HYBRID KINEMATIC CONTROL OF DUAL-ARM COOPERATIVE ROBOTS FOR OBJECT MANIPULATION

Rafael de Oliveira Faria

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadores: Fernando Cesar Lizarralde
Antonio Candea Leite

Rio de Janeiro
Setembro de 2016

HYBRID KINEMATIC CONTROL OF DUAL-ARM COOPERATIVE ROBOTS FOR OBJECT MANIPULATION

Rafael de Oliveira Faria

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Fernando Cesar Lizarralde, D.Sc.

Prof. Antonio Candea Leite, D.Sc.

Prof. Gustavo Medeiros Freitas, D.Sc.

Prof. Marco Antonio Meggiolaro, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2016

Faria, Rafael de Oliveira

Hybrid Kinematic Control of Dual-arm Cooperative Robots for Object Manipulation/Rafael de Oliveira Faria. – Rio de Janeiro: UFRJ/COPPE, 2016.

XIII, 112 p.: il.; 29, 7cm.

Orientadores: Fernando Cesar Lizarralde

Antonio Candea Leite

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2016.

Bibliografia: p. 97 – 107.

1. Sistemas robóticos cooperativos. 2. Controle híbrido cinemático de posição-força. 3. Estimação visual da pose de objetos. 4. Estimação de força. I. Lizarralde, Fernando Cesar *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Agradecimentos

Agradeço primeiramente a minha família, pelo imenso apoio e inspiração para prosseguir na vida acadêmica.

A Mauricio Galassi, que sempre demonstrou prontidão e disponibilidade, facilitando muito os trabalhos experimentais com o robô no CENPES, agradeço pela confiança e pelos conselhos.

Aos membros do LABCON e LEAD, Guilherme Carvalho, João Monteiro e Alex Fernandes, pelo auxílio e aprendizado com as conversas.

Aos amigos de trabalho Eneida, Viviane, Victor, Claudir, Ciro, Eustério, Patricia, Gustavo e Jaqueline pela força que transmitiram e aos chefes José Antonio, Pedro Saldanha e João Márcio que sempre apoiaram a continuidade e o desenvolvimento deste trabalho.

Aos grandes amigos que tive a honra de conhecer durante o mestrado, em especial: Paulo Yamasaki, por todas as conversas, desabafos, discussões e reflexões sobre praticamente tudo na vida; e Florentin, pela oportunidade de realizar um excelente trabalho em conjunto e por sempre me incentivar.

Aos membros da banca, Gustavo Freitas, que participou do início deste trabalho e contribuiu com críticas, ideias e sugestões sempre importantes para melhorar a qualidade do trabalho e Marco Antonio Meggiolaro, pelos elogios e críticas muito construtivas.

Agradeço muito aos meus orientadores. A Fernando Lizarralde, por me aceitar como aluno orientado, pela oportunidade que ofereceu de trabalhar com um robô real, pela confiança e por todos os ensinamentos ao longo destes anos de graduação e mestrado. E a Antonio Leite, por investir tempo e acreditar neste trabalho, pelos incentivos, pelas ideias inspiradoras e pela paciência infinita sempre.

À Emília Alves, agradeço pela paciência e, principalmente, por todo o amor e companheirismo que são essenciais na minha vida e foram muito importantes para vencer cada desafio durante este trabalho.

Finalmente, agradeço a todos que estiveram torcendo e desejando o sucesso deste trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

CONTROLE HÍBRIDO CINEMÁTICO DE ROBÔS COOPERATIVOS TIPO *DUAL-ARM* PARA MANIPULAÇÃO DE OBJETOS

Rafael de Oliveira Faria

Setembro/2016

Orientadores: Fernando Cesar Lizarralde
Antonio Candea Leite

Programa: Engenharia Elétrica

Nos últimos anos, o problema da manipulação robótica cooperativa para transporte de carga vem sendo estudado sob o ponto de vista da modelagem e controle. Tais estudos exigem uma metodologia que considere não somente o controle do sistema robótico, como também o problema da pegada do objeto. Visando preencher esta lacuna, o objetivo deste trabalho é propor um método de manipulação robótica cooperativa para identificar visualmente a pose (posição e orientação) de um objeto e manipulá-lo, de maneira segura e eficiente, utilizando um robô do tipo *dual-arm*.

O sistema robótico é modelado utilizando o conceito de variáveis absolutas e relativas, que é caracterizado por sua simplicidade de formulação. A estratégia de controle baseia-se em um esquema de controle híbrido cinemático cooperativo, que combina os objetivos de rastreamento de posição e regulação de força simultaneamente, sem o uso de sensores de força acoplados aos efetuadores. Nesse contexto, a força exercida pelo manipulador sobre o objeto é estimada indiretamente, por meio de medidas provenientes dos sensores de torque nas juntas do robô.

A estimação da pose do objeto a ser manipulado é obtida por meio da técnica de Homografia, utilizando marcadores fiduciais e uma câmera monocular acoplada ao efetuador de cada manipulador. Após a estimação da pose, o sistema robótico posiciona seus manipuladores a fim de realizar a pegada inicial de forma autônoma e, posteriormente, o controle híbrido cooperativo é acionado para que o objeto siga uma trajetória de referência desejada.

