

Alma Mater Studiorum Università di Bologna
Dipartimento di Elettronica Informatica e Sistemistica
Dottorato in Automatica e Ricerca Operativa (ING/INF-04)
XIX ciclo

Robotic Manipulation: Planning and Control for Dexterous Grasp

Ph.D. Thesis

Raffaella Carloni

Tutor: Prof. Claudio Melchiorri
Coordinatore: Prof. Claudio Melchiorri

Anno Accademico 2005/2006

Keywords:

Dexterous Manipulation, Visually-guided Grasping, Grasping Force Optimization, Robust Contact Detection, Hybrid Control

Raffaella Carloni

DEIS - University of Bologna

Via Risorgimento 2, 40136 Bologna.

Phone: +39 051 2093902, Fax: +39 051 2093073

Email: rcarloni@deis.unibo.it

URL: <http://www-lar.deis.unibo.it/people/rcarloni>

Copyright ©2007 by Raffaella Carloni. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording or any information storage and retrieval system, without permission in writing from the author.

Contents

Introduction	1
1 Grasp and Regrasp Planning of Planar Objects	5
1.1 Introduction	5
1.2 General Model	6
1.2.1 Object Description	6
1.2.2 Forces Description	6
1.3 Force-Closure Analysis	6
1.3.1 Force-Closure Conditions	7
1.3.2 Determination of Force-Closure Grasps	9
1.4 Regrasp Task	10
1.4.1 Regrasp Graph	10
1.4.2 Regrasp Strategy	12
1.5 Simulations	13
1.6 Conclusions and Future Work	15
1.6.1 Conclusion	15
1.6.2 Future Work	15
2 Optimal Grasp Force Distribution	19
2.1 Introduction	19
2.2 Force kinematic Model and Mathematical Background	19
2.2.1 Force Kinematic Model	20
2.2.2 Non-Linear Programming Optimization Background	21
2.3 Optimization Problem Model	23
2.3.1 Model Simplification	24
2.3.2 Convexity Analysis	26
2.3.3 Primal Form	26
2.3.4 Dual Form	27
2.3.5 Feasible Regions Method	27
2.4 Examples	30
2.5 Conclusions	33
3 Vision-Based Grasp Tracking	35
3.1 Introduction	35
3.2 Grasp Tracking	36
3.3 Invariants for Grasp Translation	38
3.3.1 Object and Grasp Descriptions	38
3.3.2 Grasp-related Invariant Features	40

3.4	Grasp Translation	42
3.4.1	Translation based on RRP Coordinates	43
3.4.2	Translation based on Relative Coordinates	43
3.4.3	Translation based on a Homography	44
3.5	Results	48
3.6	Conclusions and Future Work	55
3.6.1	Conclusions	55
3.6.2	Future Work	57
4	Robust Contact Detection and Force Regulation	59
4.1	Introduction	59
4.2	General Model	60
4.2.1	General Robotic Manipulator Model	60
4.2.2	Compliant Contact Force Model	61
4.2.3	Model Reduction	61
4.3	Hysteresis Control based on Force Measurement	62
4.4	Hybrid Controller	64
4.4.1	Position Controller	64
4.4.2	Force Controller	65
4.4.3	Control Strategy	65
4.4.4	Closed-loop System	66
4.5	Controller Design	69
4.5.1	Margin of Robustness	72
4.6	Simulations	74
4.6.1	Robustness	76
4.6.2	Change of Material	76
4.7	Controller Design with Saturation	78
4.7.1	Position Controller with Saturation	78
4.7.2	Force Controller with Saturation	79
4.7.3	Controller Design	80
4.7.4	Design of the Hysteresis Thresholds	87
4.7.5	Margin of Robustness	89
4.8	Simulations	89
4.9	Conclusions and Future Work	89
4.9.1	Conclusions	89
4.9.2	Future Work	89
	Conclusion	91
	Bibliography	92

List of Figures

1.1	Object and force geometric description: \mathcal{B}_d represents a discrete parameterization of the object boundary; f_i is a generic force exerted on the object boundary at the contact point p_i ; \hat{n}_i and \hat{t}_i are the normal and tangent unitary vectors at the boundary; \hat{r}_i and \hat{l}_i are the unitary vectors of the primitive forces f_i^r and f_i^l	7
1.2	(a) Force-closure grasp: the intersection point P between the supporting straight lines of two primitive forces lies inside the friction cone defined by $F_{k,\hat{r}}$ and $F_{k,\hat{l}}$; b) Non force-closure grasp: there is not a intersection between two primitive forces inside the friction cone defined by $F_{k,\hat{r}}$ and $F_{k,\hat{l}}$	8
1.3	Contact point p_i and p_j and definition of the region in which the third contact point p_k has to be placed. This region is given by the intersection of the two regions delimited by $p_{i'}$ and $p_{j'}$ and by p_r and p_l . p_n is the point whose normal passes through the intersection point P	9
1.4	Example of regrasp grid: (a) represents a convex object with a fixed contact point p_k ; (b) represents the grid of couples p_i and p_j for a fixed p_k	11
1.5	Example of the N grids of couples p_i and p_j for fixed values of p_k . In dashed line a path between two different force-closure grasp configuration.	12
1.6	Example of a regrasp graph A : this is a simple case in which there are only two grids I_k . The cells in the graph A are equal to 1 if there is a path between the two corresponding cells in I_1 and I_2	13
1.7	Regrasp graph: the force-closure grasp configurations are characterized by a white node and a arch with weight 1; otherwise the node is black and the corresponding arch has weight equal to ∞ . In dashed line, there is a possible path between two nodes.	14
1.8	Regrasp Strategy of a convex object: (a) initial configuration of the contact point $(p_i, p_j, p_k)_1 = [4 \ 1 \ 20]$; (b) final configuration of the contact point $(p_i, p_j, p_k)_2 = [18 \ 4 \ 22]$	15
1.9	Example of search of force-closure grasps: (a) represents a concave object with a fixed contact point p_k ; (b) represents the grid of couples p_i and p_j for a fixed p_k	16
1.10	Regrasp Strategy of a concave object: (a) initial configuration of the contact point $(p_i, p_j, p_k)_1 = [20 \ 47 \ 30]$; (b) final configuration of the contact point $(p_i, p_j, p_k)_2 = [17 \ 42 \ 35]$	17
2.1	(a) Object and contact coordinate frames; (b) Point contact with friction.	20
2.2	Relations between the primal and the dual forms: the optimal solutions are equivalent while the feasible regions are not.	29

2.3	Object, contact points and section defined by these points used in the examples with different approximations of the friction cones.	30
2.4	Evolution of the contact forces during the rotation of the object: initial position of the object and forces obtained with $m = 12$	33
3.1	A grasp-based positioning movement.	36
3.2	Experimental setups for tracking a grasp.	37
3.3	Grasp tracking procedure.	37
3.4	Location (left) and RRP (right) coordinates of contour points.	39
3.5	Grasp description using relative coordinates.	40
3.6	Grasp translation strategies.	43
3.7	Invariance of the RRP coordinates.	44
3.8	Invariance of the grasp description based on relative coordinates.	45
3.9	Use of a homography for the translation of grasp points.	45
3.10	Search of the correspondence of a given point.	47
3.11	Correction of the position of the grasp points.	48
3.12	Extraction and size-based selection of an object of interest.	50
3.13	Grasp tracking based on relative coordinates.	50
3.14	Grasp tracking based on a homography.	50
3.15	Tracking of an object and a selected grasp on a sequence of images using RRP coordinates.	54
4.1	A robotic manipulator interacting with a work environment with surface identified by S	60
4.2	Example of a switching strategy based on position information: the manipulator establishes and loses the contact in A and B , respectively. Several bounces (and switches between the position and force controllers) are present before the manipulator reaches the desired position/contact force configuration.	63
4.3	Example of switching strategy based on hysteresis. ℓ_{γ_1} and ℓ_{γ_2} define the sets for the switches based on force, ε is the margin of robustness and L_{V_F} is the largest Lyapunov function level set for closed-loop system with force controller that does not leave the right half plane.	64
4.4	Contact detection strategy with hysteresis. Constants γ_1, γ_2 define the thresholds for the change from mode $q = 0$ to mode $q = 1$ (and vice versa).	65
4.5	Flow and jump sets.	68
4.6	Hybrid control scheme. A position/force block generates the set-points f_c^d and x_1^d . The controller has as input the position of the manipulator x in the workspace coordinates and the contact force f_c measured by the sensor. Noise enters through the measurements of f_c and x	68
4.7	Computation of the maximum level set in the design of the switching strategy.	72
4.8	Compact set of initial conditions K	73
4.9	Margin of robustness ε of the switching strategy.	73
4.10	Phase diagram of the switching strategy. The plot depicts the Lyapunov functions level sets of the position and force controller, the ℓ_{γ_1} and ℓ_{γ_2} lines, and the trajectory of the system. The controller is able to avoid the bouncing off of the robotic system.	75
4.11	Phase diagram of the switching strategy. Zoom of Figure 4.10.	75
4.12	Contact forces during the interaction task. The plot depicts the interaction forces f_c , the desired contact force f_c^d and the logic variable q	76

4.13	Phase diagram of the switching strategy in presence of sensor noise and measurement errors in the manipulator position. The plot depicts the Lyapunov functions level sets of the position and force controllers, the ℓ_{γ_1} and ℓ_{γ_2} lines, and the trajectory of the system both without noise (dashed line) and different values of noise (continuous lines).	77
4.14	Phase diagram of the switching strategy. The plot depicts the different trajectories of the system for values of the environment material stiffness equal to 5 N/mm , 10 N/mm , 25 N/mm and 50 N/mm . The steady state point is changing since $x_1^F = f_c^d/kc$, in which the desired force is fixed at $f_c^d = 5 N$	77
4.15	Description of the constraints for the input-to-state stability of the system (4.19). If the first constraint in Equations (4.25) is valid, the gray zone corresponds to the input-to-state stable area.	83
4.16	Scheme of the architecture with the saturated controller.	88
4.17	Non-compact set \tilde{K} for the system with $\tilde{\kappa}_P$ position controller.	88
4.18	Phase diagram of the switching strategy with the saturated controllers. The plot depicts the ℓ_{γ_1} and ℓ_{γ_2} lines, and the trajectory of the system. The controller is able to avoid the bouncing off of the robotic system.	90

List of Tables

1.1	Regrasp sequence of example 1: convex object.	16
1.2	Regrasp sequence of example 2: concave object.	17
2.1	Determination of the minimum contact forces for different approximations of the friction cones and considering the weight of the object	31
2.2	Determination of the minimum contact forces for different approximations of the friction cones and considering an external wrench with non-null components in each direction	32
2.3	Number of iterations and computational times in the rotation of the ellipse.	34
3.1	Grasp descriptions expressed in location, RRP and relative location coordinates.	41
3.2	Detected point correspondences based on curvature matching.	49
3.3	Grasp translation based on RRP coordinates.	51
3.4	Grasp translation using relative coordinates.	52
3.5	Grasp translation based on a homography.	53
3.6	Grasp tracking along a stereo sequence of contours.	54
3.7	Grasp translation errors with the different techniques.	56
4.1	Design parameters of the controller for different values of the environment material stiffness k_c and fixed damping term b_c	78

*To the memory of my beloved grandparents
Alfonso ed Elisabetta*

Introduction

Dexterous manipulation by means of mechanical hands has become a research topic in the last two decades. The advantages of dealing with articulated mechanical hands over conventional grippers are, among others, the possibility of grasping objects of different shapes and of better distributing the grasping forces. The manipulation of *a priori* unknown objects involves several steps and problems that must be carefully considered and solved by proper planning and control algorithms. In this thesis, I have addressed both planning and control of robotic manipulation tasks.

Chapter 1 focuses on a particular kind of task of robotic manipulation, that is called regrasp. This is the highest level of dexterous manipulation and consists in changing the contact points on the object boundary in order to achieve a final desired configuration by starting from an initial one. The different steps of the regrasp sequence (i.e. a set of grasping configurations) are required to satisfy the force-closure constraint in each instant. This means that the forces applied by the fingers have to be able to balance any external force or torque on the object [5]. A methodology to plan a sequence of force-closure grasps using four fingers (three of them guaranteeing the force-closure grasps while moving the fourth) can be found in [54]. Another methodology for four fingers and including sliding movements of the fingers is presented in [69, 70]. All these works, developed for polygonal objects, imply the construction of a connectivity graph between the edges of the object and the application of search techniques to obtain the sequence of regrasp. Besides, some of the configurations obtained using four fingers can be very difficult or even impossible to be applied by a mechanical hand. The repositioning of the fingers near the initial grasp configuration considering non-polygonal objects can be found in [33], although a methodology to plan a sequence from an initial to a final configuration is not provided. A problem, related to the planning of the regrasp sequence, is the determination of independent regions of the object boundary, i.e. the force-closure is ensured if each finger of the robotic hand is placed in one of these different regions. Independent regions are used in [22] for two friction contacts and in [17] for four frictionless contacts. In this first Chapter, a new framework for planning regrasp sequences considering irregular 2D objects and three-finger grasps is presented. Some characteristic points are determined in order to establish the regions that allows force-closure grasps as well as the connectivity between them, obtaining what we define the regrasp map. By exploring the regrasp map as a shortest path problem, it is possible to determine regrasp sequence just with three fingers sliding or jumping on the object boundary and ensuring a force-closure grasp. This is a more challenging problem rather than the use of a fourth auxiliary finger.

Chapter 2 deals with the problem of distributing grasping forces, once the grasp satisfies the properties of force closure. In particular, the problem is to determine an adequate set of contact forces such that the external forces and torques are balanced and the object remains in equilibrium. This is one of the most basic requirements of a grasping action, otherwise the object may fall, and, since the grasp configuration and

the external forces often change during a manipulation action, the computational time of the algorithms has to be small enough to provide a valid response to these changes. Besides, the minimization of the contact forces is also desirable in order to avoid damages on the object. Some authors prioritized the simplicity and efficiency of the algorithms versus the optimality of the solution in order to allow the use of the algorithms in real-time procedures. Other authors focused their interest in the optimization of the solution. The main difficulty in determining the optimal grasping forces is the non-linearity of the friction models. The work in [53] presents optimization algorithms that consider the non-linearities of the friction cones, based on the primal form of a non-linear optimization problem, while in [66] the grasping forces are determined by considering the deformations of the fingertips. Nevertheless, the algorithms can only be applied off-line due to their computational cost. In [36] the friction cones is linearized by pyramids, and the search of the optimal grasping forces is formulated as a linear programming problem. The work in [13] proves that the efficiency of these methods could be improved using the duality theory of linear programming. The accuracy of these methods increases with the number of planes used in the approximation of the friction cones, but this also increases the computational cost of the optimization algorithms. A linear method that use a large number of planes in the approximation with a reasonable computational cost is proposed in [42]. The work in [9] points out that the friction cone constraints are equivalent to the positive definiteness of certain symmetric matrices. Based on these matrices, they developed gradient flow algorithms for real-time computation of optimal grasping forces. The force distribution problem is also formulated as a Linear Matrix Inequality problem in [26]. In [29], the dimension of the optimization problem is significantly reduced with a redefinition of the matrices that represent the friction cone constraints. The main drawbacks of the algorithms proposed by [9], [29] and [26] are that they require the selection of an initial solution that satisfies the friction cone constraints and the determination of the step size of the gradient algorithms. In [41] and [40] a solution for these problems is proposed and implemented. Another approach is followed in [74], in which the use of a neural network is proposed to solve the grasping force optimization problem. This second Chapter presents a new mathematical approach to efficiently solve the optimal force distribution problem. The problem is modeled as a non-linear minimization problem such that the objective function is the L_2 norm of the finger forces vector and the constraints are obtained by linearizing the friction cones. This model assures the convexity of the problem, implying that the dual theorem of non-linear programming can be applied, and the original problem is transformed into another one much easier to be solved. This method allows to use a large number of planes in the linear approximation without increasing the computational cost of the algorithm, allowing an accurate final solution. The dual theorem of non-linear optimization programming has been applied in many different fields such as economics, manufacturing, chemistry. A dual theorem in the case of cooperating robots is proposed in [38]. In this Chapter, we show that the dual theorem can also be successfully applied in grasping, in order to determine the minimum forces that the fingers have to exert on the object. This results have been presented in [20].

Once the grasp task has been planned, the control system should be able to guide the robotic manipulator towards the object in order to grasp it. **Chapter 3** proposes an image-based visual servoing procedure that grants the manipulator to reach the object, while the relative position/orientation of the object and the gripper of the robotic system changes due to the approaching movement of the robot towards the object itself. A fundamental requirement for a vision-based robotic system is the capability of detecting image features in order to control the pose of the end-effector with regard to a set of target features. In this Chapter, we analyze the problem of grasping unknown planar objects

by using an eye-in-hand system acting in an unknown dynamic environment. The visual servoing control loop should be able to trace significant points on the object, such as the grasp points, in order to achieve the desired position of the robotic gripper with respect to the target object. The proposed image-based visual servoing procedure requires three functional blocks to be executed in real time: the extraction of a set of features, i.e. the contour of the object, the selection of a stable grasp configuration and its tracking along different views of the object, and the control law whose set points are the grasp points. The problem of *tracking a grasp* consists in the search of the matching between different views of the object. We choose to compute a stable grasp configuration at the beginning and then to transpose it in the sequence of the acquired images. This has two main advantages: first of all, the computational cost of each iteration of the control loop is reduced. Secondly, by avoiding the grasp search at each time instant, the errors due to noisy sensor data decrease. In order to allow the grasp tracking, it is necessary to find a representation of the coordinates of the grasp points that is invariant in the image space according to different movement of the robotic arm. In particular, three different representations are presented: two are invariant with respect to 4 d.o.f. (translations along x , y , z and rotation about z) [64], the remaining one with respect to 6 d.o.f., [10]. Consequently, we have to use two different control laws: one that bounds the movement of the robotic arm to 4 d.o.f., the other to 6 d.o.f. The development of a visual servoing system for positioning a gripper with respect to an object has been considered in literature from a general point of view, without focusing on any particular manipulation task [4, 46, 47, 56]. A robust tracking algorithm is presented in [39], but the object shape is always supposed to vary in an affine model. Many works use the fundamental matrix and the epipolar geometry, although with the hypothesis of not having parallel views of the object to be grasped [15, 44, 75]. In the literature, the visual servoing system has generally been based on the use of features that are more easily to be extracted from the image other than the grasp points [23, 31]. For example, in [46] artificial marks explicitly set on the object and previously known are considered. In other works, geometric parameters of the object have been exploited, such as the centroid [47]. In [1] parameters related to the projection of the object, such as the measure of the image velocity at each pixel, have been adopted. The grasp search and the computation of the target position have been performed off-line in [31]. In [15], a method based on a homography transformation has been presented in which the whole shape of the object is used and the matrix of internal parameters of the camera is assumed as known. Several grasp-synthesis strategies can be found in [7, 12, 22, 28, 57, 64, 65]. Finally, regarding the design of the control law, under the assumption that the object is motionless, a simple proportional controller is the most common approach [32].

Chapter 4 focuses on the problem of the realization of the grasp, since it deal with the problem of the control of the interaction between a robotic manipulator and a generic work environment. In general, complex robotic systems, such as grasping and locomotion devices, involve the interaction between a robotic manipulator and its work environment. The control issue of such type of tasks is the regulation of the transition phase, in which the dynamic of the system is switching from the free to the constrained motion. In particular, the crucial point of the control is in the detection of contact/non-contact states since, when the manipulator gets in contact with the environment, large impulsive forces can cause the manipulator to bounce off and to become unstable. In the literature, the control of the dynamical behavior of the manipulator in interaction tasks has been a research topic for many years and several control synthesis schemes, both continuous and discontinuous, have been proposed. In the continuous case, impedance control schemes are used to establish a desired dynamic relationship between the robotic manipulator position and the force it exerts on the work environment [30]. A unified approach for motion and force control

is proposed in [37]: when the manipulator is moving freely, it is position-controlled and force-monitored; when the manipulator is constrained, it is force-controlled and position-monitored. During the impact transition, the control is a pure velocity damping so that the system can dissipate fast the impact energy and avoid bounces. In [71] an impedance control is used jointly with a hybrid system for the detection of the contact. On the other hand, the discontinuous control schemes in [11, 49, 55, 60, 72] present a switching control law in which a position controller is applied during non-contact motion while a position/force controller is applied during the transition phase and the contact stage. Commonly, the contact is detected by using position/velocity information. This work develops an innovative solution to the problem of robotic interaction. We present a hybrid control architecture that involves a position and a force controller, so that the position control law steers the manipulator to a target point in the workspace and, once the contact is detected, the control law switches to a force control that makes the manipulator exerting the desired force. The detection of the contact is realized with a strategy based on force information: when the measured contact force reaches some level, than the manipulator is supposed to be in contact. Vice versa, when the measured contact force goes below some level, than the manipulator is supposed to have detached the contact. The proposed hybrid control assures the stability of the manipulator so that, once the manipulator gets in contact with the environment, no bounces are present and it never loses the contact. This is realized by steering the robotic manipulator with a position controller into the basin of attraction generated by the the closed-loop system with the force controller [67]. The control scheme provides a margin of robustness with respect to actuators errors and to force sensors noise and it avoids chattering problems, i.e. multiple switching between the position and the force controller. This results have been presents in [59].

Chapter 1

Grasp and Regrasp Planning of Planar Objects

In this Chapter a new approach to planar objects dexterous manipulation by means of three-fingered hands is presented. In particular, we focus on a particular task: the regrasp task. In this context, it is required to plan sequences of force-closure grasps in order to achieve a desired stable configuration from an initial one. The discretized boundary of an irregular object is analyzed so that all the regions that ensures force-closure grasps are established: the connectivity between these regions provides the computation of a *regrasp graph*. The regrasp sequence is obtained either with slides or with jumps of the fingertips on the object boundary and is realized with the solution of a shortest path problem.

1.1 Introduction

In this Chapter, a new framework for planning regrasp tasks of irregular planar objects with three-finger robotic hands is presented. Given a discrete description of the object boundary, some characteristic points are determined in order to establish the regions that allow force-closure grasps as well as the connectivity between them, obtaining what we define the regrasp graph. By exploring the regrasp graph as a shortest path problem, it is possible to determine regrasp sequence just with three fingers sliding or jumping on the object boundary and ensuring a force-closure grasp in each step of the task. In this work, only one finger is supposed to be moved at each time, as in the methodologies developed for polygonal objects. The contact model is a point with hard friction and the dynamical effects produced at the contacts are not considered. As described in [2] and reviewed in [6], this effects has to be considered in the low-level of dexterous manipulation.

The Chapter is organized as follows: Section 1.2 presents the planar manipulation problem by providing the geometric description of the object and of the forces involved

in the task. Section 1.3 analyses the force-closure conditions and recalls methods for the determination of force-closure grasps. In Section 1.4, the regrasp task is presented and a solution for the regrasp planning problem is provided. Finally, simulation results with two different kind of objects, concave and convex, are reported in Section 1.5.

1.2 General Model

In this Section, we present the planar manipulation problem by giving a geometric description of the object to be grasped and of the forces exerted on it by the robot fingertips.

1.2.1 Object Description

Let \mathcal{B} be a smooth and closed curve describing a generic parameterization of the object boundary, let $\mathcal{B}_d = \{p_i, i = 1, \dots, N\}$ be a collection of N sampled points p_i that belong to \mathcal{B} and let $\theta = \{\theta_i, i = 1, \dots, N\}$ be the vector of the inward normal direction on each point p_i . The object is described by the discrete representation \mathcal{B}_d and the vector θ .

1.2.2 Forces Description

Let f_i be the force exerted by a robotic finger on a contact point p_i and let f_i^n and f_i^t be its normal and tangent components to the object boundary, which directions are given, respectively, by the unitary vectors $\hat{n}_i = [\cos \theta_i \quad \sin \theta_i]^T$ and $\hat{t}_i = [-\sin \theta_i \quad \cos \theta_i]^T$.

By assuming the Coulomb friction model, the finger slippage on the object boundary is avoided if the normal and tangent components of the finger force satisfy

$$\mu \|f_i^n\| \geq \|f_i^t\|,$$

where μ is the friction coefficient. This implies that the force exerted by the finger has to lie inside the friction cone centered on the normal direction to the object boundary and delimited by the two primitive forces f_i^r and f_i^l which directions are given, respectively, by the unitary vectors

$$\hat{r}_i = [\cos(\theta_i - \varphi) \quad \sin(\theta_i - \varphi)] \quad (1.1)$$

$$\hat{l}_i = [\cos(\theta_i + \varphi) \quad \sin(\theta_i + \varphi)] \quad (1.2)$$

where $\varphi = \arctan \mu$. We indicate the straight lines of the unitary vectors \hat{n}_i , \hat{t}_i , \hat{r}_i and \hat{l}_i with the symbol $F_{i,\hat{v}}$, where \hat{v} is the generic unitary vector.

The complete scenario of the geometric description of the object and of the forces exerted on its boundary is depicted in Figure 1.1.

1.3 Force-Closure Analysis

In this section, we analyse the conditions that guarantee a force-closure grasp by recalling some previous results presented in [19].

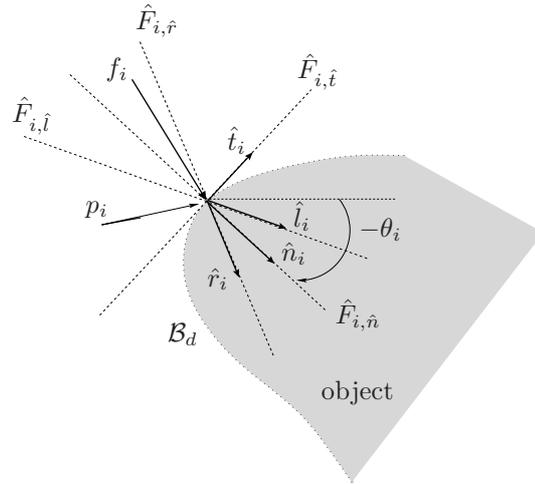


Figure 1.1: Object and force geometric description: \mathcal{B}_d represents a discrete parameterization of the object boundary; f_i is a generic force exerted on the object boundary at the contact point p_i ; \hat{n}_i and \hat{t}_i are the normal and tangent unitary vectors at the boundary; \hat{r}_i and \hat{l}_i are the unitary vectors of the primitive forces f_i^r and f_i^l .

1.3.1 Force-Closure Conditions

A set of contact points allows a force-closure grasp, if and only if the convex hull defined by the primitive wrenches contains the origin [3]. Even when this is a general necessary and sufficient condition and it can be applied considering 2D or 3D objects and any number of fingers, some authors have developed other necessary and sufficient conditions that avoid to compute the convex hull in some specific cases. We recall here the proposition in [34] that gives the necessary and sufficient condition for the force closeness in the case of grasp of 2D objects with a three-finger robotic hand.

Proposition 1.3.1. *Three contact points p_i , p_j and p_k allow a force-closure grasp if and only if: a) the unitary primitive vectors that bound the friction cones at these points positively span the force space, and b) at least one intersection point P between the supporting straight lines of the primitive forces lies inside the friction cone at the other contact point.*

Figure 1.2(a) shows an example of three contact points that satisfy the necessary and sufficient condition in Proposition 1.3.1 allowing a force-closure grasp, while Figure 1.2(b) shows an example of three contact points that do not satisfy the condition.

As stated [19], from Proposition 1.3.1, two Lemmas follows. Before, some definitions are needed.

Definition 1.3.2. *Two points p_i and p_j are primitive-primitive opposite (PPO) if a primitive force applied at p_i and a primitive force applied at p_j have opposite directions. Besides, if these forces are collinear these points are also primitive-primitive antipodal (PPA).*

Definition 1.3.3. *Two points p_i and p_j are primitive-normal opposite (PNO) if a primitive force applied at p_i and the normal force applied at p_j have opposite directions. Besides,*

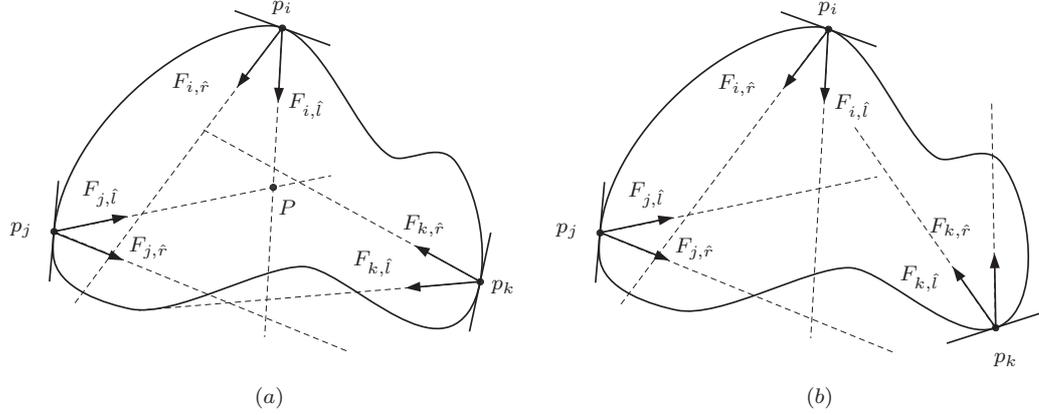


Figure 1.2: (a) Force-closure grasp: the intersection point P between the supporting straight lines of two primitive forces lies inside the friction cone defined by $F_{k,\hat{r}}$ and $F_{k,\hat{l}}$; b) Non force-closure grasp: there is not a intersection between two primitive forces inside the friction cone defined by $F_{k,\hat{r}}$ and $F_{k,\hat{l}}$.

if these forces are collinear these points are also primitive-normal antipodal (PNA).

Definition 1.3.4. Two points p_i and p_j are normal-normal opposite (NNO) if the normal force applied at p_i and the normal force applied at p_j have opposite directions. Besides, if these forces are collinear these points are also normal-normal antipodal (NNA).

Lemma 1.3.5. Let p_i and p_j be two contact point and let P be the intersection point between the straight lines of two primitive forces applied at p_i and p_j . In order to obtain a force-closure grasp, the third point p_k has to be placed into the intersection of the two regions on the object boundary defined as follows:

- The region limited by two points whose primitive force are opposite to the two primitive forces that determine P . In Figure 1.3, these two PPO points are $p_{i'}$ and $p_{j'}$.
- The region of points where the friction cone contains P . In Figure 1.3, these two points are p_r and p_l .

Lemma 1.3.6. It always exists a point p_n in the region identified by Lemma 1.3.5, whose normal force passes through P .

The two Lemmas 1.3.5 and 1.3.6 are illustrated in Figure 1.3 and are used in the following section in order to obtain the set of possible force-closure grasps.

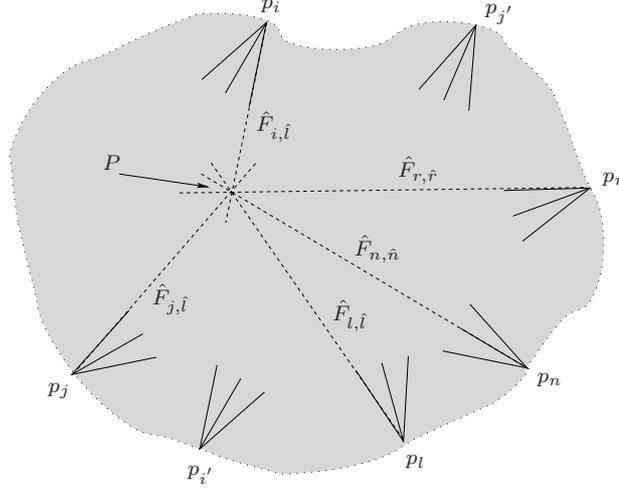


Figure 1.3: Contact point p_i and p_j and definition of the region in which the third contact point p_k has to be placed. This region is given by the intersection of the two regions delimited by $p_{i'}$ and $p_{j'}$ and by p_r and p_l . p_n is the point whose normal passes through the intersection point P .

1.3.2 Determination of Force-Closure Grasps

Recalling the work in [19], in this subsection we present a procedure for the computation of a force-closure grasp configuration for 2D objects and a three-finger robotic hand. In particular, given two contact points p_i and p_j , this procedure computes the region in which the third contact point p_k has to be placed in order to guarantee that the configuration given by p_i , p_j and p_k is a force-closure grasp. As stated in [19], this procedure has a computational cost of $O(N^2)$, since it only considers the position of the two contact points p_i and p_j and the existence of PNA points on the object boundary. Note that the search of the PNA point

From Lemma 1.3.5, the region where p_k has to be placed is limited by PPO points of p_i and p_j , i.e. $p_{i'}$ and $p_{j'}$.

Let Γ_i and Γ_j be the torques produced by the unitary normal vectors at $p_{i'}$ and $p_{j'}$ with respect to the intersection P . Therefore, Γ_i and Γ_j are given by

$$\Gamma_i = (p_i - p_{i'}) \times \hat{n}_{i'} \quad (1.3)$$

$$\Gamma_j = (p_j - p_{j'}) \times \hat{n}_{j'} \quad (1.4)$$

According to the signs of Γ_i and Γ_j , a combinations of two generic contact points p_i and p_j is classified into two types:

- *Odd Combination:* The signs of Γ_i and Γ_j satisfy the following relation:

$$\text{sign}(\Gamma_i) \neq \text{sign}(\Gamma_j) \quad (1.5)$$

This means that there is an odd number of points p_n between $p_{i'}$ and $p_{j'}$ where a normal force produces a null torque with respect to P and the sign of the torque produced by a normal force is different on each side of these points. The existence of these points between $p_{i'}$ and $p_{j'}$ guaranties that the intersection between the regions

defined by Lemma 1.3.5 is not null. Therefore, it is possible to assure that p_i and p_j together with a third contact point can produce a force-closure grasp.

