time, as robots are not designed to be mobile and need to be calibrated carefully. Finding a layout for the desired task is challenging as there is no software tool for planning reconfigurable robots available on the market. With the focus on combining mobility, flexibility, and versatility in one system a novel handling concept, called PARAGRIP (Parallel Gripping), has been developed, a design concept sketch is presented in Fig. 17.1. Its idea of manipulating objects leads to a fusion of a parallel robot and an adjustable gripper.

The system consists of several arms, featuring a mobility number of six, each. The arms can be actuated independent of each other. Contact elements located distally at the arms, e.g., electro magnets or vacuum cups, provide a temporary fixed coupling of arm and object, which allows to transfer forces and moments in all directions. After grasping, the object becomes part of the kinematic structure and can be guided like the moving platform in a closed loop manipulator, similar to the architecture of parallel robots, presented in [5, 6]. Thus, the system requires

an overall number of six actuators to perform a 6-DOF object motion. Additional drives are necessary for the grasp process. During object motion they can either be disconnected, driven passively, or support the manipulation. After finishing the manipulation the contact forces are switched off and the object is no longer integrated into the kinematic structure. As the contact element is not customized, the manipulation of objects with different shapes and sizes is possible without any physical reconstruction. This leads to a very flexible, versatile, and cost-efficient handling concept.

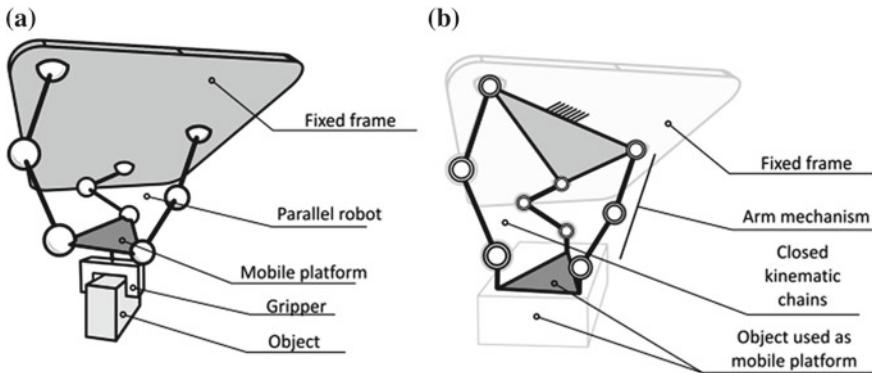
The principle of PARAGRIP offers to reconfigure the kinematic-chains before or during object movement. Reconfigurability has essential influence on the robot performance and capacity, for example, drive torque, drive velocity and positioning accuracy, as well as on the translational and rotational workspace. If the handling system consists of four or more arms, changing the configuration while moving is possible, which extends the complete workspace. Hence, it is possible to rotate large angles continuously about arbitrary axes.

Grasping, which includes the planning of the optimal grasp points depending on the motion, the computation of the grasp motion, as well as the physical grasp process itself with an underactuated robot arm, is a very important aspect of the complete robot development, computation and control.

In this chapter the novel handling system, called PARAGRIP, will be presented. After a short insight into the handling concept and the kinematic structure, the special grasp process with an underactuated arm will be investigated. Afterwards the reconfiguration possibilities and their effects on the robot performance are discussed. Here the focus is set on the configuration planning and the grasp-point optimization. Additionally, the capabilities, the challenges, and the optimizations of changing the configuration dynamically during motion, e.g., by regrasping, are presented. Several practical examples will underline the enhanced performance of a handling system where robot and gripper are combined into one unit.

## 17.2 Handling Concept

The key idea of the handling concept is to generate closed-loop kinematic chains formed by the object and the arms, similar to the architecture of a parallel robot, illustrated in Fig. 17.2. Manipulating an object with three arms in a 6-D Cartesian space requires two main actuators per arm and six in total. Compared to co-operating robots the overall number of drives can be reduced when the arms are meant to work together. This drive configuration leads to an underactuated single arm design with less drives than required to meet the mobility number of the arm. Hence, the combination of the arms to manipulate objects is the fundamental of this handling principle.

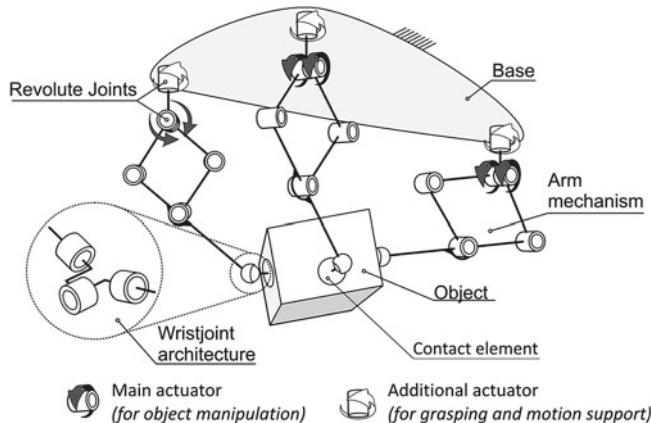


**Fig. 17.2** **a** classical parallel robot structure, **b** idea of new handling system

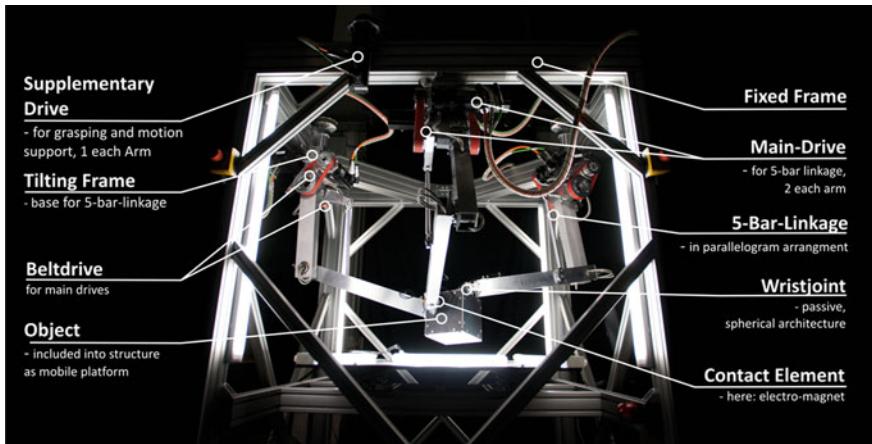
With the idea of several arms working together in a closed kinematic chain, the robot system provides the same kinetostatic performance as a parallel manipulator. Its capability is defined by the parameters of the closed loop structure, generated by the combination of arms. Hence, the single arm can be developed specifically to feature a more lightweight and mobile design in comparison to a regular industrial robot. Several kinematic parameters, e.g., dimensions of the moving platform, can be changed every time the system is grasping or regrasping an object. A temporarily jointed connection between the arms and the object can transfer forces in all directions and integrates the object completely into the robot structure. In comparison to mechanical handling systems imitating hands [7–9] the object is not clamped or clutched to handle. The possibility to locate the contact points on the surface of the handling object automatically eliminates the need for a customized gripper for each object and task. Hence, this approach enables the system to manipulate objects of different shapes and sizes without any mechanical changes.

### 17.3 Robot Architecture

The robot system is schematically represented in Fig. 17.3. It consists of three active arms, which are connected to the base by revolute joints. The object is attached to the arm mechanism via wrist joints having 3-DOF. Each arm mechanism is a five-bar linkage. The drive to rotate the pivoted frame is fixed to the base. At the end of the five-bar linkage a wrist joint is attached to the link. It features 3-DOF in rotation and a general contact element. As the axes of the wrist joints are intersecting in one center of rotation, the joint has a spherical architecture. There are two configurations for the wrist: in the first all joints are passive and in the second one axis is driven by a low-cost stepper motor. In both cases the wrist joint is under actuated, this means all passive joints have to be

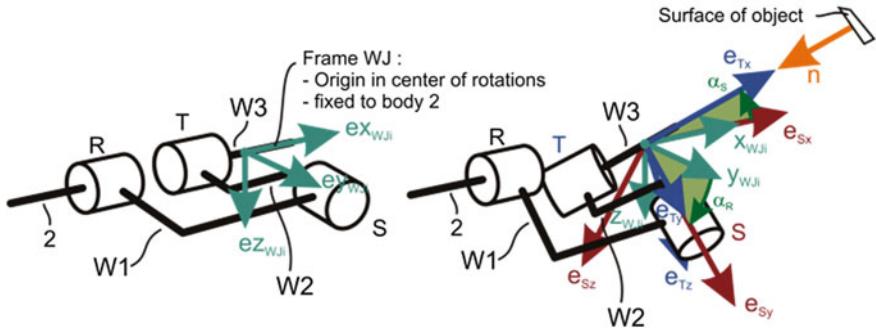


**Fig. 17.3** Concept sketch of the robot



**Fig. 17.4** Picture of the prototype

spring centered to avoid grasp failures or collision as a result of undefined joint positions. To demonstrate the feasibility of this concept a prototype was built, which is shown in Fig. 17.4. This prototype features a layout where the arms are mounted on a high-level base and hang down to manipulate the object. Wall mounted or floor orientated locations of the arms as well as a mixture of these configurations is also possible. The two cranks of the five-bar linkage in the arm are actuated by a servo drive via a belt drive and a planetary gearbox. The contact element of the wrist joint is an electro magnet in the present configuration of the arm, which offers to manipulate metal objects. For simplicity of exposition, the



**Fig. 17.5** Wrist joint structure with frames and denotations *Left* center joint position, *Right* general joint position

supplementary resting arm is not represented. The structural and dimensional syntheses are presented in detail in [10] and [11].

### 17.3.1 Grasp Process

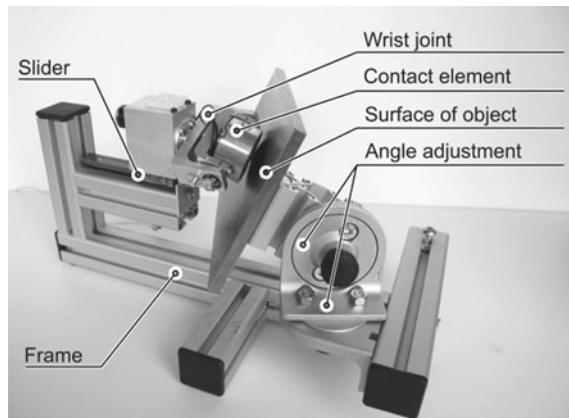
The grasp process will be presented in the following section. Because of its underactuated arm, this handling device needs a special investigation of the grasp behavior and a special grasp path to enhance the grasp capability.

One of the main benefits of the novel handling concept is the cost-effective build-up due to the reduced number of actuators. As shown in the previous chapter, the 3-DOF wrist joint can be designed free of drives. Hence, during the grasping phase this underactuated structure needs to be driven by the external contact forces to ensure that the contact elements align successfully to the surface of the object, similar to movement of the phalanx at a grasp process with an underactuated multi-finger gripper, like investigated in [12, 13]. The passive alignment here depends on several factors like the joint angles, friction conditions, grasp motion, and so on.