Simulações numéricas e ensaios experimentais, realizados com um robô Baxter®, ilustram o desempenho e a eficácia da metodologia proposta.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

HYBRID KINEMATIC CONTROL OF DUAL-ARM COOPERATIVE ROBOTS FOR OBJECT MANIPULATION

Rafael de Oliveira Faria

September/2016

Advisors: Fernando Cesar Lizarralde
Antonio Candea Leite

Department: Electrical Engineering

In the last years, the problem of cooperative robotic manipulation for load transportation has been studied from the modeling and control point of view. Such studies require a methodology which considers not only the control design but also the object grasping problem. In order to fill this gap, the objective of this work is to propose a cooperative manipulation method that visually estimates the pose (position and orientation) of the object and then manipulates it in a safe and efficient manner, using a dual-arm robot.

The robotic system is modeled using the concept of absolute and relative variables, which is characterized by its simplicity. The control strategy is based on a cooperative hybrid kinematic control scheme, which combines the objectives of position tracking and force regulation, simultaneously, without using force sensors attached to the end effectors. In this context, the contact force applied by the manipulator on the object is indirectly estimated using joint torque sensors measurements.

The estimation of the initial pose of the object to be manipulated is obtained by the Homography technique using fiducial markers and a monocular camera mounted to each end effector. After estimating the pose, the robotic system aligns its manipulators in order to perform the object grasping autonomously and, thereafter, the hybrid controller is activated ensuring that the object successfully follows a user-defined reference trajectory.

Numerical simulations and experimental tests, performed with a Baxter® robot, illustrate the performance and the effectiveness of the proposed methodology.

Contents

List of Figures **ix**

List of Tables **xiii**

1 Introduction **1**

1.1 Motivation . . . 2
1.2 Review of the State-of-the-Art . . . 3
1.3 Objectives . . . 7
1.4 Methodology . . . 7
1.5 Contributions . . . 9
1.6 Organization . . . 10

2 Kinematic Modeling of a Robotic System **11**

2.1 Forward Kinematics . . . 12
2.1.1 Joint Space and Operational Space . . . 14
2.1.2 Absolute and Relative Configuration . . . 16
2.2 Differential Kinematics and Statics . . . 18
2.2.1 Geometric Jacobian . . . 18
2.2.2 Analytical Jacobian . . . 19
2.2.3 Representation Jacobian . . . 19
2.3 Statics . . . 20
2.3.1 Force Estimation based on Joint Torques . . . 21
2.4 Cooperative Manipulators . . . 22
2.4.1 Symmetric Formulation . . . 24
2.4.2 Task-Space Formulation . . . 26

3 Cooperative Control Strategies **28**

3.1 Single-arm Kinematic Control . . . 29
3.1.1 Position Control . . . 31
3.1.2 Force Control . . . 31
3.1.3 Orientation Control . . . 32
3.2 Cooperative Control . . . 33

3.2.1 Differential Kinematics for Coordinated Motion Tasks 34
3.2.2 Numerical simulation 37
3.3 Cooperative Control using Force Feedback 43
3.3.1 Object Modeling 43
3.3.2 Contact Modeling 44
3.3.3 Natural and Artificial Constraints 46
3.3.4 Force Control and Interaction Methods 47
3.3.5 Dual-arm Hybrid Position-Force Kinematic Control 54
3.3.6 Numerical simulations 59
3.4 Conclusion 65

4 Experimental Setup 66
4.1 Problem Formulation 67
4.1.1 Baxter Robot Description - Hardware and Software 67
4.1.2 Object description 69
4.1.3 Validation of the force estimation 69
4.1.4 Sequence of operation 71
4.2 Methodology for dual-arm grasp of objects 72
4.2.1 Object identification using key features 73
4.2.2 Image Processing 75
4.2.3 Pose Estimation Algorithms 77
4.2.4 Experiments for Validation 79
4.2.5 Alignment of the Manipulators 80
4.3 Experiments and Results 82
4.4 Conclusion 94

5 Concluding Remarks and Perspectives 95
5.1 Conclusions 95
5.2 Future Work 96

A Proof of Theorem 1:
Dual-arm Hybrid Relative Position-Force Control 108

B Architecture for Teleoperation using Natural User Interface 111
B.1 Human Machine Interface 112

List of Figures

1.1 Robotics and some examples of its applications. 2
1.2 Robot manipulating a valve, one of the tasks on DARPA challenge. . 3
1.3 ATLAS robot, from Boston Dynamics, picking up a box. 4
1.4 Examples of robots with flexible joints, developed for human collaboration. 5
1.5 Experimental setup – Baxter robot and object with visual markers. . 9
1.6 The result obtained from the application of the methodology described in (Faria et al., 2015). 10

2.1 Joints and links description of a robotic manipulator. Extracted from (Leite, 2005). 13
2.2 Examples of kinematic chains. Extracted from (Siciliano et al., 2009). 13
2.3 The left $\mathcal{F}_1$ and right $\mathcal{F}_2$ end-effectors frames and the robot base frame $\mathcal{F}_b$, which is the common frame for the cooperative system. . . 16
2.4 Model of multiple arm system manipulating a load. 23
2.5 Model of a two-arm robotic system manipulating a common object. Model extracted from (Siciliano and Khatib, 2008). 24