- *Even Combination:* The signs of Γ_i and Γ_j satisfy the following relation:

$$\text{sign}(\Gamma_i) = \text{sign}(\Gamma_j) \quad (1.6)$$

This means that there can be either no one or an even number of points p_n between p_i and p_j where a normal force produces a null torque with respect to P and the sign of the torque produced by a normal force is equal on each side of these points. Therefore, it is not possible to assure that p_i and p_j together with a third contact point can produce a force-closure grasp.

Following this analysis, given two contact points p_i and p_j , it is possible to compute the region of the object boundary where the third contact point p_k can be placed.

1.4 Regrasp Task

The regrasp task consists in the planning of the movements that each finger of the robotic hand has to execute in order to realize a change in the grasp configuration. In particular, at the beginning of the regrasp task, the three-finger robotic hand is keeping the object in an initial force-closure grasp configuration. Then, in order to reach a final force-closure grasp configuration without losing the object, it is necessary to move the fingers by passing through different grasp configurations that realize a force-closure.

1.4.1 Regrasp Graph

By following the discussion of Section 1.3, given the discretized parameterization of the boundary of an irregular planar object and the inward normal direction of each point of the boundary, we are able to know all the possible force-closure grasp configurations where the contact points can be placed so that to guarantee the stability of the object. In particular, for every contact point p_k , we can compute all the couples of points (p_i, p_j) that grant a force-closure grasp. This means that for every point p_k , we can build a grid in the 2D space defined by the contact points p_i and p_j in which it is possible to distinguish between possible force-closure grasps and non possible force-closure grasps.

Figure 1.4(a) depicts a generic planar convex object whose boundary is discretized by a number of points $N = 40$. If we suppose that the contact point p_k is fixed, we can build up a grid, illustrated in Figure 1.4(b), that represents all the combination of couples of point (p_i, p_j) . In particular, the white cells represent the couples of point (p_i, p_j) that, combined with the fixed point p_k , guarantee a force-closure grasp configuration, while the dark cells represents the configurations that don't guarantee a force-closure grasp. Note that the grid in Figure 1.4(b) is symmetric with respect to the line characterized by $p_i = p_j$. The computational cost for one grid is $O(n^2/2)$.

In order to realize a change in the grasp configuration we have to plan all the movements, i.e. the steps of the regrasp sequence, that the fingers have to execute from the initial to the final configuration. Therefore we have to build up all the grids of couples of contact points (p_i, p_j) that correspond to all the possible N points p_k . Eventually, we come up with N grids, as depicted in Figure 1.5. These grids are the graphical representation of all the combinations of three contact points p_i , p_j and p_k . If the three contact points realize a force-closure grasp, then the corresponding cell is white, otherwise, if the three

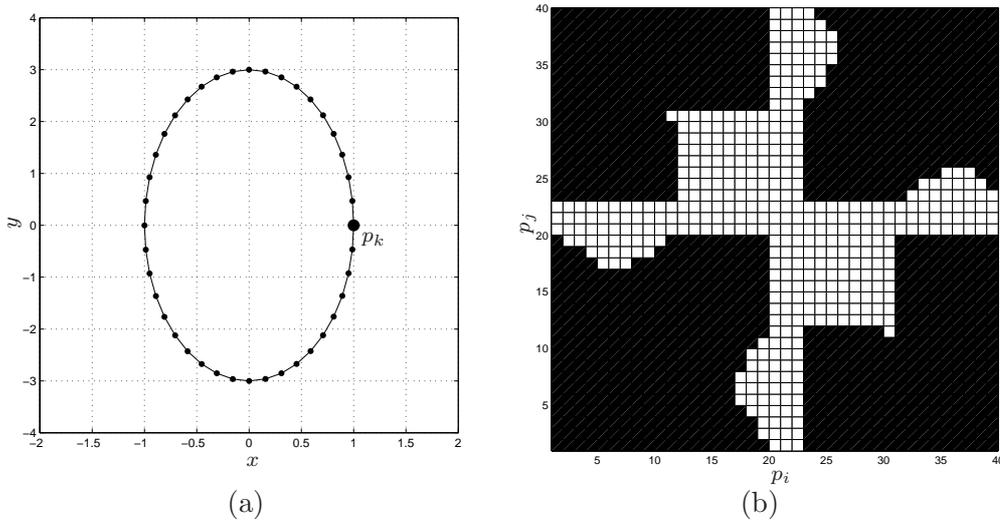


Figure 1.4: Example of regrasp grid: (a) represents a convex object with a fixed contact point p_k ; (b) represents the grid of couples p_i and p_j for a fixed p_k .

contact points do not realize a force-closure grasp, then the corresponding cell is black. Notice that the computational cost for all the grids is $O(n^3/2)$.

By exploring the grids, we can notice that there can be paths that join an initial grasp configurations $(p_i, p_j, p_k)_1$ and a final one $(p_i, p_j, p_k)_2$, by always passing through white cells. A possible path is shown in dashed line in Figure 1.5. Note that the initial and the final configurations are white cells and the path is passing through white cells. In the path, we are changing one cell at each time by movements in the same plane (characterized by a fixed p_k) or by changing plane (therefore by changing the value of p_k) but maintaining a fixed values of the contact points p_i and p_j . This means that we are assuming that each step of the planning is characterized by a single movement of one finger.

The regrasp grids can be joined in a unique 2D grid with dimension $N^3 \times N^3$, that contains all the information of the grids. In particular, this grid, denoted with A , is basically a graph that we call the *regrasp graph*.

In order to explain how to build the regrasp graph A , let's analyse a single $N \times N$ regrasp grid, that we call I_k . Each grid I_k is transformed in a $N^2 \times N^2$ grid, that we call A_k . Each generic cell (p_i, p_j) of I_k is compared with all the others cells of I_k : if both cells are white (i.e. a force-closure grasp is admitted), the corresponding cell $A_k(p_i, p_j)$ is equal to 1, otherwise is equal to 0. This means that the two cells are correlated by a path of weight 1 or 0. Note that the grid A_k is symmetric with respect $p_i = p_j$. Finally, each grid I_k has a corresponding grid A_k .

All the A_k grids are put together in order to compose the main diagonal of the regrasp graph. In the superdiagonal and the subdiagonal of the regrasp graph, there are the correlation between all the different I_k , build up with the same procedure. In particular, only the cells with the same values of (p_i, p_j) and adjacent values of p_k are compared, since we suppose to have a continuity of the finger movements. The remaining parts of the regrasp graph are cells with a 0 value, since there are no direct correlations between them.

In order to visualize this procedure, we describe in Figure 1.6 the easy case of two I_k cells (I_1 and I_2) of dimension 2×2 . On the left part of the Figure, the two grids $I_k, k = 1, 2$

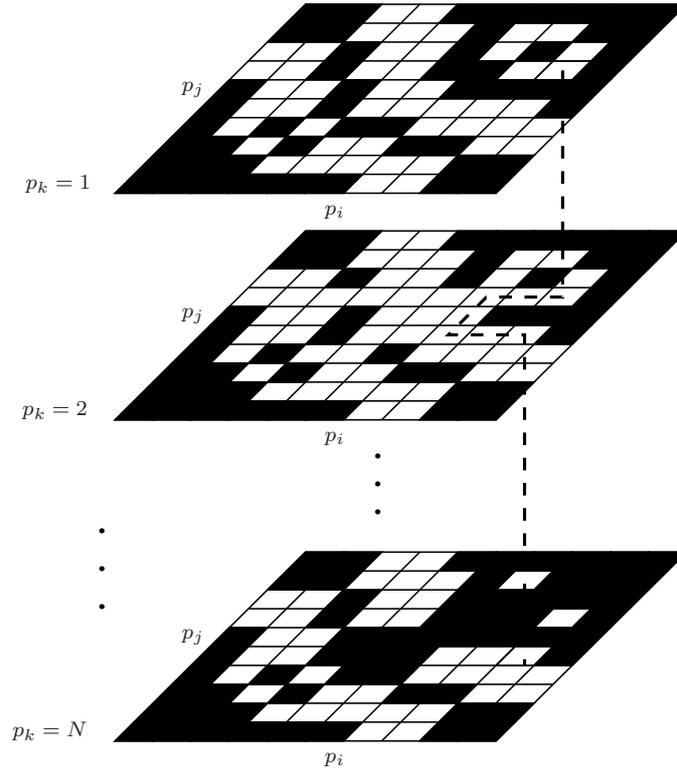


Figure 1.5: Example of the N grids of couples p_i and p_j for fixed values of p_k . In dashed line a path between two different force-closure grasp configuration.

are plotted: if a cell is white, the configuration grasp realizes a force-closure, otherwise is black (gray in the Figure). Note that, for example, the configuration $(p_i, p_j, p_k) = (1, 1, 1)$ is a force closure, while the configuration $(p_i, p_j, p_k) = (1, 2, 7)$ is not. Each cell of the grids I_k are numerated in a progression. Then, the graph A is build up. If two adjacent cells are both white, the corresponding cell in A is marked as 1, otherwise it is 0. The cells are considered adjacent if they have a common side or if they have the same position (p_i, p_j) but in different level k . Note that, for example, from cell 1 and 2 there is a path of weight equal to 1, as well as for cells $(5, 6)$, $(6, 8)$, $(1, 5)$, $(2, 6)$ since they are adjacent. Only for these combinations there is a 1 in the graph A .

More generally, Figure 1.7 depicts a schematic example of a regrasp graph. We can notice that the nodes which guarantee a force-closure grasp configuration are white, otherwise the cells are black. Each arch of the graph is oriented in both directions and it has weight equal to 1 if the change of contact point is allowed. Otherwise, if the change of contact point does not guarantee a force-closure grasp, i.e. one of the two nodes that the arch is correlating is black, the arch has a weight equal to ∞ . In Figure 1.7, a possible path between two nodes is plotted in dashed line.

1.4.2 Regrasp Strategy

Once the regrasp graph A is build, the search of the regrasp sequence is a simple linear combination problem on the graph A . It is necessary to explore the regrasp graph in order

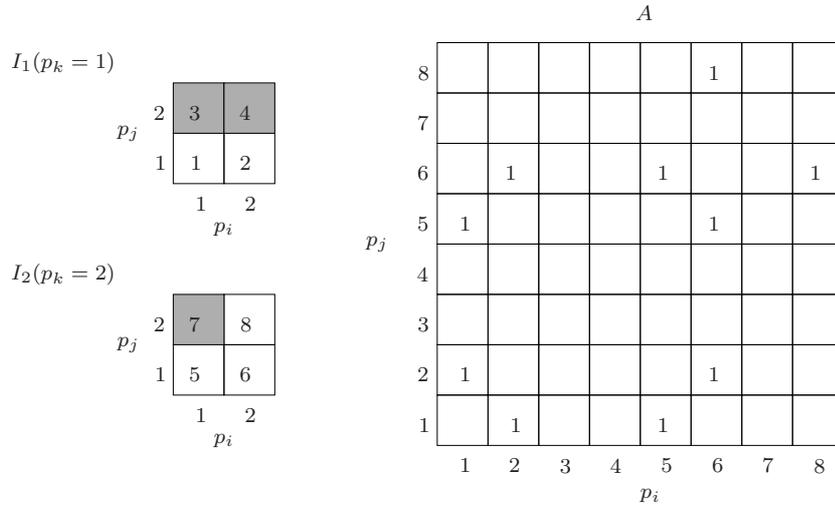


Figure 1.6: Example of a regrasp graph A : this is a simple case in which there are only two grids I_k . The cells in the graph A are equal to 1 if there is a path between the two corresponding cells in I_1 and I_2 .

to find a path from an initial configuration to a final configuration and the fastest way to find a solution is to apply the shortest path algorithm. We present this results in the following proposition.

Proposition 1.4.1. *Given a discretized parameterization of an irregular planar object \mathcal{B}_d and the vector θ of the inward direction at each point of the boundary, the regrasp sequence from an initial force-closure configuration $(p_i, p_j, p_k)_1$ to a final force-closure configuration $(p_i, p_j, p_k)_2$ is realized by building up and analysing the regrasp graph A . In particular, the path is given by the solution of the shortest path algorithm on the graph A .*

Once the regrasp path has been decided, we can distinguish among two different types of movements of the robot fingers. In particular, in the designed path we can observe which is the finger that is moving in each step of the regrasping task. If the two fingers that have to remain fixed on the object realize a force-closure grasp, we can make the finger moving with a jump, otherwise with a slide.

1.5 Simulations

In this section, two examples of the proposed methodology are presented for both a convex and a concave object. In the two examples, the friction coefficient is $\mu = 0.3$.

The steps of the simulation procedure are:

Step 1) Given the discretized parameterization \mathcal{B}_d and the vector θ of an irregular planar object, find the N grids I_k corresponding to all the configurations of force-closure grasps.

Step 2) Compute the regrasp graph A .

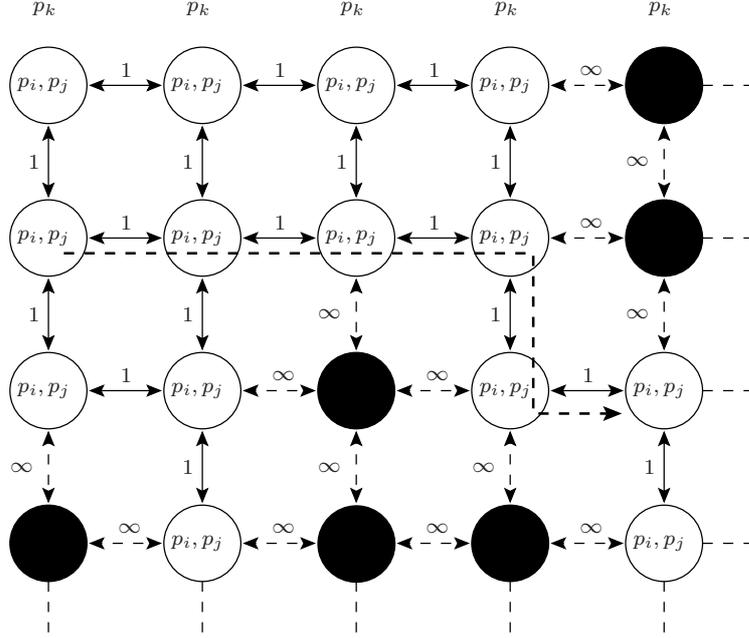


Figure 1.7: Regrasp graph: the force-closure grasp configurations are characterized by a white node and a arch with weight 1; otherwise the node is black and the corresponding arch has weight equal to ∞ . In dashed line, there is a possible path between two nodes.

Step 3) By using the library Matlab BGL v2.0, compute the shortest path on the graph A .

Example 1: Convex Object The object considered in this example is an ellipse described by $N = 40$ points, as the one shown in Figure 1.4(a).

The initial positions of the fingers are $(p_i)_1 = [0.891 \ 1.3621]$, $(p_j)_1 = [1 \ 0]$, $(p_k)_1 = [-0.9877 \ 0.4682]$, see Figure 1.8(a). The final positions of the fingers are $(p_i)_2 = [-0.8912 \ 1.361]$, $(p_j)_2 = [0.891 \ 1.3621]$, $(p_k)_2 = [-0.9876 \ -0.4706]$, see Figure 1.8(b). In order to simplify the notation, we refer to these points by using their indexes in the discrete parameterization \mathcal{B}_d , i.e. $(p_i, p_j, p_k)_1 = [4 \ 1 \ 20]$ and $(p_i, p_j, p_k)_2 = [18 \ 4 \ 22]$. The regrasp sequence is detailed in Table 1.1, in which the finger that is going to be moved and the type of movement are specified.

Example 2: Concave Object The object considered in this example is shown in Figure 1.9(a) and is described by $N = 50$ points. The grid in Figure 1.9(b) represents all the combination of couples of point p_i and p_j for a fixed value of p_k .

The initial positions of the fingers are $(p_i)_1 = [-3.43 \ 1.41]$, $(p_j)_1 = [2.67 \ -1.64]$, $(p_k)_1 = [-2.97 \ -1.47]$, see Figure 1.10(a). The final positions of the fingers are $(p_i)_2 = [-1.39 \ 1.59]$, $(p_j)_2 = [3.26 \ -4.90]$, $(p_k)_2 = [-0.17 \ -2.73]$, see Figure 1.10(b). In order to simplify the notation, we refer to these points by using their indexes in the discrete parameterization \mathcal{B}_d , i.e. $(p_i, p_j, p_k)_1 = [20 \ 47 \ 30]$ and $(p_i, p_j, p_k)_2 = [17 \ 42 \ 35]$. The regrasp sequence is detailed in Table 1.2, in which the finger that is going to be moved and the type of movement are specified.

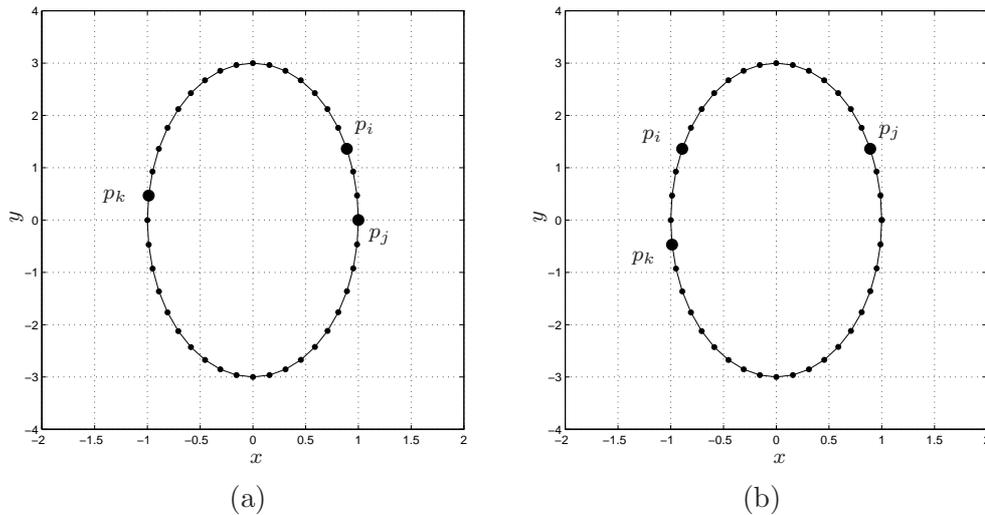


Figure 1.8: Regrasp Strategy of a convex object: (a) initial configuration of the contact point $(p_i, p_j, p_k)_1 = [4 \ 1 \ 20]$; (b) final configuration of the contact point $(p_i, p_j, p_k)_2 = [18 \ 4 \ 22]$.

1.6 Conclusions and Future Work

1.6.1 Conclusion

In this Chapter, a new method for the planning of regrasp task has been presented. In particular we have analyzed the case of the manipulation of irregular planar objects by means of three-finger robotic hands. By a graphic visualization of all the force-closure grasp configurations, it is possible to build up a regrasp graph that allow the planning of a regrasp sequence, i.e. the sequence that the fingers of the robot have to comply in order to go from an initial to a final grasp configuration. The regrasp sequence is obtained either with slides or with jumps of the fingertips on the object boundary and is realized with the solution of a shortest path problem on the regrasp graph.

1.6.2 Future Work

Future work will be aimed to the generalization of the proposed approach in order to include the kinematic constraints of the robotic hand and/or other constraints due to the requirements of a specific task. These constraints will influence the construction of the regrasp graph by reducing its dimension, since it can be that some grasp configurations are not physically feasible.

Table 1.1: Regrasp sequence of example 1: convex object.

Step	p_i	p_j	p_k	type of movement
(1)	4	1	20	jump
(2)	14	1	20	slide
(3)	14	2	20	slide
(4)	14	3	20	slide
(5)	14	4	20	slide
(6)	15	4	20	slide
(7)	16	4	20	slide
(8)	17	4	20	slide
(9)	18	4	20	jump
(10)	18	4	22	STOP

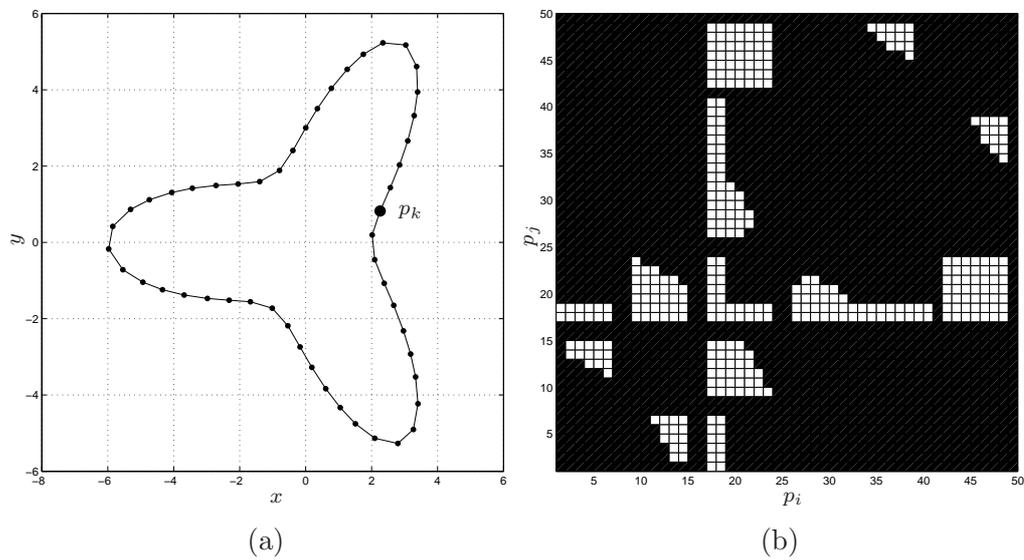


Figure 1.9: Example of search of force-closure grasps: (a) represents a concave object with a fixed contact point p_k ; (b) represents the grid of couples p_i and p_j for a fixed p_k .

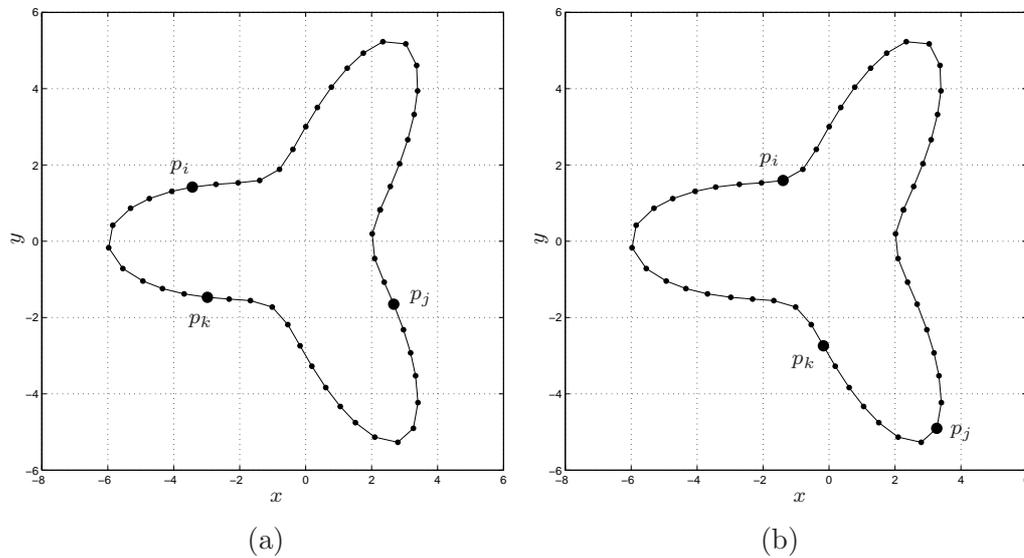


Figure 1.10: Regrasp Strategy of a concave object: (a) initial configuration of the contact point $(p_i, p_j, p_k)_1 = [20 \ 47 \ 30]$; (b) final configuration of the contact point $(p_i, p_j, p_k)_2 = [17 \ 42 \ 35]$.

Table 1.2: Regrasp sequence of example 2: concave object.

Step	p_i	p_j	p_k	type of movement
(1)	20	47	30	slide
(2)	20	47	31	slide
(3)	20	46	31	slide
(4)	20	45	31	slide
(5)	20	45	32	slide
(6)	20	45	33	slide
(7)	19	45	33	slide
(8)	19	44	33	slide
(9)	18	44	33	slide
(10)	17	44	33	slide
(11)	17	43	33	slide
(12)	17	42	33	slide
(13)	17	42	34	slide
(14)	17	42	35	STOP

Chapter 2

Optimal Grasp Force Distribution

One of the basic issues in robotic manipulation is the determination of a suitable set of grasping forces that a mechanical hand has to apply in order to balance the external forces and torques applied on a generic object and to keep it in equilibrium. In this Chapter we present a new mathematical approach to efficiently obtain the optimal solution of this problem by means of the dual theorem of non-linear optimization programming. In particular, the dual theorem can be applied to the force distribution problem only if it is modeled such as the basic convexity property is satisfied.

2.1 Introduction

This Chapter presents a new mathematical approach to efficiently solve the optimal force distribution problem. The problem is modeled as a non-linear minimization problem such that the objective function is the L_2 norm of the finger forces vector and the constraints are obtained by linearizing the friction cones. This model assures the convexity of the problem, implying that the dual theorem of non-linear programming can be applied, and the original problem is transformed into another one much easier to be solved. This method allows to use a large number of planes in the linear approximation without increasing the computational cost of the algorithm, allowing an accurate final solution.

The Chapter is organized as follows: Section 2.2 describes the kinematic model of the grasping forces and a mathematical background of non-linear programming optimization; in Section 2.3 the model of the force optimization problem is developed and the proposed solution is presented. Simulation results are reported in Section 2.4.

2.2 Force kinematic Model and Mathematical Background

In this Section, we present a kinematic model of the reaction forces, due to the interaction between a robotic hand and a generic object, and we briefly illustrate a mathematical

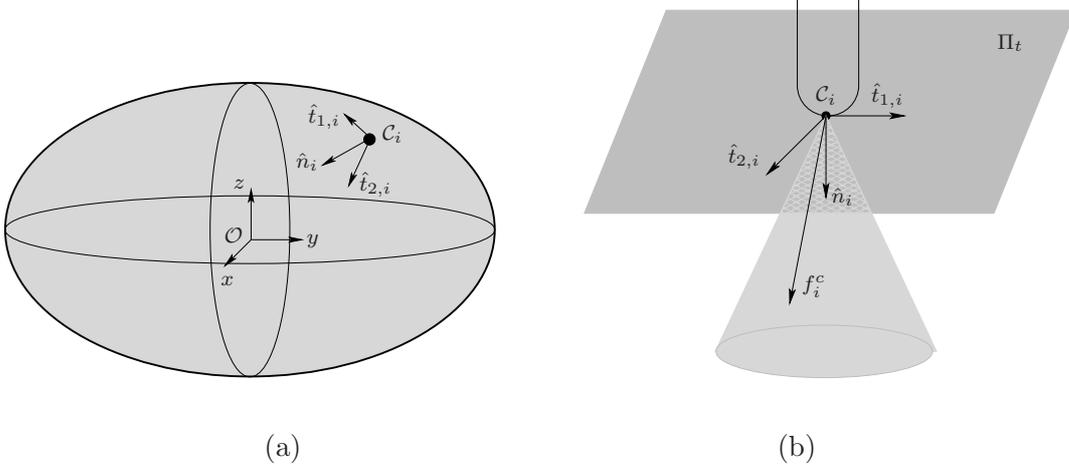


Figure 2.1: (a) Object and contact coordinate frames; (b) Point contact with friction.

background of non-linear programming optimization with inequality constraints, that is necessary in order to introduce our method of grasping force distribution.

2.2.1 Force Kinematic Model

Let \mathcal{B} be a smooth and closed curve describing the parameterization of an object boundary and let \mathcal{O} be the object reference frame attached at the object center of mass. This scenario is depicted in Figure 2.1. Figure 2.1a highlights that the contact reference frame \mathcal{C}_i is considered attached at the contact point c_i on \mathcal{B} and it is oriented so that its axes are defined by \hat{n}_i , the inward unitary vector normal to \mathcal{B} , and by $\hat{t}_{1,i}$ and $\hat{t}_{2,i}$, two orthonormal vectors belonging to the plane Π_t tangent to \mathcal{B} at c_i .

Let F_i be the set of contact forces that a robotic finger can apply on the object at c_i . The generic contact force $f_i^c \in F_i$ is represented by a three-component vector expressed with respect to the contact frame \mathcal{C}_i

$$f_i^c = (\alpha_{n_i} \ \alpha_{t_{1,i}} \ \alpha_{t_{2,i}})^T, \quad (2.1)$$

where α_{n_i} , $\alpha_{t_{1,i}}$ and $\alpha_{t_{2,i}}$ are the magnitudes of the components of f_i^c along the unitary vectors \hat{n}_i , $\hat{t}_{1,i}$ and $\hat{t}_{2,i}$. Based on the hard point contact model with friction, the contact forces must satisfy the Coulomb's law in order to avoid the finger slippage on the object boundary. Therefore, as depicted in Figure 2.1b, F_i geometrically represents a cone, called the friction cone, centered on the direction normal to the object boundary, and it is given by

$$F_i = \left\{ f_i^c \in \mathbb{R}^3 \mid \sqrt{(\alpha_{t_{1,i}})^2 + (\alpha_{t_{2,i}})^2} \leq \mu \alpha_{n_i}, \ \alpha_{n_i} \geq 0 \right\}, \quad (2.2)$$

where μ is the friction coefficient.

Let's now consider the complete robotic hand composed by N fingers. The whole contact force vector $f^c \in \mathbb{R}^{3n}$ is obtained by considering all contact forces of the fingers expressed with respect to their local contact frames

$$f^c = ((f_1^c)^T \dots (f_i^c)^T \dots (f_n^c)^T)^T. \quad (2.3)$$

In order to determine the resultant wrench w applied to the object and generated by the contact forces, it is necessary to apply a change of coordinates so that to express the complete force vector f^c from the local reference frame \mathcal{C}_i to the reference frame \mathcal{O} centered at the object center of mass. This change of coordinates is given by the *grasp map* $G \in \mathbb{R}^{6 \times 3n}$ [52]

$$w = Gf^c. \quad (2.4)$$

A stable grasp for an object is granted only if the resultant wrench w can balance an external wrench applied on the object w_{ext} . This means that, in order to guarantee that the grasped object is in equilibrium

$$-w_{ext} = Gf^c. \quad (2.5)$$

We can notice that, if the number on finger is greater than two, the dimension of vector $f^c \in \mathbb{R}^{3n}$ is greater than the dimension of $w_{ext} \in \mathbb{R}^6$. This implies that $G \in \mathbb{R}^{6 \times 3n}$ is a rectangular matrix. In this case, the solution of Equation (2.5) is

$$f^c = f^p + f^h, \quad (2.6)$$

where f^p and f^h are two orthogonal vectors. More precisely:

- f^p is called *particular solution* and is the component of the contact force f^c that is required to balance the external wrench. This means that

$$f^p = -G^+w_{ext}, \quad (2.7)$$

where G^+ is the pseudo-inverse of G .

- f^h is called *homogeneous solution*, or *internal force vector*, and is the component of the contact force f^c that is required to guarantee that $f_i^c \in F_i$. This means that

$$f^h = Nh, \quad (2.8)$$

where $N \in \mathbb{R}^{3n \times (3n-6)}$ is an orthonormal basis of the null space of G , and $h \in \mathbb{R}^{(3n-6)}$ describes the components of f^h with respect to N .

Since for a given grasp configuration and a given external wrench f^p is fixed, the force distribution problem is equivalent to the determination of the vector h . The grasping force distribution can be studied as the minimization problem of the internal force vector and, in most cases, this problem has not an unique solution.

2.2.2 Non-Linear Programming Optimization Background

Non-linear optimization and mathematical programming have been deeply addressed in the literature, see [43] among others.

A non-linear optimization problem with inequality constraints is basically the minimization of an objective function $Z(x)$, in which $x \in \mathbb{R}^n$ is the vector of the unknowns, so that a certain p -dimensional set of inequalities constraints g , eventually non-linear, has to be granted so that the solution of the problem belongs to a set Ω of valid solutions. In particular, the optimization problem is mathematically expressed by

$$\text{Min} \quad Z(x) \quad (2.9)$$

$$\text{subject to} \quad g(x) \leq 0. \quad (2.10)$$

In a constrained optimization problem, two kinds of solutions are possible: a *local minimum* and a *global minimum*, which are defined as follows.

Definition 2.2.1. A vector $x^* \in \Omega$ is said to be a local (relative) minimum of $Z(x)$ over Ω , if there is an $\varepsilon \geq 0$ such that $Z(x) \geq Z(x^*), \forall x \in \Omega$ within $|x - x^*| \leq \varepsilon$. If $Z(x) > Z(x^*), \forall x \in \Omega, x \neq x^*$ and $|x - x^*| \leq \varepsilon$, then x^* is said to be a strict local (relative) minimum of $Z(x)$ over Ω .

