

FP7-IST-60918
1 March 2013 (36months)

DR 4.1: Autonomous selection and execution of grasp pre-shaped primitives within the soft-synergy framework

Michela Antonelli, Marco Gabiccini

Centro di Ricerca “E. Piaggio”, Università di Pisa
(m.gabiccini@ing.unipi.it)

Due date of deliverable: 2014-02-28
Actual submission date: 2014-02-25
Lead partner: Università di Pisa
Revision: final
Dissemination level: PU

This report describes the activities related to the definition, selection and execution of grasp primitives. It includes work on modelling underactuated hands within the soft-synergy framework, acquiring the target grasps using synergies and learned from data gathered either from humans or simulations, synthesising grasps for novel objects using either statistical machine learning or case based reasoning, and execution of such grasps using configuration dependent impedance control. The main task addressed is Task 4.1, but for completeness we also cover initial work we have been doing that crosses over with Task 4.3.

1	Tasks, objectives, results	4
1.1	Planned work	4
1.2	Actual work performed	7
1.2.1	Natural Redundancy Resolution in Dual-Arm Manipulation using Configuration Dependent Stiffness (CDS) Control	7
1.2.2	Grasp analysis tools for synergistic underactuated robotic hands	8

1.2.3	Grasp Movement Classification via Multiple Eigenspace	9
1.2.4	Synthetic Grasp Affordances for the Pisa/IIT SoftHand	10
1.2.5	Case-based Reasoning for Grasp Synthesis	12
1.2.6	Learning Dexterous Grasps That Generalise To Novel Objects By Combin- ing Hand And Contact Models	13
1.3	Relation to the state-of-the-art	14
2	Annexes	22
2.1	Natural Redundancy Resolution in Dual-Arm Manipulation using Conguration De- pendent Stiffness (CDS) Control	22
2.2	Grasp analysis tools for synergistic underactuated robotic hands	23
2.3	Grasp Movement Classification via Multiple Eigenspaces	24
2.4	Synthetic Grasp Affordances for the Pisa/IIT SoftHand	25
2.5	Case Based Reasoning for Grasp Synthesis	26
2.6	Learning Dexterous Grasps That Generalise To Novel Objects By Combining Hand And Contact Models	27

Executive Summary

This report describes the activities – most already performed while some others planned for the next weeks – within the PaCMan consortium to devise strategies to define, select and execute grasping actions, as defined in Task 4.1 and carrying over into the beginning of Task 4.3. This includes work on grasp selection (Annexes 2.3, 2.4, 2.5, 2.6), modelling of grasps for synergetic underactuated hands (Annex 2.2) and methods for control of grasping (Annex 2.1). This deliverable thus covers the raft of grasping relevant activity in WP4.

Role of grasp selection and execution in PaCMan

This deliverable documents the effort of the consortium in devising new strategies to define practical approaches to tackle the problem of grasping under uncertainty, either caused by clutter, visual occlusions, and new object shapes. To this end, strategies are presented based on impedance control, even in dual-arm tasks, on exploiting adaptability of the Pisa/IIT SoftHand, and on learning and generalizing dexterous grasps to novel objects.

Contribution to the PaCMan scenario

Grasp methodologies that exploit the compliance of the Pisa/IIT SoftHand, its learned grasp affordances and the ability to learn and generalize dexterous grasps to novel objects are central in the scenario depicted by the PaCMan project. In addition some of the approaches are applicable with other kinds of dextrous hands used in the project.

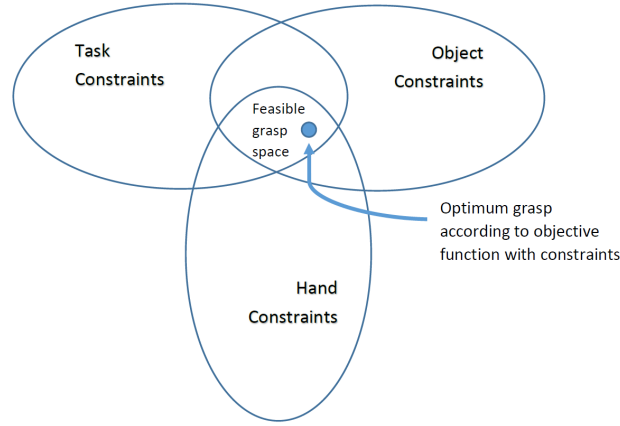


Figure 1: Feasible grasp space definition.

1 Tasks, objectives, results

1.1 Planned work

Despite optimistic claims, the task of grasping unfamiliar objects is still an open problem in the robotic community. Human beings have no trouble in selecting the correct grasp strategies even if the object to be grasped is unknown or partially seen. On the contrary, though robotics has made remarkable progress in the field of autonomous grasping, the problem of grasping novel objects in cluttered scenarios has not been completely solved yet.

In a recent review [1], Bohg et. al. classify the approaches used for autonomous grasping into two main categories, namely analytic and data-driven approaches. Analytic approaches are mathematically-based and usually use a multi-fingered robotic hand to build force-closure grasps. These hands have four important properties: they are dexterous, stable, in equilibrium and characterized by a dynamic behavior. In this case, given an object, the process of choosing a suitable grasp, referred to also as *grasp synthesis*, is usually formulated as a constrained optimization problem guided by one or more of the above four properties. As shown in Figure 1, the constraints depend mainly on three aspects: i) the task to be performed, that is the forces and motions that must be imparted, ii) the grasped object, in particular its shape, iii) the hand, that is the maximum grasp force applicable and the maximum opening of the fingers.

One of the main issues of this approach is that the grasp synthesis can be accurately studied only in simulation environments where the precise models of both the hand kinematics and object are available. Indeed, in the real world, noisy sensors and inaccurate models make the relative position of object and hand prone to errors, thus causing a lot of difficulties in the placement of the fingertips on the object.

In order to deal with uncertain data and behave robustly with respect to model errors, in Section 1.2.1, we propose a novel realtime motion control strategy to regulate the desired Cartesian stiffness profile during the execution of a bi-manual task. This approach will be further investigated and refined when facing the problem of object bi-manual passing, in future tasks within the PaCMan projects. This contributes to Task 4.3, but we report it here to give a complete picture of our efforts.

Another issue that affects the practical application of analytical approaches is the increasing complexity and number of degrees-of-freedom of the most employed robotic hands such as the UTAH/MIT hand, the Robonaut Hand, the Shadow hand and the DLR hand: in fact, as they require a large number of actuators to be controlled, this large amount of actuators makes it very difficult to find a joint control law able to perform a given task in a robust way. The need to find principled coordinated joint motion patterns appears to be inescapable.

A possible approach to reduce complexity is to cleverly cut down on the number of actuators: the outcome are underactuated hands that are simpler and more reliable than their fully actuated counterparts.

Focusing on grasping, [2] has shown that most of the grasping hand movement can be explained, in terms of statistical variance, just by the first two or three principal components or synergies. A direct consequence of this, was a new boost to the development of under-actuated and simplified hands, which embed hardware synergies in their mechanics to reduce the number of motors used to achieve most grasping tasks. The Pisa/IIT SoftHand that our group have proposed in [3] is an example of such under-actuated robotic hand. Other studies, like [4] and [5], have modelled analytically the ability of underactuated hands to realize force-closure grasps proving that the benefits associated to a reduced complexity are not necessarily coupled to a reduction in grasp performances.

In Section 1.2.2 we present a detailed analytical framework for the quasi-static analysis of synergistic underactuated grasps. Here we propose an approach where the set of equations describing the system is treated as a whole. Results are not limited to providing the controllable system variations by synergistic actuation but, in addition to this, systematic ways to discover relevant properties of the grasp are provided, even in symbolic form, and a method to define and study the feasibility of some tasks is given. This contributes centrally to all of WP4, as it provides the theoretical framework for modelling soft synergy grasps.

However, the possible applications of synergistic concept are not limited to hardware. Indeed, to simplify the joint control laws, software synergies can be simulated on a fully actuated robotic hands, as suggested by [6]. This can highly simplify the design phase of a grasp, by reducing the number of control variables. In Section 1.2.3 we introduce a global dimensionality reduction technique based on the idea of growing Multiple Eigenspaces, to obtain simple instructions to pi-

lot robotic hands in a novel way. This work, while bringing to the table a novel dimensionality reduction technique never applied before to the hand joint movement analysis, it strives to reconcile different approaches employed to categorize grasp taxonomies. In fact, even if much work is needed to obtain a more complete map of results, an attempt is made to compare data emerging hand shape patterns during grasping with visually defined grasp taxonomies, like the well-established Cutkosky's taxonomy.

On the other hand, data-driven approaches reduce the problems of determining joint control law, by avoiding the kinematical calculation of the fingertips' relative position on the object. Indeed, these approaches rely on sampling grasp candidates for specific object and ranking them according to some metrics. In this case, a grasp is usually characterized by the following four parameters:

1. the grasping point on the object which the tool center point should be aligned with;
2. the approach vector, describing the 3-D angle with which the robot hand approaches the grasping point;
3. the wrist orientation of the robotic hand;
4. an initial finger configuration.

Many data-driven approaches propose to find the suitable grasp of an object by learning from the experience acquired during grasp executions. In this framework, a knowledge base containing examples of pairs (object, grasp candidate) is a crucial aspect. Elsewhere in WP2 we have been working on ways to capture such representations from human demonstrations. Here we describe how it may be acquired by numerical simulations as explained in Section 1.2.4.

At the beginning of this project we planned to use the Columbia Grasp Database [7] as knowledge base for the learning algorithms. Actually, this database did not work out as we initially thought. Indeed, the 3D models of the objects are from the Princeton Shape Benchmark (PSB) and have not been meant to be used for robotic grasping. Furthermore, the hands contained in the database are all fully actuated while we have planned to use the Pisa/IIT SoftHand that is underactuated via the adaptive synergies [3]. For this reason, we have decided to build our own database with new methods explored in WP2 and those explained in Section 1.2.4. As said before, one of our main motivations for building a grasp database is to have a training set of grasping data to use for generating new grasp plans by means of learning strategies.

In Section 1.2.5 we explain how a Case-Base Reasoning approach (CBR) can be used to build a system able to autonomously grasp objects. We consider three categories of objects: known objects, i.e., the database contains both the 3D models and a set of associated successful grasps, familiar objects, i.e., object similar to the

ones already included in the database, and unknown objects, i.e., in the database there is no grasp experience recorded for the objects.

For categories of unknown objects, in Section 1.2.6 we propose a new method for generalizing grasps to novel objects. Indeed, the grasps are inferred from as little as one training example. The method generalises a human chosen grasp type (e.g. rim, pinch) trained from one or two examples, to a novel object of potentially quite different shape. These last two methods contribute to the selection of grasps in Task 4.1.

1.2 Actual work performed

1.2.1 Natural Redundancy Resolution in Dual-Arm Manipulation using Configuration Dependent Stiffness (CDS) Control

With the advent of higher demands on dual-arm manipulation, interaction with the environment using impedance control [7] became the focus of many studies in dual-arm systems. In this direction, object-level impedance control of cooperating robots has gained great attention due to the feasibility of controlling task forces/velocities while avoiding the occurrence of high internal forces [8].

In [8], relying on the major contribution of the arm configuration to effectively regulate the directionality of the realized endpoint stiffness ellipsoid, and its low-cost nature, we explore the role of configuration dependent stiffness (CDS) control, for dual-arm object manipulation. Therefore, the joint variables are controlled in redundant space to realize a task-specific Cartesian stiffness profile. In addition, with the purpose of obtaining a good tracking performance while rendering natural movements, we incorporate the optimality principles in human motor control in the development of proposed impedance controller. In this direction, given the growing body of evidence in support of coordinated stiffening behavior of the arm joints in humans, we utilize the concept of common mode stiffness (CMS) control.

The effectiveness of the proposed controller is evaluated in a dual-arm assembly task. The acquired good tracking performance and similarities between the natural postures of the human operators and the ones realized by the robots illustrate the contact efficiency of the proposed algorithm, while highlighting the advantages of the integration of human motor principles in redundancy resolution of dual-arm manipulation. More details on our method and on the obtained results can be found in the attached paper to appear in ICRA (Annex 2.1 [8]). This work begins to link work in Task 4.1 to Task 4.3.

1.2.2 Grasp analysis tools for synergistic underactuated robotic hands

The design of robotic hands was directed for a long time to obtaining better dexterity and capability for manipulation tasks. This goal was sought by increasing complexity and number of degrees of freedom, with remarkable examples of robotic hand designs following this trend represented by the UTAH/MIT hand, the Robonaut Hand, the Shadow hand and the DLR hand arm system. One of the main issues of this approach is that a large number of degrees of freedom implies many actuators, with a consequent growth of size, weight and cost. With such kinds of devices, another disadvantage is the difficulty in control. Finding a joint control law, able to perform a given task, becomes a difficult job, mainly due to the complex kinematic structures of the hands. Starting from these observations, different kinds of underactuation strategies were introduced and studied [9]. In this direction, a recently considered approach has been to take inspiration from the human hand, not just from a biomechanic point of view, but also for the control aspects.

In the attached paper (Annex [10]) we present a quasi-static analysis of synergistic underactuated grasps, and we propose an approach where the set of equations describing the system is treated as a whole. Results are not limited to providing the controllable system variations by synergistic actuation. In addition to this, systematic ways to discover relevant properties of the grasp are provided, even in symbolic form, and a method to define and study the feasibility of some tasks is given.

Since also the derivative terms of the Jacobian matrix are taken into account, configurations with contact force preload can be rigorously analyzed by the proposed method. Our choice is to use an object-centric formulation, obtaining a constant grasp matrix and some simplifications to describe rigid object displacements. The resulting system of equations is treated with two different approaches.

The first, is a systematic way to elaborate the system of equations. Input and output variables are defined and used in an algorithm of Gauss-Jordan elimination, extended to act on block partitioned matrices. In consequence of this, the relationships between the controls and the consequent hand/object configurations are given, both for displacements and forces, in two different ways, one more effective for numerical applications, the other more appropriate with symbolic forms of the equations.

This leads to finding relevant properties of the grasp, e.g. a basis for the controllable internal forces, controllable object displacements and the grasp compliance. Furthermore, in our work, the main categories of tasks are defined in terms of nullity or non-nullity of some variables. These definitions can be used to suitably apply the reduced row echelon form (RREF) to the solution space of the system, ascertaining their feasibility and the necessary inputs to be applied in order to perform them. More details of this work can be found in the attached paper that has



Figure 2: An example of a 1-dimensional linear space in hand posture space, i.e. one *eigenspace* of dimension one. Subject ED, spherical grasp movement.

appeared in IJRR (Annex 2.2 [10]).

1.2.3 Grasp Movement Classification via Multiple Eigenspace

In Task 4.1 we have also worked to find low dimensional representations of grasps from human data, in order to inform selection of soft synergy grasps in robot hands. Dimensionality reduction techniques have extensively been used in the field of robotic grasping, under many different forms, to obtain simple instructions to pilot robotic hands to mimic what human hands do. Human grasp analysis is becoming increasingly automated, and the trend is likely to continue during the next years. Nonetheless, there is still a gap which needs to be bridged when analyzing this sort of data, namely the classification of the type of grasp performed which is still a task mostly performed via direct inspection.

There is a large number of models and techniques which are useful for dimensionality reduction: Principal Component Analysis (PCA, [11]), Independent Component Analysis (ICA, [12]), functional PCA (fPCA, [13]), Hidden Markov Model (HMM, [14]), Gaussian Process Latent Variable Model (GPLVM, [15]), and many more. Lately, many of these techniques have been applied to postural and grasping data: for example, Peternel and Leonardis [16] showed that even movements as complex as human locomotion can be modeled by a small number of Degrees of Freedom (DoFs), i.e. 15 GMM in 4-D space. Bernardin et al. [17] approached the problem of fusing data glove and tactile sensor information applying an HMM recognizer to distinguish among different grasp types.

Still, the most widely used technique remains PCA. For example, in [18] the subjects have been asked to grasp a large number of imagined objects in order to reconstruct Principal Components, called *postural synergies*, and in [19], [20] the use of PCA is extended to *temporal* and *kinematic* postural synergies. These techniques have been applied to show that there is an underlying structure in the way humans grasp. A different procedure, namely direct inspection, has been used for

a seemingly different problem: generation of a grasp taxonomy. Efforts in this direction have been made since the 50s [21] and the 80s [22]. Still recently (see [23] and [24]), to classify grasping postures and movements, the same method used almost sixty years ago is applied, namely direct visual inspection. Some attempts at automating the procedure of data segmentation have been made in the last decade (e.g., [25], [26]), still they start by performing dimensionality reduction on the full dataset, and then look for a valid segmentation in the reduced dimensional space which is, indeed, affected by the population employed.

The work we have done aims to find a systematic, data-driven way to explain grasp taxonomy, which could then be used for automatic movement classification and hand-posture database, e.g. [27], indexing. The results of the global dimensionality reduction techniques cited above can be highly dependent on the dataset used. To avoid this dependency, we have used the Multiple Eigenspace technique [28]. Multiple Eigenspaces are a PCA generalization, which can be seen to be a single eigenspace which explains the whole dataset, while using more than one eigenspace the dataset is automatically clustered and, differently from other clustering techniques like K-means [29], each cluster can have a different dimension. Moreover, this approach gives lower dimensional subspaces than the ones given by PCA. These subspaces can be used for interpolation, thus avoiding to neglect the low dimensional local structure of the data, which is instead usually ignored.

Especially in robotics, where Programming by Demonstration is increasingly taking place [30, 31], a low number of DoFs is desirable. Using a technique which automatically groups together what can be achieved in a similar way should relieve the programmer of the choice of a suitable dataset, which could then just be big enough to contain what is wanted and the reduction would be done by the learning algorithm itself. With this in mind, our work focuses on movements rather than static postures, which can anyway be obtained as ending of a movement. Furthermore, in order to maximize the probability of grasp success, we have to consider the overall time course evolution of the hand shape, and not just the final pose to attain. The attached technical report (Annex 2.3) describes in detail the approach we have developed.

1.2.4 Synthetic Grasp Affordances for the Pisa/IIT SoftHand

For well known scenes and object models, pre-programming of autonomous grasping and manipulation tasks may be an option. To this aim, several approaches have been proposed to define the optimal finger placement on the object, either based on some geometric [44] or force [45, 46] grasp quality measures, specifically tailored to convex objects [47], with optimal on-line contact adjustments [48], and also generalized to non-convex objects [49].

On the other hand, adaptation of a grasping algorithm to formerly unknown or only

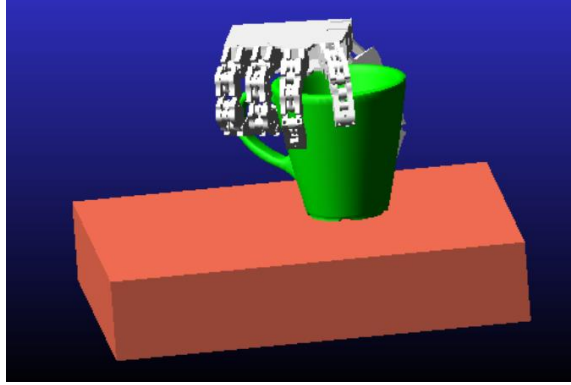


Figure 3: The Pisa/IIT SoftHand grasping a cup.

partially known scenes remains a difficult task, to which several approaches have been investigated. Some of them are model-free and propose geometrical features which indicate good grasps [50, 51], some others evaluate also topological object properties such as holes [52]. Typically, grasps are ranked on a fixed list of suitable hypotheses, do not require supervised learning, but do not adapt over time.

Other methods are based on learning the success rate of grasps given some descriptor extracted from sensor data, either evaluated on a real robotic system [53, 54], or on simulated sensor data [55, 56].

Beside vision-based methods, tactile exploration may be used for 3D reconstruction of an unknown object, as tactile sensing solves some severe limitations of computer vision, such as sensitivity to illumination and limited perspective. A combined procedure based on dynamic potential fields, that aims at reconstructing 3D object models, which are then used for grasp planning and execution was presented in [57] and recently extended in [58].

In this work, which contributes to Task 4.1, we present some results on the automatic creation of a database of grasp affordances for the Pisa/IIT SoftHand for a set of kitchenware objects. The method itself is not new since similar approaches to the grasp planning problem have been presented in the past by employing simulators as GrasIt! [59], Open-Rave [60], and OpenGRASP [61].

However, we claim that the main novelty of this contribution is related to the special hand design for which grasp affordances are sought and which we employ to reduce the complexity of the problem. In fact, due to the peculiar differential-drive type of the actuation and the built-in flexibility at the joint level, as the SoftHand grasps, a synergistic path is followed that allows adaptation to objects of various shapes using only a single motor [62]. This explains why, differently from other definitions based on contact point positions, in the current study palm poses in $SE(3)$ are defined as grasp affordances.

The adaptive hand behavior, entangled with the complex dynamical interactions between the hand phalanxes and the objects — the object are not fixated during the grasp acquisition — simply cannot be modelled using the physics simulators cited above, as the complex routing of a single tendon through all the limbs is, to be best of our knowledge, a fundamental feature which is out of the above physics engines' capabilities.

We modelled the Pisa/IIT SoftHand in the multipurpose and physically accurate MBS environment ADAMS [63]. In a first phase, candidate palm poses for the hand are generated by a procedure that employs random samples in the point cloud representation of the object, and selected based on their ability of presenting chances of grasp success (e.g., poses where the hand early interpenetrates the object or the object is ejected during hand closing are discarded). In a second phase, only the grasps robust with respect to perturbation wrenches on the object are considered successful and marked as feasible grasp poses. Additional local grasp candidates are generated around successful grasp and the evaluate process is repeated.

Despite the relatively naïve procedure, the mechanical intelligence embedded in the hand design allows to obtain a fairly satisfactory number of robust grasps, even for objects of complex shapes. Early experimental results confirm the numerical simulation outcomes. Grasp affordances automatically found by the presented procedure will be employed in a pipeline for automatic object recognition, grasping and bi-manual manipulation. The technique is described in detail in the attached technical report (Annex 2.4).

1.2.5 Case-based Reasoning for Grasp Synthesis

Case-Based Reasoning (CBR) means using old experiences to understand and solve new problems [70]. In CBR, a reasoner remembers a previous situation similar to the current one and uses that to solve the new problem. Generally, when a problem will be presented to a case-based reasoner, it searches in its memory of past cases (the case base) and attempts to find a case that has the same problem specification as the current case. If the reasoner cannot find an identical case in its case base, it will attempt to find the case or cases most similar to the current case in the case base. If an identical case is found, its solution is returned as the current solution of the problem. On the other hand, if the retrieved case is not identical to the current case, an adaptation phase occurs. In this phase, the differences between the current and retrieved case are taken into account to adapt the solution associated with the retrieved case. The adapted solution is then returned. The main advantage of a CBR system is that it does not require a lot of specific information on the problem domain: A CBR system can be created with a small amount of experience and incrementally developed, adding more cases to the case base, as they become available. Indeed it offers several benefits such as the facilitation of

the knowledge acquisition process, the capability of learning from past experience and the avoidance of the complete understanding of the domain.

In this work we have applied CBR to the problem of selecting grasps for presented objects. The objects are represented as point clouds and the grasps are parameterised using information from human grasps captured using a thimble glove from work in WP2. The work is described in detail in a technical report (Annex 2.5).

1.2.6 Learning Dexterous Grasps That Generalise To Novel Objects By Combining Hand And Contact Models

There has been progress in learning generalisable grasps. One class of approaches utilises the shape of common object parts to generalise grasps across object categories [82, 83, 84]. This works well for low DoF hands. Another class of approaches captures the global properties of the hand shape either at the point of grasping, or during the approach [85]. This global hand shape can additionally be associated with global object shape, allowing generalisation by warping grasps to match warps of global object shape [86]. This second class works well for high DoF hands, but generalisation is more limited. We have developed a method that achieves the advantages of both classes, generalising grasps across object categories with high DoF hands.

In Annex 2.6 (to appear in ICRA 2014 as [87]) we present a method that infers grasps for novel objects from as little as one training example. It generalises a human chosen grasp type (e.g. rim, pinch) trained from one or two examples, to a novel object of potentially quite different shape. We demonstrate learning and transfer of six grasps types to 31 different grasps of 18 objects, and for two different dexterous hands. The results show the ability to generalise a grasp within and across object categories, and with full or partial shape information at either testing or training stage. The method requires no knowledge of the human defined object category (e.g. cup, box) when learning or performing transfer. Our main technical innovation to achieve this is to learn two types of models from the example grasp, and then recombine them when inferring a new grasp. Both models are probability density functions. The first is a contact model of the relation between a finger part and the local object shape at its contact. We learn one such model for each rigid link in the hand. Each model can generate many possible independent placements for its link on a new query object. Given these models the next problem is to find a hand configuration that combines a good subset of placements in a kinematically feasible grasp. To address this we learn a second type of model, a hand configuration model from the example grasp.

We then use this hand configuration model to constrain the combined search space for the link placements. Given these two learned models we can infer grasps for

novel objects. When presented with a novel query object we use a Monte Carlo procedure to construct a third type of density function (called the query density) over possible contact points on the new object. We again construct a query density for every link of the hand. We then use these densities in two ways to select the new grasp. First we pick a link, and draw a contact point for it on the object from the query density. Then we sample a hand configuration to obtain the complete set of link poses. The whole solution is then refined using simulated annealing on the product of all the query densities and the hand configuration density.

More details on this new method and on the achieved results can be found in the attached paper (Annex 2.6, to appear in ICRA 2014 [87]).

1.3 Relation to the state-of-the-art

The relation has been discussed in each Section, where the detailed description of the activity performed has been given.

References

- [1] C.-H. Tsang, S. Kwong, and H. Wang, “Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection,” *Pattern Recognition*, vol. 40, no. 9, pp. 2373–2391, sep 2007.
- [2] M. Santello, M. Flanders, and J. F. Soechting, “Postural hand synergies for tool use,” *The Journal of Neuroscience*, 1998.
- [3] M. G. Catalano, G. Grioli, A. Serio, E. Farnioli, C. Piazza, and A. Bicchi, “Adaptive synergies for a humanoid robot hand.” in *Humanoids*. IEEE, pp. 7–14.
- [4] A. Bicchi, M. Gabiccini, and M. Santello, “Modelling natural and artificial hands with synergies,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 366, no. 1581, pp. 3153–3161, 2011.
- [5] M. Gabiccini, A. Bicchi, D. Prattichizzo, and M. Malvezzi, “On the role of hand synergies in the optimal choice of grasping forces,” *Autonomous Robots*, vol. 31, no. 2-3, pp. 235–252, 2011.
- [6] M. Ciocarlie, C. Goldfeder, and P. Allen, “Dexterous Grasping via Eigen-grasps: A Low-Dimensional Approach to a High-Complexity Problem,” 2007.
- [7] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, “The columbia grasp database,” in *IEEE Intl. Conf. on Robotics and Automation*, 2009.

- [8] A. A. N. G. T. J. L. M. Gabbicini and A. Bicchi, “Natural redundancy resolution in dual-arm manipulation using conguration dependent stiffness (cds) control,” in *to appear in IEEE International Conference on Robotics and Automation*, 2014.
- [9] A. Bicchi, *Hands for Dexterous Manipulation and Powerful Grasping: A Difficult Road Towards Simplicity*, G. Giralt and G. Hirzinger, Eds. Springer London, 1996.
- [10] M. Gabbicini, E. Farnioli, and A. Bicchi, “Grasp analysis tools for synergistic underactuated robotic hands,” *International Journal of Robotic Research*, vol. 32, no. 13, pp. 1553–1576, 2013.
- [11] I. Jolliffe, “Principal component analysis and factor analysis,” *Principal Component Analysis*, pp. 150–166, 2002.
- [12] A. Hyvärinen, J. Karhunen, and E. Oja, “What is independent component analysis?” *Independent Component Analysis*, pp. 145–164, 2001.
- [13] J. O. Ramsay and B. W. Silverman, *Applied functional data analysis: methods and case studies*. Springer New York, 2002, vol. 77.
- [14] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [15] N. D. Lawrence, “Gaussian process latent variable models for visualisation of high dimensional data,” *Advances in neural information processing systems*, vol. 16, no. 329-336, p. 3, 2004.
- [16] M. Peternel and A. Leonardis, “Visual learning and recognition of a probabilistic spatio-temporal model of cyclic human locomotion,” in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4. IEEE, 2004, pp. 146–149.
- [17] K. Bernardin, K. Ogawara, K. Ikeuchi, and R. Dillmann, “A hidden markov model based sensor fusion approach for recognizing continuous human grasping sequences,” in *Proc. 3rd IEEE International Conference on Humanoid Robots*, 2003, pp. 1–13.
- [18] M. Santello, M. Flanders, and J. F. Soechting, “Postural hand synergies for tool use,” *The Journal of Neuroscience*, vol. 18, no. 23, pp. 10 105–10 115, 1998.
- [19] R. Vinjamuri, M. Sun, C.-C. Chang, H.-N. Lee, R. J. Sclabassi, and Z.-H. Mao, “Temporal postural synergies of the hand in rapid grasping tasks,” *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 4, pp. 986–994, 2010.

- [20] —, “Dimensionality reduction in control and coordination of the human hand,” *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 2, pp. 284–295, 2010.
- [21] J. R. Napier, “The prehensile movements of the human hand,” *Journal of bone and Joint surgery*, vol. 38, no. 4, pp. 902–913, 1956.
- [22] M. Cutkosky, “On grasp choice, grasp models, and the design of hands for manufacturing tasks,” *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 3, pp. 269–279, 1989.
- [23] I. Bullock, R. Ma, and A. Dollar, “A hand-centric classification of human and robot dexterous manipulation,” *Haptics, IEEE Transactions on*, vol. 6, no. 2, pp. 129–144, 2013.
- [24] I. Bullock, J. Zheng, S. De La Rosa, C. Guertler, and A. Dollar, “Grasp frequency and usage in daily household and machine shop tasks,” *Haptics, IEEE Transactions on*, vol. 6, no. 3, pp. 296–308, 2013.
- [25] L. Y. Chang, N. S. Pollard, T. M. Mitchell, and E. P. Xing, “Feature selection for grasp recognition from optical markers,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 2944–2950.
- [26] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard, “Segmenting motion capture data into distinct behaviors,” in *Proceedings of Graphics Interface 2004*. Canadian Human-Computer Communications Society, 2004, pp. 185–194.
- [27] M. Bianchi and M. V. Liarokapis, “HandCorpus, a new open-access repository for sharing experimental data and results on human and artificial hands,” in *IEEE World Haptics Conference (WHC)*, April 2013.
- [28] A. Leonardis, H. Bischof, and J. Maver, “Multiple eigenspaces,” *Pattern Recognition*, vol. 35, no. 11, pp. 2613–2627, 2002.
- [29] J. A. Hartigan, *Clustering algorithms*. John Wiley & Sons, Inc., 1975.
- [30] P. Kingston and M. Egerstedt, “Time and output warping of control systems: Comparing and imitating motions,” *Automatica*, vol. 47, no. 8, pp. 1580–1588, 2011.
- [31] S. Calinon, “Robot programming by demonstration,” in *Springer Handbook of Robotics*. Springer, 2008, pp. 1371–1394.
- [32] M. Gabiccini, G. Stillfried, H. Marino, and M. Bianchi, “A data-driven kinematic model of the human hand with soft-tissue artifact compensation mechanism for grasp synergy analysis,” in *Intelligent Robots and Systems, 2013. Proceedings., 2013 IEEE/RSJ International Conference on*. IEEE, 2013.

- [33] “The princeton shape benchmark,” in *Proceedings of the Shape Modeling International 2004*, ser. SMI '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 167–178. [Online]. Available: <http://dx.doi.org/10.1109/SMI.2004.63>
- [34] A. Miller, P. Allen, V. Santos, and F. Valero-cuevas, “From robot hands to human hands: A visualization and simulation engine for grasping research,” *Industrial Robot*, vol. 32, pp. 55–63, 2005.
- [35] I. Yildirim and R. A. Jacobs, “Transfer of object category knowledge across visual and haptic modalities: Experimental and computational studies,” *Cognition*, vol. 126, no. 2, pp. 135 – 148, 2013.
- [36] A. Bicchi, J. K. Salisbury, and D. L. Brock, “Contact sensing from force measurements,” *International Journal of Robotics Research*, vol. 12, pp. 249–262, 1993.
- [37] E. Battaglia, G. Grioli, M. G. Catalano, M. Bianchi, A. Serio, M. Santello, and A. Bicchi, “Thimblesense: A new wearable tactile device for human and robotic fingers,” in *Haptics Symposium*, 2014.
- [38] E. Battaglia, G. Grioli, M. G. Catalano, M. Santello, and A. Bicchi, “Thimble Sense: an individual-digit wearable tactile sensor for experimental grasp studies,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2014.
- [39] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009.
- [40] D. W. E. A. Lorusso and R. B. Fisher, “Estimating 3-d rigid body transformations: A comparison of four major algorithms,” *Mach. Vision Appl.*, vol. 9, pp. 272–290, 1997.
- [41] E. A. Wan and R. Van Der Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000.* IEEE, 2000, pp. 153–158.
- [42] D. Teney and J. Piater, “Probabilistic object models for pose estimation in 2d images,” pp. 336–345, 2011.
- [43] A. Serio, E. Riccomini, V. Tartaglia, I. Sarakoglou, M. Gabiccini, N. Tsagarakis, and A. Bicchi, “The patched intrinsic tactile object: a tool to investigate human grasps,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Chicago, Illinois, 11/2014 2014.

- [44] C. Ferrari and J. Canny, “Planning optimal grasps,” *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pp. 2290–2295, Jan. 1992.
- [45] D. Prattichizzo, J. Salisbury, and A. Bicchi, “Contact and grasp robustness measures: Analysis and experiments,” *Experimental Robotics IV*, Jan. 1997.
- [46] C. Rosales, R. Suarez, M. Gabbicini, and A. Bicchi, “On the synthesis of feasible and prehensile robotic grasps,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 1–7.
- [47] M. Teichmann and B. Mishra, “Reactive algorithms for grasping using a modified parallel jaw gripper,” in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, 1994, pp. 1–6.
- [48] R. Platt, A. H. Fagg, R. A. Grupen, W. Bluethmann, R. Ambrose, M. Diftler, E. Huber, A. Fagg, M. Rosenstein, R. Grupen, C. Breazeal, A. Brooks, A. Lockerd, R. A. Peters, O. C. Jenkins, M. Mataric, and M. Bugajska, “Null-Space Grasp Control: Theory and Experiments,” *Robotics, IEEE Transactions on*, no. 2, pp. 1–14, 2010.
- [49] D. Wang, B. Watson, and A. Fagg, “A switching control approach to haptic exploration for quality grasps,” in *Proceedings of the Workshop on Robot*, Jan. 2007, pp. 1–8.
- [50] K. Hsiao, S. Chitta, M. Ciocarlie, and E. G. Jones, “Contact-reactive grasping of objects with partial shape information,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 1–8.
- [51] E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A. Y. Ng, and O. Khatib, “Grasping with application to an autonomous checkout robot,” *2011 IEEE International Conference on Robotics and Automation*, pp. 2837–2844, May 2011.
- [52] F. T. Pokorny, J. A. Stork, and D. Kragic, “Grasping objects with holes: A topological approach,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1100–1107, 2012.
- [53] L. Montesano, M. Lopes, A. Bernardino, J. Santos-Victor, F. S. Melo, and R. Martinez-Cantin, “Learning grasping affordances from local visual descriptors,” in *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on*, 2009, pp. 1–6.
- [54] R. Detry, E. Bařeski, M. Popović, and Y. Touati, “Learning continuous grasp affordances by sensorimotor exploration,” *From motor learning to*, Jan. 2010.
- [55] J. Bohg and D. Kragic, “Learning grasping points with shape context,” *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 362–377, Apr. 2010.

- [56] A. Saxena, J. Driemeyer, and A. Y. Ng, “Robotic Grasping of Novel Objects using Vision,” *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, Feb. 2008.
- [57] A. Bierbaum and M. Rambow, “Grasp affordances from multi-fingered tactile exploration using dynamic potential fields,” *Humanoid Robots*, Jan. 2009.
- [58] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, J. Bohg, T. Asfour, and S. Schaal, “Learning of grasp selection based on shape-templates,” *Autonomous Robots*, vol. 36, no. 1-2, pp. 51–65, 2013.
- [59] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, “Automatic grasp planning using shape primitives,” *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, pp. 1824–1829, Jan. 2003.
- [60] R. Diankov and J. Kuffner, “Openrave: A planning architecture for autonomous robotics,” pp. 1–18, Jan. 2008.
- [61] B. León, S. Ulbrich, R. Diankov, G. Puche, M. Przybylski, A. Morales, T. Asfour, S. Moio, J. Bohg, J. Kuffner, and R. Dillmann, *OpenGRASP: A Toolkit for Robot Grasping Simulation*. Springer Berlin Heidelberg, Jan. 2010.
- [62] M. Catalano, G. Giorgio, A. Serio, and E. Farnioli, “Adaptive synergies for a humanoid robot hand,” *on humanoid robots*, pp. 1–8, Jan. 2012.
- [63] M. ADAMS. (2014, Jan.) Msc software last accessed. [Online]. Available: <http://web.mscsoftware.com/>
- [64] B. Hillberry and A. Hall Jr, “Rolling contact joint,” Jan. 13 1976, uS Patent 3,932,045.
- [65] M. G. Catalano, G. Grioli, A. Serio, E. Farnioli, C. Piazza, and A. Bicchi, “Adaptive synergies for a humanoid robot hand,” in *IEEE-RAS International Conference on Humanoid Robots*, Osaka, Japan, 2012.
- [66] M. G. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi, “Adaptive synergies for the design and control of the pisa/iit soft hand,” *International Journal of Robotics Research*, Submitted.
- [67] J. Shigley and J. Uicker, *Theory of machines and mechanisms*, ser. McGraw-Hill series in mechanical engineering. McGraw-Hill, 1980.
- [68] L.-W. Tsai, *Robot Analysis and Design: The Mechanics of Serial and Parallel Manipulators*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1999.
- [69] M. Software, *Getting Started Using Adams/View*, MSC Software Corporation, 2012, software available at <http://www.mscsoftware.com/product/adams>. [Online]. Available: <http://www.mscsoftware.com/product/adams>

- [70] J. L. Kolodner, “An introduction to case-based reasoning,” *Artificial Intelligence Review*, vol. 6, no. 1, pp. 3–34, 1992.
- [71] R. Rusu and S. Cousins, “3d is here: Point cloud library (pcl),” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1–4.
- [72] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [73] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, “Towards 3d point cloud based object maps for household environments,” *Robot. Auton. Syst.*, vol. 56, no. 11, pp. 927–941, 2008.
- [74] J. Berkmann and T. Caelli, “Computation of surface geometry and segmentation using covariance techniques,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 11, pp. 1114–1116, 1994.
- [75] N. B. Radu Bogdan Rusu, Zoltan Csaba Marton and M. Beetz, “Learning informative point classes for the acquisition of object model maps,” in *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Hanoi, Vietnam, 2008.
- [76] M. B. Radu Bogdan Rusu, Nico Blodow, “Fast point feature histograms (fpfh) for 3d registration,” *Intelligent Autonomous System*, 2009, <http://files.rbrusu.com/publications/Rusu09ICRA.pdf>.
- [77] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, “Fast 3d recognition and pose using the viewpoint feature histogram,” in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 10/2010 2010.
- [78] M. V. W. Wohlkinger, “Ensemble of shape functions for 3d object classification,” in *Robotics and Biomimetics (ROBIO), IEEE International Conference on*, 2011.
- [79] F. Tombari, S. Salti, and L. Di Stefano, “Unique signatures of histograms for local surface description,” in *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*, ser. ECCV’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 356–369.
- [80] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinhellefort, and M. Beetz, “General 3D Modelling of Novel Objects from a Single View,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 18-22 2010.
- [81] K. Vehkalahti, “The concise encyclopedia of statistics by yadolah dodge,” *International Statistical Review*, vol. 76, no. 3, pp. 460–461, 2008.

- [82] R. Detry, C. H. Ek, M. Madry, and D. Kragic, “Learning a dictionary of prototypical grasp-predicting parts from grasping experience.” in *IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 601–608.
- [83] A. H. P. P. M. K. L. R. T. Asfour and S. Schaal, “Template-based learning of grasp selection.” in *IEEE International Conference on Robotics and Automation*. IEEE, 2012.
- [84] K. E. U. E. Oztop and J. Peters, “A kernel-based approach to direct action perception.” in *IEEE International Conference on Robotics and Automation*. IEEE, 2012.
- [85] H. B. Amor, O. Kroemer, U. Hillenbrand, G. Neumann, and J. Peters, “Generalization of human grasping for multi-fingered robot hands.” in *IROS*. IEEE, 2012, pp. 2043–2050.
- [86] U. Hillenbrand and M. A. Roa, “Transferring functional grasps through contact warping and local replanning.” in *IROS*. IEEE, 2012, pp. 2963–2970.
- [87] M. Kopicki, R. Detry, F. Schmidt, C. Borst, R. Stolkin, and J. L. Wyatt, “Learning dexterous grasps that generalise to novel objects by combining hand and contact models,” in *to appear in IEEE International Conference on Robotics and Automation*, 2014.

2 Annexes

2.1 Natural Redundancy Resolution in Dual-Arm Manipulation using Conguration Dependent Stiffness (CDS) Control

Paper title: Natural Redundancy Resolution in Dual-Arm Manipulation using Conguration Dependent Stiffness (CDS) Control

Paper authors: A. Ajoudani, N. G. Tsagarakis, J. Lee, M. Gabbicini and A. Bicchi

Publication info: to appear in *2014 IEEE International Conference on Robotics and Automation (ICRA)*,

Abstract: Incorporation of human motor control principles in the motion control architectures for humanoid robots or assistive and prosthesis devices will permit these systems not only to look anthropomorphic and natural at the body ware level but also to generate natural motion proles resembling those executed by humans during manipulation and locomotion. In this work, relying on the observations on human bimanual coordination, a novel realtime motion control strategy is proposed to regulate the desired Cartesian stiffness prole during the execution of bimanual tasks. The novelty of the proposed control scheme relies on the use of common mode stiffness(CMS) and conguration dependent stiffness (CDS) to regulate the size and directionality of the task space stiffness ellipsoid. Thanks to the CDS control, the proposed scheme is not only proved to be effective in regulating the desired stiffness ellipsoid but also permits to resolve the manipulator redundancy in a manner that results in motions exhibiting similarities with those performed by human during the execution of bimanual tasks. The effectiveness of the controller is evaluated in an experimental setup in which two cooperating robotic arms are executing an assembly task. Experimental results demonstrate that the proposed dual-arm CDS-CMS controller is effective in tracking the desired stiffness ellipsoids as well as in producing human-like natural motions for the two robotic arms.

Relation with the deliverable: Exploit stiffness control for performing dual-arm manipulation tasks under uncertain conditions.

2.2 Grasp analysis tools for synergistic underactuated robotic hands

Paper title: Grasp analysis tools for synergistic underactuated robotic hands

Paper authors: M. Gabiccini, E. Farnioli, and A. Bicchi.

Publication info: *International Journal of Robotic Research*, 2013, 32(13), pp.(1553-1576)

Abstract: Despite being a classical topic in robotics, the research on dexterous robotic hands still stirs a lively research activity. The current interest is especially attracted by underactuated robotic hands where a high number of degrees of freedom (DoFs), and a relatively low number of degrees of actuation co-exist. The correlation between the DoFs obtained through a wise distribution of actuators is aimed at simplifying the control with a minimal loss of dexterity. In this sense, the application of bio-inspired principles is bringing research toward a more conscious design. This work proposes new, general approaches for the analysis of grasps with synergistic underactuated robotic hands. After a review of the quasi-static equations describing the system, where contact preload is also considered, two different approaches to the analysis are presented. The first one is based on a systematic combination of the equations. The independent and the dependent variables are denoted, and cause-effect relationships between them are found. In addition, remarkable properties of the grasp, as the subspace of controllable internal force and the grasp compliance, are worked out in symbolic form. Then, some relevant kinds of tasks, such as pure squeeze, spurious squeeze and kinematic grasp displacements, are denoted, in terms of nullity or non-nullity of proper variables. The second method of analysis shows how to discover the feasibility of the pre-denoted tasks, operating a systematic decomposition of the solution space of the system. As a result, the inputs to be given to the hand, in order to achieve the desired system displacements, are found. The study of the feasible variations is carried out arriving at the discovery of all the combinations of nullity and/or non-nullity variables which are allowed by the equations describing the system. Numerical results are presented both for precision and power grasps, finding forces and displacements that the hand can impose on the object, and showing which properties are preserved after the introduction of a synergistic underactuation mechanism.

Relation with the deliverable: Quasi-analysis of synergistically underactuated grasps, fundamental for modelling the grasps realized with underactuated hands.

2.3 Grasp Movement Classification via Multiple Eigenspaces

Paper title: Grasp Movement Classification via Multiple Eigenspaces

Paper authors: H. Marino, M. Gabiccini, and A. Leonardis.

Publication info: Technical Report

Abstract: Dimensionality reduction techniques have extensively been used in the field of robotic grasping, under many different forms, to obtain simple instructions to pilot robotic hands to mimic what human hands do. Human grasp analysis is becoming increasingly automated. Nonetheless, there is still a gap which needs to be bridged when analyzing this sort of data, namely the classification of the type of grasp performed which is still a task mostly performed via direct inspection. There is a large number of models and techniques which are useful for dimensionality reduction, most popularly PCA. However, the results of such global dimensionality reduction techniques can be highly dependent on the dataset used. To avoid this dependency, we have used the Multiple Eigenspace technique [28]. This approach gives lower dimensional subspaces than the ones given by PCA. These subspaces can be used for interpolation, thus utilising the low dimensional local structure of the data, which is instead usually ignored. We describe the results of analysing more 4500 individual grasp postures.

Relation with the deliverable: Acquisition of grasp synergies from human grasp examples is one of the ways that we are selecting grasp primitives in a soft-synergy framework.

2.4 Synthetic Grasp Affordances for the Pisa/IIT SoftHand

Paper title: Synthetic Grasp Affordances for the Pisa/IIT SoftHand

Paper authors: M. Gabiccini, and M. Antonelli.

Publication info: Technical Report

Abstract: In this technical report we describe work we have done to automatically create a database of grasp affordances for the Pisa/IIT SoftHand for a set of kitchenware objects. The method itself is not new since similar approaches to the grasp planning problem have been presented in the past by employing simulators. Instead the novelty arises from the special hand design for which the grasp affordances are sought, and which is employed to reduce the complexity of the problem. Due to the differential-drive type of the actuation and the built in flexibility at the joint level, as the SoftHand grasps it adapts to the object. This allows us to define grasp affordances simply as initial hand poses relative to the object. We modelled the hand using the ADAMS simulator. Candidate palm poses for the hand are generated by a randomised procedure, and selected based on the chance of grasp success. Despite the relatively naive procedure the hand design allows a satisfactory number of grasps even for objects of complex shape.

Relation with the deliverable: This technique is later able to be used as the basis of techniques for selecting soft synergy grasps based on a library of grasps for known objects.

2.5 Case Based Reasoning for Grasp Synthesis

Paper title:Case Based Reasoning for Grasp Synthesis

Paper authors: M. Antonelli and M. Gabiccini

Publication info: Technical Report

Abstract: In this paper we describe how to build a Case Based Reasoning (CBR) system able to find, for a given pair (hand, object), the suitable grasp configuration among the set of feasible grasp candidates. Generally, each case is represented as two sets of attributes: the former specifies the problem and the latter the solution. In particular, the problem is described by the features of the point cloud of a scene and the solution by the parameters of the grasp. The data for the case based system is derived from work on example grasps gathered using methods involving a thimble glove in WP2. The parameters are the contact points, i.e., the position of the thimbles with respect to the object, the position of the wrist with respect to the object, the contact forces applied to the contact points.

Relation with the deliverable: The application of CBR to grasp selection after examples from humans allows selection of grasps for novel objects that can be executed with a dextrous hand.

2.6 Learning Dexterous Grasps That Generalise To Novel Objects By Combining Hand And Contact Models

Paper title: Learning Dexterous Grasps That Generalise To Novel Objects By Combining Hand And Contact Models

Paper authors: M. Kopicki, R. Detry, F. Schmidt, C. Borst, R. Stolkin, J.L. Wyatt

Publication info: to appear in *2014 IEEE International Conference on Robotics and Automation (ICRA)*

Abstract: Generalizing dexterous grasps to novel objects is an open problem. We show how to learn grasps for high DoF hands that generalize to novel objects, given as little as one demonstrated grasp. During grasp learning two types of probability density are learned that model the demonstrated grasp. The first density type (the contact model) models the relationship of an individual finger part to local surface features at its contact point. The second density type (the hand configuration model) models the whole hand configuration during the approach to grasp. When presented with a new object, many candidate grasps are generated, and a kinematically feasible grasp is selected that maximizes the product of these densities. We demonstrate 31 successful grasps on novel objects (an 86% success rate), transferred from 16 training grasps. The method enables: transfer of dexterous grasps within object categories; across object categories; to and from objects where there is no complete model of the object available; and using two different dexterous hands.