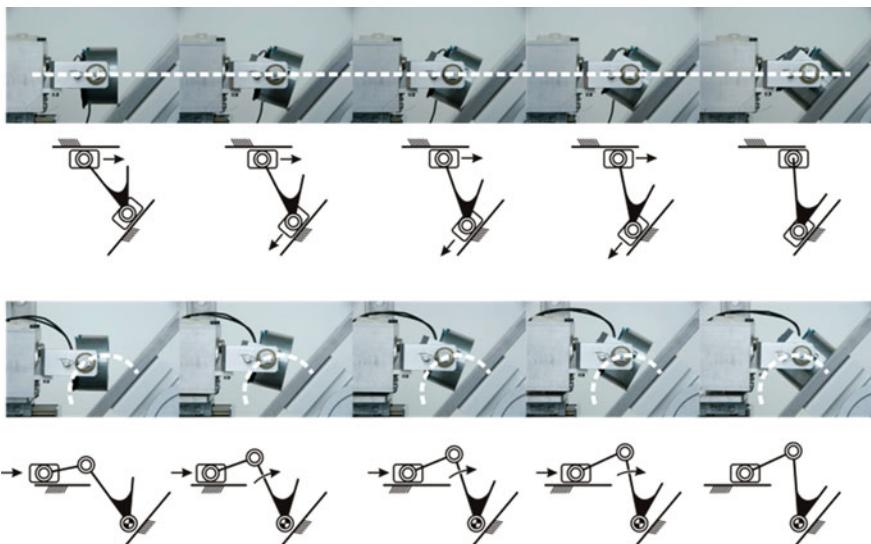
To calculate the necessary joint angles  $\alpha_R$  and  $\alpha_S$  for grasping, the orientations of the systems of coordinates S and T need to be determined, see Fig. 17.5. System S is attached to link W1, T to W3. Unit vector  $e_{S_y}$  is collinear to joint axis S, vector  $e_{T_x}$  is in the direction of joint axis T. Vector  $n$  is normal to the surface. The vectors can be computed as follows:

$$\begin{aligned} e_{S_x} &= e_{W_{Jx}}, \quad e_{S_y} = \frac{n \times e_{W_{Jx}}}{\|n \times e_{W_{Jx}}\|}, \quad e_{S_z} = e_{S_x} \times e_{S_y} \\ e_{T_x} &= -n, \quad e_{T_y} = e_{S_y}, \quad e_{T_z} = e_{T_x} \times e_{T_y} \end{aligned} \quad (17.1)$$

The joint angle  $\alpha_r$  is the angle between the vectors  $e_{W_{Jy}}$  and  $e_{S_y}$  in the direction of vector  $e_{W_{Jx}}$ ,  $\alpha_S$  between  $e_{S_x}$  and  $e_{T_x}$  in the direction of  $e_{S_y}$ .



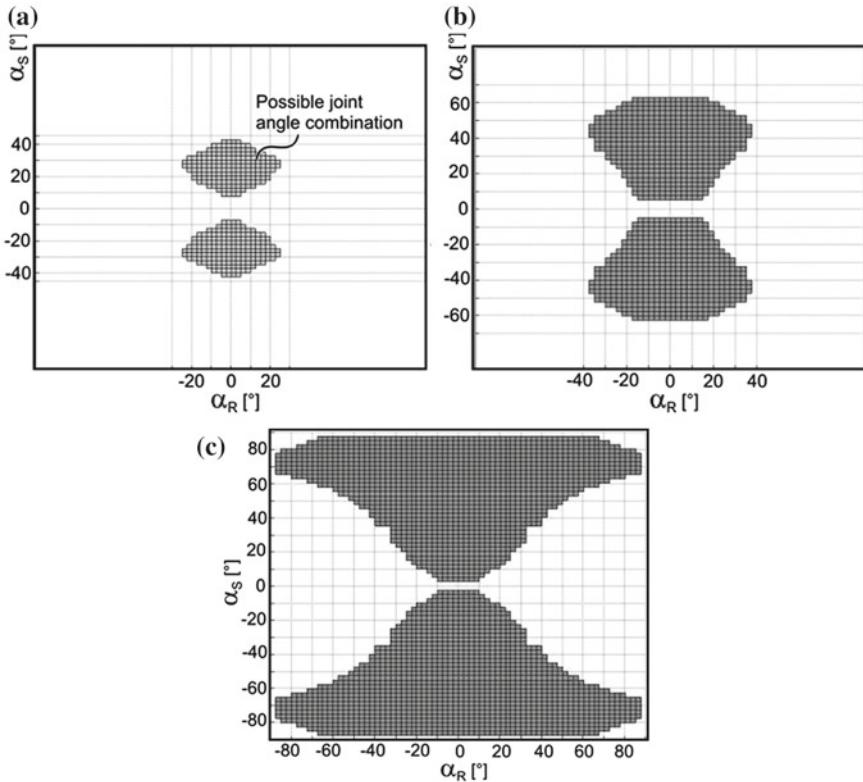
**Fig. 17.6** Wrist joint test bench



**Fig. 17.7** Linear and circular grasp motion in comparison

To investigate the grasp behavior a test bench shown in Fig. 17.6 has been developed. The wrist joint prototype is located on a slider and can be moved toward a test surface. This surface is adjustable in every combination of the angles  $\alpha_R$  and  $\alpha_S$ . The test bench can be configured in many variations to test different grasp influences, like grasp paths and friction conditions. Figure 17.7 shows a grasp sequence, where the contact element aligns passively with the test surface by rotating about the angles  $\alpha_R$  and  $\alpha_S$ .

A change in the configuration of the test bench can simulate the guidance of the joint on different paths. Figure 17.7 upper part shows, that for linear grasp motion,



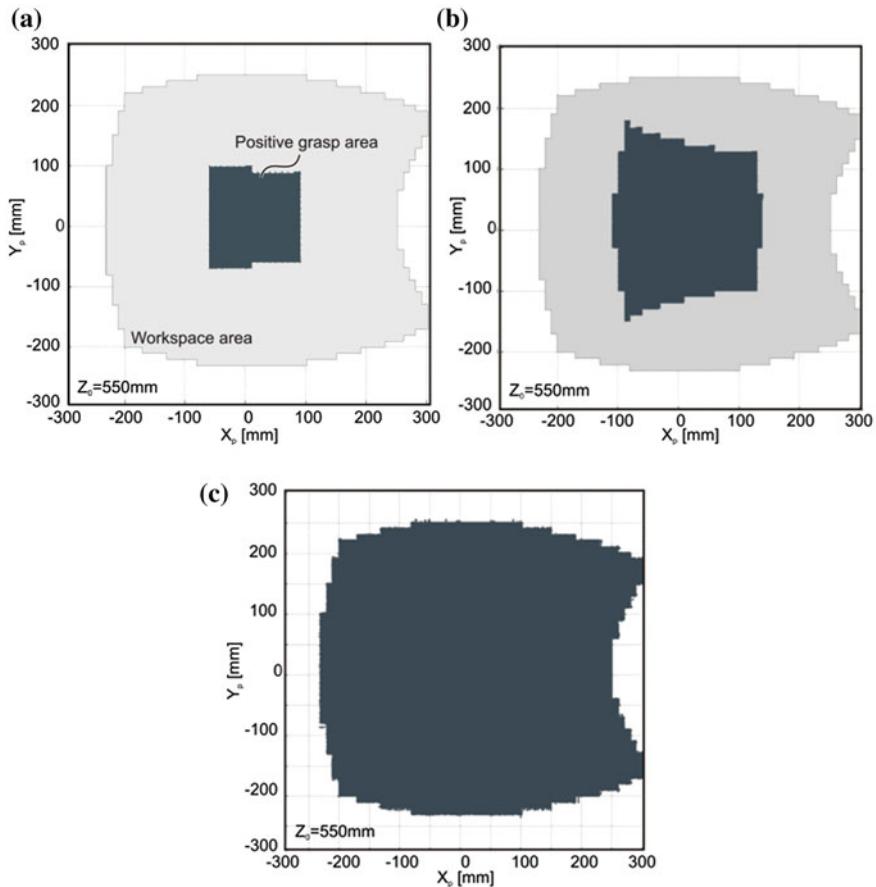
**Fig. 17.8** Successful passive alignment of the contact element for **a** linear motion ( $\mu = 0.2$ ), **b** circular motion ( $\mu = 0.2$ ) and **c** circular motion and contact modification ( $\mu = 0.6$ )

when the center of the wrist joint is guided on a straight line, the contact point needs to slide on the surface, comparable with a double-slider-mechanism. At circular motion, the contact element rolls on the fixed contact point by moving the center of the wrist joint on a radius like a reversed slider-crank-mechanism, see Fig. 17.7 lower part.

The test results show that there is a significant difference in the passive alignment of the contact element for these grasp paths. It can be shown, that independent from friction conditions the circular grasp motion is always more successful. An additional enhancement can be achieved by modifying the contact conditions using supplementary friction elements on the edge of the contact element, like rubber rings.

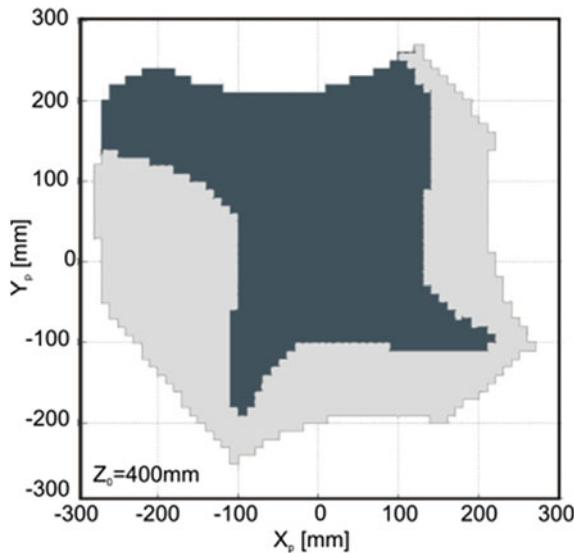
Figure 17.8 shows the area of a successful passive alignment in the space of the wrist joint angles  $\alpha_R$  and  $\alpha_S$  for friction coefficients  $\mu = 0.2$  and  $\mu = 0.6$  at contact point and different grasp motions.