3.1 Kinematic control diagram. 29
3.3 Coordinated motion: (a) absolute position and reference (b) absolute position error. 39
3.4 Coordinated motion: (a) relative position and reference (b) relative position error. 39
3.5 Coordinated motion: (a) absolute orientation error (b) relative orientation error. 40
3.6 Coordinated motion: (a) absolute position control (b) relative orientation control (c) relative position control (d) relative orientation control. 40
3.7 Coordinated motion: left arm joint control signals. 41
3.8 Coordinated motion: right arm joint control signals. 41

3.9 Coordinated motion: trajectory performed by manipulators (with reference). . . . 42
3.10 Simplified object model. . . . 44
3.11 Representation of an end effector in contact with a box and the corresponding friction cone. . . . 46
3.12 Task of manipulating an object in contact with a surface. Extracted from (Siciliano et al., 2009). . . . 47
3.13 Organization of force control methods. . . . 49
3.14 Representation of the interaction task considering an elastic surface. . 49
3.15 Stiffness Control. . . . 49
3.16 Impedance Control. . . . 50
3.17 Parallel Control. . . . 51
3.18 The original hybrid position-force control for a single manipulator. . . 52
3.19 Hybrid kinematic position-force control scheme for a single manipulator. 53
3.20 The original hybrid position-force scheme for a two arm system, considering absolute and relative variables. . . . 55
3.21 Constraint frame $\mathcal{F}_s$ represented at an example object. . . . 55
3.22 Position and force controlled variables, according to the directions. . . 56
3.23 Dual-arm Hybrid position-force scheme. . . . 58
3.24 Hybrid control: (a) absolute position and reference (b) absolute position error. . . . 60
3.25 Hybrid control: (a) relative position and reference (b) relative position error. . . . 61
3.26 Hybrid control: (a) absolute orientation error (b) relative orientation error. . . . 61
3.27 Hybrid control: (a) force applied to object (b) force error. . . . 62
3.28 Hybrid control: (a) absolute position control (b) relative orientation control (c) relative position control (d) relative orientation control. . . 62
3.29 Hybrid control: left arm joint control signals. . . . 63
3.30 Hybrid control: right arm joint control signals. . . . 63
3.31 Hybrid control: trajectory performed by manipulators (with reference). 64

4.1 The manipulation task proposed has two main steps. . . . 66
4.2 Experimental setup: robot, cameras and object. . . . 67
4.3 Links and joints representation for the Baxter arm. . . . 68
4.4 Experiment for force estimation (a) using data obtained from the torque sensors and (b) after a low-pass filter. . . . 70
4.5 Phases of the proposed manipulation task. . . . 71
4.6 Tag composed of four circular markers. . . . 73

4.7 Perspective problem using four feature points. Extracted from (Petersen, 2008*a*) . . . 74
4.8 Original image of an object with markers, oriented in a 45° angle. . . 75
4.9 Result of bitwise operation with HSV mask, applied to the original image. . . . 76
4.10 Smoother image after the Gaussian filter. . . . 76
4.11 Contours detection. . . . 77
4.12 Markers coordinates are successfully obtained. . . . 78
4.13 Input/Output model for the P4P problem. . . . 78
4.14 Output image and TAG frame. . . . 79
4.15 Vision Experiment: object different configurations. . . . 80
4.16 Results for the vision experiment. . . . 81
4.17 Alignment problem. . . . 82
4.18 Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: (a) absolute position and reference (b) absolute position error. . . . 84
4.19 Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: (a) relative position and reference (b) relative position error. . . . 85
4.20 Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: (a) absolute orientation error (b) relative orientation error. . . . 85
4.21 Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: (a) force applied to object (b) force error. . . . 86
4.22 Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: (a) absolute position control (b) relative orientation control (c) relative position control (d) relative orientation control. . . 86
4.23 Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: left arm joint control signals. . . . 87
4.24 Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: right arm joint control signals. . . . 87
4.25 Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: trajectory performed by manipulators (with reference). 88
4.26 Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: (a) absolute position and reference (b) absolute position error. . . . 89
4.27 Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: (a) relative position and reference (b) relative position error. . . . 90

4.28 Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: (a) absolute orientation error (b) relative orientation error. 90
4.29 Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: (a) force applied to object (b) force error. 91
4.30 Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: (a) absolute position control (b) relative orientation control (c) relative position control (d) relative orientation control. . . 91
4.31 Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: left arm joint control signals. 92
4.32 Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: right arm joint control signals. 92
4.33 Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: trajectory performed by manipulators (with reference). 93

B.1 System architecture 111
B.2 Kinect skeleton model. Extracted from Microsoft website. 112

List of Tables

3.1 Simulation parameters for cooperative coordination control. 38
3.2 Static friction coefficient for some materials. 45
3.3 Natural and artificial constraints for the example task. 48
3.4 Simulation parameters for cooperative hybrid position-force control. . 59
3.5 Simulation parameters for a flexible object. 59

4.1 Camera parameters 69
4.2 Statistics of visual identification experiments. 80
4.3 Experimental control parameters for cooperative hybrid position-force control. 83