Relation with the deliverable: A new method is proposed for learning grasps and generalize them to novel objects.

Natural Redundancy Resolution in Dual-Arm Manipulation using Configuration Dependent Stiffness (CDS) Control

Arash Ajoudani, Nikos G. Tsagarakis, Jinoh Lee, Marco Gabiccini and Antonio Bicchi

Abstract—Incorporation of human motor control principles in the motion control architectures for humanoid robots or assistive and prosthesis devices will permit these systems not only to look anthropomorphic and natural at the body ware level but also to generate natural motion profiles resembling those executed by humans during manipulation and locomotion. In this work, relying on the observations on human bimanual coordination, a novel realtime motion control strategy is proposed to regulate the desired Cartesian stiffness profile during the execution of bimanual tasks. The novelty of the proposed control scheme relies on the use of common mode stiffness (CMS) and configuration dependent stiffness (CDS) to regulate the size and directionality of the task space stiffness ellipsoid. Thanks to the CDS control, the proposed scheme is not only proved to be effective in regulating the desired stiffness ellipsoid but also permits to resolve the manipulator redundancy in a manner that results in motions exhibiting similarities with those performed by human during the execution of bimanual tasks. The effectiveness of the controller is evaluated in an experimental setup in which two cooperating robotic arms are executing an assembly task. Experimental results demonstrate that the proposed dual-arm CDS-CMS controller is effective in tracking the desired stiffness ellipsoids as well as in producing human-like natural motions for the two robotic arms.

I. INTRODUCTION

In recent years the fast growing interest in versatility and flexibility of robotic systems working closely and interacting with humans in co-operative tasks or acting as assisting or prosthesis systems had led to the development of a wide range of systems: from full body humanoid robot co-workers to anthropomorphic manipulator prosthesis and exoskeleton systems aiming to aid and improve the life of humans with special needs. Apart from resembling the human body in terms of kinematics and physical appearance, and to increase their acceptability and compatibility, such systems should be also capable of generating motions that look natural and demonstrate similarities with those executed by humans. To achieve this, it is profitable therefore to incorporate in the motion control architecture of these systems human principles of motor control [1].

The goal is to obtain a good performance, while rendering a human-like natural motion, which is deemed to emerge as a consequence of the stabilization of some task-related criteria. To illustrate the underlying concept, examples of

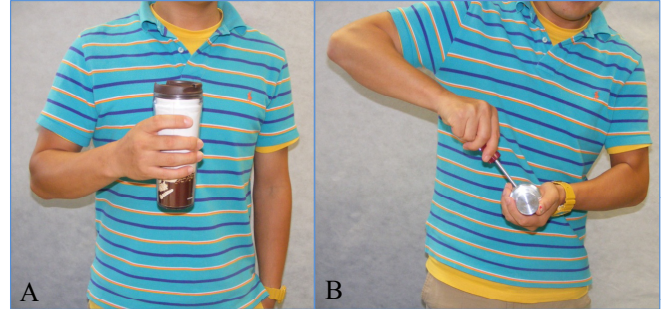


Fig. 1: Examples of acquired natural postures in humans as a consequence to the redundancy resolution for (A) minimum effort and (B) impedance control.

natural postures of the human while holding a cup or using a screwdriver are depicted in Fig. 1. A and B, respectively. In these particular examples, subtask criteria are rooted in the concepts of minimum effort (1. A) and impedance control (1. B), which lead to the generation of corresponding configurations. Inspired by similar observations, the problem of redundancy resolution has been extensively addressed in the field of robotics by establishing task-related artificial potential fields [2], [3], [4], while, only few studies have drawn attention to the acquired human-like natural motion [5], [1], [6].

With the advent of higher demands on dual-arm manipulation, interaction with the environment using impedance control [7] became the focus of many studies in dual-arm systems. In this direction, object-level impedance control of cooperating robots has gained great attention due to the feasibility of controlling task forces/velocities while avoiding the occurrence of high internal forces [8]. In [9], an internal force based impedance controller has been proposed and its effectiveness was evaluated in an assembly task in a planar setup. Following that, Bonitz et al. [10] extended the aforementioned concept to a condition in which the dynamics of the object is not known. In [11], a decentralized impedance controller is proposed and its contact efficiency has been evaluated in an industrial setup, where two cooperating manipulators carry a common object. The authors in [12], [8] have presented object-level impedance controllers with coupling stiffness which is defined by a potential function. Even though such well-established techniques have successfully demonstrated their effectiveness through reliable handling of the contact, only few have highlighted the efficiency of the

Arash Ajoudani, Marco Gabiccini and Antonio Bicchi are with Interdepartmental Research Centre "E. Piaggio", Faculty of Engineering, University of Pisa, and with the Dept. of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego 30, 16163, Genova, Italy (e-mails: arash.ajoudani@iit.it, m.gabiccini@ing.unipi.it and bicchi@centropiaggio.unipi.it).

Jinoh Lee and Nikos G. Tsagarakis are with the Dept. of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego 30, 16163, Genova, Italy (e-mail: {jinoh.lee, nikos.tsagarakis}@iit.it).

Cartesian impedance control through redundancy resolution [13], [14].

In this paper, relying on the major contribution of the arm configuration to effectively regulate the directionality of the realized endpoint stiffness ellipsoid, and its low-cost nature [15], [16], [17], we explore the role of *configuration dependent stiffness* (CDS) control [18], for dual-arm object manipulation. Therefore, the joint variables are controlled in redundant space to realize a task-specific Cartesian stiffness profile. In addition, with the purpose of obtaining a good tracking performance while rendering natural movements, we incorporate the optimality principles in human motor control in the development of proposed impedance controller. In this direction, given the growing body of evidence in support of coordinated stiffening behavior of the arm joints in humans (e.g. see [19], [15], [16]), we utilize the concept of *common mode stiffness* (CMS) control. To establish this concept, we assume a correlated stiffening behavior among the joints of each robot. A realtime optimization technique is then proposed to adjust the CMS and CDS variables and realize a desired Cartesian stiffness profile of the tool, w.r.t the object: While the CMS variations contribute to the modifications of the size of the realized stiffness ellipsoid (analogous to the role of co-contractions in human arm), CDS control will elongate the realized stiffness ellipsoid toward the desired Cartesian profile.

The effectiveness of the proposed controller is evaluated in a dual-arm assembly task. The acquired good tracking performance and similarities between the natural postures of the human operators and the ones realized by the robots illustrate the contact efficiency of the proposed algorithm, while highlighting the advantages of the integration of human motor principles in redundancy resolution of dual-arm manipulation. Therefore, utilization of similar motion control strategies in humanoids co-operating with humans or upper limb assistive exoskeleton or prosthesis devices can result in the generation of more natural motions improving the acceptability and compatibility of such devices.

The rest of the paper is structured as follows; section II describes the development of the proposed controller. The experimental setup and results of the implementation of the controller for a dual-arm assembly task are presented in sections III and IV respectively. Finally, section V addresses the conclusions.

II. CONTROLLER DESIGN

A. Dual-arm kinematics

Observations on human bimanual coordination suggest that the central nervous system (CNS) stabilizes the first synergy of the two cooperating arms to a larger degree (higher control levels) than if it did for control of each arm joints, separately. Indeed, as regards the human dual-arm activities in its most natural way, two hands cooperate in a way that they form a kinematic chain [20]: while the first synergy stabilizes the relative position/orientation of the two hands (as a dominant task requirement), lower hierarchical

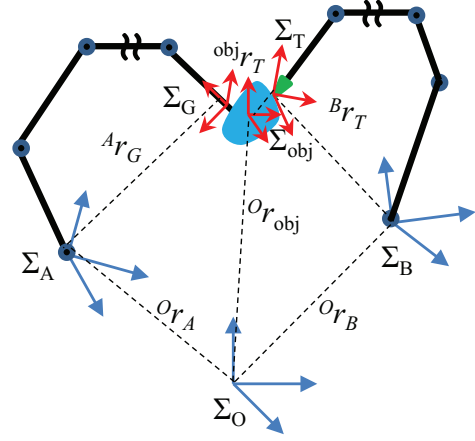


Fig. 2: Dual arm manipulation diagram.

levels of control stabilize remaining task variables by controlling the redundant degrees of freedom [21], [22]. Above observations promote the idea of utilizing a similar kinematic representation in dual-arm robotic manipulation of the object to accommodate the stabilization of the task variables as well as rendering a natural redundancy resolution.

To that end, in our setup, a relative Jacobian is used which creates a unified, coordinated control between the two arms. Normally, the expression of the relative Jacobian combines the individual Jacobians of each arm, such that the resulting Jacobian maps the joint velocities of the two arms to the relative velocity between the end-effectors [23]. This allows users to directly specify the relative trajectory between the two end-effectors such that coordinating the trajectories of the two arms becomes an easy task. Such representation have been incorporated with force-guided control [24] and impedance control [25] as well.

The dual-arm manipulation setup is shown in figure (2). In our setup, $q_G \in \mathbb{R}^{n_A}$ and $q_T \in \mathbb{R}^{n_B}$ denote the joint variables of the manipulators A and B, respectively. Here, we assume that manipulator A is rigidly holding the object, while manipulator B is executing the defined task with the tool. The position vectors of the origins of gripper frame (Σ_G) and the tool frame (Σ_T) with respect to the base frames of the two manipulators (Σ_A and Σ_B) are denoted as ${}^A r_G$ and ${}^B r_T$, respectively. In addition, the position vectors of the origins of the tool and the object frame with respect to Σ_A and Σ_B are denoted as ${}^{\text{obj}} r_T$ and ${}^{\text{obj}} r_G$, respectively. According to [23], the transformation of the joint velocities $\dot{q} = [\dot{q}_G^T \dot{q}_T^T]^T \in \mathbb{R}^n$, onto the task space linear and angular velocities $v = [{}^{\text{obj}} \dot{r}_T \text{ } {}^{\text{obj}} \dot{\omega}_T]^T \in \mathbb{R}^6$ is performed by the relative Jacobian, J_R , as follows

$$v = J_R(q)\dot{q}, \quad (1)$$

with $n = n_A + n_B$, where n_A and n_B denote the number of DoFs of the two manipulators.

B. Impedance control of dual-arm

To establish the mapping between the forces acting on the tool, ${}^{\text{obj}} F_T$, referenced from the object frame, and the

required joint torques, τ , we can exploit the principle of virtual work. Therefore, we can write

$$\begin{aligned}\Delta W_\tau &= \tau^T \Delta q, \\ \Delta W_F &= {}^{\text{obj}}F_T^T \Delta^{\text{obj}}x_T.\end{aligned}\quad (2)$$

with ΔW_τ and ΔW_F , denoting the work done by the joint torques and displacements, and forces acting on the object and relative displacement of the tool w.r.t. object frame of reference, $\Delta^{\text{obj}}x_T$, respectively. By combining the equation (1) and (2), we acquire

$$\tau = J_R^T(q) {}^{\text{obj}}F_T = J_R^T(q) K_c \Delta^{\text{obj}}\tilde{x}_T.$$

with $\Delta^{\text{obj}}\tilde{x}_T = {}^{\text{obj}}x_{Td} - {}^{\text{obj}}x_T$. On the other hand, to establish the stiffness mapping between the joint and Cartesian spaces, we employ the following expression [26],

$$K_J = \frac{\partial \tau}{\partial q} = \frac{\partial (J_R^T(q) K_c \Delta^{\text{obj}}\tilde{x}_T)}{\partial q},$$

The above equation can be written as follows, around the equilibrium position,

$$K_c = [J_R(q)(K_J - K_g)^{-1}J_R(q)^T]^{-1}, \quad (3)$$

where the diagonal elements of the joint stiffness matrix $K_J \in \mathbb{R}^{n \times n}$ are formed by the diagonal elements of the joint stiffness matrices of the two robots (K_{J_G} and K_{J_T}). Here, we assume that all joint stiffness matrices do not have coupling terms, i.e. they are always of diagonal shape. The stiffness matrix K_c specifies the Cartesian stiffness profile which is defined with respect to Σ_G , and $K_g = \frac{\partial J_R^T(q)}{\partial q} {}^{\text{obj}}F_T \in \mathbb{R}^{n \times n}$, captures the effect of geometry in presence of external forces.

A very interesting outcome of the above relation is the definition of the desired Cartesian stiffness (the Cartesian stiffness between Σ_T and Σ_{obj})¹ in relative coordinates. This means that the need for transferring the configuration-dependent, desired Cartesian stiffness matrices of the two robots with respect to the world frame to the task coordinates is simplified, thanks to the definition of the relative Jacobian. Such realization will be profitable for dual-arm asymmetric manipulation tasks using tools (e.g. carving, peeling, peg-in-hole, etc) [20], since the specification of the task parameters in relative coordinates is more intuitive.

To realize the desired Cartesian stiffness matrix (by tracking 21 elements in the symmetric matrix K_{cd}), in absence of preload forces (i.e. $K_g = 0$), we differentiate the Eq. (3) w.r.t. time as follows²:

$$\begin{aligned}\dot{K}_c &= -K_c \left[\frac{dJ_R(q)}{dt} K_J^{-1} J_R(q)^T + J_R(q) \frac{d(K_J^{-1})}{dt} J_R(q)^T \right. \\ &\quad \left. + J_R(q) K_J^{-1} \frac{dJ_R(q)^T}{dt} \right] K_c,\end{aligned}\quad (4)$$

with the first and the last terms of the above equation being the transpose of each other. To that end, if we take

¹since Σ_G is considered rigidly attached to Σ_{obj} , the components of the stiffness matrix are expressed in Σ_{obj} .

²By taking into account that for a square and invertible matrix X , $\partial X^{-1} = -X^{-1}(\partial X)X^{-1}$.

into account that $\dot{q}$ is allowed to vary in the null-space of the relative Jacobian, while complying with the prescribed relative motion of the two end-effectors, we can write

$$\dot{q} = J_R^+ v + N_J \dot{\gamma}, \quad (5)$$

where J_R^+ denotes the pseudo-inverse of the relative Jacobian, $\frac{dJ_R}{dt} = \frac{dJ_R}{dq} \dot{q}$, and $N_J \in \mathbb{R}^{n \times n-6}$ is a basis of the nullspace projector $(I - J_R^+ J_R)$ of the relative Jacobian. $\dot{\gamma} \in \mathbb{R}^{n-6}$ is the vector of nullspace velocities, and $I \in \mathbb{R}^n$ is the identity matrix. Now, by combining equations (5) and (4), and defining

$$\begin{aligned}P_1 &= \frac{dJ_R(q)}{dq} [J_R^+ v] K_J^{-1} J_R(q)^T \\ P_2 &= \frac{dJ_R(q)}{dq} [N_J \dot{\gamma}] K_J^{-1} J_R(q)^T \\ P_3 &= J_R(q) \frac{d(K_J^{-1})}{dt} J_R(q)^T \\ &= J_R(q) \dot{K}_J^{-1} J_R(q)^T,\end{aligned}$$

we can write

$$\dot{K}_c = -K_c [P_1 + P_2 + P_3 + P_1^T + P_2^T] K_c. \quad (6)$$

Now, if we decompose the nullspace velocity vector, we can write

$$\begin{aligned}\dot{\gamma} &= \begin{bmatrix} 1 \\ \bar{0}_{n-7} \end{bmatrix} \dot{\gamma}_1 + \dots + \begin{bmatrix} \bar{0}_{n-7} \\ 1 \end{bmatrix} \dot{\gamma}_{n-6}, \\ &= I_{\gamma_1} \dot{\gamma}_1 + \dots + I_{\gamma_{n-6}} \dot{\gamma}_{n-6}.\end{aligned}\quad (7)$$

Here, relying on the growing body of evidence in support of coordinated stiffening behavior of the arm joints in humans (e.g. see [19], [15], [16]), we explore the concept of common mode stiffness introduced in [18], in our dual-arm setup. Therefore, we assume that the joint stiffness values are controlled in a coordinated manner. To replicate similar behavior, we choose a constant diagonal scaling matrix, K_{scales} , (in descending order from base joint to the end-effector [27]) for each arm while introducing two independently controlled common mode stiffness values, k_{cms_1} and k_{cms_2} . Therefore we write:

$$\dot{K}_J = K_{\text{scales}_1} \dot{k}_{\text{cms}_1} + K_{\text{scales}_2} \dot{k}_{\text{cms}_2}. \quad (8)$$

By combining Eqs. (6), (7) and (8), while factoring w.r.t. the scalar nullspace velocities, $\dot{\gamma}_i$, and common mode stiffness changes, $\dot{k}_{\text{cms}_i}$, we define

$$\begin{aligned}J_{\gamma_i} &= \text{vec} \left\{ \frac{dJ_R(q)}{dq} [N_J I_{k_{j_i}}] J_R(q)^T + \sim^T \right\} \dot{\gamma}_i \\ J_{K_{\text{cms}_i}} &= \text{vec} \{ -J_R(q) K_J^{-1} K_{\text{scales}_i} K_J^{-1} J_R(q)^T + \sim^T \} \dot{k}_{\text{cms}_i}\end{aligned}$$

where the operator vec extracts the 21 independent elements of 6×6 symmetric matrix, and the symbol $\sim$ denotes the same term, placed in the same set of brackets. Now, by defining $\dot{\tilde{K}}_c = \dot{K}_c + K_c [P_1 + P_1^T] K_c$, we can write

$$\text{vec} \{ \dot{\tilde{K}}_c \} = \begin{bmatrix} J_\gamma & J_{K_{\text{cms}}} \end{bmatrix} \begin{bmatrix} \dot{\gamma} \\ \dot{k}_{\text{cms}} \end{bmatrix} =: J_{\dot{\gamma}} \dot{\gamma}. \quad (9)$$

with $J_\gamma = [J_{\gamma_1} \cdots J_{\gamma_{n-6}}]$, $J_{K_{cms}} = [J_{K_{cms_1}} J_{K_{cms_2}}]$, $\dot{\gamma} = [\dot{\gamma}_1 \cdots \dot{\gamma}_{n-6}]^T$, and $\dot{k}_{cms} = [\dot{k}_{cms_1} \dot{k}_{cms_2}]^T$.

The above equation resembles the structure of inverse kinematics problem, in which we are dealing with the tracking of $\dot{K}_c \in \mathbb{R}^{21}$ by using common mode stiffness values, k_{cms} , and degrees of kinematic redundancy in a dual-arm setup. Now, by defining the vectorial stiffness error $e_s = \text{vec}\{K_{cd} - K_c\}$, we can set up classical update laws. Consequently, by exploiting the update law based on the pseudo-inverse of J_y we get

$$\dot{y} = J_y^+ [\text{vec}\{\dot{K}_c\} + K_{ps}e_s], \quad (10)$$

where K_{ps} is the gain associated to the stiffness error e_s . The proposed controller adapts the CMS (k_{cms}) and CDS (γ) related variables to realize a desired Cartesian stiffness profile, while complying with the desired motion of the tool w.r.t. the gripper frame of reference.

C. Task Prioritization

To execute a dual-arm manipulation task, the object must be hold in a reasonable position, inside the workspace, w.r.t. the world frame of reference. This is accompanied by the the desired relative motion of the tool w.r.t. the object frame, and cost-efficient motion of the remaining degrees of freedom in dual-arm kinematic chain. Therefore, the task can be decomposed into subtasks with prioritized order of occurrence.

An efficient solution for stabilization of the task variables in prioritized order is presented in [4]. Following that, to preserve a desired position of the object w.r.t. the world frame, the first priority subtask is established as follow

$$\dot{q}_G = J_1^+ \dot{r}_1, \quad (11)$$

where $J_1 \in \mathbb{R}^{3 \times n_A}$ is the first-priority task Jacobian and $\dot{r}_1 \in \mathbb{R}^3$ is the velocity vector of the origin of the gripper frame³. Our secondary subtask establishes the relative movement of the two end effectors and its kinematic relationship is defined by the relative Jacobian of the two cooperative manipulators (Eq. 1). Now, given the two subtasks, we setup the task-priority based kinematic control [4], as follows⁴

$$\dot{q} = J_1^+ \dot{r}_1 + \hat{J}_2^+ (\dot{r}_2 - J_2 J_1^+ \dot{r}_1) + (I - J_1^+ J_1)(I - \hat{J}_2^+ \hat{J}_2) \dot{\gamma}, \quad (12)$$

where I is the identity matrix, $J_2 = J_R$, $\dot{r}_2 = \dot{v}$, and $\hat{J}_2 = J_2(I - J_2^+ J_2)$. To ensure robustness against kinematic singularities, we use the damped least squares inverse solution which is defined by $B^+ = B^T(BB^T + \lambda I)^{-1}$, with $\lambda \in \mathbb{R}$ denoting the damping factor [28].

The third term in above equation projects the vector $\dot{\gamma}$ (Eq. 9), into a subspace ($C \in \mathbb{R}^{n-9}$, assuming $n > 9$) which is formed by remaining DoF that do not affect neither one of the subtask variables, while, in the meantime decreases the error between the desired and realized relative stiffness profile.

³here we do not consider any constraints on the orientation of the first subtask.

⁴such representation accommodates the incorporation of additional subtasks such as joint and torque limit avoidance and etc.

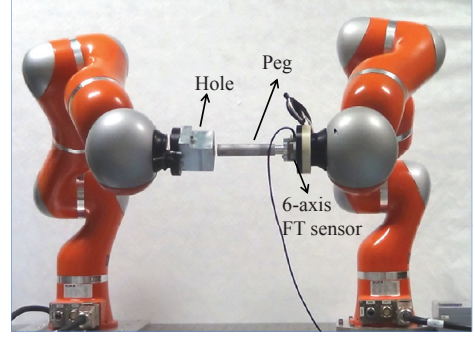


Fig. 3: Dual-Arm assembly (peg-in-hole) setup.

Due to the fact that the proposed algorithm incorporates human impedance regulation principles in realization of the desired Cartesian stiffness profile, we expect to obtain similar performance and natural redundancy resolution as compared with the humans.

The proposed CDS controller can also be utilized for humanoid robots or upper body assistive devices with joints with fixed-passive stiffness values to fairly realize a desired Cartesian stiffness profile w.r.t. the object frame. In addition, once the concept of common mode stiffness is integrated in the hardware⁵, the CDS controller can be adopted to further reduce the error between the realized and desired Cartesian stiffness profiles.

III. EXPERIMENTAL SETUP

The efficiency of the proposed approach to cope with contact stability issues while generating natural motions in redundant space is evaluated in a Peg-in-Hole task, a classical benchmark for spatial planning with uncertainties.

The dual-arm setup incorporated two seven DoF KUKA LWR, with DLR's Fast Research (FR) Interface [29]. A peg and a hole were designed and mounted on the end effectors of the two robots. To illustrate the capabilities of the proposed controller in effective modulations of the size and directionality of the realized endpoint stiffness in task coordinates, the hole was mounted along different (x, y and z) directions of Σ_G . Then, three different desired Cartesian, $K_{c_{des}}$, stiffness profiles (stiff along the direction of the hole, while realizing a compliant profile along other directions to avoid high interaction forces) were defined and tracked by our proposed controller. A 6-axis force and torque (F/T) sensor (ATIMini-45) was mounted in between the peg and the robot end effector to monitor the interaction forces. Performance analysis of the proposed algorithm was carried out considering the realized stiffness matrix error and normalized to the norm of $K_{c_{des}}$ as follows

$$e_K = \frac{\|K_c - K_{c_{des}}\|}{\|K_{c_{des}}\|}.$$

⁵by designing robotic arms with $n+1$ actuators, with n being the number of joints and 1 corresponding the actuator for modifying the stiffness of all joints in a coordinated manner.

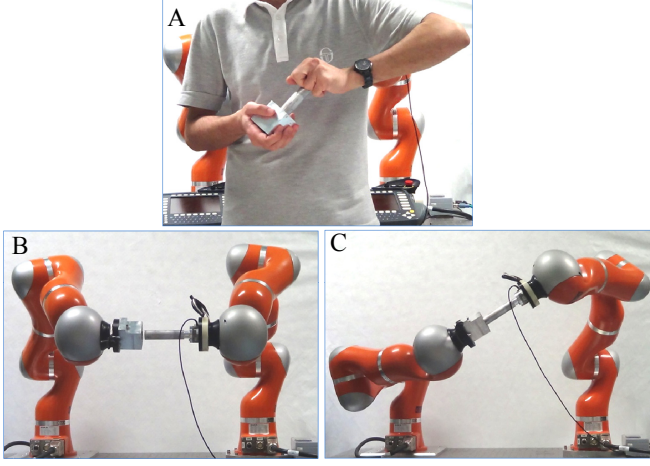


Fig. 4: A. Human performing a bimanual peg-in-hole task. Initial configuration (B), and realized configuration as a result of proposed controller (C). Human and robots' ending configuration coincide which gives evidence to the similarity in control principles.

In addition, to provide a graphical representation of the stiffness profiles, the translational part of the realized and desired stiffness matrices were projected into xy , xz and yz planes of the object frame of reference, Σ_{obj} , resulting in three stiffness matrices ($K_{ij} \in \mathbb{R}^2, [i, j] = [x, y, z]$). Following that, the major, α_{max} , and minor, α_{min} , axis of the ellipsoids along with the major axis direction, ϕ_{ij} , were determined using singular value decomposition:

$$\begin{aligned}\phi_{ij} &= \text{atan2}(U_{\max_j}, U_{\max_i}) \\ \alpha_{\max_{ij}} &= [E_{\max}(K_{ij}^T K_{ij})]^{\frac{1}{2}} \\ \alpha_{\min_{ij}} &= [E_{\min}(K_{ij}^T K_{ij})]^{\frac{1}{2}},\end{aligned}$$

with E denoting the eigenvalue operator and $K_{ij} = U_{ij} S_{ij} V_{ij}^T$.

The joint damping values were all considered fixed $D_i = 0.7\text{Ns/m}$. The control and synchronization interface between the KUKA controllers and the six axis F/T sensor were developed in C++.

IV. RESULTS

Figure 4. A demonstrates the natural posture of the human operator executing an assembly task. As observed here, the redundant DoFs in both arms are adapted to elongate the stiffness profile in the direction of the hole, to overcome the frictional forces, while rendering a compliant behavior in remaining directions to avoid high interaction forces.

The proposed algorithm establishes the realtime tracking of the desired stiffness profile in relative coordinates through the optimization of the CMS and CDS. Starting from an initial configuration (Fig. 4. B), the CDS control drives the redundant joints to a configuration which is depicted in Fig. 4. C.

The desired (red, dashed) and realized (black, solid) ellipsoids are projected into the object's frame of reference and are depicted in Fig. 5. As seen in the plots, efficient elongation of the stiffness ellipsoid is perceived due to

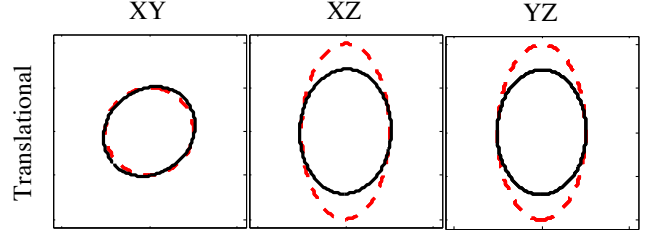


Fig. 5: Desired (red, dashed) and realized (black, solid) translational stiffness ellipsoids, corresponding to configuration C in Fig. 4. In this experiment, the translational stiffness matrix $K_{c_{des}} = [300, 300, 1000]\text{N/m}$, is defined in relative coordinates. All off diagonal elements are set to zero. The hole is mounted along the z axis of Σ_G .

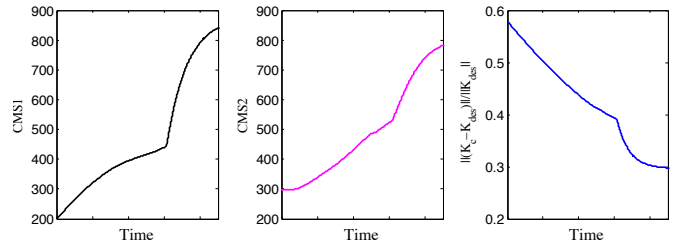


Fig. 6: Common mode stiffness parameters of the two robots (left and middle plots) are adapted in addition to the CDS to decrease the error between the desired and realized stiffness ellipsoids (most right plot), in realtime.

the CDS control. Fig. 6 illustrates the adaptation of the CMS values of the two robots (most left and middle plots), synchronized with CDS control of the dual-arm setup. As a result, the error between the desired and realized stiffness ellipsoids is decreased in realtime (Fig. 6, most right plot).

In another experiment, the hole was mounted along the y axis of Σ_G and the desired stiffness ellipsoid was defined to render a stiff behavior along the direction of the hole, while being compliant along other directions. Starting from an initial configuration depicted in Fig. 7. B, the CDS control drives the redundant joints to a configuration which is depicted in Fig. 7. C. Since the position of the hole is considered fixed, the CDS controller has maximally adjusted the configuration of the manipulator on the left, not to violate the constraints on the position of the hole. At the same time, efficient control of the configuration of the manipulator on the right has been achieved. Similarities between the natural posture of the human operator (Fig. 7. A), and the one realized by the proposed controller, once again, give evidence to the analogy between incorporated control principles. The normalize error between the realized and desired task stiffness profile was $e_K = 27\%$. This error includes all 21 elements of the stiffness matrices. Projected stiffness ellipsoids of the this experiment are brought in Fig. 8.

Fig. 9 illustrates the projected stiffness ellipsoids, as

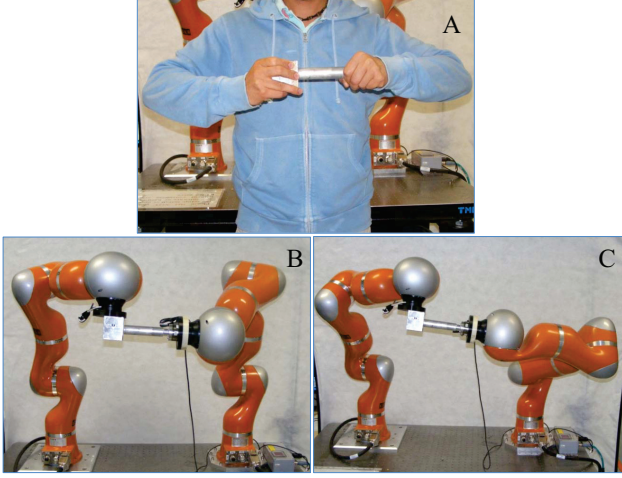


Fig. 7: A. Human performing a bimanual peg-in-hole task. Initial configuration (B), and realized configuration as a result of proposed controller (C). The hole was kept in a fixed position (subtask one), while the orientation was allowed to change due to the CDS. Human and robots' ending configuration coincide which gives evidence to the similarity in control principles.

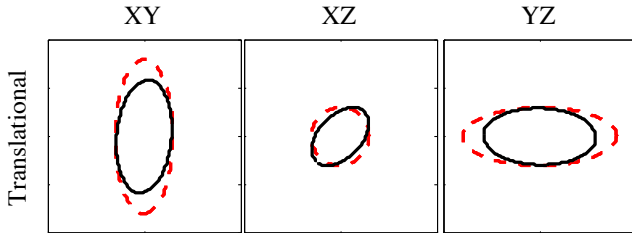


Fig. 8: Desired (red, dashed) and realized (black, solid) translational stiffness ellipsoids, corresponding to configuration C in Fig. 4. In this experiment, the translational stiffness matrix $K_{c_{des}} = [300, 1000, 300] \text{N/m}$, is defined in relative coordinates. All off diagonal elements are set to zero. The hole is mounted along the Y axis of Σ_G .

results of the tracking of different elongation of the desired stiffness ellipsoid in relative coordinates (the hole was mounted along x in this experiment). The CDS and CMS values are adapted to effectively align the realized stiffness ellipsoid with the desired one. This lead to the normalized error of $e_K = 30\%$.

Effective modulations of the size and the directionality of the realized stiffness ellipsoid in above examples illustrate the capabilities of the proposed controller in realization of the desired Cartesian stiffness profile. In addition, our proposed CDS controller achieves a natural redundancy resolution of the dual-arm setup, which highlights its application in control of the humanoids or upper body assistive or prosthesis devices.

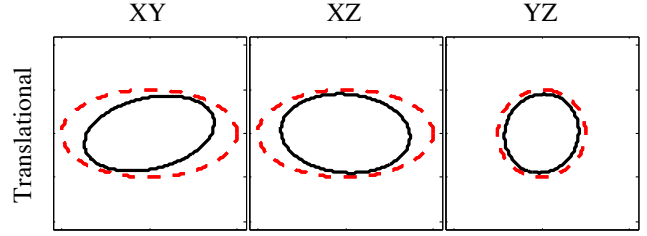


Fig. 9: Desired (red, dashed) and realized (black, solid) translational stiffness ellipsoids with different elongation w.r.t. Fig. 5. In this experiment, the translational stiffness matrix $K_{c_{des}} = [1000, 300, 300] \text{N/m}$, is defined in relative coordinates. All off diagonal elements are set to zero.

V. CONCLUSIONS

The paper proposed a new on-line motion controller which regulates the Cartesian stiffness ellipsoid in a manner inspired by human motor control and impedance regulation principles. The presented scheme not only achieves effective task space stiffness control but also intrinsically perform redundancy resolution of the manipulator arm in a manner that results in the generation of natural motion resembling those executed by human in similar tasks. In particular, given the growing body of evidence on coordinated stiffening behavior of the human arm joints, two common mode stiffness (CMS) controllers were utilized. The CMS concept was introduces in our dual arm controller by changing the joint stiffness values of each robotic arm in a correlated manner. This allows to regulate the size of the realized Cartesian stiffness ellipsoid for each arm. Simultaneously, to align the realized stiffness ellipsoid with the desired one, we explored the concept of configuration dependent stiffness (CDS) in dual arm impedance control. Effectiveness of the proposed algorithm in efficient modifications of the realized size and directionality of the task stiffness ellipsoid, while generating natural movements, was evaluated in three assembly experiments. Implementation of the CMS-CDS controller resulted in good tracking performance, in addition to the generation of natural movements, similar to the ones realized by human operators.

VI. ACKNOWLEDGMENTS

This work is supported in part by the European Research Council under the Advanced Grant SoftHands “A Theory of Soft Synergies for a New Generation of Artificial Hands” no. ERC-291166, and by the FP7-ICT-2011-9 program “Cognitive Systems” (grant number 600918 “PaCMan - Probabilistic and Compositional Representations of Objects for Robotic Manipulation”, and EU FP7-ICT project, WALKMAN, “Whole-body Adaptive Locomotion and Manipulation”, no. 611832.

REFERENCES

- [1] O. Khatib, E. Demircan, V. De Sapio, L. Sentis, T. Besier, and S. Delp, “Robotics-based synthesis of human motion,” *Journal of Physiology-Paris*, vol. 103, no. 3, pp. 211–219, 2009.

- [2] B. Cao, G. I. Dodds, and G. W. Irwin, "Redundancy resolution and obstacle avoidance for cooperative industrial robots," *Journal of Robotic Systems*, vol. 16, no. 7, pp. 405–417, 1999.
- [3] H. Seraji, "Task options for redundancy resolution using configuration control," in *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on*. IEEE, 1991, pp. 2793–2798.
- [4] Y. Nakamura, *Advanced robotics: redundancy and optimization*. Addison-Wesley Longman Publishing Co., Inc., 1990.
- [5] D. Matsui, T. Minato, K. F. MacDorman, and H. Ishiguro, "Generating natural motion in an android by mapping human motion," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 3301–3308.
- [6] A. M. Zanchettin, P. Rocco, L. Bascetta, I. Symeonidis, and S. Peldschus, "Kinematic analysis and synthesis of the human arm motion during a manipulation task," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2692–2697.
- [7] C. Ott, *Cartesian impedance control of redundant and flexible-joint robots*. Springer, 2008, vol. 49.
- [8] T. Wimböck, C. Ott, A. Albu-Schäffer, and G. Hirzinger, "Comparison of object-level grasp controllers for dynamic dexterous manipulation," *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 3–23, 2012.
- [9] S. Schneider and R. Cannon Jr, "Object impedance control for cooperative manipulation: Theory and experimental results," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 383–394, 1992.
- [10] R. Bonitz and T. Hsia, "Internal force-based impedance control for cooperating manipulators," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 78–89, 1996.
- [11] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani, "Six-dof impedance control of dual-arm cooperative manipulators," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 5, pp. 576–586, 2008.
- [12] A. Dietrich, T. , and A. Albu-Schaffer, "Dynamic whole-body mobile manipulation with a torque controlled humanoid robot via impedance control laws," in *Proceedings of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept. 2011, pp. 3199–3206.
- [13] A. Albu-Schaffer, M. Fischer, G. Schreiber, F. Schoeppe, and G. Hirzinger, "Soft robotics: what cartesian stiffness can obtain with passively compliant, uncoupled joints?" in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.*, vol. 4. IEEE, 2004, pp. 3295–3301.
- [14] W. Owen, E. Croft, and B. Benhabib, "Acceleration and torque redistribution for a dual-manipulator system," *Robotics, IEEE Transactions on*, vol. 21, no. 6, pp. 1226–1230, 2005.
- [15] R. Trumbower, M. Krutky, B. Yang, and E. Perreault, "Use of self-selected postures to regulate multijoint stiffness during unconstrained tasks," *PLoS One*, vol. 4, no. 5, 2009.
- [16] T. Milner, "Contribution of geometry and joint stiffness to mechanical stability of the human arm," *Experimental Brain Research*, vol. 143, pp. 515–519, 2002.
- [17] E. Perreault, R. Kirsch, and P. Crago, "Voluntary control of static endpoint stiffness during force regulation tasks," *Journal of Neurophysiology*, vol. 87, pp. 2808–2816, 2002.
- [18] A. Ajoudani, M. Gabiccini, N. G. Tsagarakis, A. Albu-Schäffer, and A. Bicchi, "TeleImpedance: Exploring the role of common-mode and configuration-dependant stiffness," in *IEEE International Conference on Humanoid Robots*, 2012.
- [19] A. Ajoudani, N. G. Tsagarakis, and A. Bicchi, "Tele-Impedance: Teleoperation with impedance regulation using a body-machine interface," *International Journal of Robotics Research*, vol. 31(13), pp. 1642–1655, 2012, <http://www.youtube.com/watch?v=KPO6IO7Tr-Q>.
- [20] Y. Guiard, "Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model," *Journal of motor behavior*, vol. 19, pp. 486–517, 1987.
- [21] K. Hinckley, R. Pausch, D. Proffitt, J. Patten, and N. Kassell, "Cooperative bimanual action," in *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. ACM, 1997, pp. 27–34.
- [22] Y. Guiard, "The kinematic chain as a model for human asymmetrical bimanual cooperation," *Advances in Psychology*, vol. 55, pp. 205–228, 1988.
- [23] A. Mohri, M. Yamamoto, and G. Hirano, "Cooperative path planning for two manipulators," in *Robotics and Automation, 1996. Proceedings, 1996 IEEE International Conference on*, vol. 3. IEEE, 1996, pp. 2853–2858.
- [24] J.-D. Choi, S. Kang, M. Kim, C. Lee, and J.-B. Song, "Two-arm cooperative assembly using force-guided control with adaptive accommodation," in *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on*, vol. 2, 1999, pp. 1253–1258.
- [25] J. Lee, P. Chang, and R. Jamisola, "Relative impedance control for dual-arm robots performing asymmetric bimanual tasks," *IEEE Transactions on Industrial Electronics*, 2013, in press.
- [26] S.-F. Chen and I. Kao, "Conservative congruence transformation for joint and cartesian stiffness matrices of robotic hands and fingers," *The International Journal of Robotics Research*, vol. 19, no. 9, pp. 835–847, 2000.
- [27] H. Gomi and M. Kawato, "Human arm stiffness and equilibrium-point trajectory during multi-joint movement," *Biological cybernetics*, vol. 76, no. 3, pp. 163–171, 1997.
- [28] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 13, no. 3, pp. 398–410, 1997.
- [29] G. Schreiber, A. Stemmer, and R. Bischoff, "The fast research interface for the kuka lightweight robot," in *IEEE Conference on Robotics and Automation (ICRA)*, 2010.

Grasp analysis tools for synergistic underactuated robotic hands

The International Journal of
Robotics Research
32(13) 1553–1576
© The Author(s) 2013
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364913504473
ijr.sagepub.com


Marco Gabiccini^{1,2,3}, Edoardo Farnioli^{1,3} and Antonio Bicchi^{1,3,4}

Abstract

Despite being a classical topic in robotics, the research on dexterous robotic hands still stirs a lively research activity. The current interest is especially attracted by underactuated robotic hands where a high number of degrees of freedom (DoFs), and a relatively low number of degrees of actuation co-exist. The correlation between the DoFs obtained through a wise distribution of actuators is aimed at simplifying the control with a minimal loss of dexterity. In this sense, the application of bio-inspired principles is bringing research toward a more conscious design. This work proposes new, general approaches for the analysis of grasps with synergistic underactuated robotic hands.

After a review of the quasi-static equations describing the system, where contact preload is also considered, two different approaches to the analysis are presented. The first one is based on a systematic combination of the equations. The independent and the dependent variables are defined, and cause–effect relationships between them are found. In addition, remarkable properties of the grasp, as the subspace of controllable internal force and the grasp compliance, are worked out in symbolic form. Then, some relevant kinds of tasks, such as pure squeeze, spurious squeeze and kinematic grasp displacements, are defined, in terms of nullity or non-nullity of proper variables. The second method of analysis shows how to discover the feasibility of the pre-defined tasks, operating a systematic decomposition of the solution space of the system. As a result, the inputs to be given to the hand, in order to achieve the desired system displacements, are found. The study of the feasible variations is carried out arriving at the discovery of all the combinations of nullity and/or non-nullity variables which are allowed by the equations describing the system. Numerical results are presented both for precision and power grasps, finding forces and displacements that the hand can impose on the object, and showing which properties are preserved after the introduction of a synergistic underactuation mechanism.

Keywords

Grasp analysis, compliant grasps, underactuated robotic hands, soft synergies, structural properties, grasp taxonomy

1. Introduction

The design of robotic hands was directed for a long time to obtaining better dexterity and capability for manipulation tasks. This goal was sought by increasing complexity and number of degrees of freedom, with remarkable examples of robotic hand designs following this trend represented by the UTAH/MIT hand (Jacobsen et al., 1984), the Robonaut Hand (Lovchik and Diftler, 1999), the Shadow hand (Shadow Robot Company Ltd., 2009) and the DLR hand arm system (Grebstein et al., 2012). One of the main issues of this approach is that a large number of degrees of freedom implies many actuators, with a consequent growth of size, weight and cost. With such kinds of devices, another disadvantage is the difficulty in control. Finding a joint control law, able to perform a given task, becomes a difficult job, mainly due to the complex kinematic structures of the hands. Starting from these observations, different kinds of

underactuation strategies were introduced and studied (Bicchi, 2000; Birglen et al., 2008). In this direction, a recently considered approach has been to take inspiration from the human hand, not just from a biomechanic point of view, but also for the control aspects.

Despite significant differences in the definitions and in the requirements of the investigated tasks, many neuroscience studies such as Fish and Soechting (1992), Angelaki and Soechting (1993), Soechting and Flanders (1997), Santello et al. (1998), Latash et al. (2005), Thakur et al.

¹Research Center “E. Piaggio,” Università di Pisa, Italy

²Department of Civil and Industrial Engineering, Università di Pisa, Italy

³Department of Advanced Robotics, Istituto Italiano di Tecnologia, Italy

⁴Department of Information Engineering, Università di Pisa, Italy

Corresponding author:

Edoardo Farnioli, Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy.
Email: edoardo.farnioli@iit.it

(2008) and Castellini and van der Smagt (2013) to mention only a few, share a main observation: simultaneous motion of multiple digits occurs in a consistent fashion, even when the task may require a fairly high degree of movement individuation such as grasping a small object or typing.

One of the main quantitative results is that a large variety of common grasps that humans can perform is well described by just five synergies,¹ the first two being important enough to explain 80% of the variance in grasp postures. This suggests the possibility of changing the basis of the description, from the joint space to the human-inspired postural synergy space. In Ciocarlie et al. (2007) this idea was implemented via software to control the hand for grasp planning, interrupting the rigid coordination of a finger when it arrives in contact with the object to be grasped. A second way, proposed in Brown and Asada (2007), follows the road of a hardware implementation, where two kinds of synergies can be achieved by changing the set of actuation pulleys. In this case, when a first contact with the object is achieved, either stable or not, the hand has to stop its motion. Differently from the previous one, the *soft synergies* approach Gabiccini et al. (2011) avoids the problems due to a strict correlation between joint motions. In this case the synergistic movement is imposed on a virtual hand, attracting the real one via a generalized spring. The influence of synergies in motion and forces is investigated in Prattichizzo et al. (2011). The contact force optimization problem was investigated in Gabiccini et al. (2011), taking into account the limitation imposed in the controllability by the underactuation. Recently, the possibility to obtain a soft synergy like controllability using adaptive synergies was studied in Catalano et al. (2012) and Grioli et al. (2012), in order to achieve both the results to get human-inspired movement and an easier mechanical implementation.

This paper, based in part on Gabiccini et al. (2012), deals with the quasi-static analysis of synergistic underactuated grasps, and proposes an approach where the set of equations describing the system is treated as a whole. Results are not limited to providing the controllable system variations by synergistic actuation. In addition to this, systematic ways to discover relevant properties of the grasp are provided, even in symbolic form, and a method to define and study the feasibility of some tasks is given. In Section 2 a review of the analytical description of the problem, based on the screw theory (Murray et al., 1994; Davidson and Hunt, 2004) is presented. Since also the derivative terms of the Jacobian matrix are taken into account, configurations with contact force preload can be rigorously analyzed by the proposed method. Our choice is to use an object-centric formulation, obtaining a constant grasp matrix and some simplifications to describe rigid object displacements. The resulting system of equations is treated with two different approaches. The first, in Section 3, presents a systematic way to elaborate the system of equations. *Input* and *output* variables are defined and used in an algorithm of Gauss–Jordan elimination, extended to act on block partitioned matrices. In

consequence of this, the relationships between the controls and the consequent hand/object configurations are given, both for displacements and forces, in two different ways, one more effective for numerical applications, the other more appropriate with symbolic forms of the equations. This leads to finding relevant properties of the grasp, e.g. a basis for the *controllable internal forces*, *controllable object displacements* and the *grasp compliance*, discussed in Section 4. In Sections 5 and 6, the main categories of tasks are defined in terms of nullity or non-nullity of some variables. These definitions can be used to suitably apply the *reduced row echelon form* (RREF) to the solution space of the system, ascertaining their feasibility and the necessary inputs to be applied in order to perform them. The idea to distinguish different kinds of solutions is expanded in Section 7, where the equations describing the kineto-static behavior of the system are analyzed to discover which combinations of nullity or non-nullity are allowed, until a complete classification of the solution space is obtained. In Section 8 some words are spent describing how the methods proposed in the paper could be applied to the study of robotic hands characterized by other types of (under-)actuation. To conclude, in Section 9, numerical examples are presented for tip and power grasps. Both cases are firstly studied as if the hand were completely actuated, finding out the hand manipulation capabilities. Then, the synergistic underactuation is introduced verifying which possibilities are lost and which others still persist.

2. System description

A basic assumption in the paper is that all bodies are modelled as rigid. Thanks to this, it is possible to use the wide variety of mathematical tools developed to describe rigid body motions. However, the presence of compliance in the system is allowed in a lumped way. For example, as we will further discuss later, the softness of the contact fingertip and/or the grasped object can be described by introducing contact virtual springs.

It is worth observing that, as usually done in literature, we separate the investigation of the system capabilities, e.g. the contact forces on the object actually controllable by the hand, to the restriction imposed by the friction limits, assuming that a subsequent classical force optimization procedure can lead to a feasible solution.

With reference to Figure 1, we model a hand as a collection of serial robots manipulating an object. An inertial frame $\{A\} = \{O_a; x_a, y_a, z_a\}$ is attached to the palm. On the i th of the p contact points we place a frame $\{C_i^h\} = \{O_{c_i^h}; x_{c_i^h}, y_{c_i^h}, z_{c_i^h}\}$ attached to the finger link, and a frame $\{C_i^o\} = \{O_{c_i^o}; x_{c_i^o}, y_{c_i^o}, z_{c_i^o}\}$ attached to the object.