Definition 2.2.2. A vector $x^* \in \Omega$ is said to be a global minimum of $Z(x)$ over Ω , if $Z(x) \geq Z(x^*), \forall x \in \Omega$. If $Z(x) > Z(x^*), \forall x \in \Omega$ and $x \neq x^*$, then x^* is said to be a strict global minimum of $Z(x)$ over Ω .

In optimization problems without constraints, the gradient vector of the objective function is null at the local and global minima, while, if the objective function is subjected to constraints, this is not always guaranteed.

Any constrained optimization problem can be transformed into a system of equations thanks to the Kuhn-Tucker conditions, that are based on the concept of the *Lagrange multipliers vector*. Before introducing the Kuhn-Tucker conditions, it is necessary to give the two following definitions of *active constraints* and *regular vector*.

Definition 2.2.3. Let x be a vector that satisfies Equation (2.10). An inequality constraint $g_j(x) \leq 0$ is said active if $g_j(x) = 0$ and inactive if $g_j(x) < 0$.

Definition 2.2.4. Let x be a vector satisfying Equation (2.10) and let J be the set of indexes j of the active constraints. In this case, x is said to be a regular vector for the constraints if the gradient vectors $\nabla g_j(x), j \in J$ are linearly independent.

Theorem 2.2.5. (Kuhn-Tucker Conditions or First-Order Necessary Conditions). Let x^* be a local minimum of the optimization problem described by Equations (2.9) and (2.10) and suppose that x^* is regular for the constraints. Then, there exists a Lagrange multipliers vector λ such that

$$\nabla Z(x^*) + \lambda^T \nabla g(x^*) = 0 \quad (2.11)$$

$$\lambda g(x^*) = 0 \quad (2.12)$$

$$\lambda \geq 0. \quad (2.13)$$

Note that Equation (2.12) determines the active constraints of the optimization problem, since $\lambda_j \neq 0$ implies $g_j = 0$. Then, a strictly positive Lagrange multiplier is associated to the active constraints.

Since the Kuhn-Tucker conditions are necessary conditions, any local minimum of the optimization problem has to satisfy them. For a strict local minimum, the Kuhn-Tucker conditions are also sufficient if the following theorem is verified.

Theorem 2.2.6. (Second-Order Sufficient Condition) A vector x^* satisfying the Kuhn-Tucker conditions is a strict local minimum of the problem described by Equations (2.9) and (2.10) if the Hessian matrix

$$H(x^*) = \nabla^2 Z(x^*) + \lambda^T \nabla^2 g(x^*) \quad (2.14)$$

is positive definite on the subspace

$$M' = \{y \mid \nabla g_j(x^*)y = 0 \forall j \in J\}, \quad (2.15)$$

where

$$J = \{j \mid \nabla g_j(x^*)y = 0, \lambda_j > 0\}. \quad (2.16)$$

Although the Kuhn-Tucker conditions transform any non-linear optimization problem in a system of equations, it may be difficult to solve the obtained system. Moreover, since the identification of the active constraints is a combinatorial problem, several systems of equations, each one referred to a different active constraint, have to be solved.

In the literature, there are different methods which can be applied for the determination of x^* . These methods are called *primal methods*, since the original (*primal*) form of the optimization problem is considered. Among the most used methods, we can mention the *feasible direction*, the *gradient projection* or the *penalty and barrier methods*. These methods involve issues such as the computation of a feasible initial solution, of a search direction and a step size.

In the literature, other kind of methods, called *dual methods*, are used for the determination of the optimal solution.

The dual form of an optimization problem transforms the original problem into an equivalent one in which the fundamental unknowns are the Lagrange multipliers. Once the Lagrange multipliers are known, the determination of the final solution is simple. The methods based on the dual form are applicable only to a subclass of non-linear optimization problems, i.e. convex problems. The dual methods are based on the following theorem.

Theorem 2.2.7. (Duality Theorem) *Let x^* be a relative local minimum of the optimization problem described by Equations (2.9) and (2.10), and let λ^* be the corresponding Lagrange multipliers vector. Suppose also that x^* is regular and that the Hessian matrix $H(x^*)$ is positive definite. Then, the dual problem*

$$\text{Max} \quad Z_d(\lambda) = \min[Z(x) + \lambda^T g(x)] \quad (2.17)$$

$$\text{subject to} \quad \lambda \geq 0 \quad (2.18)$$

has a local maximum at λ^* with corresponding value x^* .

Note that the Duality Theorem can be applied only when the problem is convex, i.e. $H(x^*)$ is positive definite.

The same techniques used to solve the primal form can also be applied for the dual form of the optimization problem. Note that the constraints of the dual form are easier than the constraints of the primal form, therefore, in general, solving the dual form of the problem is easier than the primal form.

2.3 Optimization Problem Model

In this Section, we apply the mathematical programming background presented in Section 2.2 in order to propose a solution to the optimal force distribution problem. In particular, the solution we are proposing is based on the minimization of the contact forces so that to guarantee the equilibrium of the grasped object by balancing a generic external wrench exerted on it.

The minimization of the contact forces can be expressed as a constrained optimization problem in which the objective function to be minimized is the module of the contact force vector and the constraints are the object equilibrium and the friction model. The optimization problem is

$$\text{Min} \quad Z(h) = \|f^c\| \quad (2.19)$$

$$\text{subject to} \quad Gf^c - w_{ext} = 0 \quad (2.20)$$

$$\sqrt{(\alpha_{t1,i})^2 + (\alpha_{t2,i})^2} - \mu\alpha_{n_i} \leq 0 \quad \text{with } i = 1, \dots, N \quad (2.21)$$

where f^c , α_{n_i} , $\alpha_{t1,i}$ and $\alpha_{t2,i}$ are functions of the sought h , i.e. the component of the internal force vector f^h expressed with respect to the basis N .

2.3.1 Model Simplification

The initial form of the optimization problem expressed in Equations 2.19, 2.20 and 2.21 is mathematically manipulated in the following part of the section so that to obtain a simplified form in order to apply non-linear programming techniques and to solve it.

Proposition 2.3.1. (Equivalent Objective Function) *The minimization of the module of the contact force vector $\|f^c\|$ is equivalent to the minimization of the square of the module of the internal force vector f^h expressed with respect to the orthonormal basis N , $\|h\|^2$.*

Sketch of Proof: Let's consider the objective function $Z = \|f^c\|$. By substituting Equation (2.6), the objective function is given by

$$Z = \|f^c\| = \sqrt{(f^p)^T f^p + (f^h)^T f^h + 2(f^p)^T f^h}. \quad (2.22)$$

This expression can be simplified by observing that the square root required to compute Z does not affect the result of the minimization problem. Moreover, since f^p and f^h are two orthogonal vectors, $(f^p)^T f^h = 0$; finally, since $(f^p)^T f^p$ is a constant value, it can be removed from the minimization problem. Therefore, Z is reduced to:

$$Z = (f^h)^T f^h. \quad (2.23)$$

By substituting Equation (2.8), the objective function is given by

$$Z = \sum_{i,j=1}^{3n} (\text{Col}_i(N))^T \text{Col}_j(N) h_i h_j, \quad (2.24)$$

where $\text{Col}_i(N)$ and $\text{Col}_j(N)$ are the i and j columns, respectively, of N .

Since N is an orthonormal basis, the module of each column is 1, and any product between two of them is 0:

$$(\text{Col}_i(N))^T \text{Col}_j(N) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \quad (2.25)$$

It follows that Equation (2.24) can be simplified in

$$Z = \sum_{i=1}^{3n} h_i^2 = \|h\|^2, \quad (2.26)$$

which implies that the minimization of $\|h\|^2$ is equivalent to minimize the module of the contact force vector $\|f^c\|$.

In order to further simplify the optimization model, we proceed with the linearization of the constraint in Equation (2.21). In particular, the friction cone represented by this equation is approximated with a pyramid of m faces. The base of the pyramid is considered at a distance α_{n_i} from the contact point, so that the base of the pyramid is described by a polygon of m vertexes inscribed in a circumference of radius $\mu\alpha_{n_i}$. Therefore, the relation between the area of the polyhedron and the area of the circumference can be used as a measure of the accuracy of the approximation.

Let v_k , $k = 1, \dots, m$, be the vertexes of the base of the pyramid. Since these vertexes belong to the circumference of radius $\mu\alpha_{n_i}$, they are described by:

$$v_k = \mu\alpha_{n_i}(\cos \phi_k, \sin \phi_k), \quad \phi_k = k \frac{2\pi}{m} \quad (2.27)$$

Furthermore, since $\alpha_{n_i} \geq 0$, for any two consecutive vertexes v_k and v_{k+1} , the values of $\alpha_{t_{1,i}}$ and $\alpha_{t_{2,i}}$ define a point inside the base of the pyramid if the following relation is satisfied:

$$-\mu(\sin \phi_{k+1} \cos \phi_k - \sin \phi_k \cos \phi_{k+1})\alpha_{n_i} + (\sin \phi_k - \sin \phi_{k+1})\alpha_{t_{1,i}} + (\cos \phi_{k+1} - \cos \phi_k)\alpha_{t_{2,i}} \leq 0 \quad (2.28)$$

By using Equation (2.3), Equation (2.28) can be expressed in a matricial form as:

$$A_i^c f_i^c \leq 0, \quad A_i^c \in \mathbb{R}^{m \times 3} \quad (2.29)$$

where

$$A_i^c = \begin{pmatrix} -\mu(\sin \phi_2 \cos \phi_1 - \sin \phi_1 \cos \phi_2) & (\sin \phi_1 - \sin \phi_2) & (\cos \phi_2 - \cos \phi_1) \\ \vdots & \vdots & \vdots \\ -\mu(\sin \phi_m \cos \phi_{m-1} - \sin \phi_{m-1} \cos \phi_m) & (\sin \phi_{m-1} - \sin \phi_m) & (\cos \phi_m - \cos \phi_{m-1}) \\ -\mu(\sin \phi_1 \cos \phi_m - \sin \phi_m \cos \phi_1) & (\sin \phi_m - \sin \phi_1) & (\cos \phi_1 - \cos \phi_m) \end{pmatrix} \quad (2.30)$$

(note that the last row takes into account that the first and the last vertexes are also consecutive).

The fulfillment of Equation (2.29) implies that the finger forces lie within the pyramid used to linearize the friction cone. Applying the same procedure for the N fingers and staking the results in a single matrix,

$$A^c f^c \leq 0 \quad (2.31)$$

where

$$A^c = \begin{pmatrix} A_1^c & 0 & \dots & 0 \\ 0 & A_2^c & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & A_n^c \end{pmatrix} \in \mathbb{R}^{nm \times 3n}. \quad (2.32)$$

In order to obtain the constraints as a function of h , we substitute Equations (2.6) and (2.7) in Equation (2.31):

$$A^c N h + A^c f^p \leq 0. \quad (2.33)$$

Finally, the problem of minimizing the grasping forces can be expressed with the following minimization problem:

$$\text{Min} \quad Z(h) = \|h\|^2 \quad (2.34)$$

$$\text{subject to} \quad Ah + b \leq 0, \quad (2.35)$$

where

$$A = A^c N, \quad (2.36)$$

$$b = A^c f^p. \quad (2.37)$$

2.3.2 Convexity Analysis

Following the discussion in Section 2.2.2, we have to verify that the minimization problem defined by Equations (2.34) is convex.

By using Equation (2.14), the Hessian matrix H associated to the optimization problem is given by

$$H(h) = \nabla^2 \|h\|^2 + \lambda \nabla^2 (Ah + b). \quad (2.38)$$

Operating adequately, $H(h)$ can be reduced to the following expression:

$$H(h) = 2I_n, \quad (2.39)$$

where I_n is the N -identity matrix.

Let h^* be the optimal solution of the minimization problem and let λ^* be the Lagrange multipliers vector associated to h^* . Since $H(h)$ is constant and positive definite, the convexity of the problem is always assured. This implies that the following properties of h^* and λ^* are also satisfied:

- From Theorem 2.16, h^* is the strictly global minimum of the minimization problem defined by Equations (2.34) and (2.35). Then, h^* is the only value of h that satisfies the Kuhn-Tucker conditions stated in Theorem 2.2.5.
- From Theorem 2.2.7, λ^* is the optimal solution of the dual form of the minimization problem. Then, the methods based on the dual form are also valid to obtain the optimal solution h^* (once λ^* is obtained, h^* can be easily determined).

2.3.3 Primal Form

Since the convexity of the minimization problem is assured, the global minimum h^* satisfies the Kuhn-Tucker conditions:

$$2I_n h^* + A^T \lambda^* = 0 \quad (2.40)$$

$$\lambda^* (Ah^* + b) = 0 \quad (2.41)$$

$$\lambda^* \geq 0 \quad (2.42)$$

In this set of equations, the number of constraints (Nm) is larger than the number of variables (N). This implies that, in order to have a regular solution, the maximum number of active constraints has to be N . Therefore, at least $Nm - N$ Lagrange multipliers are equal to zero. The determination of the minimum contact forces using the Kuhn-Tucker

conditions represents a combinatorial problem with the maximum number of combinations bounded by

$$\binom{Nm}{N} = \frac{(Nm)!}{(Nm-N)!N!} \quad (2.43)$$

Each combination implies to solve a N-linear system of equations. Note that the number of combinations increases exponentially with respect to the number of faces used to linearize the friction cones. Although the Kuhn-Tucker conditions in their pure form can be used when the friction cones are linearized with a low number of faces, it should be used jointly with another method to improve the convergence (feasible direction, gradient projection or penalty and the barrier methods).

2.3.4 Dual Form

Since the convexity of the minimization problem is assured, Theorem 2.2.7 can be applied. This implies that the solution of the minimization problem defined by Equations (2.34) and (2.35) is also the solution of the following maximization problem:

$$\text{Max} \quad Z_d(\lambda) = \lambda^T A_d \lambda + \lambda^T b \quad (2.44)$$

$$\text{subject to} \quad \lambda \geq 0, \quad (2.45)$$

where $A_d \in \mathbb{R}^{N \times Nm}$ is defined by:

$$A_d(i, j) = \begin{cases} -\frac{1}{4} \sum_{k=1}^n [A(i, k)]^2 & \text{if } i = j \\ -\frac{1}{4} \sum_{k=1}^n [A(i, k)][A(j, k)] & \text{if } i \neq j \end{cases} \quad (2.46)$$

2.3.5 Feasible Regions Method

The maximization problem defined by Equations (2.44) and (2.45) can be solved by using an algorithm based on the feasible regions method. This method reaches the optimal value of the objective function applying iteratively the following equation:

$$\lambda^{(k+1)} = \lambda^{(k)} + \delta^{(k)} d^{(k)}, \quad (2.47)$$

where k is the step number, $d^{(k)}$ is the search direction and $\delta^{(k)}$ is the size of the step in this direction. In order to reach the maximum of the objective function, $d^{(k)}$ and $\delta^{(k)}$ have to be chosen such that $Z_d(\lambda^{(k+1)}) > Z_d(\lambda^{(k)})$ with $\lambda^{(k+1)} \geq 0$.

Any algorithm based on feasible regions method has four main issues to be determined: an initial feasible solution $\lambda^{(1)}$, a search direction $d^{(k)}$, a step size $\delta^{(k)}$ and a stop condition. Using the dual form defined by Equations (2.44) and (2.45) these aspects are solved as follows:

Initial solution: A good initial solution is the trivial solution $\lambda^{(1)} = 0$ since it is inside the feasible region defined by the constraints and since at least $Nm - N$ Lagrange multipliers are equal to zero for a regular solution of the primal form.

Search direction: Let $d^{(k)} = [d_1^{(k)} \dots d_i^{(k)} \dots d_{nm}^{(k)}]^T$ be the direction in which $\lambda^{(k+1)}$ is determined. In a typical gradient flow algorithm, this direction is determined by the gradient of the objective function. Since in the optimization problem tackled here $\lambda \geq 0$, the determination of $d^{(k)}$ is slightly modified as follows:

1. Compute $d^{(k)} = \nabla Z_d(\lambda^{(k)})$, the following result is obtained:

$$d^{(k)} = 2A_d\lambda^{(k)} + b, \quad (2.48)$$

2. If $\lambda_i^{(k)} = 0$, then $d_i^{(k)} = \max\{0, d_i^{(k)}\}$ for $i = 1, \dots, nm$ (i.e., the search direction can not be negative when $\lambda_i^{(k)} = 0$, otherwise $\lambda_i^{(k+1)}$ would be negative).

Step size: In order to increase the convergence of the algorithm, the step size $\delta^{(k)}$ is determined solving the following maximization problem:

$$\text{Max} \quad Z_d(\lambda^{(k)} + \delta^{(k)}d^{(k)}) \quad (2.49)$$

$$\text{subject to} \quad \lambda^{(k)} + \delta^{(k)}d^{(k)} \geq 0. \quad (2.50)$$

This problem can be solved with the following steps:

1. Initialize $\delta^{(k)}$ with the value that maximizes $Z_d(\lambda^{(k)} + \delta d^{(k)})$. This value can be analytically determined as:

$$\delta^{(k)} = -\frac{2(d^{(k)})^T A_d d^{(k)} + (d^{(k)})^T b}{2(d^{(k)})^T A_d d^{(k)}}, \quad (2.51)$$

2. Check whether $\lambda^{(k)} + \delta^{(k)}d^{(k)} \geq 0$.
 - (a) If this condition is satisfied, then $\delta^{(k)}$ is the value determined by Equation (2.51).
 - (b) Else, $\delta^{(k)}$ is the minimum positive value so that $\lambda^{(k)} + \delta^{(k)}d^{(k)} = 0$.

Stop condition: When $d^{(k)} = 0$, $\lambda^{(k+1)} = \lambda^{(k)}$ and the objective function is not further maximized, obtaining the optimal value. Let ε be a tolerance parameter on the final value of $d^{(k)}$. The optimal value is reached when $\|d^{(k)}\| < \varepsilon$.

As a result, the following algorithm is proposed to solve the dual form of the optimization problem.

Once λ^* is obtained, the optimal solution of the primal problem h^* can be easily determined from Equation (2.40)

$$h^* = -\frac{1}{2}A_d^T\lambda^*. \quad (2.53)$$

Then, the internal force vector and the minimum contact forces are determined from Equations (2.8) and (2.6), respectively.

Remark 2.3.2. *The proposed methodology, using the dual form of the optimization problem, provides some useful mathematical advantages with respect to the methods based on the primal form. Nevertheless, this formulation has a lack of physical meaning and the accomplishment of the constraints of the dual problem does not mean the accomplishment of the friction constraints. Only in the optimum case it is possible to assure that the finger forces lie inside the friction cones. Figure 2.2 schematizes the relation between the primal and the dual form.*

Algorithm 1 Determination of the minimum contact forces

The maximization problem defined by Equations (2.44) and (2.45) is solved with the following steps:

Step 1 Chose the tolerance parameter $\varepsilon > 0$, initialize $k = 1$ and $\lambda^{(k)} = 0$ (i.e. the initial feasible solution).

Step 2 Compute the coefficient matrix A_d and the vector b (by using Equations (2.46) and (2.37), respectively) of the dual objective function described by Equation (2.44).

Step 3 Determine the search direction $d^{(k)}$ and the step size $\delta^{(k)}$, as described above.

Step 4 Update the Lagrange multipliers vector:

$$\lambda^{(k+1)} = \lambda^{(k)} + \delta d^{(k)} \quad (2.52)$$

Step 5 Compute $\|d^{(k)}\|$:

- (a) If $\|d^{(k)}\| < \varepsilon$, then $\lambda^* = \lambda^{(k+1)}$ is the optimal solution of the dual problem.
- (b) Else $k = k + 1$ and go to Step 3.

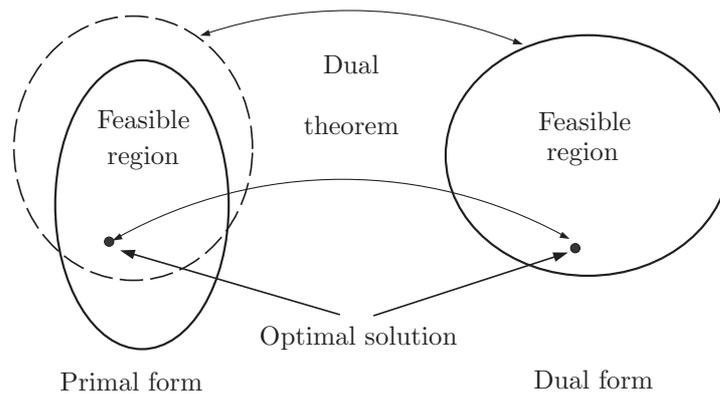


Figure 2.2: Relations between the primal and the dual forms: the optimal solutions are equivalent while the feasible regions are not.

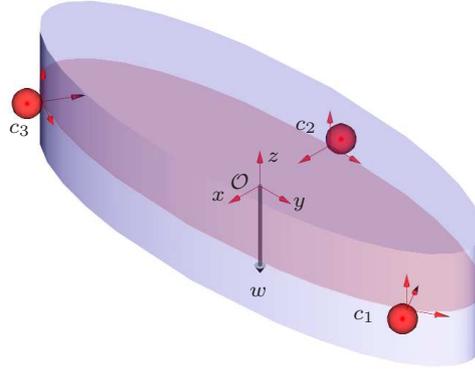


Figure 2.3: Object, contact points and section defined by these points used in the examples with different approximations of the friction cones.

2.4 Examples

This section presents some simulation results of the proposed force distribution methodology. The grasping forces are determined for given contact point positions and for a given external wrench (w_{ext}) exerted on the object. The methodology has been implemented using Matlab 6.5 in a Pentium Centrino at $1.6GHz$. Therefore, the code is not optimal in terms of efficiency and the computational times included in the examples can only be considered as qualitative values.

Figure 2.3 shows the object and the three grasping points used in the examples. The grasp map G determined by these contact points is:

$$G = \begin{pmatrix} -0.86 & 0.50 & 0 & 1 & 0 & 0 & -0.86 & 0.50 & 0 \\ -0.50 & -0.86 & 0 & 0 & 1 & 0 & 0.50 & 0.86 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 2.59 & 0 & 0 & 0 & 0 & 0 & -2.59 \\ 0 & 0 & -0.50 & 0 & 0 & 1 & 0 & 0 & -0.50 \\ 2 & -1.73 & 0 & 0 & -1 & 0 & -2 & 1.73 & 0 \end{pmatrix} \quad (2.54)$$

Tables 2.1 and 2.2 show the results of applying Algorithm 1 considering two different external wrenches w_{ext} and m number of faces in the approximation of the friction cones. In the first table only the weight, w , of the object is considered, while in the second one an external wrench with non-null components in each direction of the object reference frame acts on the object.

Figure 2.4 shows another example where the object is rotated clockwise 2π radians with respect to the z -axis keeping the same contact points (Figure 2.4a shows the initial position of the object). The forces are computed while the object is being rotated since the direction of the weight with respect to the contact points varies (it is considered that the object is rotated with a constant low velocity and there are not inertial moments). The movement has been discretized with 100 sampled points, which implies that the forces are recomputed at each 0.0628 radians. In order to improve the efficiency of the gradient method, λ is initialized with the solution of the previous sampled point instead of the null vector. The example has been done considering $m = 12$ (Figure 2.4b). The average and maximum number of iterations and computational times are summarized in Table 2.3. In

Table 2.1: Determination of the minimum contact forces for different approximations of the friction cones and considering the weight of the object

$w_{ext} = w$	$[0 \ 0 \ -1 \ 0 \ 0 \ 0]^T$			
m	4	8	12	20
f_1^c	$\begin{bmatrix} 1.1111 \\ 0.0000 \\ 0.3333 \end{bmatrix}$			
f_2^c	$\begin{bmatrix} 1.9245 \\ 0.0000 \\ 0.3333 \end{bmatrix}$			
f_3^c	$\begin{bmatrix} 1.1111 \\ 0.0000 \\ 0.3333 \end{bmatrix}$			
$\ f^c\ $	2.5507	2.5507	2.5507	2.5507
Number of iterations	143	32	55	72
Computational time (seconds)	0.0200	0.0100	0.0100	0.0100

Table 2.2: Determination of the minimum contact forces for different approximations of the friction cones and considering an external wrench with non-null components in each direction

w_{ext}	$[1 \ -2 \ 5 \ -4 \ 1 \ 2]^T$			
m	4	8	12	20
f_1^c	$\begin{bmatrix} 5.2555 \\ -1.0131 \\ -0.5635 \end{bmatrix}$	$\begin{bmatrix} 4.8792 \\ -1.2303 \\ -0.5635 \end{bmatrix}$	$\begin{bmatrix} 4.7853 \\ -1.2846 \\ -0.5635 \end{bmatrix}$	$\begin{bmatrix} 4.2417 \\ -1.1234 \\ -0.5635 \end{bmatrix}$
f_2^c	$\begin{bmatrix} 10.1291 \\ 0.2452 \\ -2.3333 \end{bmatrix}$	$\begin{bmatrix} 9.9119 \\ -0.1310 \\ -2.3333 \end{bmatrix}$	$\begin{bmatrix} 9.8576 \\ -0.2250 \\ -2.3333 \end{bmatrix}$	$\begin{bmatrix} 9.2770 \\ -1.1292 \\ -2.3333 \end{bmatrix}$
f_3^c	$\begin{bmatrix} 7.0104 \\ 0.0000 \\ -2.1031 \end{bmatrix}$	$\begin{bmatrix} 7.0104 \\ 0.0000 \\ -2.1031 \end{bmatrix}$	$\begin{bmatrix} 7.0104 \\ 0.0000 \\ -2.1031 \end{bmatrix}$	$\begin{bmatrix} 7.3711 \\ 0.6833 \\ -2.1031 \end{bmatrix}$
$\ f^c\ $	13.8071	13.5246	13.4575	13.0988
Number of iterations	276	148	243	162
Computational time (seconds)	0.0400	0.0300	0.0500	0.030

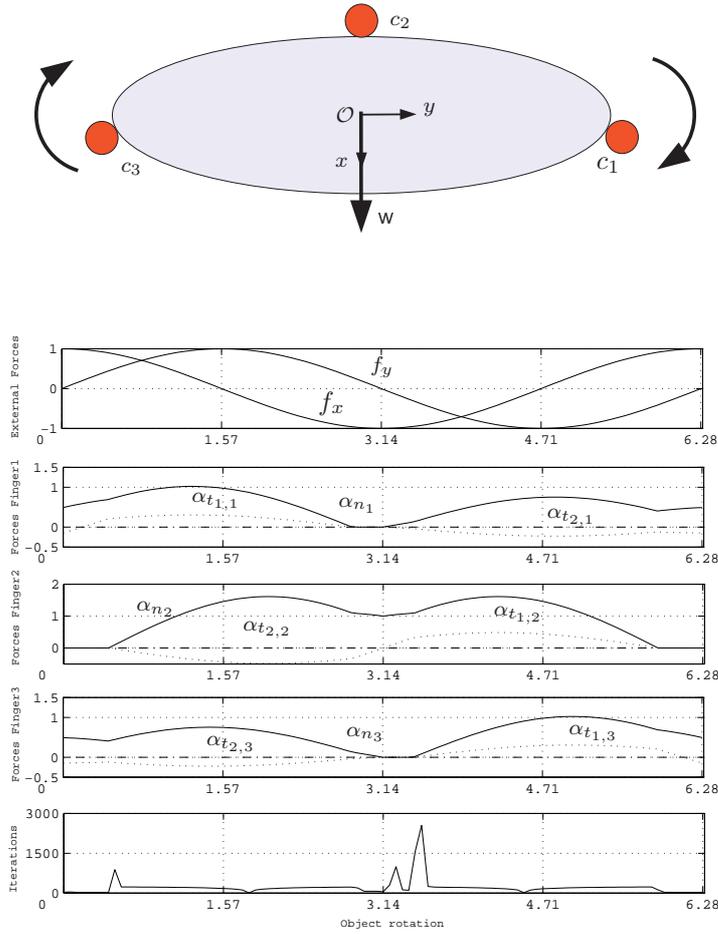


Figure 2.4: Evolution of the contact forces during the rotation of the object: initial position of the object and forces obtained with $m = 12$.

this case, the differences between the results obtained with the two approximations are very small.

The results obtained with the different approximations of the friction cones indicate that there is not a direct relation between the performance of Algorithm 1 and the number of faces m used in the approximation when $4 \leq m \leq 20$. For higher values of m the computational cost also increases, although it is difficult to mathematically determine this relation and the improvement of the minimum grasping forces is very small. In general, a good trade-off between performance and accuracy in the final result is obtained when $12 \leq m \leq 20$. Other examples of the proposed methodology considering several objects are present in [16].

2.5 Conclusions

A new mathematical approach to solve the force distribution problem in a grasp has been presented. This approach is based on the dual theorem of non-linear programming,

Table 2.3: Number of iterations and computational times in the rotation of the ellipse.

m		12	20
Number of iterations	Maximum	4374	4022
	Average	104	135
Computational time (seconds)	Maximum	0.8512	0.8013
	Average	0.0219	0.0317

which can only be applied when the convexity of the problem is assured. By adequately modeling the problem and applying the dual theorem, the original problem is transformed to another one much easier to solve. The examples show the efficiency of the proposed methodology. Even when the code can not be considered optimal in terms of efficiency, the provided computational times are of the order or even smaller than those of some of the most popular algorithms described by [40].

The L_2 norm of the contact force vector has been taken as objective function. It should be notice that other quadratic norms can also be used, since in this case the Hessian matrix is always convex and therefore the convexity is still guaranteed.

The determination of the contact forces has been done linearizing the friction constraints. The development of a similar approach using the original problem with non-linear constraints will considered in future works. In this case, it is necessary to study the convexity of the problem in order to assure that the dual theorem can also be applied in this case, and the optimal solution of the dual form corresponds to the optimal solution of the original problem.

Chapter 3

Vision-Based Grasp Tracking

In robotics, the manipulation of *a priori* unknown objects involves several steps and problems that must be carefully considered and solved by proper planning and control algorithms. For example, once suitable contact points have been computed, the control system should be able to track them in the approach phase, i.e. while the relative position/orientation of the object and the gripper of the robotic system change due to the approaching movement of the robot towards the object itself. In this Chapter, we discuss a practical method for the tracking of grasp points in image space that is based on transferring previously computed grasp points from an initial image to subsequent ones and on the analysis of the new grasp configuration. In order to obtain an efficient visually guided grasping, three basic techniques are used together. In particular, two of them are based on a grasp description that is invariant with respect to the relative movement between two object views, one of these strategies being used jointly with an object tracking method. The third proposed alternative is based on a homography computed between two object views.

3.1 Introduction

A fundamental requirement for a vision-based robotic system is the capability of detecting image features in order to control the pose of the end-effector with regard to a set of target features. In this Chapter, we analyze the problem of grasping unknown planar objects by using an eye-in-hand system acting in an unknown dynamic environment. The visual servoing control loop should be able to trace significant points on the object, such as the grasp points, in order to achieve the desired position of the robotic gripper with respect to the target object.

As shown in Figure 3.1, the proposed image-based visual servoing procedure requires three functional blocks to be executed in real time: the extraction of a set of features, i.e.

the contour of the object, the selection of a stable grasp configuration and its tracking along different views of the object, and the control law whose set points are the grasp points. In particular, in this Chapter we focus on the design of the grasp tracking module.

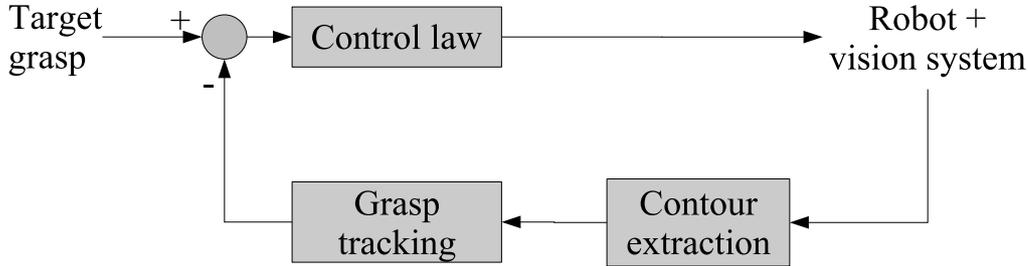


Figure 3.1: A grasp-based positioning movement.

In order to allow the grasp tracking, it is necessary to find a representation of the coordinates of the grasp points that is invariant in the image space according to different movement of the robotic arm. In particular, three different representations are presented: two are invariant with respect to 4 d.o.f. (translations along x , y , z and rotation about z) [64], the remaining one with respect to 6 d.o.f., [10]. Consequently, we have to use two different control laws: one that bounds the movement of the robotic arm to 4 d.o.f., the other to 6 d.o.f.

The vision system is considered uncalibrated so that the internal camera parameters are not known or used. The proposed strategy allows both to transpose the grasp points along a sequence of images and to find correspondences between pairs of stereo images [64]. The object is not known *a priori* and no model of it is available. It is assumed to be rigid and planar: ideally, it is a shape that lies on a given plane, so it is considered relatively flat [50]. The tests have been performed considering only the external contour of the object, ignoring possible holes. Nevertheless, the grasp descriptions considered can also be applied to objects with holes.