Chapter 1

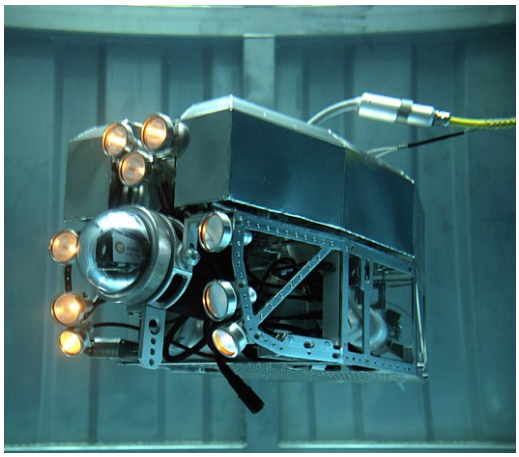
Introduction

Robotics has been essential to meet the growing need for industry to perform repetitive tasks, which require great precision, as well as activities in inhospitable environments such as deep sea or space. Furthermore, the use of robots allows to replace humans in hazardous or detrimental to health environments. The study of robotic systems is still in progress and it has great potential for mankind evolution, since these systems can be applied and generate benefits in several fields of science.

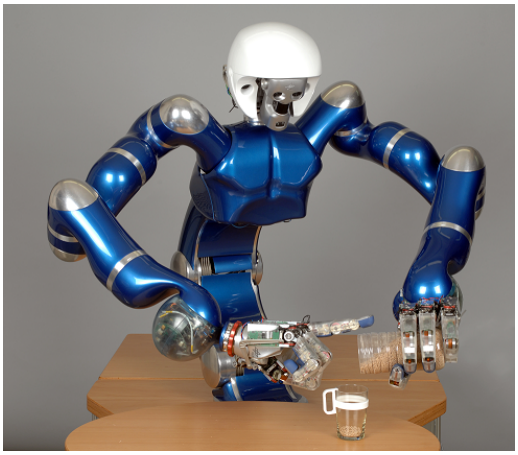
In this context, some applications can be pointed out among the main areas of research in robotics. For example, robots can be used for remote inspection and preventive maintenance of nuclear power plants, avoiding human exposure to radiation, such as the SUSI robot (Figure 1.1a) used for inspection inside nuclear pressure vessels; transportation of waste from nuclear generation to treatment and storage (Iborra et al., 2003). They can also be employed for rescuing of people, monitoring and inspection of environments after accidents or disasters, such as the robot *iRobot 710 Warrior* used in the Fukushima nuclear accident (Qian et al., 2012). Remarkably, projects as the DARPA challenge conduct research on the use of humanoid robots in unstructured environment for rescue and manipulation tasks (Figure 1.2) (Pratt and Manzo, 2013).

The use of multiple manipulators is essential in the execution of diverse tasks. Even activities which can be performed with a single manipulator, such as wreckage remove, can be improved if multiple arms are available. It also extends the range of robotic applications, allowing to perform tasks which require coordination and collaborative use of manipulators. For instance, in Medicine, robots with multiple arms are used for minimally invasive surgery, improving the accuracy of surgery and reducing recovery time for the patient, such as the robot *DaVinci* and the robot *DLR MiroSurge* (Hattori et al., 2002; Taylor and Stoianovici, 2003; Hagn et al., 2010).

Lately, new dual-arm robots have been conceived for human interaction and for cooperation to perform tasks, such as the robots *Justin* (Figure 1.1b.), which



(a) SUSI, from AREVA



(b) Justin, from DLR

(c) YUMI, from ABB

Figure 1.1: Robotics and some examples of its applications.

was developed for performing repair of satellites; *Yumi* (Figure 1.1c) and *Baxter* (Figure 1.4b.), which are safe and collaborative robots meant to be used in industry working alongside with humans or in research labs supporting the development of new strategies (De Santis et al., 2008).

1.1 Motivation

Several applications in industrial automation require a robotic manipulator to interact with the environment or to perform operations on a surface, such as polishing and drilling. However, some applications as load transport are limited by weight and size of the object, in case only one manipulator is used (see Figure 1.3). Assembly tasks may require the use of multiple robots working together, cooperatively (Sujan and Meggiolaro, 2004).

Besides, the use of human-like skills in industry, such as dual-arm object ma-

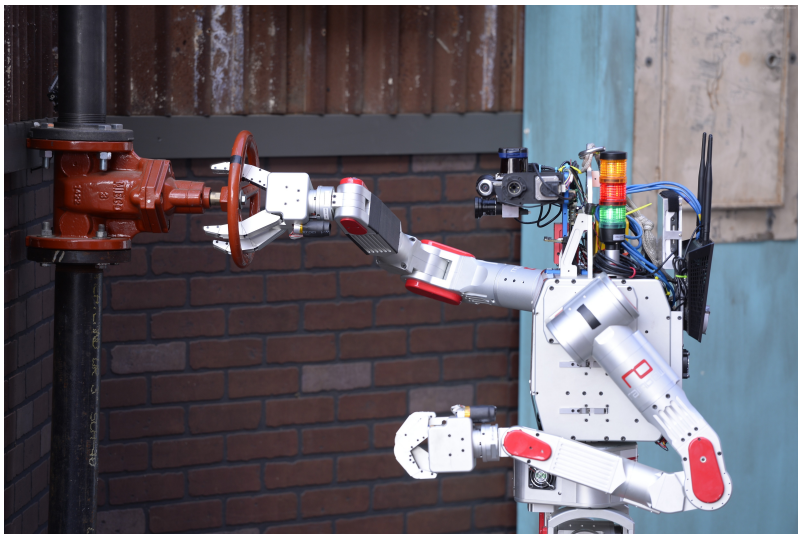


Figure 1.2: Robot manipulating a valve, one of the tasks on DARPA challenge.

nipulation associated with remote operation, is particularly important to replace humans in unhealthy and hazardous environments. For example, nuclear power plants, which contain areas with high radioactivity, and oil platforms, which may present risk of explosion, toxic gases and high temperatures.