A reference frame $\{B\} = \{O_b; x_b, y_b, z_b\}$ is fixed on the grasped object: later in the text we will refer to it as the *body* frame. With the symbols ξ_{xy}^z we indicate the twist of the frame $\{Y\}$ w.r.t. $\{X\}$, as described by an observer attached

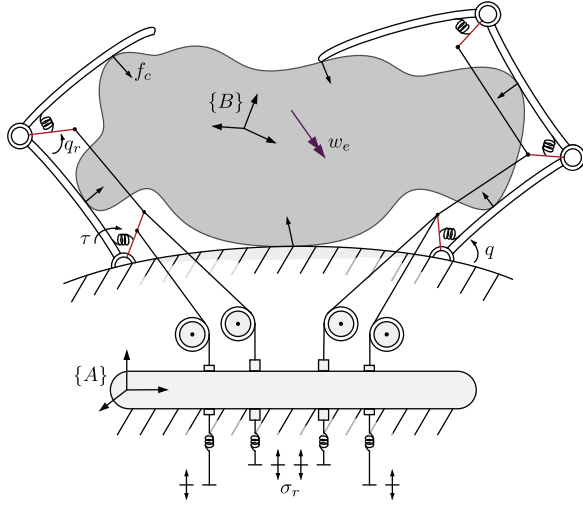


Fig. 1. Compliant grasp by an underactuated robotic hand.

to $\{Z\}$. Denoting with $g_{tz} \in SE(3)$ the posture of frame $\{Z\}$ with respect to $\{T\}$, according to Murray et al. (1994), we define the *adjoint matrix*, $Ad_g \in \mathbb{R}^{6 \times 6} : se(3) \rightarrow se(3)$, as an operator able to transform twists by the law

$$\xi_{xy}^t = Ad_{g_{tz}} \xi_{xy}^z \quad (1)$$

Indicating with w_x^y the wrench that the frame $\{X\}$ exerts on the frame $\{Y\}$, and in $\{Y\}$ components, we introduce the *co-adjoint matrix* operator, $Ad_g^{-T} \in \mathbb{R}^{6 \times 6} : se^*(3) \rightarrow se^*(3)$, transforming wrenches according to the law

$$w_x^t = Ad_{g_{ty}}^{-T} w_x^y \quad (2)$$

For the reader's convenience, the explicit form of the homogeneous and adjoint operators is given in 10. Other definitions and the notation used in the text are summarized in Table 1.

2.1. Equilibrium equation of the object

Let $w_e^b \in \mathbb{R}^6$ be the external wrench acting on the object with components expressed in $\{B\}$. Similarly, we indicate with $w_{c_i}^{c_i^o}$ the wrench that the hand exerts on the i th contact point on the object, with components in $\{C_i^o\}$. The equilibrium of the frame $\{B\}$ requires that the sum of all force/moment contributions, with components in $\{B\}$, is null. This condition can be written as

$$w_e^b + \sum_{i=1}^p Ad_{g_{bc_i}}^{-T} w_{c_i}^{c_i^o} = 0 \quad (3)$$

The hand/object interaction can be described as a six-dimensional constraint for the position and orientation of the local contact frame. Usually, however, depending on the nature of the contact type, the constrained directions are less than six. In this way, we assume to describe the contact interaction with a vector $f_{c_i}^{c_i^o} \in \mathbb{R}^{c_i}$, with $c_i \leq 6$, that

Table 1. Notation for grasp analysis.

Notation	Definition
δv	Variation of variable v
$\bar{v}$	Value of variable v in reference configuration
$\#v$	Dimensions of vector v
$\{A\}$	Palm (inertial) frame
$\{C_i^h\}$	i th Contact frame attached to the hand
$\{C_i^o\}$	i th Contact frame attached to the object
$\{B\}$	Object frame
$q \in \mathbb{R}^{\#q}$	Joint variables
$q_r \in \mathbb{R}^{\#q}$	Joint reference variables
$\tau \in \mathbb{R}^{\#q}$	Joint torques
$\sigma \in \mathbb{R}^{\#\sigma}$	Synergy variables
$\sigma_r \in \mathbb{R}^{\#\sigma}$	Synergy reference variables
$\eta \in \mathbb{R}^{\#\sigma}$	Synergy generalized forces
p	Number of contact points
c	Number of hand/object interaction constraints
$u \in \mathbb{R}^6$	Position and orientation of the object frame
$\xi_{xy}^z \in \mathbb{R}^6$	In the motion of the frame $\{Y\}$ relative to $\{X\}$, it represents the twist of the frame $\{Z\}$ as attached to $\{Y\}$, with components expressed in $\{Z\}$
$f_{c_i}^{c_i^o}$	Contact force/torque exerted from $\{C_i^h\}$ to $\{C_i^o\}$ with components and moments relative to the frame $\{C_i^o\}$
$w_x^y \in \mathbb{R}^6$	Wrench exerted by $\{X\}$ on $\{Y\}$, with components and moments relative to $\{Y\}$
$aJ \in \mathbb{R}^{c \times \#q}$	Hand Jacobian matrix in inertial frame
$c^o J \in \mathbb{R}^{c \times \#q}$	Hand Jacobian matrix in object contact frames
$S \in \mathbb{R}^{\#q \times \#\sigma}$	Synergy matrix
${}^b G \in \mathbb{R}^{6 \times c}$	Grasp matrix in body frame

takes into account only the directions constrained. More details will be provided in Section 2.5. For each contact point we introduce a matrix $B_i \in \mathbb{R}^{6 \times c_i}$ able to map the local interaction in a complete wrench as follows

$$w_{c_i}^{c_i^o} = B_i f_{c_i}^{c_i^o} \quad (4)$$

Substituting (4) in (3), we obtain

$$w_e^b + \sum_{i=1}^p Ad_{g_{bc_i}}^{-T} B_i f_{c_i}^{c_i^o} = 0 \quad (5)$$

Defining $c = \sum_{i=1}^p c_i$ as the total number of contact constraints acting on the object, we define the *grasp matrix in body frame*, ${}^b G \in \mathbb{R}^{6 \times c}$, as

$${}^b G = \begin{bmatrix} Ad_{g_{bc_1}}^{-T} B_1 & \cdots & Ad_{g_{bc_i}}^{-T} B_i & \cdots & Ad_{g_{bc_p}}^{-T} B_p \end{bmatrix} \quad (6)$$

Equation (6) allows us to express the *object equilibrium equation* in the form

$$w_e^b + {}^b G f_{c_i}^{c_i^o} = 0 \quad (7)$$

where $f_{ch}^{co} = [f_{c_1}^{coT}, \dots, f_{c_p}^{coT}]^T \in \mathbb{R}^c$ is a vector collecting all the actions performed by the hand on the object. It is important to underline that the grasp matrix in body frame is a constant matrix if the object contact points do not change. In consequence of this, to find a law relating small perturbations of the variables involved, we can calculate the differential of (7). It follows that the perturbed equilibrium equation can be written as

$$\delta w_e^b + {}^b G \delta f_{ch}^{co} = 0 \quad (8)$$

2.2. Congruence equation of the object

The twist of the i th contact frame $\{C_i^o\}$ on the object can be described as a function of the twist of the object frame $\{B\}$. To do this we use the adjoint matrix, presented in (1), as follows

$$\xi_{ab}^{co} = \text{Ad}_{g_{c_i^o b}} \xi_{ab}^b \quad (9)$$

As briefly discussed before, usually the contact constraints do not involve all the displacement directions. As explained in Bicchi (1994), the matrix B_i^T is able to select the terms of velocity violating the contact constraints, and the matrix B_i in (4) can expand the force/moment vector in a complete wrench. Introducing the vector $v_{ab}^{co} \in \mathbb{R}^{c_i}$, containing only the terms of the i th velocity violating constraints, the equation relating the body frame twist and the velocity of the i th contact point frame can be expressed as

$$v_{ab}^{co} = B_i^T \xi_{ab}^{co} = B_i^T \text{Ad}_{g_{c_i^o b}} \xi_{ab}^b \quad (10)$$

Defining $v_{ab}^{co} = [v_{ab}^{coT} \dots v_{ab}^{c_i^oT} \dots v_{ab}^{c_p^oT}]^T \in \mathbb{R}^c$, using (9) and (10), we can write

$$v_{ab}^{co} = \left[(B_1^T \text{Ad}_{g_{c_1^o b}})^T \dots (B_i^T \text{Ad}_{g_{c_i^o b}})^T \dots (B_p^T \text{Ad}_{g_{c_p^o b}})^T \right]^T \xi_{ab}^b \quad (11)$$

By a direct comparison between (11) and the definition of the grasp matrix in (6), it immediately follows that

$$v_{ab}^{co} = {}^b G^T \xi_{ab}^b \quad (12)$$

Remembering that the rotational part of the term $\xi_{ab}^b dt$ does not represent a variation of coordinates, we have to introduce a vector $u \in \mathbb{R}^6$ able to parametrize $SE(3)$. A possible approach is to choose a parametrization vector composed by the same translational part of ξ_{ab}^b and Euler's angles in the rotational part, as described by Sciavicco and Siciliano (2000). A convenient approach is to introduce a virtual kinematic chain, describing the object configuration with respect to the absolute frame. In this case the *object Jacobian* matrix, that is the Jacobian of the virtual chain, can be used to parametrize the object twist, as follows

$$\xi_{ab}^b = J_o(u) \dot{u} \quad (13)$$

Defining the variation of the contact frame of the object as $\delta C_{ab}^{co} = v_{ab}^{co} dt$, we can substitute (13) in (12). Introducing the

matrix ${}^b G_t^T(u) = {}^b G^T J_o(u)$, we can write the congruence equation of the object as

$$\delta C_{ac}^{co} = {}^b G_t^T(u) \delta u \quad (14)$$

2.3. Congruence equation of the hand

Letting the vector $\xi_{ac_i}^a \in \mathbb{R}^6$ be the twist of the frame $\{C_i^h\}$ with respect to $\{A\}$, with components expressed in $\{A\}$, as described by Murray et al. (1994) and summarized in 10, we introduce the *spatial Jacobian matrix* ${}^a J_i(q_i) \in \mathbb{R}^{6 \times \sharp q_i}$, where $\sharp q_i$ is the number of joints that precedes the i th hand contact point in its own serial chain. This allows us to write the relationship

$$\xi_{ac_i}^a = {}^a J_i(q_i) \dot{q}_i \quad (15)$$

An object-centric description is obtained mapping (15) from the inertial frame $\{A\}$ to object frame $\{C_i^o\}$, thus using the adjoint operator as follows

$$\xi_{ac_i}^{co} = \text{Ad}_{g_{c_i^o a}(u)} \xi_{ac_i}^a = \text{Ad}_{g_{c_i^o a}(u)} {}^a J_i(q_i) \dot{q}_i \quad (16)$$

The selection matrix B_i^T can be used, as in (10), to maintain the terms of velocity in the directions violating contact constraints. This results in

$$v_{ac_i}^{co} = B_i^T \text{Ad}_{g_{c_i^o a}(u)} {}^a J_i(q_i) \dot{q}_i = {}^c J_i(q_i, u) \dot{q}_i \quad (17)$$

Similarly to what previously performed, defining $v_{ac}^{co} = [v_{ac_1}^{coT} \dots v_{ac_i}^{coT} \dots v_{ac_p}^{coT}]^T \in \mathbb{R}^c$ and collecting all the joint parameters in the vector $q \in \mathbb{R}^{\sharp q}$, from (17) we can write

$$v_{ac}^{co} = {}^c J(q, u) \dot{q} \quad (18)$$

where ${}^c J(q, u)$ is the *hand Jacobian* referred to the object contact frames. Multiplying (18) by dt , we find the congruence equation of the hand as

$$\delta C_{ac}^{co} = {}^c J(q, u) \delta q \quad (19)$$

2.4. Equilibrium equation of the hand

From the kineto-static duality, coherently with the expression of the hand Jacobian, the map from the forces exerted by the hand f_{ch}^{co} to the joint torques $\tau \in \mathbb{R}^{\sharp q}$ is given by

$$\tau = {}^c J^T(q, u) f_{ch}^{co} \quad (20)$$

Small perturbations of the system can be described by differentiation of (20), obtaining

$$\begin{aligned} \delta \tau = & \frac{\partial {}^c J^T(q, u) f_{ch}^{co}}{\partial q} \delta q + \frac{\partial {}^c J^T(q, u) f_{ch}^{co}}{\partial u} \delta u \\ & + \frac{\partial {}^c J^T(q, u) f_{ch}^{co}}{\partial f} \delta f_{ch}^{co} \end{aligned} \quad (21)$$

Defining the matrices

$$\begin{aligned} Q &= \frac{\partial {}^c J^T(q, u) f_{ch}^{c^o}}{\partial q} \in \mathbb{R}^{\sharp q \times \sharp q}, \\ U &= \frac{\partial {}^c J^T(q, u) f_{ch}^{c^o}}{\partial u} \in \mathbb{R}^{\sharp q \times 6} \end{aligned} \quad (22)$$

the equation describing the perturbed equilibrium of the hand can be expressed as

$$\delta \tau = Q \delta q + U \delta u + {}^c J^T \delta f_{ch}^{c^o} \quad (23)$$

Evaluating (23) in an equilibrium configuration, the terms Q and U become negligible if the initial contact forces are small or null. In 10 we present a method to compute the derivatives in (22), based only on the knowledge of the elements composing (17), that is the spatial Jacobian and adjoint transformations, in an algorithmic way. From a practical point of view, this implies that symbolic calculations are not required to compute (22).

2.5. Hand/object interaction model

The *contact* between the hand and the object can be interpreted as a motion constraint. Typically, depending on the nature of materials involved, not all the directions of motions are limited. As an example, for the case of *contact point without friction*, only the direction normal to the contact surface is forbidden. This contact type is equivalent to having only one constraint; thus the dimension of the vector $f_{ch}^{c^o}$ is $c_i = 1$. Similarly, the *contact point with friction*, or *hard finger*, allows the presence of three components of force, on the contact, but no moment, $c_i = 3$. As a last example we cite the *soft finger* case. With respect to the hard finger, it adds the possibility to transmit a moment around the normal vector to the contact surface, $c_i = 4$.

In general, we are not justified in assuming that the total number of motion constraints imposed by the interaction with the hand is equal to the degrees of freedom of the grasped object (six in a 3D case, three in a planar case). On the contrary, we usually have more constraints than degrees of freedom, and this makes the problem hyperstatic, also known as statically indeterminate. In these cases, the equilibrium equations are not sufficient to univocally determine the contact forces. To tackle this problem, we introduce a contact force model which is linear in the interference between the hand and the object. More specifically, for the i th contact point we introduce a *stiffness matrix* $K_{c_i} \in \mathbb{R}^{c_i \times c_i}$ containing the characteristics of the virtual contact spring. Accordingly, the i th force that the hand exerts on the object can be expressed as

$$\delta f_{ch}^{c^o} = K_{c_i} (\delta C_{ac_i}^{c^o} - \delta C_{ac_i}^{c^o}) \quad (24)$$

Introducing $K_c = \text{blkdiag}(K_{c_1}, \dots, K_{c_p}) \in \mathbb{R}^{c \times c}$, as a block diagonal matrix, where the i th block element is K_{c_i} , the constitutive equation of the contact forces becomes

$$\delta f_{ch}^{c^o} = K_c (\delta C_{ac}^{c^o} - \delta C_{ac}^{c^o}) \quad (25)$$

2.6. Joint actuation model

For the i th joint we introduce an elastic model relating the real configuration with a *reference* value, $q_{r_i} \in \mathbb{R}$, described by a torsional spring with constant stiffness $k_{q_i} \in \mathbb{R}$

$$\tau_i = k_{q_i} (q_{r_i} - q_i) \quad (26)$$

Defining $K_q = \text{diag}(k_{q_i}) \in \mathbb{R}^{\sharp q \times \sharp q}$ as the matrix collecting all the joint stiffness, and introducing $q_r \in \mathbb{R}^{\sharp q}$ as the vector collecting the joint reference variables, by differentiation we can write the quasi-static actuation law as

$$\delta \tau = K_q (\delta q_r - \delta q) \quad (27)$$

2.7. Underactuation of the hand

The equations presented up to here allow the description of a hand with elastic joints, but with an equal number of degrees of actuation and freedom. However, it is possible to extend the problem to include underactuated systems as well. To this end, we impose that the joint reference positions q_r evolve on a manifold $\mathbb{S} \subseteq \mathbb{R}^{\sharp q}$ as a function of some *postural synergy* values $\sigma \in \mathbb{R}^{\sharp \sigma}$, where $\sharp \sigma \leq \sharp q$. Thus, in general we can write

$$q_r = f(\sigma) \quad (28)$$

where $f: \mathbb{R}^{\sharp \sigma} \rightarrow \mathbb{S} \subseteq \mathbb{R}^{\sharp q}$ is the *synergy function*. Calculating the differential of (28), and defining the *synergy matrix* as $S(\sigma) = \frac{\partial f(\sigma)}{\partial \sigma} \in \mathbb{R}^{\sharp q \times \sharp \sigma}$, the joint reference displacements are described as follows

$$\delta q_r = S(\sigma) \delta \sigma \quad (29)$$

Again, by virtue of the kineto-static duality, we can find that the vector $\eta \in \mathbb{R}^{\sharp \sigma}$ of the generalized forces at the synergy level is related to the joint torques by the relationship

$$\eta = S^T(\sigma) \tau \quad (30)$$

Differentiating (30) and introducing the matrix $\Sigma(\sigma, \tau) = \frac{\partial S^T(\sigma) \tau}{\partial \sigma} \in \mathbb{R}^{\sharp \sigma \times \sharp \sigma}$, for the actuation forces we obtain

$$\delta \eta = S^T(\sigma) \delta \tau + \Sigma(\sigma, \tau) \delta \sigma \quad (31)$$

For the sake of generality, we can introduce an elastic model even for the synergistic underactuation, similarly to what was achieved before for the joints. Defining the *synergy stiffness* as the matrix $K_\sigma = \text{diag}(k_{\sigma_i}) \in \mathbb{R}^{\sharp \sigma \times \sharp \sigma}$, and introducing the vector of *synergy references* $\sigma_r \in \mathbb{R}^{\sharp \sigma}$, the actuation law appears as

$$\delta \eta = K_\sigma (\delta \sigma_r - \delta \sigma) \quad (32)$$

2.8. The Fundamental Grasp Equation

The set of equations composed collecting (8), (14), (19), (23), (25), (27), (29), (31) and (32) is able to describe a

linear approximation of all the possible system displacements around an equilibrium configuration. The result is the *Fundamental Grasp Equation* (FGE) and it appears as

$$\begin{bmatrix} I_{\#f} & 0 & 0 & 0 & -K_c & K_c & 0 & 0 & 0 & 0 & 0 \\ -{}^cJ^T & I_{\#t} & 0 & -\bar{U} & 0 & 0 & -\bar{Q} & 0 & 0 & 0 & 0 \\ 0 & -\bar{S}^T & I_{\#n} & 0 & 0 & 0 & 0 & 0 & -\bar{\Sigma} & 0 & 0 \\ {}^bG & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{\#w} & 0 \\ 0 & 0 & 0 & 0 & I_{\#Ch} & 0 & -{}^cJ & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -{}^b\bar{G}_t^T & 0 & I_{\#Co} & 0 & 0 & 0 & 0 & 0 \\ 0 & I_{\#t} & 0 & 0 & 0 & 0 & K_q & -K_q & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I_{\#qr} & -\bar{S} & 0 & 0 & 0 \\ 0 & 0 & I_{\#s} & 0 & 0 & 0 & 0 & 0 & K_\sigma & 0 & -K_\sigma \end{bmatrix} \begin{bmatrix} \delta f_{ch}^{co} \\ \delta \tau \\ \delta \eta \\ \delta u \\ \delta C_{ac}^{co} \\ \delta C_{ac}^{ch} \\ \delta q \\ \delta q_r \\ \delta \sigma \\ \delta w_e^b \\ \delta \sigma_r \end{bmatrix} = 0 \quad (33)$$

where we indicated with $\bar{v}$ the value of the variable v in the starting equilibrium point, and with the symbol $I_{\#v}$ an identity matrix of dimension $\#v$.

Equation (33) is a system of *linear* and *homogeneous* equations in the form $A\delta y = 0$, where the coefficient matrix $A \in \mathbb{R}^{r_a \times c_a}$ is the FGM and the vector $y \in \mathbb{R}^{c_a}$ collecting all the variable is the *augmented configuration*.

It is trivial to note that the constraint identified by (33) does not change its physical implications using a different order of both the equations and the variables. For reasons that will be clarified in Section 3.2.4, the chosen order of the equations and of the *augmented configuration* components derives from the specification to have square blocks on the main diagonal. However, it is always possible to obtain this characteristic from any other order of equations and variables, simply applying suitable permutation matrices.

3. Controllable system perturbations

It is easy to find that the dimension r_a and c_a of the Fundamental Grasp Matrix (FGM) in (33) correspond to

$$\begin{aligned} r_a &= \#w + 3\#t + 3\#f + 2\#\sigma \\ c_a &= 2\#w + 3\#t + 3\#f + 3\#\sigma \end{aligned} \quad (34)$$

In the very general cases of interest, it produces $\text{rank}(A) = r_a$, and we will assume this in the rest of the text. Exceptions are analytically possible but they refer to unusual situations of poor practical relevance.

All the solutions of (33) can be written as a basis for the nullspace of the FGM. We indicate this with the symbol $\Gamma \in \mathbb{R}^{r_a \times c_\gamma}$. Taking into account (34), it immediately follows that

$$\begin{aligned} r_\gamma &= c_a \\ c_\gamma &= c_a - r_a = \#w + \#\sigma \end{aligned} \quad (35)$$

This implies that we can find a unique solution of (33) when it is known a number of independent variables, or *inputs* for the system, equal to $c_\gamma = \#w + \#\sigma$.

In continuity with the grasp analysis literature, we choose to consider the external wrench variation δw_e^b as known or

measurable. Moreover, we consider the synergy references as position-controlled. This corresponds to the practical case of an actuation by servo motors to control the synergy displacements $\delta \sigma_r$. Other choices are possible, maintaining constant c_γ , for example by substituting the external wrench δw_e^b with the object displacement δu , or the reference synergy positions $\delta \sigma_r$ with synergy force actuation $\delta \eta$. The results of our analysis can be easily adapted to the above mentioned situations as well.

The next sections will show two different methods to obtain the complete configuration of the system as a function of the input variables. The first method, called *Find-X*, presented in Section 3.1, has the advantage of a simple and versatile application, very effective from a numerical point of view. On the contrary, the second one, called GEROME-B, in Section 3.2, has the advantage to produce input/output equations with relevant physical sense, more effective when presented in symbolic form. Furthermore, it results an useful systematic method to merge some subspaces of interest in the grasp analysis, and we will better discuss these in Sections 4.2 and 4.3.

3.1. Find-X method

From previous considerations, it follows that all the solution of (33) can be obtained as a combination of the columns of the solution space Γ . Thus, introducing a vector of parameters $x \in \mathbb{R}^{\#w + \#\sigma}$, every augmented configuration variation such that

$$\delta y = \Gamma x \quad (36)$$

will be a solution of the system.

While δy is composed by a set of variables, we can also divide the matrix Γ in the corresponding portions as follows

$$\begin{bmatrix} \delta f_{ch}^{co} \\ \delta \tau \\ \delta \eta \\ \delta u \\ \delta C_{ac}^{co} \\ \delta C_{ac}^{ch} \\ \delta q \\ \delta q_r \\ \delta \sigma \\ \delta w_e^b \\ \delta \sigma_r \end{bmatrix} = \begin{bmatrix} \Gamma_f \\ \Gamma_\tau \\ \Gamma_\eta \\ \Gamma_u \\ \Gamma_{C^h} \\ \Gamma_{C^o} \\ \Gamma_q \\ \Gamma_{q_r} \\ \Gamma_\sigma \\ \Gamma_w \\ \Gamma_{\sigma_r} \end{bmatrix} x \quad (37)$$

where, for the block relative to the variable v , it results $\Gamma_v \in \mathbb{R}^{\#v \times c_\gamma}$.

Under the assumption that the inputs of the system are the external wrench and the synergy references, it is possible to calculate the vector x in (37), responsible of the column combination of Γ , simply as

$$x = \begin{bmatrix} \Gamma_w \\ \Gamma_{\sigma_r} \end{bmatrix}^{-1} \begin{bmatrix} \delta w_e^b \\ \delta \sigma_r \end{bmatrix} \quad (38)$$

The perturbed configuration of the system can be found by substituting (38) in (36) obtaining

$$\delta y = \Gamma \begin{bmatrix} \Gamma_w \\ \Gamma_{\sigma_r} \end{bmatrix}^{-1} \begin{bmatrix} \delta w_e^b \\ \delta \sigma_r \end{bmatrix} \quad (39)$$

The invertibility of the matrix $\begin{bmatrix} \Gamma_w^T & \Gamma_{\sigma_r}^T \end{bmatrix}^T$ is acceptable with the same generality of the assumption that the coefficient matrix in (33) is full rank. In the eventuality of different inputs, it is immediate to find the corresponding portions of Γ to calculate vector x in the same way as in (38).

3.2. GEROME-B: The Gauss elementary row operation method for a block partitioned matrix

Since the system in (33) is linear and homogeneous, we can act on the coefficient matrix performing a Gauss–Jordan elimination. To do this maintaining the integrity of the submatrices composing the FGM, we need to extend the *Gauss elementary row operations* to process a block partitioned matrix. We can write the three Gauss elementary block operations as:

- exchanging the i th block row with the j th block row;
- pre-multiplying the i th block row for a full column rank matrix Δ ;
- adding the j th block row, eventually multiplied for a suitable matrix Λ to according dimensions, to the i th block row.

As well as the original elimination method, we need to apply these rules without affecting the solution space of the system. To this aim, we remind the reader of the following

Theorem 3.1. *Given the matrices $P \in \mathbb{R}^{p \times c_p}$ and $M \in \mathbb{R}^{m \times c_m}$, with $c_m = r_p$, it results $\mathcal{N}(MP) = \mathcal{N}(P)$, if and only if M is full column rank.*

More details about this can be found in Meyer (2000). In the following sections, we will get three full-rank matrices able to translate the Gauss elementary operations as pre-multiplication matrices of a generic block partitioned matrix P , coefficient matrix of linear and homogeneous system of the type $Pz = 0$. Thanks to Theorem 3.1, we can guarantee the integrity of its own solution space.

3.2.1. First gauss operation To translate the first rule, considering P composed by m block rows, we start from a partitioned identity matrix $I_p \in \mathbb{R}^{r_p \times r_p}$, such that the i th of m blocks, I_i , has the same number of rows of the corresponding i th block row of P , $\forall i \in \{1, \dots, m\}$. To obtain the

effect of exchanging the i th and the j th block row of P , we have to pre-multiply it for a matrix in the form

$$M_{ij}^1 = \begin{bmatrix} I_1 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \ddots & 0 \\ 0 & \dots & I_{i-1} & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & I_j & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & I_{i+1} & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & \ddots & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & \ddots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & I_{j-1} & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & I_i & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & I_{j+1} & \dots & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & I_m \end{bmatrix} \quad (40)$$

3.2.2. Second gauss operation A matrix performing the second rule is obtained starting again from the partitioned identity matrix I_p , and substituting the i th identity block with a full rank matrix Δ . The searched pre-multiplication matrix appears as

$$M_{ii}^2(\Delta) = \begin{bmatrix} I_1 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & I_{i-1} & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & \Delta & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & I_{i+1} & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & \ddots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & I_m \end{bmatrix} \quad (41)$$

3.2.3. Third gauss operation At last, to sum two rows, we place a matrix Λ , of appropriate dimensions, at the intersection of the i th row and j th column of I_p , obtaining

$$M_{ij}^3(\Lambda) = \begin{bmatrix} I_1 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & I_i & 0 & \dots & 0 & \Lambda & \dots & 0 \\ 0 & \dots & 0 & \ddots & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & I & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & \ddots & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & I_j & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \ddots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & I_m \end{bmatrix} \quad (42)$$

3.2.4. The GEROME-B algorithm In the scalar case, the Gauss–Jordan elimination acts using pivots to eliminate the

Algorithm 1 GEROME-B

```

for  $h = 1 \rightarrow m$  do
   $\Delta = \hat{P}_{hh}^{-1}$ 
   $\hat{P} = M_{hh}^2(\Delta) \hat{P}$ 
  for  $k = 1 \rightarrow m$  do
    if  $h = k$  then
       $\Lambda = -\hat{P}_{kh}$ 
       $\hat{P} = M_{kh}^3(\Lambda) \hat{P}$ 
    end if
  end for
end for

```

other elements on the same column. Similarly, our extension needs *block pivot* elements. They must follow the following definition.

Definition 3.2. A block of the matrix P can be a pivot if and only if

- it is a full-rank square block;
- it is the only pivot in its row and column;
- it does not multiply an independent variable of the system.

The block elimination procedure, called GEROME-B – Gauss elementary row operation method for a block partitioned matrix – is explained in Algorithm 1. Without losing generality, we implement a particular permutation, $\hat{P}$, of the initial matrix P , obtained using matrices of the type (40), where $\hat{P}$ has all the pivots on the main diagonal, and the blocks multiplying the input variables in the last columns. When the algorithm is completed, the permutation can be inverted to replace the pivots and variables in the original order.

In order to apply Algorithm 1 to the FGM in (33), permutations are not needed if we take the elements on the main diagonal as pivots. In fact, in this case all the blocks on the main diagonal are square and full rank matrices. The only exception is the fourth row, where a null pivot element appears. Unfortunately, this is an obliged choice, because that is the only square matrix in its own column. To tackle this problem, we can consider the following steps:

1. Pre-multiplying the FGM for $M_{41}^3(-^bG)$, this is equivalent to substituting the fourth row with the fourth row minus bG times the first row.
2. Applying the matrix $M_{45}^3(-^bGK_c)$, this is equivalent to substituting the (new) fourth row with the fourth row minus bGK_c times the fifth row.
3. Applying the matrix $M_{46}^3(^bGK_c)$, this is equivalent to substituting the (new) fourth row with the fourth row plus bGK_c times the sixth row.

Applying these operations, from (33) we obtain the form

$$\begin{bmatrix} I_{\#f} & 0 & 0 & 0 & -K_c & K_c & 0 & 0 & 0 & 0 & 0 \\ -^c\tilde{o}J^T & I_{\#\tau} & 0 & -\tilde{U} & 0 & 0 & -\tilde{Q} & 0 & 0 & 0 & 0 \\ 0 & -\tilde{S}^T & I_{\#\eta} & 0 & 0 & 0 & 0 & 0 & -\tilde{\Sigma} & 0 & 0 \\ 0 & 0 & 0 & -^bGK_c^b\tilde{G}_t^T & 0 & 0 & ^bGK_c^c\tilde{J} & 0 & 0 & I_{\#w} & 0 \\ 0 & 0 & 0 & 0 & I_{\#Ch} & 0 & -^c\tilde{o}J & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -^b\tilde{G}_t^T & 0 & I_{\#Co} & 0 & 0 & 0 & 0 & 0 \\ 0 & I_{\#\tau} & 0 & 0 & 0 & 0 & K_q & -K_q & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{\#qr} & -\tilde{S} & 0 & 0 \\ 0 & 0 & I_{\#\sigma} & 0 & 0 & 0 & 0 & 0 & K_\sigma & 0 & -K_\sigma \end{bmatrix} \begin{bmatrix} \delta f_{ch}^{co} \\ \delta \tau \\ \delta \eta \\ \delta u \\ \delta C_{ac^o}^{co} \\ \delta C_{ac^o}^{co} \\ \delta q \\ \delta q_r \\ \delta \sigma \\ \delta w_e^b \\ \delta \sigma_r \end{bmatrix} = 0 \quad (43)$$

briefly indicated as $A_1 \delta y = 0$.

3.3. The FGM in canonical form

The new coefficient matrix A_1 in (43) satisfies all the necessary hypothesis to apply Algorithm 1, in terms of pivots collocation and rank. Before seeing the final result, we want to underline the following

Remark 1. GEROME-B allows us to obtain a new suitable form of the coefficient matrix. This corresponds to a new set of equations, where each dependent variable is a function only of the independent ones. If the pivots are on the main diagonal, the result is a quasi-diagonal matrix, or, otherwise, a general permutation of a quasi-diagonal matrix.

From previous considerations, it is clear that applying the Algorithm 1 to the coefficient matrix in (43), we obtain the form

$$\begin{bmatrix} I_{\#f} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & W_f & R_f \\ 0 & I_{\#\tau} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & W_\tau & R_\tau \\ 0 & 0 & I_{\#\eta} & 0 & 0 & 0 & 0 & 0 & 0 & W_\eta & R_\eta \\ 0 & 0 & 0 & I_{\#\mu} & 0 & 0 & 0 & 0 & 0 & W_\mu & R_\mu \\ 0 & 0 & 0 & 0 & I_{\#Ch} & 0 & 0 & 0 & 0 & W_{Ch} & R_{Ch} \\ 0 & 0 & 0 & 0 & 0 & I_{\#Co} & 0 & 0 & 0 & W_{Co} & R_{Co} \\ 0 & 0 & 0 & 0 & 0 & 0 & I_{\#q} & 0 & 0 & W_q & R_q \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{\#qr} & 0 & W_{qr} & R_{qr} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{\#\sigma} & W_\sigma & R_\sigma \end{bmatrix} \begin{bmatrix} \delta f_{ch}^{co} \\ \delta \tau \\ \delta \eta \\ \delta u \\ \delta C_{ac^o}^{co} \\ \delta C_{ac^o}^{co} \\ \delta q \\ \delta q_r \\ \delta \sigma \\ \delta w_e^b \\ \delta \sigma_r \end{bmatrix} = 0 \quad (44)$$

Equation (44) is briefly expressed as $A_c \delta y = 0$, where $A_c \in \mathbb{R}^{r_a \times c_a}$, is the *Canonical Form of the Fundamental Grasp*

Matrix (cFGM). For the reader's convenience, we refer to 10 for the complete expression of the matrices W_v and R_v .

It is worth observing that the previous results are relevant especially considering an application in symbolic form. In fact, similar conclusions are not difficult to achieve with a numerical approach. In this sense, using the subscript d to indicate portions relative to the dependent variables, and i to indicate the independent ones, we can observe that (33) can even be expressed as

$$A\delta y = \begin{bmatrix} A_d & A_i \end{bmatrix} \begin{bmatrix} \delta y_d \\ \delta y_i \end{bmatrix} = 0 \quad (45)$$

where $A_d \in \mathbb{R}^{r_a \times r_a}$, $A_i \in \mathbb{R}^{r_a \times (c_a - r_a)}$, $\delta y_d \in \mathbb{R}^{r_a}$, $\delta y_i \in \mathbb{R}^{(c_a - r_a)}$. Under the same previous hypothesis, the canonical form of the system can be numerically obtained premultiplying (45) for A_d^{-1} , as follows

$$A_d^{-1} \begin{bmatrix} A_d & A_i \end{bmatrix} \begin{bmatrix} \delta y_d \\ \delta y_i \end{bmatrix} = \begin{bmatrix} I & A_d^{-1} A_i \end{bmatrix} \begin{bmatrix} \delta y_d \\ \delta y_i \end{bmatrix} = A_c \delta y = 0 \quad (46)$$

4. Relevant properties of the FGM in canonical form

In the next sections, we will analyze the relevant properties of some blocks in (44). For the sake of completeness, in 10 the cases of a grasp performed by (a) a fully actuated hand, (b) a hand equipped with elastic joints and (c) a soft synergy underactuated hand are considered. The expression of the elements of the canonical forms of the FGMs are presented there for all the cases.

4.1. Controllable internal forces

From (44), we consider the expression of the contact force variation, that is

$$\delta f_{ch}^{c_o} + W_f \delta w_e^b + R_f \delta \sigma_r = 0 \quad (47)$$

Defining as *internal* the solutions of (44), or equivalently of (33), not involving the external wrench variation, it immediately follows that the span of the matrix R_f is the subspace of *controllable internal forces*.

4.2. Contact force transmission caused by an external wrench

Again from (47), we note that W_f is a map from the external wrench to the contact forces, when the inputs on the hand are kept constant. Therefore, $-W_f$ represents the *contact force transmission* of the external wrench.

Both controllable internal forces and contact force distribution have great relevance in problems as the contact force optimization and the evaluation of the force closure properties of the grasp.

4.3. Controllable internal object displacements

Especially in the case of underactuated hands, it can be tricky to find what kind of motion the hand can impose upon the object. This problem is solved by considering the fourth equation of (44), which gives us a description of the object displacements as

$$\delta u + W_u \delta w_e^b + R_u \delta \sigma_r = 0 \quad (48)$$

Easily we can deduce that the range space of the matrix R_u represents the subspace of the *controllable internal object displacements*.

4.4. Grasp compliance

Again from (48), we can find that $-W_u$ is the *grasp compliance*. In other words, $C_g = -W_u$ represents the compliance that a 6D spring should have to behave like the hand, maintaining the same object displacement consequent to an external wrench variation.

It is worth observing on passing that, applying Algorithm 1 to (33), substituting the displacement δu to the wrench δw_e^b as input variable, we would have dually found the *grasp stiffness*.

5. Main categories of manipulation tasks

In the previous sections, we have shown how to describe the system and to determine a perturbed configuration consequent to an input variable variation. Algorithm 1 allowed us to find some particular subsets of solutions of the system, e.g. the *controllable internal forces*. This suggested the idea to search other possible relevant solution subsets. In this way, we will show how to define some relevant types of grasps, based on the nullity or non-nullity patterns of the system variables. However, the given definitions involve only some variables. In Section 7, we will proceed to discover all the nullity patterns that could be present inside the solution set of (33).

To help us in a clear exposition of reasoning and results, we introduce a specific notation. For a variable variation $\delta v \in \mathbb{R}^{\sharp v}$ we could have the cases:

- $\delta v = 0$, if all the elements of the vector are null,
- $\delta v = *$, if at least one element of the vector is non-null,
- $\delta v = \otimes$, if the vector can be indifferently null or not.

5.1. Internal forces and displacements

As explained in Section 4, a variation of the augmented configuration is called *internal*² if it does not involve any variation of the external wrench acting on the object, that is

$$\delta w_e^b = 0 \quad (49)$$

5.2. External forces and displacements

On the contrary, a perturbation δy is called *external* if it involves a variation of the external wrench acting on the object, that is

$$\delta w_e^b = * \quad (50)$$

5.3. Pure squeeze

We define the *pure squeeze* case as the situation in which the hand performs a contact force variation on the object, without moving it. Synthetically, the pure squeeze occurs when

$$\begin{cases} \delta w_e^b = 0 \\ \delta f_{ch}^{co} = * \\ \delta u = 0 \end{cases} \quad (51)$$

5.4. Spurious squeeze

If the contact force variation is accompanied by an object displacement, we will talk of *spurious squeeze*. This case corresponds to having

$$\begin{cases} \delta w_e^b = 0 \\ \delta f_{ch}^{co} = * \\ \delta u = * \end{cases} \quad (52)$$

5.5. Kinematic grasp displacements

Still inside the subspace of internal forces and displacements, we identify those perturbations δy that do not involve any violation of the (rigid) kinematic contact constraints and, equivalently, those perturbations that do not involve any variation of the elastic potential energy stored in the contact springs, with respect to the reference configuration. This effect is obtained with solutions involving object displacements, but no contact force variation. Synthetically, we look for solutions in the form

$$\begin{cases} \delta w_e^b = 0 \\ \delta f_{ch}^{co} = 0 \\ \delta u = * \end{cases} \quad (53)$$

It is worth observing that, if we regard the displacements of the extremities of the contact springs as descriptive of elastic deformation of the grasped object and of the fingertip, the kinematic grasp, requiring a null variation of contact forces, implies a null variation of the object shape after the displacement. In this interpretation, the definition of *rigid object displacements* can be recovered. We want to remark that, in order to give consistent definitions, it is necessary to describe the contact forces in the object reference frame. Otherwise, describing forces in the absolute frame, not all the contact force variations necessarily correspond to contact spring (or object shape) deformations. A simple example of this fact is the pure rotation of the grasped object. To preserve its shape, the initial contact forces have to rotate

with it. Thus, if contact forces were expressed in an absolute frame, a variation would have occurred just because of the relative rotation.

5.6. Redundant motion of the hand

It could be of interest to find the motions of the hand not involving the object, e.g. to move the joints away from the limit values. This corresponds to the conditions

$$\begin{cases} \delta w_e^b = 0 \\ \delta f_{ch}^{co} = 0 \\ \delta u = 0 \\ \delta q = * \end{cases} \quad (54)$$

For the hand, this kind of movement represent to the so-called *nullspace motions*. Nevertheless, in the rest of the paper, we will not use this name in order to avoid confusion with other definitions.

5.7. External structural forces

In the subspace of external actions, it is possible to look for perturbations δy that are characterized by variations of contact forces δf_{ch}^{co} , but do not involve any modifications of the synergistic actions and references. This can be equivalently described as

$$\begin{cases} \delta w_e^b = * \\ \delta f_{ch}^{co} = * \\ \delta \eta = 0 \\ \delta \sigma_r = 0 \end{cases} \quad (55)$$

5.8. Resultant decomposition of the solution space

The previous definitions correspond to particular forms of solutions for the system (33). Since the basis Γ in (36) contains all the solution of the system, even these particular forms can be searched for there. This corresponds to obtaining a decomposition of the solution space as follows

$$\begin{bmatrix} \delta f_{ch}^{co} \\ \delta \tau \\ \delta \eta \\ \delta u \\ \delta C_{ac^b}^{co} \\ \delta C_{ac^o}^{co} \\ \delta q \\ \delta q_r \\ \delta \sigma \\ \delta w_e^b \\ \delta \sigma_r \end{bmatrix} = \begin{bmatrix} * & * & * & 0 & 0 & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & 0 & * \\ * & 0 & * & * & 0 & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & * & * \\ * & * & * & * & * & 0 & * \end{bmatrix} x \quad (56)$$

As we can see, each column (from the 2nd one to the 6th) corresponds to a definition given above. Exceptions are the first and the last column, that are introduced to complete the space of internal and external forces respectively.

The careful reader will observe that from some nullity patterns previous defined, taking into account (33), it can follow that other variables are necessarily null or non-null. This fact will be analyzed in Section 7, where all the constraints expressed by (33) will be taken into account to define a complete and consistent taxonomy (classification) of the solution space. Here we want to focus the attention only on the necessary conditions on which the definitions hold.

6. Block decomposition of the nullspace matrix

From linear algebra, it is known that every matrix $C \in \mathbb{R}^{r_c \times c_c}$, with $\rho_c = \text{rank}(C)$, can be expressed, by a Gauss–Jordan elimination, in its *reduced row echelon form* (RREF), via a suitable permutation matrix $\Pi \in \mathbb{R}^{r_c \times r_c}$, such that

$$\Pi C = \begin{bmatrix} U \\ 0 \end{bmatrix} \quad (57)$$

where $U \in \mathbb{R}^{\rho_c \times c_c}$ is a staircase matrix, and the zero block has consequent dimensions. It is worth recalling that the reduced row echelon form of a given matrix is unique. The reader can find other details about the RREF in Meyer (2000).

As explained in Bicchi et al. (1995), this particular form can be used to search nullity patterns, as in (56), inside general matrices. In order to explain the method, we show operatively the details of each step to find a basis for the internal solution of the system.

In later discussion, we will use the assumption to have access to a function `rref(X)` able to return the reduced row echelon form of its argument X .³ The procedure to define a basis for the internal system perturbations ($\delta w_e^b = 0$), consists of the steps:

1. Partitioning the basis of the solution space Γ , isolating the portion relative to the variable that we want to nullify (or not nullify). In the example of internal actions, from (37), we obtain

$$\delta y = \begin{bmatrix} \Gamma_{\tilde{w}_1} \\ \Gamma_w \\ \Gamma_{\tilde{w}_2} \end{bmatrix} x \quad (58)$$

where the symbols $\Gamma_{\tilde{w}_1}$ and $\Gamma_{\tilde{w}_2}$ correspond to blocks that are not relative to the external wrench variable. From this, we extract the block $\Gamma_w \in \mathbb{R}^{\rho_w \times c_y}$ relative to the external wrench.

2. Obtaining a permutation matrix. As a consequence of the properties of the RREF, detailed in Meyer (2000), running the function `rref` ($[\Gamma_w^T \mid I_{c_y}]$), where $I_{c_y} \in \mathbb{R}^{c_y \times c_y}$ is an identity matrix, gives the result $\begin{bmatrix} U_w \\ 0 \end{bmatrix} \Pi_w$, where $U_w \in \mathbb{R}^{\rho_w \times \rho_w}$, with $\rho_w = \text{rank}(\Gamma_w)$. The block $\Pi_w \in \mathbb{R}^{c_y \times c_y}$ is the permutation matrix such that $\Pi_w \Gamma_w^T = U_w$.

3. Using these results to find the new form ${}^1\Gamma \in \mathbb{R}^{r_\gamma \times c_\gamma}$ for the basis Γ as

$${}^1\Gamma = \Gamma \Pi_w^T = \begin{bmatrix} \Gamma_{\tilde{w}_1} \\ \Gamma_w \\ \Gamma_{\tilde{w}_2} \end{bmatrix} \quad (59)$$

where, in detail, ${}^1\Gamma_w = U_w^T$, ${}^1\Gamma_{\tilde{w}_1} = \Gamma_{\tilde{w}_1} \Pi_w^T$, and ${}^1\Gamma_{\tilde{w}_2} = \Gamma_{\tilde{w}_2} \Pi_w^T$.

By the way, we observe that the matrix ${}^1\Gamma$ is essentially a permutation and combination of the columns of Γ . This assures that the two matrices have the same column space, or, in other words, that both are a solution basis for the system (33).

With simple considerations, it is evident that (59) can be written as

$${}^1\Gamma = [{}^1\Gamma_{\text{ext}} \quad {}^1\Gamma_{\text{int}}] \quad (60)$$

where ${}^1\Gamma_{\text{ext}} \in \mathbb{R}^{r_\gamma \times \rho_w}$ and ${}^1\Gamma_{\text{int}} \in \mathbb{R}^{r_\gamma \times (c_\gamma - \rho_w)}$, which result a basis for *external* and *internal* solution respectively. Thus we have obtained the desired result.

Similarly, the procedure can be iterated to discover other subspaces defined in previous sections, searching nullity or non-nullity of the variable of interest each time.

7. Complete taxonomy of the grasp

In Section 5 some relevant kinds of system configuration variations are defined by imposing to some variables to be null or non-null. This approach can be further investigated, arriving to discern all the combinations of null or non-null variables, which can compose a feasible solution of the system (33).

To this aim, we will use a sort of *sieve of Eratosthenes* (Cojocaru et al., 2005): starting from all the possible combinations, we will proceed eliminating those forbidden by the equations composing (33).

To easily identify the forbidden solutions, it is useful to reorganize the system variables in *levels* as follows:

- *object level*, involving the external wrench δw_e^b , and the object displacement δu ;
- *hand/object interface level*, involving the contact forces $\delta f_{ch}^{c^o}$, the displacement of the object contact frame $\delta C_{ac^o}^{c^o}$, and the displacement of the hand contact frame $\delta C_{ac^h}^{c^o}$;
- *hand level*, involving the joint torques $\delta \tau$, the joint configuration δq , and the joint reference configuration δq_r ;
- *synergy level*, involving the synergy forces $\delta \eta$, the synergy configuration $\delta \sigma$, and synergy references $\delta \sigma_r$.

7.1. In-level constraints

Since a grasp performed by a synergistic underactuated hand is described by eleven variables, restricting the discussion to consider the possibility that they can only be null or non-null, all the possible combinations are $2^{11} = 2048$.

Table 2. Object level combinations.

	δw_e^b	δu
<i>O.A</i>	*	*
<i>O.B</i>	*	0
<i>O.C</i>	0	*
<i>O.D</i>	0	0

Table 3. Hand/object interface level combinations.

	δf_{ch}^{co}	δC_{aco}^{co}	δC_{ach}^{co}
<i>I.A</i>	*	*	*
<i>I.B</i>	*	*	0
<i>I.C</i>	*	0	*
<i>I.D</i>	*	0	0
<i>I.E</i>	0	*	*
<i>I.F</i>	0	*	0
<i>I.G</i>	0	0	*
<i>I.H</i>	0	0	0

However, each equation composing (33) can be interpreted as a constraint that the feasible nullity patterns have to satisfy. In other words, as we will show shortly, some combinations of null or non-null variables are forbidden by the equations of the system. These combinations can be found and eliminated in order to bring out all the combinations allowed.