The concepts presented in this Chapter have been applied in two different laboratory setups, shown in Figure 3.2: the Robotic Intelligence Lab at the University Jaume I in Castellón (Spain) and the Laboratory of Automation and Robotics at the University of Bologna (Italy).

3.2 Grasp Tracking

The *grasp tracking* module of Figure 3.1 can be subdivided as shown in Figure 3.3, and can be implemented as outlined in the proposed algorithm 2.

During the approaching movement of the robotic arm towards the object to be grasped, a sequence of contours is acquired in real time by the vision system. At the beginning of this sequence, as no grasp is available, an initial stable grasp is computed in the first contours (*grasp synthesis*) [18, 45, 50, 61, 64]. Then, for each new incoming contour, the algorithm tries to apply a previously computed grasp to the current image. This process is called here *grasp translation*. This involves a translation of that grasp configuration to the current image and an evaluation of the stability of the translated grasp, i.e. a *grasp analysis* [61, 62].

In order to evaluate the stability of the new grasp, proper criteria must be used, and the new grasp should be discharged if these criteria fails. In particular, due to noise and

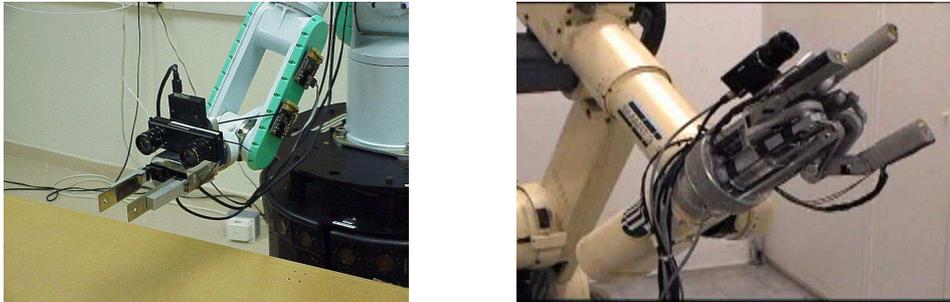


Figure 3.2: Experimental setups for tracking a grasp.

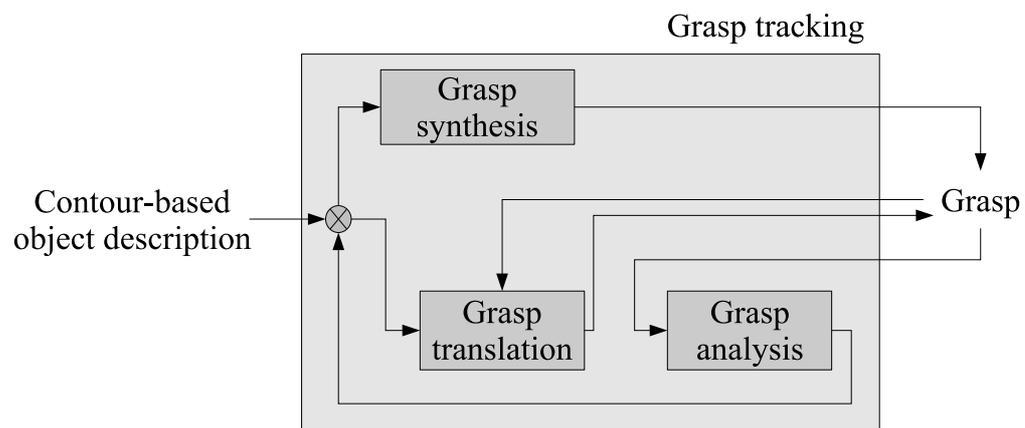


Figure 3.3: Grasp tracking procedure.

data extraction errors, three different criteria have been applied in parallel [61, 62], and a grasp is disregarded if two or more of them fail for more than two consecutive images. In this case, a new grasp configuration must be evaluated.

Algorithm 2 Grasp tracking

```

if no previously computed grasp available then
  search for a new grasp (grasp synthesis);
else
  translate grasp from reference view to current object view (grasp translation);
  evaluate translated grasp (grasp analysis);
  if negative grasp evaluation then
    search for a new grasp (grasp synthesis);
  end if
end if

```

In this Chapter we focus on three different grasp translation strategies, which are described in Section 3.4. They are based on the description of the grasp using features that are invariant with respect to the different views of the object acquired during the motion.

3.3 Invariants for Grasp Translation

In order to carry out the visual translation, the target object is to be described in term of invariants, i.e. properties that remain unchanged under appropriate classes of transformations. This Section describes the different sets of invariants considered for the grasp translation, and their relationship with the grasp description.

3.3.1 Object and Grasp Descriptions

As the grasp points belong to the contour of the object, a mechanism to describe the contour is required. A generic object consists of a list of internal and external contours each of which is considered as a list of points. Two alternatives have been considered for indexing these points:

- *Location coordinates*, see Figure 3.4 (left). The location coordinates correspond to a couple of indexes:
 - c : the index of the contour, within the list of internal/external contours, the point belongs to;
 - p : the index of the point, within the complete list of the points belonging to contour c .
- *RRP coordinates (Relative Reference Points)*, see Figure 3.4 (right). The contour is considered as a polygon. The vertexes of this polygon are named *reference points* and the contour points that lie on the segment joining the reference points are the *interpolated points*. The RRP coordinates include three indexes:
 - c : the index of the contour;
 - ref : the index of the reference point, within the list of reference points corresponding to contour c ;

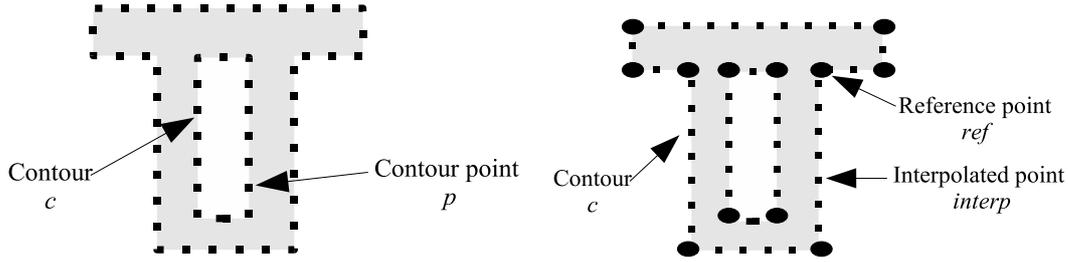


Figure 3.4: Location (left) and RRP (right) coordinates of contour points.

- *interp*: the index of the point, interpolated between reference point *ref* and the following reference point.

A grasp description consists of a collection of points, called *grasping points* [35, 65] or *contact points* [21, 58], and depends on the geometry of the gripper and on the particular manipulation task. Each of them identifies the center of the region of the object contour in which the robot fingers should be placed. The description proposed includes:

- Coordinates of the grasp points. Three alternatives have been considered:
 - Location coordinates. For the *i*-th grasp point, location coordinates are made of a couple (c_i, p_i) ;
 - RRP coordinates. For the *i*-th grasp point, the RRP coordinates are made of a triple $(c_i, ref_i, interp_i)$;
 - Relative coordinates (only for antipodal grasps). This description is based on the specification of the relative location of the grasp points and the grasp line (the line joining the two grasp points) with respect to a reference frame in which the axes are the minimum (I_{min}) and maximum (I_{max}) inertia axes of the silhouette of the object and the center is the centroid (p_c), as illustrated in Figure 3.5 [63]. This grasp description is composed of:
 - * d_c : distance of the intersection between the grasp line and I_{min} , from the centroid of the object along the direction vector of the I_{min} axis;
 - * α_m : angle between the grasp line and the I_{min} axis, measured from I_{min} towards the grasp line;
 - * d_i : distance, along the direction vector of the grasp line, between one of the two grasp points, p_{g_i} , and the intersection between the grasp line and I_{min} .

Distances are measured in pixels and, to ensure invariance with respect to changes of scale, are normalized using the area of the object. Angles are measured in degrees. The computation of I_{min} , I_{max} , p_c and other object features from second-order moments is described in other works [25].

- Type of grasp:
 - *Squeezing grasp*, executed by closing the gripper fingers around the object;
 - *Expansion grasp*, executed by opening the gripper fingers.
- An index selecting the object to be grasped (in case of multiple objects in the image).

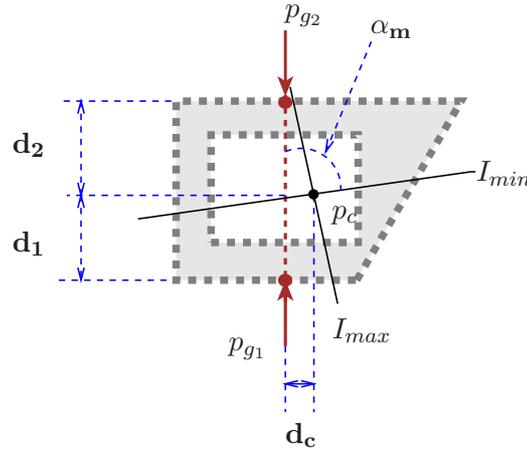


Figure 3.5: Grasp description using relative coordinates.

- An index selecting the gripper the grasp is computed for (in case of multiple grippers).

Several grasp descriptions expressed in location, RRP and relative location coordinates are shown in Table 3.1. The Figure in the last row of this Table is the same one considered in [22].

3.3.2 Grasp-related Invariant Features

Three alternative sets of invariant features related to the grasp description have been considered:

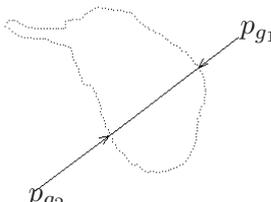
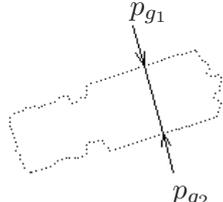
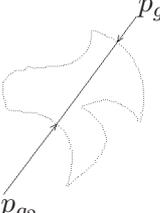
- The grasp description using RRP coordinates.
- The grasp description using relative coordinates.
- An homography related two object views, taking advantage of the fact that the object is assumed to be planar.

In the case of the grasp description based on location and RRP coordinates, the invariance cannot be ensured unless the whole set of contour points is rigidly translated from one image to the next one, since these coordinates are based on the location of the grasp points within the list of contour points. Therefore, the tracking of the grasp points has to be performed within the tracking of the entire object. The movement of the robot is bounded by the d.o.f. of the tracking algorithm.

The grasp description based on relative coordinates is based on the computation of second-order normalized moments of the object silhouette, which are invariant under four kinds of movements of the robotic system with respect to the image plane [25], so that the robot movement is restricted to 4 d.o.f.:

- Translations along the two axis that define the image plane (2 d.o.f.);
- Rotations with respect to an axis perpendicular to the image plane (1 d.o.f.);

Table 3.1: Grasp descriptions expressed in location, RRP and relative location coordinates.

Grasp	Location $p_{g_i} = (c_i, p_i)$	RRP $p_{g_i} = (c_i, r_i, t_i)$	Relative $(d_c, \alpha_m, d_1, d_2)$
	(0, 161) (0, 340)	(0, 53, 2) (0, 113, 1)	(0.0016, -87.1873°, -0.0035, 0.0038)
	(0, 145) (0, 287)	(0, 48, 1) (0, 95, 2)	(-0.0040, -267.5700°, -0.0038, 0.0039)
	(0, 116) (0, 543)	(0, 38, 2) (0, 181, 0)	(0.0002, -210.4850°, -0.0025, 0.0050)

- Changes of scale (1 d.o.f.: translation along an axis perpendicular to the image plane).

The invariance of the relative coordinates also requires that the whole object can be observed in each image. Partial occlusions of the shape of the object or enlargements, due to overlapping of other elements, would cause the loss of the invariance of the geometric moments between pairs of images, so that this description of the grasp could not be used.

Finally, a *homography* is a linear projective transformation between two planes [27,51]. This transformation produces, given a point in a plane, its corresponding point in the other plane. In general, a projective transformation between two projective planes I and I' can be represented by a general linear transformation in projective space:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \mathbb{H} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbb{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (3.1)$$

with $\mathbf{x} = [x_1 \ x_2 \ x_3]^T \in I$, $\mathbf{x}' = [x'_1 \ x'_2 \ x'_3]^T \in I'$. Equation 3.1 can then be rewritten as:

$$\begin{bmatrix} \lambda_i u'_i \\ \lambda_i v'_i \\ \lambda_i \end{bmatrix} = \mathbb{H} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}, \quad i \in [1, \dots, 4] \quad (3.2)$$

where $\mathbf{p} = [u \ v]^T$ and $\mathbf{p}' = [u' \ v']^T$ are the coordinates in Cartesian space of \mathbf{x} and \mathbf{x}' , respectively. They can be obtained as $u = x_1/x_3$, $v = x_2/x_3$, $u' = x'_1/x'_3$ and $v' = x'_2/x'_3$.

The homography matrix is invariant with respect to a 6 d.o.f. movement between two object views, i.e. the invariance of the grasp configuration is ensured under any kind of movement of the robotic manipulator in the free space.

3.4 Grasp Translation

The purpose of the grasp translation is to obtain, given a grasp in a first contour-based object description, the position of that grasp in a second object description. Three alternatives are proposed, as shown in Figure 3.6:

- Translation based on an invariant grasp description using RRP coordinates;
- Translation based on an invariant grasp description using relative coordinates;
- Translation based on the computation of the homography between the contours of the object in the two views.

In general, as the grasp is initially expressed in the contour of the object in the first view as a number of points, these points are used to compute a set of features that is invariant with respect to the relative movement between the two views of the object. This set is then applied on the contour of the object in the second view in order to compute there the position of the grasp points. The main difference between the above proposed strategies lies in the definition of this invariant set of features. In addition, the precision of the grasp translation will depend on the real invariance of the selected set. Algorithm 3 provides an outline of this general procedure.

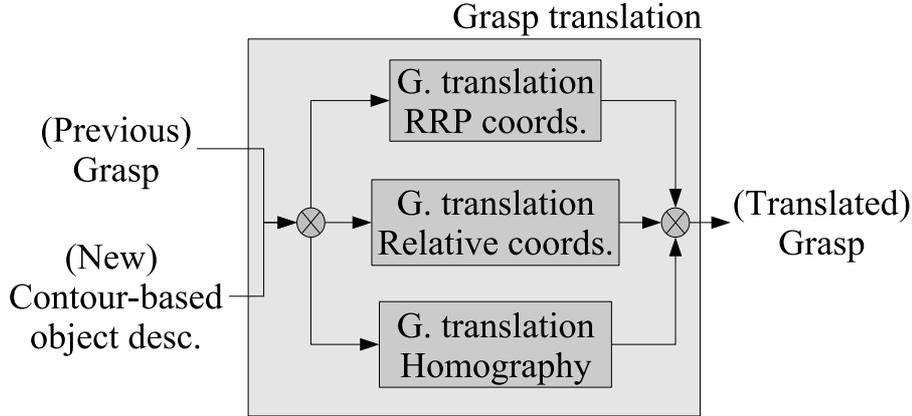


Figure 3.6: Grasp translation strategies.

Algorithm 3 Grasp translation

convert grasp points on first contour to invariant description;
 apply invariant description to second contour;
 compute grasp points on second contour from invariant description.

3.4.1 Translation based on RRP Coordinates

Following the specifications given in Section 3.3.2, this translation strategy has been applied together with an object tracking algorithm. In particular, B-splines are used as *active contours* [8] for the object tracking and the RRP coordinates provide a point indexing of these contours: the points sampled from the spline are used as *reference points* and the rest of the contour points can be computed as *interpolated points*. The object tracking algorithm is described in [64] and allows movements of 4 d.o.f. of the robot. The transformations corresponding to these 4 d.o.f. are: translations on the image plane, rotations around the normal to this plane and changes of scale. Therefore, the grasp description based on RRP coordinates is invariant with respect to the above four kind of transformations. Figure 3.7 shows the application of this translation strategy.

The tests of this translation strategy have considered only the external contour of the object. However, this does not imply a loss of generality of this strategy, which could be easily extended to the use of both external and internal contours since the RRP coordinates include a component indicating the contour each point belongs to.

3.4.2 Translation based on Relative Coordinates

This strategy is an alternative to the previous one, when no object tracking is performed. In this case, the description of each contour does not provide directly an invariant framework for expressing the location of the grasp points, since the order and the number of contour points may be different. Therefore other object features have to be considered.

The relative coordinates of the grasp points provide a description $(d_c, \alpha_m, d_1, d_2)$ that is invariant with respect to 4 d.o.f., which coincide with those mentioned in Section 3.4.1. As the grasp is initially expressed in the first object as a pair of points, those points are used to compute the above relative coordinates. These coordinates are applied to the second

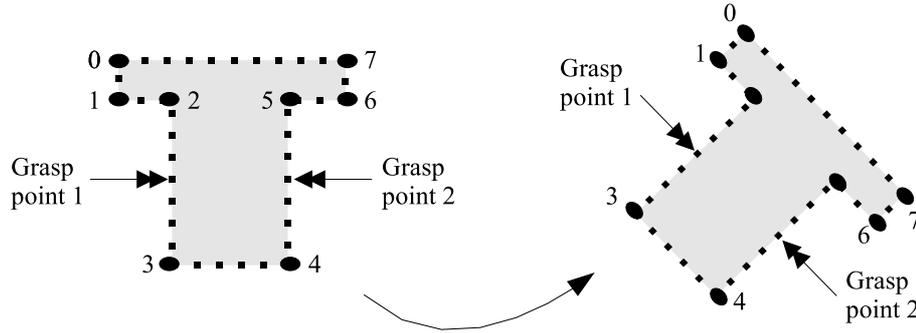


Figure 3.7: Invariance of the RRP coordinates. With a contour index 0, the RRP coordinates of the grasp points in both contours are $(0, 2, 4)$ and $(0, 4, 4)$.

object and then used to compute the position of the grasp points in that object. If the relative movement between the object and the camera is restricted to be under the above d.o.f., then, as shown in Figure 3.8, the grasp description in one contour using relative coordinates can also be applied to the other one.

3.4.3 Translation based on a Homography

In this strategy, a homography is used for translating the grasp points. Since projective points are equivalent up to a scalar λ , one element of the homography matrix \mathbb{H} can be arbitrarily set to a fixed value (in particular, $h_{33} = 1$ has been chosen), so that this matrix has eight independent parameters. It is thus necessary to find at least four initial point correspondences, provided that no three of them are collinear, between two projectively transformed planes to define it uniquely. With more than four correspondences, it would be possible to compute it using iterative methods, less sensitive to measurement errors [51]. Algorithm 4 provides an adaptation of algorithm 3 for this translation strategy.

Algorithm 4 Homography-based grasp translation

- extract four correspondences from the contours;
 - compute the homography matrix by using those correspondences;
 - perform the grasp translation using the homography;
 - refine the grasp translation.
-

Figure 3.9 illustrates the application of this algorithm to a translation of two grasp points. Four correspondences \mathbf{p}_i and \mathbf{p}'_i on the contour of a planar object are selected in each image I and I' in order to compute the homography matrix \mathbb{H} , which relates both object views. This matrix is then used to translate the grasp points, \mathbf{p}_{g1} and \mathbf{p}_{g2} , from image I to image I' , that is, to find the correspondences \mathbf{p}'_{g1} and \mathbf{p}'_{g2} in image I' of the grasp points \mathbf{p}_{g1} and \mathbf{p}_{g2} computed in image I .

The above algorithm is independent on the actual procedure considered for extracting the correspondences between the two contours. The goal of the refinement step included in this algorithm is to make the translation more robust to errors in the computation of the homography matrix and to ensure that the translated grasp belongs to the contour of the object.

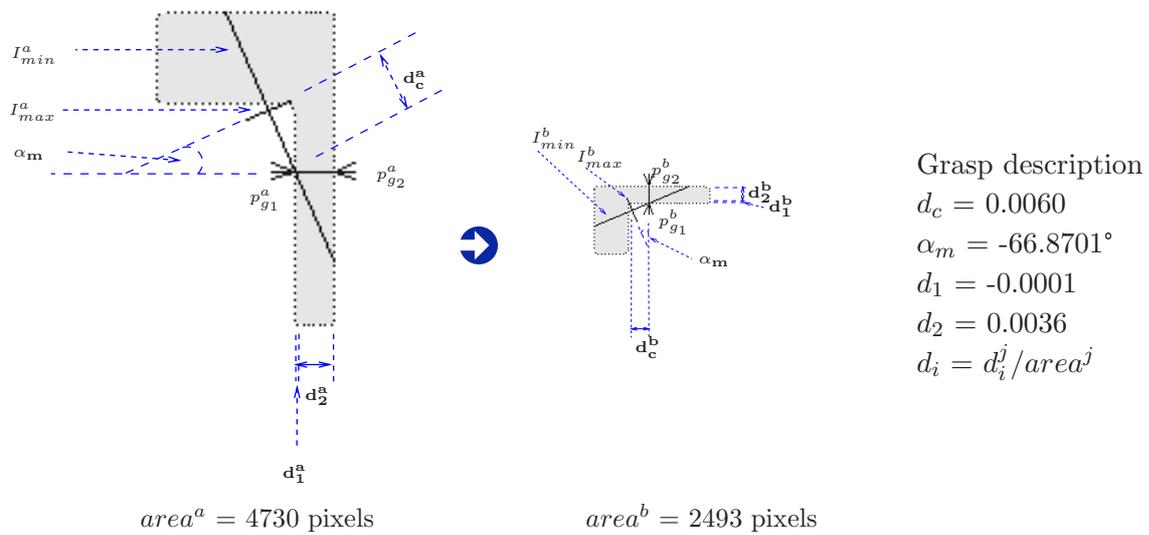


Figure 3.8: Invariance of the grasp description based on relative coordinates.

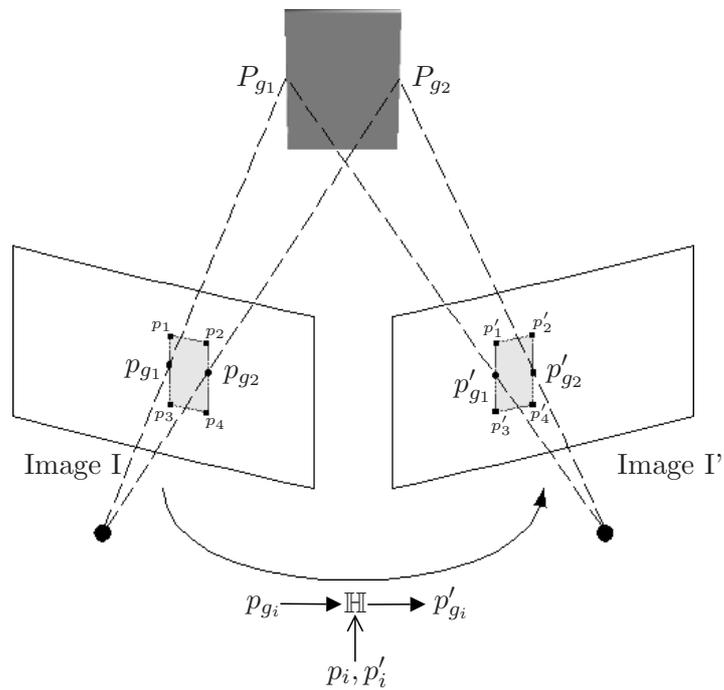


Figure 3.9: Use of a homography for the translation of grasp points.

Computation of the Homography

Let the four corresponding points be $[u_1 v_1]^T$, $[u_2 v_2]^T$, $[u_3 v_3]^T$, $[u_4 v_4]^T$, in plane I and $[u'_1 v'_1]^T$, $[u'_2 v'_2]^T$, $[u'_3 v'_3]^T$, $[u'_4 v'_4]^T$ in plane I' . The substitution of these points in Equation 3.2 produces a linear system that can be used to compute the components of \mathbb{H} .

In order to reduce the computational cost of obtaining \mathbb{H} , a direct, non-iterative method has been used to solve the linear system given by Equation 3.2. This method is sensitive to errors in the position of the correspondences; however, this is compensated by the refinement step.

Extraction of Correspondences

The search of correspondences between two views of the same scene is a typical problem of computer vision. Some authors have used matching of image sub-windows [75], epipolar geometry [44, 48] or B-splines [14] to find them. In most cases, a manual selection is required in the first view or there must exist some specific relationship between both views.

An automatic procedure is proposed for selecting correspondences between two contours. For simplicity, only the external contours of the object have been considered.

The proposed procedure is outlined by algorithm 5. First, as four correspondences are required, four points are selected in one of the contours. For this purpose, the curvature of each contour is analyzed [10]. This curvature is described by a *curvature vector*, which contains the result of evaluating a curvature function at each contour point. The proposed matching procedure is independent of the chosen curvature function. The size of the interval considered for computing this curvature in each contour is normalized with respect to the length of the contour, in order to ease the comparison between curvature vectors of different contours.

The matching procedure analyses the curvature vector of that contour and selects the points corresponding to the four peaks of highest curvature. Table 3.2 shows, for each object, the four peaks selected on the curvature vector of the first contour (–), as well as their four associated contour points (*).

Algorithm 5 Extraction of correspondences

```

compute the curvature vector of each contour;
select 4 points  $p_i$  with highest curvature in one contour;
for each point  $p_i$  do
    find its correspondence  $p_i$  on the other contour.
end for

```

Next, the procedure tries to match those points on the other contour. For each point \mathbf{p}_i , it tries to find the match between the curvature vectors of both contours. As it can be observed in Figure 3.10, it considers a neighborhood in the curvature vector of the first contour, centered at \mathbf{p}_i , and tries to find a matching neighborhood in the curvature vector of the other contour.

The point $\mathbf{p}'_{i_{\text{corr}}}$ of the second contour that is used as a starting point for the search of the matching neighborhood is at the same position as point \mathbf{p}_i within the list of points of their respective contours. A scale factor κ is used in order to take into account differences in the length of both contours:

$$i_c = \kappa i, \quad \kappa = l'_c / l_c \quad (3.3)$$

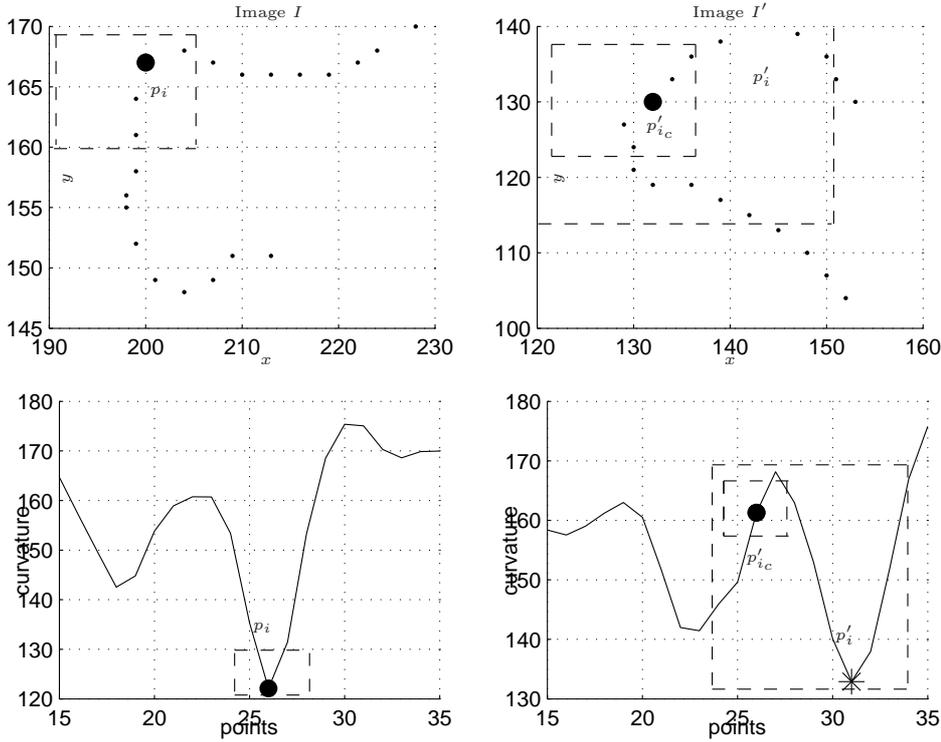


Figure 3.10: Search of the correspondence of a given point.

where l_c and l'_c are the number of points of the first and second contours, respectively. If $l_c = l'_c$, then $i_c = i$, that is, \mathbf{p}_i and $\mathbf{p}'_{i_{\text{corr}}}$ have the same location coordinates.

The search will be performed within an interval $[i_c - \kappa\xi, i_c + \kappa\xi]$, where ξ is the radius of this interval when $l_c = l'_c$. Let \mathbf{p}'_{j_i} , with $j_i \in [i_c - \kappa\xi, i_c + \kappa\xi]$, be a point of the second contour that has to be evaluated as candidate correspondence of \mathbf{p}_i . The matching error $d(\mathbf{p}_i, \mathbf{p}'_{j_i})$ between \mathbf{p}_i and \mathbf{p}'_{j_i} is computed as:

$$d(\mathbf{p}_i, \mathbf{p}'_{j_i}) = \sum_{k=-n}^n |v(i+k) - v'(j_i + \kappa k)|, \quad (3.4)$$

where $v(x)$ and $v'(x)$ are the values of the x -th element of the curvature vector of the first and the second contours, respectively, and n is the radius of the neighborhood considered for comparing both curvature vectors.

The matching neighborhood is considered to be that with the lowest matching error. The center \mathbf{p}'_i of such neighborhood is taken as the point correspondence of \mathbf{p}_i :

$$\mathbf{p}'_i = \min_{\mathbf{p}'_{j_i}} d(\mathbf{p}_i, \mathbf{p}'_{j_i}) \quad (3.5)$$

Table 3.2 shows the correspondences \mathbf{p}'_i (\bullet) in the second contour of the points \mathbf{p}_i ($*$) ($i \in [1, \dots, 4]$) selected in the first contour, as well as the centers ($-$) of their respective matching neighborhoods in their curvature vectors. To avoid ambiguities in the search of correspondences due to shape similarities within the same contour, such as the corners of the first Figure of this Table, the displacement between the two object views should be

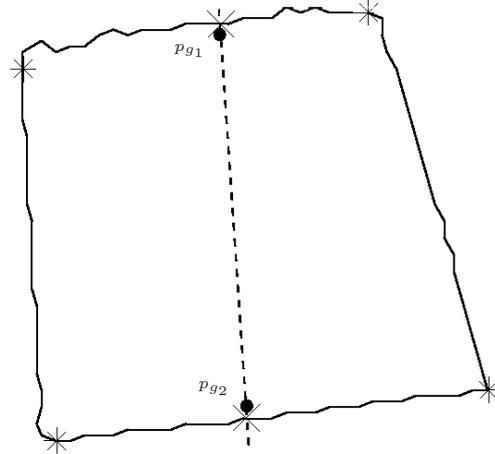


Figure 3.11: Correction of the position of the grasp points.

relatively small. Alternatively, a larger radius n for computing the matching error could be considered.

Application of the Homography and Grasp Refinement

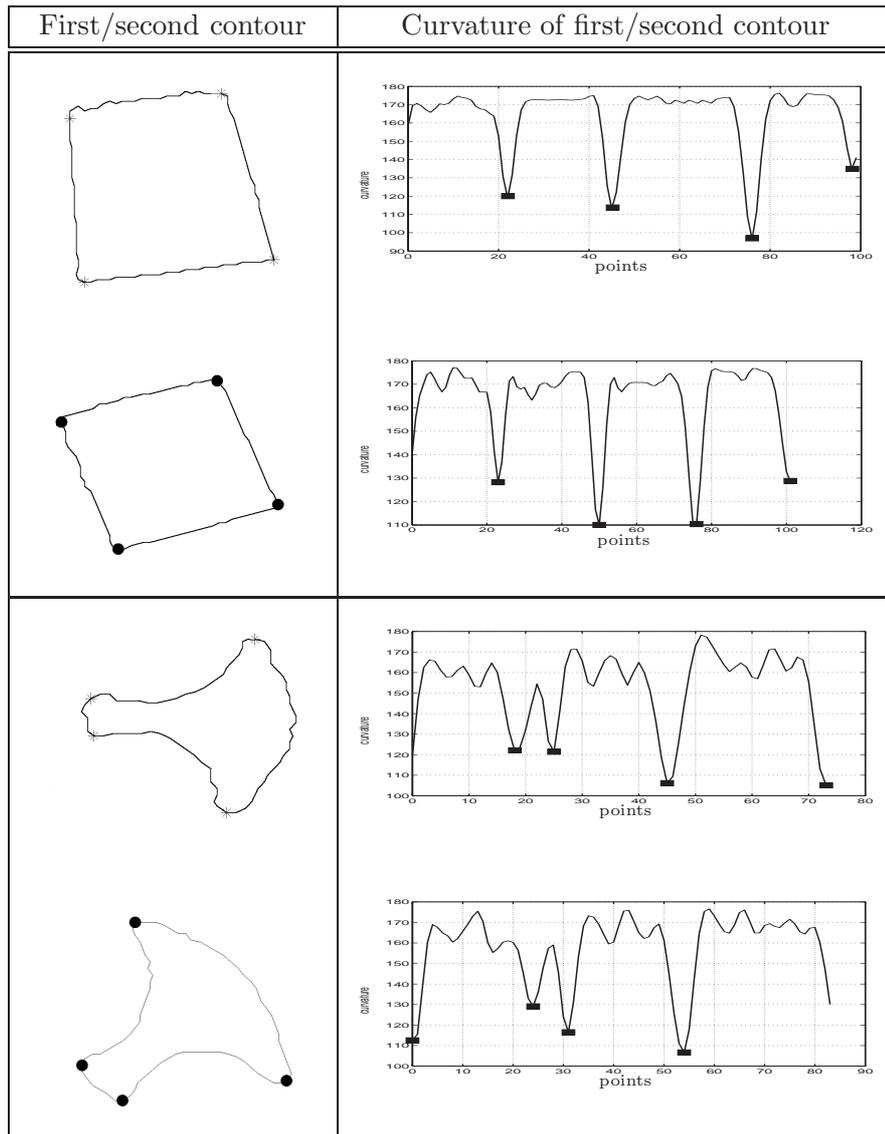
Once the homography matrix has been computed, the grasp points $\mathbf{p}_{\mathbf{g}_i}$ from the first contour, available in image I , are translated into image I' . Ideally, the translated points should lie on the corresponding contour extracted from I' .