Furthermore, dual-arm manipulation is also interesting for domestic applications, such as transfer a patient from bed to wheelchair (Mukai et al., 2010). It can also help people to transport heavy objects (Suda et al., 2003) or automate routine transport tasks.

Therefore, control methods must allow the robotic system to receive desired commands and, at the same time, to safely handle the manipulated object. The improvement of these methods and their application in dual-arm cooperative manipulation is the main motivation of this work.

1.2 Review of the State-of-the-Art

The interest in robotic systems composed by multiple manipulators emerged since the early days of robotics. Single arm systems have limited capacity to perform certain tasks. The use of more than one manipulator cooperatively makes it possible, for example, to handle large or heavy loads. Hence, the research of this type of system, starting in the early 70s, extends to the present days due to the problem complexity and the interest in new applications, for example in unstructured and/or harsh environments.

The cooperative task of handling a load using a teleoperated robotic system requires synchronism of the end effectors and prompt response to the operator commands. In this context, not only movement of the end effector and the contact

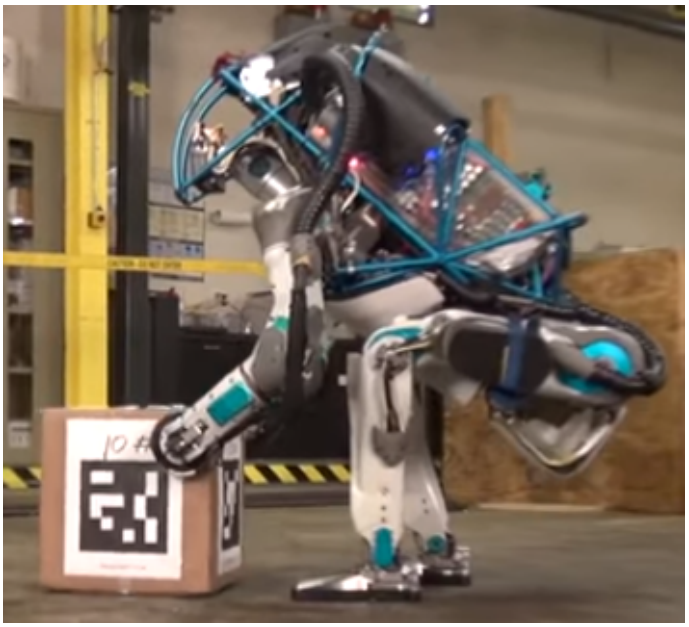


Figure 1.3: ATLAS robot, from Boston Dynamics, picking up a box.

force must be controlled simultaneously, but it is also necessary that the pose of the manipulated object can be modified during task execution (Siciliano et al., 2009).

Early works in this area considered important issues such as master/slave scheme (Soares et al., 2008; Nakano et al., 1974) and force/compliance control (Mason, 1981). In master/slave scheme, the slave arm, controlled by force, follows the master arm, controlled by position. This scheme, however, is not robust enough, since it depends on low impedance of the slave arm for smooth tracking (Jafari and Ryu, 2016) and it also has problems to reverse the role of master and slave arms during operation. In this manner, further works considered non-master/slave approaches, which are more natural, since they consider the desired motion of the object in the task-space as a reference for positioning the arms (Caccavale and Uchiyama, 2008).

Some authors have studied the modeling and control of multiple arms systems as a closed kinematic chain (Freitas et al., 2011; McClamroch, 1986; Tarn et al., 1988). The difficulties for parametrization of the force and moment constraints of the object have motivated the development of other modeling formulations, such as the *symmetric formulation*, proposed by Uchiyama and Dauchez (1988), which uses the concept of *virtual sticks* to simplify the kineto-static model. Several works (Dauchez and Uchiyama, 1987; Uchiyama and Dauchez, 1987, 1988, 1992) have modeled the cooperative system using this approach. However, this formulation can only be used with rigid objects and also depends on the knowledge of the object geometry (Caccavale and Uchiyama, 2008).

On the other hand, the *cooperative task-space formulation*, proposed by Chiacchio et al. (1996, 1993), uses absolute and relative variables to describe the cooperative system and also the direct and inverse kinematic relationships by means of the ab-

(a) DLR LWR-III, from German Aerospace Center

(b) Baxter, from Rethink Robotics

Figure 1.4: Examples of robots with flexible joints, developed for human collaboration.

solute and relative Jacobian. Using the kinematic control strategy, the coordination of the arms configurations can be performed even without a real object being manipulated and does not depend on any assumption for the object geometry (Smith et al., 2012; Caccavale and Uchiyama, 2008). In a posterior publication, Caccavale et al. (2000) has adopted the unit quaternion in the cooperative task-space formulation, to represent the orientation, using a PD-type scheme to control the dual-arm system. Despite the contribution for modeling the cooperative system, these works have not considered the force control problem.