For theoretical investigations of the forces occurring during grasp process, see [14]. More interesting than investigations of positive grasp behavior in joint

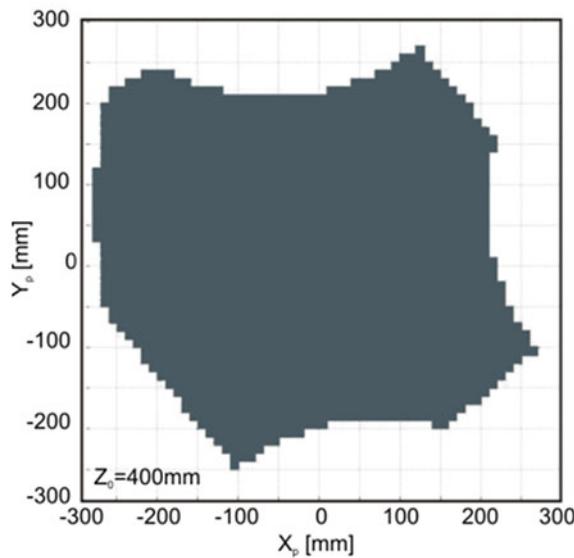


**Fig. 17.9** Positive grasp and workspace area for linear motion (a) circular motion (b) and circular motion with contact element modifications (c)

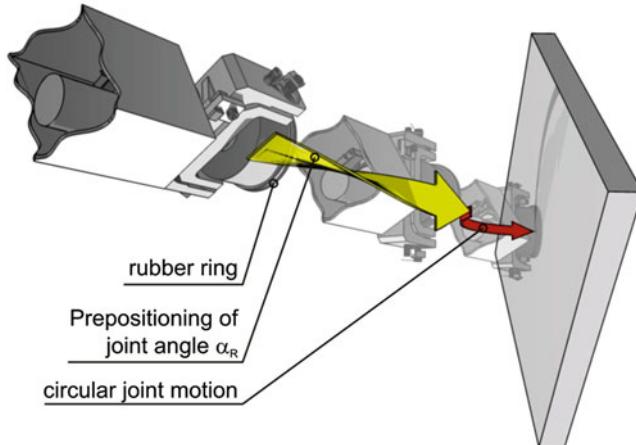
space is a consideration in Cartesian space. Figure 17.9 shows the workspace for object manipulation in a single layer (light gray part) and the area (dark gray) where a cube with an edge length of 150 mm can be grasped. Figure 17.9 (a) shows the result for linear motion, (b) for circular motion and (c) for a circular motion as well but additionally with a modified contact element. In the last case the higher friction causes a very good grasp behavior, so that the object can be picked up all over the workspace. Investigations with different object orientations on the ground show that even with a circular grasp motion and an enhanced contact element, some object poses at the edges of the workspace area cannot be grasped because of an improper combination of the joint angles  $\alpha_R$  and  $\alpha_S$ , see Fig. 17.10. This drawback can be avoided by using the above-mentioned wrist joint configuration II with an actuator in joint axis  $R$ . This actuator is a very small, low cost stepper drive, which controls actively the position of the angle  $\alpha_R$  before



**Fig. 17.10** Grasping behavior for object orientation  $\varphi = 45^\circ$  wrist joint configuration I

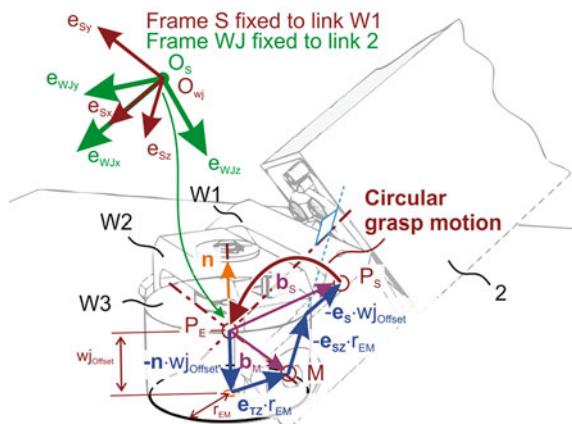


**Fig. 17.11** Enhanced grasping and with object orientation  $\varphi = 45^\circ$  and wrist joint configuration II



**Fig. 17.12** Enhancements for successful grasping

**Fig. 17.13** Computation of circular grasp motion

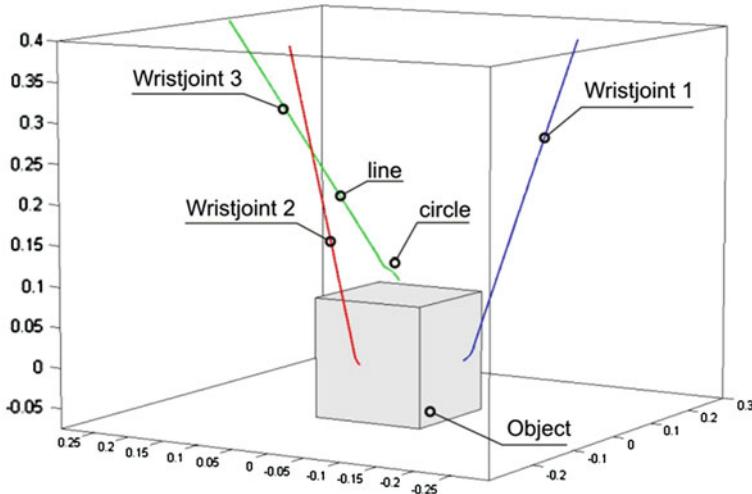


or while grasping. During object manipulation, the stepper is driven passively by the closed-loop kinematic chains of the complete robot architecture.

Figure 17.11 depicts that prepositioning of joint angle  $\alpha_R$  provides to pick up the object in any pose on this workspace layer.

To sum up, the tests show that a positive passive alignment of the contact element can be achieved for any object pose on ground level by the three modifications depicted in Fig. 17.12:

- Guiding the wrist joint center on a circle
- Enhancing the frictional conditions at contact point by modifying the contact element
- Prepositioning of joint angle  $\alpha_R$  by a low cost stepper



**Fig. 17.14** Visualization of computed example grasp paths

The friction conditions can be enhanced by using a ring of rubber on the edge of the contact element. The necessary joint angle  $\alpha_R$  for grasping can be computed implicit with the help of Eq. (17.2) and is used for prepositioning.

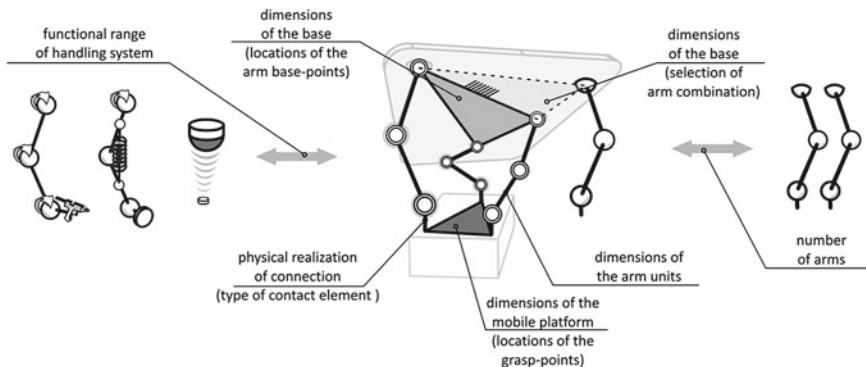
Finally, for optimal grasp behavior a circular path is needed, once the magnet touches the object. The center of the circle is coincident with this touch point. The exact circular path in 3-D space needs to be determined. This can be done with the vectors in Fig. 17.13 starting from origin  $O_{wj}$ :

$$\begin{aligned} \mathbf{bs} &= -\mathbf{e}_n \cdot \omega \mathbf{j}_{\text{of fset}} + \mathbf{e}_{TZ} \cdot \mathbf{r}_{EM} - \mathbf{e}_{SZ} \cdot \mathbf{r}_{EM} - \mathbf{e}_{SX} \cdot \omega \mathbf{j}_{\text{of fset}} \\ \mathbf{b}_M &= -\mathbf{e}_n \cdot \omega \mathbf{j}_{\text{of fset}} + \mathbf{e}_{TZ} \cdot \mathbf{r}_{EM} \end{aligned} \quad (17.2)$$

where  $wj_{Object}$  is the distance between the joint center and the contact surface and  $r_{EM}$  is the radius of the contact element. The computed complete grasp paths of the wrist joint centers for all three arms are shown, Fig. 17.14. The first arbitrary part, here a straight motion, and second part, the circle may have different curvature at transition. This needs to be considered in trajectory planning to avoid vibrations.

## 17.4 Reconfiguration

To attain the required flexibility, a fast and easy reconfiguration of the handling system is necessary. In the case of the presented manipulation concept, the system configuration can be designed by changing the parameters and modules illustrated in Fig. 17.15.



**Fig. 17.15** Parameters and modules for configuration

Three fundamental possibilities to adapt the configuration to a certain task are given within this “modular-robot-system”. The first one concerns the variation of specific kinematic parameters, like the link lengths inside one robot arm and the joint positions on the mobile platform and the base of all arms together. Changes here have great effect on the workspace and the overall performance of the complete robot system, such as wrench and velocity capability, stiffness and accuracy.

The second possibility is to enhance the flexibility by scaling the system. In this sense, scaling does not mean “variation in the size” but rather the “variation in the quantity” of the units. By removing and adding supplementary arm units a wide range of objects with different sizes and weights can be handled. Redundancy in kinematics or actuation improves stiffness and payload and offers to support large and sensitive objects at many points.

The third possibility to affect the capacity of the robot system is to vary modules with different functions. The integration of measurement, manufacturing, joining, or balancing units expands the overall functionality.

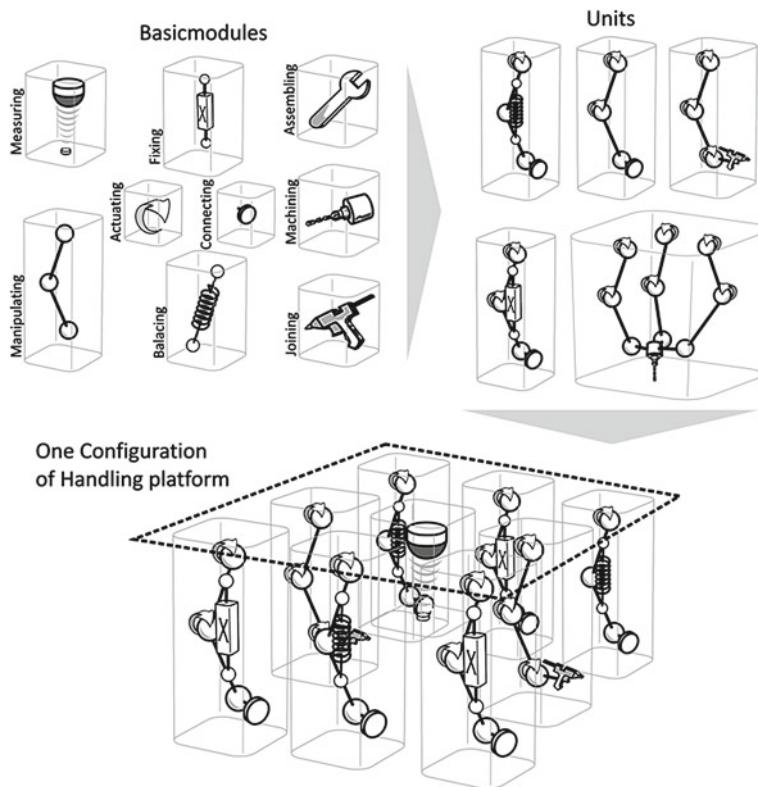
Combining these three aspects in one concept a “modular-robot-system” can be designed as schematically illustrated in Fig. 17.16.

All these configuration parameters have different influence on the performance, the functionality and the capability of the robot. Here we focus on an alterable kinematic structure, which is mentioned above as the first configuration possibility.