In Table 2 we present all the possible (non-)nullity combinations involving the object level variables. By comparison with the equations grouped in (33), the result is that no combination is forbidden.

In Table 3, we represent all the possible combinations for the variables at the hand/object interface level. Taking into account (25), from an algebraic point of view, under the assumption that the contact stiffness matrix K_c is square and full rank, the result is that it is not possible to have a contact force without any contact frame displacements. Similarly, it is not possible to have a displacement of only one contact frame, without contact forces.

In other terms, from a physical point of view, the combination *I.D* corresponds to a contact force variation that appears without a corresponding deformation of the virtual contact spring. On the contrary, the combinations *I.F* and *I.G* describe situations where the contact spring is necessarily deformed, since one contact frame is fixed but the other moves; despite this fact, there are no contact forces. Therefore it is clear that the combinations *I.D*, *I.F* and *I.G*, corresponding to situations without physical meaning, are not allowed (highlighted in grey in Table 3).

Even for the *hand level*, whose (non-)nullity combinations are listed in Table 4, similar considerations hold. Equation (27) enables us to eliminate the cases *H.D*, *H.F* and *H.G*, if the joint stiffness matrix is square and full rank. As similarly discussed for the hand/object interface level,

Table 4. Hand level combinations.

	$\delta \tau$	δq	δq_r
<i>H.A</i>	*	*	*
<i>H.B</i>	*	*	0
<i>H.C</i>	*	0	*
<i>H.D</i>	*	0	0
<i>H.E</i>	0	*	*
<i>H.F</i>	0	*	0
<i>H.G</i>	0	0	*
<i>H.H</i>	0	0	0

Table 5. Synergy level combinations.

	$\delta \eta$	$\delta \sigma$	$\delta \sigma_r$
<i>S.A</i>	*	*	*
<i>S.B</i>	*	*	0
<i>S.C</i>	*	0	*
<i>S.D</i>	*	0	0
<i>S.E</i>	0	*	*
<i>S.F</i>	0	*	0
<i>S.G</i>	0	0	*
<i>S.H</i>	0	0	0

the combinations forbidden by (27) correspond to the presence of a joint torque variation without changing the joint or joint reference configuration (*H.D*), and, on the contrary, a variation of the joint actuation spring length, without a corresponding torque variation (*H.F* and *H.G*). With a similar physical motivation, we can eliminate the cases *S.D*, *S.F* and *S.G* from Table 5, under the assumption that the synergy stiffness matrix in (32) is square and full rank.

7.2. Intra-level constraints

With the previous discussion, we have considerably reduced the number of acceptable combinations. In fact, only four possibilities for the object level and five for all the other three levels remain. The total amount is now $4 \times 5^3 = 500$ possible combinations. But also the other equations of the system (33) can be taken into account, allowing us to eliminate some interconnection between the levels. In fact, from (8), we can observe that, if the grasp matrix bG is full row-rank, as it is usually the case, it is not possible to have an external wrench variation and a null variation of the contact force. This means that the groups *O.A* and *O.B* can be present in combination with the groups *I.A*, *I.B*, *I.C* and *I.D* (the last eliminated already in Table 3), but not with the others. The cases eliminated correspond to situations in which an external wrench would not have an effect on the contact. In other words, we restrict our analysis just to the cases of isostatic and hyperstatic equilibrium of the object, thus to the situation in which the number of independent constraints imposed on the object by the interaction with the hand is equal to or greater than its degrees of freedom.

Similarly, from (14) it follows that, in the presence of an object contact frame displacement δC_{ac}^{co} , there is always an object displacement δu . This corresponds to assuming that every object displacement has effect on at least one virtual contact spring.

From these considerations, it results that

- the group $O.A$ can occur only in conjunction with $I.A$ or $I.B$;
- the group $O.B$ can occur only in conjunction with $I.C$;
- the group $O.C$ can occur only in conjunction with $I.A$, $I.B$ or $I.E$;
- the group $O.D$ can occur only in conjunction with $I.C$ or $I.H$.

The variables of the interface and the hand level are related by (19). To satisfy this relationship, it is necessary that, if the variable δC_{ach}^{co} is non-zero, δq must also be non-zero. This is equivalent to imposing that all hand contact frame displacements are caused by a hand configuration variation. Instead, the possibility to have a joint movement not involving a hand contact frame displacement is allowed. This situation corresponds to reconfiguration motions of the hand. Then, it follows that

- the group $I.A$ can occur only in conjunction with $H.A$, $H.B$ or $H.E$;
- the group $I.B$ can occur in conjunction with $H.A$, $H.B$, $H.C$, $H.E$ or $H.H$ (all the hand level combinations allowed);
- the group $I.C$ can occur only in conjunction with $H.A$, $H.B$ or $H.E$;
- the group $I.E$ can occur only in conjunction with $H.A$, $H.B$ or $H.E$;
- the group $I.H$ can occur in conjunction with $H.A$, $H.B$, $H.C$, $H.E$ or $H.H$ (all the hand level combinations allowed).

Equations (29) and (31) relate the hand level with the synergy level variables. Assuming that the synergy matrix is full column rank, we eliminate all the combinations in which a joint reference displacement is present without a synergistic variation, and vice versa. The connections admitted can be summarized as

- the group $H.A$ can occur only in conjunction with $S.A$, $S.B$ or $S.E$;
- the group $H.B$ can occur only in conjunction with $S.C$ or $S.H$;
- the group $H.C$ can occur only in conjunction with $S.A$, $S.B$ or $S.E$;
- the group $H.E$ can occur only in conjunction with $S.A$, $S.B$ or $S.E$;
- the group $H.H$ can occur only in conjunction with $S.H$.

Finally, equation (23) relates three levels and impose that, if $\delta \tau$ is non-null, the variables δu , δf_{ch}^{co} and δq cannot be null at the same time or, in other words, impose that the

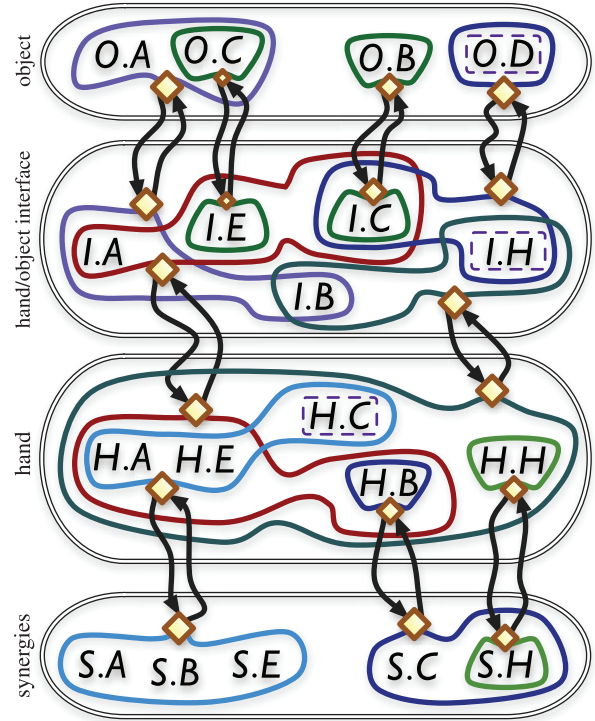


Fig. 2. Complete taxonomy of the grasp.

presence of a joint torque variation needs an explanation in a contact force or in a change of configuration. Taking into account the previously found constraints, it follows that it is not possible to get the condition $O.D$ in conjunction with $I.H$ and $H.C$ at the same time.

The total amount of the remaining combinations is 57, or 56 without considering the trivial solution composed by the combination $O.D$, $I.H$, $H.H$ and $S.H$, where no displacement of the system actually exists.

7.3. The taxonomic labyrinth

The set of all the acceptable combinations of the groups presented in Tables 2–5 defines the complete taxonomy of the grasp with soft synergy underactuation, sketched in Figure 2. Despite the fact that the figure appears as a labyrinth, it is not difficult to read it. It is possible to start indifferently from a group located at the top or at the bottom level. From this, one can move toward the next level, downward or upward respectively, moving from the *bubble* containing the starting group (e.g. $O.B$ or $S.E$). Each of these bubbles has a *square door* that allows access to another bubble, in the next level, containing all the admissible groups that can be chosen. From the new group, all the bubbles containing it define a new set of possible doors can be used. Throughout, it is forbidden to go back to the previous level.

From the chosen door, only new bubbles can be accessed, from where a new group can be chosen. The process is then

repeated, until arriving at the last level, with the sole limitation that it is not possible to have together all the elements *O.D.*, *I.H.* and *H.C.* (in the dotted box in Figure 2).

As a simple example, let us explain how to search for the *redundant motion of the hand*, defined in (54). From the conditions on the external wrench ($\delta w_e^b = 0$) and on the object displacement ($\delta u = 0$), by a direct comparison with Table 2, it follows that we can start from the element *O.D.*, on the top level of the Figure 2. The square door, connecting to the *hand/object interface* level, allows access to the groups *I.C.* and *I.H.* By their definitions, in Table 3, we note that the group *I.C.* requires a contact force variation ($\delta f_{ch}^{co} = *$), while for the group *I.H.* it holds that $\delta f_{ch}^{co} = 0$. By the definition in (54), the condition on the interface level ($\delta f_{ch}^{co} = 0$) is satisfied only choosing the group *I.H.* Considering the bubble containing it, the door allows every element of the hand level to be chosen. Again from (54), for the hand level variables we have to guarantee a non-null joint displacement ($\delta q = *$); thus only the groups *H.A.*, *H.B.* or *H.E.* are acceptable. To make things clearer, we take the group *H.E.*, corresponding to a particular redundant motion of the hand that does not require changing the joint torques. If no other condition is needed, finally we find that we can connect to *S.A.*, *S.B.* or *S.E.* By a comparison with Table 5, we can give an interpretation of these three cases. While the group *S.A.* correspond to the most general case, where all synergistic actuation is involved, the other two cases deserve a more detailed discussion. A solution of the system including the group *S.B.* indicates the presence of an equilibrium configuration of the system where the synergy reference configuration is held changing the generalized forces at the synergy level. The situation can occur in the case where the force transmission ratio between the joint torques and synergistic forces can vary with the synergy configuration. The presence of a solution including the group *S.E.*, indeed, points out the possibility of reconfiguring the synergy actuation without changing the generalized forces provided at the synergy level.

8. Other types of (under-)actuation

The *soft synergy* underactuation mechanism is currently attracting much attention and interest. However, we cannot ignore that it is not the only actuation mechanism worth studying. Over the obvious interest to study fully actuated robotic hands, other actuation approaches deserve attention and need suitable analysis tools, for instance *eigen-grasps* (Ciocarlie et al., 2007), parallel structure based (Birglen et al., 2008) and *adaptive synergies* (Catalano et al., 2012). Despite the fact that some parts of this paper concern strictly the *soft synergy underactuation*, the framework presented is general enough to cope with the above-mentioned underactuated situations.

The basic requirement to apply the methods we propose, is to write all the equations describing the kineto-static behavior of the particular system under investigation.

Casting together all the relationships, we basically obtain a new type of FGM, for which it is straightforward to apply GEROME-B and Find-X methods. Similarly, the decomposition of the nullspace by using the RREF, can be performed. However, it is necessary to pay attention when classifying the main categories of manipulation tasks. The definitions proposed, basically involve the variables from the object to the hand level, and these can be used for any kind of (under-)actuation considered. One exception is the subspace of the *external structural forces*. We can simply recover this concept considering instead the conditions

$$\begin{cases} \delta w_e^b = * \\ \delta f_{ch}^{co} = * \\ \delta \tau^* = 0 \\ \delta q^* = 0 \end{cases} \quad (61)$$

where δq^* and $\delta \tau^*$ are the generalized displacements and forces at the (under-)actuation level for the present case.

As a simple example, the FGM for the case in which no underactuation is considered on the hand can be obtained by taking (8), (14), (19), (23), (25), as in (121). The presence of elastic actuators can be introduced in the previous model, restoring (27), as in (125). For these examples, the block elements composing the canonical form of the FGM are described in 10. As a final example, the *eigen-grasp* case can be considered adding, to the system describing the fully actuated case (121), expressions of the type (29) and (31), but involving the joint displacements δq instead of the references δq_r , that is in the form

$$\begin{aligned} \delta q &= S(\sigma) \delta \sigma \\ \delta \eta &= S^T(\sigma) \delta \tau + \Sigma(\sigma, \tau) \delta \sigma \end{aligned} \quad (62)$$

After the proper FGM for the system in interest is built, taking into account the previous considerations, the approaches proposed in this paper can be adopted to obtain symbolic relationships from the input and the output variables (GEROME-B). Computationally efficient methods to find whole system perturbations (Find-X), and main categories of the manipulation tasks (nullspace block decomposition by the use of the RREF) can again be applied.

9. Numerical examples

9.1. Precision grasp

Let us consider the 2D example shown in Figure 3, where a two-fingered hand is grasping a circular object of radius R with its fingertips. Each finger is composed by two links of length L . Globally, the hand has four revolute joints $[j_1, \dots, j_4]$. Attached to the palm we fix an inertial frame $\{A\}$ with origin on the intersection between the axis of j_1 and the plane of the figure. On the k th contact point we introduce two reference frames $\{C_k^h\}$ and $\{C_k^o\}$, fixed with the hand and the object, respectively, and aligned with $\{A\}$ in the reference configuration.

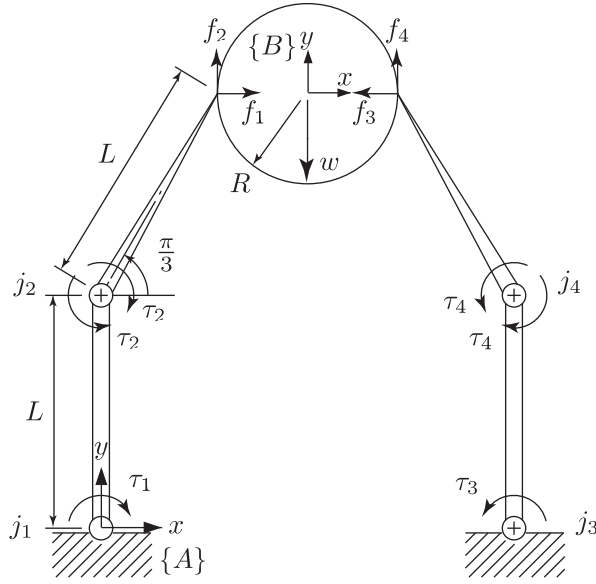


Fig. 3. Grasp of a circular object by a two-fingered robotic hand.

Attached to the object we pose a frame $\{B\}$ with origin in the center of the circle, initially parallel to $\{A\}$. The generic displacement of $\{B\}$ with respect to $\{A\}$ is described by the vector $\delta u^T = [\delta u_x^T, \delta u_y^T, \delta u_\alpha^T]^T$, where the first two elements indicate a linear displacement along the axes x and y , respectively, and the last one an angular displacement around the axis z . It is worth observing that the displacement of the object is described in coordinates expressed in $\{A\}$.

As numerical values for the parameters of the problem we consider $L = R = 1$ m, $k_q = 10$ N/rad, $k_c = 10$ N/m, $k_s = 10$ N/m, and we assume the preload $\bar{f}_{c^h} = [1, 1, -1, 1]^T$ N, as sketched in Figure 3. By the definition of the grasp matrix in body frame (6), we need to consider the contact model. In the case of *planar hard fingers*, we can admit that the hand can exert only forces along the axes x and y , without moments. In other words, the vectors describing the interaction at the contact is composed of two elements only. This implies that, to transform the local interaction in a complete wrench, for each contact we need a matrix

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (63)$$

The co-adjoint matrix in (6), by its definition in (91), can be computed considering the rotation matrices

$$R_{bc_1^o} = R_{bc_2^o} = I_3 \quad (64)$$

where I_3 is an identity matrix of dimension 3, and the vectors

$$d_{bc_1^o} = \begin{bmatrix} -R \\ 0 \\ 0 \end{bmatrix}, \quad d_{bc_2^o} = \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} \quad (65)$$

describe the position of the object contact frames $\{C_1^o\}$ and $\{C_2^o\}$, respectively, in frame $\{B\}$. Selecting only the components of the grasp matrix describing actions lying on the plane, the grasp matrix in body frame for the parameters values we choose, finally appears as

$${}^bG = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad (66)$$

In order to compute the hand Jacobian, we basically need to compute the spatial Jacobian matrices of the two contact frames, indicated as $\mathcal{J}_1$ and $\mathcal{J}_2$ respectively. Considering the vector $\omega = [0, 0, 1]^T$ of the joint axes, and the geometry of the system, from (102) we obtain the twists

$$\xi_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \xi_2 = \begin{bmatrix} L \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \xi_3 = \begin{bmatrix} 0 \\ -L - 2R \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \xi_4 = \begin{bmatrix} L \\ -L - 2R \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (67)$$

Setting to zero the value of the joint parameters q in the reference configuration of Figure 3, from (101) it follows that $\xi'_2 = \xi_2$ and $\xi'_4 = \xi_4$. Finally, from the transformation in (17), for the geometric parameters that we are using, the hand Jacobian matrix in object contact frames results in

$${}^oJ = \begin{bmatrix} -1.866 & -0.866 & 0 & 0 \\ 0.500 & 0.500 & 0 & 0 \\ 0 & 0 & -1.866 & -0.866 \\ 0 & 0 & -0.500 & -0.500 \end{bmatrix} \quad (68)$$

The derivatives of the Jacobian matrix, defined in (22), can be easily computed by (111) and (120), obtaining

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & -1 & 3.366 \\ 1 & -1 & 2.366 \\ 1 & 1 & 3.366 \\ 1 & 1 & 2.366 \end{bmatrix} \quad (69)$$

9.1.1. Perturbed configuration for fully actuated hand The aim here is to characterize some of the structural properties of the fully actuated grasp. Therefore, we perform the decomposition of the nullspace matrix as in (56). All the above definitions hold defining a synergy matrix $S = I_{\#q}$, or equivalently using the FGM for a fully actuated hand in (121). In this case, all the previous definitions given in Section 5 can be recovered, with the only exception of the external structural forces, where we have to impose the nullity of the hand level variables δq and $\delta \tau$, instead of $\delta \sigma_r$.

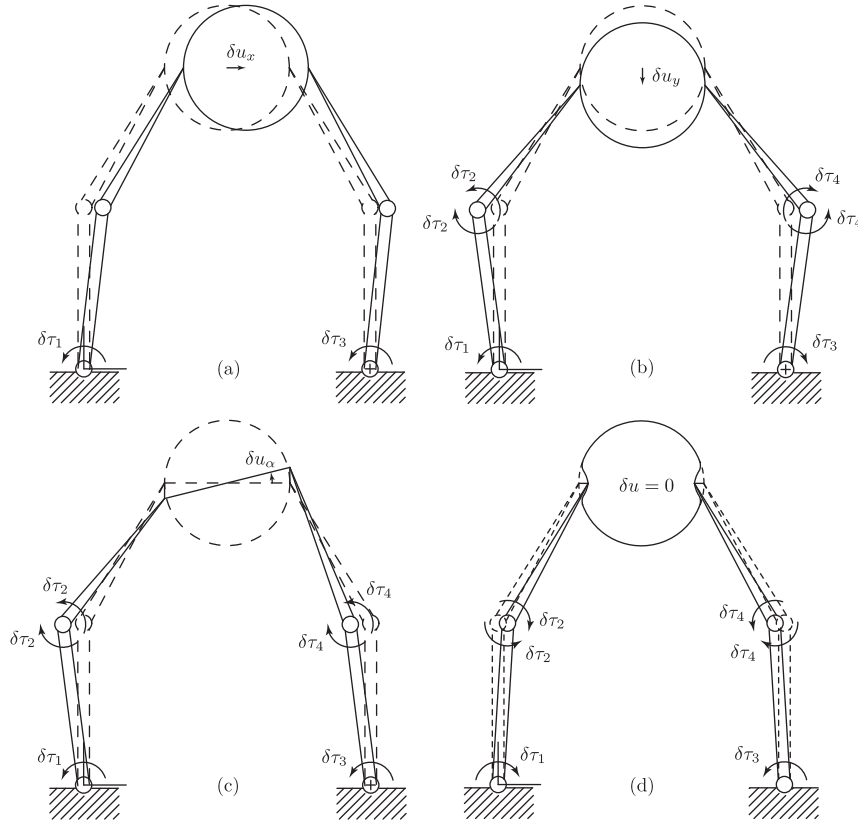


Fig. 4. Graphic representations of numerical results for the internal precision grasp variations. (a–c) represent the *kinematic grasp displacements*, i.e. the object movements that the hand can impose to the object without modifying the contact forces; (d) represents a *pure squeeze*, i.e. a contact force variation obtained without moving the grasped object.

and $\delta\eta$. In this example, the nullspace of the FGM has dimension $\sharp w + \sharp q = 7$. Within this seven-dimensional subspace, the subspace of *kinematic grasp displacements* has dimension three, the *pure squeeze* subspace has dimension one: together they complete the subspace of *internal system perturbations* (no *spurious squeeze* and *redundant motion of the hand* are present). The *external structural force* subspace has null dimension, meaning that every external perturbation needs a counterpart on the hand actuation to preserve the equilibrium of the system. For the *kinematic grasp displacements*, a finite displacement of the object $\delta u_x = 0.001$ in the x -direction, as in Figure 4(a), is possible with the corresponding variations in joint torques and angles

$$\begin{aligned} \delta\tau &= [0.001 \ 0 \ 0.001 \ 0]^T \\ \delta q &= [-0.001 \ 0.001 \ -0.001 \ 0.001]^T \end{aligned} \quad (70)$$

For an object displacement $\delta u_y = -0.001$, Figure 4(b), the corresponding variations in the joint torques and joint angles are

$$\begin{aligned} \delta\tau &= [-0.0010 \ -0.0027 \ 0.0010 \ 0.0027]^T \\ \delta q &= [0.0017 \ -0.0037 \ -0.0017 \ 0.0037]^T \end{aligned} \quad (71)$$

For an object rotation $\delta u_\alpha = 0.001$, Figure 4(c), the variations in the joint torques and joint angles are then

$$\begin{aligned} \delta\tau &= [0.0033 \ 0.0040 \ 0.0033 \ 0.0040]^T \\ \delta q &= [0.0017 \ -0.0037 \ 0.0017 \ -0.0037]^T \end{aligned} \quad (72)$$

Finally, the variations of joint torques and joint angles for the *squeeze*, represented in Figure 4(d), are

$$\begin{aligned} \delta\tau &= [-0.0018 \ -0.0010 \ 0.0018 \ 0.0010]^T \\ \delta q &= [-0.0001 \ 0.0001 \ 0.0001 \ -0.0001]^T \end{aligned} \quad (73)$$

9.1.2. A synergy in the precision grasp Having inspected the properties of the fully actuated system, we can then evaluate the effect of underactuation on the system properties by introducing only one synergy. The synergies proposed both here and later in section 9.2.2, are generated (by a reverse-engineering process) after the nullspace decomposition for the fully actuated system with elastic joints has been carried out, by properly combining some solutions in order to synthesize a desired task. GEROME-B or Find-X can be profitably employed here to verify the results.

Introducing a single column synergy matrix as follows

$$S = [-0.2866 \ 0.0034 \ 0.2866 \ -0.0034]^T \quad (74)$$

we can write the FGM introduced in (43), for the synergistic underactuated case. Decomposing the nullspace, we can verify that we are left one pure squeeze, and three external forces and displacements, which together complete the basis of the nullspace. To elicit the nature of this synergy, we can evaluate the system response to a predictable input. To this aim, we can use the Find-X method or GEROME-B, obtaining, of course, the same results. Since the pure squeeze still exists for this system, with the above synergy it is always possible for a certain $\delta\sigma_r$ to keep the object in its starting configuration ($\delta u = 0$) for a null external perturbation $\delta w_e^b = [0, 0, 0]^T$. In fact, by applying $\delta\sigma_r = 1$, we get $\delta f_c = [0.9039, 0, -0.9039, 0]$ with $\delta u = [0, 0, 0]^T$. This is because S was defined as proportional to the (single) column of the pure squeeze found for the fully actuated system.

9.2. Power grasp

As a second test case, we consider a square of side $2L$ grasped by a spider-like hand composed of two fingers, and a total of eight joints $[j_1, \dots, j_8]$. The notation is similar to the previous example. Figure 5 shows the starting configuration of the hand and the initial contact preload. All the force vectors have unit lengths and the directions depicted. We consider every contact frame initially parallel to the frame $\{A\}$. The contact points, with respect to the frame $\{B\}$, are placed respectively in

$$d_{bc_1^o} = \begin{bmatrix} -L \\ -L \\ 0 \end{bmatrix}, \quad d_{bc_2^o} = \begin{bmatrix} -L \\ L \\ 0 \end{bmatrix},$$

$$d_{bc_3^o} = \begin{bmatrix} L \\ -L \\ 0 \end{bmatrix}, \quad d_{bc_4^o} = \begin{bmatrix} L \\ L \\ 0 \end{bmatrix} \quad (75)$$

As in the previous example, we consider the numerical values $L = 1$ m, $k_q = 10$ N/rad, $k_c = 10$ N/m, $k_s = 10$ N/m. For hard finger-like contacts, for which we can use (63), the grasp matrix in body frame results in

$${}^bG = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & -1 & 1 \end{bmatrix} \quad (76)$$

From the geometry in Figure 5, considering that every joint has $\omega = [0, 0, 1]^T$, we obtain the twists

$$\xi_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \xi_2 = \begin{bmatrix} L \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \xi_3 = \begin{bmatrix} L \\ -L \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \xi_4 = \begin{bmatrix} (5/2)L \\ (1/2)L \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

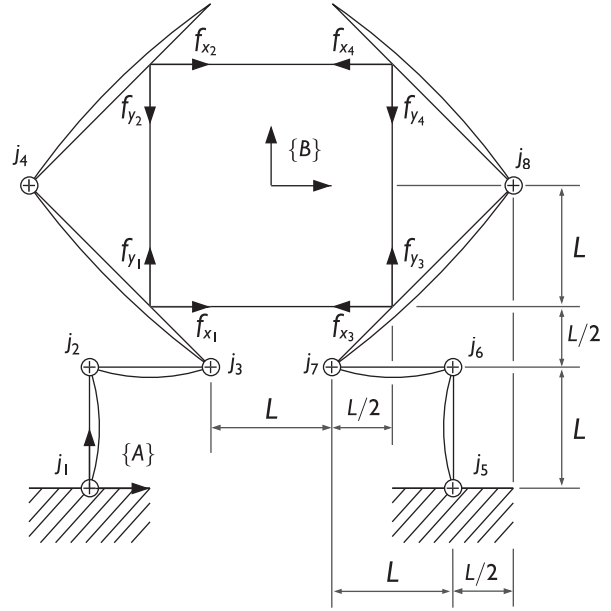


Fig. 5. Grasp of a square by a two fingered spider-like hand.

$$\xi_5 = \begin{bmatrix} 0 \\ -3L \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \xi_6 = \begin{bmatrix} L \\ -3L \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \xi_7 = \begin{bmatrix} L \\ -2L \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \xi_8 = \begin{bmatrix} (5/2)L \\ -(7/2)L \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (77)$$

From (77), considering a null value for the joint parameters q in the reference configuration, according to (17), the hand Jacobian results in

$${}^cJ = \begin{bmatrix} -1.500 & -0.500 & -0.500 & 0 & 0 & 0 & 0 & 0 \\ 0.500 & 0.500 & -0.500 & 0 & 0 & 0 & 0 & 0 \\ -3.500 & -2.500 & -2.500 & -1 & 0 & 0 & 0 & 0 \\ 0.500 & 0.500 & -0.500 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.500 & -0.500 & -0.500 & 0 \\ 0 & 0 & 0 & 0 & -0.500 & -0.500 & 0.500 & 0 \\ 0 & 0 & 0 & 0 & -3.500 & -2.500 & -2.500 & -1 \\ 0 & 0 & 0 & 0 & -0.500 & -0.500 & 0.500 & -1 \end{bmatrix} \quad (78)$$

The derivatives of the Jacobian matrix, computed by the rules (111) and (120), appear as follows

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & -2 & -3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3 & -2 & -3 & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & -2 & 3 \\ 0 & -2 & 3 \\ 0 & -2 & 1 \\ -1 & -1 & 2 \\ 0 & 2 & 3 \\ 0 & 2 & 3 \\ 0 & 2 & 1 \\ -1 & 1 & 2 \end{bmatrix} \quad (79)$$

9.2.1. Perturbed configuration for fully actuated hand By employing the matrices presented in the previous section,

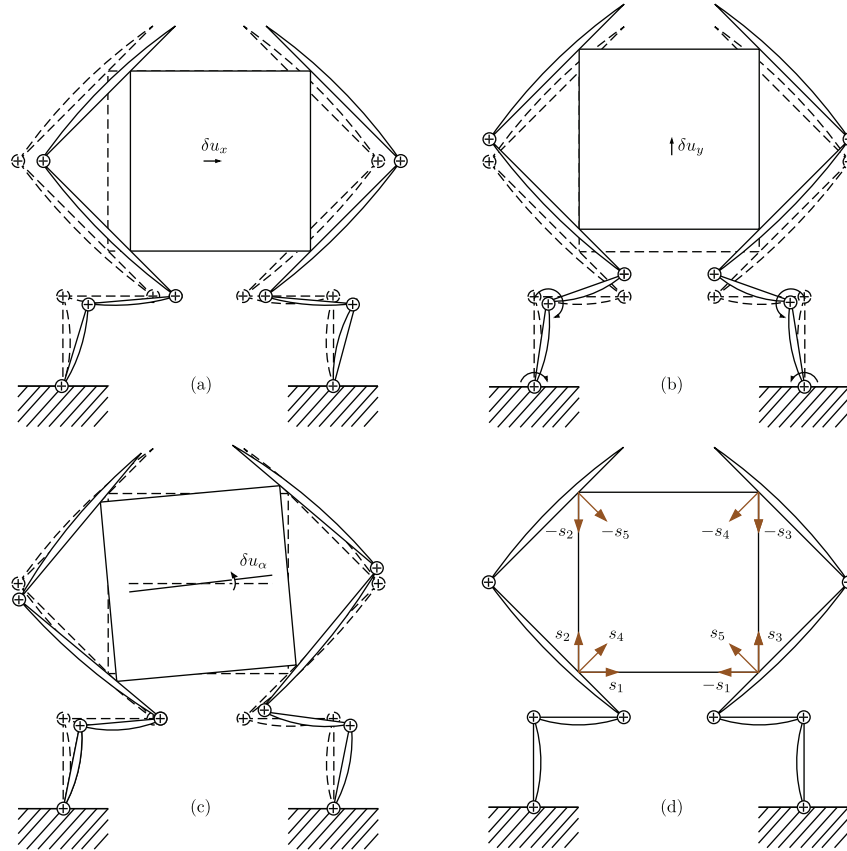


Fig. 6. Graphic representations of numerical results for the internal power grasp variations. (a–c) represent the *kinematic grasp displacements*, i.e. the object movements that the hand can impose to the object without modifying the contact forces; (d) represents the *pure squeeze*. In this case the two diagonals of the square and three of the four sides (left, bottom and right) of the object can be individually squeezed, without moving the object. This is not possible for the top side.

we can build the FGM for the fully actuated case as in (121). In this case we obtain that $A \in \mathbb{R}^{43 \times 54}$ and, consequently, the basis for the $N(A)$ is $\Gamma \in \mathbb{R}^{54 \times 11}$. Performing the decomposition of the nullbasis Γ , as in (56), we obtain that the *kinematic grasp* subspace has dimension three, the *pure squeeze* subspace has dimension five, and together they complete the subspace of internal forces and displacements. Conversely, the subspace of external forces and displacements has dimension three.

For the *kinematic grasp displacements*, simulation results show that it is possible to have a finite displacement of the object $\delta u_x = 0.001$, as in Figure 6(a), with no torque variations, but with a joint angle displacement of

$$\delta q = 10^{-3} [-1 \ 1 \ 0 \ 0 \ -1 \ 1 \ 0 \ 0]^T \quad (80)$$

For $\delta u_y = 0.001$, the corresponding variations in the joint torques and joint angles are

$$\begin{aligned} \delta \tau &= 10^{-3} [-2 \ -2 \ 0 \ 0 \ 2 \ 2 \ 0 \ 0]^T \\ \delta q &= 10^{-3} [0 \ 1 \ -1 \ 0 \ 0 \ -1 \ 1 \ 0]^T \end{aligned} \quad (81)$$

with pictorial representations sketched in Figure 6(b). For an object rotation $\delta u_\alpha = 0.001$, Figure 6(c), the variations in the joint torques and joint angles are then

$$\begin{aligned} \delta \tau &= 10^{-3} [3 \ 3 \ 0 \ 0 \ 3 \ 3 \ 0 \ 0]^T \\ \delta q &= 10^{-3} [-1.5 \ 1 \ 1.5 \ 0 \ -1.5 \ 1 \ 1.5 \ 0]^T \end{aligned} \quad (82)$$

The five possibilities for the pure squeeze are sketched in Figure 6(d), where the i th couple of forces s_i and $-s_i$ correspond to the i th achievable squeeze.

9.2.2. A synergy in the power grasp Underactuating the hand with the following single-column synergy matrix

$$S = \begin{bmatrix} -0.6500 & 0 & -0.3200 & -0.4000 \\ 0.6500 & 0 & 0.3200 & 0.4000 \end{bmatrix}^T \quad (83)$$

we get one possibility for the squeeze and three external solutions. To unveil the properties of this synergy we can study how the system responds to different inputs. In the absence of an external interaction, $\delta w_e^b = 0$, and with an

unitary synergistic actuation, $\delta\sigma_r = 1$, contact forces and object displacements become

$$\delta f_{ch}^{co} = \begin{bmatrix} 0.5043 & 0.5043 & 0.5043 & -0.5043 \\ -0.5043 & 0.5043 & -0.5043 & -0.5043 \end{bmatrix}^T \quad (84)$$

$$\delta u = [0 \quad 0 \quad 0]^T \quad (85)$$

indicating that we are purely squeezing the object along both diagonals. It is worth noting that the above synergy was constructed by summing two columns of the pure squeeze relative to the fully actuated system.

10. Conclusions

In this work, we presented a framework to model and study the structural properties of a grasp, performed by a general robotic hand, in a quasi-static setting. Synergistic underactuation is included in the analysis, as well as compliant characteristics at various levels, in the fingerpads and in the actuation system, both for joints and synergies. The mathematical model presented considers also the derivatives of the Jacobian matrix and employs an object-centric formulation of the grasp matrix, since other definitions may lead to potentially misleading conclusions. Considering all the system equations, the FGE and the FGM are defined. An extension of the Gauss–Jordan elimination method, called GEROME-B, able to act on block partitioned matrices, is presented. Thanks to this, the FGMc is found, corresponding to a new set of equations where each *dependent variable* of the system is a function only of the *independent* ones. Main relevant properties of the grasp can be discovered in this form, as a basis for the *controllable internal forces* and the *grasp compliance*. It is important to note that their expressions can be found even in symbolic form, as a functions of the basic matrices of the system, such as the Jacobian matrix, the grasp matrix, and so on. As GEROME-B acts in symbolic form, another method, called Find-X, is presented as a numerical efficient counterpart to find input–output relationships between the variables of the system.

Other more specific categories of system perturbations are later defined, in terms of nullity or non-nullity of some variables, such as the *pure squeeze*, when the hand acts on the object without changing its configuration, or *kinematic object displacements*, when the object is rigidly moved. An algorithm based on the extensive employment of the RREF has been presented, that allows to discover them inside the solution space of the system, and even to find the particular inputs that the system needs to perform them.

The search of feasible solutions was further discussed, considering all the equations of the system as constraints for the feasible (non-)nullity patterns of variables, until arriving at a complete taxonomy of the grasp performed by synergistic underactuated hands. In order to assess the validity of the proposed method, some numerical tests have been presented for two different grasps: a precision grasp and a power grasp. These demonstrated (a) the importance

of the terms originated by the derivatives of the Jacobian matrix in the presence of a preload contact force to obtain physically meaningful results, and (b) the influence of the synergistic actuation on the structural properties of a grasp configuration.

Funding

This work was supported by the European Commission (CP-IP grant number 248587, “THE Hand Embodied”), within the FP7-2007-2013 program “Cognitive Systems and Robotics” (ERC advanced grant number 291166, “SoftHands – A Theory of Soft Synergies for a New Generation of Artificial Hands”) and within the FP7-ICT-2011-9 program “Cognitive Systems” (grant number 600918 “PaCMan – Probabilistic and Compositional Representations of Objects for Robotic Manipulation”).

Notes

1. The term synergies in human hand control studies refers to various phenomena and in ways that reflect the level at which the analysis is performed, ranging from the common divergent inputs to multiple neurons at the neural level, the activity of concurrently active motor units at the muscle level, to covariation patterns among digit joints during reach-to-grasp and manipulation tasks at the kinematic level (Santello et al., 2013). In the present paper, as usually common in the robotics community, we will adopt the kinematic level definition of synergies, e.g. finger joint covariation patterns in grasping and manipulation tasks.
2. It is worth remarking that our definition of *internal forces* implies the use of object frame components for the external wrench. This choice allows to filter out the influence of object motions that, otherwise, would have come across in a spatial description of the same quantity.
3. This is a typical situation using the most popular computational platforms, e.g. `rref(X)` in MATLAB and `RowReduce(X)` in Mathematica.

References

- Angelaki DE and Soechting JF (1993) Non-uniform temporal scaling of hand and finger kinematics during typing. *Experimental Brain Research* 92(2): 319–329.
- Bicchi A (1994) On the problem of decomposing grasp and manipulation forces in multiple whole-limb manipulation. *International Journal of Robotics and Autonomous Systems* 13: 127–147.
- Bicchi A (2000) Hands for dextrous manipulation and powerful grasping: A difficult road towards simplicity. *IEEE Transactions on Robotics and Automation* 6: 652–662.
- Bicchi A, Melchiorri C and Balluchi D (1995) On the mobility and manipulability of general multiple limb robotic systems. *IEEE Transactions on Robotics and Automation* 11(2): 215–228.
- Birglen L, Laliberté T and Gosselin C (2008) *Underactuated robotic hands*, (Springer Tracts in Advanced Robotics, Vol. 40). Berlin: Springer Verlag.
- Brown CY and Asada HH (2007) Inter-finger coordination and postural synergies in robot hands via mechanical implementation of principal components analysis. In: *2007 IEEE/RSJ*

- International conference on intelligent robots and system, San Diego, CA, 29 October-2 November 2007, pp.2877–2882.
- Castellini C and van der Smagt P (2013) Evidence of muscle synergies during human grasping. *Biological Cybernetics* 107(2): 233–245. 10.1007/s00422-013-0548-4.
- Catalano MG, Grioli G, Serio A, Farnioli E, Piazza C and Bicchi A (2012) Adaptive synergies for a humanoid robot hand. In: *IEEE-RAS International conference on humanoid robots*, Osaka, Japan, 29 November-1 December 2012, pp.7–14.
- Ciocarlie M, Goldfeder C and Allen P (2007) Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem. In: *Proceedings of the robotics: science and systems 2007 workshop – sensing and adapting to the real world*, Atlanta, GA, USA, 27–30 June 2007.
- Cojocar A, Bruce JW and Murty R (2005) *An Introduction to Sieve Methods and Their Applications*. Cambridge: Cambridge University Press.
- Davidson J and Hunt K (2004) *Robots and Screw Theory: Applications of Kinematics and Statics to Robotics*. Oxford University Press on Demand. Available at: <http://books.google.it/books?id=OQq67Tr7D0cC>.
- Fish J and Soechting JF (1992) Synergistic finger movements in a skilled motor task. *Experimental Brain Research* 91(2): 327–334.
- Gabbicini M, Bicchi A, Prattichizzo D and Malvezzi M (2011) On the role of hand synergies in the optimal choice of grasping forces. *Autonomous Robots [Special Issue on RSS2010]* 31(2–3): 235–252. 10.1007/s10514-011-9244-1.
- Gabbicini M, Farnioli E and Bicchi A (2012) Grasp and manipulation analysis for synergistic underactuated hands under general loading conditions. In: *International conference of robotics and automation (ICRA 2012)*, Saint Paul, MN, USA, 14–18 May 2012. pp. 2836–2842.
- Grebstein M, Chalon M, Friedl W, Haddadin S, Wimböck T, Hirzinger G and Siegwart R (2012) The hand of the DLR hand arm system: Designed for interaction. *International Journal of Robotic Research* 31(13): 1531–1555. Available at: <http://dblp.uni-trier.de/db/journals/ijrr/ijrr31.html#GrebsteinCFHWHS12>.
- Grioli G, Catalano M, Silvestro E, Tono S and Bicchi A (2012) Adaptive synergies: An approach to the design of underactuated robotic hands. In: *IROS/RIS International conference on intelligent robots and systems*, pp.1251–1256. Available at: <https://dl.dropbox.com/u/5583591/SecondHand.pdf>.
- Hall B (2003) *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. (Graduate Texts in Mathematics). Springer. Available at: <http://books.google.it/books?id=m1VQi8HmEwcC>.
- Jacobsen S, Wood J, Knutt D and Biggers K (1984) The Utah/MIT dextrous hand: Work in progress. *The International Journal of Robotics Research* 3(4): 21–50.
- Latash ML, Krishnamoorthy V, Scholz JP and Zatsiorsky VM (2005) Postural synergies and their development. *Neural plasticity* 12(2–3): 119–130; discussion 263–272. 10.1155/NP.2005.119. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2565460&tool=pmcentrez&rendertype=abstract>.
- Lovchik C and Diftler M (1999) The robonaut hand: A dextrous robot hand for space. In: *Proceedings of 1999 IEEE International Conference on Robotics and Automation*, Vol. 2. Detroit, MI, USA, 10–15 May 1999, pp. 907–912. 10.1109/ROBOT.1999.772420.
- Meyer CD (2000) *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA: Society for Industrial and Applied Mathematics. Available at: <http://matrixanalysis.com/DownloadChapters.html>.
- Murray R, Li Z and Sastry S (1994) *A Mathematical Introduction to Robotic Manipulation*. CRC Press. Available at: http://books.google.it/books?id=D_PqGKR07oIC.
- Prattichizzo D, Malvezzi M and Bicchi A (2011) On motion and force controllability of grasping hands with postural synergies. In: *Robotics: Science and Systems VI*. Zaragoza, Spain: The MIT Press, pp.49–56.
- Santello M, Baud-Bovy G and Jörntell H (2013) Neural bases of hand synergies. *Frontiers in Computational Neuroscience* 7: 23. 10.3389/fncom.2013.00023.
- Santello M, Flanders M and Soechting J (1998) Postural hand synergies for tool use. *The Journal of Neuroscience* 18(23): 10105–10115.
- Sciavicco L and Siciliano B (2000) *Modelling and Control of Robot Manipulators*. London: Springer.
- Selig JM (2005.) *Geometric Fundamentals of Robotics*, 2nd edition (Monographs in Computer Science). New York: Springer.
- Soechting JF and Flanders M (1997) Flexibility and repeatability of finger movements during typing: Analysis of multiple degrees of freedom. *Journal of Computational Neuroscience* 4(1): 29–46. Available at: <http://dblp.uni-trier.de/db/journals/jcns/jcns4.html#SoechtingF97>.
- Shadow Robot Company Ltd (2009) Shadow hand. Retrieved from Shadowhand: <http://shadowhand.com>.
- Thakur PH, Bastian AJ and Hsiao SS (2008) Multidigit movement synergies of the human hand in an unconstrained haptic exploration task. *Journal of Neuroscience* 28(6): 1271–1281. 10.1523/jneurosci.4512-07.2008.

Appendix A. Homogeneous transformations and adjoint operators

Given a point $p \in \mathbb{R}^3$, we define its homogeneous form as

$$\bar{p} = \begin{bmatrix} p \\ 1 \end{bmatrix} \quad (86)$$

Considering two reference frames $\{A\} = \{O_a; x_a, y_a, z_a\}$ and $\{B\} = \{O_b; x_b, y_b, z_b\}$, let $d_{ab} \in \mathbb{R}^3$ be the vector going from O_a to O_b , and $R_{ab} \in \mathbb{R}^{3 \times 3}$ be the rotation matrix describing the configuration of $\{B\}$ with respect to $\{A\}$. The descriptions of the point p in the two reference frames are related by the expression

$$\bar{p}_a = \begin{bmatrix} R_{ab} & d_{ab} \\ 0 & 1 \end{bmatrix} \bar{p}_b := g_{ab} \bar{p}_b \quad (87)$$

where we implicitly defined the *homogeneous transformation matrix* as

$$g_{ab} = \begin{bmatrix} R_{ab} & d_{ab} \\ 0 & 1 \end{bmatrix} \quad (88)$$

With the symbol $\hat{\bullet}$ we indicate the *cross product matrix*, thus such that, for $x, y \in \mathbb{R}^3$, it holds

$$\hat{x}y = x \times y \quad (89)$$

The *adjoint matrix* and the *co-adjoint matrix*, whose properties are explained in Section 2, have the following expressions

$$\text{Ad}_{g_{ab}} = \begin{bmatrix} R_{ab} & \hat{d}_{ab}R_{ab} \\ 0 & R_{ab} \end{bmatrix} \quad (90)$$

$$\text{Ad}_{g_{ab}}^{-T} = \begin{bmatrix} R_{ab} & 0 \\ \hat{d}_{ab}R_{ab} & R_{ab} \end{bmatrix} \quad (91)$$

Introducing the operator $(\bullet)$ such that $(\hat{x})^\vee = x$, by direct calculation, from (89) it follows that

$$x \times y = (\hat{x}\hat{y} - \hat{y}\hat{x})^\vee \quad (92)$$

Expanding the definition of the operator $\hat{\bullet}$ to act on twists $\xi = [v^T, \omega^T]^T \in se(3)$ as

$$\hat{\xi} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} \quad (93)$$

in analogy with (92), we define the *Lie bracket* as an operator acting on two elements of $se(3)$ according with the rule

$$[\xi_1; \xi_2] = (\hat{\xi}_1\hat{\xi}_2 - \hat{\xi}_2\hat{\xi}_1)^\vee \quad (94)$$

It is worth observing that the result of the Lie bracket of two twists it is also a twist. The operation described in (94), can be realized by a proper multiplication operation in the form

$$[\xi_1; \xi_2] = \begin{bmatrix} \hat{\omega}_1 & \hat{v}_1 \\ 0 & \hat{\omega}_1 \end{bmatrix} \begin{bmatrix} v_2 \\ \omega_2 \end{bmatrix} := \text{ad}(\xi_1)\xi_2 \quad (95)$$

where we introduced the *adjoint matrix* (in lower case), also called *Lie adjoint* hereafter, as

$$\text{ad}(\xi) = \begin{bmatrix} \hat{\omega} & \hat{v} \\ 0 & \hat{\omega} \end{bmatrix} \quad (96)$$

For later use, we show that between the adjoint matrix (Ad) and the Lie adjoint (ad), the following relationships hold

$$\text{Ad}_{e^{\hat{\xi}}} = e^{\text{ad}(\hat{\xi})} \quad (97)$$

$$\text{Ad}_{e^{\hat{\xi}_1}} e^{\text{ad}(\hat{\xi}_2)} \text{Ad}_{e^{\hat{\xi}_1}}^{-1} = \text{ad}(\text{Ad}_{e^{\hat{\xi}_1}} \hat{\xi}_2) \quad (98)$$

More details about the exponential of a twist are presented in the next section.