In order to achieve object trajectory tracking and to guarantee a secure grasp, the use of force control in conjunction with motion control has to be conciliated, since the distinct objectives may imply conflicting commands. The force decomposition strategy was investigated by Mason (1981) and emerged as an interesting solution to this problem. In this sense, the hybrid position-force control proposed in (Raibert and Craig, 1981), initially applied to single manipulators, is a method that presents uncoupled force and position control loops, which can be designed independently (Enríquez and Alejo, 2015; Tinos et al., 2006; Yamano et al., 1998). One challenge that can be solved by using hybrid control method is, for example, the interaction of end effector with uncertain geometry contact surfaces (Doulgeri and Karayiannidis, 2007; Leite et al., 2009).

The simplicity and efficiency of the hybrid method led to studies to extend the applicability of single arm hybrid control to cooperative hybrid control (Hayati, 1986; Uchiyama et al., 1987). Several other works followed this research line: (Jafari and Ryu, 2016; Kruse et al., 2015; Farooq and Wang, 2008; Yamano et al., 2004; Sun and Mills, 2002; Uchiyama and Dauchez, 1988; Dauchez et al., 1989, 1991; Uchiyama

and Dauchez, 1992; Fisher and Mujtaba, 1992; Bonitz and Hsia, 1994).

Since the hybrid method uses an explicit force feedback, it is classified as a *direct* interaction method (Siciliano and Khatib, 2008). Another direct method that has been developed is the parallel position-force control (Caccavale et al., 2013; Chiaverini and Sciavicco, 1993). In the parallel scheme, the position and force control loops act simultaneously in the task-space, i.e, there is no selection of force and motion directions, although the force control loop has to dominate the control action (Chiaverini et al., 1994). This characteristic makes the scheme more robust for manipulation of unstructured environments or uncertain objects, at the expense of steady-state position error (Siciliano et al., 2009). On the other hand, *indirect* interaction methods do not require force measurement. The most used is impedance (or compliance) control (Bonitz and Hsia, 1996; Schneider and Cannon, 1992; Hogan, 1984). In this scheme, a desired stiffness behavior is designed and the contact force is estimated according to the deviations in the task-space or joint-space variables, caused by the interaction. Then, a inner motion control loop controls the arms to achieve the desired positioning and the desired compliant behavior (Siciliano and Khatib, 2008; Caccavale et al., 2008). Several works have developed cooperative systems using this type of control (Sieber et al., 2015; Erhart and Hirche, 2014; Erhart et al., 2013; Sadeghian et al., 2012; Caccavale et al., 2008).

The original schemes of hybrid, parallel and impedance control use the robot and the object dynamics. However, some researches have worked on less complex control schemes. Adorno et al. (2010) has proposed kinematic control strategies for dual-arm manipulation and has modeled the cooperative system using the compact representation of dual-quaternions. This work did not consider, though, the forces acting when a object is grasped by the dual-arm system. In addition, Leite et al. (2010) have proposed a purely kinematic hybrid position-force control scheme with independent orientation control, for a single manipulator, and they have developed its stability analysis.

In addition to the discussed methods, other approaches have also emerged in the study of cooperative systems. Although not in the scope of the present work, some advanced control techniques, such as adaptive control (Aghili, 2011; Liu and Arimoto, 1998), neural networks (Panwar et al., 2012) and sliding mode control (Herrmann et al., 2014) treated the problem in a different perspective.

In general, to achieve successful interaction control, the simplest way is to use force sensors coupled to the end effectors. However, these sensors requires customized adaptation of the robot end effector, they add mass to the system and they also increase the costs of the project (Linderoth et al., 2013; Le et al., 2013). External force estimation can be performed using motor torques (Wahrburg et al., 2014; Linderoth et al., 2013; Choi et al., 2012). However, the measures can be noisy under

high reduction rates and compensation for gravity and friction is required (Stolt et al., 2012).

Under this scenario, studies have considered the use of autonomous grasp of objects (Ye et al., 2013; Miller et al., 2003). Among different technologies that can be used to identify an unknown object, the use of cameras attached to the end effectors is an interesting option considering cost/performance and quick setup (Lippiello et al., 2013; Saxena et al., 2008). One feasible solution for pose identification of objects is to adopt fiducial markers (for example, LEDs (Faria et al., 2015)). The projection of the markers in the camera frame is used, in a Perspective n-Point problem, to estimate the relative pose of the object (Lepetit et al., 2009; Petersen, 2008*b*; Oberkampf et al., 1996; Fischler and Bolles, 1981).

1.3 Objectives

In this work, we present a methodology to deal with the problem of cooperative manipulation of objects. The control goal is to grasp and to manipulate an object located inside a robot workspace, according to the references provided by an operator.

The robotic system must be able to estimate the initial object pose, which is used for a proper grasp. After the grasping phase, a hybrid position-force control scheme is employed for the object manipulation. The control design must guarantee that a desired constant contact force is applied to the object, while the robot manipulators conduct the object for tracking a reference trajectory.