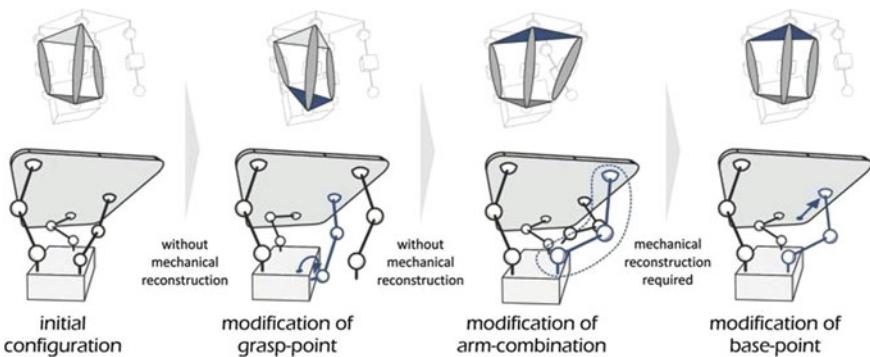
To be more precise, in the following cases, the kinematic lengths of a single arm are assumed to be fixed and “only” the grasp points on the object, the base points of the arms, and the selection of the arm combination which guides the object, are variable and have to be determined.

By modifying these parameters the kinematic dimensions of fixed and mobile platform are adjustable, as illustrated schematically in Fig. 17.17.

As described above, the object is integrated into the kinematic structure and becomes the mobile platform of the parallel manipulator. Hence, changing grasp points means changing kinematic dimensions of this platform. As capabilities of

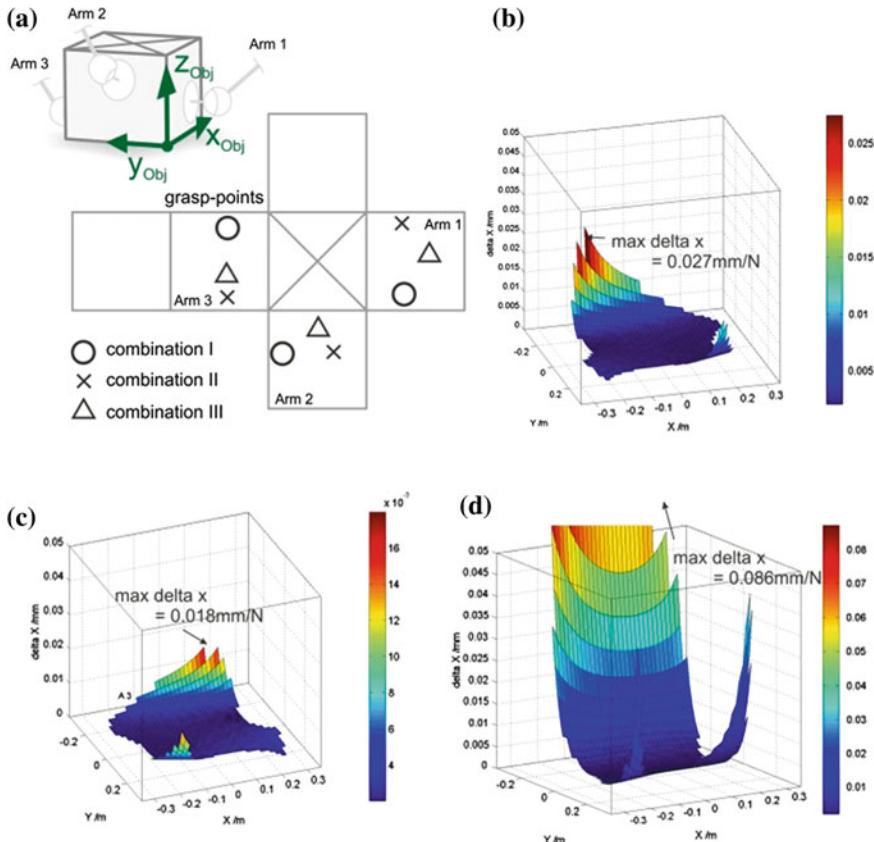


**Fig. 17.16** Schematic view of a “modular-robot-system”



**Fig. 17.17** Modifying base and mobile platform

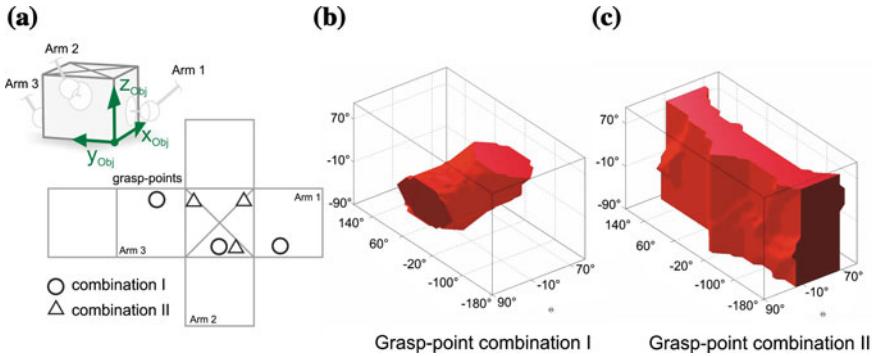
parallel manipulators are very sensitive to these dimensions and as grasping the object is required before every motion, the selection of appropriate grasp points is the most important configuration parameter of this system.



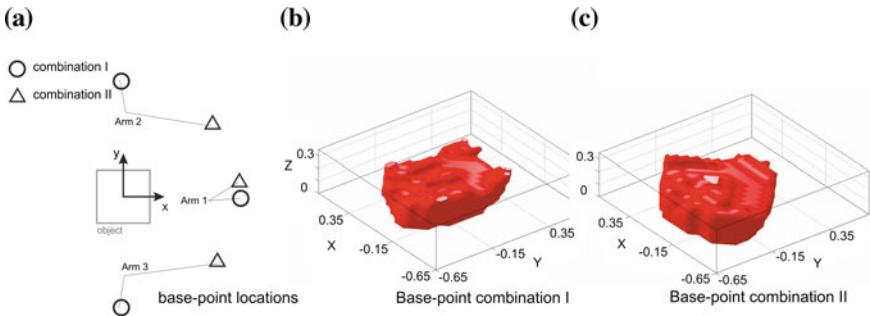
**Fig. 17.18** Different stiffness performance with different grasp-points. **a** Location of Grasp-Point combinations on the objectssurface, **b** Grasp-point combination I, **c** Grasp-point combination II , **d** Grasp-point combination III

Figure 17.18 presents the impact of different grasp combinations (a) on the capacity to withstand a unit force of 1 N on the object, here in  $x$  direction. The diagrams (b)–(d) show the position error of the object in the  $x$  direction caused by the external force and the compliant elements in the robot arm in one layer of the workspace.

Comparing combination I (b) and II (c), it can be seen that the stiffness of the robot varies over 477 % in some poses by modifying platform dimensions. Although the grasp points are different, the translational workspaces are approximately the same. In contrast to this effect, the rotational workspace, which describes the manipulator's capability to orientate objects freely in space, is strongly affected by the location of the grasp-points. Figure 17.19b and c show the rotational workspaces for the two grasp combinations in Fig. 17.19a about the



**Fig. 17.19** Effects of grasp-point locations on rotational workspace

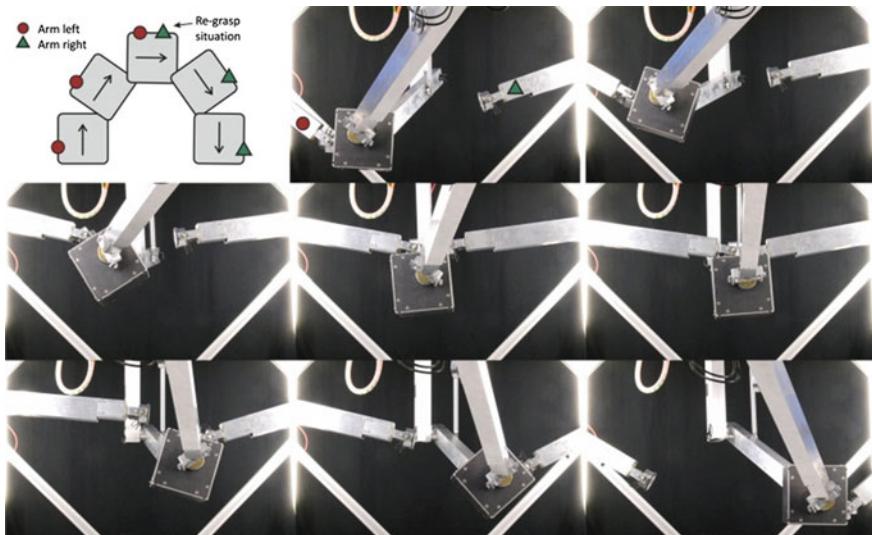


**Fig. 17.20** Effects of base-point locations on translational workspace

Euler angles  $\varphi, \theta, \psi$  ( $z, y', z''$ ). In all two cases the object is a cube, which is located in the center position of the translational workspace. Combination II (c) allows a large rotation about the angle  $\Phi$ . If one grasp point is on the top surface the versatility is enhanced in comparison to I and a large rotation of about  $\pm 90^\circ$  about the  $\theta$  axis becomes possible.

The selection of suitable base points of the single arm units is of high importance as well. These points can be varied by moving the mobile designed arms to a different location or by selecting an arm combination out of a group with more than three arms. The base-point locations of the selected group mainly define the translational workspace of the robot as shown in Fig. 17.20. Here, the constant orientation workspace is presented for a robot with two different base-point combinations, handling a slim object, like a plate.

It is obvious, but it should be mentioned, that changing some kinematic parameters of a parallel manipulator has effects on both, translational and rotational workspace, as well as on every performance criteria of the robot. Here, changing grasp or base points has primarily the shown impacts, but affects all aspects of motion capabilities as well. The combination of both, grasp- and



**Fig. 17.21** Regrasping an object for large rotation motion

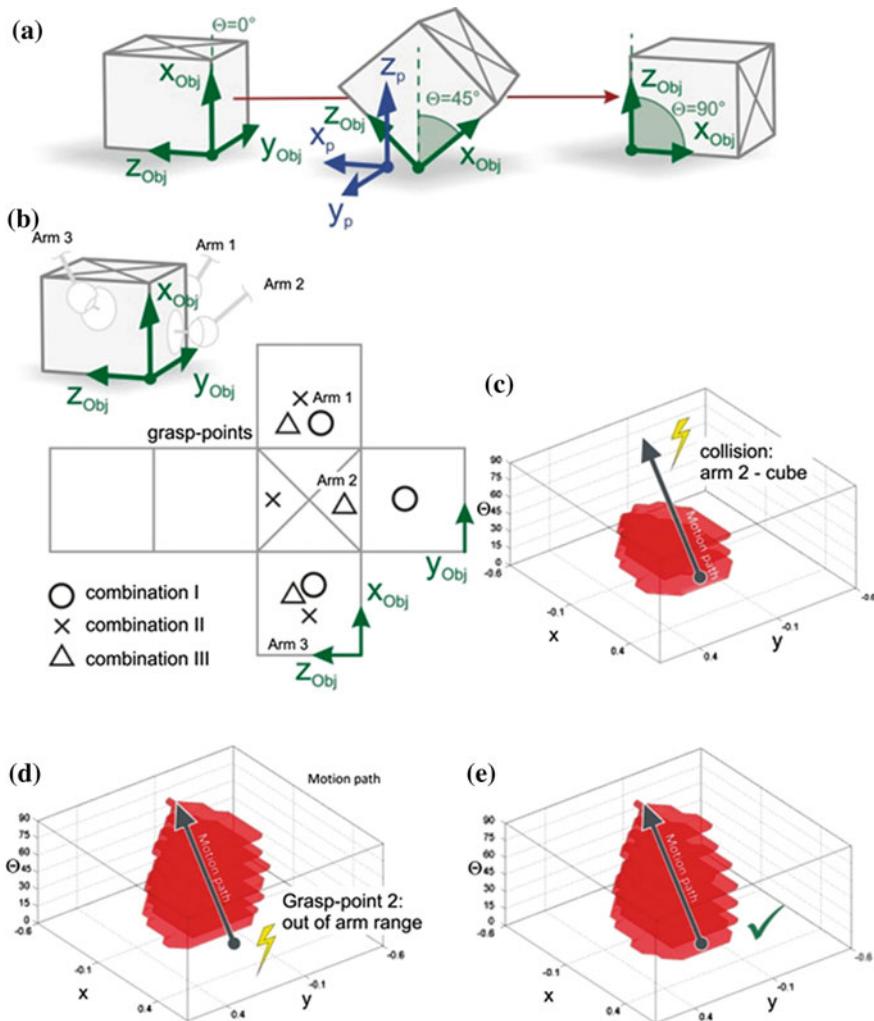
base-point selection, offers the user a wide range of possibilities to design the robot's versatility and performance.