Nevertheless, due to errors in the contour extraction in I and I' , in the selection of the corresponding points, and/or in the computation of the homography, the translated points \mathbf{p}'_i may happen not to be in that contour, but close to it. The correction in the position of these points, so that they lie on the contour can be achieved by using known restrictions in the relationship between the grasp points. Figure 3.11 shows the case in which the grasp points have been computed for a two-finger gripper: the line joining the two grasp points is considered to be the translated grasp line and the intersections between this line and the contour are computed. At least, there should be two intersections. The two intersection points $\mathbf{p}'_{\mathbf{g}_1}$ and $\mathbf{p}'_{\mathbf{g}_2}$ that are closest to the two translated are considered to be the grasp points in image I' .

3.5 Results

As mentioned in Section 3.4, the proposed grasp tracking procedures use a contour-based representation of the object as input. In the grasp translations based on the relative coordinates and on the homography, the contour of the object of interest is independently extracted in each image of the sequence. For this reason, object selection criteria, such as size or position in the image, are needed [61]; for example, in Figure 3.12, the object of interest is chosen, because of its size, as the largest present in the image. An example of grasp tracking without object tracking is given in Figure 3.13, where the grasp translation

Table 3.2: Selected points (*) and contour curvature on the first contour and their correspondences (●).



is based on the relative coordinates of the grasp points. Figure 3.14 shows a homography-based grasp tracking on a sequence of object views; the pairs of images considered for computing the homography are consecutive along the sequence.



Figure 3.12: Extraction and size-based selection of an object of interest.

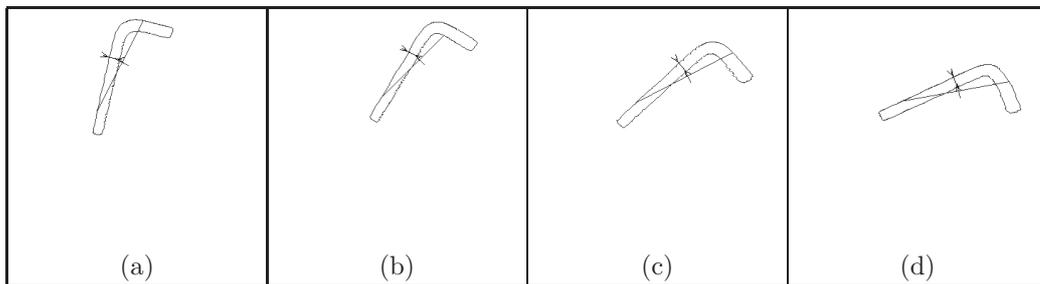


Figure 3.13: Grasp tracking based on relative coordinates.

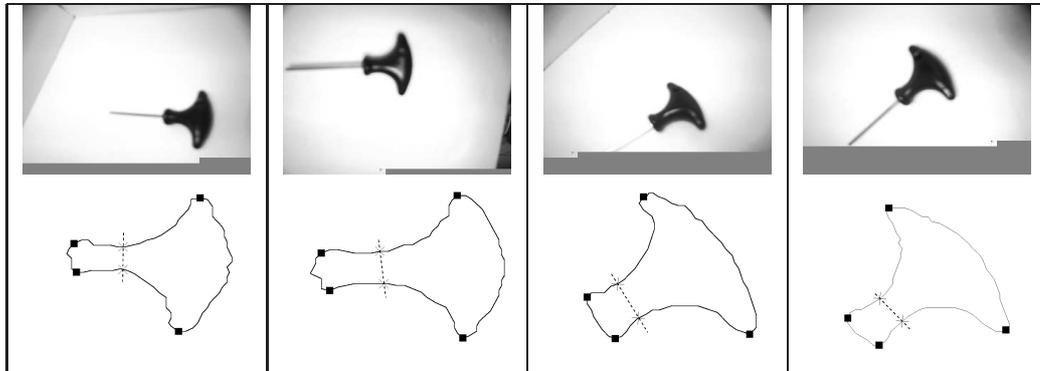


Figure 3.14: Grasp tracking based on a homography.

On the other side, in the grasp translation based on RRP coordinates, an object tracking method is used for selecting the object of interest. Figure 3.15 provides an example of a grasp tracking based on RRP coordinates and integrated within an object tracking procedure.

Table 3.3 presents examples of grasp translation based on RRP; in this case, the sampled points of a B-spline are used to set the reference points. Some examples of grasp translation based on relative coordinates are provided in Table 3.4, while Table 3.5 shows the grasp translation based on the homography.

The proposed tracking procedures can also be extended to track a grasp along a stereo sequence of images, as shown in Table 3.6. In this case, an initial grasp search is per-

Table 3.3: Grasp translation based on RRP coordinates.

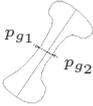
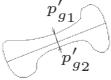
Original/translated grasp	Location $p_{g_i} = (c_i, p_i)$	RRP $p_{g_i} = (c_i, r_i, t_i)$	Relative $(d_c, \alpha_m, d_1, d_2)$
	(0, 587) (0, 242)	(0, 40, 6) (0, 15, 25)	(-0.0001, -95.9817°, -0.0013, 0.0007)
	(0, 607) (0, 243)	(0, 40, 6) (0, 15, 25)	(-0.0001, -87.8033°, -0.0012, 0.0007)
	(0, 64) (0, 102)	(0, 14, 4) (0, 24, 3)	(-0.0124, -95.6864°, -0.0072, 0.0077)
	(0, 80) (0, 102)	(0, 14, 4) (0, 24, 3)	(-0.0100, -98.4322°, -0.0049, 0.0057)

Table 3.4: Grasp translation using relative coordinates.

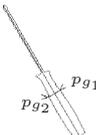
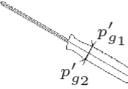
Original/translated grasp	Location $p_{g_i} = (c_i, p_i)$	RRP $p_{g_i} = (c_i, r_i, t_i)$	Relative $(d_c, \alpha_m, d_1, d_2)$
 	$(0, 100)$ $(0, 235)$ $(0, 95)$ $(0, 224)$	$(0, 16, 4)$ $(0, 39, 1)$ $(0, 15, 5)$ $(0, 37, 2)$	$(-0.0061, -102.7960^\circ,$ $-0.0089, -0.0025)$ $(-0.0061, -102.7960^\circ,$ $-0.0089, -0.0025)$
 	$(0, 152)$ $(0, 296)$ $(0, 172)$ $(0, 347)$	$(0, 25, 2)$ $(0, 49, 2)$ $(0, 28, 4)$ $(0, 57, 5)$	$(0.0009, -89.9220^\circ,$ $-0.0041, 0.0035)$ $(0.0009, -89.9220^\circ,$ $-0.0041, 0.0035)$

Table 3.5: Grasp translation based on a homography.

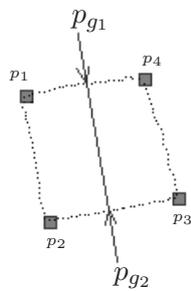
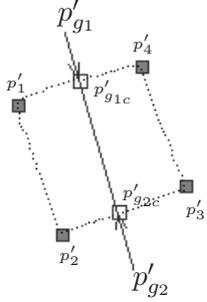
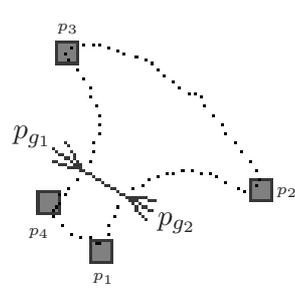
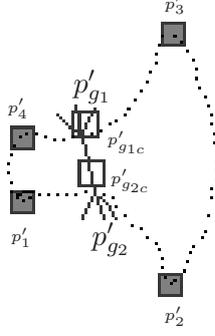
Original grasp	Translated grasp	
 <p data-bbox="279 795 614 996">Selected correspondences: (158, 97) (173, 173) (251, 157) (229, 86)</p> <p data-bbox="279 1030 486 1153">Selected grasp: $p_{g1} = (189, 85)$ $p_{g2} = (202, 160)$</p>	 <p data-bbox="670 795 1005 996">Selected correspondences: (140, 96) (167, 174) (242, 143) (214, 72)</p> <p data-bbox="670 1030 1244 1153">Grasp translation: Applied grasp $p'_{g1c} = (172, 77)$ $p'_{g1} = (171, 76)$ $p'_{g2c} = (195, 155)$ $p'_{g2} = (194, 156)$</p>	
 <p data-bbox="279 1568 614 1803">Image coordinates Selected correspondences: (200, 183) (253, 163) (187, 118) (183, 169)</p> <p data-bbox="279 1836 486 1960">Selected grasp: $p_{g1} = (187, 152)$ $p_{g2} = (201, 161)$</p>	 <p data-bbox="670 1568 1005 1803">Image coordinates Selected correspondences: (114, 97) (167, 126) (163, 42) (114, 76)</p> <p data-bbox="670 1836 1244 1960">Grasp translation: Applied grasp: $p'_{g1c} = (131, 68)$ $p'_{g1} = (130, 72)$ $p'_{g2c} = (133, 84)$ $p'_{g2} = (134, 90)$</p>	

Table 3.6: Grasp tracking along a stereo sequence of contours.

Left image				
Right image				

formed in the first left contour. The grasp selected in this search is then translated to the corresponding right contour. After that, the same grasp obtained for the initial left contour is translated to the next left contour in the sequence and the resulting grasp is applied to the corresponding right contour. Each tracked grasp along the sequence of left contours is checked to determine if it is still stable. For simplicity, no such checking has been performed on the grasps translated to the right contours.

As it can be observed in Table 3.3 and 3.4, both for the relative and the RRP coordinates, the original and the translated grasp are, as expected, similar. The existing differences are due to the discretization errors during the sampling or the extraction of the contour and the computation of the target features with respect to which the grasp is described. However, these differences are more reduced in the case of the RRP-based translation (Table 3.3), thanks to the use of an object tracking algorithm, which preserves better the shape of the object. With respect to the homography-based translation, the results from Figure 3.11 and Table 3.5 show that this strategy is sensitive to the errors in the search of the correspondences for computing the homography. In the search method proposed, these errors depend, in turn, on the accuracy of the object-extraction procedures. This remarks the importance of the refinement step as an error-compensating method.

More precisely, in order to provide an experimental evaluation of the proposed techniques, let's consider a two-finger grasp characterized by the grasp points p_{g_1} and p_{g_2} in the image I . In general, by using one of the procedures of grasp translation presented, these two points correspond to the points p'_{g_1} and p'_{g_2} in image I' . If this grasp configuration is translated back to image I , the two points p''_{g_1} and p''_{g_2} are obtained. At this point, it is

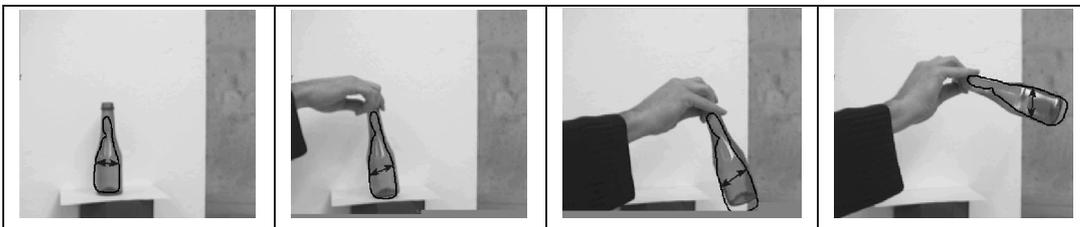


Figure 3.15: Tracking of an object and a selected grasp on a sequence of images using RRP coordinates.

possible to define the distance:

$$dist_i = \sqrt{(p_{x_{g_i}} - p''_{x_{g_i}})^2 + (p_{y_{g_i}} - p''_{y_{g_i}})^2}$$

in which $p_{x_{g_i}}$ and $p_{y_{g_i}}$ are the x and y components of the point p_{g_i} . Consequently, the average error and the maximum error are defined as:

$$e_{av} = \sum_{i=1}^2 \frac{dist_i}{2} \quad e_{max} = \max_{i \in \{1,2\}} dist_i$$

Table 3.7 provides the error values of the different grasp translation techniques for some of the examples presented above. It can be easily noticed that all of the three strategies grant satisfactory results.

Regarding the computational cost of the proposed grasp translation methods, those based on RRP and relative coordinates are the fastest, with computation times respectively below 15 and 35 ns, for a two-finger gripper and contours of around 300 points on a 2.5-GHz Pentium Celeron. The homography-based method is slower, requiring 290 ns under the same configuration, with 60% of the time spent in the search of correspondences. The grasp analysis, performed together with the translation during tracking, adds only 370 ns. The total time is in all cases below 1 ms.

In general, the results shown in this Section indicate that a very fast grasp translation can be achieved, with reasonable accuracy, by integrating the translation based on RRP coordinates with an object tracking procedure. Among the methods not requiring object tracking, the use of the relative coordinates of the grasp has the disadvantage of being only valid for two-finger grippers, while the homography-based method, although slower, offers a high degree of flexibility, since it can be used with virtually any gripper configuration, it does not rely on any particular correspondence/searching method and includes an error correction step. However, all these methods are fast enough to be used in real-time together with other image processing procedures, so the method selection may eventually depend on the whole set of such procedures to be applied.

3.6 Conclusions and Future Work

3.6.1 Conclusions

A method for the tracking of grasp points in image space has been described. This method is based on the application of a grasp translation procedure between pairs of images, for which three alternatives have been proposed. Among other applications, this tracking method enables the use of grasp points as control features within a visual servoing loop [10, 62]; it avoids repeating the grasp search at each iteration of the control loop and ensures that the same grasp is considered during the positioning movement.

The method proposed is really interesting in practical applications, as shown in the results. Each translation strategy has shown some advantages and disadvantages. In particular, the RRP-based grasp translation requires the use of an object tracking method, but it benefits from a more stable description of the object with respect to which the grasp is located. The grasp translation using relative coordinates does not rely on the tracking of the object, but is quite dependent on the full observability of the contour of the object, as well as of the robustness of the contour-extraction method. Finally, the homography-based translation has the advantages of allowing the tracking of a variable number of

Table 3.7: Grasp translation errors with the different techniques.

Original grasps		RRP	Relative Coord.	Homogr.
Figure 3.13 (a)-(b) $(p_{x_{g_1}}, p_{y_{g_1}}) = (121, 69)$ $(p_{x_{g_2}}, p_{y_{g_2}}) = (134, 73)$	$(p''_{x_{g_1}}, p''_{y_{g_1}})$	(121, 69)	(121, 69)	(121, 69)
	$(p''_{x_{g_2}}, p''_{y_{g_2}})$	(134, 73)	(134, 73)	(134, 73)
	$dist_1$	0	0	0
	$dist_2$	0	0	0
	e_{av}	0	0	0
	e_{max}	0	0	0
First Figure in Table 3.3 $(p_{x_{g_1}}, p_{y_{g_1}}) = (254, 156)$ $(p_{x_{g_2}}, p_{y_{g_2}}) = (287, 174)$	$(p''_{x_{g_1}}, p''_{y_{g_1}})$	(254, 156)	(254, 156)	(254, 156)
	$(p''_{x_{g_2}}, p''_{y_{g_2}})$	(287, 174)	(287, 174)	(287, 174)
	$dist_1$	0	0	0
	$dist_2$	0	0	0
	e_{av}	0	0	0
	e_{max}	0	0	0
Second Figure in Table 3.3 $(p_{x_{g_1}}, p_{y_{g_1}}) = (24, 162)$ $(p_{x_{g_2}}, p_{y_{g_2}}) = (20, 178)$	$(p''_{x_{g_1}}, p''_{y_{g_1}})$	(24, 162)	(27, 163)	(26, 163)
	$(p''_{x_{g_2}}, p''_{y_{g_2}})$	(20, 178)	(25, 179)	(23, 177)
	$dist_1$	0	3.16228	2.23607
	$dist_2$	0	5.09902	3.16228
	e_{av}	0	4.13065	2.69917
	e_{max}	0	5.09902	3.16228
First Figure in Table 3.4 $(p_{x_{g_1}}, p_{y_{g_1}}) = (132, 152)$ $(p_{x_{g_2}}, p_{y_{g_2}}) = (131, 146)$	$(p''_{x_{g_1}}, p''_{y_{g_1}})$	(132, 152)	(132, 152)	(132, 152)
	$(p''_{x_{g_2}}, p''_{y_{g_2}})$	(131, 146)	(131, 146)	(131, 146)
	$dist_1$	0	0	0
	$dist_2$	0	0	0
	e_{av}	0	0	0
	e_{max}	0	0	0
Second Figure in Table 3.4 $(p_{x_{g_1}}, p_{y_{g_1}}) = (138, 188)$ $(p_{x_{g_2}}, p_{y_{g_2}}) = (156, 176)$	$(p''_{x_{g_1}}, p''_{y_{g_1}})$	(138, 188)	(138, 188)	(139, 190)
	$(p''_{x_{g_2}}, p''_{y_{g_2}})$	(156, 176)	(155, 175)	(155, 177)
	$dist_1$	0	0	2.23607
	$dist_2$	0	1.41421	1.41421
	e_{av}	0	0.707107	1.82514
	e_{max}	0	1.41421	2.23607
First Figure in Tab 3.5 $(p_{x_{g_1}}, p_{y_{g_1}}) = (166, 144)$ $(p_{x_{g_2}}, p_{y_{g_2}}) = (243, 126)$	$(p''_{x_{g_1}}, p''_{y_{g_1}})$	(166, 144)	(166, 144)	(166, 144)
	$(p''_{x_{g_2}}, p''_{y_{g_2}})$	(243, 126)	(243, 126)	(243, 126)
	$dist_1$	0	0	0
	$dist_2$	0	0	0
	e_{av}	0	0	0
	e_{max}	0	0	0

grasp points and, with a careful selection of the point correspondences, of not necessarily requiring a view of the whole shape of the object. Nevertheless, its computation is more complex than that of the other methods.

3.6.2 Future Work

Future activity will involve, among others, the analysis of the integration of the proposed grasp translation and tracking procedures in a visual servoing control loop such as the one outlined in Figure 3.1. An extension to objects with holes will also be considered, which will require a matching of their internal contours. Finally, the grasp tracking described can be a useful tool for the search of 3D grasps.

Chapter 4

Robust Contact Detection and Force Regulation

The Chapter presents an innovative hybrid system approach for the control of robotic interaction. This hybrid architecture can control a manipulator during the stages of the interaction: the free motion, the transition phase and the constrained motion. The switching logic is based on a contact detection which is robust with respect to measurement noise so to guarantee that, during the robotic task, no bounces are present. The design of the controller is based on a Lyapunov analysis and depends on the viscoelastic parameters of the environment. The hybrid control can guarantee good performance in both cases of stiff and compliant contact model with a constraint on the velocity of impact.

4.1 Introduction

In this Chapter an innovative solution to the problem of robotic interaction is developed. We present a hybrid control architecture that involves a position and a force controller, so that the position control law steers the manipulator to a target point in the workspace and, once the contact is detected, the control law switches to a force control that makes the manipulator exerting the desired force. The detection of the contact is realized with a strategy based on force information: when the measured contact force reaches some level, than the manipulator is supposed to be in contact. Vice versa, when the measured contact force goes below some level, than the manipulator is supposed to have detached the contact. The proposed hybrid control assures the stability of the manipulator so that, once the manipulator gets in contact with the environment, no bounces are present and it never loses the contact. A model of the environment is required and the maximum admitted velocity of impact depends on its viscoelastic parameters. The control scheme provides a margin of robustness with respect to actuators errors and to force sensors noise

and it avoids chattering problems, i.e. multiple switching between the position and the force controller.

The Chapter is organized as follows: Section 4.2 describes the dynamical model of a generic manipulator and the compliant contact force model. The main idea of the proposed hybrid control architecture is given in Section 4.3 and more technical details are presented Section 4.4. Section 4.5 goes through the theoretical part of the design of the control synthesis and, finally, simulation results are reported in Section 4.6. In Section 4.7, the hybrid control architecture is enriched by introducing saturations in the control loop. Finally, in Section 4.8, simulation results of the saturated case are illustrated.

4.2 General Model

In this section, we present a dynamical model of a generic robotic manipulator and a model of the reaction forces due to the interaction between the manipulator and the environment. This scenario is depicted in Figure 4.1. The figure highlights that the reference frame is considered attached at the contact point on the surface S of the environment and that, thanks to its compliance, the surface is compressed, when the manipulator gets in contact with it, so that the desired position x_1^d can be reached.

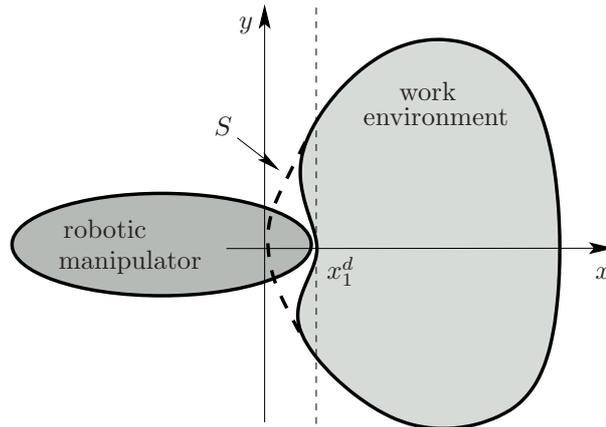


Figure 4.1: A robotic manipulator interacting with a work environment with surface identified by S .

4.2.1 General Robotic Manipulator Model

Let $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be a smooth and invertible mapping describing the forward kinematics of the robotic manipulator and relating the generalized vector of the manipulator joints angles $\theta \in \mathbb{R}^n$ and the workspace variables $x \in \mathbb{R}^m$.

From the Euler-Lagrange equations, it follows that the dynamics of the manipulator are:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) = \tau - J(\theta)^T f_c, \quad (4.1)$$

where:

- $M(\theta) \in \mathbb{R}^{n \times n}$ is the manipulator inertia matrix that is symmetric and positive definite.

- $C(\theta, \dot{\theta}) \in \mathbb{R}^{n \times n}$ is the Coriolis matrix. The vector $C(\theta, \dot{\theta})\dot{\theta}$ gives the Coriolis and centrifugal force terms.
- $N(\theta, \dot{\theta}) \in \mathbb{R}^n$ includes gravity terms and other forces that act at the joints.
- $\tau \in \mathbb{R}^n$ is the vector of the actuators torques.
- $J(\theta) \in \mathbb{R}^{m \times n}$ is the Jacobian matrix relating the joint space velocity to the workspace velocity, i.e. $\dot{x} = J(\theta)\dot{\theta}$.
- $f_c \in \mathbb{R}^m$ is the vector of the reaction forces due to the interaction between the manipulator and the environment.

Since we are interested in the interaction between the manipulator and the work environment, we rewrite the dynamic equation of motion (4.1) in workspace coordinates:

$$\tilde{M}(\theta)\ddot{x} + \tilde{C}(\theta, \dot{\theta})\dot{x} + \tilde{N}(\theta, \dot{\theta}) = F - f_c,$$

where

- $\tilde{M}(\theta) = J^{-T}(\theta)M(\theta)J^{-1}(\theta)$ is the manipulator inertia matrix.
- $\tilde{C}(\theta, \dot{\theta}) = J^{-T}(\theta)(C(\theta, \dot{\theta})J^{-1}(\theta) + M(\theta)\frac{d}{dt}J^{-1}(\theta))$ is the Coriolis matrix;
- $\tilde{N}(\theta, \dot{\theta}) = J^{-T}(\theta)N(\theta, \dot{\theta})$ includes gravity terms and other forces which act at the joints;
- $F := J^{-T}(\theta)\tau$ is the vector of forces and torques applied at the end-effector of the manipulator.

4.2.2 Compliant Contact Force Model

We consider the linear contact model presented in [24] to characterize the relationship between the bodies penetration and the reaction force. In this model, the viscoelastic material of the environment is described as the mechanical parallel of a linear spring and a damper which are represented, respectively, by a stiffness matrix $K_c \in \mathbb{R}^{m \times m}$ and a damper matrix $B_c \in \mathbb{R}^{m \times m}$. (This follows the simple 1-D linear model of Kelvin-Voigt, [24]).

Let $s : \mathbb{R}^m \rightarrow \mathbb{R}$ be such that $s(x) \leq 0$ if x is a point in the work environment and $s(x) > 0$ otherwise. Then, $W = \{x \in \mathbb{R}^m \mid s(x) \leq 0\}$ is the work environment and $S = \{x \in \mathbb{R}^m \mid s(x) = 0\}$ is its surface. With this definition, the contact force model is given by

$$f_c = \begin{cases} K_c x + B_c \dot{x} & \text{if } s(x) \leq 0 \\ 0 & \text{if } s(x) > 0. \end{cases}$$

4.2.3 Model Reduction

In many applications, with the knowledge of some of the parameters and state of the manipulator, it is possible to design an inner feedback loop that stabilizes some of the internal and external forces of the manipulator. Such technique is commonly used in the control of robotic manipulators literature, e.g. [11,37,72] to just list a few. Following these

references, let u be the input control force in the workspace coordinates and let the inner feedback law be given by

$$F = u + \tilde{C}(\theta, \dot{\theta})\dot{x} + \tilde{N}(\theta, \dot{\theta}) .$$

This feedback law is basically a state feedback linearization law that reduces the dynamics of the manipulator to

$$\tilde{M}(\theta)\ddot{x} = u - f_c. \quad (4.2)$$

As further described in [72], it is possible to distinguish between constrained and unconstrained direction of the motion of a manipulator interacting with an environment. Following [72], without loss of generality, we consider the case in which the interaction between the manipulator and its environment occurs along a normal direction. In this way, the manipulator consists simply of a mass with motion constrained to a straight line. The interaction with the work environment occurs at some point on that line. We further assume that the mass is unitary. Then, the system gets reduced to the second-order system

$$\ddot{x} = u - f_c \quad (4.3)$$

where $x := [x_1 \ x_2] \in \mathbb{R}^n$, x_1 is the position, x_2 the velocity, and f_c is the contact force which is given by

$$f_c = \begin{cases} k_c x_1 + b_c x_2 & \text{if } s(x_1) \leq 0 \\ 0 & \text{if } s(x_1) > 0 \end{cases} \quad (4.4)$$

where $k_c \in (0, +\infty)$ and $b_c \in (0, +\infty)$ are the elastic and the viscous parameters of the contact.

4.3 Hysteresis Control based on Force Measurement

We consider the problem of controlling a manipulator in the task of establishing contact with a work environment and of regulating the interaction force to a pre-specified value. This problem has been previously addressed in the literature, see e.g. [11, 37, 55, 68, 72] to just list a few.

Perhaps the simplest strategy to accomplish the task described above is (whether contact/non-contact has been detected between the manipulator and the environment) to switch between a position controller, which steers the manipulator close to the work environment, and a force controller, which regulates the force to a desired value. When the position of the manipulator is $x_1 < 0$, then the strategy is to apply the position controller with a set-point equal to $x_1^d > 0$ and, when the position of the manipulator reaches $x_1 \geq 0$, then the strategy is to apply the force controller in order to make the robotic manipulator exerting the desired contact force on the work environment. This means that the strategy of commutation between the two controllers is based on position information, and in particular on the measurement of x_1 .

The phase plot in Figure 4.2 depicts a trajectory resulting from this switching logic. Notice that several bounces, and therefore several switches, are present before the manipulator reaches the desired position/contact force configuration. The manipulator establishes contact with the environment at the point A and the control law is switched to the force controller. The manipulator loses contact in the point B and the control law is switched back from the position to the force controller. Eventually, the trajectories approach the steady state at which the contact force is stabilized to the desired set-point. Note that this is possible after multiple bounces off the work environment have occurred.

More importantly, with such control strategy, the closed-loop system may be vulnerable to measurement noise in the position and velocity. To illustrate this note that if, when the manipulator is close to the environment, there is a small measurement noise in the position so that the manipulator seems to be already in contact, the force controller will be selected sooner than desired. In the same way, once in contact, small measurement noise can indicate that no contact has been established and therefore a switch back to the position controller can be triggered. In this way, chattering on the controller can arise when the manipulator is in a neighborhood of the contact surface.

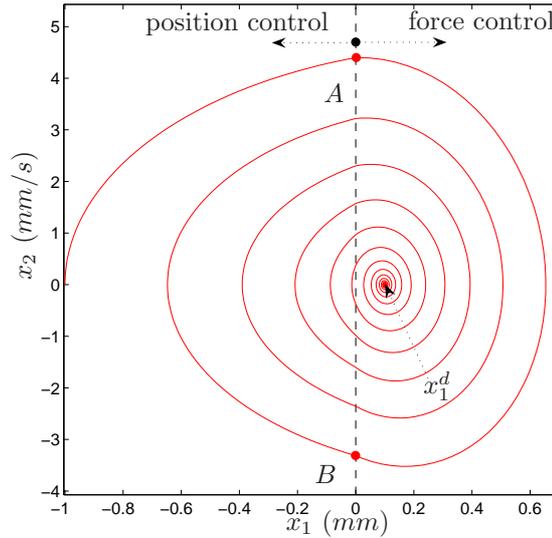


Figure 4.2: Example of a switching strategy based on position information: the manipulator establishes and loses the contact in A and B , respectively. Several bounces (and switches between the position and force controllers) are present before the manipulator reaches the desired position/contact force configuration.

In this paper, we propose a hybrid controller that minimizes such issues. The control strategy is to switch to the force controller only when a level of force is detected indicating contact between the robotic manipulator and the work environment. During the detaching phase, the control strategy is to switch back to the position controller when the measured force is below a certain level. This mechanism introduces hysteresis in the switches between the two controllers and, for this reason, it corresponds to a hybrid strategy. More precisely, the control strategy is as follows. Let $0 < \gamma_1 < \gamma_2$,

1. If the contact force f_c is smaller or equal than the positive constant thresholds γ_2 , apply the position controller until the contact force reaches γ_2 , and in that event, switch to the force controller.
2. If the contact force f_c is larger or equal than the positive constant thresholds γ_2 , apply the force controller. If f_c is below γ_1 , then switch to the position controller.

With the contact model (4.4), the contact force f_c is a linear combination of the two state variable x_1 and x_2 , that identify the position and the velocity of the robotic manipulator. This means that the conditions $f_c \geq \gamma_2$ and $f_c \leq \gamma_1$ for switching between the two controllers, correspond to half planes in the phase diagram. In particular these half planes

are defined by the lines $l_{\gamma_1} : \{(x_1, x_2) \mid x_2 = -\frac{k_c}{b_c}x_1 + \frac{\gamma_1}{b_c}\}$ and $l_{\gamma_2} : \{(x_1, x_2) \mid x_2 = -\frac{k_c}{b_c}x_1 + \frac{\gamma_2}{b_c}\}$ that we depict in Figure 4.3. Note that these lines have fixed slope given by $-\frac{k_c}{b_c}$, for given viscoelastic parameters k_c and b_c of the work environment.

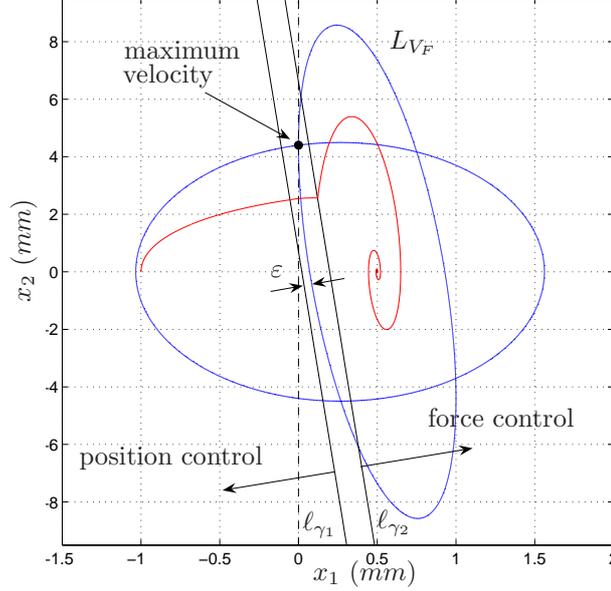


Figure 4.3: Example of switching strategy based on hysteresis. l_{γ_1} and l_{γ_2} define the sets for the switches based on force, ε is the margin of robustness and L_{V_F} is the largest Lyapunov function level set for closed-loop system with force controller that does not leave the right half plane.