Furthermore, an analytical method to estimate interaction force between end effector and the object, using the torque measured at the joints and the vector of gravity forces on the robot manipulators, is presented. The proposed solution can be applied to robotic systems composed of two arms with torque sensors at the joints and cameras attached to each robot wrist.

The main objective, in long term, is to provide grants to develop new advanced control strategies for robotic systems capable of manipulating, safely and efficiently, different types of objects in hazardous or human-populated environments.

1.4 Methodology

In this work, the problem of dual-arm robotic systems acting cooperatively in a load transport task is considered. The system is modeled based on the *cooperative task-space formulation*, proposed in Caccavale et al. (2000) and Chiacchio et al. (1996). While other approaches, as the symmetric formulation, consider the object to be manipulated as a part of the kinematic model, the cooperative task-space

formulation defines meaningful variables – absolute and relative configurations – for the cooperative task, which can be simply computed from the end-effectors position and orientation with respect to a inertial frame. An important advantage of this formulation is to achieve coordination of the arms even without an object to close the kinematic chain.

In order to achieve robust and stable grasp behavior and reduce the end-effectors position accuracy needed during manipulation, interaction control has to be considered. In order to conciliate the position and force objectives, the hybrid position-force control strategy uses selection matrices to divide the task-space in force-controlled and position-controlled directions. Opposed to other methods, as impedance and parallel control, hybrid control presents the advantage of independent control loops design/action and it allows specification of a desired grasp force. The original hybrid control scheme uses an inverse dynamics control law (Villani and De Schutter, 2008; Uchiyama et al., 1987). However, the lack of a simple and efficient method motivated the development of a hybrid position-force kinematic control scheme with orientation (Leite et al., 2010; Leite, 2005). Based on this scheme, a cooperative control method named *dual-arm hybrid position-force kinematic control* is proposed here.

In a brief, the main features of the hybrid scheme developed are: (1) the use of kinematic approach, (2) the orientation problem is time-variant and expressed using quaternions, (3) the manipulation task is 6-DoF (4) the modeling does not assume the object has a known and fixed geometry and (5) the control scheme is centralized and it gives priority to accomplish the cooperative task objective, rather than each manipulator positioning. Additionally, in this work, a method for indirect estimation of the environment forces, using torque sensor information, provides the force feedback required by the hybrid controller.

As part of the cooperative manipulation, dual-arm autonomous grasp of objects is also considered. For this purpose, visual methods using monocular cameras for object pose estimation offer good cost/performance and ease of use over other technologies. Also, the use of passive fiducial markers attached to the object greatly simplify the estimation problem. Following the procedure described in Faria et al. (2015), the problem is formulated as a perspective 4-Point problem and the algorithm entitled Efficient Perspective-n-Point Camera Pose Estimation (EPnP) (Lepetit et al., 2009) is used to obtain the solution. Then, using this pose estimative, the manipulators are repositioned for the grasp closure.

The feasibility of the proposed dual-arm hybrid relative position-force kinematic control method is first verified through simulation (Matlab and Simulink). Then, the experiments are conducted using a real robotic system (Baxter robot – see Figure (1.5)). The robot is composed by two 7-DoF manipulators and each joint contains

a series elastic actuator. Also, Baxter uses the *Robot Operating System* (ROS) as the structural programming software. The results confirm the effectiveness of the proposed method.

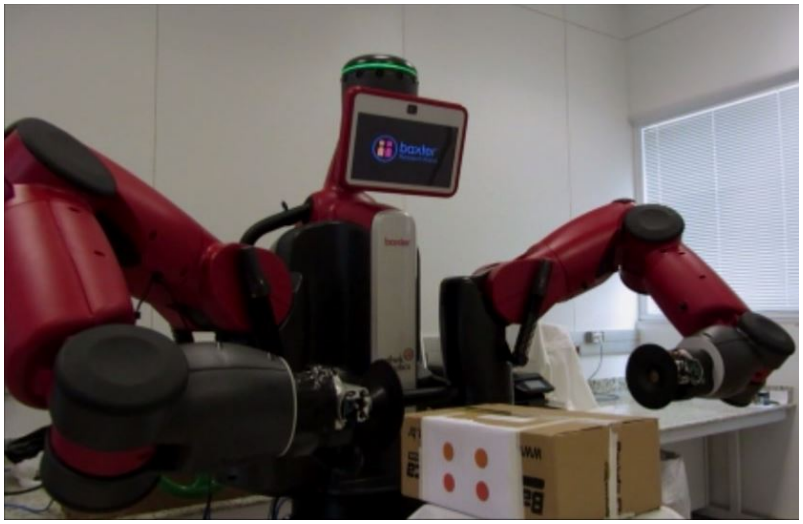


Figure 1.5: Experimental setup – Baxter robot and object with visual markers.

1.5 Contributions

The main contributions of this work are:

- Development of a kinematic-based hybrid position-force control scheme for cooperative manipulation, based on the hybrid control scheme proposed in Leite et al. (2010) for a single manipulator. The scheme proposed here considers the problem of bi-manual cooperative manipulation using the concept of absolute and relative variables (Chiacchio et al., 1996).