Furthermore, if more than three arms are available in the basic layout, the configuration can be adapted even during dynamic motion, by integrating, removing or resetting arms. This “re-grasping” is similar to the motion behavior of a human hand flipping objects between the fingers. Reconfiguration by regrasping objects in motion creates completely new potential for novel and difficult manipulation tasks, like unlimited object rotations about any auxiliary axis in space. Figure 17.21 shows a motion sequence when manipulating a cubic object with a large 180 degrees rotation, with one additional arm.

## 17.5 Reconfiguration Examples

### 17.5.1 Case 1

The reconfiguration possibilities and their effects on the capability of the robot have been discussed above. In this chapter three examples of realistic motion cases are presented. For each case, the workspace criterion is analyzed by examining the actuated and the passive joint angles as well as the collision restrictions. For the sake of clarity the kinetostatic performance criterion is not considered here, but in general, they have to be evaluated to find an optimal robot configuration.



**Fig. 17.22** **a** Motion case with grasp-point variation, **b** Grasp-point combination, **c** grasp-point combination I, **d** grasp-point combination II, **e** grasp-point combination III

As shown in Fig. 17.22a, the first case represents a simple linear translational movement of a cubical object along  $x_p$  direction, combined with a  $90^\circ$  rotation about the rectangular axis  $y_p$ . The workspaces in  $X_p$ ,  $Y_p$ ,  $\theta$  dimension for the three grasp-point combinations are illustrated in Fig. 17.22c–e.

Grasp-point combination (I) allows to pick up the cube, but causes a collision at the end of the motion. This can be seen in Fig. 17.22c, when the motion path, illustrated by the bold black line, leaves the workspace slices. In all Fig. 17.22c–e,

the translational workspace along  $x_P$  and  $y_P$  direction as well as the rotational workspace about  $y_P$  axis is depicted.

The attempt to grasp in combination (II) involves a failure directly at the beginning of the motion, because the grasp point of arm 2 is not inside the required workspace. By an automatic planning algorithm, the solution (III) can be found which offers an appropriate workspace to perform the motion without any collision.

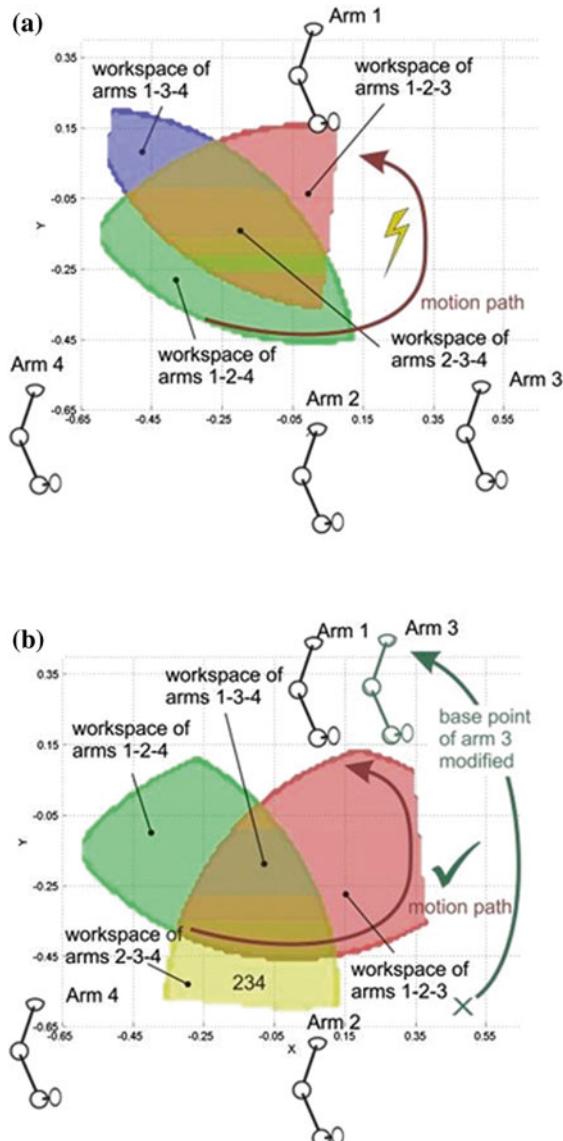
### 17.5.2 Case 2

In the next case, an object with the shape of a plate should be moved along a given path with a constant orientation. The present robot layout consists of four arms being located on the base. Here, the grasp positions are given and not variable.

Figure 17.23a shows the simplified XY-planar workspaces for the four-arm combinations in different colors and the required motion path together in one illustration. The workspace of the arm combination 2–3–4 is not visible, because it is located exactly in the intersection area of the other three. One can see that none of these combinations are capable of performing the motion from the beginning to the end. Furthermore, one segment of the path is unreachable in each three arm combination. This implies that even regrasping will not succeed here. A rearrangement of arm 3 is the solution, as presented in Fig. 17.23b. Here the four translational workspaces are displayed as well. The combination of arm 1, 2, and 3 can accomplish the task.

### 17.5.3 Case 3

In the last case a very complex motion shall be performed. For the sake of clarity in illustration of the workspaces, here, the object orientation during motion remains constant. It should be pointed out that even large rotations with  $360^\circ$  or more about any auxiliary axis are possible, but very unclear to picture. Figure 17.24 shows a layout with eight robot arms. Finding appropriate combinations of three arms, respectively, yields to a large combined workspace, which covers the complete path. This motion can be performed by regrasping, which means changing the arm combination in the areas, where the individual workspaces overlap. Here at regrasping only one arm is changed at once. Replacing directly two or even all three arms is sometimes possible, when layout features the appropriate number and locations of the arms [15].

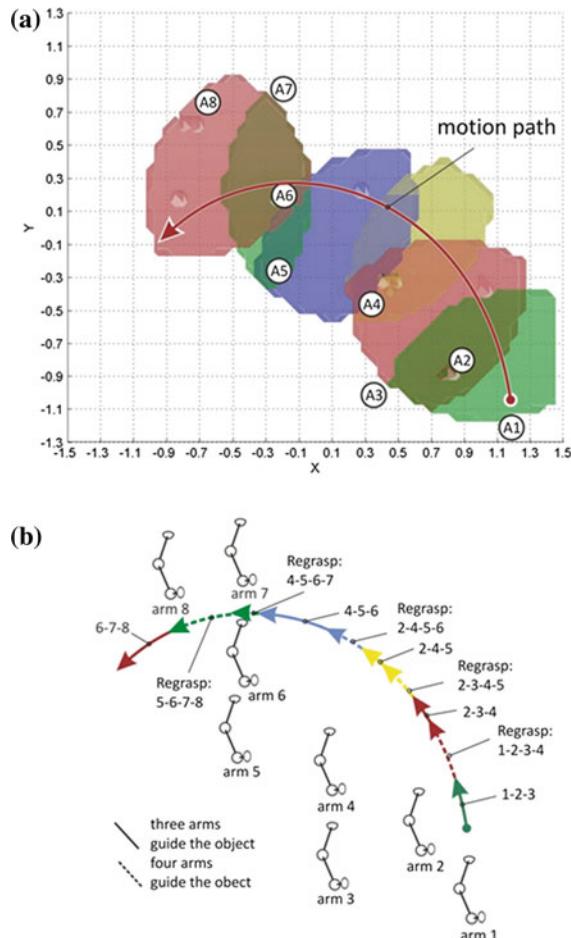


**Fig. 17.23** Motion case with base-point variation. **a** Initial configuration, **b** optimized configuration

## 17.6 Configuration Planning

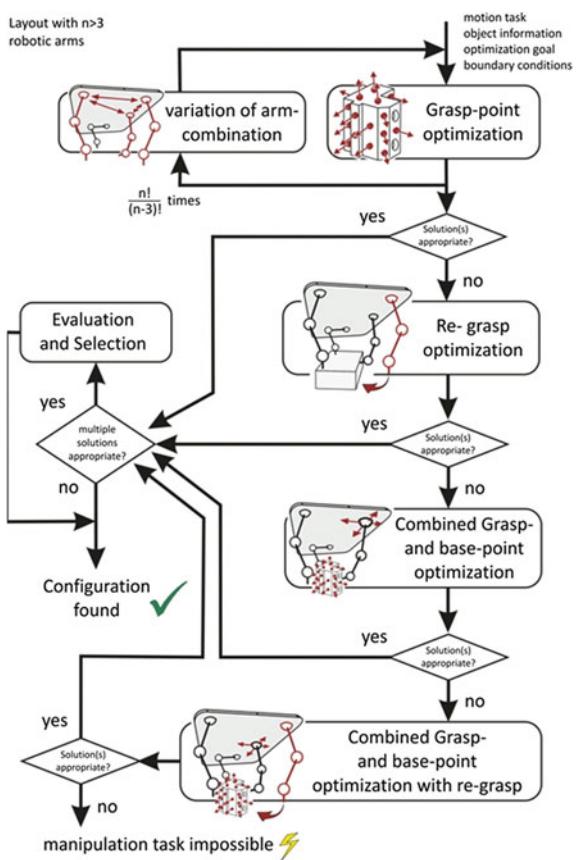
A handling system with an alterable mechanical layout offers flexibility on the one hand, but also increases the complexity on the other. Adjustable parameters imply supplementary DOF. These need to be planned carefully or optimized automatically

**Fig. 17.24** Motion case with base-point variation.  
**a** Base points and workspaces of arm-combinations,  
**b** planning of motion segments



as, in general, there are unlimited configurations feasible. The economic efficiency of such a system is strongly dependent on time and effort of the required planning, reconfiguration sequence and calibration of the kinematic layout. A lightweight and special arm design features a quick connection module for a fast and easy manual rearrangement of the arms on the base. High precision of the user is not required here, because the location of all arms and the complete layout configuration is detected automatically by a measurement system. In the automatic control system, the kinematic model of the structure is adapted to the real layout of the present handling system.