Appendix B. Spatial Jacobian matrix

For the reader's convenience, we briefly present how to compute the spatial Jacobian matrix of serial manipulator, that is a matrix ${}^sJ(q) \in \mathbb{R}^{6 \times n_q}$ such that

$$\xi_{ae}^a = {}^sJ(q)\dot{q} \quad (99)$$

where the vector $\xi_{ae}^a \in \mathbb{R}^6$ is the twist, with components in $\{A\}$, of a frame of reference that is instantly superimposed to the absolute frame, but moves rigidly with the movement of

the end-effector. It is useful to remember that the Jacobian matrix can be written as

$$\mathcal{J}(q) = [\xi_1 \quad \xi_2' \quad \dots \quad \xi_n'] \quad (100)$$

where

$$\xi_k' = \text{Ad}_{(e^{\hat{\xi}_1 q_1} \dots e^{\hat{\xi}_{k-1} q_{k-1}})} \xi_k \quad (101)$$

It is worth noting that the terms ξ_k in (101) correspond to the k th joint twist, and ξ_k' are their transformation imposed by the robot configuration. The twists for a prismatic and a revolute joint have the forms

$$\xi_p = \begin{bmatrix} v_p \\ 0 \end{bmatrix}, \quad \xi_r = \begin{bmatrix} -\omega_r \times p_r \\ \omega_r \end{bmatrix} = \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \quad (102)$$

where $v_p \in \mathbb{R}^3$ is a unit vector pointing in direction of the axis of the prismatic joint, $\omega_r \in \mathbb{R}^3$ is a unit vector identifying the axis of the revolute joint and p_r is a point on this axis. It is possible to demonstrate that the exponential of the twists in (102), necessary to compute (101) and (100), are given by

$$e^{\hat{\xi}_p q_p} = \begin{bmatrix} I & v_p q_p \\ 0 & 1 \end{bmatrix}, \quad e^{\hat{\xi}_r q_r} = \begin{bmatrix} e^{\hat{\omega}_r q_r} (I - e^{\hat{\omega}_r q_r}) (\omega_r \times v_r) + \omega_r \omega_r^T v_r q_r & \\ 0 & 1 \end{bmatrix} \quad (103)$$

where

$$e^{\hat{\omega}_r q_r} = I + \hat{\omega}_r \sin q_r + \hat{\omega}_r^2 (1 - \cos q_r) \quad (104)$$

Equation (104) corresponds to a rotation of an angle q_r around an axis identified by the vector ω_r , and it is known as *Rodrigues's formula*.

Appendix C. Derivative terms of the Jacobian matrix

Appendix C1. Derivative of the Jacobian matrix with respect to the hand configuration

The definitions given in Appendices 10 and 10 are useful now to calculate the derivatives of the Jacobian matrix, seen in (22). Other details about the results provided in this section can be found in Murray et al. (1994), Hall (2003) and in Selig (2005). For the sake of simplicity, but without loss of generality, in this section we will find the derivatives of a Jacobian matrix of a serial robot, such as, for example, a finger of the hand. The reader will easily extend the results to the whole hand Jacobian matrix.

Let us consider first the derivative with respect to the hand configuration, that is the variable q , of the Jacobian matrix describing the twist of the i th contact frame, as defined in (17). In other words we are looking for a direct calculation method for

$$Q_i = \frac{\partial {}^{c_i^o}J_i^T(q, u) f_{c_i^h}^{c_i^o}}{\partial q} \quad (105)$$

or equivalently

$$Q_i = \begin{bmatrix} \frac{\partial {}^{c_o}J_i^T(q,u)f_{c_i^h}^{c_o}}{\partial q_1} & \frac{\partial {}^{c_o}J_i^T(q,u)f_{c_i^h}^{c_o}}{\partial q_2} & \dots & \frac{\partial {}^{c_o}J_i^T(q,u)f_{c_i^h}^{c_o}}{\partial q_{n_i}} \end{bmatrix} \quad (106)$$

where n_i is the number of joints affecting the motion of the contact frame. From (17) it follows that the j th term in (106) can be even expressed as

$$\frac{\partial \left(B_i^T \text{Ad}_{g_{c_i^b}} \text{Ad}_{g_{ba}(u)} \mathcal{J}_i(q) \right)^T f_{c_i^h}^{c_o}}{\partial q_j} = \left(\frac{\partial \mathcal{J}_i(q)}{\partial q_j} \right)^T \text{Ad}_{g_{c_i^b}^T(u)}^T B_i f_{c_i^h}^{c_o} \quad (107)$$

where we used the fact that the operations of transposition and derivative with respect to a scalar quantity commute. Considering (107), the problem is now reduced to find the derivative of the spatial Jacobian for a serial robot, as in (100).

It is possible to approach the problem by searching the derivative of the twists composing it, that is

$$\begin{aligned} \frac{\partial \xi_k'}{\partial q_j} &= \frac{\partial}{\partial q_j} \left(\text{Ad}_{(e^{\hat{\xi}_1 q_1} \dots e^{\hat{\xi}_{k-1} q_{k-1}})} \xi_k \right) \\ &= \frac{\partial}{\partial q_j} \left(\text{Ad}_{e^{\hat{\xi}_1 q_1}} \dots \text{Ad}_{e^{\hat{\xi}_j q_j}} \dots \text{Ad}_{e^{\hat{\xi}_{k-1} q_{k-1}}} \xi_k \right) \end{aligned} \quad (108)$$

Remembering the property in (97), from (108) it directly follows that

$$\begin{aligned} \frac{\partial \xi_k'}{\partial q_j} &= \text{Ad}_{e^{\hat{\xi}_1 q_1}} \dots \frac{\partial e^{\text{ad}(\xi_j) q_j}}{\partial q_j} \dots \text{Ad}_{e^{\hat{\xi}_{k-1} q_{k-1}}} \xi_k \\ &= \text{Ad}_{e^{\hat{\xi}_1 q_1}} \dots \text{ad}(\xi_j) e^{\text{ad}(\xi_j) q_j} \dots \text{Ad}_{e^{\hat{\xi}_{k-1} q_{k-1}}} \xi_k \end{aligned} \quad (109)$$

where the last term can be easily rewritten as

$$\begin{aligned} &\text{Ad}_{e^{\hat{\xi}_1 q_1}} \dots \text{Ad}_{e^{\hat{\xi}_{j-1} q_{j-1}}} \text{ad}(\xi_j) \text{Ad}_{e^{-\hat{\xi}_{j-1} q_{j-1}}} \dots \\ &\dots \text{Ad}_{e^{-\hat{\xi}_1 q_1}} \text{Ad}_{e^{\hat{\xi}_1 q_1}} \dots \text{Ad}_{e^{\hat{\xi}_{k-1} q_{k-1}}} \xi_k \\ &= \text{ad} \left(\text{Ad}_{(e^{\hat{\xi}_1 q_1} \dots e^{\hat{\xi}_{j-1} q_{j-1}})} \xi_j \right) \xi_k' = \text{ad}(\xi_j') \xi_k' \end{aligned} \quad (110)$$

The results can be finally summarized as

$$\frac{\partial \xi_k'}{\partial q_j} = \begin{cases} \text{ad}(\xi_j') \xi_k' & \text{if } j < k \\ 0 & \text{otherwise.} \end{cases} \quad (111)$$

Thanks to (111), using the definition (96), it is now trivial to compute (107), thus also (105).

Appendix C2. Derivative of the Jacobian matrix with respect to the object configuration

Let us consider now the problem of calculating the derivative of the Jacobian matrix of the i th contact frame with respect to the object configuration u , that is the term

$$U_i = \frac{\partial {}^{c_o}J_i^T(q,u)f_{c_i^h}^{c_o}}{\partial u} \quad (112)$$

As a first step, considering that (112) can be equivalently written as

$$U_i = \begin{bmatrix} \frac{\partial {}^{c_o}J_i^T(q,u)f_{c_i^h}^{c_o}}{\partial u_1} & \frac{\partial {}^{c_o}J_i^T(q,u)f_{c_i^h}^{c_o}}{\partial u_2} & \dots & \frac{\partial {}^{c_o}J_i^T(q,u)f_{c_i^h}^{c_o}}{\partial u_6} \end{bmatrix} \quad (113)$$

by virtue of (17), we reduce the problem to the calculation of

$$\begin{aligned} &\frac{\partial \left(B_i^T \text{Ad}_{g_{c_i^b}} \text{Ad}_{g_{ba}(u)} \mathcal{J}_i(q) \right)^T f_{c_i^h}^{c_o}}{\partial u_j} \\ &= \mathcal{J}_i(q)^T \left(\frac{\partial \text{Ad}_{g_{ba}(u)}}{\partial u_j} \right)^T \text{Ad}_{g_{c_i^b}^T(u)}^T B_i f_{c_i^h}^{c_o} \end{aligned} \quad (114)$$

that is finally

$$\frac{\partial \text{Ad}_{g_{ba}(u)}}{\partial u_j} = \frac{\partial \text{Ad}_{g_{ab}^{-1}(u)}}{\partial u_j} \quad (115)$$

As seen in (13), a parametrization for the object twist is needed. We can obtain it introducing the Jacobian of a virtual kinematic chain described by a suitable Jacobian matrix of the form

$$\begin{aligned} J_o(u) &= \begin{bmatrix} \xi_{o_1} & \xi'_{o_2} & \dots & \xi'_{o_6} \end{bmatrix} \\ \xi'_{o_k} &= \text{Ad}_{(e^{\hat{\xi}_{o_1} u_1} \dots e^{\hat{\xi}_{o_{k-1}} u_{k-1}})} \xi_{o_k} \end{aligned} \quad (116)$$

The same parametrization can be used to describe the homogeneous matrix $g_{ab}(u)$ as product of exponentials

$$g_{ab}(u) = e^{\hat{\xi}_1 u_1} e^{\hat{\xi}_2 u_2} \dots e^{\hat{\xi}_6 u_6} \quad (117)$$

Substituting (117) into (115), considering the properties of the adjoint matrix, it is easy to obtain

$$\begin{aligned} \frac{\partial \text{Ad}_{g_{ab}^{-1}(u)}}{\partial u_j} &= \frac{\partial}{\partial u_j} \text{Ad}_{(e^{-\hat{\xi}_{o_6} u_6} e^{-\hat{\xi}_{o_5} u_5} \dots e^{-\hat{\xi}_{o_1} u_1})} \\ &= \text{Ad}_{e^{-\hat{\xi}_{o_6} u_6}} \dots \text{Ad}_{e^{-\hat{\xi}_{o_{j+1}} u_{j+1}}} \frac{\partial e^{\text{ad}(-\xi_{o_j} u_j)}}{\partial u_j} \\ &\quad \text{Ad}_{e^{-\hat{\xi}_{o_{j-1}} u_{j-1}}} \dots \text{Ad}_{e^{-\hat{\xi}_{o_1} u_1}} \end{aligned} \quad (118)$$

The derivative term in (118) can be also expressed as

$$\begin{aligned} \frac{\partial e^{\text{ad}(-\xi_{o_j} u_j)}}{\partial u_j} &= \text{ad}(-\xi_{o_j}) e^{\text{ad}(-\xi_{o_j} u_j)} = \text{ad}(-\xi_{o_j}) \text{Ad}_{e^{-\hat{\xi}_{o_j} u_j}} \\ &= \text{Ad}_{e^{-\hat{\xi}_{o_j} u_j}} \dots \text{Ad}_{e^{-\hat{\xi}_{o_1} u_1}} \text{Ad}_{e^{\hat{\xi}_{o_1} u_1}} \dots \text{Ad}_{e^{\hat{\xi}_{o_j} u_j}} \\ &\quad \text{ad}(-\xi_{o_j}) \text{Ad}_{e^{-\hat{\xi}_{o_j} u_j}} \end{aligned} \quad (119)$$

Substituting (119) in (118), considering the properties seen in (97) and in (98), we finally obtain

$$\begin{aligned} \frac{\partial \text{Ad}_{g_{ab}^{-1}(u)}}{\partial u_j} &= \text{Ad}_{(e^{-\hat{\xi}_{o_6} u_6} \dots e^{-\hat{\xi}_{o_1} u_1})} \text{ad}(-\xi'_{o_j}) \\ &= \text{Ad}_{g_{ab}^{-1}(u)} \text{ad}(-\xi'_{o_j}) \end{aligned} \quad (120)$$

From (120), it is now trivial to compute (114), and thus (112).

Appendix D. Block elements of the canonical form of the FGM

Appendix D1. FGM for a fully actuated hand grasping an object

The FGE relative to the case of a grasp performed by a fully actuated hand can be obtained considering (8), (14), (19), (23) and (25). This appears in the form

$$\begin{bmatrix} I_{ff}^f & 0 & 0 & -K_c & K_c & 0 & 0 \\ -{}^c\mathcal{J}^T & I_{\tau\tau} & -\bar{U} & 0 & 0 & 0 & -\bar{Q} \\ 0 & 0 & -{}^bGK_c {}^b\bar{G}_t^T & 0 & 0 & I_{\tau w} {}^bGK_c {}^c\mathcal{J} \\ 0 & 0 & 0 & I_{\tau Ch} & 0 & 0 & -{}^c\mathcal{J} \\ 0 & 0 & -{}^b\bar{G}_t^T & 0 & I_{\tau Co} & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta f_{ch}^{co} \\ \delta \tau \\ \delta u \\ \delta C_{ac}^{co} \\ \delta C_{ac}^{co} \\ \delta w_e^b \\ \delta q \end{bmatrix} = 0 \quad (121)$$

Applying the algorithm GEROME-B to (121), this becomes in quasi-diagonal form as follows

$$\begin{bmatrix} I_{ff}^f & 0 & 0 & 0 & 0 & W_f^f & R_f^f \\ 0 & I_{\tau\tau} & 0 & 0 & 0 & W_\tau^f & R_\tau^f \\ 0 & 0 & I_{\tau w} & 0 & 0 & W_u^f & R_u^f \\ 0 & 0 & 0 & I_{\tau Ch} & 0 & W_{Ch}^f & R_{Ch}^f \\ 0 & 0 & 0 & 0 & I_{\tau Co} & W_{Co}^f & R_{Co}^f \end{bmatrix} \begin{bmatrix} \delta f_{ch}^{co} \\ \delta \tau \\ \delta u \\ \delta C_{ac}^{co} \\ \delta C_{ac}^{co} \\ \delta w_e^b \\ \delta q \end{bmatrix} = 0 \quad (122)$$

The superscript f was introduced to distinguish the blocks relative to the fully actuated case, from the cases with elastic joints and synergistic underactuation, respectively, presented later in this section. The blocks composing (122) have the following expression

$$\begin{aligned} W_f^f &= K_c {}^b\bar{G}_t^T ({}^bGK_c {}^b\bar{G}_t^T)^{-1} \\ W_\tau^f &= ({}^c\mathcal{J}^T K_c {}^b\bar{G}_t^T - \bar{U}) ({}^bGK_c {}^b\bar{G}_t^T)^{-1} \\ W_u^f &= -({}^bGK_c {}^b\bar{G}_t^T)^{-1} \\ W_{Ch}^f &= 0 \\ W_{Co}^f &= -{}^b\bar{G}_t^T ({}^bGK_c {}^b\bar{G}_t^T)^{-1} \end{aligned} \quad (123)$$

$$\begin{aligned} R_f^f &= K_c ({}^b\bar{G}_t^T ({}^bGK_c {}^b\bar{G}_t^T)^{-1} {}^bGK_c - I_f) {}^c\mathcal{J} \\ R_\tau^f &= -\bar{Q} - {}^c\mathcal{J}^T K_c {}^c\mathcal{J} + ({}^c\mathcal{J}^T K_c {}^b\bar{G}_t^T - \bar{U}) ({}^bGK_c {}^b\bar{G}_t^T)^{-1} {}^bGK_c {}^c\mathcal{J} \\ R_u^f &= -({}^bGK_c {}^b\bar{G}_t^T)^{-1} {}^bGK_c {}^c\mathcal{J} \\ R_{Ch}^f &= -{}^c\mathcal{J} \\ R_{Co}^f &= -{}^b\bar{G}_t^T ({}^bGK_c {}^b\bar{G}_t^T)^{-1} {}^bGK_c {}^c\mathcal{J} \end{aligned} \quad (124)$$

Appendix D2. FGM for a fully actuated hand with elastic joints grasping an object

The FGE describing the case of a hand with elastic joints can be obtained adding (27) to the system (122), resulting in

$$\begin{bmatrix} I_{ff}^f & 0 & 0 & -K_c & K_c & 0 & 0 & 0 \\ -{}^c\mathcal{J}^T & I_{\tau\tau} & -\bar{U} & 0 & 0 & -\bar{Q} & 0 & 0 \\ 0 & 0 & -{}^bGK_c {}^b\bar{G}_t^T & 0 & 0 & {}^bGK_c {}^c\mathcal{J} & I_{\tau w} & 0 \\ 0 & 0 & 0 & I_{\tau Ch} & 0 & -{}^c\mathcal{J} & 0 & 0 \\ 0 & 0 & -{}^b\bar{G}_t^T & 0 & I_{\tau Co} & 0 & 0 & 0 \\ 0 & I_{\tau\tau} & 0 & 0 & 0 & K_q & 0 & -K_q \end{bmatrix} \begin{bmatrix} \delta f_{ch}^{co} \\ \delta \tau \\ \delta u \\ \delta C_{ac}^{co} \\ \delta C_{ac}^{co} \\ \delta q \\ \delta w_e^b \\ \delta q_r \end{bmatrix} = 0 \quad (125)$$

The canonical form is achieved applying GEROME-B to the coefficient matrix of (125). The final form of the system appears as

$$\begin{bmatrix} I_{ff}^f & 0 & 0 & 0 & 0 & 0 & W_f^e & R_f^e \\ 0 & I_{\tau\tau} & 0 & 0 & 0 & 0 & W_\tau^e & R_\tau^e \\ 0 & 0 & I_{\tau w} & 0 & 0 & 0 & W_u^e & R_u^e \\ 0 & 0 & 0 & I_{\tau Ch} & 0 & 0 & W_{Ch}^e & R_{Ch}^e \\ 0 & 0 & 0 & 0 & I_{\tau Co} & 0 & W_{Co}^e & R_{Co}^e \\ 0 & 0 & 0 & 0 & 0 & I_{\tau q} & W_q^e & R_q^e \end{bmatrix} \begin{bmatrix} \delta f_{ch}^{co} \\ \delta \tau \\ \delta u \\ \delta C_{ac}^{co} \\ \delta C_{ac}^{co} \\ \delta q \\ \delta w_e^b \\ \delta q_r \end{bmatrix} = 0 \quad (126)$$

The blocks composing (122), expressed as function of the terms in (123) and (124), have the form

$$\begin{aligned} W_f^e &= W_f^f + R_f^f (K_q - R_\tau^f)^{-1} W_\tau^f \\ W_\tau^e &= W_\tau^f + R_\tau^f (K_q - R_\tau^f)^{-1} W_\tau^f \\ W_u^e &= W_u^f + R_u^f (K_q - R_\tau^f)^{-1} W_\tau^f \\ W_{Ch}^e &= R_{Ch}^f (K_q - R_\tau^f)^{-1} W_\tau^f \\ W_{Co}^e &= W_{Co}^f + R_{Co}^f (K_q - R_\tau^f)^{-1} W_\tau^f \\ W_q^e &= - (K_q - R_\tau^f)^{-1} W_\tau^f \end{aligned} \quad (127)$$

$$\begin{aligned} R_f^e &= R_f^f (K_q - R_\tau^f)^{-1} K_q \\ R_\tau^e &= R_\tau^f (K_q - R_\tau^f)^{-1} K_q \\ R_u^e &= R_u^f (K_q - R_\tau^f)^{-1} K_q \\ R_{Ch}^e &= R_{Ch}^f (K_q - R_\tau^f)^{-1} K_q \\ R_{Co}^e &= R_{Co}^f (K_q - R_\tau^f)^{-1} K_q \\ R_q^e &= - (K_q - R_\tau^f)^{-1} K_q \end{aligned} \quad (128)$$

Appendix D3. FGM for a synergistic underactuated hand grasping an object

Using the result in (122), the FGM in (43) can be rewritten as

$$\begin{bmatrix} I_{ff}^* & 0 & 0 & 0 & 0 & 0 & R_f^f & 0 & 0 & W_f^f & 0 \\ 0 & I_{\tau\tau}^* & 0 & 0 & 0 & 0 & R_\tau^f & 0 & 0 & W_\tau^f & 0 \\ 0 & -\tilde{S}^T & I_{\eta\eta}^* & 0 & 0 & 0 & 0 & 0 & -\tilde{\Sigma} & 0 & 0 \\ 0 & 0 & 0 & I_{uu}^* & 0 & 0 & R_u^f & 0 & 0 & W_u^f & 0 \\ 0 & 0 & 0 & 0 & I_{C^h}^* & 0 & R_{C^h}^f & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{C^o}^* & R_{C^o}^f & 0 & 0 & W_{C^o}^f & 0 \\ 0 & I_{\tau\tau}^* & 0 & 0 & 0 & 0 & K_q & -K_q & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{qr}^* & -\tilde{S} & 0 & 0 \\ 0 & 0 & I_{\sigma\sigma}^* & 0 & 0 & 0 & 0 & 0 & K_\sigma & 0 & -K_\sigma \end{bmatrix} \begin{bmatrix} \delta f_{C^h}^{c^o} \\ \delta \tau \\ \delta \eta \\ \delta u \\ \delta C_{C^h}^{c^o} \\ \delta C_{C^o}^{c^o} \\ \delta q \\ \delta q_r \\ \delta \sigma \\ \delta w_e^b \\ \delta \sigma_r \end{bmatrix} = 0 \quad (129)$$

Applying GEROME-B to (129), we arrive to the formulation seen in (44). For later use, we define the matrices

$$\begin{aligned} H &= (K_q - R_\tau^f)^{-1} \\ L &= (K_\sigma + \tilde{\Sigma} - \tilde{S}^T R_\tau^f H K_q \tilde{S})^{-1} \\ T &= \tilde{S}^T (I + R_\tau^f H) W_\tau^f \end{aligned} \quad (130)$$

With these simplifications, the matrices composing (44) can be expressed as follows

$$\begin{aligned} W_f^s &= W_f^f + R_f^f H (W_\tau^f + K_q \tilde{S} L T) \\ W_\tau^s &= W_\tau^f + R_\tau^f H (W_\tau^f + K_q \tilde{S} L T) \\ W_\eta^s &= \tilde{S}^T W_\tau^f + \tilde{\Sigma} L T + \tilde{S}^T R_\tau^f H (W_\tau^f + K_q \tilde{S} L T) \\ W_u^s &= W_u^f + R_u^f H (W_\tau^f - K_q \tilde{S} L T) \\ W_{C^h}^s &= R_{C^h}^f H (W_\tau^f + K_q \tilde{S} L T) \\ W_{C^o}^s &= W_{C^o}^f + R_{C^o}^f H (W_\tau^f + K_q \tilde{S} L T) \\ W_q^s &= -H (W_\tau^f + K_q \tilde{S} L T) \\ W_{q_r}^s &= -\tilde{S} L T \\ W_\sigma^s &= -L T \end{aligned} \quad (131)$$

$$\begin{aligned} R_f^s &= R_f^f H K_q \tilde{S} L K_\sigma \\ R_\tau^s &= R_\tau^f H K_q \tilde{S} L K_\sigma \\ R_\eta^s &= (\tilde{S}^T R_\tau^f H K_q \tilde{S} - \tilde{\Sigma}) L K_\sigma \\ R_u^s &= R_u^f H K_q \tilde{S} L K_\sigma \\ R_{C^h}^s &= R_{C^h}^f H K_q \tilde{S} L K_\sigma \\ R_{C^o}^s &= R_{C^o}^f H K_q \tilde{S} L K_\sigma \\ R_q^s &= -H K_q \tilde{S} L K_\sigma \\ R_{q_r}^s &= -\tilde{S} L K_\sigma \\ R_\sigma^s &= -L K_\sigma \end{aligned} \quad (132)$$

Grasp Movement Classification via Multiple Eigenspaces

Hamal Marino, Marco Gabiccini, Ales Leonardis

PaCMan Technical Report - February 2014

1 Introduction

Dimensionality reduction techniques have extensively been used in the field of robotic grasping, under many different forms, to obtain simple instructions to pilot robotic hands to mimic what human hands do. Human grasp analysis is becoming increasingly automated, and the trend is likely to continue during the next years.

Nonetheless, there is still a gap which needs to be bridged when analyzing this sort of data, namely the classification of the type of grasp performed which is still a task mostly performed via direct inspection.

There is a large number of models and techniques which are useful for dimensionality reduction: Principal Component Analysis (PCA, [1]), Independent Component Analysis (ICA, [2]), functional PCA (fPCA, [3]), Hidden Markov Model (HMM, [4]), Gaussian Process Latent Variable Model (GPLVM, [5]), and many more.

Lately, many of these techniques have been applied to postural and grasping data: for example, Peternel and Leonardis [6] showed that even movements as complex as human locomotion can be modeled by a small number of Degrees of Freedom (DoFs), i.e. 15 GMM in 4-D space. Bernardin et al. [7] approached the problem of fusing data glove and tactile sensor information applying an HMM recognizer to distinguish among different grasp types.

Still, the most widely used technique remains PCA. For example, in [8] the subjects have been asked to grasp a large number of imagined objects in order to reconstruct Principal Components, called *postural synergies*, and in [9], [10] the use of PCA is extended to *temporal* and *kinematic* postural synergies.

These techniques have been applied to show that there is an underlying structure in the way humans grasp.

A different procedure, namely direct inspection, has been used for a seemingly different problem: generation of a grasp taxonomy. Efforts in this direction have been made since the 50s [11] and the 80s [12]. Still recently (see [13] and [14]), to classify grasping postures and movements, the same method used almost sixty years ago is applied, namely direct visual inspection.



Figure 1: An example of a 1-dimensional linear space in hand posture space, i.e. one *eigenspace* of dimension one. Subject ED, spherical grasp movement.

Some attempts at automating the procedure of data segmentation have been made in the last decade (e.g., [15], [16]), still they start by performing dimensionality reduction on the full dataset, and then look for a valid segmentation in the reduced dimensional space which is, indeed, affected by the population employed.

This work aims to find a systematic, data-driven way to explain grasp taxonomy, which could then be used for automatic movement classification and hand-posture database, e.g. [17], indexing.

The results of the global dimensionality reduction techniques cited above can be highly dependent on the dataset used. To avoid this dependency, we have used the Multiple Eigenspace technique [18]. In Figure 1 we show an example of an eigenspace in hand posture space and in Figure 2 it is shown a comparison of Multiple Eigenspaces and PCA on a particular 2-D dataset.

Multiple Eigenspaces are a PCA generalization, which can be seen to be a single eigenspace which explains the whole dataset, while using more than one eigenspace the dataset is automatically clustered and, differently from other clustering techniques like K-means [19], each cluster can have a different dimension. Moreover, the used approach gives lower dimensional subspaces than the ones given by PCA. These subspaces can be used for interpolation, thus to avoid to neglect the low dimensional local structure of the data, which is instead usually ignored.

Especially in robotics, where Programming by Demonstration is increasingly taking place [20, 21], a low number of DoFs is desirable. Using a technique which automatically groups together what can be achieved in a similar way should relieve the programmer of the choice of a suitable dataset, which could then just be big enough to contain what is wanted and the reduction would be done by the learning algorithm itself.

With this in mind, we focus on movements rather than static postures, which can anyway be obtained as ending of a movement. Furthermore, in order to maximize the probability of grasp success, we have to consider the overall time course evolution of the hand shape, and not just the final pose to attain.

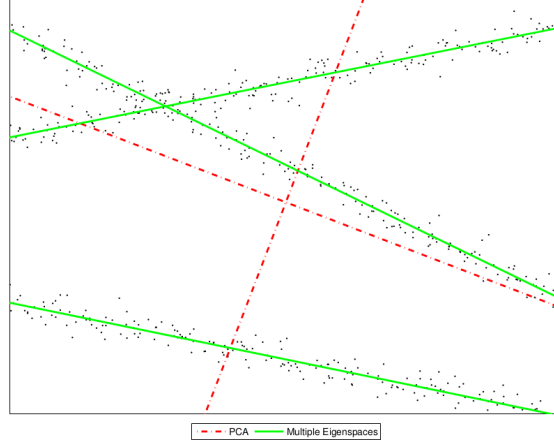


Figure 2: When data are along multiple low dimensional subspaces, a global dimensionality reduction technique may give poor results in terms of ability to explain the data, even if the error in reconstructing each datapoint is very low.

2 Data Gathering

To test the proposed method, we exploit the procedure adopted in [22] where the grasping movement data are obtained from timed sequences of postural data. In particular, the procedure can be described as follow:

- a volunteer (*subject*) has his/her hand prepared with active markers placed on the skin;
- a motion capture system, (Phase Space Impulse system) is used to record the movements at 480 Hz;
- upon timed intervals, a random image from a set of possible objects is shown to the subject for a few seconds (see Table 1 for a partial list of the objects used, or [8] for a full list of the 57 objects used);
- after each image disappears, the subject is asked to perform the grasp as if the object just shown was in front of him/her;
- after the experiment, a calibration phase estimates the parameters of the specific subject hand;
- keeping constant the calibration data, an identification with an Extended Kalman Filter based on a 26 DoFs kinematic hand model is performed on the whole movement, down-sampled to 15 Hz.

From the 26 DoFs data, which includes also 2 wrist DoFs, only the remaining 24 “inner-hand” DoFs are considered. The full timeseries obtained are then directly inspected to extract the beginning and the end of each movement; each subsequence, usually lasting from 15 to 25 frames, is then resampled via piecewise linear approximation to have a constant length of 20 frames.

1. Bucket	11. Hammer
2. Calculator	12. Ice cube
3. Chalk	13. Jar lid
4. Cherry	14. Light bulb
5. Computer mouse	15. Pen
6. Dinner plate	16. Rope
7. Espresso cup	17. Telephone handset
8. Fishing rod	18. Tennis racket
9. Frisbee	19. Toothpick
10. Hair dryer	20. Wrench
21 ÷ 57 ...	

Table 1: A partial list of objects used for data gathering.

Notice that, in the following, the words *posture* and *datapoint* will be used interchangeably to refer to one of the 20 frames generated this way, i.e. one vector of 24 numbers representing the joint angles of a 24 DoFs kinematic hand model.

Data from two subjects (ED and VB, both right handed and unimpaired, between 20 and 30 years old), each performing the full experiment twice, have been used, i.e. a total of $2 \times 2 \times 57 \times 20 = 4560$ postures have been analyzed. The ordering of the dataset is, from inner to outer: frames, trials, objects, subjects; this means that each trial is composed of 20 adjacent postures which are temporally close frames, and each object is represented by two successive trials.

3 Multiple Eigenspaces

The word *Eigenspace* stands for a representation of a subset of the data which consists of a mean datapoint and a certain number of linear directions, taken as the directions of maximum variance in the data: this number is called *dimension* of the eigenspace.

The problem of generating the eigenspaces has two main issues:

- i) which postures belong to the same eigenspace
- ii) what the dimension of each eigenspace should be.

To take this aspect into account, the multiple eigenspaces are generated by the following cyclical procedure:

- 1) generate a high number of seeds with adjacent postures in the dataset (**Seed Generation**),
- 2) repeat until an Eigenspace cannot further grow:

- grow them independently of each other (**Eigenspace Independent Growing**)
- reduce the number of datapoints in each eigenspace based on reconstruction error (**Datapoint Reduction**)
- prune the eigenspaces using a selection procedure (**Eigenspace Selection**)

3) keep each datapoint in no more than one eigenspace (**Final Selection and Unification**).

As regards the procedure shown in [18], there are three main differences with the one we are proposing: i) datapoints included in one eigenspace can be taken out of it during the datapoint reduction procedure (to avoid too many overlaps); ii) zero-order eigenspaces are forcefully removed (to focus on movements); and iii) each posture belong to one eigenspace only (to maximize the ability of that eigenspace to explain that posture).

In the following we describe the details of each step of the procedure.

Seeds generation Seeds, which are the initial stage of the eigenspaces, are generated from the dataset (ordered as explained in Section 2) with a proximity criterion: once the initial scope of the seeds has been chosen (i.e., the number of postures to initially include), corresponding postures are incorporated in an eigenspace which, at the beginning, simply represents their mean value (i.e., has dimension *zero*).

The scope has to be chosen small enough to let the seeds be free to evolve in the best possible direction as dictated by the data. We denote as E_j^t the j -th eigenspace at stage t , thus E_j^0 represents the j -th seed.

Eigenspace independent growing Each eigenspace is independently grown inserting the postures which are more closely related to it, considering their reconstruction error δ . The δ_{ij}^t error of posture i w.r.t. the eigenspace j at stage t is simply the norm of the distance between the posture $\mathbf{x}_i$ and its reconstruction $\hat{\mathbf{x}}_{ij}^t$ obtained in E_j^t

$$\delta_{ij}^t = \|\mathbf{x}_i - \hat{\mathbf{x}}_{ij}^t\| \quad (1)$$

where the reconstruction $\hat{\mathbf{x}}_{ij}^t$ is the projection of $\mathbf{x}_i$ onto the eigenspace.

An allowable error level σ (see Table 2), has to be chosen depending on the data (i.e., what we consider to be an average level of error in a group of postures). The δ error of the posture has thus to be below a pre-specified threshold δ_{th} to avoid inserting in the eigenspace postures which are too far from it, still trying to expand the scope; that is why we use $\delta_{th} = 2.0\sigma$.

To also avoid having too many overlaps in the data, the maximum number of postures we allow each time to enter an eigenspace is initially equal to the scope of the seeds. We then double this every time, to allow a further growth of the eigenspaces.

Another important measure to consider is the overall reconstruction error of E_j^t , called ρ error

$$(\rho_j^t)^2 = \frac{1}{\#E_j^t} \sum_{\mathbf{x}_i \in E_j^t} (\delta_{ij}^t)^2, \quad (2)$$

where the symbol $\#$ means the number of postures in E_j^t , i.e. its scope.

This value is used to decide whether an eigenspace, after the postures found as previously described are inserted (i.e., the eigenspace E_j^{t+1}), is expanding correctly, and when it would be useful to increase its dimension; this is achieved using two thresholds $\rho_{th1} \leq \rho_{th2}$ (chosen as in Table 2) and the following procedure:

if the eigenspace E_j^{t+1} is already a good representation for the postures in it, i.e. $\rho_j^{t+1} < \rho_{th2}$, accept the last set of postures inserted and continue

else try increasing the dimension of E_j^{t+1} by one (generating $\hat{E}_j^{t+1}$) and compute again its error $\hat{\rho}_j^{t+1}$

if the error of E_j^{t+1} is significantly reduced, i.e. $\hat{\rho}_j^{t+1} < \rho_{th1}$, accept E_j^{t+1} with all postures present in it

else discard the last set of inserted posture, revert the eigenspace to its previous stage (assigning $\hat{E}_j^{t+1} = \hat{E}_j^t$) and stop growing it.

Datapoint reduction After all eigenspaces have passed a stage of growing, a datapoint reduction procedure is performed to keep each posture in at most two eigenspaces. This is done by keeping each posture in the two eigenspaces which best reconstruct it, that is selecting the two lowest δ errors amongst all possible δ errors across the various eigenspaces which include that posture. We reduce the dimension of the eigenspace until the condition on the error threshold is respected.

This step, which was not done in the algorithm of [18], serves as a preparation to the selection procedure, which can handle well *overlaps* (datapoints present in more than one eigenspace) up to the 2nd order (i.e., pairwise overlaps), but suffers when much higher order overlaps exists.

Eigenspace selection This procedure is used to take some eigenspaces out. The choice of which eigenspaces have to be preserved, is based on a Minimum Description Length (MDL) principle that considers the following cost function:

$$F(\mathbf{h}) = \mathbf{h}^T C \mathbf{h} = \mathbf{h}^T \begin{bmatrix} c_{11} & \cdots & c_{1r} \\ \vdots & & \vdots \\ c_{r1} & \cdots & c_{rr} \end{bmatrix} \mathbf{h} \quad (3)$$

where c_{jj} is the *saving* associated with the preserving of the eigenspace j , and c_{jk} is the *saving* associated with the preserving of both eigenspaces j and k . For the sake of readability, all superscripts t are omitted in the following equations. c_{jj} is defined as:

$$c_{jj} = K_0(\#E_j) - (K_{1,d}d_j + K_{1,m} + K_2(\#E_j)d_j + K_3(\#E_j)\rho_j), \quad (4)$$

where d_j is the dimension of E_j , while K_0 , $K_{1,d}$, $K_{1,m}$, K_2 , and K_3 are constant (see Table 2). K_0 is related to the cost of encoding one posture in the absence of the eigenspace, $K_{1,m}$ to the cost of encoding the mean posture of the eigenspace, and $K_{1,d}$ to the cost of encoding

Parameters	Description	Value
σ	level of allowed error	$[0.3 \div 0.5]$
δ_{th}	threshold: allow a datapoint in	2.0σ
$\rho_{\text{th}2}$	threshold: request a dimension upgrade	σ
$\rho_{\text{th}1}$	threshold: accept a dimension upgrade	1.2σ
K_2/K_0	cost of a coefficient	$1/24$
K_3/K_0	cost of the reconstruction error	1.1

Table 2: Parameters used in the eigenspace generation procedure.

each eigenposture: these costs are considered to be equal, as all postures have 24 DoFs. K_2 is the cost related to encoding the coefficients of each posture in the eigenspace, and it has been chosen equal to $1/24$ of K_0 (1 coefficient versus 24 DoFs per posture). Finally, K_3 is related to the average cost due to the error, and is a parameter which have to be chosen appropriately. c_{jj} can thus be simplified to:

$$c_{jj} = K_0(\#E_j - d_j - 1) - (\#E_j)d_j - K_3(\#E_j)\rho_j. \quad (5)$$

Elements c_{jk} out of the diagonal are elements which are used to consider that savings only occur only once even when the postures are inserted in more than one eigenspace. Thus, we consider the number of overlaps and subtract the associated savings once (that is why this procedure works well with pairwise overlaps, but fails when higher order overlaps are present)

$$c_{jk} = -K_0(\#E_{j \cap k}) + (\#E_{j \cap k})d_{jk} + K_3(\#E_{j \cap k})\rho_{jk}, \quad (6)$$

where $E_{j \cap k}$ represent the datapoint intersection of E_j and E_k , and d_{jk} and ρ_{jk} can be chosen to be (d_j, ρ_j) or (d_k, ρ_k) on the basis of the higher error, computed only on the overlapping postures.

Final selection and unification

When the growing procedure terminates, because or no more points can be added to the eigenspaces or the eigenspaces reached a steady state (all datapoints added in the growth phase are then taken out by the datapoint reduction phase), we unify the eigenspaces, allowing each posture to belong to no more than one eigenspace. This step reduces the overlaps to zero.

Moreover, since we are interested in movements rather than just static posture classification, before performing this merging, we remove all the zero-order eigenspaces, that is the postures where they are represented only by a mean value).

An example of how the eigenspace growing procedure evolves is shown in Figure 3 where different stages, from seed formation to final selection, are shown. Each figure therein have on the x-axis the posture indexes, and on the y-axis the seed/eigenspace indexes. Comparing these figures with similar ones in [18], where eigenspaces were growing taking adjacent datapoints in at each stage, would reinforce how the characteristics of the dataset are different and why the procedure had to be adapted to obtain meaningful results.

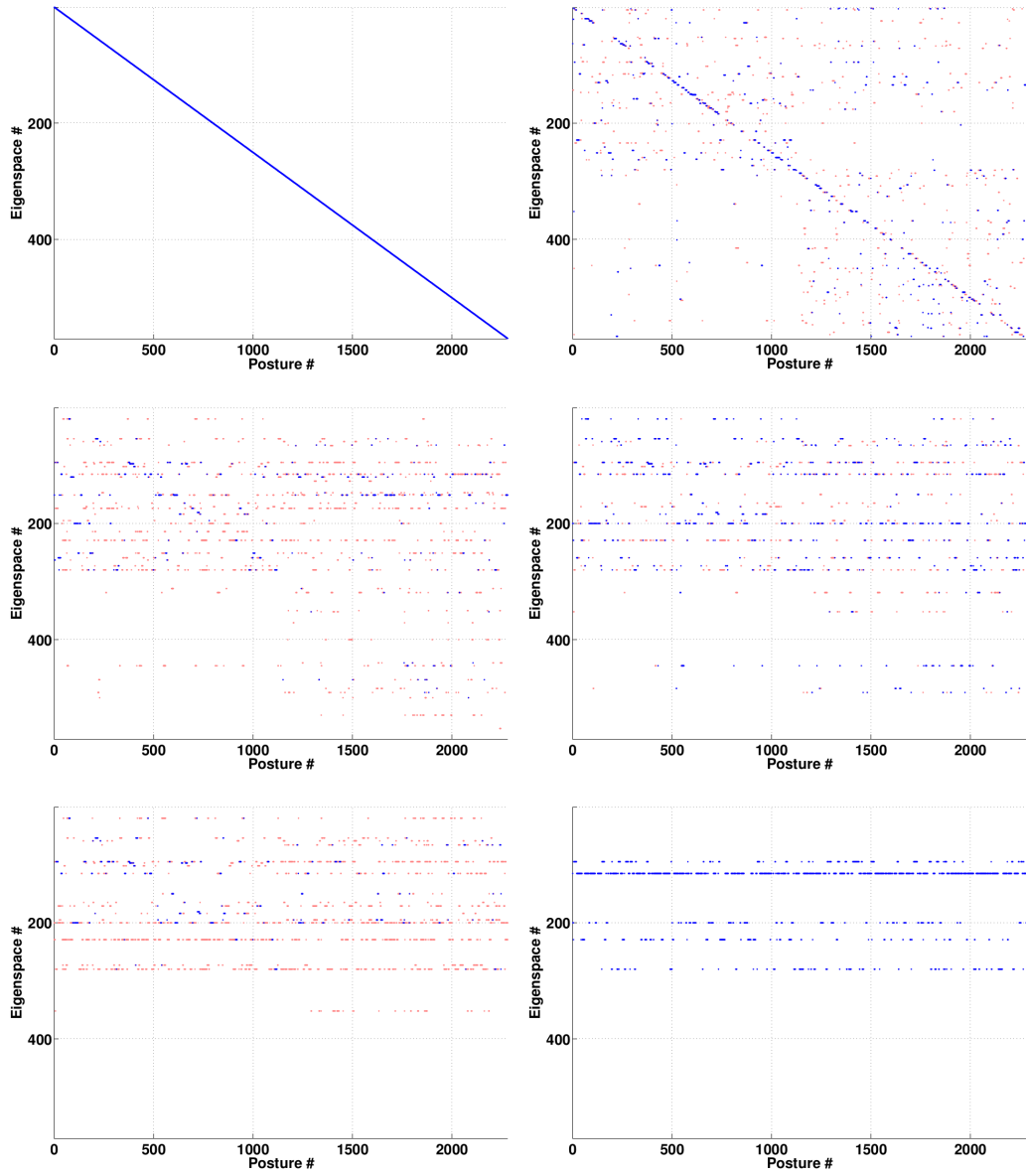


Figure 3: Growing and selection of eigenspaces. Left to right, top to bottom: seeds, stage 5, 10, 15, 20, final selection. Example of subject ED. A total of 2280 datapoints are clustered into 5 eigenspaces, 4 of which are 1-D and 1 is 2-D, starting from 570 initial seeds of 4 postures each. Each figure has posture indexes on the x-axis, and eigenspace indexes on the y-axis. During intermediate stages, light (red) dots represent overlapping postures (datapoints present in more than one eigenspace): notice that these are absent in the first and last stages.



Figure 4: Subject ED: medium wrap grasping; hand closing from left to right.

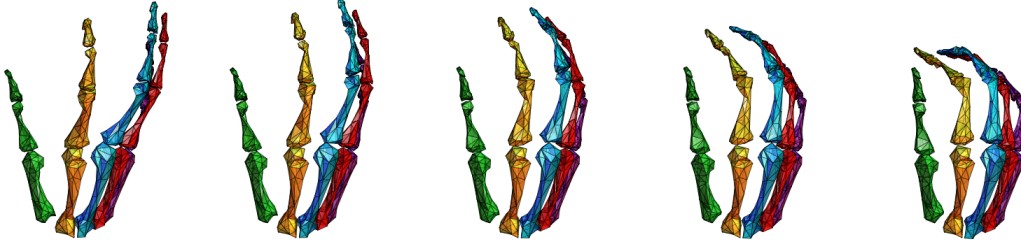


Figure 5: Subject ED: light tool grasping; hand closing from left to right.

4 Results

The technique described in the previous Section has been applied to the dataset obtained as explained in Section 2.

We observe that, independently of the value given to the allowed error threshold σ , a gross clustering always resulted in mainly different eigenspaces to represent each subject. Keeping this aspect in mind, all the following considerations are made on two separate eigenspace sets, i.e. one per subject. Seeds are, thus, 570 per subject, comprising 4 postures each.

Evolution of the eigenspaces In Figure 3, seed formation, four intermediate stages, and final selection, are shown for subject ED with $\sigma = 0.34$. After applying the final selection procedure, the number of eigenspaces considered at each stage decreases from 570 seeds to only 5, 4 of which are 1-D and one is 2-D. Similar results have been obtained for VB, where the final selection procedure extract 8 1-D eigenspaces for $\sigma = 0.43$. The values chosen for σ are different, depending on the dataset to be considered, and a general trend is that higher values of σ allow grouping into bigger clusters, and vice-versa.

Grasp taxonomy classification of eigenspaces Eigenspaces resulting from this procedure have been directly inspected to assess their similarity to entries of a classical grasp taxonomy. Most similar grasps from the taxonomy in [12] are:

ED: medium wrap (see Figure 4), tripod, light tool (see Figure 5), thumb - 4 finger, sphere (see Figure 1);

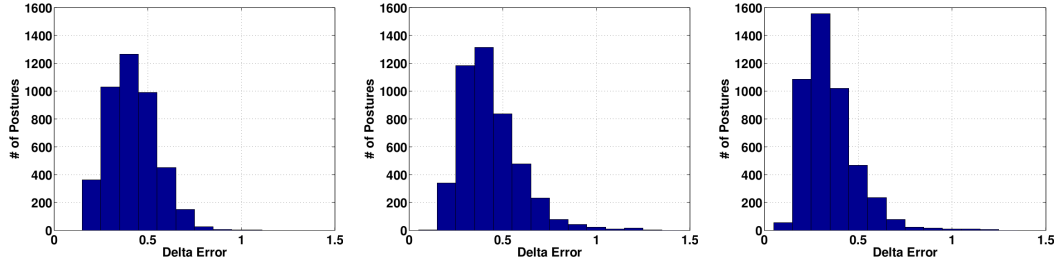


Figure 6: Histograms of delta error across datapoints. From left to right: delta error obtained with a Multiple Eigenspace representation; delta error with 5 PCs from PCA; again PCA, taking into account 7 PCs. Notice that the total of postures in the first (left-most) figure is lower than the other two, as only postures represented by the selected eigenspaces are considered.

VB: tripod, lateral pinch, medium wrap, thumb - 2 finger, large diameter, thumb-index, prismatic with adducted thumb, light tool (see the attached video for a graphic representation of these movements).

4.1 Comparison with PCA

As the proposed multiple eigenspaces algorithm is a generalization of PCA, the results are compared with the ones obtained by using the PCA on the overall dataset. For the PCA, we choose to employ a number of 5 or 7 Principal Components in order to show results of the same orders.

In Table 3, a comparison of explained variance across the postures included in each eigenspace is reported. The first column indicates the subject to whom the data belong, then the second column says whether we consider a PCA or an eigenspace, and the third column shows the number of PCs (or eigenspace dimension). Fourth and fifth columns are, respectively, number of postures and explained variance, in percentage over total variance, of the corresponding dataset. Finally, last column, inserted for completeness, displays how many PCs coming from a PCA on the full dataset would be needed to explain the same amount of percentage variance on all the postures. This last measure can be misleading, as averaging effects can take place when considering a dataset which is much broader than the ones of each eigenspace.

A fairer comparison can be made using the reconstruction error (δ error) of each posture. In Figure 6 the histograms of δ error using, from left to right, multiple eigenspaces, 5 PCs, and 7 PCs respectively, are shown. It is possible to see that, using multiple eigenspaces, the amount of postures reconstructed with an error up to 0.5 is similar to a PCA in which 5 PCs are considered. Using more PCs obviously leads to better results on the reconstruction. Finally, in the first of these images, there are no postures with a δ error as high as in the other two, and this is because datapoints with such a high error have been automatically discarded by the procedure.

Subject(s)		dim.	# postures	variance	# of corr. PCs
VB + ED	PCA	5	4560	75%	-
VB + ED	PCA	7	4560	84%	-
VB	E_1	1	191	66%	4
VB	E_2	1	293	67%	4
VB	E_3	1	243	57%	3
VB	E_4	1	369	58%	3
VB	E_5	1	166	63%	3
VB	E_6	1	373	63%	3
VB	E_7	1	307	66%	4
VB	E_8	1	129	79%	6
ED	E_1	1	287	70%	5
ED	E_2	2	1302	60%	3
ED	E_3	1	236	61%	3
ED	E_4	1	210	40%	1
ED	E_5	1	182	44%	2

Table 3: Comparison of percentage explained variance values using PCA or Multiple Eigenspaces. PCA is performed on the whole dataset, while eigenspaces are generated on separate subjects as they become clustered from the very beginning. Number of Principal Components (PCs) used, or dimension of the eigenspace, are shown, along with number of datapoints included and percentage of total variance in the data explained. In the last column, the corresponding number of PCs required to explain the same amount of total variance across the whole dataset is reported.