4.4 Hybrid Controller

In this section, we present the hybrid controller which implements the control strategy outlined in Section 4.3. We start by describing the position and force controllers embedded in the hybrid controller.

4.4.1 Position Controller

Let x_1^d be the desired set point for the position of contact. We consider a position controller for set-point stabilization of the position x_1 of the manipulator in (4.3) that relies on position and velocity measurements of the manipulator and is given by the proportional/derivative control law

$$\kappa_P(x, x_1^d) = -k_p(x_1 - x_1^d) - k_d x_2 \quad (4.5)$$

where x_1^d is the set-point, and $k_p, k_d \in \mathbb{R}$ are constants. Proportional/derivative controllers have been previously used in the literature for set-point stabilization of manipulators, e.g. [37, 49, 72].

4.4.2 Force Controller

Let f_c^d be the desired set-point for the contact force. We consider a force controller that only relies on measurements of the contact force and that is given by the proportional/feedforward control law

$$\kappa_F(f_c, f_c^d) = f_c + k_f(f_c^d - f_c) \quad (4.6)$$

where $k_f \in \mathbb{R}$ is a constant. Similar force control strategies have been considered in [49,52]. We will denote the maximum allowed contact force by \hat{f}_c , and assume that $f_c^d \leq \hat{f}_c$.

4.4.3 Control Strategy

The main idea of the control strategy for contact detection outlined in Section 4.3 is to switch from position to force controller (and vice versa) relying only on information of measurements of the contact force f_c . The key feature of this strategy, which grants to the closed-loop system a margin of robustness to measurement noise, is that the controller selection depends on the memory of the controller; hence, it is a (logic-based) hybrid controller.

We implement the control strategy in a hybrid controller which we denote by \mathcal{H}_c . The state of the controller is given by the logic variable $q \in \{0, 1\}$. Let the constants $\gamma_1, \gamma_2 \in \mathbb{R}_{>0}$ be the parameters of \mathcal{H}_c , the threshold levels.

As depicted in Figure 4.4, the update law for the logic variable q is so that it switches between 0 and 1 based on the value of f_c with hysteresis levels defined by γ_1, γ_2 . Two different transitions are possible:

- $q = 0 \rightarrow 1$ (path: $0 \rightarrow A \rightarrow B \rightarrow C$):
The logic variable q can only be mapped to 1 when the measured contact force reaches the threshold γ_2 , i.e. when $f_c \geq \gamma_2$.
- $q = 1 \rightarrow 0$ (path: $C \rightarrow B \rightarrow D \rightarrow 0$):
The logic variable q can only be mapped to 0 when the measured contact force is below the threshold γ_1 , i.e. when $f_c \leq \gamma_1$.

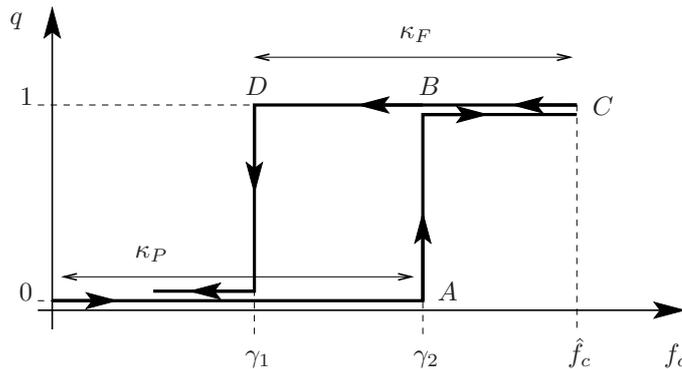


Figure 4.4: Contact detection strategy with hysteresis. Constants γ_1, γ_2 define the thresholds for the change from mode $q = 0$ to mode $q = 1$ (and vice versa).

The dynamics of the hybrid controller \mathcal{H}_c are as follows.

Jumps:

- From $q = 0$ to $q = 1$ (i.e. $q^+ = 1$)
When $q = 0$ and $f_c \geq \gamma_2$, the logic variable q is mapped to 1.
- From $q = 1$ to $q = 0$ (i.e. $q^+ = 0$)
When $q = 1$ and $f_c \leq \gamma_1$, the logic variable q is mapped to 0.

Flows:

- $\dot{q} = 0$
When $q = 0$ and $f_c \leq \gamma_2$, or when $q = 1$ and $f_c \geq \gamma_1$, the logic variable remains constant. The output of the controller is given by

$$u := \kappa(x, f_c, x_1^d, f_c^d, q) := \begin{cases} \kappa_P(x, x_1^d) & \text{if } q = 0 \\ \kappa_F(f_c, f_c^d) & \text{if } q = 1 \end{cases} .$$

Remark 4.4.1. *The logic variable q , basically, indicates whether it is “safe” or not to switch from the position controller κ_P to the force controller κ_F . More precisely, if the position and force controllers are designed so that bounces off the work environment are not possible when a certain level of contact force has been achieved, switching from the position controller to the force controller is enabled. Note that the hybrid controller \mathcal{H}_c switches from κ_P to κ_F if the logic variable q makes a transition from 0 to 1. This is possible only if the position controller is able of generating a contact force that is larger than γ_2 and if the measurement of f_c experiences an increase of at least $\gamma_2 - \gamma_1 > 0$. On the other hand, the hybrid controller \mathcal{H}_c switches from κ_F to κ_P if the logic variable q makes a transition from 1 to 0. This is possible only if the measurement of f_c experiences a decrease of at least $\gamma_2 - \gamma_1 > 0$.*

Remark 4.4.2. *Note that the set-point x_1^d of the position controller is a parameter we have to design. In order to guarantee that there is at least one switch from the position to the force controller, we have to grant that the steady state point x_1^P (that depends on x_1^d) is located on the right of the line ℓ_{γ_2} in the phase diagram. If so, we are sure that, while the system is controlled by κ_P and therefore it is approaching x_1^P , the trajectories hit the line ℓ_{γ_2} , ensuring the switch.*

4.4.4 Closed-loop System

The closed-loop system, denoted by \mathcal{H}_{cl} and results of controlling (4.3) with the hybrid controller \mathcal{H}_c , has continuous dynamics given by

$$\begin{aligned} \dot{x}_1 &= x_2 & \text{if } & \begin{cases} q = 0 \text{ and } k_c x_1 + b_c x_2 \leq \gamma_2 \text{ and } x_1 \geq 0 \\ \text{or} \\ q = 0 \text{ and } x_1 \leq 0 \\ \text{or} \\ q = 1 \text{ and } k_c x_1 + b_c x_2 \geq \gamma_1 \text{ and } x_1 \geq 0 \end{cases} \\ \dot{x}_2 &\in F_2(x, q) \\ \dot{q} &= 0 \end{aligned}$$

and jump dynamics given by

$$\begin{aligned} x_1^+ &= x_1 \\ x_2^+ &= x_2 \quad =: G_0(x, q) \quad \text{if} \quad \left\{ \begin{array}{l} q = 0 \quad \text{and} \quad k_c x_1 + b_c x_2 \geq \gamma_2 \quad \text{and} \quad x_1 \geq 0 \\ q^+ = 1 \end{array} \right. \\ \\ x_1^+ &= x_1 \\ x_2^+ &= x_2 \quad =: G_1(x, q) \quad \text{if} \quad \left\{ \begin{array}{l} q = 1 \quad \text{and} \quad k_c x_1 + b_c x_2 \leq \gamma_1 \quad \text{and} \quad x_1 \geq 0 \\ \text{or} \\ q = 1 \quad \text{and} \quad x_1 \leq 0 \end{array} \right. \\ q^+ &= 0 \end{aligned}$$

where $F_2 : \mathbb{R}^2 \times \{0, 1\} \rightarrow \mathbb{R}$ is a set-valued function given by

$$F_2(x, q) = (1 - q)\kappa_P(x, x_1^d) + \begin{cases} [(k_c x_1 + b_c x_2)(1 - k_f) + k_f f_c^d] q - (k_c x_1 + b_c x_2) & \text{if } x_1 \geq 0 \\ k_f f_c^d q & \text{if } x_1 \leq 0 \end{cases}$$

More precisely, the closed loop system \mathcal{H}_{cl} is defined by $\mathcal{H}_{cl} = (F, G, C, D)$ on the state space \mathbb{R}^2

$$\mathcal{H}_{cl} \quad \left\{ \begin{array}{l} \left[\begin{array}{l} \dot{x} \\ \dot{q} \end{array} \right] \in F(x, q) \quad \text{when } (x, q) \in C \\ \left[\begin{array}{l} x^+ \\ q^+ \end{array} \right] \in G(x) \quad \text{when } (x, q) \in D \end{array} \right. \quad (4.7)$$

in which $F(x, q) = \begin{bmatrix} x_2 \\ F_2(x, q) \\ 0 \end{bmatrix}$ is the flow map, $G(x, q) = \begin{cases} G_0(x, q) & \text{if } q = 0 \\ G_1(x, q) & \text{if } q = 1 \end{cases}$ is the jump map, C is a closed set in \mathbb{R}^2 that defines the flow set given by

$$C := C_0 \cup C_1$$

where:

$$\begin{aligned} C_0 &:= \{(q, x) \mid q = 0 \quad \text{and} \quad (x_1 \leq 0 \quad \text{or} \quad (x_1 \geq 0 \quad \text{and} \quad k_c x_1 + b_c x_2 \leq \gamma_2))\} \\ C_1 &:= \{(q, x) \mid q = 1 \quad \text{and} \quad x_1 \geq 0 \quad \text{and} \quad k_c x_1 + b_c x_2 \geq \gamma_1\}. \end{aligned}$$

and D is a closed set in \mathbb{R}^2 that defines the jump set given by

$$D := D_0 \cup D_1$$

where:

$$\begin{aligned} D_0 &:= \{(q, x) \mid q = 0 \quad \text{and} \quad (x_1 \geq 0 \quad \text{and} \quad k_c x_1 + b_c x_2 \geq \gamma_2)\} = \overline{\mathbb{R}^2 \setminus C_0} \\ D_1 &:= \{(q, x) \mid q = 1 \quad \text{and} \quad (x_1 \leq 0 \quad \text{or} \quad (x_1 \geq 0 \quad \text{and} \quad k_c x_1 + b_c x_2 \leq \gamma_1))\} = \overline{\mathbb{R}^2 \setminus C_1}. \end{aligned}$$

Figure 4.5 illustrates the flow and the jump sets.

Figure 4.6 depicts the block diagram of the closed-loop system \mathcal{H}_{cl} .

The idea is that, in order to guarantee that the trajectories do not bounce off when the switch occurs, we design a force controller and we choose γ_2 such that, at switching, the trajectories stay inside the largest Lyapunov function level set L_{V_F} that does not leave

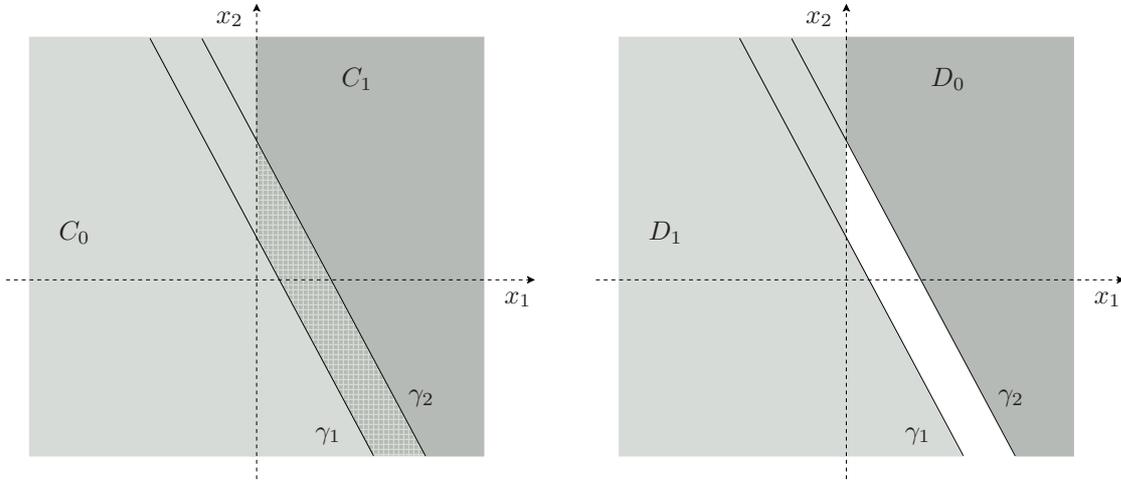


Figure 4.5: Flow and jump sets.

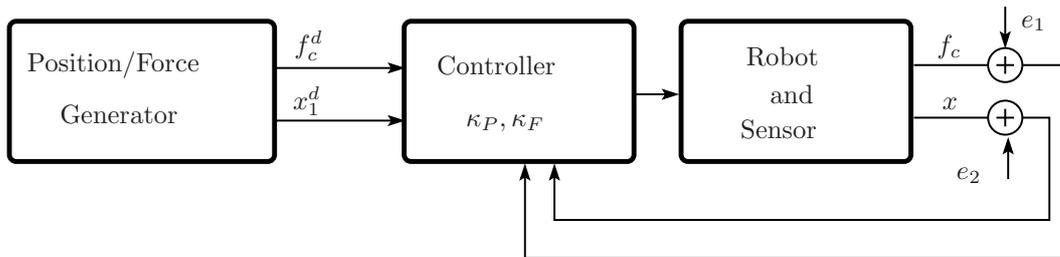


Figure 4.6: Hybrid control scheme. A position/force block generates the set-points f_c^d and x_1^d . The controller has as input the position of the manipulator x in the workspace coordinates and the contact force f_c measured by the sensor. Noise enters through the measurements of f_c and x .

the right half plane. Vice versa, we choose γ_1 so that the line referred to this threshold is outside the level set of the closed-loop system with the force controller. This implies that the system leaves this level set only when some changes in the set-points require a detachment of the contact.

Note that with hysteresis, once the contact has been established, there is no chattering between the two controllers and, therefore, there is no bouncing off. In fact, once the force controller is activated, the trajectories need to reach the set defined by l_{γ_1} , in order to switch back to the position controller. Since by design, in this contact condition, trajectories do not leave L_{V_F} , the margin of robustness to measurement noise, denoted with ε in Figure 4.3, is given by the minimum distance between L_{V_F} and l_{γ_1} . The only requirement of this architecture is that the velocity of impact has to be bounded. According to that, the design of the position controller has to guarantee this constraint on the impact velocity that is basically due to the viscoelastic properties of the environment.

4.5 Controller Design

We now present a result that guarantees the existence of parameters for the hybrid controller \mathcal{H}_c for contact detection. We go through the steps of the proof since they highlight the design of our hybrid control strategy.

Theorem 4.5.1. *Given parameters k_c, b_c of the work environment and desired contact force $f_c^d < \hat{f}_c$, there exist*

- A compact set $K \subset \mathbb{R}^2$ of initial conditions of the manipulator;
- Parameters $k_p, k_d, k_f, \gamma_1, \gamma_2$ of the hybrid controller \mathcal{H}_c ;
- Set-point x_1^d for the regulation of the position;

such that for every initial condition $x^0 \in K$, the solutions (x, q) to \mathcal{H}_{cl} approach the compact set $\{x_1^F\} \times \{1\}$. Moreover, the parameters satisfy

$$k_p, k_d, k_f > 0, \quad x_1^d > \frac{cx_1^F b_c (k_p + k_c)}{k_p k_c b} \quad (4.8)$$

$$Q_F = \begin{bmatrix} 2ck_c k_f & (bk_c + cb_c)k_f - a \\ (bk_c + cb_c)k_f - a & 2(bb_c k_f - c) \end{bmatrix} > 0$$

$$\gamma_1 \in \left(0, k_c \left(x_1^F - \sqrt{\frac{2r}{p_1}} \right) \right) \quad (4.9)$$

$$\gamma_2 \in \left(\frac{cb_c x_1^F}{b}, \frac{k_c k_p}{k_p + k_c} x_1^d \right) \quad (4.10)$$

where

$$\theta = \beta - \frac{\pi}{2}, \quad \beta = \arctan \left(-\frac{k_c}{b_c} \right) \in \left(-\frac{\pi}{2}, 0 \right) \quad (4.11)$$

$$x_1^F = \frac{f_c^d}{k_c}, \quad r = \frac{ab - c^2}{2b} (x_1^F)^2 \quad (4.12)$$

$$a, b, c > 0, \quad c = (b - a) \frac{\sin \beta + \cos \beta}{\sin^4 \beta - \cos^4 \beta}$$

$$\begin{bmatrix} \frac{b \cos^2 \beta - a \sin^2 \beta}{\cos^4 \beta - \sin^4 \beta} & 0 \\ 0 & \frac{b \sin^2 \beta - a \cos^2 \beta}{\sin^4 \beta - \cos^4 \beta} \end{bmatrix} > 0.$$

We proceed by constructing Lyapunov functions for the closed-loop system resulting when the position and force controller are in the loop. These Lyapunov functions decrease along trajectories that stay in $\{x \in \mathbb{R}^2 \mid x_1 \geq 0\}$.

The closed-loop system with the κ_P controller is given by

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -(k_p + k_c) x_1 + (-k_d - b_c) x_2 + k_p x_1^d \quad (4.13)$$

Note that the steady state of the position is given by $x_1^P := \frac{k_p}{k_p+k_c}x_1^d$. Let $z_P := \begin{bmatrix} z_{P1} \\ z_{P2} \end{bmatrix} = \begin{bmatrix} x_1 - x_1^P \\ x_2 \end{bmatrix}$ and $V_P : \mathbb{R}^2 \rightarrow \mathbb{R}_{\geq 0}$ be given by

$$V_P(z_P) = z_P^T P_P z_P = \frac{1}{2} z_P^T \begin{bmatrix} a_P & 0 \\ 0 & b_P \end{bmatrix} z_P$$

where $a_P > 0$ and $b_P > 0$. It follows that, if $\frac{a_P}{b_P} = k_p + k_c$,

$$\langle \nabla V_P(z_P), \dot{z}_P \rangle = -(k_d + b_c) b_P z_{P2}^2.$$

Hence, the equilibrium point $[x_1^P \ 0]^T$ is stable for the closed-loop system in (4.13). By Krasovskii-LaSalle's invariance principle, trajectories that stay in $\{x \in \mathbb{R}^2 \mid x_1 \geq 0\}$ converge to that point.

Similarly for the force controller, the closed-loop system with κ_F controller is given by

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -k_f k_c x_1 - k_f b_c x_2 + k_f f_c^d \quad (4.14)$$

Note that the steady state of the position is given by $x_1^F := \frac{f_c^d}{k_c}$. Let $z_F := \begin{bmatrix} z_{F1} \\ z_{F2} \end{bmatrix} = \begin{bmatrix} x_1 - x_1^F \\ x_2 \end{bmatrix}$. Given the positive definite diagonal matrix

$$P_o := \frac{1}{2} \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix},$$

let $a := p_1 \sin^2 \beta + p_2 \cos^2 \beta$, $b := p_1 \cos^2 \beta + p_2 \sin^2 \beta$, $c := (p_2 - p_1) \sin \beta \cos \beta$, $p_2 < p_1$ and define $V_F : \mathbb{R}^2 \rightarrow \mathbb{R}_{\geq 0}$ be given by

$$V_F(z) = z_F^T P_F z_F = \frac{1}{2} z_F^T \begin{bmatrix} a & c \\ c & b \end{bmatrix} z_F$$

where $P_F > 0$. Note that P_F corresponds to the matrix P_o rotated clockwise by β . It follows that

$$\langle \nabla V_F(z_F), \dot{z}_F \rangle = -z_F^T Q_F z_F.$$

The equilibrium point $[x_1^F \ 0]^T$ is asymptotically stable for the closed-loop system in (4.14) provided that $Q_F > 0$. Such a condition holds if $k_f > 0$ is chosen so that the determinant of Q_F is positive, and

$$\frac{c}{a} < \frac{b_c}{k_c}, \quad b_c c \neq k_c b,$$

which in terms of the parameters p_1 and p_2 in P_o yields

$$\frac{(p_2 - p_1) \sin \beta \cos \beta}{a_1 \sin^2 \beta + b_1 \cos^2 \beta} < \frac{b_c}{k_c} \quad (4.15)$$

$$b_c (p_2 - p_1) \sin \beta \cos \beta \neq k_c (a_1 \cos^2 \beta + b_1 \sin^2 \beta). \quad (4.16)$$

With the design of the Lyapunov function V_F above, we can design the threshold for jumps determined by γ_1 and γ_2 . We compute the maximum level set $L_{V_F}(r)$ that is

contained in the right-half plane in the x -coordinates. The value of the level set is given by

$$r = \frac{ab - c^2}{2b} (x_1^F)^2 = \frac{p_1 p_2}{2(p_1 \cos^2 \beta + p_2 \sin^2 \beta)} (x_1^F)^2. \quad (4.17)$$

In Figure 4.7, we depict the computation of the maximum level set. Note that the jump condition $f_c \geq \gamma_2$ in the jump set of the hybrid controller \mathcal{H}_c can explicitly expressed in terms of x_1 and x_2 by using the definition of the contact force in (4.4) which corresponds to a hyperplane in \mathbb{R}^2 as depicted in Figure 4.7. We also denote with $\underline{\ell}_{\gamma_2}$ the line which passes through the intersection of the r -level set of V_F and the $x_1 = 0$ axes. Since the jumps from position controller occur when the contact force is above the threshold γ_2 , this threshold defines a line in the right half plane of the x -coordinates which we denote by ℓ_{γ_2} . Note that $\underline{\ell}_{\gamma_2}$ and ℓ_{γ_2} are parallel. To guarantee that after a jump that triggers a transition from position to force controller the trajectories do not leave the right half plane, we design the threshold γ_2 so that the line ℓ_{γ_2} is above the line $\underline{\ell}_{\gamma_2}$. This implies

$$\gamma_2 \geq \frac{cb_c x_1^F}{b}$$

However, since a very large value of γ_2 would set the line ℓ_{γ_2} quite, it could be that the position controller is not able to generate such contact force and no jump that enables the force controller occurs. Therefore, we design the line ℓ_{γ_2} so that it is to the right of the line $\underline{\ell}_{\gamma_2}$ and so that it crosses the $x_2 = 0$ axes at the left of the point $\min\{x_1^P, x_1^F\}$. We denote with $\bar{\ell}_{\gamma_2}$ the line which passes through the $\min\{x_1^P, x_1^F\}$. In this way, the jump from position controller to force controller is always guaranteed and after the jump, the trajectories remain in the $L_{V_F}(r)$. It follows that

$$\gamma_2 \leq \frac{k_c k_p}{k_p + k_c} x_1^d$$

and to guarantee that the x_1^P is in a feasible location, it is required that x_1^d satisfies

$$x_1^d > \frac{cx_1^F b_c (k_p + k_c)}{k_p k_c b}.$$

For the design of the threshold γ_1 , we will consider lines ℓ_{γ_1} that are parallel to the line ℓ_{γ_2} and that do not intersect the r level set of the Lyapunov function V_F . For that purpose γ_1 has to satisfy:

$$\gamma_1 < k_c \left(x_1^F - \sqrt{\frac{2r}{p_1}} \right). \quad (4.18)$$

We denote with $\bar{\ell}_{\gamma_1}$ the line which is tangent to the level set of the Lyapunov function V_F and with $\underline{\ell}_{\gamma_1}$ the line which passes through the origin.

Figure 4.7 shows the plot of the maximum level of the Lyapunov function V_F , the lines $\underline{\ell}_{\gamma_2}$, $\bar{\ell}_{\gamma_2}$, $\underline{\ell}_{\gamma_1}$, $\bar{\ell}_{\gamma_1}$, the maximum admitted velocity of impact x_2^* and the steady state of the position controller x_1^P .

The general Lyapunov function $V : \mathbb{R}^2 \rightarrow \mathbb{R}_{\geq 0}$ for the hybrid closed loop system \mathcal{H}_{cl} is given by

$$V(x) = (q - 1)V_P + qV_F.$$

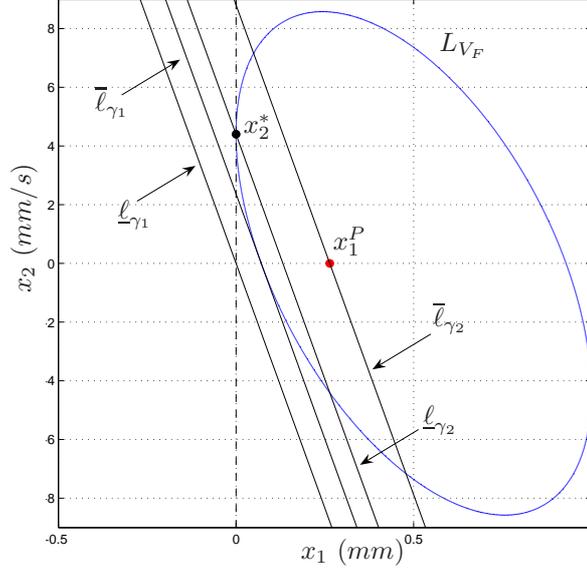


Figure 4.7: Computation of the maximum level set in the design of the switching strategy.

Since for every initial condition $x^0 \in K$,

$$\begin{aligned} \langle \nabla V(x), F(x) \rangle &\leq 0 \quad \forall x \in C \\ V(G(x)) - V(x) &\leq 0 \quad \forall x \in D, \end{aligned}$$

the solutions (x, q) to \mathcal{H}_{cl} approach to the compact set $\{x_1^F\} \times \{1\}$. By LaSalle's invariance principle, the compact set $\{x_1^F\} \times \{1\}$ is locally asymptotically stable for the hybrid closed loop system \mathcal{H}_{cl} .

Remark 4.5.2. *The compact set K is defined so that, for all the initial condition x_1^0, x_2^0 of the manipulator and for given parameters of the position controller, the manipulator will reach the surface of the environment with a bounded value of the impact velocity, denoted by x_2^* and identified, as maximum value, by f_c^d/b_c . This compact set can be enlarged by more sophisticated position control design that are described in Section 4.7.*

In particular, the compact set K is depicted in Figure 4.8.

4.5.1 Margin of Robustness

The margin of robustness to measurement noise obtained with our control strategy is defined as the minimum distance between the set L_{V_F} and ℓ_{γ_1} , and it is denoted with the parameter $\varepsilon > 0$, as shown in Figure 4.9.

If a desired margin of robustness ε is specified, one can design the line ℓ_{γ_1} so that it is at least a ε -apart from the $\bar{\ell}_{\gamma_1}$. It follows that, to accomplish the desired margin of robustness, instead of (4.18), γ_1 needs to satisfy

$$\gamma_1 \leq k_c \left(x_1^F - \sqrt{\frac{2r}{p_1}} - \varepsilon \right).$$

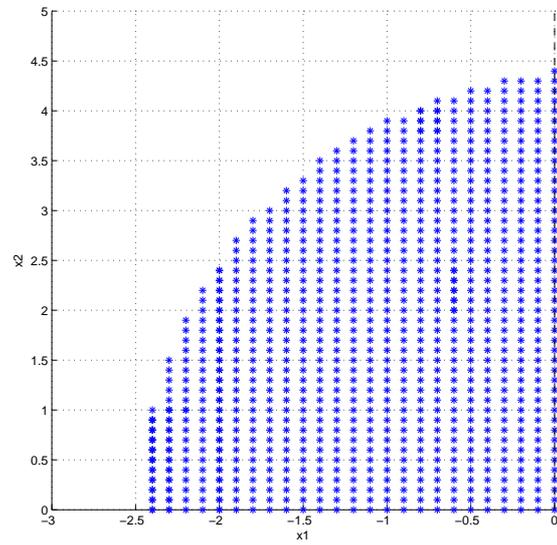


Figure 4.8: Compact set of initial conditions K .

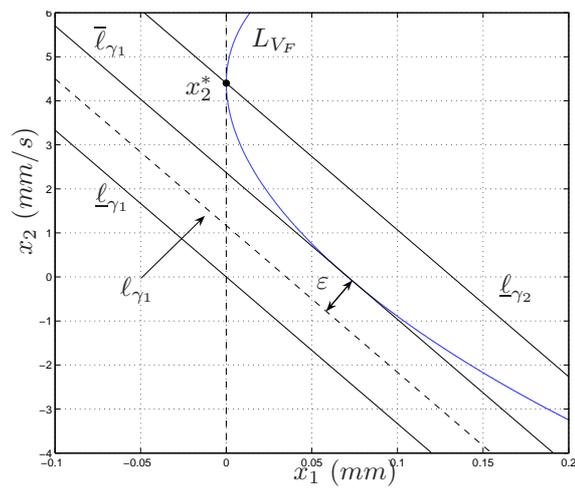


Figure 4.9: Margin of robustness ε of the switching strategy.

Moreover, since $\gamma_1 > 0$, in order to guarantee that this margin of robustness is feasible, it is required that $k_c \left(x_1^F - \sqrt{\frac{2r}{p_1}} - \varepsilon \right) > 0$. This is granted if the parameters of the controller are designed so that

$$(x_1^F)^2 p_2 < (x_1^F - \varepsilon)^2 (p_1 \cos^2 \beta + p_2 \sin^2 \beta).$$

The maximum margin of robustness possible is given by:

$$\varepsilon_{max} = x_1^F - \sqrt{\frac{2r}{p_1}} - \frac{\gamma_1}{k_c}$$

where, as in (4.17), r is a function of p_1, p_2 .

4.6 Simulations

In this section, we describe the simulations of the presented hybrid control architecture by going through the steps of the design of the control algorithm:

- Step 1)** Given the viscoelastic parameters k_c, b_c , by using (4.11), it is possible to compute the parameters θ and β , that decide the slope of the ℓ -lines;
- Step 2)** From the force set-point f_c^d , the force controller steady state x_1^F is also fixed from the first of (4.12);
- Step 3)** Build the maximum level set of the Lyapunov function V_F all contained in the right half plane, we have degrees of freedom in the choice of p_1 and p_2 , unless of the constraints (4.15) and (4.16); then, once p_1 and p_2 are chosen, we can compute the value r of the level set by using the second of (4.12). This guarantee that the parameters a, b and c are fixed and also the positive gain of the force control k_f ;
- Step 4)** From the analysis of the position controller and, according to (4.8), pick parameters k_p, k_d and x_1^d ;
- Step 5)** Choose γ_1 and γ_2 according to (4.9) and (4.10).

The simulation results are shown in the following. In particular, in the interaction task we are going to simulate, we suppose that, at the beginning of the task, the manipulator is in the free space at an initial position $x_1^0 = -1mm$ and initial velocity $x_2^0 = 0mm/s$ with respect to the point to be touched in the environment, that is considered at $x_1 = 0$. The desired force we want to apply is $f_c^d = 5N$. Moreover, we suppose to deal with a soft material characterized by stiffness parameter $k_c = 10N/mm$ and damping term $b_c = 0.3Ns/mm$. We will consider later the case of stiff and very stiff materials.

Figure 4.10, and its zoom in Figure 4.11, illustrate the trajectory of the system in the phase diagram. As it can be noticed, the position control makes the trajectory hitting the environment at a velocity that is below the bound x_2^* imposed by the material. The manipulator is driven by the position control until the measured contact force is equal to the threshold γ_2 . At this point, the controller switches to the force control without bouncing outside the environment, since the γ_1 line has been designed outside the Lyapunov level set of the closed-loop system with the force controller. At this point, the velocity of the system increases due to the switch between the two controllers; however the velocity

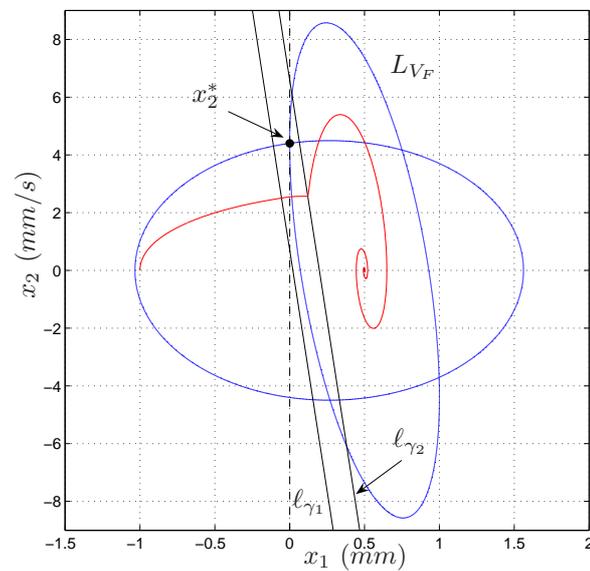


Figure 4.10: Phase diagram of the switching strategy. The plot depicts the Lyapunov functions level sets of the position and force controller, the l_{γ_1} and l_{γ_2} lines, and the trajectory of the system. The controller is able to avoid the bouncing off of the robotic system.