**Fig. 17.25** Flowchart for complete layout planning



With regard to a consistent efficiency in all reconfiguration steps, the automatic planning tool for an optimal system layout is essential, because a user cannot manage this high complexity on his own. A complete workflow for a motion with a reconfigured robot layout has five main steps:

1. Automatic or manual input of object, path, and trajectory data as well as other boundary conditions
2. Automatic configuration planning
3. If necessary: manual addition, removal, or rearrangement of the arms
4. Self-calibration of the layout with measurement system
5. Realization of the manipulation task.

When taking a closer look at the planning software, it is obvious to imagine that finding an optimal layout solution is not a straightforward problem, as it can be compared with a dimensional synthesis of a parallel manipulator. In our case we define “optimal” as the solution, which offers minimal reconfiguration time and effort as well as the achievement of a user given goal, like minimal resulting

actuator power or velocities, minimal motion errors, or high stiffness performances and in addition satisfies all the given boundary conditions. To reduce the duration for a complete workflow, the optimization algorithms have a priority list that is linked to the flow chart, depicted in Fig. 17.25.

If more than three arms are available on an adjustable base, the least reconfiguration effort is given, when no arm needs to be relocated. Hence, for the present and unmodified base-points the planning algorithms first search for appropriate arm combinations and corresponding grasp points. If one or more boundary conditions could not be satisfied with a static arm combination, the algorithms try to find solutions with alternating arm combinations and changing grasp points during motion, realized by regrasping. As the base points still remain at the same location any mechanical rearrangement in the layout is not required.

Moving one or more base points is the next optimization step if still no adequate solution is found. New grasp points, new arm selections and arm locations, as well as new regrasp configurations need to be optimized in combination.

The layout planning tool works with an object and base description in a discrete and continuous form, respectively. As pointed out in [13], optimizing in continuous space finds solutions with slightly better performance, but at a longer computation time.

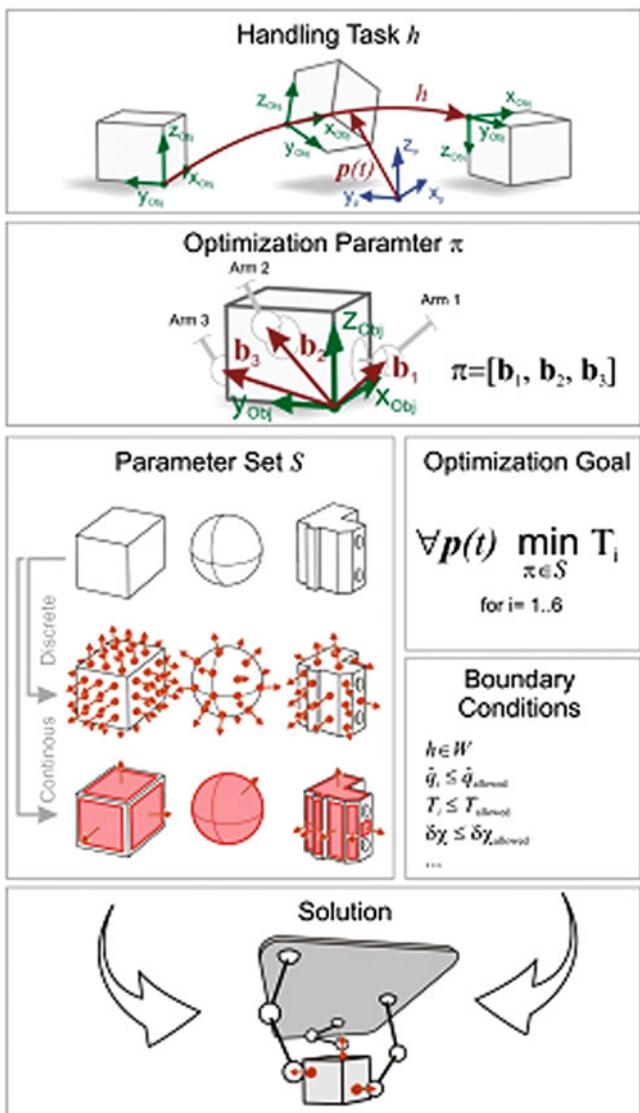
After introducing the possibilities of reconfiguration planning for the robot system, the grasp-point optimization and the algorithm for dynamic regrasp will be presented in detail in the following sections. Thus the main aspects of the flowchart will be shown.

## 17.7 Grasp-Point Optimization

As the handling concept is based on the integration of the object into the kinematic structure, some robot performances can be directly affected by choosing adequate positions of the connection points while grasping. The influence of a bad choice of these grasp points becomes very clear when taking a look at the following two negative examples. If the grasp points lie at or close to an imaginary line the complete robot configuration becomes uncertain in every object pose. In this singular situation the robot structure cannot withstand a wrench on the object and would collapse. In a different case, case 1 in the above mentioned reconfiguration cases, the grasp points are positioned well for force transmission but may reduce the mobility of the object, as for example large rotations could cause collisions.

That means, by choosing the connection points, some kinematic dimensions can be designed depending on the task to enhance workspace and dexterity as well as force and velocity transmission, accuracy, and stiffness performance. As the selection of an appropriate grasp-point combination is very challenging to a normal user an automatic optimization strategy is required.

A feasible optimization sequence which supports the user when planning the robot motion is shown in Fig. 17.26.



**Fig. 17.26** Grasp-point optimization

Among the information about the manipulation task, the optimization aim and the restrictions, especially the possible grasp positions on the object, are of importance. Therefore, these positions could be generated automatically by scanning CAD data or be given manually by the user. The optimization algorithms can handle both, discrete and continuous descriptions of possible grasp areas, like

a list of coordinates or a set of boundary for given surfaces. After computing an appropriate configuration the object motion can be performed.

The finding of adequate kinematic lengths on the object is similar to a dimensional synthesis of parallel manipulators. In [11, 16, 17], some optimization strategies are discussed which deals with enhancing kinetostatic performance indices all over the workspace to obtain design solutions that perform well for any trajectory.

These computations, which are used during the design process of the robot, could take several hours or even days to find one or a set of solutions depending on the complexity of the problem. This type of time-consuming optimization for grasp planning is not tolerable, as the user needs to wait after defining the handling tasks in the robot interface. The same applies to industrial applications with assembly belts featuring optical object detection via cameras, where a time critical planning of robot motion is required.

Depending on the information about possible grasp areas two approaches have been investigated. The first one deals with a given set of discrete grasp points and tries to find the best solution by browsing through these points. The second one uses a gradient-based optimization algorithm to search in a combination of continuous surfaces. Both algorithms are implemented in MATLAB and integrated in a graphical user interface to control robot motion. Figure 17.26 shows schematically the optimization process, starting from the definition of the handling task  $h$  in Workspace  $W$  and the optimization parameter vector  $\pi$ , going on with the specification of the parameter set  $S$ , the optimization goal and the boundary conditions, to run the optimization algorithm and find a good solution for coupling points on the object.

The optimization ensures that the chosen points can be grasped successfully with an underactuated wrist joint architecture by checking the angle combination  $\alpha_R$  and  $\alpha_S$ , see Fig. 17.8. Furthermore, collisions as well as the permissible range of joint motions are observed. These restrictions guaranty, that the workspace criterion is not violated. An optimization goal could be, for example, to find the grasp points which features the minimal actuator torque  $T_i$  for all actuators along the path. As all kinematic performances influence each other, additional boundary conditions need to be defined. These could be limits on maximal tolerable actuator velocities  $q_{i\_allowed}$  and torques  $T_{i\_allowed}$ , necessary object pose accuracy  $\delta\chi_{allowed}$  or required stiffness  $f_{allowed}$  of the robot.

To reduce computation time in contrast to general dimensional syntheses of parallel robots three simple steps are included in the grasp-point optimization.

As in most cases, the handling task is explicitly given, an optimization for each object pose in the complete workspace is not target aimed and too time consuming. Hence, all performance values are only computed for the object poses, velocities, and accelerations along the given path. The roughness of the path discretization determines significantly the computation time. The main time benefit arises from a pre-selection of possible grasp points or areas. Here, single points or areas are checked regarding the orientations of wrist joint vector  $e_{wJ_x}$  and surface vector  $n$  (see Fig. 17.26) without determining the robot performance and are listed in a new

**Table 17.1** Results of an optimization example

Task definition							
Path	cubic spline	P1	P2	P3			
	$[x,y,z]^T$ in [m]	$[-0,15;0,1;0]^T$	$[-0,05;0,15;0,1]^T$	$[0,1;0,15;0]^T$			
Trajectory	$[\varphi,\dot{\varphi},\ddot{\varphi}]^T$ in [deg]	$[0,0,0]^T$					
Motion	law of motion	5th order polynomial					
		duration	1	s			
		length	0,48	m			
		max. transl. velocity	0,91	m/s			
		max. transl. acceleration	6,52	$m/s^2$			
Object	Shape	Cube					
	Size	150x150x150	mm x mm x mm				
	Mass	1	kg				
Optimization goal							
minimize drive torque along path							
Boundary conditions							
Restrictions		Grasp criterion	true				
		Workspace criterion	true				
Limits		Maximal drive torque	15	Nm			
		Maximal drive velocity	1000	rpm			
		Maximal object error $\delta X$	3	mm			
Results							
Optimization method	Case	Grasp point vectors $b_i$ [xyz] in respect to frame F_OBi in [m]	Optimization Time in [s] @2.66GHz	Grasp criterion	Workspace criterion	max T in [Nm]	max q in [rpm]
		$b_1$ -0,075 -0,060 0,030		true	true	31,0	784,4
non	A	$b_2$ 0,075 0,060 0,060	0	-	-	2,2	-
	B	$b_3$ 0,060 0,075 0,010	0	true	true	11,9	690,1
	C	$b_4$ 0,060 0,060 -0,060	0	true	false	-	-
	D	$b_5$ 0,060 0,060 -0,060	0	false	false	-	-
discrete	E	$b_6$ 0,060 0,060 0,060	0,000	734,59	true	8,6	647,1
	F	$b_7$ 0,060 0,075 -0,060	0,000	0,51	true	true	1,2
	G	$b_8$ 0,060 0,075 -0,060	0,000	0,10	true	true	13,3
continuous	H	$b_9$ 0,049 -0,049 -0,060	0,060 -0,035 0,060	4757	true	true	7,5
	I	$b_{10}$ 0,049 -0,049 -0,060	0,060 -0,035 0,060	84,50	true	true	7,5
	J	$b_{11}$ 0,060 0,075 -0,060	0,060 0,029 0,075	0,75	true	true	8,2

set of feasible grasp combinations. An additional option to make the grasp planning even faster can be chosen by the user when the determination of adequate point is very time critical. Here the algorithm does not search for the best point combination, but tries to find a solution which just satisfies the given limits, as fast as possible.

Table 17.1 shows the results of different optimization strategies for a given task. The first four solutions (A–D) are chosen arbitrarily, the next three (E–G) are optimized from a set of discrete start parameters and the last three solutions (H–J) were optimized gradient-based on continuous areas.