5 Conclusions

In this work, the algorithm of Multiple Eigenspaces from [18] has been adapted and applied to a dataset of grasping movement postures. Data obtained following the procedure from [22] are processed with the proposed algorithm, and achieved results are shown. Noteworthy, grasping movements analogous to classical grasp taxonomy entries (from, e.g., [12]) are automatically found from the data, as shown in Figures 1, 4, and 5.

A comparison with PCA is presented, keeping in mind that these are different approaches and that the inherent low dimensionality of Multiple Eigenspaces, together with their generalization ability, comes into play when interpretation of the data is crucial. Especially notice that this procedure, differently from PCA, discards points which are considered to be outliers, and clusters the remaining to be as meaningful as possible.

The presented algorithm builds in the direction of increasingly automating classification of movements, and could be used for purposes related to Programming by Demonstration dataset generation, and data-driven posture database indexing.

Due to ongoing research, we expect to be able to extend the ability of automatic taxonomy-like

grasp movement generation, dealing with different levels in the taxonomy hierarchy using a systematic way to change the parameters.

Moreover, future work also include the idea of using Multiple Eigenspaces procedure, which is based on PCA, building upon more complex (maybe nonlinear) dimensionality reduction techniques.

References

- [1] I. Jolliffe, “Principal component analysis and factor analysis,” *Principal Component Analysis*, pp. 150–166, 2002.
- [2] A. Hyvärinen, J. Karhunen, and E. Oja, “What is independent component analysis?” *Independent Component Analysis*, pp. 145–164, 2001.
- [3] J. O. Ramsay and B. W. Silverman, *Applied functional data analysis: methods and case studies*. Springer New York, 2002, vol. 77.
- [4] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [5] N. D. Lawrence, “Gaussian process latent variable models for visualisation of high dimensional data,” *Advances in neural information processing systems*, vol. 16, no. 329–336, p. 3, 2004.
- [6] M. Peternel and A. Leonardis, “Visual learning and recognition of a probabilistic spatio-temporal model of cyclic human locomotion,” in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4. IEEE, 2004, pp. 146–149.
- [7] K. Bernardin, K. Ogawara, K. Ikeuchi, and R. Dillmann, “A hidden markov model based sensor fusion approach for recognizing continuous human grasping sequences,” in *Proc. 3rd IEEE International Conference on Humanoid Robots*, 2003, pp. 1–13.
- [8] M. Santello, M. Flanders, and J. F. Soechting, “Postural hand synergies for tool use,” *The Journal of Neuroscience*, vol. 18, no. 23, pp. 10 105–10 115, 1998.
- [9] R. Vinjamuri, M. Sun, C.-C. Chang, H.-N. Lee, R. J. Sclabassi, and Z.-H. Mao, “Temporal postural synergies of the hand in rapid grasping tasks,” *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 4, pp. 986–994, 2010.
- [10] —, “Dimensionality reduction in control and coordination of the human hand,” *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 2, pp. 284–295, 2010.
- [11] J. R. Napier, “The prehensile movements of the human hand,” *Journal of bone and Joint surgery*, vol. 38, no. 4, pp. 902–913, 1956.
- [12] M. Cutkosky, “On grasp choice, grasp models, and the design of hands for manufacturing tasks,” *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 3, pp. 269–279, 1989.

- [13] I. Bullock, R. Ma, and A. Dollar, “A hand-centric classification of human and robot dexterous manipulation,” *Haptics, IEEE Transactions on*, vol. 6, no. 2, pp. 129–144, 2013.
- [14] I. Bullock, J. Zheng, S. De La Rosa, C. Guertler, and A. Dollar, “Grasp frequency and usage in daily household and machine shop tasks,” *Haptics, IEEE Transactions on*, vol. 6, no. 3, pp. 296–308, 2013.
- [15] L. Y. Chang, N. S. Pollard, T. M. Mitchell, and E. P. Xing, “Feature selection for grasp recognition from optical markers,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 2944–2950.
- [16] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard, “Segmenting motion capture data into distinct behaviors,” in *Proceedings of Graphics Interface 2004*. Canadian Human-Computer Communications Society, 2004, pp. 185–194.
- [17] M. Bianchi and M. V. Liarokapis, “HandCorpus, a new open-access repository for sharing experimental data and results on human and artificial hands,” in *IEEE World Haptics Conference (WHC)*, April 2013.
- [18] A. Leonardis, H. Bischof, and J. Maver, “Multiple eigenspaces,” *Pattern Recognition*, vol. 35, no. 11, pp. 2613–2627, 2002.
- [19] J. A. Hartigan, *Clustering algorithms*. John Wiley & Sons, Inc., 1975.
- [20] P. Kingston and M. Egerstedt, “Time and output warping of control systems: Comparing and imitating motions,” *Automatica*, vol. 47, no. 8, pp. 1580–1588, 2011.
- [21] S. Calinon, “Robot programming by demonstration,” in *Springer Handbook of Robotics*. Springer, 2008, pp. 1371–1394.
- [22] M. Gabiccini, G. Stillfried, H. Marino, and M. Bianchi, “A data-driven kinematic model of the human hand with soft-tissue artifact compensation mechanism for grasp synergy analysis,” in *Intelligent Robots and Systems, 2013. Proceedings., 2013 IEEE/RSJ International Conference on*. IEEE, 2013.

Synthetic Grasp Affordances for the Pisa/IIT SoftHand

Marco Gabiccini, Michaela Antonelli

PaCMan Technical Report - February 2014

1 Introduction

For well known scenes and object models, pre-programming of autonomous grasping and manipulation tasks may be an option. To this aim, several approaches have been proposed to define the optimal finger placement on the object, either based on some geometric [1] or force [2, 3] grasp quality measures, specifically tailored to convex objects [4], with optimal on-line contact adjustments [5], and also generalized to non-convex objects [6].

On the other hand, adaptation of a grasping algorithm to formerly unknown or only partially known scenes remains a difficult task, to which several approaches have been investigated. Some of them are model-free and propose geometrical features which indicate good grasps [7, 8], some others evaluate also topological object properties such as holes [9]. Typically, grasps are ranked on a fixed list of suitable hypotheses, do not require supervised learning, but do not adapt over time.

Other methods are based on learning the success rate of grasps given some descriptor extracted from sensor data, either evaluated on a real robotic system [10, 11], or on simulated sensor data [12, 13].

Beside vision-based methods, tactile exploration may be used for 3D reconstruction of an unknown object, as tactile sensing solves some severe limitations of computer vision, such as sensitivity to illumination and limited perspective. A combined procedure based on dynamic potential fields, that aims at reconstructing 3D object models, which are then used for grasp planning and execution was presented in [14] and recently extended in [15].

In this work, which contributes to Task 4.1, we present some results on the automatic creation of a database of grasp affordances for the Pisa/IIT SoftHand for a set of kitchenware objects. The method itself is not new since similar approaches to the grasp planning problem have been presented in the past by employing simulators as GraspIt! [16], Open-Rave [17], and Open-GRASP [?].

However, we claim that the main novelty of this contribution is related to the special hand design for which grasp affordances are sought and which we employ to reduce the complexity of the problem. In fact, due to the peculiar differential-drive type of the actuation and the built-in

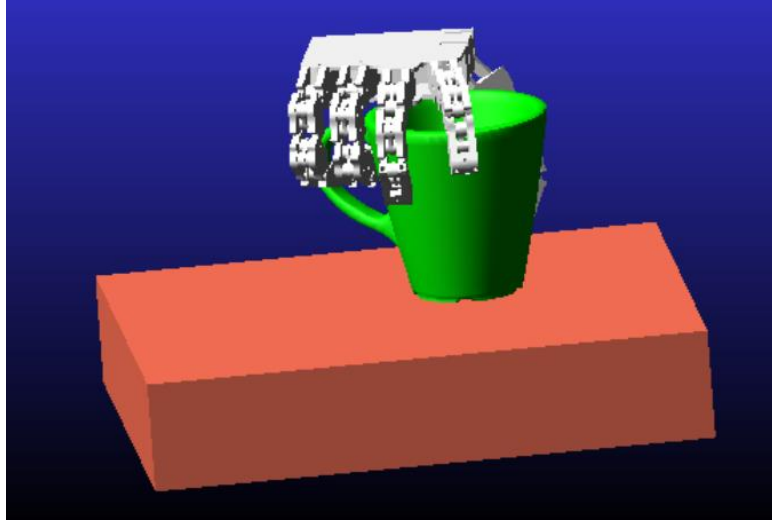


Figure 1: The Pisa/IIT SoftHand grasping a cup.

flexibility at the joint level, as the SoftHand grasps, a synergistic path is followed that allows adaptation to objects of various shapes using only a single motor [18]. This explains why, differently from other definitions based on contact point positions, in the current study palm poses in $SE(3)$ are defined as grasp affordances.

The adaptive hand behavior, entangled with the complex dynamical interactions between the hand phalanxes and the objects — the object are not fixated during the grasp acquisition — simply cannot be modelled using the physics simulators cited above, as the complex routing of a single tendon through all the limbs is, to be best of our knowledge, a fundamental feature which is out of the above physics engines' capabilities.

We modelled the Pisa/IIT SoftHand in the multipurpose and physically accurate MBS environment ADAMS [19]. In a first phase, candidate palm poses for the hand are generated by a procedure that employs random samples in the point cloud representation of the object, and selected based on their ability of presenting chances of grasp success (e.g., poses where the hand early interpenetrates the object or the object is ejected during hand closing are discarded). In a second phase, only the grasps robust with respect to perturbation wrenches on the object are considered successful and marked as feasible grasp poses. Additional local grasp candidates are generated around successful grasp and the evaluate process is repeated.

Despite the relatively naïve procedure, the mechanical intelligence embedded in the hand design allows to obtain a fairly satisfactory number of robust grasps, even for objects of complex shapes. Early experimental results confirm the numerical simulation outcomes. Grasp affordances automatically found by the presented procedure will be employed in a pipeline for automatic object recognition, grasping and bi-manual manipulation.

2 The model

In the the Pisa/IIT SoftHand two types of joints are used: the classical revolute joint and the Hillberry's joint, also known as *rolling contact joint* (RC joint) [20], represented in Figure 2. The RC joint is composed by two parts, one rolling on the other. In its original version the Hillberry's joint is able to maintain the contact between the two rolling parts by using metallic strips. Instead, in the Pisa/IIT SoftHand prototype the integrity of the RC joints, as well as of the whole structure, is assured by the tension of the actuation tendon and by the presence of elastic beams (Figure 3).

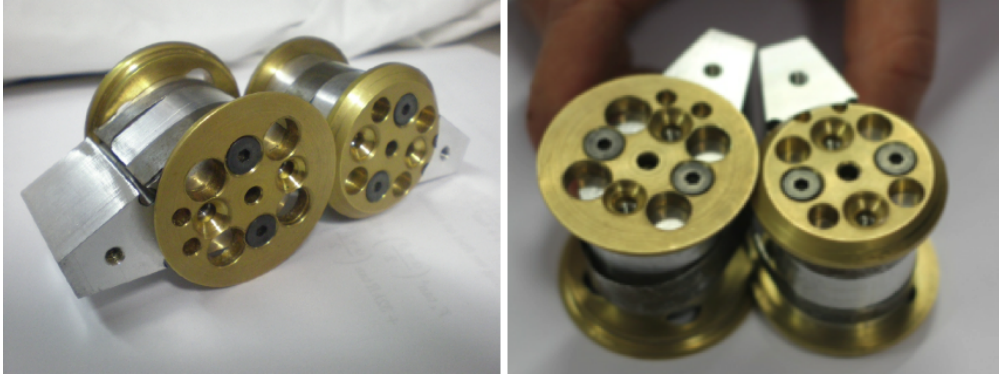


Figure 2: The Hillberry's joint, also called *rolling contact* (RC) joint. One link can move with respect to the other rolling on a circular surface.

The distribution and orientation of the joints is shown Figure 3. Other details about the Pisa/IIT SoftHand prototype can be found in [21] and in [22].

The adduction/abduction movement of the four fingers, from the index to the little, is allowed by a revolute joint with axis directed as z of the palm frame (see Figure 3). For each of these fingers, three RC joints make possible the flex/extension movement. The thumb has an opposition movement assured by a revolute joint with axis parallel to y . Moreover, two RC joints with axis directed as z in the rest configuration, allow the closure movement of the thumb.

Rolling Contact Joint From a kinematic point of view, the RC joint is equivalent to an epicyclic gearing. To make simple the study of this system, it is convenient to consider the *carrier* fixed, as in Figure 4. In this case, a relationship between the relative rotations can be expressed by the *Willis formula* as:

$$\frac{\theta_{i+1} - \theta_i}{\theta_{i-1} - \theta_i} = \frac{R_i}{R_{i+1}}. \quad (1)$$

More details about epicyclic gearing can be found in [23].

Despite the carrier is not physically present, it is convenient to consider it as a virtual link connecting the centers of the two pulleys composing the RC joint. With this interpretation, introducing q_i as the relative angle between the virtual link (carrier) and the first link, and q_{i+1}

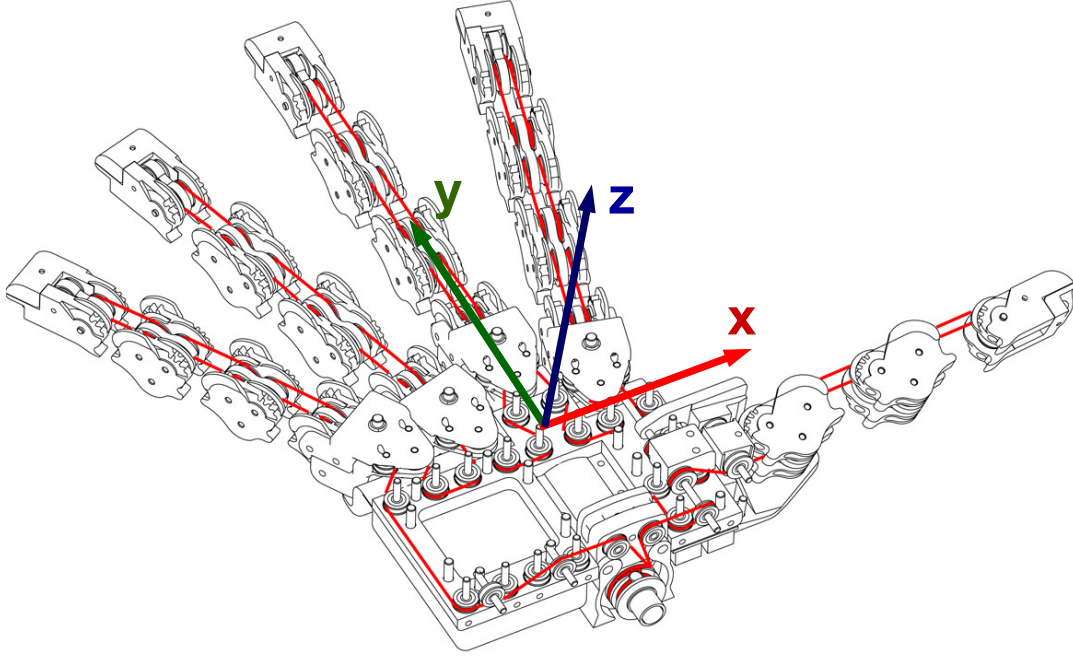


Figure 3: An exploded draw viewing of the Pisa/IIT SoftHand. In red the actuation tendon, implementing a differential effect on the finger closure, allowing the adaptability of the hand.

as the angle between the the third (considering also the virtual one), and the carrier, from (1) we obtain the formula that link the values of the joint variables.

$$R_{i+1}q_{i+1} = R_i q_i, \quad (2)$$

Tendon Routing

The closure movement of the hand is entrusted to a tendon that goes through the whole hand.

As sketched in Figure 3, the tendon starts from the unique motor of the hand. Initially it engages the pulleys on one side of the thumb. At the fingertip it turns around another pulley, before coming back toward the palm, engaging pulleys on the other side of the thumb. The tendon path goes on similarly across the other fingers, until arriving again to the motor.

Let us considering the relationship between the tendon and the joint angles displacement. More details about the equation presented in this section can be found in [24].

With similar discussion, also the relationship between tendon extremities displacements and the abduction angles can be found. It is possible to verify that this equation results nonlinear, but a linear approximation can be used without introducing relevant errors in the model.

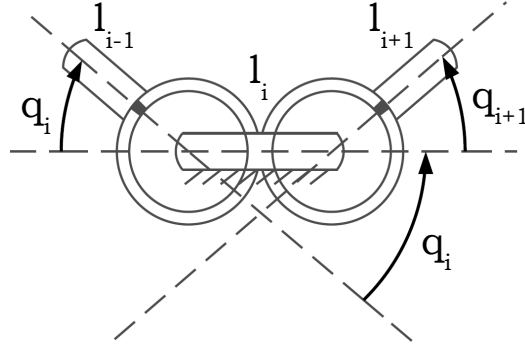


Figure 4: A schematic representation of the RC joint.

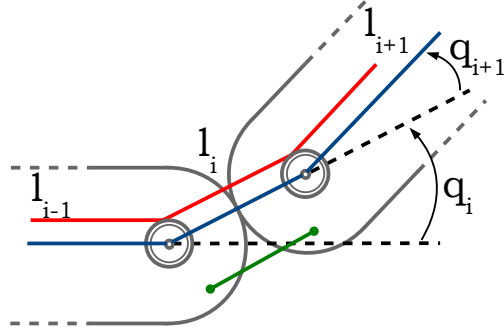


Figure 5: Lateral viewing of the use of RC joint in the Pisa/IIT SoftHand. In red the actuation tendon engaging the pulleys. In green the elastic beam connecting two consecutive links.

Consequently, globally the tendon extremity displacement $s = s^r + s^l$ (right and left tendon displacements, respectively), can be related to the joint configuration vector by:

$$s = Rq, \quad (3)$$

where $R \in \mathbb{R}^{1 \times 33}$, since that 5 are the revolute joints of the abduction and 28 the variables describing all the RC joints.

The elastic beams As stated above, the closure movement of the hand is assured by the tendon routing. On the contrary, when the actuation tendon is released by the motor, the hand can come back in the rest configuration thanks to the presence of elastic beams at joints, applied in different fashion for the RC joint and the revolute joint.

Let us consider the RC joint whose virtual link is the i^{th} , as shown in Figure 5. The elastic beam has one extremity attached to the $(i-1)^{\text{th}}$ link, and the other to the $(i+1)^{\text{th}}$. Despite the fact that the forces product by the elastic beam are internal to the structure, the simplest way to describe

their effect on the joint is to consider the two forces on the links as external, with the constraint of being equal.

Since the elastic force is actually internal to the structure, it is possible to find that only torques at the i^{th} and the $(i + 1)^{\text{th}}$ joints are needed for the equilibrium.

To demonstrate this, we can consider the spatial Jacobian (left superscript s in later equations) of the two points where elastic forces are applied.

The force application points will be called c_{i-1} and c_{i+1} , respectively, by virtue of the link name where they are attached to, as in Figure 5. The point velocities can be expressed as

$$v_{c_{i-1}} = {}^s J_{c_{i-1}} \dot{q} = [{}^s J_{i-1} \ 0] \dot{q}, \quad (4)$$

$$v_{c_{i+1}} = {}^s J_{c_{i+1}} \dot{q} = [{}^s J_{i-1} \ {}^s J_{(i-1)(i+1)}] \dot{q}, \quad (5)$$

where the matrix ${}^s J_{i-1}$ is the same in both the expressions by virtue of the description used.

By kineto-static duality, considering $f_{c_{i-1}} = -f_{c_{i+1}} = f_{el}$, it immediately follows that the total joint torque is given by

$$\tau = - \begin{bmatrix} 0 \\ {}^s J_{(i-1)(i+1)}^T \end{bmatrix} f_{el}, \quad (6)$$

that proves our statement.

In order to describe the elastic force at the joint, we indicate with $k_{el} \in \mathbb{R}$ the stiffness of the elastic beam. The elastic force can be computed as

$$f_{el} = k_{el} (\|C_{i+1} - C_{i-1}\| - l_0) \text{vers}(C_{i+1} - C_{i-1}), \quad (7)$$

where C_{i-1} and C_{i+1} are the positions of the points where the elastic forces are applied, and l_0 is the length of the elastic band in the rest configuration.

3 Model implementation in ADAMS

The simulation software ADAMS [25] gives many powerful tools to describe complex mechanical systems. However, some simplifications are needed to simulate the behavior of the Pisa/IIT SoftHand. To this aim, the kinematic study of the hand previously seen finds application. The CAD model of the parts composing the hand can be easily imported in ADAMS.

In Figure 6 the steps required to implement the RC joint in ADAMS are shown. More in detail, in Figure 6 (a) a virtual link is inserted between the two links. The virtual link is connected the real ones by virtue of a revolute joints (Figure 6 (b)). Finally the coherence between the movements of the simulated RC joint and the real one is assured introducing a gearwheel-type constraint (Figure 6 (c)).

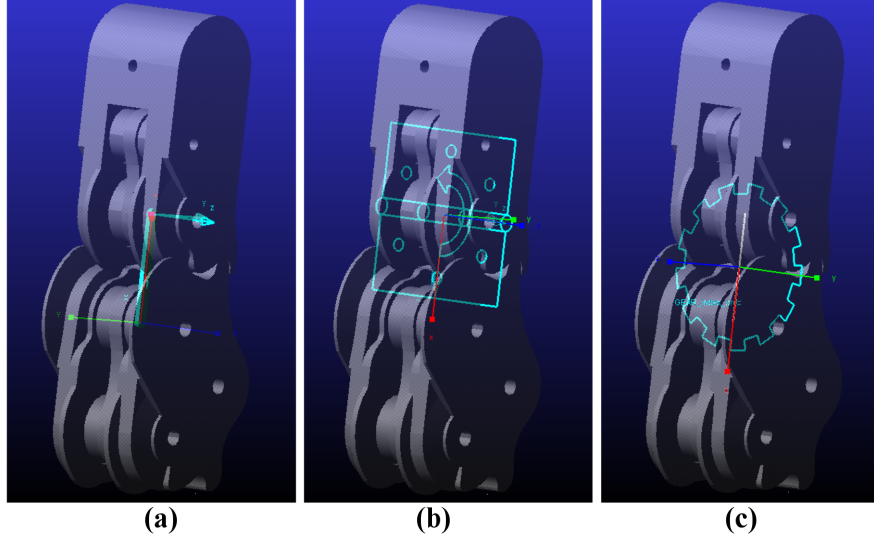


Figure 6: Implementation of the RC joint in ADAMS. A virtual link is introduced between two parts (a); a revolute joint connects the virtual link and each real part (b); a gear-wheel constraint is imposed to the two revolute joints (c).

In order to properly describe the behavior of the Pisa/IIT SoftHand, in addition to the RC joint, it is important to take into account also the kinematic constraint imposed by the tendon routing, as well as the effects of the elastic beams at the joints.

Despite the last versions of ADAMS give the possibility to introduce tendons in the model, we considered more effective taking advantage of the kinematic relationships previously seen, that is, consider the effect of the tendon kinematic constraint described in (3).

Moreover, for each RC joint we can consider a constraint relationship as in (2). From (2) it follows that, for each RC joint (14 in the hand), the number of variables can be reduced from two to one. To this aim, we introduce the reduced variable vector $q^* \in \mathbb{R}^{19}$, containing the 5 abduction joint variables, and one for each RC joint.

Let $C \in \mathbb{R}^{28 \times 19}$ be the *joint correlation matrix*, such that $q = Cq^*$. By substituting Cq^* into (3), we can easily obtain

$$s = RCq^*. \quad (8)$$

Let $\eta \in \mathbb{R}$ be the motor torque provided to the extremities of the tendon, from (8), by kineto-static duality considerations, we can write

$$\tau^* = C^T R^T \eta, \quad (9)$$

where $\tau^* \in \mathbb{R}^{19}$ is the vector of the *reduced joint torques*.

In order to take into account also the contribution of the elastic beams, we can consider a linear approximation of (6), without affecting the results. In this case, casting together the contribu-

tion of elastic forces to all the joints, and considering the correlation between joints, we can obtain

$$\tau^* = C^T R^T \eta - K_{q^*} q^*, \quad (10)$$

where $K_{q^*} \in \mathbb{R}^{19 \times 19}$ is the *reduced joint stiffness matrix*, describing how the motor torque (η) is transmitted to the joints, considering also the (negative) contribution of the elastic beams.

ADAMS allows imposing the desired function to describe to the torque for each joint, that can be computed by the knowledge of the motor torque curve versus time and the matrix C .

The contribution of the elastic torque is introduced in the model by using the ADAMS rotational spring, with also the advantage of having a damping term, useful to avoid bouncing effects in simulation, that actually are not present in the real hand due to the friction.

4 Simulation Results

In order to implement the set of simulations for the Pisa/IIT SoftHand the ADAMS model was fully parameterized via a template script (that will be described in the following) that allows us to modify all the simulation parameters, like object inertia, contact parameters, joint stiffness and friction, object and hand pose.

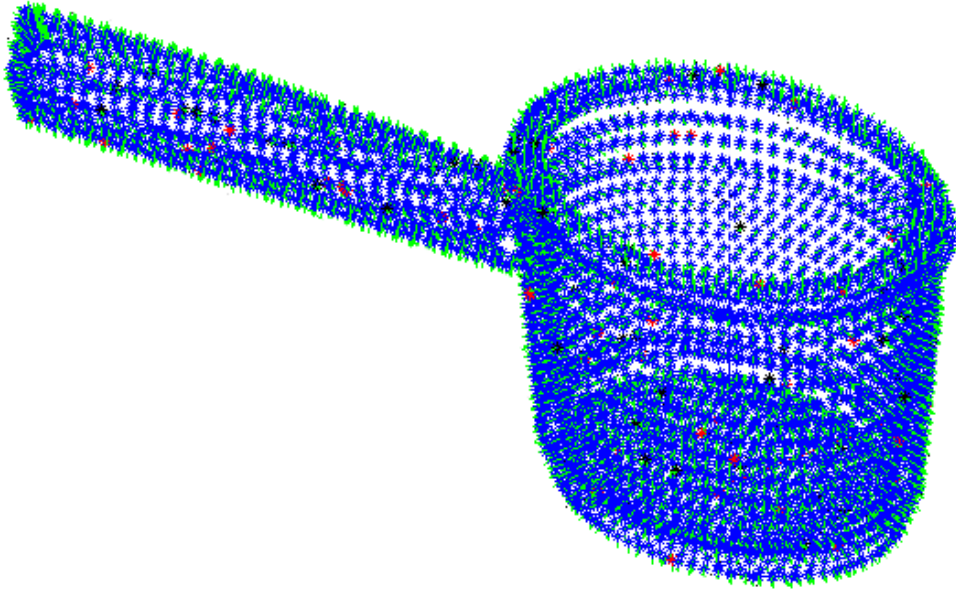


Figure 7: MATLAB representation of the points composing the surface mesh of the pot object. All points and normals are considered as candidate for the definition of the hand frame posture in order to attempt a grasp.

Since the design parameters were properly chosen to mimic as closely as possible the real robotic hand, the sole parameters to be modified are the pose of the hand with respect to the object. To this aim, and keep the number of simulations low, without sacrificing the quality of the results, a certain strategy had to be devised.

Initial Grasping Pose Selection In order to attempt object grasps, we first need to define the configuration of the hand. Initially, an extensive investigation strategy was attempted, sampling four out of the six DoFs necessary to define the position and orientation of the palm with respect to the object. More in details, the position of the origin of the palm frame was parameterized in spherical coordinates (radius, azimuth and elevation) with respect to the object frame. The z axis of the palm frame always points toward the center of the sphere (origin of the object frame). The remaining DoF was the rotation of the hand frame around its z axis. Sample hand postures were defined by defining a grid with: 4 levels on the radius, 8 on the azimuth, 8 on the elevation, 8 on the angle of rotation of the hand frame about its z axis.

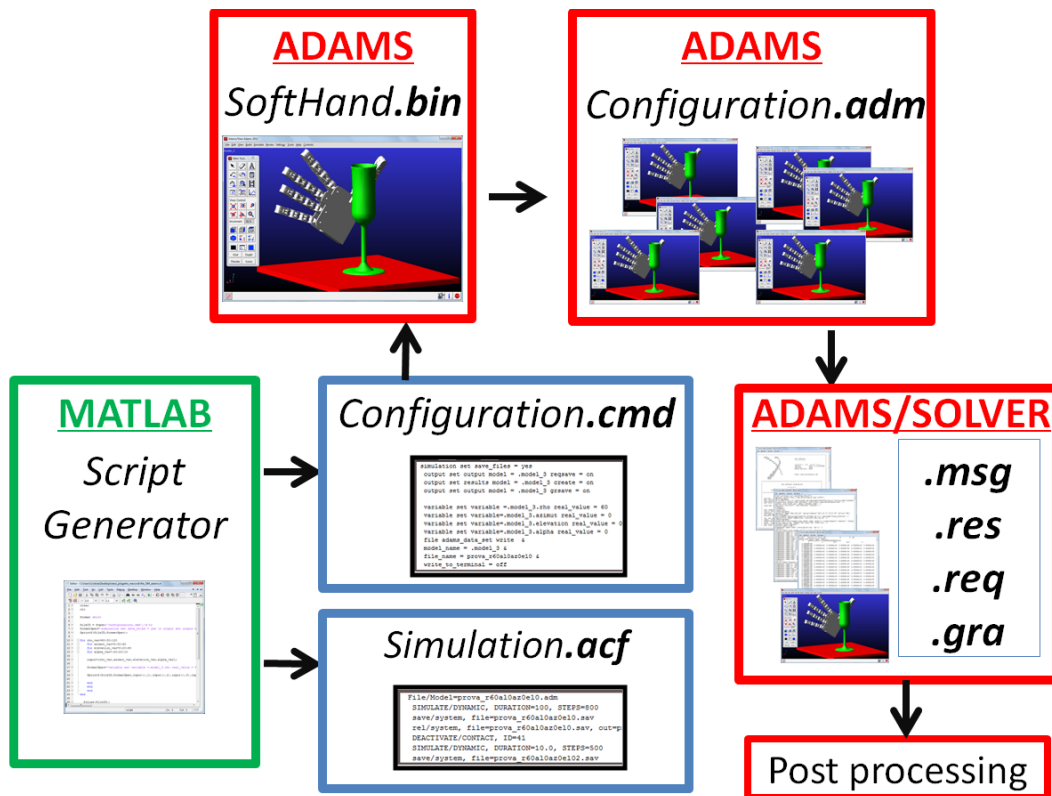


Figure 8: A flow chart of the use of ADAMS and MATLAB for running batch simulations. Firstly, a *.bin* file containing the object/hand model is loaded in ADAMS. In MATLAB a *.cmd* file is used to define the hand configurations in which attempting the grasp. By loading this file in ADAMS it is possible to obtain a file (*.adm*) for each configuration. Running the *.acf* file, the simulations start in batch mode. Simulation variables (joint angles, contact forces, etc...) are available in the exit files (*.msg*, *.res*, *.req*, *.gra*) for post-processing operations.

With this strategy a large number of attempts were unsuccessful because, in the starting configuration, the hand was either already interpenetrating the object or too far from it. In the first situation the simulation was skipped, because of the numerical instabilities due to the large contact forces which would have occurred, while in the latter the object was out of reach or, even if in contact, the ejection of the object occurred. The main reason is that a sphere is not a good approximation of many object geometry, such as a pot or a cup. This suggested us to redesign the strategy taking into account the geometry of the object: therefore, we focused the attention in the mesh describing the geometry of the object.

From the CAD files we extracted all points and normals of the mesh of each object, and considered all the points in the mesh as candidates for the position of the hand palm.

The pose of the hand frame was chosen positioning its local frame origin 5 mm outside the surface mesh along the outward normal. The palm z axis was aligned with the outward normal but with opposite direction (the palm always faces the object), and 8 configurations were selected rotating the hand around its z axis. The procedure was repeated for 50 randomly selected points of the mesh. The final number of selected poses of this initial guess is 400. A MATLAB script was written with these information, generating all the necessary files (cmd, acf and adm) for running the simulation in batch mode using the ADAMS/Solver (see Figure 8).

All the simulations was performed in two phases. In the first phase, the object to be grasped lies on a plane, orthogonal to the gravity vector, in order to give stability to object without imposing over-constraint on it. No contact was set up between the hand and the table, in order to do not preclude any possible approach direction.

This phase is executed in three seconds, that is approximately a second more of the time of free closure of the hand. The second phase starts from the final configuration of the first simulation, keeps active the hand motor torque and remove the table. This phase executes for 2 seconds.

At the end of the second phase, the velocity of the object is read, from the output files, and the grasp is considered as successful and stable if the velocity is smaller than a tolerance value (i.e., the object is still grasped by the hand).

Search Refinement

From the results of the first set of simulations, the points that brought to a stable grasp were extracted.

In order to evaluate and rank each grasping pose, we further search around the good points, by sampling their neighbors.

This task was performed by choosing randomly 10 of the 40 closest points of the mesh, around the successful one. For each new point, again 8 rotations around the palm normal was considered to attempt the grasp. Again, the simulations was performed in two parts, and the stability of the grasps were evaluated as previously explained.

It is important to note that the use of a dynamic simulator allows us to do not explicitly manage

	successful postures	successful points	clouds	best cloud
Cup	30	13	8	10
Pot	138	41	16	35
Strainer	21	6	2	14
Plate	112	29	10	37

Table 1: The table summarizes the simulation results: (i) the number of initial postures bringing to a grasp (successful), (ii) the number of successful individual starting points for the hand placement, (iii) the number of clouds studied and (iv) the maximum number of successful configurations found in a cloud (best cloud).

the incipient contact situation. This fact, together with the adaptability of the Pisa/IIT Soft-Hand, brings to an automatic management both of the contact point positioning and of the grasp achievement phases, that, otherwise, may have turned out to be a tricky phase.

In Tables 1 and 2 some simulation results are summarized. More in details, in Table 1, for every object we list:

- (i) the number of palm frame configurations (green frames in Figures 13-16) bringing to a stable grasp (**successful postures**),
- (ii) the number of different origin positions of hand frame, multiple orientations for the same origin position are not counted (**successful postures**),
- (iii) the number of clouds investigated (**clouds**),
- (iv) the maximum number of successful grasp in a cloud (**best cloud**).

Sketches of the tested configuration are represented in Figures 13-16. In particular: (a) figures represent a MATLAB plot of the object mesh points (blue) and of the outward normal vector to the surface (green vector).

(b) figures show the results after the investigation of the first 50 points: in green, we depict the initial points that were successful, bringing to a stable grasp, in red the others.

(c) figures show the results after the investigation of the neighbors of the initial successful points.

(d) figures show the only successful configuration found.

As we can see, the best result in terms of number of grasps achieved, as well as in terms of individual successful points, is obtained for the pot. However, the best cloud found is for the

	CQ Index %	Closure Index (min-max) %	Net Force (N)
Cup	11.3	40-73	1-57
Pot	39.7	48-68	2-36
Strainer	15.9	54-67	11-52
Plate	42.1	57-69	6-28

Table 2: The table summarizes some simulation results and quality measures. The *Cloud Quality* (CQ) index reports the percentage of successes (stable grasp achieved), with respect to the number of postures attempted, in the best cloud. The *Closure* index (CI) indicates how much the hand is near to the complete closure configuration, the minimum and maximum values found for all the successful grasps. The *Net Force* (NF) measures the amount of contact force exceeding the weight of the object (minimum and maximum).

plate.

In Table 2, simple quality indexes are listed. In particular, the *cloud quality* (CQ) index reports the highest percentage of stable grasps achieved in the tested clouds. More in details, the CQ index is computed as:

$$CQ = \frac{g_b}{c_b} 100, \quad (11)$$

where g_b is the number of stable grasp achieved and the c_b is the number of hand palm configurations tested, both considered for the best cloud. The *closure* index (CI) is a measure of how much the hand is closed at the end of the simulation. In Table 2 are shown the minimum and maximum values found for each object. The index is computed as

$$CI = \frac{100}{n_{qc}} \sum_{i=1}^{n_{qc}} \frac{q_{f_i}}{q_{c_i}}, \quad (12)$$

where q_{f_i} is the final configuration of the i^{th} joint during the grasp, and q_{c_i} is the maximum reachable value, allowed by the mechanical constraints, for the i^{th} joint, n_{qc} is the number of *flex-extension joints*, that is the total number of joints minus the abduction joints.

The *Net Force* (NF) is a measure of the amount of internal forces produced in the grasp. It is computed as

$$NF = \left(\sum_{i=1}^c f_{c_i}^T f_{c_i} - w_o^T w_o \right)^{\frac{1}{2}}, \quad (13)$$

where c is the number of contact points, f_{c_i} is the force at the i^{th} contact point, and w_o is the weight of the grasped object.

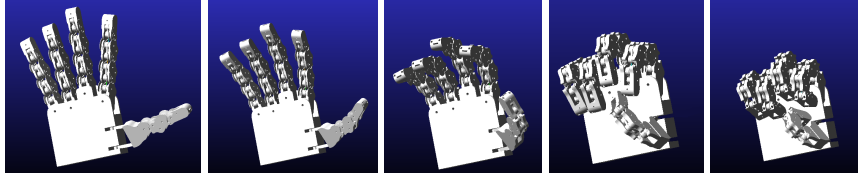


Figure 9: Sketches of the free closure movement of the Pisa/IIT SoftHand in ADAMS simulation.

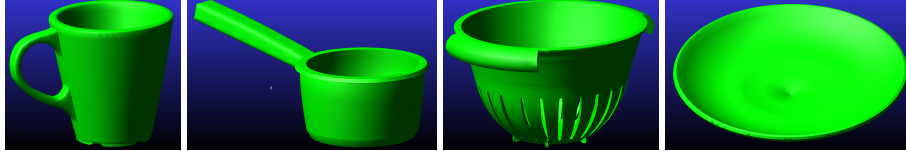


Figure 10: The set of objects used to attempt the grasps.

As it is evident from Table 2, the best *cloud quality* index is achieved for the plate. Also the highest values for the *closure index* are realized grasping the plate. The reason of this can be due to the fact that all the tests are performed with the same torque-time motor curve. In some cases, specially for the plate, the results show that the amount of motor torque is enough to overcome the friction and to bring the hand in a closure configuration (see also last panel in Figure 20). The greater amount of the NF index, is realized to grasp the strainer. The reason of this is, again, an excessive level of motor torque, in particular with respect to the low weight of the object.

Generally speaking, the not excellent results of the strainer (just 2 clouds and 21 successful configurations) can be explained with the difficulty of randomly selecting points near to the upper edge or near the handles. Moreover, the meshes of the objects, obtained from CAD files, are characterized by the presence of (nominally) normal vectors, that, indeed, are not orthogonal to the object surface. In some cases, an hand/object compenetrations can occur, caused by a palm normal orientation not orthogonal to the object surface, and the selected points (potentially good) is discarded. For similar reasons, we can consider the cup and the pot penalized. Potentially, more points could be found on their handle having a better representation of both the mesh points and the normal vectors to the surface.

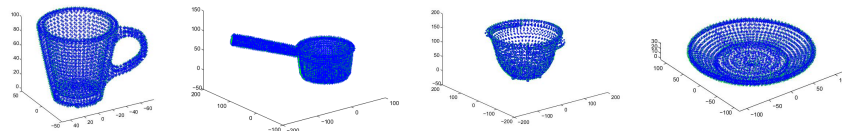


Figure 11: MATLAB representation of the objects tested.

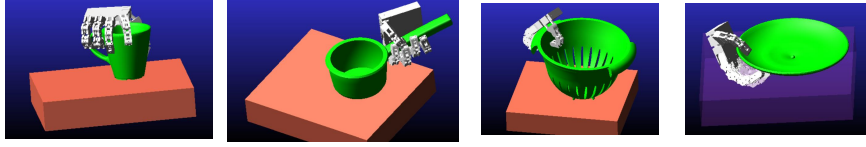


Figure 12: Examples of stable grasps achieved with the Adams simulations.

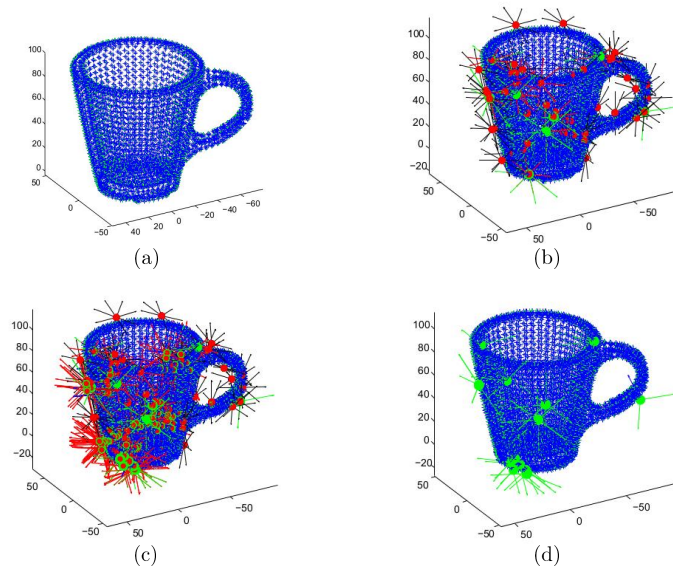


Figure 13: Matlab representation of grasping attempt results for the cup. Red points correspond to palm configuration that did not bring to a stable grasp. Green points correspond to successful hand configurations. Figure (b) shows the results for the first 50 points, Figure (c) shows the results for all the points tested. Figure (d) shows only successful configurations.

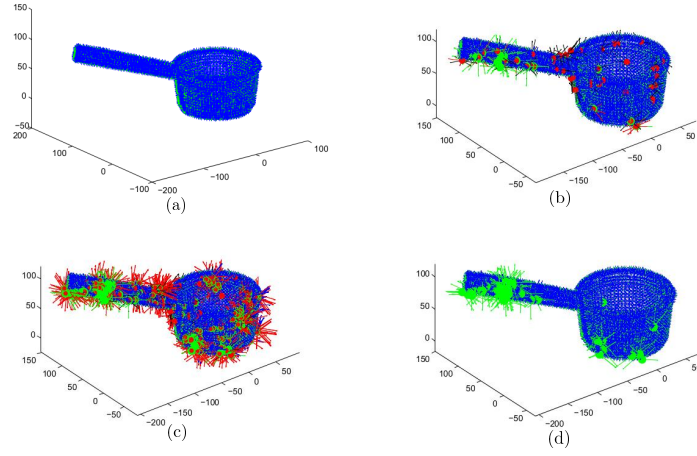


Figure 14: MATLAB representation of grasping attempt results for the pot. The order of the figures follows the same criterion of Figure 13.

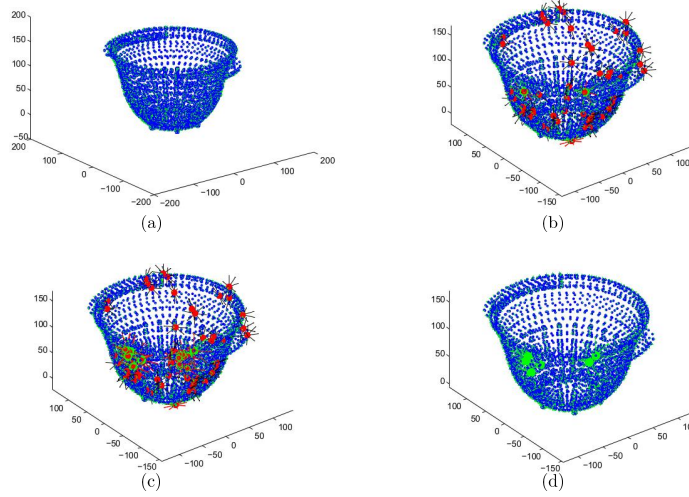


Figure 15: MATLAB representation of grasping attempt results for the strainer. The order of the figures follows the same criterion of Figure 13.

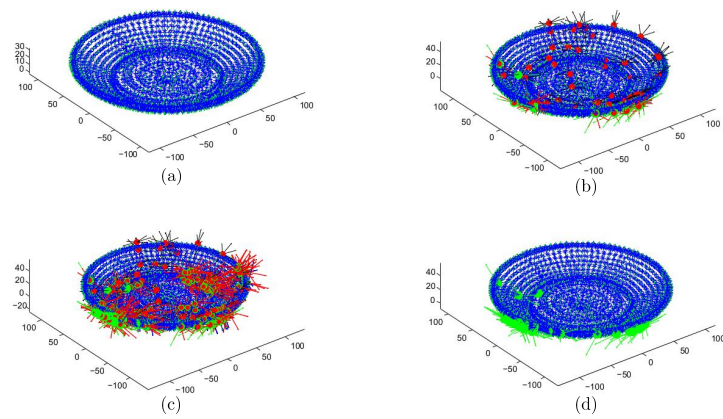


Figure 16: MATLAB representation of grasping attempt results for the plate. The order of the figures follows the same criterion of Figure 13.

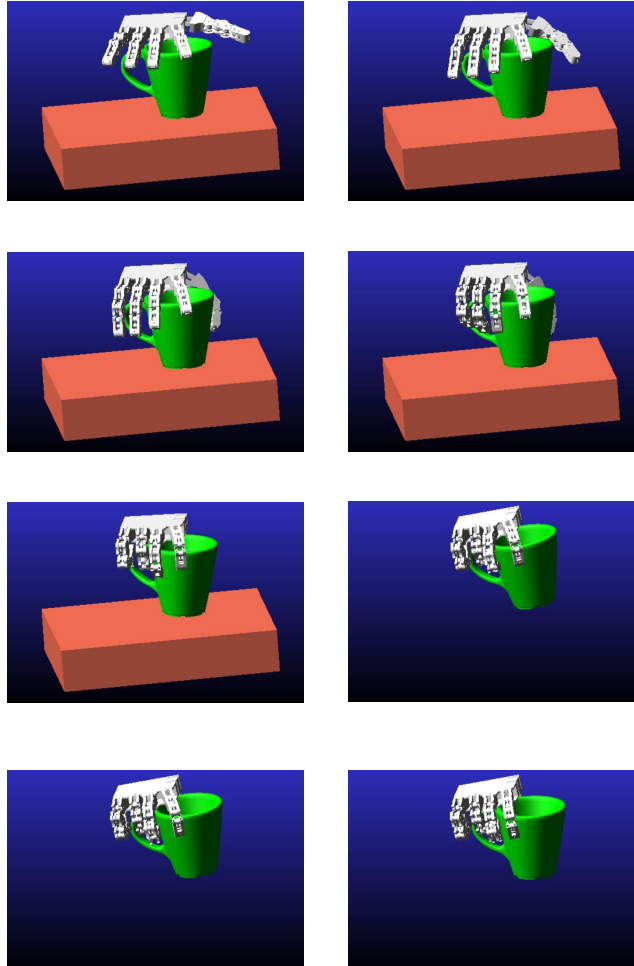


Figure 17: Slide show of a successful grasp of the cup. In the first three seconds of simulation the object rests on a table (slides 1 – 5). Later the table is removed to verify the stability of the grasp achieved (slides 6 – 8).

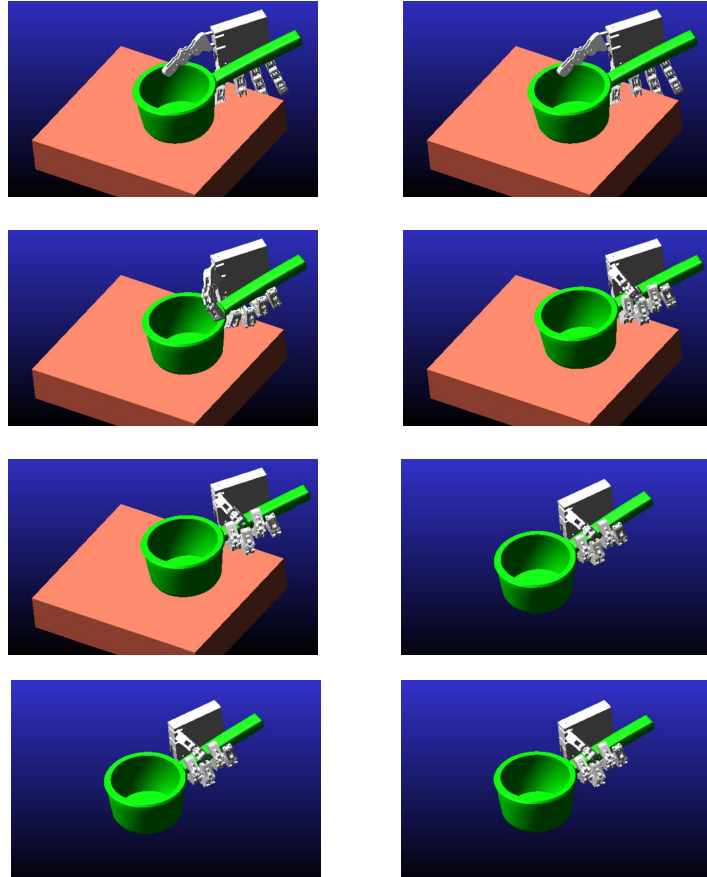


Figure 18: Slide show of a successful grasp of the pot. In the first three seconds of simulation the object rests on a table (slides 1 – 5). Later the table is removed to verify the stability of the grasp achieved (slides 6 – 8).