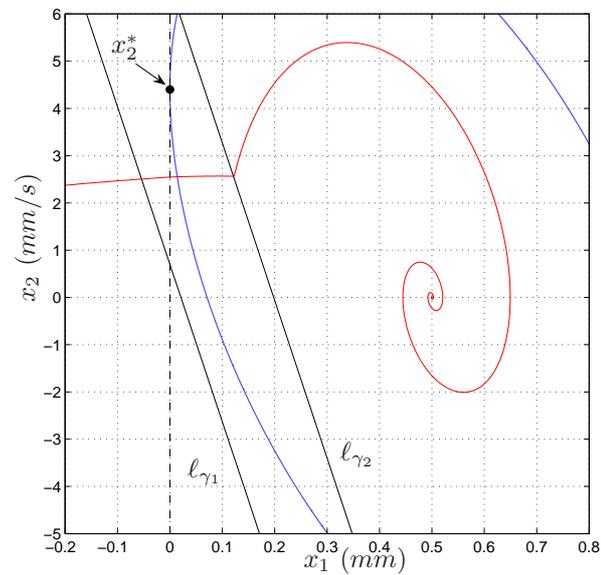


Figure 4.11: Phase diagram of the switching strategy. Zoom of Figure 4.10.

remains bounded because the trajectory is constrained inside the designed level set L_{V_F} . Finally, the steady state x_1^F is reached without chattering.

Note that the initial position x_1^0 can be greater than the one considered in the simulation. In particular, for this example, x_1^0 can be chosen not below $-15mm$.

Figure 4.12 presents the measurement of the interaction forces during the task. We can notice that the involved forces remain bounded. On the top, the plot of the desired contact force f_c^d and the logic variable q is added: once the contact is detected the variable q switches to 1 and, since the contact is never released, it never goes down to 0.

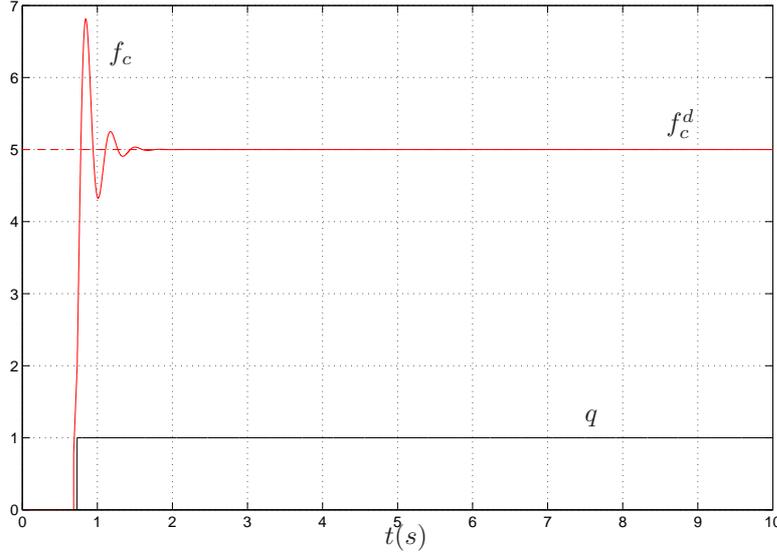


Figure 4.12: Contact forces during the interaction task. The plot depicts the interaction forces f_c , the desired contact force f_c^d and the logic variable q .

4.6.1 Robustness

In order to test the robustness of the architecture design, we can add in our simulation noise in the force sensor and errors in the measurement of the manipulator position.

Figure 4.13 depicts in dashed line the trajectory of the system without noise/errors and in continuous line the trajectories of the system with different values of noise/errors. In particular, we have added a Gaussian noise in both sensor and manipulator position measurements with null mean value and variation of $\sigma = 0.01$, $\sigma = 0.5$, $\sigma = 1$, $\sigma = 2$. It is easy to notice that the system continues to converge to the steady state x_1^F without bouncing off.

4.6.2 Change of Material

In Figure 4.14, we illustrate how the strategy is working for different kind of material, i.e. while varying the parameter k_c from soft to stiff and very stiff materials. In particular, as the value of k_c is increasing, the trajectory becomes more flattened to the surface of the environment, but the algorithm is still guaranteeing good performances.

Table 4.1 shows how the design parameter changes for different values of the environment material stiffness. We can notice that in order to avoid bounces, as the stiffness

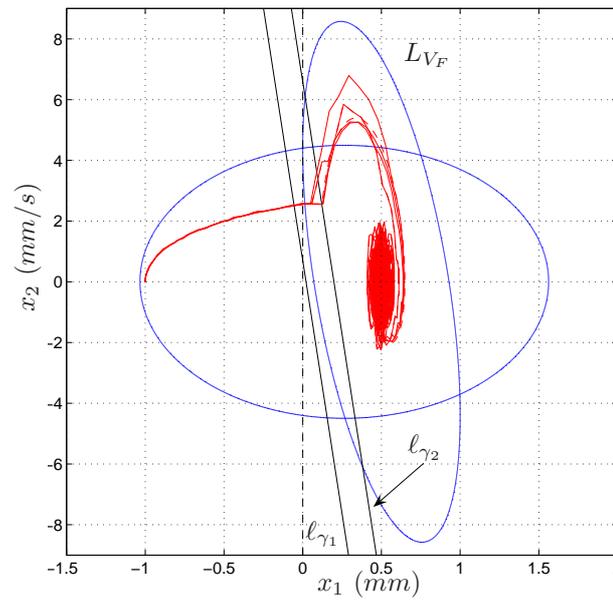


Figure 4.13: Phase diagram of the switching strategy in presence of sensor noise and measurement errors in the manipulator position. The plot depicts the Lyapunov functions level sets of the position and force controllers, the l_{γ_1} and l_{γ_2} lines, and the trajectory of the system both without noise (dashed line) and different values of noise (continuous lines).

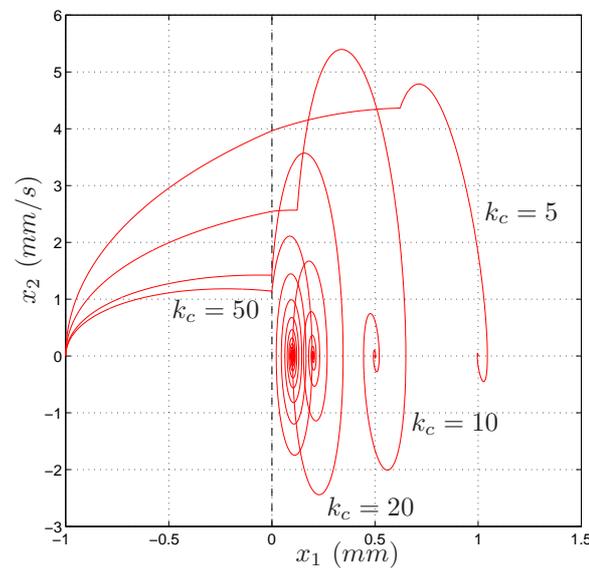


Figure 4.14: Phase diagram of the switching strategy. The plot depicts the different trajectories of the system for values of the environment material stiffness equal to 5 N/mm, 10 N/mm, 25 N/mm and 50 N/mm. The steady state point is changing since $x_1^F = f_c^d / k_c$, in which the desired force is fixed at $f_c^d = 5$ N.

of the environment is increasing, the desired position and the admitted impact velocity decrease. Moreover, for fixed environment parameters, there is a trade-off between the maximum admitted impact velocity and the control gain k_f of the force controller: in particular, in order to have a large impact velocity, a large gain is required.

Table 4.1: Design parameters of the controller for different values of the environment material stiffness k_c and fixed damping term b_c .

k_c (N/mm)	x_1^d (mm)	x_2^* (mm/s)	k_f
1	14.55821	16.1757	430
5	4.1208	9.8115	80
10	0.2934	0.9054	16
20	0.4530	1.3727	20
50	0.0736	0.2360	8
100	0.0182	0.0596	4
200	0.0045	0.0149	2
500	0.0007	0.0024	0.8

4.7 Controller Design with Saturation

4.7.1 Position Controller with Saturation

Following the discussion in Section 4.3, one of the properties that we want the position controller to confer to the closed-loop is a bounded impact velocity. Unfortunately, the position controller in (4.5) does not guarantee such a property for the closed-loop system. Using techniques in [73], we extend the control law κ_P with a saturation function to guarantee boundedness of the impact velocity. The resulting position control law is given by

$$\tilde{\kappa}_P(x, x_1^d) = \text{sat}(\kappa_P(x, x_1^d)) - kx_2$$

for every $k \in \mathbb{R}_{>0}$ and for every $s \in \mathbb{R}$, $\text{sat}(s) = \text{sign}(s) \min\{|s|, u_P\}$, where u_P is the saturation level.

The closed-loop system with the $\tilde{\kappa}_P$ controller is given by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -k_c x_1 - (b_c + k)x_2 + \text{sat}(-k_p(x_1 - x_1^d) - k_d x_2) \end{aligned} \quad (4.19)$$

Analysis of the Closed-loop System with the Saturated Position Controller

From the closed-loop system described by Equations (4.19) we can observe that, if the saturation level $u_P \leq \frac{k_c k_p}{k_c + k_p} x_1^d$, the region of stability is given by $[x_1^P \ 0]^T$, where $x_1^P \leq \frac{k_p}{k_c + k_p} x_1^d$. On the other hand, if the saturation level $u_P > \frac{k_c k_p}{k_c + k_p} x_1^d$, the region of stability

is given by $[x_1^P \ 0]^T$, where $x_1^P = \frac{k_p}{k_c+k_p}x_1^d$. This means that, in general, the region of stability of the closed-loop system with the saturated position controller is given by:

$$[x_1^P \ 0]^T, \text{ where } x_1^P \leq \frac{k_p}{k_c+k_p}x_1^d.$$

In order to analyze the stability of the closed-loop system (4.19), let's proceed with a change of coordinates. Let be $w_P := \begin{bmatrix} w_{P1} \\ w_{P2} \end{bmatrix} = \begin{bmatrix} x_1 - x_1^d \\ k_p(x_1 - x_1^d) + k_d x_2 \end{bmatrix}$, then system (4.19) becomes:

$$\begin{aligned} \dot{w}_{P1} &= -\frac{k_p}{k_d}w_{P1} + \frac{1}{k_d}w_{P2} \\ \dot{w}_{P2} &= \left(-\frac{k_p^2}{k_d} - k_c k_d + k_p(k + b_c)\right)w_{P1} + \left(\frac{k_p}{k_d} - k - b_c\right)w_{P2} - k_d \text{sat}(w_{P2}) - k_c k_d x_1^d \end{aligned} \quad (4.20)$$

Let be:

$$\begin{aligned} k_{1P} &= \frac{k_p}{k_d} \\ k_{2P} &= \frac{1}{k_d} \\ k_{3P} &= \frac{k_p^2}{k_d} + k_c k_d - k_p(k + b_c) \\ k_{4P} &= \frac{k_p}{k_d} - k - b_c \\ k_{5P} &= k_c k_d x_1^d, \end{aligned}$$

then system (4.20) becomes:

$$\begin{aligned} \dot{w}_{P1} &= -k_{1P}w_{P1} + k_{2P}w_{P2} \\ \dot{w}_{P2} &= -k_{3P}w_{P1} + k_{4P}w_{P2} - \frac{1}{k_{2P}}\text{sat}(w_{P2}) - k_{5P} \end{aligned} \quad (4.21)$$

4.7.2 Force Controller with Saturation

Following the discussion in Section 4.3, one of the properties that we want to confer to the closed-loop is to grant a bounded control input. Unfortunately, due to the switch between the position and the force controller, the control input can be high and it can have dangerous consequences for the mechanical system. In the same way as discussed in 4.7.1, we can extend the control law κ_F in (4.6) with a saturation function to guarantee boundedness of the control signal. The resulting force control law is given by

$$\tilde{\kappa}_F(f_c, f_c^d) = \text{sat}(\kappa_F(f_c, f_c^d))$$

for every $s \in \mathbb{R}$, $\text{sat}(s) = \text{sign}(s) \min\{|s|, u_F\}$, where u_F is the saturation level.

The closed-loop system with the $\tilde{\kappa}_F$ controller is given by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -k_c x_1 - b_c x_2 + \text{sat}(k_c x_1 + b_c x_2 + k_f(f_c^d - k_c x_1 - b_c x_2)) \end{aligned} \quad (4.22)$$

Analysis of the Closed-loop System with the Saturated Force Controller

From the closed-loop system described by Equations (4.22) we can observe that, with the condition that the desired force f_c^d is below the the saturation level u_F ($f_c^d \leq u_F$), the closed-loop system with the saturated force controller has a steady state given by:

$$[x_1^F \ 0]^T, \text{ where } x_1^F = \frac{f_c^d}{k_c}.$$

In order to analyze the stability of the system, let's proceed with a change of coordinates. Let be $w_F := \begin{bmatrix} w_{F1} \\ w_{F2} \end{bmatrix} = \begin{bmatrix} x_1 \\ k_c(1 - k_f)x_1 + b_c(1 - k_f)x_2 + k_f f_c^d \end{bmatrix}$, then system (4.22) becomes:

$$\begin{aligned} \dot{w}_{F1} &= -\frac{k_c}{b_c}w_{F1} + \frac{1}{b_c(1-k_f)}w_{F2} - \frac{k_f f_c^d}{b_c(1-k_f)} \\ \dot{w}_{F2} &= -\frac{k_c^2}{b_c}(1 - k_f)w_{F1} + \left(\frac{k_c}{b_c} - b_c\right)w_{F2} + b_c(1 - k_f)\text{sat}(w_{F2}) + \left(b_c - \frac{k_c}{b_c}\right)k_f f_c^d \end{aligned} \quad (4.23)$$

Let be:

$$\begin{aligned} k_{1F} &= \frac{k_c}{b_c} \\ k_{2F} &= -\frac{1}{b_c(1-k_f)} \\ k_{3F} &= -\frac{k_c^2}{b_c}(1 - k_f) \\ k_{4F} &= \frac{k_c}{b_c} - b_c \\ k_{5F} &= k_f f_c^d k_{2F} \\ k_{6F} &= -k_f f_c^d k_{4F} \end{aligned}$$

then system (4.23) becomes:

$$\begin{aligned} \dot{w}_{F1} &= -k_{1F}w_{F1} - k_{2F}w_{F2} + k_{5F} \\ \dot{w}_{F2} &= k_{3F}w_{F1} + k_{4F}w_{F2} - \frac{1}{k_{2F}}\text{sat}(w_{F2}) + k_{6F} \end{aligned} \quad (4.24)$$

4.7.3 Controller Design

We now present a result that guarantees the existence of parameters for the hybrid controller \mathcal{H}_c for contact detection. We go through the steps of the proof since they highlight the design of our hybrid control strategy.

Theorem 4.7.1. *Given parameters k_c, b_c of the work environment and desired contact force $f_c^d < \hat{f}_c$, there exist*

- A compact set $\tilde{K} \subset \mathbb{R}^2$ of initial conditions of the manipulator;
- Parameters $k_p, k_d, k, k_f, \gamma_1, \gamma_2$ of the hybrid controller \mathcal{H}_c ;
- Set-point x_1^d for the regulation of the position;

such that for every initial condition $x^0 \in K$, the solutions (x, q) to \mathcal{H}_{cl} approach the compact set $\{x_1^d\} \times \{1\}$. Moreover, the parameters satisfy

$$\begin{aligned} k_p, k_d, k &> 0 \\ k_f &> 1 \frac{k_p^2}{k_d} + k_c k_d - k_p(k + b_c) > 0 \quad (\text{i.e. } k_{3P} > 0) \\ \sqrt{\frac{1}{k_d} \left(\frac{k_p^2}{k_d} + k_c k_d - k_p(k + b_c) \right)} - \frac{k_p}{k_d} + k + b_c &> 0 \quad (\text{i.e. } \sqrt{k_{2P} k_{3P}} - k_{4P} > 0) \end{aligned}$$

We proceed by constructing a Lyapunov function for the system (4.20) resulting when the position controller is in the loop and we start the analysis by considering the unforced closed-loop system. This Lyapunov function decreases along trajectories that stay in $\{x \in \mathbb{R}^2 \mid x_1 \geq 0\}$.

Let $V_P : \mathbb{R}^2 \rightarrow \mathbb{R}_{\geq 0}$ be given by

$$V_P(w_P) = \frac{1}{2} w_P^T P_P w_P + k_{2P} \int_0^{w_{P2}} \text{sat}(y) dy = \frac{1}{2} w_P^T \begin{bmatrix} a_P & -c_P \\ -c_P & b_P \end{bmatrix} w_P + k_{2P} \int_0^{w_{P2}} \text{sat}(y) dy.$$

where:

$$\begin{aligned} a_P &= 2k_{1P}k_{4P}^2 - 4k_{1P}k_{4P}\sqrt{k_{2P}k_{3P}} + k_{1P}k_{2P}k_{3P} + 2k_{2P}k_{3P}\sqrt{k_{2P}k_{3P}} + \\ &\quad 2(k_{4P} - \sqrt{k_{2P}k_{3P}})\sqrt{k_{1P}^2k_{4P}^2 - k_{2P}^2k_{3P}^2 - 2k_{1P}k_{4P}\sqrt{k_{2P}k_{3P}} + 2k_{1P}k_{2P}k_{3P}\sqrt{k_{2P}k_{3P}}} > 0 \\ b_P &= k_{2P}^2(2\sqrt{k_{2P}k_{3P}} - k_{4P}) > 0 \\ c_P &= k_{2P}^2k_{3P} > 0. \end{aligned}$$

Remark 4.7.2. The variable a_P , b_P and c_P can be also written as:

$$\begin{aligned} a_P &= \frac{1}{k_{1P}}(c_1^2 + k_{3P}c_P) \\ b_P &= \frac{1}{k_{4P}}(-c_2^2 + k_{2P}c_P) \\ c_P &= k_{2P}^2k_{3P} \end{aligned}$$

with $c_1 < 0$ and $c_2 > 0$ given by:

$$\begin{aligned} c_1^2 &= \left(k_{1P}k_{4P} - k_{1P}\sqrt{k_{2P}k_{3P}} + \sqrt{k_{1P}^2k_{4P}^2 - k_{2P}^2k_{3P}^2 - 2k_{1P}k_{4P}\sqrt{k_{2P}k_{3P}} + 2k_{1P}k_{2P}k_{3P}\sqrt{k_{2P}k_{3P}}} \right)^2 \\ c_2^2 &= k_{2P}^2(\sqrt{k_{2P}k_{3P}} - k_{4P})^2 \\ c_1 &= -(k_{1P}k_{4P} - k_{1P}\sqrt{k_{2P}k_{3P}} + \sqrt{k_{1P}^2k_{4P}^2 - k_{2P}^2k_{3P}^2 - 2k_{1P}k_{4P}\sqrt{k_{2P}k_{3P}} + 2k_{1P}k_{2P}k_{3P}\sqrt{k_{2P}k_{3P}}}) \\ c_2 &= k_{2P}(\sqrt{k_{2P}k_{3P}} - k_{4P}). \end{aligned}$$

It follows that the derivative of V_P along the trajectories of the system is given by:

$$\begin{aligned} \langle \nabla V_P(w_P), \dot{w}_P \rangle &= (-k_{1P}a_P + k_{3P}c_P)w_{P1}^2 + (k_{4P}b_P - k_{2P}c_P)w_{P2}^2 \\ &\quad + (-k_{3P}b_P + k_{2P}a_P + k_{1P}c_P - k_{4P}c_P)w_{P1}w_{P2} \\ &\quad + \left(\frac{c_P}{k_{2P}} - k_{2P}k_{3P}\right)w_{P1}\text{sat}(w_{P2}) + \left(-\frac{b_P}{k_{2P}} + k_{2P}k_{4P}\right)w_{P2}\text{sat}(w_{P2}) - \text{sat}^2(w_{P2}) \\ &= -(c_1w_{P1} - c_2w_{P2})^2 - 2k_{2P}(\sqrt{k_{2P}k_{3P}} - k_{4P})w_{P2}\text{sat}(w_{P2}) - \text{sat}^2(w_{P2}) \leq 0. \end{aligned}$$

Since $\dot{V}_P \leq 0$, we can assert that, by Krasovskii-LaSalle's invariance principle, the equilibrium point $[x_1^P \ 0]^T$ is globally asymptotically stable for the unforced system corresponding to (4.19).

Let's now come back to the complete forced system (4.20). We proceed now by using the same Lyapunov function V_P construed for the unforced closed-loop system. This Lyapunov function decreases along trajectories.

The derivative of V_P along the trajectories of the system is given by:

$$\begin{aligned} \langle \nabla V_P(w_P), \dot{w}_P \rangle &= -(c_1w_{P1} - c_2w_{P2})^2 - 2k_{2P}(\sqrt{k_{2P}k_{3P}} - k_{4P})w_{P2}\text{sat}(w_{P2}) - \text{sat}^2(w_{P2}) \\ &\quad - k_{5P}(-c_Pw_{P1} + b_Pw_{P2} + k_{2P}\text{sat}(w_{P2})) \end{aligned}$$

Let be $\theta_P \in (0, 1)$, it follows that:

$$\begin{aligned}
\langle \nabla V_P(w_P), \dot{w}_P \rangle &= -(1 - \theta_P)(c_1 w_{P1} - c_2 w_{P2})^2 - \theta_P(c_1 w_{P1} - c_2 w_{P2})^2 \\
&\quad - 2k_{2P}(\sqrt{k_{2P}k_{3P}} - k_{4P})w_{P2}\text{sat}(w_{P2}) - \text{sat}^2(w_{P2}) \\
&\quad - k_{5P}(-c_P w_{P1} + b_P w_{P2} + k_{2P}\text{sat}(w_{P2})) \\
&= -(1 - \theta_P)(c_1 w_{P1} - c_2 w_{P2})^2 - \theta_P c_1^2 w_{P1}^2 - \theta_P c_2^2 w_{P2}^2 + 2\theta_P c_1 c_2 w_{P1} w_{P2} \\
&\quad - 2k_{2P}(\sqrt{k_{2P}k_{3P}} - k_{4P})w_{P2}\text{sat}(w_{P2}) - \text{sat}^2(w_{P2}) \\
&\quad - k_{5P}(-c_P w_{P1} + b_P w_{P2} + k_{2P}\text{sat}(w_{P2}))
\end{aligned}$$

By using the Young inequality, $2w_{P1}w_{P2} \leq w_{P1}^2 + w_{P2}^2$, we can state:

$$\begin{aligned}
\langle \nabla V_P(w_P), \dot{w}_P \rangle &\leq -(1 - \theta_P)(c_1 w_{P1} - c_2 w_{P2})^2 - \theta_P c_1^2 w_{P1}^2 - \theta_P c_2^2 w_{P2}^2 + 2\theta_P c_1 c_2 (w_{P1}^2 + w_{P2}^2) \\
&\quad - 2k_{2P}(\sqrt{k_{2P}k_{3P}} - k_{4P})w_{P2}\text{sat}(w_{P2}) - \text{sat}^2(w_{P2}) \\
&\quad - k_{5P}(-c_P w_{P1} + b_P w_{P2} + k_{2P}\text{sat}(w_{P2})) \\
&= -(1 - \theta_P)(c_1 w_{P1} - c_2 w_{P2})^2 \\
&\quad + ((-\theta_P c_1^2 + 2c_1 c_2 \theta_P)w_{P1}^2 + k_{5P}c_P w_{P1}) \\
&\quad + ((-\theta_P c_2^2 + 2c_1 c_2 \theta_P)w_{P2}^2 - k_{5P}b_P w_{P2}) \\
&\quad - 2k_{2P}(\sqrt{k_{2P}k_{3P}} - k_{4P})w_{P2}\text{sat}(w_{P2}) - \text{sat}^2(w_{P2}) - k_{2P}k_{5P}\text{sat}(w_{P2}).
\end{aligned}$$

Let's define

$$\bar{w}_{P1} := \frac{k_{5P}c_P}{\theta_P |c_1^2 - 2c_1 c_2|}, \quad \bar{w}_{P2} := \frac{k_{5P}b_P}{\theta_P |c_2^2 - 2c_1 c_2|},$$

if

$$|w_{P1}| \geq \bar{w}_{P1} \quad \text{and} \quad |w_{P2}| \geq \bar{w}_{P2}, \quad (4.25)$$

it follows that

$$\begin{aligned}
\langle \nabla V_P(w_P), \dot{w}_P \rangle &\leq -(1 - \theta_P)(c_1 w_{P1} - c_2 w_{P2})^2 - 2k_{2P}(\sqrt{k_{2P}k_{3P}} - k_{4P})w_{P2}\text{sat}(w_{P2}) \\
&\quad - \text{sat}^2(w_{P2}) - k_{2P}k_{5P}\text{sat}(w_{P2}).
\end{aligned}$$

Finally, if

$$-\text{sat}^2(\bar{w}_{P2}) - k_{2P}k_{5P}\text{sat}(\bar{w}_{P2}) \leq 0, \quad (4.26)$$

$$\langle \nabla V_P(w_P), \dot{w}_P \rangle \leq 0.$$

Since $\dot{V}_P \leq 0$, we can assert that the equilibrium point $[x_1^P \ 0]^T$ is input-to-state stable for the forced system (4.19) with the conditions (4.25) and (4.26).

We can observe here that if $\bar{w}_{P2} \geq u_P$ the system is not allowed to work in the linear zone. Therefore, we add the condition

$$\bar{w}_{P2} \leq u_P. \quad (4.27)$$

This implies that \bar{w}_{P2} belongs to the linear zone of the system and the constraint (4.26) can be written as

$$\bar{w}_{P2} \geq k_{2P}k_{5P}. \quad (4.28)$$

At this point, we can state that the equilibrium point $[x_1^P \ 0]^T$ is input-to-state stable for the forced system (4.19) with the conditions (4.25), (4.27) and (4.28).

Remark 4.7.3. In order to better understand the conditions that guarantee the input-to-state stability of the forced system (4.19), we can observe that, according the evolution of the system and according the position of the reference frame in Figure 4.1, the robotic manipulator is approaching the work environment from a negative value of the position. This implies that the first constraint in Equations (4.25), can be simplified in:

$$w_{P1} \leq -\bar{w}_{P1}. \quad (4.29)$$

In particular, if the constraint in (4.29) is valid, we can describe in Figure 4.15 the other constraints. In particular, the figure depicts that, when the system is working in the saturation zone, the system is always input-to-state stable. However, when the system is working in the linear zone, the system can be unstable.

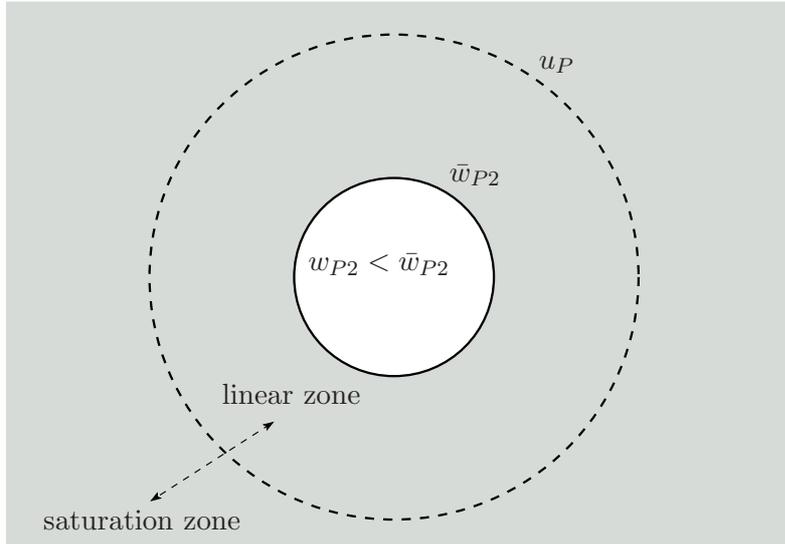


Figure 4.15: Description of the constraints for the input-to-state stability of the system (4.19). If the first constraint in Equations (4.25) is valid, the gray zone corresponds to the input-to-state stable area.

Let's analyse the system working in the linear zone, that is when $|k_p(x_1 - x_1^d) + k_d x_2| \leq u_P$. In particular, the system (4.19) becomes:

$$\begin{aligned} \dot{w}_{P1} &= -k_{1P}w_{P1} + k_{2P}w_{P2} \\ \dot{w}_{P2} &= -k_{3P}w_{P1} + \left(k_{4P} - \frac{1}{k_{2P}}\right)w_{P2} - k_{5P} \end{aligned}$$

If we analyze the Lyapunov function $V_P(w_P)$ when the system is in the linear zone, we find out that the derivative of V_P along the trajectories of the system is always negative.

We can conclude that the equilibrium point $[x_1^P \ 0]^T$ is input-to-state stable for the system (4.19) if

$$w_{P1} \leq -\bar{w}_{P1} \quad \text{and} \quad k_{2P}k_{5P} \leq \bar{w}_{P2} \leq u_P.$$

From these inequalities, it follows that the parameter θ_P has to satisfy the constraint

$$\theta_P < \min\left\{1, \frac{b_P}{k_{2P} |c_2^2 - 2c_1c_2|}\right\}.$$

We proceed by constructing a Lyapunov function for the system (4.24) resulting when the force controller is in the loop and we start the analysis by considering the unforced closed-loop system. This Lyapunov function decreases along trajectories.

Let $V_F : \mathbb{R}^2 \rightarrow \mathbb{R}_{\geq 0}$ be given by

$$V_F(w_F) = \frac{1}{2}w_F^T P_F w_F + k_{2F} \int_0^{w_{F2}} \text{sat}(y) dy = \frac{1}{2}w_F^T \begin{bmatrix} a_F & c_F \\ c_F & b_F \end{bmatrix} w_F + k_{2F} \int_0^{w_{F2}} \text{sat}(y) dy.$$

where:

$$\begin{aligned} a_F &= \frac{2k_{1F}k_{4F}^2 - 4k_{1F}k_{4F}\sqrt{k_{2F}k_{3F}} + k_{1F}k_{2F}k_{3F} + 2k_{2F}k_{3F}\sqrt{k_{2F}k_{3F}} +}{2(k_{4F} - \sqrt{k_{2F}k_{3F}})\sqrt{k_{1F}^2k_{4F}^2 - k_{2F}^2k_{3F}^2 - 2k_{1F}^2k_{4F}\sqrt{k_{2F}k_{3F}} + 2k_{1F}k_{2F}k_{3F}\sqrt{k_{2F}k_{3F}}} > 0 \\ b_F &= k_{2F}^2(2\sqrt{k_{2F}k_{3F}} - k_{4F}) > 0 \\ c_F &= k_{2F}^2k_{3F} > 0. \end{aligned}$$

Remark 4.7.4. The variable a_F , b_F and c_F can be also written as:

$$\begin{aligned} a_F &= \frac{1}{k_{1F}}(c_3^2 + k_{3F}c_F) \\ b_F &= \frac{1}{k_{4F}}(-c_4^2 + k_{2F}c_F) \\ c_F &= k_{2F}^2k_{3F} \end{aligned}$$

with $c_3 < 0$ and $c_4 > 0$ given by:

$$\begin{aligned} c_3^2 &= \left(k_{1F}k_{4F} - k_{1F}\sqrt{k_{2F}k_{3F}} + \sqrt{k_{1F}^2k_{4F}^2 - k_{2F}^2k_{3F}^2 - 2k_{1F}^2k_{4F}\sqrt{k_{2F}k_{3F}} + 2k_{1F}k_{2F}k_{3F}\sqrt{k_{2F}k_{3F}}} \right)^2 \\ c_4^2 &= k_{2F}^2(\sqrt{k_{2F}k_{3F}} - k_{4F})^2 \\ c_3 &= -(k_{1F}k_{4F} - k_{1F}\sqrt{k_{2F}k_{3F}} + \sqrt{k_{1F}^2k_{4F}^2 - k_{2F}^2k_{3F}^2 - 2k_{1F}^2k_{4F}\sqrt{k_{2F}k_{3F}} + 2k_{1F}k_{2F}k_{3F}\sqrt{k_{2F}k_{3F}}) \\ c_4 &= k_{2F}(\sqrt{k_{2F}k_{3F}} - k_{4F}). \end{aligned}$$

It follows that the derivative of V_F along the trajectories of the system is given by:

$$\begin{aligned} \langle \nabla V_F(w_F), \dot{w}_F \rangle &= (-k_{1F}a_F + k_{3F}c_F)w_{F1}^2 + (k_{4F}b_F - k_{2F}c_F)w_{F2}^2 \\ &\quad + (k_{3F}b_F - k_{2F}a_F - k_{1F}c_F + k_{4F}c_F)w_{F1}w_{F2} \\ &\quad + \left(-\frac{c_F}{k_{2F}} + k_{2F}k_{3F}\right)w_{F1}\text{sat}(w_{F2}) + \left(-\frac{b_F}{k_{2F}} + k_{2F}k_{4F}\right)w_{F2}\text{sat}(w_{F2}) - \text{sat}^2(w_{F2}) \\ &= -(c_3w_{F1} - c_4w_{F2})^2 - 2k_{2F}(\sqrt{k_{2F}k_{3F}} - k_{4F})w_{F2}\text{sat}(w_{F2}) - \text{sat}^2(w_{F2}) \leq 0. \end{aligned}$$

Since $\dot{V}_F < 0$, we can assert that, by Krasovskii-LaSalle's invariance principle, the equilibrium point $[x_1^F \ 0]^T$ is globally asymptotically stable for the unforced system corresponding to (4.22).

Let's now come back to the complete forced system (4.24), we proceed now by using the same Lyapunov function V_F construed for the unforced closed-loop system. This Lyapunov function decreases along trajectories.