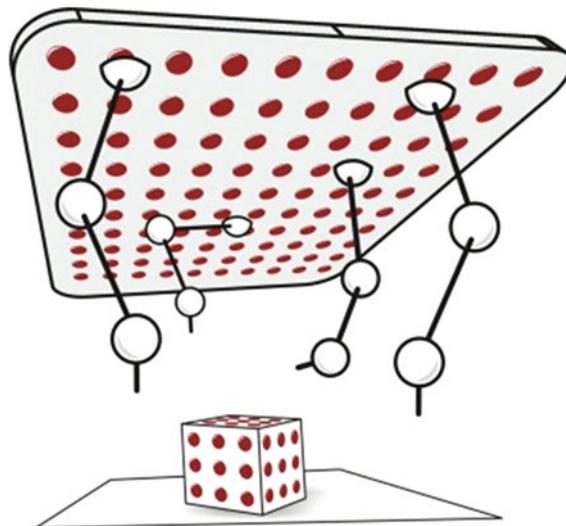
Although the selection of user-defined grasp points takes no computation time, the choice may be unsatisfying. Grasping the object in the configuration (A) leads to a high maximal actuator torque during motion. Otherwise this motion can be accomplished but with bad performance. The same applies to point combination (B), where the low accuracy of the object pose is not tolerable. However, in cases (C and D) the motion is not possible at all. In C the object can only be picked up, but the grasp points would cause collision between the robot links and the object during motion. Furthermore, the object in (D) cannot even be grasped in the initial pose.

Automatic grasp planning prevents the user from choosing bad points. For cases (E–G) a set of 54 grasp points on the cube is given, which can be combined theoretically in 148,824 ( $=54!/51!$ ) ways by the three arms. In (E) all those combinations are evaluated. This method finds the best solution in the set, but is, with 735 s duration, very time consuming. In (F) the pre-selecting process minimizes the number of points to be analyzed, which must lead to the same solution as in (E) but with a significant computing time reduction to only 0.068 %. Aborting the grasp-planning process when the first solution is found (G) enhances the computation speed additionally to 19.6 % total time in contrast to (F). The solution found here is usually not the best one in the set, but still satisfies the requirements as defined in the boundary conditions.

Analog to (E–G) the last three cases also show one optimization without pre-selection (H), one with pre-selection (I), and one with pre- and “first-best” selection (J). In these cases, the parameter vector  $\pi$  is optimized over continuous areas by using the gradient-based MATLAB function *fgoalattain*. In contrast to a discrete search these algorithms can also find solutions in between the set, which leads to better performances, however, the computations take more time. To prevent optimization from finding solutions only in local minima, each area is checked from several different start points.

In the end, the user has to decide which optimization method is best for each individual case, by trading off for performance goal and computation duration.

The base-point optimization works quite similar to the grasp-point optimization. In this case, the grasp points are given and the algorithm has to alter the base-point combinations, a combination of both just takes more time.



**Fig. 17.27** Initial configuration for discrete optimization

## 17.8 Dynamic Re-Grasping

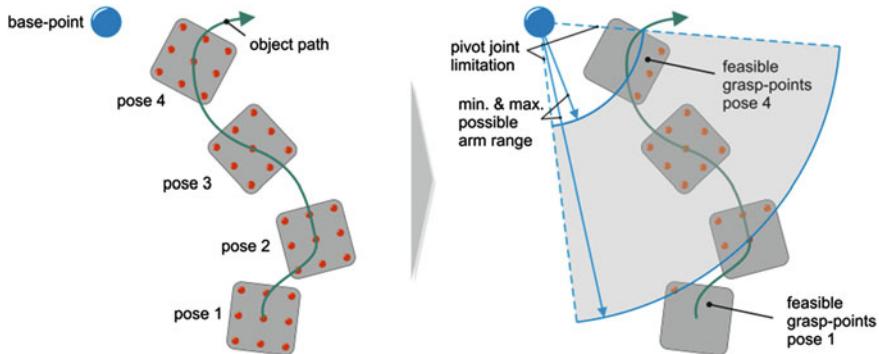
As shown in the flowchart in Fig. 17.25, an algorithm to find an appropriate set of regrasps is needed, if no solution is found after executing the grasp-point optimization. Case 3 in the above-mentioned example is an example for a desired motion, which cannot be performed without regrasping the object.

In the beginning of this section the motivation to reduce the field of solutions will be demonstrated. Afterwards some constraints for reduction will be presented. At the end of this section, a search algorithm will be introduced and a first performance measure will be made.

The following easy example should illustrate the complexity of a reconfiguration planning process with regrasps. As depicted in Fig. 17.27, four arms are available, which can be located on 10 times 10 base positions and can grasp a cube on nine points on each side. The object motion takes 10 s and the discretization for regrasp planning is 0.1 s. Here, it is not permitted to relocate the arms on the base while the object is in motion.

By variation of these parameters, three arms can theoretically grasp the object in  $1.4 \times 10^5$  ( $= 54!/51!$ ) combinations. The four arms available can be combined in four groups at three arms each. There are theoretically  $94 \times 10^6$  ( $= 100!/96!$ ) ways to arrange the four arms on the base.

During this sample motion, there are 99 times the possibility to evaluate an integration of the supplementary arm and a removal of an active one. There are 51 grasp positions left to connect to the object and three possibilities to release. To sum this up, referred to Eq. (17.3), there are  $2,69 \times 10^{229}$  theoretic configurations possible to manage this motion task with this few given customizable parameters.



**Fig. 17.28** Preselection of grasp points

$$C = \left( \frac{g!}{(g-h)!} \frac{n!}{h!(n-h)!} \frac{b!}{(b-n)!} \right) ((g-h)h)^{(m-1)} \quad (17.3)$$

with:

$n$  total number of arms in the layout (here:  $n = 4$ )

$h$  number of arms for manipulation (here:  $h = 3$ )

$g$  number of grasp points on the object (here:  $g = 54$ )

$b$  number of base-point locations ( $b = 100$ )

$m$  number of discrete motion points ( $m = 100$ )

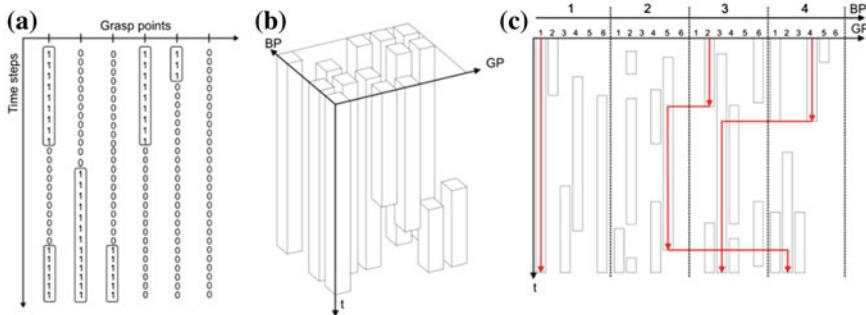
$C$  total number of theoretic configuration possibilities

The almost infinite configuration possibilities enabled by regrasping, turn a manual planning into a very challenging and inefficient operation for a nonskilled user. The demand for a quick, user friendly and economic reconfiguration of the system does also include the planning process. An automatic layout planning tool needs to optimize regrasp motions as well.

In our case, the positions of the base point should be defined before motion and are fixed. Thus, the number of theoretic combinations is reduced, see Eq. (17.4).

$$C = \left( \frac{g!}{(g-h)!} \frac{n!}{h!(n-h)!} \right) ((g-h)h)^{(m-1)} \quad (17.4)$$

In the above mentioned example a 10 s motion should be performed. With regrasping and without changing the base-points position there are  $1.14 \times 10^{222}$  theoretic possible configuration sets to perform this motion task. It is obvious that computing all possible combinations and evaluating their performance is unfeasible. A fast pre-selection algorithm can reduce the number of possible solution sets by separating feasible from unfeasible sets. The pre-selection algorithm is realized in three main steps. The first step scans the orientation of the surfaces. If any grasp-point surface is showing in a prohibited direction, which would violate the permissible joint angles, it is rejected. In the second step, the distance between



**Fig 17.29** a Recognition as segments for one arm, b segments for four arms, c computed configuration set

grasp and base point is analyzed and compared with the motion range of arm, respectively. In the last step all the critical joint angles are computed, to ensure that the point is inside the workspace. The third step only executes if the point passes the first two checks, which saves approximately 30 % of computation, in comparison to only a single phase preselection. As simplified illustrated in Fig. 17.28 the pre-selection scans a given number of motion points.

To compute the number of possible solution sets left after preselecting, a “regular” case can be approximated. If only one side of the cube is accessible for each arm the total set of solutions shrinks to  $3.26 \times 10^{145}$ . This number is still far away from any realistic and computable solution set. Hence, some boundary conditions have to be specified or added to constrain the field of solutions.

In Fig. 17.29a, the path planning matrix is shown for an object with only six grasp points (GP). The “ones” represent reachable points. A satisfying approach to downsize the field of solutions is to recognize serial points as segments, which are illustrated in the figure using black lined boxes. This method allows the planning tool to change the configuration of the arm combination only at a specific position of the segment, e.g., at the end of the segment. By computing the minimum time needed to regrasp, the segments which are shorter than this, can be rejected. The authors decided to use this deterministic approach, as it is easier to eliminate areas where no solution can be found, than with a probabilistic approach. In the actual state of the application this approach is practical, if in further applications the planning algorithm is more time critical, other approaches, for e.g. rapid random tree (RRT) method or probabilistic road map method (PRM), presented in [18], should be investigated.

After those steps of merging hundred motion points to just a few motion segments, the field of solutions is limited to a manageable size, which can be evaluated by a search algorithm. The task of the search algorithm is to find three completely independent ways through the segments to the end of the motion without violating the boundary conditions. Three boundary conditions are defined as follows: Only one segment per base point can be used at the same time, only one reconfiguration operation at the same time and the minimum distance between

the grasp points should not be smaller than the diameter of the wrist joint, to avoid a collision. In Fig. 17.29c one solution is illustrated. Here the red vertical arrows are representing the correlation of base and grasp point. The horizontal ones are marking the configuration changes. In this example the system is regrasping three times. In the first step, the costs of a solution are defined by the overall number of regrasp operations. Because the remaining regrasp operations cannot be estimated there is no heuristic available. Thus, only uninformed search algorithms can be considered. Because of its completeness and space efficiency, the iterative deepening depth-first search (DFID) is a suitable algorithm to solve this problem [19]. The depth of the algorithm matches the number of regrasp operations in our case. The algorithm first goal is to minimize the number of regrasp operations. In case the algorithm finds more than one solution set, the sets are compared and optimized regarding to a user given performance criteria, e.g., maximal drive torque, maximum accuracy, or maximum stiffness. These kinetostatic criteria as well as the algorithms used for optimizing the configuration parameters can be derived from the approach of a dimensional synthesis of parallel manipulators, as presented in [11, 20]. The algorithm solves the above-mentioned scenario, with a 10 s motion, at a 2.66 GHz CPU in less than 4 s.

## 17.9 Conclusion

In this chapter a novel handling concept, called PARAGRIP, where robot and gripper are combined in one unit was introduced. The structural synthesis brought a three arm system with a hybrid structure in each arm and an underactuated wrist joint. Because of its underactuated wrist joint a special grasp process was derived from practical tests with the wrist joint, to increase the grasping capability. As several kinematic parameters can be changed easily, the handling system can be reconfigured economically to match the tasks requirements. The configuration of the handling system can be defined by grasping the object and stays the same over the handling process or a dynamic changing of the arm configuration is possible, by integrating and removing arms during manipulation. On one hand, this provides the possibility to extend or to reshape the workspace dynamically on the other this extends the number of theoretical possible configuration sets to an incomputable number. Thus, an intelligent algorithm is needed to solve the problem in a practical feasible timespan.