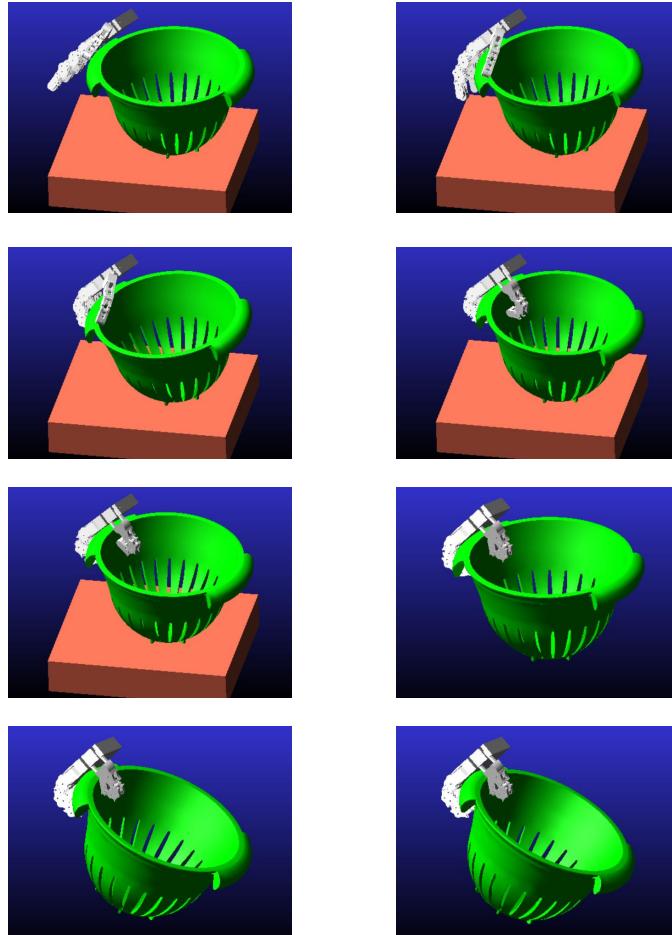


Figure 19: Slide show of a successful grasp of the strainer. In the first three seconds of simulation the object rests on a table (slides 1 – 5). Later the table is removed to verify the stability of the grasp achieved (slides 6 – 8).

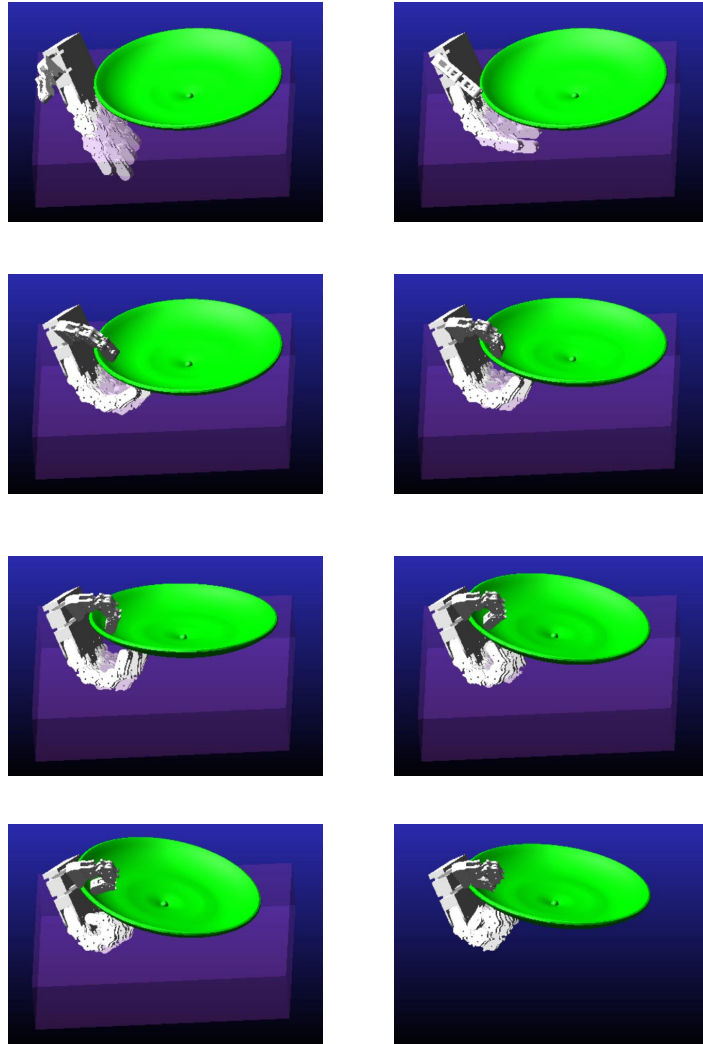


Figure 20: Slide show of a successful grasp of the plate. In the first three seconds of simulation the object rests on a table (slides 1 – 7). Later the table is removed to verify the stability of the grasp achieved (slide 8).

5 Conclusions

The Pisa/IIT SoftHand is a novel prototype of robotic hand that thanks to its adaptivity characteristics can successfully grasp a wide variety of every day objects. This property was initially verified in the first prototypes attaching the hand to mechanical interface, allowing a human operator to directly control the pose of its wrist. However, in order to use the SoftHand in a more complex robotic scenario, some strategies about how posing the palm with respect the object to grasp is needed. For this kind of application, a systematic way to investigate to goodness of the hand poses is useful also to evaluate to probability to achieve a stable as well as to build an initial data set, useful base for later learning methods. Thanks to the adaptivity of the SoftHand, stable grasps can be searched investigating the hand-object relative configurations space, without needing an exhaustive research of the possible contact points. During the closure phase, the differential mechanism governing the synergistic movement of the hand allows a natural enclosing of the object shape, when it is possible, or to reject the object, otherwise.

References

- [1] C. Ferrari and J. Canny, “Planning optimal grasps,” *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pp. 2290–2295, Jan. 1992.
- [2] D. Prattichizzo, J. Salisbury, and A. Bicchi, “Contact and grasp robustness measures: Analysis and experiments,” *Experimental Robotics IV*, Jan. 1997.
- [3] C. Rosales, R. Suarez, M. Gabiccini, and A. Bicchi, “On the synthesis of feasible and prehensile robotic grasps,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 1–7.
- [4] M. Teichmann and B. Mishra, “Reactive algorithms for grasping using a modified parallel jaw gripper,” in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, 1994, pp. 1–6.
- [5] R. Platt, A. H. Fagg, R. A. Grupen, W. Bluethmann, R. Ambrose, M. Diftler, E. Huber, A. Fagg, M. Rosenstein, R. Grupen, C. Breazeal, A. Brooks, A. Lockerd, R. A. Peters, O. C. Jenkins, M. Mataric, and M. Bugajska, “Null-Space Grasp Control: Theory and Experiments,” *Robotics, IEEE Transactions on*, no. 2, pp. 1–14, 2010.
- [6] D. Wang, B. Watson, and A. Fagg, “A switching control approach to haptic exploration for quality grasps,” in *Proceedings of the Workshop on Robot*, Jan. 2007, pp. 1–8.
- [7] K. Hsiao, S. Chitta, M. Ciocarlie, and E. G. Jones, “Contact-reactive grasping of objects with partial shape information,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 1–8.

- [8] E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A. Y. Ng, and O. Khatib, “Grasping with application to an autonomous checkout robot,” *2011 IEEE International Conference on Robotics and Automation*, pp. 2837–2844, May 2011.
- [9] F. T. Pokorny, J. A. Stork, and D. Kragic, “Grasping objects with holes: A topological approach,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1100–1107, 2012.
- [10] L. Montesano, M. Lopes, A. Bernardino, J. Santos-Victor, F. S. Melo, and R. Martinez-Cantin, “Learning grasping affordances from local visual descriptors,” in *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on*, 2009, pp. 1–6.
- [11] R. Detry, E. Başeski, M. Popović, and Y. Touati, “Learning continuous grasp affordances by sensorimotor exploration,” *From motor learning to*, Jan. 2010.
- [12] J. Bohg and D. Kragic, “Learning grasping points with shape context,” *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 362–377, Apr. 2010.
- [13] A. Saxena, J. Driemeyer, and A. Y. Ng, “Robotic Grasping of Novel Objects using Vision,” *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, Feb. 2008.
- [14] A. Bierbaum and M. Rambow, “Grasp affordances from multi-fingered tactile exploration using dynamic potential fields,” *Humanoid Robots*, Jan. 2009.
- [15] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, J. Bohg, T. Asfour, and S. Schaal, “Learning of grasp selection based on shape-templates,” *Autonomous Robots*, vol. 36, no. 1-2, pp. 51–65, 2013.
- [16] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, “Automatic grasp planning using shape primitives,” *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, pp. 1824–1829, Jan. 2003.
- [17] R. Diankov and J. Kuffner, “Openrave: A planning architecture for autonomous robotics,” pp. 1–18, Jan. 2008.
- [18] B. León, S. Ulbrich, R. Diankov, G. Puche, M. Przybylski, A. Morales, T. Asfour, S. Moio, J. Bohg, J. Kuffner, and R. Dillmann, *OpenGRASP: A Toolkit for Robot Grasping Simulation*. Springer Berlin Heidelberg, Jan. 2010.
- [19] M. Catalano, G. Giorgio, A. Serio, and E. Farnioli, “Adaptive synergies for a humanoid robot hand,” *on humanoid robots*, pp. 1–8, Jan. 2012.
- [20] M. ADAMS. (2014, Jan.) Msc software last accessed. [Online]. Available: <http://web.mscsoftware.com/>
- [21] B. Hillberry and A. Hall Jr, “Rolling contact joint,” Jan. 13 1976, uS Patent 3,932,045.
- [22] M. G. Catalano, G. Grioli, A. Serio, E. Farnioli, C. Piazza, and A. Bicchi, “Adaptive synergies for a humanoid robot hand,” in *IEEE-RAS International Conference on Humanoid Robots*, Osaka, Japan, 2012.

- [23] M. G. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi, "Adaptive synergies for the design and control of the pisa/iit soffthand," *International Journal of Robotics Research*, Submitted.
- [24] J. Shigley and J. Uicker, *Theory of machines and mechanisms*, ser. McGraw-Hill series in mechanical engineering. McGraw-Hill, 1980.
- [25] L.-W. Tsai, *Robot Analysis and Design: The Mechanics of Serial and Parallel Manipulators*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1999.
- [26] M. Software, *Getting Started Using Adams/View*, MSC Software Corporation, 2012, software available at <http://www.mscsoftware.com/product/adams>. [Online]. Available: <http://www.mscsoftware.com/product/adams>

Case Based Reasoning for Grasp Synthesis

Michaela Antonelli and Marco Gabiccini

PaCMan Technical Report - February 2014

1 Introduction

Case-Based Reasoning (CBR) means using old experiences to understand and solve new problems [1]. In CBR, a reasoner remembers a previous situation similar to the current one and uses that to solve the new problem. Generally, when a problem will be presented to a case-based reasoner, it searches in its memory of past cases (the case base) and attempts to find a case that has the same problem specification as the current case. If the reasoner cannot find an identical case in its case base, it will attempt to find the case or cases most similar to the current case in the case base. If an identical case is found, its solution is returned as the current solution of the problem. On the other hand, if the retrieved case is not identical to the current case, an adaptation phase occurs. In this phase, the differences between the current and retrieved case are taken into account to adapt the solution associated with the retrieved case. The adapted solution is then returned. Figure 1 shows a typical block diagram of a CBR system. The case-based reasoning mechanism is usually split into two major parts, namely the case retriever and the case reasoner. The former searches for appropriate cases in the case base, while the latter uses the retrieved cases to find a solution to the given problem case (derived solution).

The main advantage of a CBR system is that it does not require a lot of specific information on the problem domain: A CBR system can be created with a small amount of experience and incrementally developed, adding more cases to the case base, as they become available.

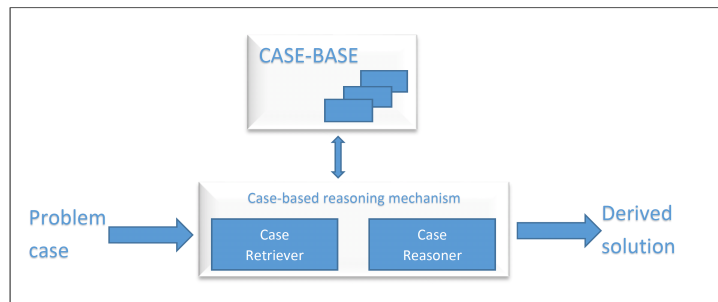


Figure 1: A Case-Based Reasoning system.

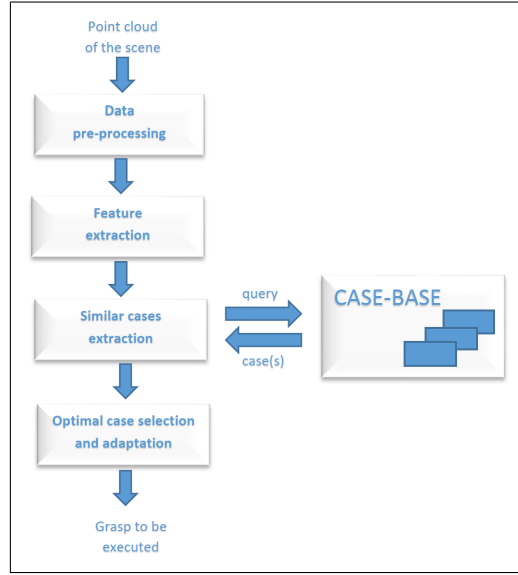


Figure 2: CBR model for grasp synthesis.

Indeed it offers several benefits such as the facilitation of the knowledge acquisition process, the capability of learning from past experience and the avoidance of the complete understanding of the domain.

Our aim is to build a CBR system able to find, for a given pair (hand, object), the suitable grasp configuration among the set of feasible grasp candidates. Generally, each case is represented as two sets of attributes: the former specifies the problem and the latter the solution. In particular, the problem is described by the features of the point cloud of a scene and the solution by the parameters of the grasp. These parameters are:

- i) the contact points, i.e., the position of the thimbles w.r.t. the object;
- ii) the position of the wrist w.r.t the object;
- iii) the contact forces applied to the contact points;

In Figure 2 we show the model of the proposed CBR-based system for grasp synthesis. The input of the system is a point cloud of a scene containing the object to be grasped, and the output is the set of the parameters of the grasp to be executed. In the following subsections we explain in detail the single modules of the CBR-based system.

2 The data pre-processing module

The data pre-processing module performs the segmentation of the object from the scene. In particular, the input of the module is the point cloud containing all the scene and its output is the

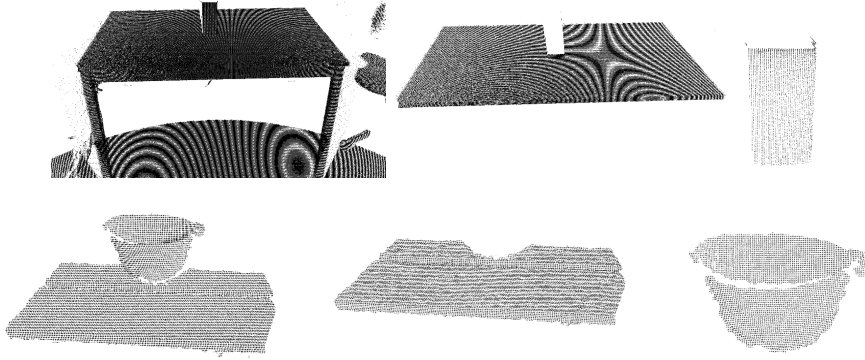


Figure 3: Top row: example glass segmentation. Bottom row: example container segmentation.

point cloud containing only the object. At the moment, we are considering very simple scenes consisting of only one object on the table. In this case, the point cloud related to the object can be extracted by using some functions defined in the PCL library [2]. In particular, we apply the following procedure:

1. find the area related to the table. This means to look for cluster of points that form planar shapes. In our experiments, the planar cluster will be related to both the floor and the table. Within these two clusters, we identify the cluster representing the table as the one with the lower number of points;
2. calculate the average value of the height of the table (i.e., the distance between the floor and the table);
3. find all other clusters related to the other objects that can be recorded by the kinect sensor;
4. identify the object as the cluster that has the same height of the table.

In Figure 3 we show an application of this procedure to two example scenes. In each row, the first image corresponds to the scene, the second to the cluster of the table, and the last to the cluster of the object.

2.1 Feature extraction

Once isolated the point cloud of the object, we extract a set of meaningful features for describing the case represented by the object. To this aim we use two different types of approaches, namely a *global approach* and a *local approach*. The former describes the object by means of features calculated on the point cloud taken as a whole (i.e., a single descriptor is calculated for each object), while the latter determines some points of interest on the object, called keypoints and calculates the descriptors on each circular area around these keypoints. In particular, we use the SIFT Keypoint Detector algorithm [3] to extract the keypoint from the point cloud. For local approaches, an object is described by a collection of descriptors. As shown in Figure 4, both the

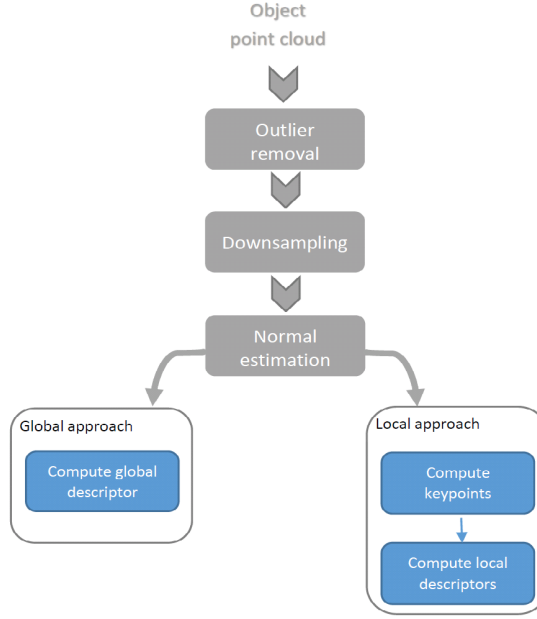


Figure 4: Pipeline for feature extraction.

approaches require some preliminary steps.

The first step removes outlier from the point cloud. An outlier is a point that is distant from other points; it could arise from sensor noise or simply from a bad acquisition (bad light). In the literature there are several methods for outlier removal: one of the most used is based on a statistical analysis of each point neighborhood [4].

Let p be a point of the point cloud P , P^K the set of k nearest points of p , and $\bar{d}_p$ the mean distance between p and each point in P^K . We calculate the mean μ_p and the standard deviation σ_p of $\bar{d}_p$ and we remove from the point cloud the points that are not in the following set P^* :

$$P^* = \{p^* \in P \mid (\mu_p - \alpha * \sigma_p) \leq d^* \leq (\mu_p + \alpha * \sigma_p)\} \quad (1)$$

The parameter α is a weight that controls the width of the interval, making the filtering process more or less restrictive (we consider $\alpha = 1$ in our experiment) . In Figure 5 we show an application of this algorithm. We used the implementation in the pcl library:

(http://pointclouds.org/documentation/tutorials/statistical_outlier.php).

After the outliers have been removed, in the following step the down-sampling of the point cloud is applied so as to reduce the number of points and therefore to speed up the next step of normal estimation.

In order to preserve the shape of the object, we cannot just remove the points randomly. Indeed, the random downsampling may damage/collapse the original models, affecting the final outcome

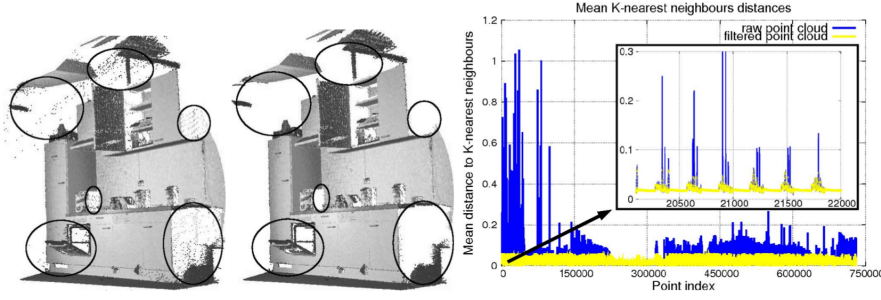


Figure 5: Example of outlier filter procedure: a point cloud with many outliers (left), the resulting point cloud after the filter (center) using $\alpha = 1$ and a graphic with mean k-nearest neighbor distances in a point neighborhood before and after filtering (right).

of the process. Thus, we need to employ an appropriate subsample procedure to produce a point cloud with points uniformly distributed.

To this aim, we apply the Voxel Grid filter, defined in the pcl library, which consists of a smart down-sampling of point clouds. The points are sub-divided into a set of voxels (volumetric pixels): all points inside each voxel are approximated with the coordinates of their centroid. The voxel size defines the amount of down-sampling applied. By default, we choose the voxel size to be 3mm.

The step of normal estimation is based on the first order 3D plane fitting technique introduced in [5]. In this method, the problem of determining the normal to a point on the surface is approximated by the problem of estimating the normal of a plane tangent to the surface, which is a least-square plane fitting estimation problem. The initial problem is therefore reduced to an analysis of the eigenvectors and eigenvalues of a covariance matrix created from the nearest neighbors of the query point. More specifically, for every point p_i the following covariance matrix is calculated:

$$C = \frac{1}{k} \sum_{i=1}^k (p_i - \bar{p}) \cdot (p_i - \bar{p})^T \quad (2)$$

where k is the number of neighbors considered in the neighborhood of p_i ; $\bar{p}$ represents the 3D centroid of the nearest neighbors, and is calculated as:

$$\bar{p} = \frac{1}{k} \cdot \sum_{i=1}^k p_i \quad (3)$$

Since the covariance matrix C is symmetric and positive semi-definite, its eigenvalues are positive real numbers and consequently the associated eigenvectors form an orthogonal frame. So if the eigenvalues are ordered ($0 \leq \lambda_0 \leq \lambda_1 \leq \lambda_2$) the eigenvector $\mathbf{v}_0$ corresponding to the smallest eigenvalue λ_0 is therefore the approximation of the normal which we were searching for.

Global Methods	Local Methods
PFH	PFH
VFH	FPFH
ESF	SHOT
	RSD

Table 1: Methods for estimating 3D descriptors.

The choice of the number k of points belonging to the neighborhood of a point p is a crucial aspect for both the step of down-sampling and normal estimation. In Figure 6 we show the effect of selecting a smaller radius of neighborhood versus a larger radius of neighborhood in the normal estimation step.

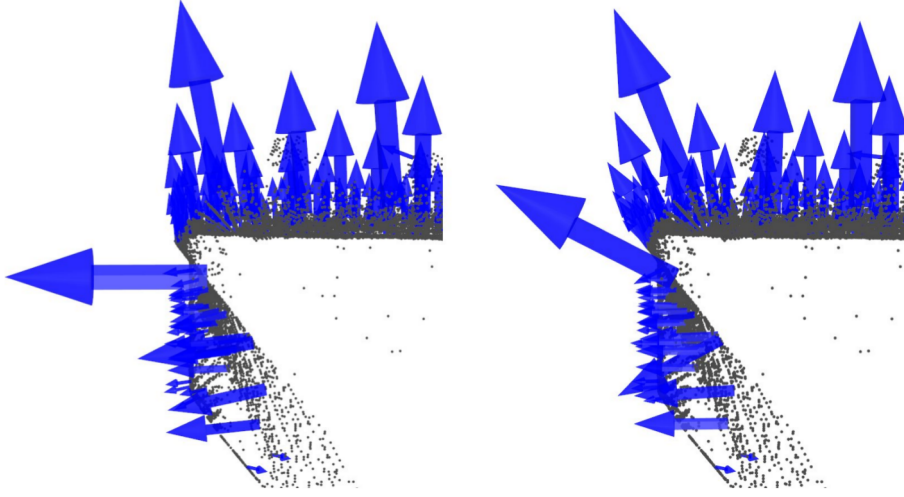


Figure 6: Normal estimation in two cases: (left) a reasonably good choice of radius, (right) the radius chosen is too big and the normals get estimated over points from different surfaces.

For our purposes, the radius for the determination of a point's neighborhood has to be selected based on the level of detail required. As an example, if the curvature at the edge between the handle of a mug and the cylindrical part is relevant, the radius needs to be small enough to catch those details.

Once the normals at each point of the cloud have been calculated, we can carry on with the feature extraction step by using local or global approaches. In Table 1 we show some of these approaches which estimate three-dimensional descriptors of a point cloud.

In the following, we will discuss in detail all the approaches shown in Table 1.

Point Feature Histogram (PFH) The PFH method [6] computes a histogram which encodes the local neighborhood's geometrical properties of a point p . The algorithm can be either local or global depending on the choice of the radius r_k that defines the k -neighbors around p .

Let r_m be the minimum radius that contains all the points of the cloud. If the radius r_k is equal to r_m , then the algorithm is considered global as it calculates a single global descriptor for the object. On the other hand, if the radius r_k is smaller than r_m , many local descriptors will be computed for the point cloud and the approach is considered local.

Let $KP = \{kp_1, kp_2, \dots, kp_n\}$ be a set of keypoints, (where $n = 1$ for the global approach and $n=keypoints_number$ for the local approach), the algorithm PFH can be summarized as follows:

For each keypoint kp_i in the cloud:

- i) Collect all points p_j (neighbored of Kp_i) within a sphere of radius r_k centered in kp_i
- ii) For each pair q_z of points p_{j1}, p_{j2} in the neighborhood of kp_i , evaluate the estimated normals η_{j1}, η_{j2} , respectively. Select a source p_s and a target p_t , the source being the one having the smaller angle between the associated normal and the line connecting the points:
 - (a) if $\langle \eta_{j1}, p_{j2} - p_{j1} \rangle \leq \langle \eta_{j2}, p_{j1} - p_{j2} \rangle$ then $p_s = p_{j1}, p_t = p_{j2}, \eta_s = \eta_{j1}, \eta_t = \eta_{j2}$
else $p_s = p_{j2}, p_t = p_{j1}, \eta_s = \eta_{j2}, \eta_t = \eta_{j1}$
 - (b) define the Darboux frame (u, v, w) as:
 $u = \eta_s, v = (p_t - p_s) \times u / \|p_t - p_s\|, w = u \times v$ and with the origin in the source point (see Figure 7)
 - (c) calculate four features, that measure the differences between the points' normal and the distance vector between them.

$$\begin{aligned} f_{z,0} &= \langle v, n_t \rangle \\ f_{z,1} &= \|p_t - p_s\| \\ f_{z,2} &= \langle u, p_t - p_s \rangle / f_{z,1} \\ f_{z,3} &= \arctan(\langle w, n_t \rangle, \langle u, n_t \rangle) \end{aligned} \quad (4)$$

- iii) compute the minimum $f_{i_{min}}$ and maximum $f_{i_{max}}$ of each feature f_i
- iv) compute for each pair q_z the idq_z value defined as

$$idq_z = \sum_{i=0}^{i \leq 3} \left\lfloor \frac{f_{zi} \cdot d}{f_{i_{max}} - f_{i_{min}}} \right\rfloor \cdot d^i \quad (5)$$

where $\lfloor \cdot \rfloor$ is the integer part of the fraction, f_i is the i -th feature, and d is the number of subdivision of the range $f_{i_{max}} - f_{i_{min}}$

Fast Point Feature Histogram (FPFH) The Fast Point Feature Histogram [7] is an optimization of the PFH method. It reduces the computational time of the descriptor calculation by ordering the point cloud, thus avoiding to re-calculate previously computed values of the features.

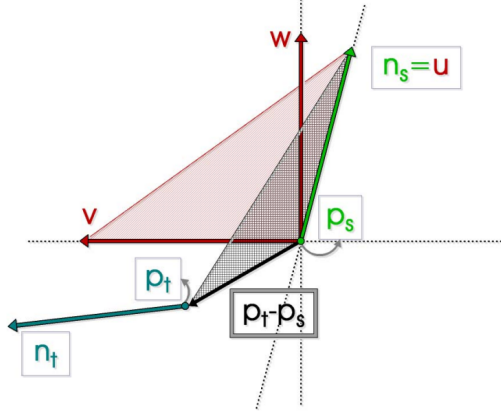


Figure 7: The Darboux frame placed at p_s .

In particular, given two points, p_1 and p_2 , if they are neighbors, not all the features have to be recomputed in their neighborhoods. Indeed, many points of the p_1 's neighborhood will also be part of the p_2 's neighborhood. We can share the histograms corresponding to these points by saving them when they are computed for p_1 and re-using them for p_2 .

In this way the algorithm has a complexity equal to $O(n \cdot k)$, rather than $O(n \cdot k^2)$ of PFH, where n is the number of points of the cloud and k the number of neighbors considered for each point.

The FPFH algorithm can be summarized as follows:

- i) Collect all neighboring points within a sphere of radius r_k centered in kp_i
- ii) Calculate the Darboux frame as in the PFH method but with some approximations. In particular, for each keypoints (kp_i), consider only the relationships between itself and its neighbors (in PFH, for each (kp_i), we consider pairs of kp_i and its neighbors and also between kp_i 's neighbors). This simplified version is called Simplified Point Feature Histogram (SPFH).
- iii) calculate three of the four features used in PFH, leaving out f_1 .
- iii) Bin the features into the histogram as in PFH.
- iv) For each point re-determine its k neighbors and use the neighboring SPFH values to weight the final histogram of kp_i (called FPFH):

$$FPFH(kp_i) = SPFH(kp_i) + \frac{1}{k} \sum_{i=1}^k \frac{1}{\omega_k} \cdot SPFH(p_k) \quad (6)$$

where w_k is the distance between kp_i and the neighbor point p_k

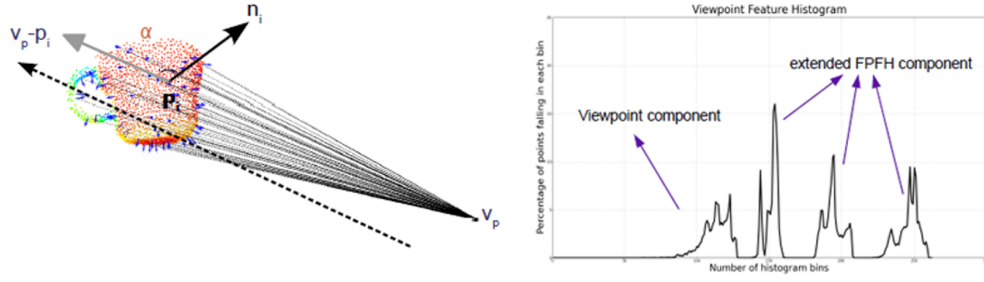


Figure 8: Viewpoint Feature Histogram computation.

Viewpoint Feature Histogram (VFH) The Viewpoint Feature Histogram algorithm [8] extends the FPFH by including viewpoint information into the resulting histogram, so as to be used to recognize both the object and its pose. For this reason, VFH is made of two components (see Figure 8): a viewpoint component and an extended FPFH component that are generated as described in the following.

- i) Calculate the centroid P_c of the point cloud and its normal N_c , calculate the vector between the viewpoint and the centroid V_c and normalize it.
- ii) To map the viewpoint component iterate over all points in the cloud and calculate the angle between their normal and V_c , incrementing the corresponding histogram bin.
- iii) For the FPFH component simply calculate the FPFH at the centroid P_c considering as neighbors all the points of the point cloud.
- vi) Sum the two histograms.

Ensemble of Shape Functions (ESF) The Ensemble of Shape Functions [9] is a method that calculates a shape global descriptor without requiring normal estimations, but just using three distinct shape functions describing distance, angle and area distributions on the surface of the point clouds. In particular, we consider the following functions:

D2: measures the distances between two randomly chosen points of the surface;

D3: measures the square root of the area of the triangle between three random points on the surface;

A3: measures the angle between three random points on the surface of the 3D model.

For each shape function we consider three sub-histograms: IN, OUT, MIXED that will be filled as explained below.

- i) for $i = 1$ to N (in [9] $N = 40.000$) do
- ii) select three random points p_{ri}, p_{rj}, p_{rk}

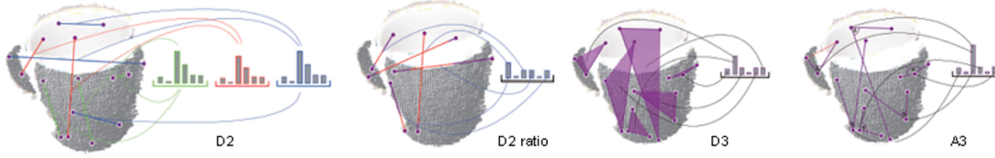


Figure 9: Illustration of ESF histogram creation.

- iii) calculate the distance between p_{ri} and p_{rj} (D2), check if the line connecting the two points lies entirely on the surface (IN), outside the surface (OUT) or both (MIXED); increment the corresponding D2's sub-histograms (either IN, OUT or MIXED) at the previously calculated distance bin. To further exploit the information captured by the lines classified as MIXED, the ratios of these lines are used to construct another distinct shape histogram (RATIO)
 - iv) calculate D3 considering the area of the triangle between p_{ri} , p_{rj} and p_{rk} , then classify the area into IN, OUT and MIXED, considering for all the three lines, the classification used in the previous step, increment the corresponding histogram bins of the D3 sub-histograms.
 - v) calculate A3, and classify this function into IN, OUT and MIXED, considering the position of the line opposite to the angle measured by A3; increment the corresponding A3 sub-histogram bin.
1. At the end of the loop we end up with a global descriptor containing 10 sub-histograms (64 bins each): D2 (IN, OUT, MIXED, RATIO), D3 (IN, OUT, MIXED), A3 (IN, OUT, MIXED).

In Figure 9 an example of ESF histogram calculation is shown.

Signature of Histogram of Orientation (SHOT) SHOT [10] is a 3D local descriptor calculated on each keypoint, that encodes histograms of basic first-order differential entities (i.e. the normals of the points within the support). Furthermore, it codifies also geometric information concerning the location of the points.

For each keypoint, a 3D grid is defined and the histogram is built by using the function of the angle θ_i between the normal at each point within the corresponding part of the grid, $\mathbf{n}_{vi}$, and the normal at the keypoint, $\mathbf{n}_u$. The function is $\cos \theta_i$ and the grid superimposed on each keypoint is an isotropic spherical grid as shown in Figure 10 that encompasses partitions along the radial, azimuth and elevation axes.

Radius-based Surface Descriptor (RSD) The RSD descriptor [11] estimates the minimum and maximum curvature radii from angle/distance pairs ($r_{\min}$ and $r_{\max}$) respectively. In particular, given a point p_i RDS calculate the radius of the fitting curves to its local neighborhood.

The problem of finding $r_{\min}$ and $r_{\max}$ is resolved by assuming that, when $\alpha \in [0, \pi/2]$ the relation between the distance d of two points and the angle α between the points' normal shown in equation (7) can be simplified as in equation (8) by the Taylor decomposition ($O(\alpha^5)$ indicates

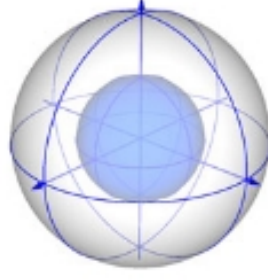


Figure 10: Structure of the SHOT grid.

the existence of elements with order 5 and upwards).

$$d_{(\alpha)} = \sqrt{2r} \sqrt{1 - \cos(\alpha)} \quad (7)$$

$$d_{\alpha} = r \cdot \alpha + \frac{r \cdot \alpha^3}{24} + O(\alpha^5) \quad (8)$$

Similar cases extraction

When a new case is presented to the system, after the feature extraction step, we need to find within the case base those cases that are the closest to the current case. This step needs criteria to decide which case is the most similar (e.g., to determine how close the current and stored cases are), and a mechanism to control how the case base is searched.

As an example, we can use a validation retrieval: first we exploit the main features of the query case for retrieving all cases that appear to be relevant to the problem. Then, we use more discriminative features on the group of retrieved cases to determine whether they are valid for the current problem. In the first experiments, we use as similarity measure on the 3D descriptors the Chi Square Distance [12].

In order to make some preliminary experiments on how these descriptors work, we build a small database containing a small number of objects in different poses. To this aim we employed a Virtual Scanner, that is, a software that virtually acquires point clouds of an object from its mesh model by simulating an acquisition from a sensor like a kinect placed in a specified position w.r.t. the object. The different positions, where the sensor could be placed around the object, are simulated by three parameters, namely a radius, a latitude and longitude angle.

To create the point cloud from a particular point of view, the program loads the mesh model of the object and creates a grid of points, perpendicular to the viewpoint direction and behind the object. In this way the object results to be in the middle between the grid and the viewpoint (see Figure 11).

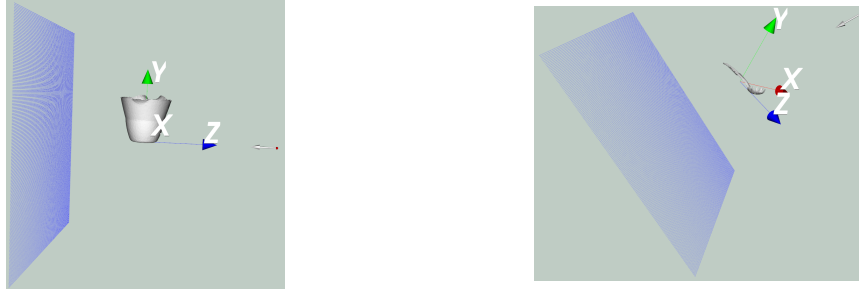


Figure 11: Example of grid positioning

After the grid is created, the Virtual Scanner projects virtual lines from the virtual position of the camera to every point in the grid and finds the intersections between these lines and the object mesh as shown in Figure 12. Every intersection point found along each line is considered as part of the object point cloud resulting from this particular point of view.

Figures 13.a and 13.b depict the mesh of the object and the corresponding point cloud calculated by the virtual scanner, respectively.

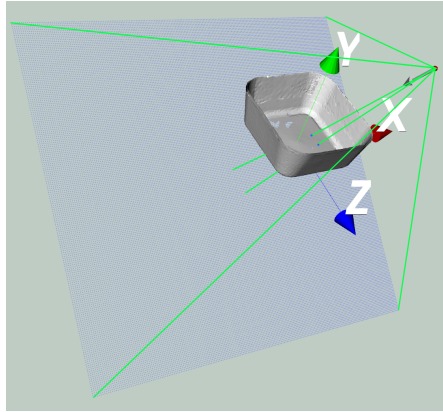


Figure 12: Example of intersections that generate the virtual point cloud.

As shown in Figure 14, the point cloud generated by the virtual scanner is not very similar to the one generated by the Kinect sensor. Indeed, the latter looks like sliced by planes because of the sensor depth resolution that can vary from 2.5 mm to 50 mm, depending on the distance from the camera. To reproduce this behavior also in the synthetic point clouds generated by the virtual scanner we apply a post-processing step. We superimpose a set of parallel planes, with a distance of 2.5mm between two consecutive planes, and normal to the viewpoint direction (see Figure 16). All the points that do not belong to any of these planes are projected on the nearest plane as shown in Figure 17. In this way we are able to “slice” the virtual point cloud. In Figure 15, a virtual point cloud is shown before and after applying this procedure.

The virtual scanner allows us to easily generate point clouds of different objects from different

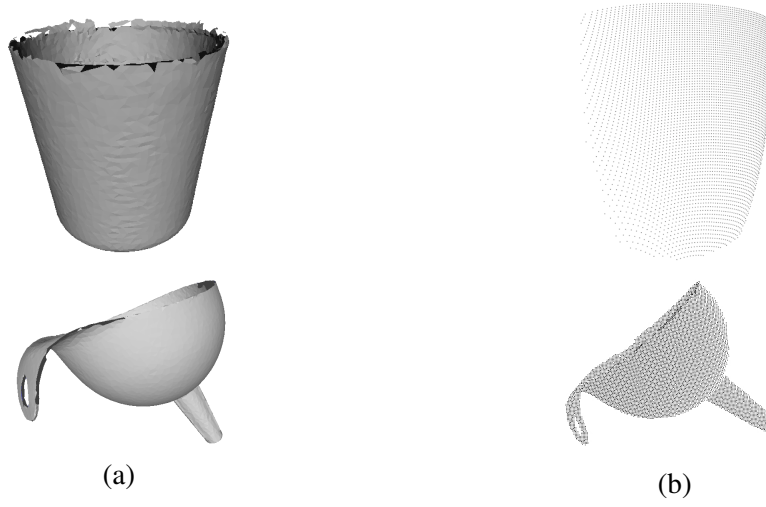


Figure 13: (a) Mesh of the object, (b) point cloud calculated by the Virtual Scanner.

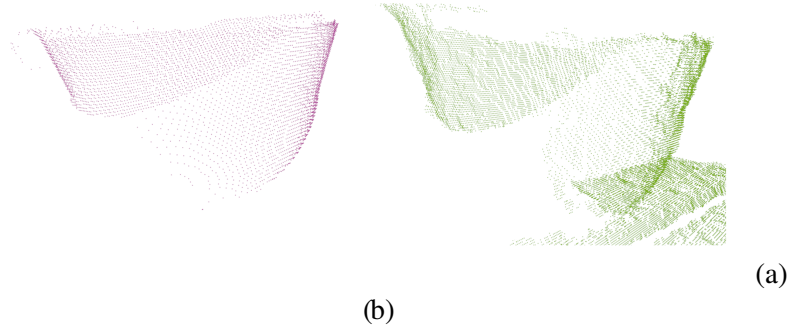


Figure 14: Difference between a virtual point cloud (a) and a real point cloud (b).

points of view. In this way we can test also how much the 3D descriptors introduced in the above subsection change as a function of the viewpoint change.

In the preliminary experiments we used three 3D local descriptors, namely SHOT, FPFH, and RSD. We recall that these local descriptors are evaluated on each keypoint extracted from the point cloud. For this reason, in order to evaluate how much are two point clouds similar, we have compare the two set of keypoints extracted from the two objects.

To this aim we apply the following procedure. Let K_s and K_b be the two keypoint sets containing N_s and N_b keypoints, respectively, with $N_s \leq N_b$. We define the dissimilarity $\overline{D}$ between K_s and K_b the ratio:

$$\overline{D} = \frac{\sum_{i=1}^{N_s} \min_{j \in [1 \dots N_b]} \|k_i, k_j\|}{N_s} \quad (9)$$

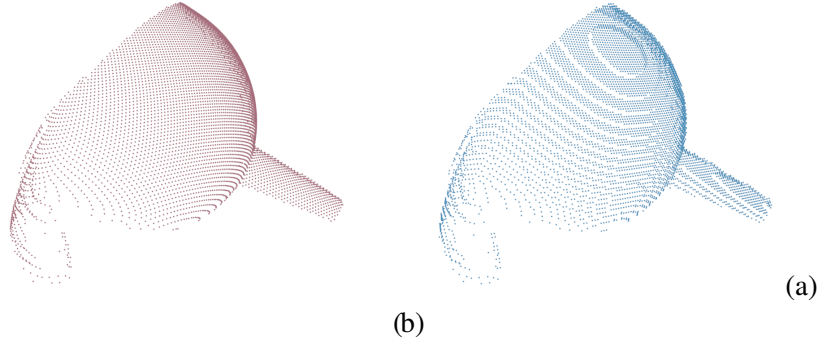


Figure 15: A virtual point cloud before (a) and after (b) applying the slicing procedure.

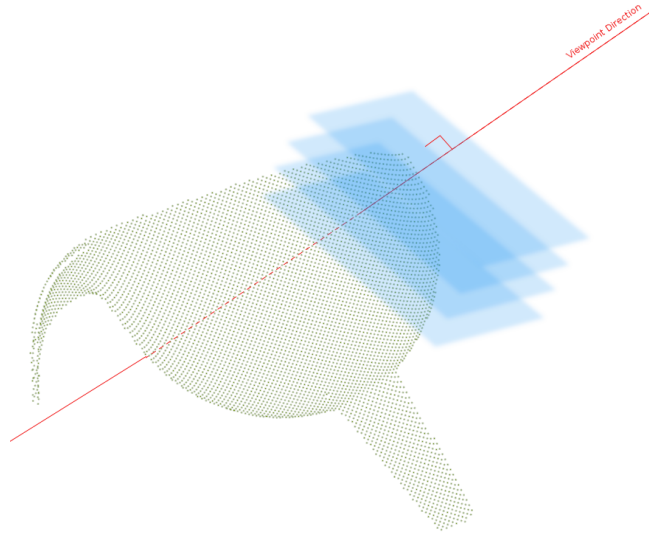


Figure 16: Parallel plane superimposed on the object.

where $\min \|k_i, k_j\|$ represents the minimum distance between a keypoint k_i of K_s and all the keypoints of K_b .

In the preliminary experiment we calculate the 3D shape descriptor on the virtual point cloud of the following objects:

- container;
- jug;
- kettle;
- mug;
- pan;

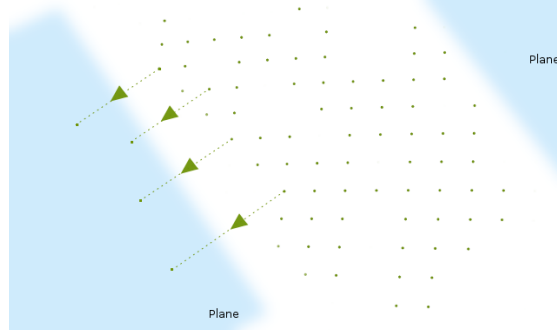


Figure 17: Projection of virtual points on the nearest plane.

- spoon.

In Table 2 we show the value of $\overline{D}$ calculated considering a reference virtual point cloud of an object of the database PO and the virtual point cloud of the same object rotated along its latitude and longitude PO_r . In that Table, for each object (represented in the first column), we show in separate rows the value of $\overline{D}$ between PO and PO_r rotated by **lat** and **lon** in latitude and longitude, respectively. The column **size** represents the number of points of PO_r , and t is the execution time in ms for the 3D descriptor calculator.

On the other hand, in Table 3 we show the value of $\overline{D}$ calculated between two different objects Ob_1 and Ob_2 .

Once the value of $\overline{D}$ has been computed for two sets of keypoints, we have to define a threshold τ so as, if $\overline{D} < \tau$, the two sets can be considered belonging to the same object. On the other hand, as shown in Tables 2, and 3, the magnitude of the $\overline{D}$ values changes depending on the algorithm used to calculate the 3D descriptors. Thus, different values of τ have to be chosen for different algorithms.

However, from tables 2, and 3 we observe that the value of $\overline{D}$ allows us to distinguish different objects (e.g., a container from a jug) and a different pose of the same object.

Table 2: Value of S between the same object in different poses

Container			SHOT		FPFH		FPFH	
Lat	Lon	Size	ms	S	ms	S	ms	S
-5	0	13373	38	0.0826	2	34.9997	0	7.09E-06
-5	-10	13402	56	0.1054	4	117.9347	1	1.754E-05
-5	10	13381	50	0.1022	2	46.4375	0	1.901E-05
-5	-5	13370	52	0.0979	3	105.744	1	6.64E-06
-5	5	13402	41	0.0862	2	40.727	0	1.712E-05
0	-10	13947	44	0.073	2	33.4	0	1.115E-05
0	10	13973	42	0.0774	2	39.333	0	8.25E-06
0	-5	13959	46	0.0638	3	28.0092	1	1.175E-05
0	5	13943	46	0.0666	3	32.239	1	0.000014
5	0	15268	53	0.1094	3	237.5579	1	7.12E-06
5	-10	15268	52	0.1142	2	252.9672	0	1.589E-05
5	10	15307	53	0.1247	2	233.1722	1	7.22E-06
5	-5	15275	59	0.096	3	246.5449	0	1.081E-05
5	5	15268	45	0.0899	2	233.9556	0	6.77E-06
-60	0	8627	51	0.4401	2	748.1966	0	0.0006236
-60	-115	8960	76	0.4795	4	626.8929	1	0.0001076
-60	115	8854	77	0.6552	6	3307.9083	0	1.392E-05
-60	-5	8588	49	0.4456	2	723.2629	0	0.000363
-60	5	8636	58	0.4174	2	637.1517	0	0.0002822
-50	-125	8350	78	0.4424	4	646.8398	1	0.0001631
-50	125	8228	97	0.5461	5	3315.9954	1	1.499E-05
-45	-130	8057	56	0.4324	3	604.8397	0	0.0001456
-45	130	8010	47	0.4154	2	503.9899	0	8.061E-05
-55	-120	8695	84	0.5472	9	3307.5521	1	8.76E-06
-55	120	8608	76	0.6252	6	3306.2933	0	1.398E-05

Table 3: Value of S between different objects

			SHOT		FPFH		FPFH	
Ob_1	Ob_2	Size	ms	S	ms	S	ms	S
container	jug	22677	66	0.2191	4	1180.8042	0	0.0007399
container	kettle	14660	62	0.5398	4	1362.2914	0	0.0014625
container	mug	25145	95	0.3576	13	3915.529	1	0.0013104
container	pan	6495	54	0.4296	3	2482.8016	0	0.0005989
container	spoon	1564	34	0.6229	4	5217.8184	0	0.0009352

3 Optimal case selection and case adaptation

Once a set of similar cases has been retrieved, we choose as optimum case the one that is the most similar to the current case. Then, we determine if this selected case is close enough to the problem case or whether the search parameters need to be modified and the search performed again.