Let $V_F : \mathbb{R}^2 \rightarrow \mathbb{R}_{\geq 0}$ be given by

$$V_F(w_F) = \frac{1}{2}w_F^T P_F w_F + k_{2F} \int_0^{w_{F2}} \text{sat}(y) dy = \frac{1}{2}w_F^T \begin{bmatrix} a_F & c_F \\ c_F & b_F \end{bmatrix} w_F + k_{2F} \int_0^{w_{F2}} \text{sat}(y) dy.$$

It follows that the derivative of V_F along the trajectories of the system is given by:

$$\begin{aligned} \langle \nabla V_F(w_F), \dot{w}_F \rangle &= -(c_3 w_{F1} - c_4 w_{F2})^2 - 2k_{2F}(\sqrt{k_{2F}k_{3F}} - k_{4F})w_{F2}\text{sat}(w_{F2}) - \text{sat}^2(w_{F2}) \\ &\quad + (k_{5F}a_F + k_{6F}c_F)w_{F1} + (k_{6F}b_F + k_{5F}c_F)w_{F2} + k_{2F}k_{6F}\text{sat}(w_{F2}) \end{aligned}$$

Let be $\theta_F \in (0, 1)$, it follows that:

$$\begin{aligned} \langle \nabla V_F(w_F), \dot{w}_F \rangle &= -(1 - \theta_F)(c_3 w_{F1} - c_4 w_{F2})^2 - \theta_F(c_3 w_{F1} - c_4 w_{F2})^2 \\ &\quad - 2k_{2F}(\sqrt{k_{2F}k_{3F}} - k_{4F})w_{F2}\text{sat}(w_{F2}) - \text{sat}^2(w_{F2}) \\ &\quad + (k_{5F}a_F + k_{6F}c_F)w_{F1} + (k_{6F}b_F + k_{5F}c_F)w_{F2} + k_{2F}k_{6F}\text{sat}(w_{F2}) \\ &= -(1 - \theta_F)(c_3 w_{F1} - c_4 w_{F2})^2 - \theta_F c_3^2 w_{F1}^2 - \theta_F c_4^2 w_{F2}^2 + 2\theta_F c_3 c_4 w_{F1} w_{F2} \\ &\quad - 2k_{2F}(\sqrt{k_{2F}k_{3F}} - k_{4F})w_{F2}\text{sat}(w_{F2}) - \text{sat}^2(w_{F2}) \\ &\quad + (k_{5F}a_F + k_{6F}c_F)w_{F1} + (k_{6F}b_F + k_{5F}c_F)w_{F2} + k_{2F}k_{6F}\text{sat}(w_{F2}) \end{aligned}$$

By using the Young inequality, $2w_{F1}w_{F2} \leq w_{F1}^2 + w_{F2}^2$, we can state:

$$\begin{aligned} \langle \nabla V_F(w_F), \dot{w}_F \rangle &\leq -(1 - \theta_F)(c_3 w_{F1} - c_4 w_{F2})^2 - \theta_F c_3^2 w_{F1}^2 - \theta_F c_4^2 w_{F2}^2 + 2\theta_F c_3 c_4 (w_{F1}^2 + w_{F2}^2) \\ &\quad - 2k_{2F}(\sqrt{k_{2F}k_{3F}} - k_{4F})w_{F2}\text{sat}(w_{F2}) - \text{sat}^2(w_{F2}) \\ &\quad + (k_{5F}a_F + k_{6F}c_F)w_{F1} + (k_{6F}b_F + k_{5F}c_F)w_{F2} + k_{2F}k_{6F}\text{sat}(w_{F2}) \\ &= -(1 - \theta_F)(c_3 w_{F1} - c_4 w_{F2})^2 + ((-\theta_F c_3^2 + 2c_3 c_4 \theta_F)w_{F1}^2 + (k_{5F}a_F + k_{6F}c_F)w_{F1}) \\ &\quad + ((-\theta_F c_4^2 + 2c_3 c_4 \theta_F)w_{F2}^2 + (k_{6F}b_F + k_{5F}c_F)w_{F2}) \\ &\quad - 2k_{2F}(\sqrt{k_{2F}k_{3F}} - k_{4F})w_{F2}\text{sat}(w_{F2}) - \text{sat}^2(w_{F2}) + k_{2F}k_{6F}\text{sat}(w_{F2}). \end{aligned}$$

Let's define

$$\bar{w}_{F1} := \frac{|k_{5F}a_F + k_{6F}c_F|}{\theta_F |c_3^2 - 2c_3 c_4|}, \quad \bar{w}_{F2} := \frac{|k_{6F}b_F + k_{5F}c_F|}{\theta_F |c_4^2 - 2c_3 c_4|}$$

if

$$|w_{F1}| \geq \bar{w}_{F1} \quad \text{and} \quad |w_{F2}| \geq \bar{w}_{F2}, \quad (4.30)$$

it follows that

$$\begin{aligned} \langle \nabla V_F(w_F), \dot{w}_F \rangle &\leq -(1 - \theta_F)(a w_{F1} - b w_{F2})^2 - 2k_{2F}(\sqrt{k_{2F}k_{3F}} - k_{4F})w_{F2}\text{sat}(w_{F2}) \\ &\quad - \text{sat}^2(w_{F2}) + k_{2F}k_{6F}\text{sat}(w_{F2}). \end{aligned}$$

Finally, if

$$-\text{sat}^2(\bar{w}_{F2}) + k_{2F}k_{6F}\text{sat}(\bar{w}_{F2}) \leq 0, \quad (4.31)$$

$$\langle \nabla V_F(w_F), \dot{w}_F \rangle \leq 0.$$

Since $\dot{V}_F \leq 0$, we can assert that the equilibrium point $[x_1^F \ 0]^T$ is input-to-state stable for the forced system (4.22) with the conditions (4.30) and (4.31).

We can observe here that if $\bar{w}_{F2} \geq u_F$ the system is not allowed to work in the linear zone. Therefore, we add the condition

$$\bar{w}_{F2} \leq u_F. \quad (4.32)$$

This implies that \bar{w}_{F2} belongs to the linear zone of the system and the constraint (4.31) can be written as

$$\bar{w}_{F2} \geq k_{2F} | k_{6F} |. \quad (4.33)$$

At this point, we can state that the equilibrium point $[x_1^F \ 0]^T$ is input-to-state stable for the forced system (4.22) with the conditions (4.30), (4.32) and 4.33).

Remark 4.7.5. *In order to better understand the conditions that guarantee the input-to-state stability of the forced system (4.22), we can observe that, according the evolution of the system and according the position of the reference frame in Figure 4.1, the robotic manipulator is interacting with the work environment when the position has a positive value. This implies that the first constraint in Equations (4.30), can be simplified in:*

$$w_{F1} \geq -\bar{w}_{F1}. \quad (4.34)$$

In particular, if the constraint in (4.34) is valid, we can state that (similarly to the case of the saturated PD controller), when the system is working in the saturation zone, the system is always input-to-state stable. However, when the system is working in the linear zone, the system can be unstable.

Let's analyse the system working in the linear zone, that is when $|k_c(1-k_f)x_1 + b_c(1-k_f)x_2 + k_f f_c^d| \leq u_F$. In particular, the system (4.22) becomes:

$$\begin{aligned} \dot{w}_{F1} &= -k_{1F}w_{F1} - k_{2F}w_{F2} + k_{5F} \\ \dot{w}_{F2} &= k_{3F}w_{F1} + (k_{4F} - \frac{1}{k_{2F}})w_{F2} + k_{6F} \end{aligned}$$

If we analyze the Lyapunov function $V_F(w_F)$ when the system is in the linear zone, we find out that the derivative of V_F along the trajectories of the system is always negative.

We can conclude that the equilibrium point $[x_1^F \ 0]^T$ is input-to-state stable for the system (4.22) if

$$w_{F1} \geq \bar{w}_{F1} \quad \text{and} \quad k_{2F} | k_{6F} | \leq \bar{w}_{F2} \leq u_F.$$

From this inequality, it follows that the parameter θ_F has to satisfy the constraint

$$\theta_F < \min\left\{1, \frac{|k_{6F}b_F + k_{5F}c_F|}{k_{2F} | k_{6F} | |c_4^2 - 2c_3c_4|}\right\}.$$

4.7.4 Design of the Hysteresis Thresholds

At this point we are able to design the thresholds γ_1 and γ_2 for the jump conditions of the hybrid controller \mathcal{H}_c . Also in the case of saturated controllers, the jumps can be explicitly expressed in terms of x_1 and x_2 by using the definition of the contact force in (4.4) which corresponds to a hyperplane in \mathbb{R}^2 .

With the same notation used in Section 4.5, the threshold γ_2 defines a line ℓ_{γ_2} in the right half plane of the x -coordinates that discriminates the switch from the position to the force controller, $f_c \geq \gamma_2$. In the same way, the threshold γ_1 defines a line ℓ_{γ_1} in the right half plane of the x -coordinates that discriminates the switch from the force to the position controller, $f_c \leq \gamma_1$.

As depicted in Figure 4.16, we design the threshold γ_2 so that the line ℓ_{γ_2} is bounded by two lines, $\underline{\ell}_{\gamma_2}$ and $\bar{\ell}_{\gamma_2}$, that respectively crosses the point $(x_1^d - \bar{w}_{P1}, 0)$ and $(x_1^P, 0)$. In this way, we assure that there is always a switch from the position to the force controller. Therefore, it follows that:

$$k_c(x_1^d - \bar{w}_{P1}) \leq \gamma_2 \leq k_c x_1^P$$

To guarantee that this design is achievable, it is necessary to satisfy the condition $\bar{w}_{F1} \leq x_1^d - \bar{w}_{P1} \leq x_1^P$, that can be also written as

$$\bar{w}_{F1} \leq \frac{k_p}{k_c} \bar{w}_{P1}.$$

Moreover, in this way, we define the maximum admitted velocity of impact x_2^* , which is given by the intersection of $\underline{\ell}_{\gamma_2}$ and the line $x_1 = \bar{w}_{F1}$

$$x_2^* = -\frac{k_c}{b_c}(x_1^d - \bar{w}_{P1} + \bar{w}_{F1}).$$

For the design of the threshold γ_1 , we will consider lines ℓ_{γ_1} that are parallel to the line ℓ_{γ_2} so that $\bar{\ell}_{\gamma_1}$ is the line that crosses the point $(\bar{w}_{F1}, -x_2^*)$ and $\underline{\ell}_{\gamma_1}$ is the line which crosses the origin. Therefore, it follows that:

$$0 \leq \gamma_1 \leq k_c(x_1^d - \bar{w}_{P1} + 2\bar{w}_{F1}). \quad (4.35)$$

Figure 4.16 shows the lines $\underline{\ell}_{\gamma_2}$, $\bar{\ell}_{\gamma_2}$, $\underline{\ell}_{\gamma_1}$, $\bar{\ell}_{\gamma_1}$, the maximum admitted velocity of impact x_2^* and the steady state of the position controller x_1^P .

By proceeding in the same way as in the non saturated control, we can define the general Lyapunov function $V : \mathbb{R}^2 \rightarrow \mathbb{R}_{\geq 0}$ for the hybrid closed loop system \mathcal{H}_{cl} that is given by

$$V(x) = (q - 1)V_P + qV_F.$$

Since for every initial condition $x^0 \in K$,

$$\begin{aligned} \langle \nabla V(x), F(x) \rangle &\leq 0 \quad \forall x \in C \\ V(G(x)) - V(x) &\leq 0 \quad \forall x \in D, \end{aligned}$$

the solutions (x, q) to \mathcal{H}_{cl} approach to the compact set $\{x_1^F\} \times \{1\}$. By LaSalle's invariance principle, the compact set $\{x_1^F\} \times \{1\}$ is locally asymptotically stable for the hybrid closed loop system \mathcal{H}_{cl} .

Remark 4.7.6. *In the case of the controllers with saturation, the set \tilde{K} of initial conditions is not compact and is enlarged thanks to the presence of the saturation. In Figure 4.17, the compact set \tilde{K} is depicted.*

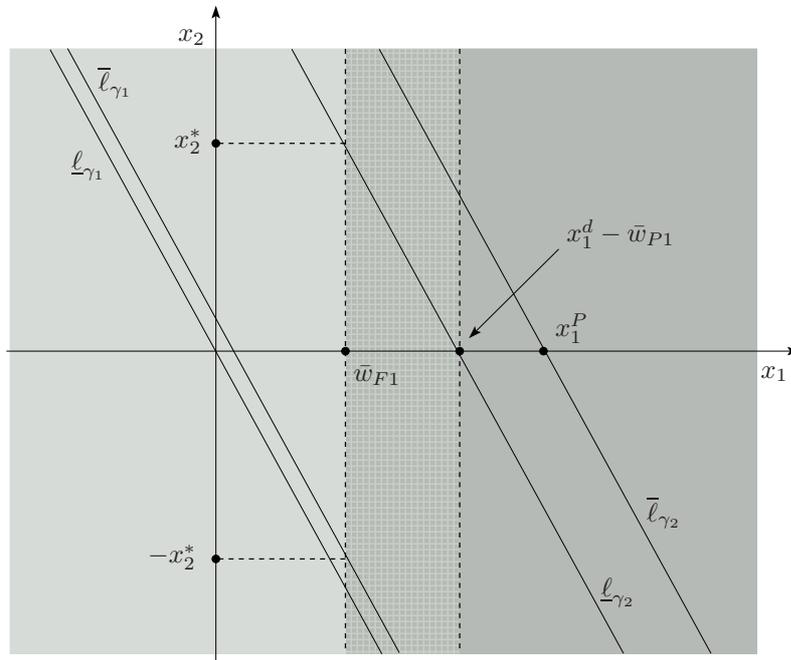


Figure 4.16: Scheme of the architecture with the saturated controller.

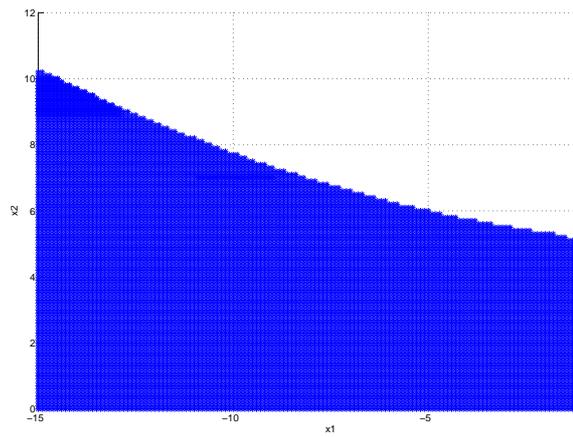


Figure 4.17: Non-compact set \tilde{K} for the system with $\tilde{\kappa}_P$ position controller.

4.7.5 Margin of Robustness

The margin of robustness to measurement noise obtained with this control strategy with saturation is defined as the minimum distance between ℓ_{γ_1} and $\bar{\ell}_{\gamma_1}$, and it is denoted with the parameter $\varepsilon > 0$.

If a desired margin of robustness ε is specified, one can design the line ℓ_{γ_1} so that it is at least a ε -apart from the $\bar{\ell}_{\gamma_1}$. It follows that, to accomplish the desired margin of robustness, instead of (4.35), γ_1 needs to satisfy

$$\gamma_1 \leq -\varepsilon\sqrt{k_c^2 + b_c^2} + k_c(x_1^d - \bar{w}_{P1} + 2\bar{w}_{F1}).$$

The maximum margin of robustness possible is given by:

$$\varepsilon_{max} = \frac{k_c(x_1^d - \bar{w}_{P1} + 2\bar{w}_{F1}) - \gamma_1}{\sqrt{k_c^2 + b_c^2}}$$

4.8 Simulations

In this section, we describe the simulations of the case in which the saturated controllers are taken into account.

The simulation results are shown in the following. In particular, in the interaction task, we suppose that, at the beginning of the task, the manipulator is in the free space at an initial position $x_1^0 = -8mm$ and initial velocity $x_2^0 = 6mm/s$ with respect to the point to be touched in the environment, that is considered at $x_1 = 0$. The desired force we want to apply is $f_c^d = 5N$. Moreover, we suppose to deal with a soft material characterized by stiffness parameter $k_c = 10N/mm$ and damping term $b_c = 0.3Ns/mm$. Figure 4.18 illustrate the trajectory of the system in the phase diagram.

4.9 Conclusions and Future Work

4.9.1 Conclusions

An innovative hybrid system approach for the control of robotic interaction has been proposed. The hybrid control involves a position and a force controller and the switching strategy is based only on force measurement information. The logic of switching is realized by introducing hysteresis in the control loop; this guarantees a margin of robustness with respect to measurement noise. The proposed architecture requires a contact model for the environment and a bounded impact velocity. The controller guarantees that the interaction forces are bounded and that no bounces are present when the manipulator gets in contact with the work environment.

4.9.2 Future Work

Future work will be aimed to the generalization of the described strategy to a full model of interaction in six degree of freedom in order to extend this kind of controller to application of complex robotics, such as locomotion and dexterous manipulation. Moreover, since the presented hybrid control system requires bounded impact velocity, we will analyze how to design the position controller by using techniques in [73], in order to introduce a saturation function in the control of the impact velocity.

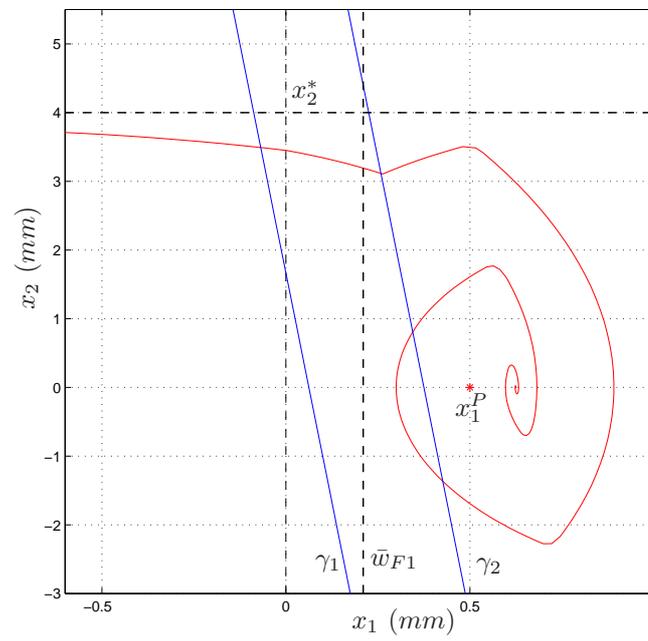


Figure 4.18: Phase diagram of the switching strategy with the saturated controllers. The plot depicts the l_{γ_1} and l_{γ_2} lines, and the trajectory of the system. The controller is able to avoid the bouncing off of the robotic system.

Conclusion

In this thesis, both planning and control of robotic manipulation tasks have been addressed. In particular, in Chapter 1, a new approach to planar objects manipulation by dexterous regrasping of three-fingered hands has been presented. In the regrasping task, it is required to plan sequences of force-closure grasps in order to achieve a desired stable configuration from an initial one. The discretized boundary of an irregular object is analyzed so that all the regions that ensures force-closure grasps are established: the connectivity between these regions provides the computation of the *regrasp map*. The regrasp sequence is obtained either with slides or with jumps of the fingertips on the object boundary and is realized with the solution of a shortest path problem.

In Chapter 2 we have addressed the problem of the determination of a suitable set of grasping forces that a mechanical hand has to apply in order to balance the external forces and torques applied on a generic object and to keep it in equilibrium. In this chapter we present a new mathematical approach to efficiently obtain the optimal solution of this problem by means of the dual theorem of non-linear optimization programming. In particular, the dual theorem can be applied to the force distribution problem only if it is modeled such as the basic convexity property is satisfied.

Chapter 3 we have discussed a practical method for the tracking of grasp points in image space that is based on transferring previously computed grasp points from an initial image to subsequent ones and on the analysis of the new grasp configuration. In order to obtain an efficient visually guided grasping, three basic techniques are used together. In particular, two of them are based on a grasp description that is invariant with respect to the relative movement between two object views, one of these strategies being used jointly with an object tracking method. The third proposed alternative is based on a homography computed between two object views.

Finally, in Chapter 4 we have presented an innovative hybrid system approach for the control of robotic interaction. This hybrid architecture can control a manipulator during the stages of the interaction: the free motion, the transition phase and the constrained motion. The switching logic is based on a contact detection which is robust with respect to measurement noise so to guarantee that, during the robotic task, no bounces are present. The design of the controller is based on a Lyapunov analysis and depends on the viscoelastic parameters of the environment. The hybrid control can guarantee good performance in both cases of stiff and compliant contact model with a constraint on the velocity of impact.

Bibliography

- [1] P.K. Allen, A. Timcenko, B.H. Yoshimi, and P. Michelman. Automatic tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. on Robotics and Automation*, 9(2):152–165, April 1993.
- [2] N. Smaby A.M. Okamura and M.R. Cutkosky. An overview of dexterous manipulation. In *IEEE International Conference on Robotics and Automation*, pages 255–262, 2000.
- [3] J.T. Schwartz B. Mishra and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica, Special Issue: Robotics*, 2(4):541–558, 1987.
- [4] F. Berry, P. Martinet, and J. Gallice. Real time visual servoing around a complex object. *IEICE Trans. on Information and Systems*, E83-D(7):1358–1368, July 2000. Special Issue on Machine Vision Applications.
- [5] A. Bicchi. On the closure properties of robotics grasping. *International Journal of Robotics Research*, 14(4):319–334, 1995.
- [6] A. Bicchi and V. Kumar. Robotic grasping and contact: A review. In *IEEE International Conference on Robotics and Automation*, pages 348–353, 2000.
- [7] A. Blake. A symmetry theory of planar grasp. *The Intl. Journal of Robotics Research (Special Issue on Integration Among Planning, Sensing, and Control)*, 14(5):425–444, October 1995.
- [8] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998.
- [9] M. Buss, H. Hashimoto, and J. B. Moore. Dexterous hand grasping force optimization. *IEEE Transactions on Robotics and Automation*, 12(3):406–418, 1996.
- [10] R. Carloni, G. Recatalá, C. Melchiorri, P.J. Sanz, and E. Cervera. Homography-based grasp tracking for planar objects. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 795–800, New Orleans (Louisiana), USA, April 2004.
- [11] M. C. Çavuşoğlu, J. Yan, and S. S. Sastry. A hybrid system approach to contact stability and force control in robotic manipulators. In *In Proceedings of the 12th IEEE International Symposium on Intelligent Control*, pages 143–148, 1997.
- [12] I.M. Chen and J.W. Burdick. Finding antipodal point grasps on irregularly shaped objects. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 2278–2283, May 1992.

- [13] F. Cheng and D. E. Orin. Efficient algorithm for optimal force distribution - the compact-dual lp method. *IEEE Transactions on Robotics and Automation*, 6(2):178–187, 1990.
- [14] G. Chesi, E. Malis, and R. Cipolla. Collineation estimation from two unmatched views of an unknown planar contour for visual servoing. In *Proc. British Machine Vision Conference*, pages 224–233, Nottingham, UK, September 1999.
- [15] G. Chesi, E. Malis, and R. Cipolla. Automatic segmentation and matching of planar contours for visual servoing. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, volume 3, pages 2753–2758, San Francisco (California), USA, April 2000.
- [16] J. Cornellà. *Geometrical Approaches for Optimal grasp synthesis*. PhD Thesis, Technological University of Catalonia, 2006.
- [17] J. Cornellà and R. Suárez. Determining independent grasp regions on 2d discrete objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2936–2941, 2005.
- [18] J. Cornellà and R. Suárez. Fast and flexible determination of force-closure independent regions to grasp polygonal objects. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 778–783, Barcelona, Spain, April 2005.
- [19] J. Cornellà and R. Suárez. A new framework for planning three-finger grasps of 2d irregular objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5688–5694, 2006.
- [20] J. Cornellà, R. Suárez, R. Carloni, and C. Melchiorri. Grasping force optimization using dual methods. In *IFAC Symposium on Robot Control*, 2006.
- [21] C. Davidson and A. Blake. Error-tolerant visual planning of planar grasp. In *Proc. 6th Intl. Conf. on Computer Vision*, pages 911–916, 1998.
- [22] B. Faverjon and J. Ponce. On computing two-finger force-closure grasps of curved objects. In *IEEE International Conference on Robotics and Automation*, pages 424–429, 1991.
- [23] H. Fujimoto, L.-C. Zhu, and K. Abdel-Malek. Image-based visual servoing for grasping unknown objects. In *Proc. IECON 2000*, pages 876–881, 2000.
- [24] G. Gilardi and I. Sharf. Literature survey of contact dynamics modelling. *Pergamon, Mechanism and Machine Theory*, 18:1213–1239, 2002.
- [25] R.C. González and P. Wintz. *Digital Image Processing*. Addison-Wesley, Reading (Massachusetts), USA, 2nd edition, 1987.
- [26] L. Han, J. C. Trinkle, and Z. X. Li. Grasp analysis as linear matrix inequality problems. *IEEE Transactions on Robotics and Automation*, 16(6):663–674, 2000.
- [27] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, second edition, 2004.
- [28] A. Hauck, J. Rüttinger, M. Sorg, and G. Färber. Visual determination of 3D grasping points on unknown objects with a binocular camera system. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 272–278, Kyongju, Korea, 1999.

- [29] U. Helmke, K. Huper, and J. B. Moore. Quadratically convergent algorithms for optimal dextrous hand grasping. *IEEE Transactions on Robotics and Automation*, 18(2):138–146, 2002.
- [30] N. Hogan. Impedance control: An approach to manipulation: Part i theory. *ASME Journal of Dynamic Systems, Measurement and Control*, 107:335–345, 1985.
- [31] R. Horaud, F. Dornaika, and B. Espiau. Visually guided object grasping. *IEEE Trans. on Robotics and Automation*, 14(4):525–532, August 1998.
- [32] S. Hutchinson, G.D. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.
- [33] B. Mishra J. Hong, G. Lafferriere and X. Tan. Fine manipulation with multifinger hands. In *IEEE International Conference on Robotics and Automation*, pages 1568–1573, 1990.
- [34] H. Liu J.W. Li and H.G. Cai. On computing three-finger force-closure grasps of 2-d and 3-d objects. *IEEE Transaction on Robotics and Automation*, 19(1):155–161, 2003.
- [35] I. Kamon, T. Flash, and S. Edelman. Learning to grasp using visual information. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 2470–2476, Minneapolis (Minnesota), USA, April 1996.
- [36] J. Kerr and B. Roth. Analysis of multifingered hands. *International Journal of Robotics Research*, 4(4):3–17, 1986.
- [37] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Transactions on Robotics and Automation*, 3(1):43–53, 1987.
- [38] W. Kwon and B.H. Lee. Optimal force distribution of multiple cooperating robots using nonlinear programming dual method. In *IEEE International Conference on Robotics and Automation*, pages 2408–2413, 1996.
- [39] P. Li, F. Chaumette, and O. Tahri. A shape tracking algorithm for visual tracking. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, Barcelona, Spain, 2005.
- [40] G. Liu and Z. Li. Real-time grasping-force optimization for multifingered manipulation: Theory and experiments. *IEEE/ASME Transactions on Mechatronics*, 9(1):65–77, 2004.
- [41] G. Liu, J. Xu, and Z. Li. On geometric algorithms for real-time grasping force optimization. *IEEE Transactions on Control Systems Technology*, 12(6):843–859, 2004.
- [42] Y.-H. Liu. Qualitative test and force optimization of 3d frictional form-closure grasps using linear programming. *IEEE Transactions on Robotics and Automation*, 15(1):163–173, 1999.
- [43] D.G. Luenberger. *Introduction to Linear and Nonlinear Programming*. Addison-Wesley Publishing Company, 1973.
- [44] Q.-T. Luong and O.D. Faugeras. The fundamental matrix: Theory, algorithms and stability analysis. *International Journal of Computer Vision*, 17:43–75, 1996.

- [45] A. Macchelli, C. Melchiorri, R. Carloni, and M. Guidetti. Space robotics: an experimental set-up based on RTAI-linux. In *4th Real-Time Linux Workshop*, Boston, USA, December 2002.
- [46] E. Malis, F. Chaumette, and S. Boudet. Multi-cameras visual servoing. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, volume 4, pages 3183–3188, San Francisco (California), USA, April 2000.
- [47] P. Martinet and E. Cervera. Stacking jacobians properly in stereo visual servoing. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 717–722, Seoul, Korea, May 2001.
- [48] P.R.S. Mendonça, K.-Y.K. Wong, and R. Cipolla. Epipolar geometry from profiles under circular motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):604–616, June 2001.
- [49] J. K. Mills and D. M. Lokhorst. Stability and control of robotic manipulators during contact/noncontact task transition. *IEEE Transactions on Robotics and Automation*, 9(3):335–345, 1993.
- [50] A. Morales, G. Recatalá, P.J. Sanz, and A.P. del Pobil. Heuristic vision-based computation of planar antipodal grasps on unknown objects. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 583–588, Seoul, Korea, May 2001.
- [51] J.L. Mundy and A. Zisserman, editors. *Geometric Invariance in Computer Vision*. The MIT Press, Cambridge (Massachusetts), USA, 1992.
- [52] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [53] Y. Nakamura, K. Nagai, and T. Yoshikawa. Dynamics and stability in coordination of multiple robotic mechanisms. *International Journal of Robotics Research*, 8(2):44–61, 1989.
- [54] T. Omata and K. Nagata. Planning reorientation of an object with multifingered hand. In *IEEE International Conference on Robotics and Automation*, pages 3104–3110, 1994.
- [55] M. Tomizuka P. R. Pagilla. Contact transition control of nonlinear mechanical systems subject to a unilateral constraint. *ASME Journal of Dynamic Systems, Measurement and Control*, 119:749–759, 1997.
- [56] N. Papanikolopoulos and C. Smith. Computer vision issues during eye-in-hand robotic tasks. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, volume 3, pages 2989–2994, Nagoya, Japan, 1995.
- [57] D. Perrin, C.E. Smith, O. Masoud, and N.P. Papanikolopoulos. Unknown object grasping using statistical pressure models. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 1054–1059, San Francisco (California), USA, April 2000.
- [58] J. Ponce, D. Stam, and B. Faverjon. On computing force-closure grasps of curved two dimensional objects. *The Intl. J. of Robotics Research*, 12(3):263–273, June 1993.

- [59] A. R. Teel R. Carloni, R. G. Sanfelice and C. Melchiorri. A hybrid control strategy for robust contact detection and force regulation. In *In Proceedings of the 26th IEEE American Control Conference*, 2007.
- [60] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, 103:126–133, 1981.
- [61] G. Recatalá. *Visual determination, tracking and execution of 2D grasps using a behavior-inspired approach*. PhD thesis, “Jaume I” University, Castellón, Spain, November 2003.
- [62] G. Recatalá, P.J. Sanz, E. Cervera, and A.P. del Pobil. Grasp-based visual servoing for gripper-to-object positioning. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 118–123, Sendai, Japan, September 2004.
- [63] G. Recatalá, P.J. Sanz, E. Cervera, and A.P. del Pobil. Grasp-based visual servoing for the manipulation of planar objects. In *Proc. Intl. Conf. on Intelligent Manipulation and Grasping*, pages 549–554, Genoa, Italy, July 2004.
- [64] G. Recatalá, M. Sorg, J. Leupold, P.J. Sanz, and A.P. del Pobil. Visual grasp determination and tracking in 2D dynamic scenarios. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 25–30, Lausanne, Switzerland, October 2002.
- [65] P.J. Sanz, A.P. del Pobil, J.M. Iñesta, and G. Recatalá. Vision-guided grasping of unknown objects for service robots. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 3018–3025, Leuven, Belgium, May 1998.
- [66] P.R. Sinha and J.M. Abel. A contact stress model for multifingered grasps of rough objects. *IEEE Transactions on Robotics and Automation*, 8(1):7–22, 1992.
- [67] M. Spong. Swing up control of the acrobot. In *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 2356–2361, 1994.
- [68] M. Spong. The swing up control problem for the acrobot. *IEEE Control Systems Magazine*, 15(2):49–55, 1995.
- [69] A. Sudsang and T. Phoka. Regrasp planning for a 4-fingered hand manipulating a polygon. In *IEEE International Conference on Robotics and Automation*, pages 2671–2676, 2003.
- [70] N. Niparnan T. Phoka, P. Pipattanasomporn and A. Sudsang. Regrasp planning of four-fingered hand for parallel grasp of a polygonal object. In *IEEE International Conference on Robotics and Automation*, pages 7961–2676, 2005.
- [71] M. Buss T. Schlegl and G. Schmidt. A hybrid systems approach toward modeling and dynamical simulation of dextrous manipulation. *IEEE/ASME Transaction on Mechatronics*, 8(3):352–361, 2003.
- [72] N. Xi T. Tarn, Y. Wu and A. Isidori. Force regulation and contact transition control. *IEEE Control Systems Magazine*, 16(1):32–40, 1996.
- [73] A. R. Teel. A nonlinear small gain theorem for the analysis of control systems with saturation. *IEEE Transactions on Automatic Control*, 41(9):1256–1270, 1996.

- [74] Y. Xia, J. Wang, and L. Fok. Grasping-force optimization for multifingered robotic hands using a recurrent neural network. *IEEE Transactions on Robotics and Automation*, 20(3):549–554, 2004.
- [75] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Technical Report 2273, INRIA, Sophia-Antipolis, France, May 1994.