**Acknowledgments** As parts of this work are within the scope of the cluster of excellence “Integrative production technology for high-wage countries (EXC 128)”, the authors thank the German Research Foundation for their support.

## References

1. Möller, N (2008) Bestimmung der Wirtschaftlichkeit wandlungsfähiger Produktionssysteme, Forschungsbericht IWB, Band 212
2. Müller R, Riedel M, Vette M, Corves B, Esser M, Huesing M (2010) Reconfigurable‘ self-optimizing handling system, precision assembly technologies and systems. In: Ratchev S (ed) 5th international precision assembly seminar, IPAS 2010, 1st edn. Springer, Berlin, pp 255–262. ISBN 978-3-642-11597-4
3. Nyhuis P (2008) Wandlungsfähige Produktionssysteme. Heute die Industrie von morgen gestalten. Garbsen, PZH, Produktionstechnisches Zentrum
4. Feldmann K, Ziegler C, Michl M (2007) Bewegungssteuerung für kooperierende Industrieroboter in der Montageautomatisierung. *wt werkstattstechnik online*, Jg. 97, Nr. 9 S.713
5. Gosselin CM (1988) Kinematic analysis, optimisation and programming of parallel robotic manipulators. Ph.D. Thesis, Mc Gill University. Montreal, 15 June
6. Ebert-Uphoff I, Gosselin CM (1998) Kinematic study of a new type of special parallel platform mechanism. DECT98/MECH-5962. In: Proceedings of ASME design engineering technical conference
7. Lee J-J, Tsai L-W (2002) Structural synthesis of multi-fingered hands. *J Mech Design* 124:272–276
8. Bicchi A (2000) Hands for dexterous manipulation and powerful grasping: a difficult road toward simplicity. *IEEE Trans Robot Automat* 16:652–662
9. Bicchi A (1995) On the closure properties of robotic grasping. *Int J Robot Res* 14:319–334
10. Nefzi M, Riedel M, Corves B (2006) Development and design of a multi-fingered gripper for dexterous manipulation. *Mechatron Syst* 4(1), Heidelberg. doi:[10.3182/20060912-3-DE-2911.00026](https://doi.org/10.3182/20060912-3-DE-2911.00026)
11. Riedel M, Nefzi M, Corves B (2010) Performance analysis and dimensional synthesis of a six DOF reconfigurable parallel manipulator. In: Proceedings of the IFToMM symposium on mechanism design for robotics, Mexico City
12. Laliberté T, Birglen L, and Gosselin CM (2002) Underactuation in robotic grasping hands. *J Mach Intell Robot Control* 4(3):77–87
13. Krut S (2005) A force-isotropic underactuated finger. In: Proceedings of the 2005 IEEE international conference on robotics and automation, Barcelona, Spain
14. Riedel M, Nefzi M, Corves B (2010) Grasp planning for a reconfigurable parallel robot with an under actuated arm structure. *Mech Sci* 1:33–42, Göttingen. ISSN: 2191-9151, 2191-916X
15. Riedel M, Mannheim T, Corves B (2011) More flexibility and versatility in automated handling process with an alterable parallel manipulator, accepted at DECT2011/MECH 47929. In: Proceedings of the 35th mechanisms and robotics conference
16. Angeles J, Park FC (2008) Performance evaluation and design criteria. Springer handbook of robotics, pp 229–244
17. Hao F, Merlet J-P (2005) Multi-criteria optimal design of parallel manipulators based on interval analysis. *Mech Mach Theor* 40:51–171
18. LaValle S (2006) Planning algorithms, Chapter 5. pp 228–248. Cambridge. ISBN 9780521 862059
19. Korf RE (1985) Depth-first iterative-deepening: an optimal admissible tree search. *Artif Intell* 27:97–109
20. Nefzi M, Riedel M, Corves B (2008) Towards an automated and optimal design of parallel manipulators. Arreguin, J.M.R. Automation and Robotics. I-Tech, Wien, pp 143–156. ISBN 978-3-902613-41-7

# Index

## A

- Active gripper, 428
- Adhesive bond, 414
- Agricultural robot, 385, 387, 388, 391, 393, 394, 404, 405
- Agricultural task, 385–388, 404
- Agriculture, 385–387, 389, 391, 393, 404
- Arm-mounted camera, 269, 270, 280, 297, 298
- ASIBOT, 223, 224, 235–237, 240, 242, 244
- Attached problem, 160

## B

- Body-coordinate formulation, 57–59, 62, 65, 69

## C

- Camera-robot calibration, 273–275, 277, 278, 280, 284
- CaPaMan, 305, 308, 314–324, 326
- Capillary grippers, 423
- Cardan angles, 199, 200, 203–206, 209, 211, 212
- Cartesian stiffness matrix, 17, 19, 21, 24, 52
- Clear-grip test, 300, 301
- Closed-chain, 65, 68–70
- Closed-loop robot-vision architectures, 269
- Collision-free grasping, 282, 287, 299
- Collision-free motions, 333, 350
- Collision-free object grasping, 269
- Constraint equation, 61, 63
- Cutkosky taxonomy, 385
- Cutting, 390, 392–397, 399, 400, 403, 404

## D

- Design grippers, 107–109, 125, 128, 130
- Dextrous manipulation, 89, 375, 380
- Differential-algebraic equations, 57
- Direct visual servoing, 269, 276
- Dynamic analysis, 210
- Dynamic manipulability, 211–215

## E

- Electrostatic microgrippers, 421
- End-effector, 385, 387, 390, 391, 393, 394, 397, 398, 401–404
- End point closed-loop (ECL) system, 277, 280, 281
- End point open-loop (EOL) system, 277, 280, 281

## F

- Feature tracking, 273–275, 278, 287
- Finger, 387, 394–401, 403, 405
- Finger mechanism, 134–138, 140–142, 145, 146, 148–152, 154, 155
- Finger orientation, 167, 170
- Fingerprint model, 270, 299–302
- Flange, 364, 365
- Force closure, 414
- Force control, 42, 43, 99, 148, 173, 225, 226, 230, 232, 233, 257, 260, 262, 266, 371, 376
- Force element, 63, 64
- Force sensing, 81, 89
- Form fit, 414

**F (cont.)**

Forward dynamic analysis, 73  
Freezing grippers, 414

**G**

Genetic algorithm optimization, 312  
Global stiffness index, 25–27  
Grasping, 329–333, 344, 346–350, 352–357  
Grasping model, 270, 290, 293, 299  
Grasping tasks, 306  
Grasp planning, 330–334, 337, 341, 342, 356  
Grasp point optimization, 455, 456, 457  
Gripper jaws, 366  
Guidance vision, 269, 270, 273, 292, 293, 295

**H**

Hand, 366, 369, 375, 380–383  
History of grasping systems, 3, 4  
grippers, 4–13, 15  
hands, 3–5, 7–9, 13–15  
mechanics of grasp, 10, 11  
mechanisms, 4, 6, 8–11, 13  
History of mechanical engineering, 354  
Hot melt adhesive, 411, 427  
Humanoid robots, 329, 330, 333, 356

**I**

Image feature parameters, 272, 278  
Image features, 270, 272–277, 281–283, 298  
Impactive gripper, 388  
Industrial grippers, 109, 115, 117–120, 128, 129  
Inverse-dynamic analysis, 73  
Inverse dynamics model, 208

**J**

Jacobian matrix, 62, 72, 205, 215  
Joint-coordinate formulation, 57, 58, 64, 65, 70–72

**K**

Kilo floating-point operations per second (KFLOPS), 249  
Kinematic analysis, 73  
Kinematic joints, 58, 60, 61, 65, 66

Kinematic synthesis, 194

**L**

Local stiffness index, 24–26  
Look-and-move servoing, 276  
Lumped stiffness parameters, 20, 21, 23, 24, 26, 28, 29, 32

**M**

Manipulability ellipsoids, 211, 213, 214  
Mechanical microgrippers, 415  
Medial axis, 329, 330, 332–339, 341, 356  
Microgrippers, 411  
Microgrippers within microassembly, 425  
Million floating-point operations per second (MFLOPS), 250, 255  
Million instructions per second (MIPS), 249, 250, 252  
Modular robot system, 445, 446  
Motion planning, 329, 331–333, 356, 435, 444, 451, 453–455, 457, 461, 462  
Multibody dynamics, 57, 63, 74  
Multi-objective optimization, 305, 308, 309

**N**

Newton–Euler equations, 57, 62, 65

**O**

Object finder, 301  
Object space, 272  
Open-chain, 65–69, 71  
Open loop robot-vision architectures, 269  
Ordinary differential equations, 57, 58  
Orientation mechanism, 159, 165, 167, 169, 170, 187

**P**

PARAGRIP, 433, 434, 435, 463  
Parallel kinematic machines, 189, 192, 210  
Parallel manipulator, 18, 22, 23, 31, 46, 49  
Parallel robots, 434, 457  
Passive and active microgripping, 411  
Path planning, 305–308, 311, 312, 319, 320, 326  
Piezoactuators, 417  
Pneumatic actuation, 417  
Precision grasping, 394, 395, 397  
Prehensile grasping, 388, 394, 398, 399

**R**

- Rapidly-exploring random trees, 329
- Reconfigurable parallel robot, 434, 435
- Re-grasping, 330, 333, 352–356, 449, 460
- Revolute joint, 61, 64, 66, 67, 70, 72
- Robot grasping, 77, 78, 85, 92, 102, 192, 269, 270, 287, 292, 293, 295
- Robotic hand, 18, 23, 29, 226, 363, 364, 380
- Robot-object calibration, 277, 282, 283, 285–287

Rotational inertia matrix, 63

**S**

- Safe gripping, 366
- Sensors, 366–368, 371, 376, 379, 382
- Serial manipulator, 18, 21, 22, 28
- Shape memory alloy (SMA), 417
- Spherical joint, 61, 67, 72, 73
- Spherical parallel machines, 192, 194
- Spherical wrist, 189, 194, 215, 216, 436
- Staying function, 394–396, 399, 400
- Stiffness, 17–29, 31–40, 42, 44–46, 48, 49, 52
- Stiffness analysis, 17, 19, 20, 28, 29, 52
- Stiffness model, 28, 29, 34, 36, 37, 52
- Surface mount devices, 420

**T**

- Tactile sensing, 81–84, 89, 102
- TEO, 223, 224, 240–244
- Thermal actuators, 417
- Tool changer, 365
- Tracking object for grasping, 270, 273–275, 278
- Two-finger gripper, 18, 23, 52

**U**

- Underactuated architectures, 223, 224, 244, 435, 436, 457
- Underactuated finger mechanism, 135, 137, 151, 152, 154
- Underactuated grasp process, 435, 438, 463
- Underactuated hand, 166, 187
- Underactuated mechanism, 134, 135, 139

**V**

- Vacuum microgrippers, 420
- Vegetable material, 391, 393, 398, 404, 405
- Velocity transformation matrix, 67–70
- Visual positioning task, 278
- Visual servoing, 269, 270, 275, 276