If the case is appropriate to be adapted, it undergoes the phase of adaptation, where the retrieved solution is translated into the solution appropriate for the current problem. This step is the most important step of CBR and adds intelligence to what would otherwise be a simple pattern matching.

After an appropriate solution is adapted, the CBR systems usually enter in a learning phase where the solution is tested in the reality and the outcome is classified as success or failure. This assessment can be used to update the CBR system to take into account any new information uncovered in the processing of the new solution.

Regarding these steps we have not performed a deep study yet. Indeed, in order to decide how to adapt a retrieved case, we need to evaluate how much this case is far from the problem case, and therefore, we need the database of grasps to start the study of this part.

References

- [1] J. L. Kolodner, “An introduction to case-based reasoning,” *Artificial Intelligence Review*, vol. 6, no. 1, pp. 3–34, 1992.
- [2] R. Rusu and S. Cousins, “3d is here: Point cloud library (pcl),” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1–4.
- [3] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [4] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, “Towards 3d point cloud based object maps for household environments,” *Robot. Auton. Syst.*, vol. 56, no. 11, pp. 927–941, 2008.
- [5] J. Berkmann and T. Caelli, “Computation of surface geometry and segmentation using covariance techniques,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 11, pp. 1114–1116, 1994.
- [6] N. B. Radu Bogdan Rusu, Zoltan Csaba Marton and M. Beetz, “Learning informative point classes for the acquisition of object model maps,” in *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Hanoi, Vietnam, 2008.

- [7] M. B. Radu Bogdan Rusu, Nico Blodow, “Fast point feature histograms (fpfh) for 3d registration,” *Intelligent Autonomous System*, 2009, <http://files.rbrusu.com/publications/Rusu09ICRA.pdf>.
- [8] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, “Fast 3d recognition and pose using the viewpoint feature histogram,” in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 10/2010 2010.
- [9] M. V. W. Wohlking, “Ensemble of shape functions for 3d object classification,” in *Robotics and Biomimetics (ROBIO), IEEE International Conference on*, 2011.
- [10] F. Tombari, S. Salti, and L. Di Stefano, “Unique signatures of histograms for local surface description,” in *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*, ser. ECCV’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 356–369.
- [11] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinhellefort, and M. Beetz, “General 3D Modelling of Novel Objects from a Single View,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 18-22 2010.
- [12] K. Vehkalahti, “The concise encyclopedia of statistics by yadolah dodge,” *International Statistical Review*, vol. 76, no. 3, pp. 460–461, 2008.

Learning Dexterous Grasps That Generalise To Novel Objects By Combining Hand And Contact Models

Marek Kopicki, Renaud Detry, Florian Schmidt, Christoph Borst, Rustam Stolkin, and Jeremy L. Wyatt

Abstract—Generalising dexterous grasps to novel objects is an open problem. We show how to learn grasps for high DoF hands that generalise to novel objects, given as little as one demonstrated grasp. During grasp learning two types of probability density are learned that model the demonstrated grasp. The first density type (the contact model) models the relationship of an individual finger part to local surface features at its contact point. The second density type (the hand configuration model) models the whole hand configuration during the approach to grasp. When presented with a new object, many candidate grasps are generated, and a kinematically feasible grasp is selected that maximises the product of these densities. We demonstrate 31 successful grasps on novel objects (an 86% success rate), transferred from 16 training grasps. The method enables: transfer of dexterous grasps within object categories; across object categories; to and from objects where there is no complete model of the object available; and using two different dexterous hands.

I. INTRODUCTION

Transferring dexterous grasps to novel objects is an open problem. We present a method that infers grasps for novel objects from as little as one training example. It generalises a human chosen *grasp type* (e.g. rim, pinch) trained from one or two examples, to a novel object of potentially quite different shape. We demonstrate learning and transfer of six grasps types to 31 different grasps of 18 objects, and for two different dexterous hands. The results show the ability to generalise a grasp within and across object categories, and with full or partial shape information. The method requires no knowledge of the human defined object category (e.g. cup, box) when learning or performing transfer.

There has been progress in learning generalisable grasps. One class of approaches utilises the shape of common object parts to generalise grasps across object categories [8], [11], [15]. This works well for low DoF hands. Another class of approaches captures the global properties of the hand shape either at the point of grasping, or during the approach [2]. This global hand shape can additionally be associated with global object shape, allowing generalisation by warping grasps to match warps of global object shape [12]. This second class works well for high DoF hands, but



Fig. 1. Objects used: the four objects on the left were used solely as example grasps, the ten objects on the right were solely used as novel test objects. The four at the back were used as either training or test, but not for both at the same time, e.g. we transferred a pinch grasp from the coke bottle to the vitamin tube, and a pinch support grasp vice versa.

generalisation is more limited. We achieve the advantages of both classes, generalising grasps across object categories with high DoF hands.

Our main technical innovation to achieve this is to learn two types of models from the example grasp, and then recombine them when inferring a new grasp. Both models are probability density functions. The first is a *contact model* of the relation between a finger part and the local object shape at its contact. We learn one such model for each rigid link in the hand. Each model can generate many possible independent placements for its link on a new “query” object. Given these models the next problem is to find a hand configuration that combines a good subset of placements in a kinematically feasible grasp. To address this we learn a second type of model, a *hand configuration model* from the example grasp. We then use this hand configuration model to constrain the combined search space for the link placements.

Given these two learned models we can infer grasps for novel objects. When presented with a novel query object we use a Monte Carlo procedure to construct a third type of density function (called the *query density*) over possible contact points on the new object. We again construct a query density for every link of the hand. We then use these densities in two ways to select the new grasp. First we pick a link, and draw a contact point for it on the object from the query density. Then we sample a hand configuration to obtain the complete set of link poses. The whole solution is then refined using simulated annealing on the product of all the query densities and the hand configuration density. The paper starts with a survey of related work (Sec II); continues with the representations employed and the learning process (Sec III); followed by a description of the process for finding a grasp

Marek Kopicki, Rustam Stolkin and Jeremy L. Wyatt are with the University of Birmingham, England. Email: {msk, stolkinr, jlw}@cs.bham.ac.uk. Renaud Detry is with the EECS department, University of Liege, Belgium. Email: renaud.detry@ulg.ac.be. Florian Schmidt and Christoph Borst are with the German Aerospace Center (DLR), Wessling, Germany, Email: {florian.schmidt, christoph.borst}@dlr.de. The work described here is protected by Patent Application (UKIPO 1309156.6). We gratefully acknowledge support by the FP7-IST-600918 (PaCMan), FP7-IST-248273 (GeRT), the Swedish Research Council (VR), the Belgian National Fund for Scientific Research (FNRS).

for a novel object. We finish with an experimental study (Sec VI) and a discussion.

II. RELATED WORK

In robotics, mainstream grasp planning has traditionally relied on force analysis [3]. Unfortunately, the strengths of force analysis become mitigated in scenarios where the object models that the robot can recover (from memory or sensing) are incomplete or erroneous. Some authors have explored ways of directly linking perceptions to action parameters. Symmetry principles have been used to reconstruct the occluded side of unknown objects [23], and thereby allow the definition of grasp contact points on occluded surfaces [4]. Others have parameterised grasps by looking for shapes that are likely to fit into the robot’s gripper [9], [17]. Popovic et al. [17] computed grasps applied to object edges detected in 2D images. Another class of methods search 3D range data for shapes closely matching the geometry of the gripper [9].

Instead of hard-coding the function that computes grasp parameters from vision, a growing number of researchers have focused on methods that learn the perception-action mapping from experimental data [7], [16]. A subclass of these focused on learning a mapping from incomplete object views to grasp parameters [1], [8], [11], [15], [14], [19], while another subclass has aimed at transferring grasps across objects whose complete 3D shape is known [2], [12]. When addressing the problem of grasping a partially-perceived object, authors have developed means of learning how to place the wrist of the gripper with respect to the part of the object that is perceived by the robot’s vision system [8], [11], [15], [14]. When working with two-finger grippers, the wrist completely parametrises the gripper pose [11]. In order to plan grasps for a three-finger hand, Kroemer et al. [15] and Detry et al. [8] let the robot learn a hand preshape, and relied on compliance or force sensing to close the hand. Alternately, Kroemer et al. [14] also relied on control policies to adapt the fingers to the visual input. Saxena et al. [19] learned a three-finger grasp success classifier from a bank of photometric and volumetric object features.

Hillenbrand et al. [12] addressed the problem of transferring a multi-finger grasp between two objects of known 3D shape. A known object’s geometry is warped until it matches that of a novel object, thereby also warping grasp points on the surface of the known object onto candidate grasp points on the novel object. Ben Amor et al. [2] exploit this warping method to transfer grasps taught by a human hand (using a data glove) to contact points for a robot hand on a novel object. Our work is, in spirit, most similar to the work of Ben Amor et al. [2] and Hillenbrand et al. [12], and to the work of Saxena et al. [19]. We compute a full hand grasping configuration for a novel object, using a grasp model that is learned from a single or a few example grasps. Our method performs best when a full shape model of the target object is available, but it is also applicable to partially-modelled objects. One difference with [12] is that they transfer grasps within the same object shape category, whereas we are able to transfer grasps to different object categories.

III. REPRESENTATIONS

This paper addresses the problem of transferring grasps across objects. Given one or more grasp examples, we aim to capture the configuration of the hand and its fingers with respect to neighbouring object surfaces, and to adapt the hand and fingers to the shape of a novel object.

We assume that the robot’s hand is formed of a palm, and a number of finger phalanges. We refer to the palm and phalanges as *parts* of the hand, denoted by H_i ($i = 1, \dots, N$). Our approach relies on a pair of models that capture two complementary aspects of the grasping process. The first model, referred to as the *contact model*, captures the configuration of one hand part H_i relative to local features from the object’s surface. Given the 3D model of an object that the robot has not grasped yet, contact model M_i allows the robot to compute a set of poses for the hand part H_i . When a robot is equipped with a N -part hand, up to N contact models can be instantiated.

The second model, referred to as the *hand configuration model*, captures a set of whole hand shapes close to the whole hand shape in the example grasp. The role of the hand configuration model in inferring a grasp for a novel object will be to rapidly constrain the search space for the hand part positions. The hand configuration model is learned from the same grasp data as the contact models.

A. Kernel Density Estimation

Much of our work relies on the joint probabilistic modeling of $SE(3)$ poses and surface descriptors. As a result, this paper makes extensive use of probability density functions (PDFs) defined on $\mathbb{R}^3 \times SO(3) \times \mathbb{R}^2$, where $SO(3)$ is the group of 3D rotations, and surface descriptors are parametrized by two real numbers. This section explains how we define these density functions.

We represent PDFs non-parametrically with a set of pose-descriptor pairs (or particles) x_ℓ

$$\{x_\ell : x_\ell \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^2\}_{\ell \in [1, L]}. \quad (1)$$

The probability density in a region of space is determined by the local density of the particles in that region. The underlying PDF is created through *kernel density estimation* [20], by assigning a kernel function $\mathcal{K}$ to each particle supporting the density, as

$$\text{pdf}(x) \simeq \sum_{\ell=1}^L w_\ell \mathcal{K}(x|\mu_\ell, \sigma_\ell), \quad (2)$$

where μ_ℓ is the kernel mean point, σ_ℓ is the kernel bandwidth and w_ℓ is a weight associated to x_ℓ such that $\sum_\ell w_\ell = 1$. We use a kernel that factorises into three functions defined on the three components of our domain, namely $\mathbb{R}^3$, $SO(3)$, and $\mathbb{R}^2$. Denoting the separation of x into a position, orientation, and feature by (p, q, r) , we define our kernel with

$$\mathcal{K}(x|\mu, \sigma) = \mathcal{N}_3(p|\mu_p, \sigma_p) \Theta(q|\mu_q, \sigma_q) \mathcal{N}_2(r|\mu_r, \sigma_r) \quad (3)$$

with kernel mean point $\mu = (\mu_p, \mu_q, \mu_r)$ and kernel bandwidth $\sigma = (\sigma_p, \sigma_q, \sigma_r)$, and where $\mathcal{N}_k$ is k -variate isotropic

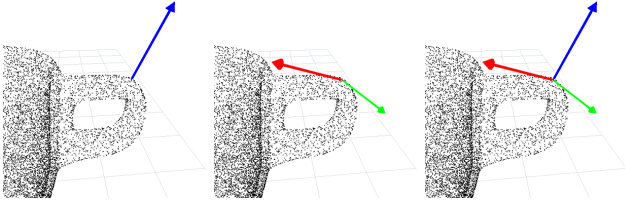


Fig. 2. An example object point cloud (black dots) with a selected point, its surface normal (blue axis), 1-st highest principal curvature (red axis), second lowest principal curvature (green axis), composing a complete frame shown in the right panel.

Gaussian kernel, Θ corresponds to a pair of antipodal von-Mises Fisher distributions which forms a Gaussian-like distribution on $SO(3)$ (for details see [10], [22]). The value of Θ is given by

$$\Theta(q|\mu_q, \sigma_q) = C_4(\sigma_q) \frac{e^{\sigma_q \mu_q^T q} + e^{-\sigma_q \mu_q^T q}}{2} \quad (4)$$

where $C_4(\sigma_q)$ is a normalising constant.

B. Surface Features

All objects considered in the paper are represented by point clouds. Each point x is characterised by a *feature*, comprising a 3D position p , a 3D orientation q , and a surface descriptor r :

$$x = (v, r), \quad v = (p, q) \quad (5)$$

The surface descriptor corresponds to the local *principal curvatures* [21]. Curvatures at point p are recorded along two directions that both lie in the plane tangential to the object's surface, i.e. perpendicular to the surface normal at p . The first direction, k_1 , is the direction of highest curvature. The second direction, k_2 , is perpendicular to k_1 . The curvatures along k_1 and k_2 are denoted by r_1 and r_2 respectively, forming a 2-dimensional feature vector $r = (r_1, r_2)$. The surface normals and curvatures are computed using the PCL library [18].

The surface normals and curvatures allow us to define a reference frame v at each object point p , with the exception of $r_1 \approx 0$, when the principal directions are chosen randomly, as it is in the case of planar surfaces. Fig. 2 illustrates surface normals and curvatures.

The procedure described above allows the computation of a set of features (pose-descriptor pairs) $\{(v_\ell, r_\ell)\}$ from a given object point cloud. In turn, the set of features defines a joint probability distribution, further referred to as the *object model*:

$$O(v, r) \equiv \mathbf{pdf}^O(v, r) \simeq \sum_{\ell=1}^L w_\ell \mathcal{K}(v, r | \mu_{x_\ell}, \sigma_x) \quad (6)$$

where O is short for $\mathbf{pdf}^O$, $x_\ell = (v_\ell, r_\ell)$, kernels $\mathcal{K}$ are defined in Eq. (3), and all weights are equal $w_\ell = 1/L$.

C. Contact Model

A contact model M_i encodes the joint probability distribution of surface features and the 3D pose of the i^{th} hand part H_i . Let us consider the hand grasping some given object. The (object) contact model of hand part H_i is denoted by

$$M_i(U, R) \equiv \mathbf{pdf}_i^M(U, R) \quad (7)$$

where M_i is short for $\mathbf{pdf}_i^M$, R is the random variable modeling surface features, and U models the pose of H_i relative to a surface feature. In other words, denoting realisations of R and U by r and u , $M_i(u, r)$ is proportional to the probability of finding H_i at pose u relative to the frame of a nearby object surface feature that exhibits feature vector equal to r .

Given a set of surface features $\{(v_\ell, r_\ell)\}$, a contact model M_i is constructed by selecting a set of features from the neighbourhood of the hand part H_i (Fig. 3). We proceed with a soft selection which associates, to each surface feature $x_\ell = ((p_\ell, q_\ell), r_\ell)$, a weight that exponentially decreases with a squared distance to H_i (Fig. 4). The weight is given by

$$w_{i\ell} = \exp(-\lambda \|p_\ell - a_{i\ell}\|^2), \quad (8)$$

where $\lambda \in \mathbb{R}^+$ and $a_{i\ell}$ is a point on the surface of H_i , the closest to p_ℓ .

The contact model is approximated as

$$M_i(u, r) \simeq \sum_{\ell} w_{i\ell} \mathcal{N}_3(p | \mu_{p_{i\ell}}, \sigma_{p_i}) \Theta(q | \mu_{q_{i\ell}}, \sigma_{q_i}) \mathcal{N}_2(r | \mu_{r_\ell}, \sigma_r) \quad (9)$$

where all weights are normalised $\sum_{\ell} w_{i\ell} = 1$, $u = (p, q)$, and where $(\mu_{p_{i\ell}}, \mu_{q_{i\ell}}) = u_{i\ell}$ is the centre of a kernel computed from ℓ -th feature pose v_ℓ and the i -th hand part pose s_i from their mutual geometric relation (see Fig. 3)

$$s_i = v_\ell \circ u_{i\ell} \quad (10)$$

or equivalently $u_{i\ell} = v_\ell^{-1} \circ s_i$, multiplying both sides of (10) by v_ℓ^{-1} . $u_{i\ell}$ represents frame s_i of hand part H_i relative to feature frame v_ℓ and *independently* on the world frame encoded by s_i itself.

Sum 9 involves only terms for which $x_\ell = ((p_\ell, q_\ell), r_\ell)$ belong to the neighbourhood of H_i , $\{x_\ell : \|p_\ell - a_{i\ell}\| \leq \delta, \delta \in \mathbb{R}^+\}$. If the neighbourhood of a particular hand part H_i is empty, the corresponding contact model is not instantiated and it is excluded from any further computation.

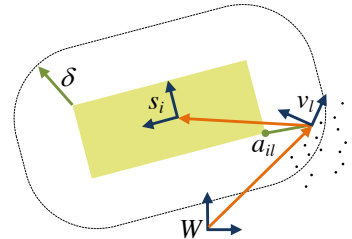


Fig. 3. Contact model: i -th hand part (solid rectangle) with frame s_i , its neighbourhood (rounded dashed contour at distance δ from the hand part), point cloud (dots), world frame W , ℓ -th feature with frame v_ℓ and the corresponding closest point on the part surface $a_{i\ell}$.

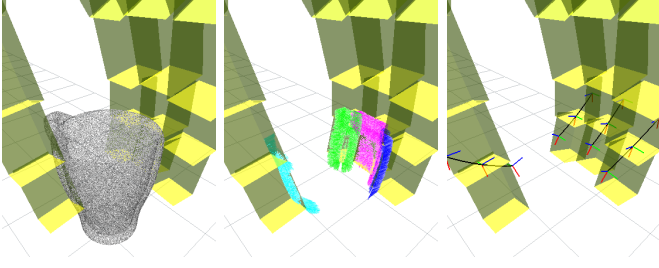


Fig. 4. Example top grasp of a mug represented by a point cloud (left). The colorful regions at contacts between fingers and the object are rays between features and the closest hand part surfaces (middle). The black curves with frames at the fingertips represent the range of hand configurations in Eq. (12) (right).

λ in (8) and the bandwidths σ_{p_i} , σ_{q_i} , σ_r in (9) were hand tuned and kept fixed in all the experiments. The time complexity for learning each contact model from an example grasp is $\Omega(|r||O||Tr|)$ where $|Tr|$ is the number of triangles in the tri-mesh describing the hand part, $|O|$ is the number of points in the object model.

D. Hand Configuration Model

The hand configuration model, denoted by C , encodes a set of configurations of the hand joints $h \in \mathbb{R}^D$ (i.e. joint angles), that are particular to example grasps. The purpose of this model is to allow us to restrict the grasp search space (during grasp transfer) to hand configurations that resemble those observed while training the grasp.

In order to boost the generalisation capability of the grasping algorithm the hand configuration model encodes the hand configuration that was observed when grasping the training object, but also a set of configurations recorded during the approach towards the object. Let us denote by h^t the joint angles at some small distance *before* the hand reached the training object, and by h^g the hand joint angles at the time when the hand made contact with the training object. We consider a set of configurations interpolated between h^t and h^g , and extrapolated beyond h^g , as

$$h(\gamma) = (1 - \gamma)h^g + \gamma h^t \quad (11)$$

where $\gamma \in \mathbb{R}$. For all $\gamma < 0$, configurations $h(\gamma)$ are beyond h^g (see Fig. 4). The hand configuration model C is constructed by applying kernel density estimation to $\{h(\gamma) : \gamma \in [-\beta, \beta], \beta \in \mathbb{R}^+\}$:

$$C(h) \equiv \sum_{\gamma \in [-\beta, \beta]} w(h(\gamma)) \mathcal{N}_D(h|h(\gamma), \sigma_h) \quad (12)$$

where $w(h(\gamma)) = \exp(-\alpha \|h(\gamma) - h^g\|^2)$ and $\alpha \in \mathbb{R}^+$. α and β were hand tuned and kept fixed in all the experiments. The hand configuration model computation has time complexity $\Omega(|\Gamma|)$ where Γ is the size of the set of values of γ used in Eq. 12.

IV. INFERRING GRASPS FOR NOVEL OBJECTS

After acquiring the contact model and the hand configuration model the robot is now presented with a new query

object to grasp. The aim is that the robot finds a grasp such that its parts H_i are well-placed with respect to the object surface, while preserving similarity to the example grasps.

First of all we combine the contact models with the observed point cloud for the object to obtain a set of *query densities*, one for each hand part. The i^{th} query density Q_i is a density over where the i -th hand part H_i can be on the surface of new object (see Fig. 5). From the query densities, a hand pose is generated as follows. We randomly pick a hand part j . We randomly sample, from the corresponding query density, a pose for part j . We sample poses for all other hand parts, from the hand configuration model aligned to match the pose of part i , yielding a kinematically feasible hand configuration that is compatible with the pose selected for part i . We refine the grasp by performing a simulated annealing search in the hand configuration space, to locally maximise the grasp likelihood measured as the product of the hand configuration density and the query densities for all the hand parts, thereby ensuring adjustments of the remaining hand parts to make contacts with surface patches of similar local curvature to their contact models. We repeat the entire process a fixed number of times, and select the most likely kinematically feasible grasp.

The optimisation procedure generates several possible grasps, each with its likelihood. Each grasp has a set of hand part poses that independently comply with the contact models, while jointly complying to the hand configuration model. The following subsections explain in detail how to create approximate query densities for a given query object, and how grasp optimisation is carried out using simulated annealing.

A. Query Density

This section explains how query densities are constructed. A query density emerges from the combination of a contact model with an object point cloud O . The i^{th} query density Q_i models where the i -th hand part H_i can be placed to grasp the object, given the features observed on the object's surface. Specifically, Q_i models the pose distribution of H_i in the world frame.

The contact model M_i models viable poses for H_i relatively to surface features. Intuitively, a contact model is constructed by finding pairs of matches between the features that belong to M_i and the features of O , and representing the distribution of poses for H_i that are suggested by all matching pairs of features.

The above procedure can be described in probabilistic terms as an integration over $(v, r) \in O$ and $(u, r) \in M_i$, yielding query density Q_i :

$$Q_i(s) \equiv Z_i \int P_i(s|v, u) O(v, r) M_i(u, r) dv du dr \quad (13)$$

where Z_i is a normalisation constant, while O and M_i satisfy

$$\int O(v, r) dv dr = 1, \quad \int M_i(u, r) du dr = 1 \quad (14)$$

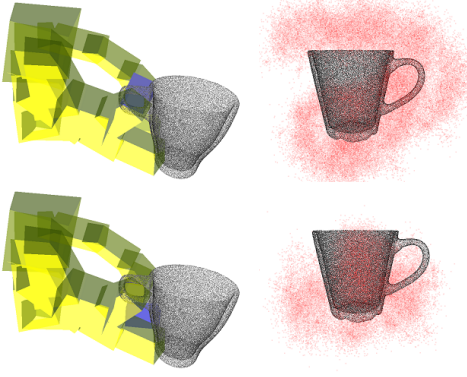


Fig. 5. Visualisation of two query densities (right) for two contact models (left) of a handle grasp. Each contact model is created for a single hand part (blue) for an example mug (left). A query density is a distribution over poses of the corresponding hand part (red cloud) for a new “query” mug.

Furthermore, the conditional density P_i represents pose multiplication $s = v \circ u$, O and M_i contribute to the weight of pose s through the common feature variable r .

Using Bayes rule the integral (13) can be rewritten as follows:

$$Q_i(s) = Z_i \int P_i(s|v, u) O(r|v) O(v) M_i(u|r) M_i(r) dv du dr \quad (15)$$

If the involved objects are represented simply as a collection of surface features (5) without uncertainties, $O(r|v)$ becomes a mapping $r(v) : v \rightarrow r$ which for any choice of v uniquely determines r , effectively removing integration over r . Eq. (15) can be approximated as

$$Q_i(s) \simeq \int P_i(s|v, u) O(v) M_i(u|r(v)) M_i(r(v)) dv du \quad (16)$$

Eq. (16) can be computed directly via Monte Carlo integration [5], [6], which generates a number of weighted kernels, by repeating the following steps:

- 1) Sample $(\hat{v}_\ell, \hat{r}_\ell) \sim O$, i.e. just pick up random surface feature from the point cloud
- 2) Sample from conditional density $(\hat{u}_{i\ell}) \sim M_i(u|\hat{r})$
- 3) Compute kernel centre $\hat{s}_{i\ell} = \hat{v}_\ell \circ \hat{u}_{i\ell}$
- 4) Compute kernel weight $w_{i\ell} = M_i(\hat{r}_\ell)$

Query density (13) can be finally approximated as

$$Q_i(s) \simeq \sum_{\ell} w_{i\ell} \mathcal{N}_3(p|\mu_{\hat{p}_{i\ell}}, \sigma_{p_i}) \Theta(q|\mu_{\hat{q}_{i\ell}}, \sigma_{q_i}) \quad (17)$$

with ℓ -th kernel centre $(\mu_{\hat{p}_{i\ell}}, \mu_{\hat{q}_{i\ell}}) = \hat{s}_{i\ell}$, and where all weights were normalised $\sum_{\ell} w_{i\ell} = 1$.

The bandwidths σ_{p_i} and σ_{q_i} in (17) were hand tuned and kept fixed in all the experiments.

Fig. 5 visualises two example query densities created for two contact models of a handle grasp (see also Fig. 10).

B. Grasp Optimisation

We now describe how to compute the most desirable grasp configuration for a new object. Let us denote by $\mathbf{s} = (s_1, \dots, s_N)$ the configuration of the hand in terms of a set of hand part poses $s_i \in SE(3)$. Let us also denote

by $\mathbf{h} = (h_w, h_j)$ the hand pose in term of a wrist pose $h_w \in SE(3)$ and joint configuration $h_j \in \mathbb{R}^D$. Finally, let $k^{\text{for}}(\cdot)$ denote the forward kinematic function of the hand, with

$$\mathbf{s} = k^{\text{for}}(\mathbf{h}), \quad s_i = k_i^{\text{for}}(\mathbf{h}) \quad (18)$$

We infer a desirable set of grasp parameters by maximising a product

$$h = \underset{(h_w, h_j)}{\operatorname{argmax}} C(h_j) \prod_i Q_i(k_i^{\text{for}}(h_w, h_j)). \quad (19)$$

where $C(h_j)$ is the hand configuration model (12) and Q_i are query densities (17).

In other words, we search for the hand pose $\mathbf{h}$ which maximises the pose probability of hand parts H_i given surface features observed on the object, while limiting the search to hand configurations that resemble those from training.

Product 19 is maximised using a simulated annealing procedure which is initialised as follows:

- 1) pick up random hand part i
- 2) find the corresponding part pose s_i by sampling from Q_i
- 3) pick up random hand configuration h_j by sampling from $C(h_j)$
- 4) compute remaining parts poses s and hand pose $\mathbf{h}$ using forward kinematics
- 5) evaluate Eq. 19, retain best scoring $\mathbf{h}$.

In the next iterations, the simulated annealing procedure skips steps 1) and 2) directly generating wrist pose h_w in some neighbourhood of the previous best scoring one.

The entire procedure is then repeated 1000 times, generating a large number of solutions (grasps) which are then clustered and ranked, yielding a variety of candidate hand poses (“grasp clusters”) which are then checked against arm reachability and workspace constraints. The remaining best scoring hand pose is then used to generate an approach trajectory. The computation of the query distribution has time complexity $\Omega(|r||O||L|)$ where L is the number of kernels in the query density. The query density is calculated once for each new object for each hand part prior to grasp optimisation.

V. EXPERIMENTAL METHOD

We tested our method with 18 real objects and two robot platforms. We performed three experiments. In these we use two types of point cloud models of objects. Type 1 (a full object model) is a precise high density point cloud created by sampling from a CAD model of the object. Type 2 (a partial object model) is a point cloud gathered from two views of the object on the fly, using a depth camera, and is thus more sparse, and more noisy. In Experiment 1 we trained and tested grasps on objects with full models. In Experiment 2 we trained on objects with full models, but transferred to objects with partial models. In Experiment 3 we trained and tested on objects with partial models.

The method adapts a known grasp type to a novel object, and so the training set was organised by grasp type. Six

Grasps on	#Succ.	#Fail	Success rate
Training objects	14	2	88%
Novel objects	31	5	86%

TABLE I

ALL EXPERIMENTS: GRASP SUCCESS RATES.

grasp types were used: rim, pinch, pinch with support, power, handle grasp with support, and a top grasp. A training object was placed in front of the robot, and a partial point cloud constructed from two views. In Experiments 1 and 2 pose estimation for the full object model was then performed with a model fitting procedure [13]. In Experiment 3 only the partial point cloud gathered on the fly was used for training. The training grasp was then taught on the localised object model (Experiment 1,2) or the partial point cloud (Experiment 3), and the contact model, and hand configuration model learned. The training grasp was then executed on the real object. This real execution was not used for the learning, but to test the residual failure rate due to localisation errors. After training, test objects were presented in sequence, a grasp type selected for each by hand. For each test object the grasp adaptation procedure ran 1000 times to generate 1000 candidate grasps, taking a total of 5 secs to compute. These grasps were ranked by the score h (Eq. 19) and working down the list each was tested for kinematic feasibility using collision checking and reachability analysis for the final hand pose. The highest scoring feasible grasp was then selected to be performed. Two robot platforms were used. The first was the DLR Justin mobile robot (Fig. 11), equipped with a 15-DoF DLR II hand with 4 fingers. The second robot platform (Fig. 6) had a 20-DoF DLR-HIT II hand with 5 fingers.

VI. RESULTS

Success rates aggregated over all three experiments are shown in Table I, 88% of the grasps executed on training objects succeeded, while 86% succeeded on novel objects. Success here corresponds to the ability to lift up an object at least 20cm above the table. The training case failures are due to localisation error. Table II shows the training and test sets for Experiment 1. For Experiment 2 we trained two grasps: power and pinch on the bottle and dumpling box respectively, and transferred those successfully the moisturiser and anti-perspirant (pinch), and spray tin (power) (Fig. 12). In Experiment 3 we trained on the stapler (rim) and successfully tested on the bowl and the cutlery holder (Fig. 13). Pictures of results for Experiment 1 are shown in (Figs. 8-10).

VII. CONCLUSIONS AND DISCUSSION

In this paper we have presented a method that generalises a single grasp (contacts and trajectories) of an object to several grasps of other objects with different global shapes. One essential element of our approach is learning a separate *contact model* for each finger phalange how of its pose relative to the surface is related to local surface feature. Another is that we learn a *hand configuration model* based

Platform	Training	Test
DLR-HIT 2	glass (r), bottle (p), tube (ps) dumpling box (w)	mug2 (r), bottle (ps), toothpaste (p,ps), cup (r) herring tin (w,ps), chocsticks box (w), drops box (ps), tube (p),
Justin	mug1 (t,h) bottle (ps)	mug2 (t,h), glass (t) cup (t), tube (ps)

TABLE II

EXPERIMENT 1 SETUP. MUG1 AND 2 WERE ALSO SWAPPED. PINCH=P, RIM=R, PINCH SUPPORT=PS, POWER=W, TOP=T, HANDLE=H.

on sampling poses near to those on the approach trajectory in the training example.

The empirical studies we have performed show that: i) our method can learn from one or two example grasps of a particular type (e.g. power, pinch, pinch with support); ii) the system creates grasps for objects we tested with globally different shapes from the training objects, but which share some local shapes; iii) for a new object many new grasps can be generated, ordered by likelihood, and these form grasp clusters, allowing the selection of grasps that satisfy workspace constraints; iv) successful new grasps can also be generated even where shape recovery is incomplete for the new object; v) this model works for a total of 31 transferred grasps of novel objects made with two different multi-fingered robot systems.

Future planned work includes automating selection between types of trained grasps. At the moment we only show that a trained grasp type can generalise to new objects, and not that we can automatically choose a grasp type.

REFERENCES

- [1] C. Bard and J. Troccaz. Automatic preshaping for a dextrous hand from a simple description of objects. In *IEEE IROS*, 1990.
- [2] H. Ben Amor, O. Kroemer, U. Hillenbrand, G. Neumann, and J. Peters. Generalization of human grasping for multi-fingered robot hands. In *IROS*, pages 2043–2050. IEEE, 2012.
- [3] A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *IEEE International Conference on Robotics and Automation*, 2000.
- [4] J. Bohg, M. Johnson-Roberson, B. León, J. Felip, X. Gratal, N. Bergstrom, D. Kragic, and A. Morales. Mind the gap – robotic grasping under incomplete observation. In *IEEE ICRA*, 2011.
- [5] R.E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7:1–49, 1998.
- [6] A. Chiasso and S. Soatto. Monte carlo filtering on lie groups. In *Proc. of IEEE CDC*, volume 1, pages 304–309, December 2000.
- [7] J. Coelho, J. Piater, and R. Grun. Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot. In *Robotics and Autonomous Systems*, volume 37, pages 7–8, 2000.
- [8] R. Detry, C. Henrik Ek, M. Madry, and D. Kragic. Learning a dictionary of prototypical grasp-predicting parts from grasping experience. In *IEEE ICRA*, 2013.
- [9] D. Fischinger and M. Vincze. Empty the basket – a shape based learning approach for grasping piles of unknown objects. In *IEEE IROS*, 2012.
- [10] R. A. Fisher. Dispersion on a sphere. In *Proc. Roy. Soc. London Ser. A.*, 1953.
- [11] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, T. Asfour, and S. Schaal. Template-based learning of grasp selection. In *IEEE ICRA*, 2012.
- [12] U. Hillenbrand and M.A. Roa. Transferring functional grasps through contact warping and local replanning. In *IROS*, pages 2963–2970. IEEE, 2012.

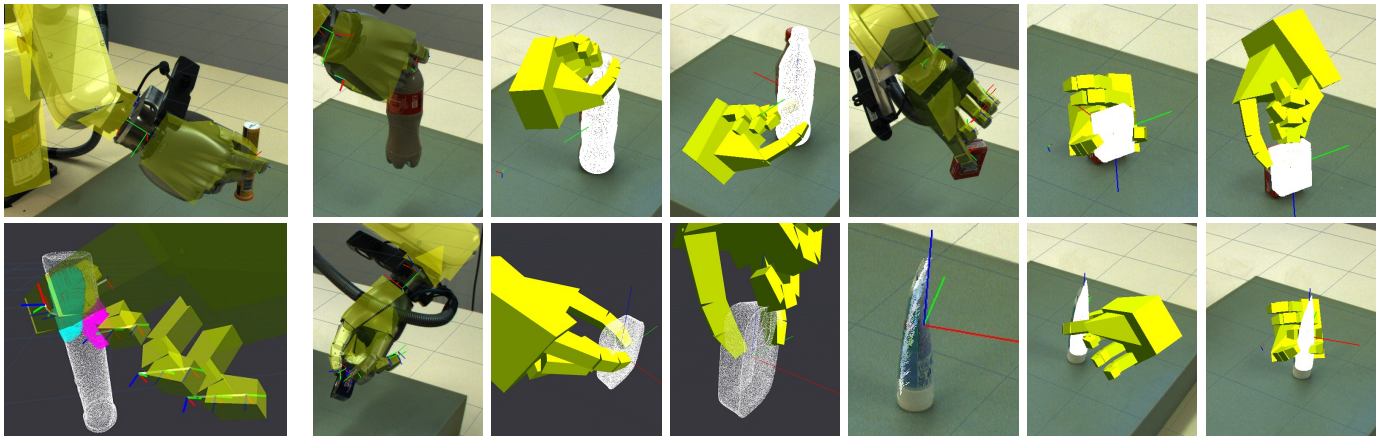


Fig. 6. Pinch with support grasp. The first column shows the demonstrated grasp and resulting learned contact models for a tablet tube. Remaining figures in middle and bottom rows show grasps subsequently generated by the agent for - coke bottle, orange-drops box and herring tin objects (executed grasps plus 2 alternative grasp clusters for each object), and a toothpaste tube object (executed grasp plus one alternative grasp cluster - both shown as simulator images due to poor camera viewing angle during execution).

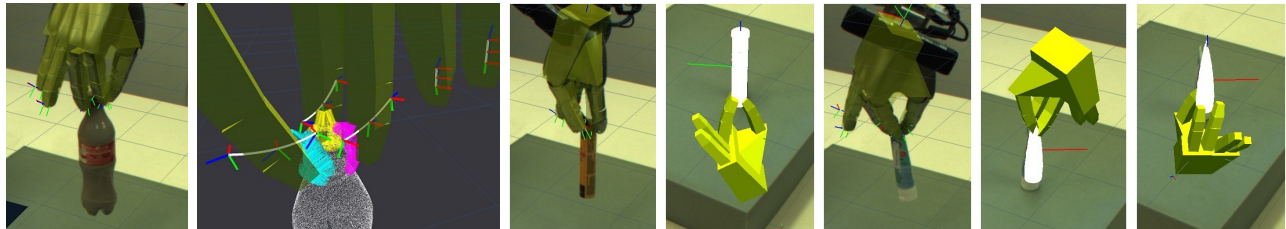


Fig. 7. Pinch grasp. Left to right: demonstrated grasp on coke bottle and resulting learned contact models; transferred grasp to tablet tube plus one additional grasp cluster; transferred grasp to toothpaste tube plus two additional grasp clusters.

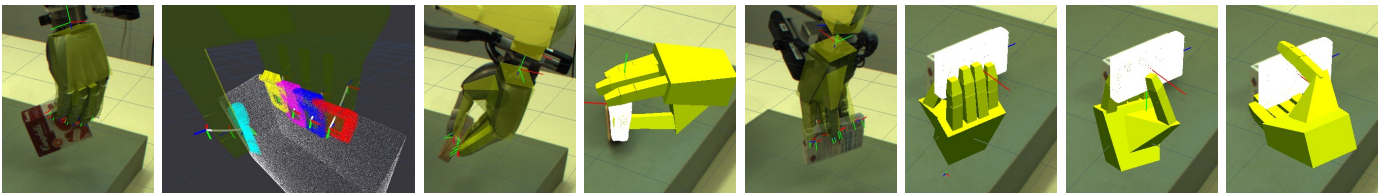


Fig. 8. Power grasp. Left to right: demonstrated grasp on instant dumplings box and learned contact models; transfer to herring tin (executed grasp plus one additional grasp cluster); transfer to chopsticks box (executed grasp plus three additional grasp clusters).

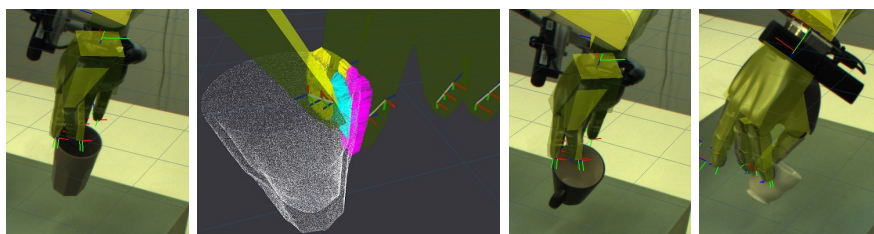


Fig. 9. Rim grasp. Left to right: demonstrated grasp on water glass and learned contact models; transferred grasps to two different tea-cups.

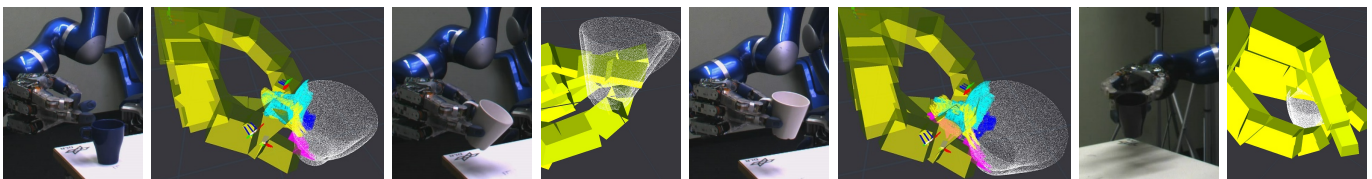


Fig. 10. Mug handle grasp. Left to right: demonstrated grasp mug 1 (fails!) and learned contact models; transfer to mug 2 (executed plus one additional cluster); demonstrated grasp mug 2; transfer to mug 1 (executed plus one additional cluster).

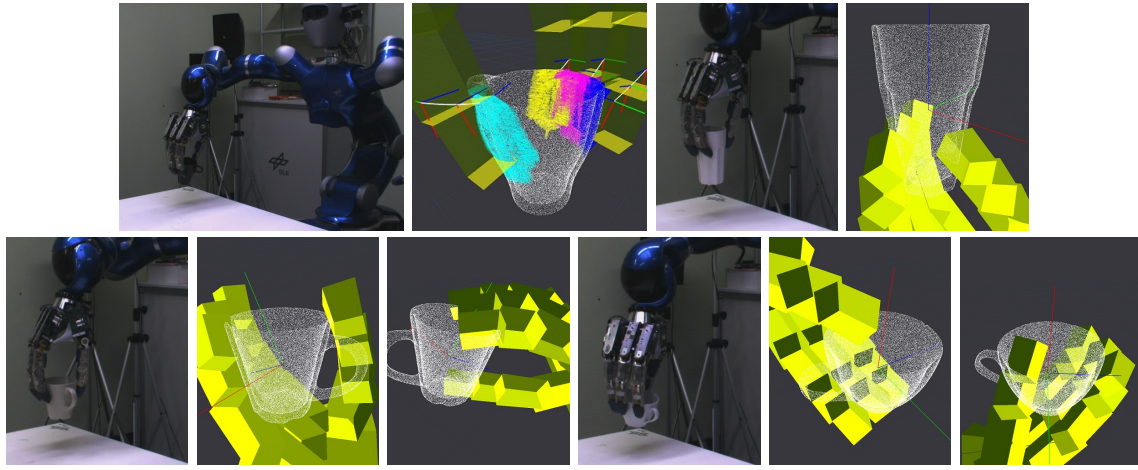


Fig. 11. Mug top grasp. Top row - demonstration and learned contact model for mug 1; transferred grasp plus one additional grasp cluster for water glass. Bottom row - transferred grasps plus two additional grasp clusters for mugs 2 and 3.

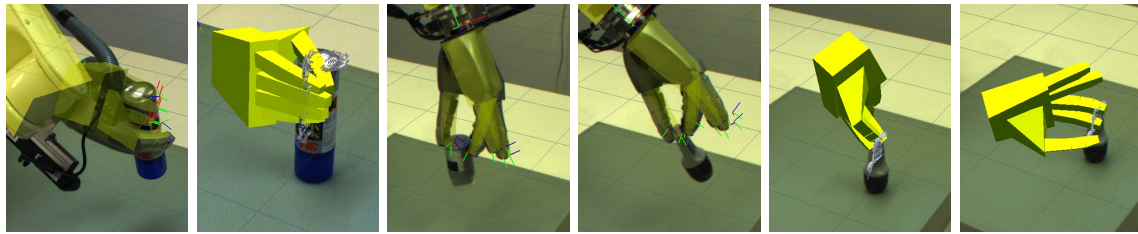


Fig. 12. Grasp transfers from complete point cloud models to objects represented by incomplete point clouds. Left to right: power grasp of lubricant spray (executed grasp plus one additional grasp cluster); pinch grasp on moisturiser; pinch grasp on antiperspirant (executed grasp plus two additional grasp clusters).

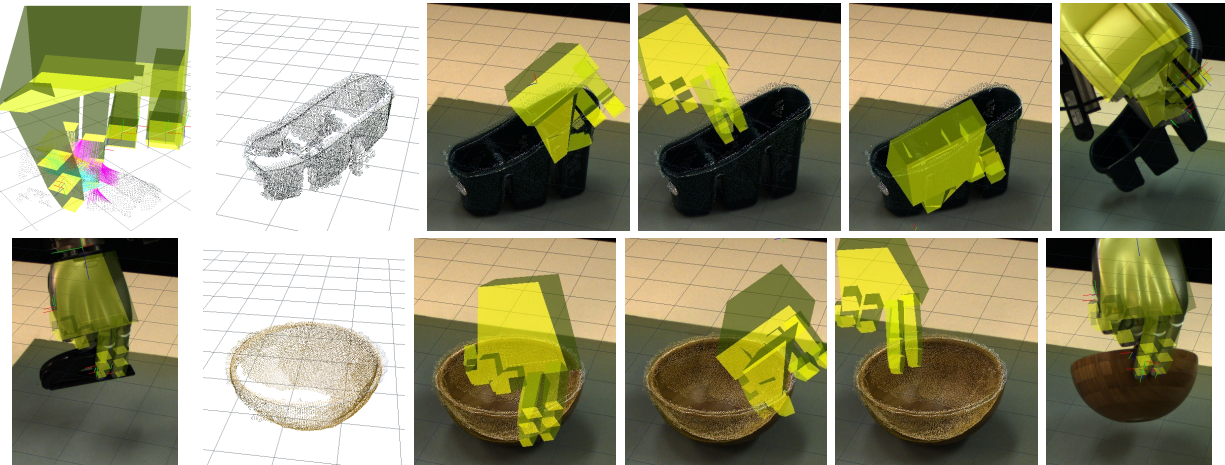


Fig. 13. Pinch grasp transfer from a stapler incomplete point cloud model (top and bottom left images) into two objects also represented by incomplete point clouds: a cutlery holder (top) and a wooden bowl (bottom).

- [13] Ulrich Hillenbrand. Non-parametric 3d shape warping. In *Pattern Recognition (ICPR)*, pages 2656–2659. IEEE, 2010.
- [14] O. Kroemer, R. Detry, J. Piater, and J. Peters. Combining active learning and reactive control for robot grasping. *RAS*, 58, 2010.
- [15] O. Kroemer, E. Ugur, E. Oztup, and J. Peters. A kernel-based approach to direct action perception. In *IEEE ICRA*, 2012.
- [16] A. Morales, E. Chinellato, A. H. Fagg, and A. P. del Pobil. Using experience for assessing grasp reliability. *Int. Journ. of Humanoid Robotics*, 1(4):671–691, 2004.
- [17] M. Popović, D. Kraft, L. Bodenhagen, E. Başeski, N. Pugeault, D. Kragic, T. Asfour, and N. Krüger. A strategy for grasping unknown objects based on co-planarity and colour information. *RAS*, 2010.
- [18] R. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *ICRA 2011*, Shanghai, China, May 9-13 2011.
- [19] A. Saxena, L. Wong, and A.Y. Ng. Learning grasp strategies with partial shape information. In *AAAI*, 2008.
- [20] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, 1986.
- [21] M. Spivak. *A comprehensive introduction to differential geometry*, volume 1. Publish or Perish Berkeley, 1999.
- [22] Erik B. Sudderth. *Graphical models for visual object recognition and tracking*. PhD thesis, MIT, Cambridge, MA, 2006.
- [23] S. Thrun and B. Wegbreit. Shape from symmetry. In *IEEE ICCV*, volume 2, pages 1824–1831, 2005.