  The main contributions of this hybrid scheme are: (1) to extend the kinematic hybrid control from single to multiple manipulators, (2) to demonstrate the feasibility of using, for cooperative manipulation, the hybrid position-force control only in the relative variables, which are significant for maintaining a secure grasp and (3) to give a different interpretation for the constraint frame, originally developed for representing the normal direction of a surface, which now is used to represent the desired force-controlled direction on the object. The constraint frame is assumed to be time-variant and it is updated according to the current object orientation.

- Development of a methodology for autonomous grasping of objects by a dual-arm robot, using visual estimation of the 3D pose of objects. Based on the

ideas proposed in Faria et al. (2015) (see Figure (1.6)) for autonomous manipulation of valves, a single monocular camera is used to determine the pose of an object, equipped with colored visual markers, located in the robot workspace. Then, a grasping algorithm, developed in this work, allows the robot arms to grasp the object autonomously.

(a) Input (b) Output

Figure 1.6: The result obtained from the application of the methodology described in (Faria et al., 2015).

1.6 Organization

This work is organized as follows:

- Chapter 2 - Reviews the *kinematic modeling* of robotic manipulators, including cooperative robots.
- Chapter 3 - Presents the concepts and theoretical development related to the *control* of manipulators, including the cooperative hybrid position-force control proposed. Numerical simulations demonstrate the feasibility of the control schemes.
- Chapter 4 - Proposes an experimental setup. The robot and the environment are described. It also presents the concepts and methods related to visual estimation of the pose of objects. Experiments demonstrate the effectiveness of the visual method and their results are discussed. In the end, the experimental results for the proposed hybrid control method, performed with a Baxter robot, are presented and discussed.
- Chapter 5 - Summarizes the overall conclusions of this work and proposals for future works.

Chapter 2

Kinematic Modeling of a Robotic System

This chapter presents an overview of the main concepts used for the kinematic modeling of robotic systems. It is based on the contents of the books (Murray et al., 1994; Siciliano and Khatib, 2008; Siciliano et al., 2009) as well as the thesis (Leite, 2005, 2011). Some definitions, ideas and techniques presented in these works will be restated here, contributing for a self-contained text.

A robotic manipulator can be interpreted as a chain of rigid bodies and links, connect by joints. At the beginning of this chain is the robot base – usually assigned as the inertial reference – and at the end is the end effector, responsible for interaction with the environment. Each link is moved by a joint and the composition of each joint displacement causes the final movement of the end effector. For object manipulation, it is necessary to describe the end-effector position and orientation (*pose*).

Motion control of the end effector requires analysis of the manipulator structure to elaborate suitable control strategies. This analysis aims to obtain mathematical models representing the manipulator *kinematics* or *dynamics*.

The robot kinematics defines the geometric relationship between the joints movement and the corresponding end-effector movement in task space. Meanwhile, the robot dynamics describes the relationship between generalized forces and the movement driven by joint accelerations, velocities and position (Siciliano et al., 2009).

The goal of this chapter is to present to the reader the theoretical basis to deal with the study of cooperative robotic systems. First, rigid body analysis is presented. After, the forward and differential kinematics models for manipulators are developed. A brief study of robots with *flexible joints* is presented and, finally, modeling of a two-arm robotic system is described.

2.1 Forward Kinematics

The kinematics analysis of a robotic manipulator deals with the description of its movement, without considering the forces and moments which cause it. In this context, the end-effector configuration in *Cartesian space* is completely described by its position and orientation with respect to a reference frame $\mathcal{F}$ (Siciliano et al., 2009).

Remark 1. For a better comprehension, the following notation is adopted in this work:

- $\mathcal{F} = [\vec{x}\ \vec{y}\ \vec{z}]$ denotes an orthonormal frame and $\vec{x}$, $\vec{y}$, $\vec{z}$ denote the unitary vectors which represent the axis of frame $\mathcal{F}$.
- The elements of a vector $\nu \in \mathbb{R}^n$ are denoted as $\nu_i \in \mathbb{R}$ as in:

$$\nu = \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_n \end{bmatrix} \qquad i = 1, \cdots, n \ .$$

- A vector ν expressed in the base frame $\mathcal{F}_b$ is simply written as ν. In the cases a vector is more conveniently expressed in the other frame $\mathcal{F}_i$, it is written as ${}^i\nu$. If ν is used as a desired value in control, the index d is placed at the right upper side as in ν^d.
- The skew-symmetric operator $\mathcal{S}(\cdot)$, used to perform the cross product operation, when applied to a vector $\nu \in \mathbb{R}^3$ is constructed as:

$$\mathcal{S}(\nu) = \begin{bmatrix} 0 & -\nu_z & \nu_y \\ \nu_z & 0 & -\nu_x \\ -\nu_y & \nu_x & 0 \end{bmatrix} \ .$$

A robotic manipulator is a sequence of rigid bodies (links) which are connected by joints. The most common types of joints are the revolute and prismatic joints, but there is also the cylindrical and spherical joints. The links together form a kinematic chain containing at one extremity the robot base and at the other the end effector (Figure 2.1), that is used for manipulation and interaction with the environment. Considering the typological description, the kinematic chain may be of two types: it is called open (Figure 2.2a) when there is only one sequence of links connecting the base and the end effector and **closed** (Figure 2.2b) when the links form a loop and, therefore, there is more than one way to connect the base to the end effector.

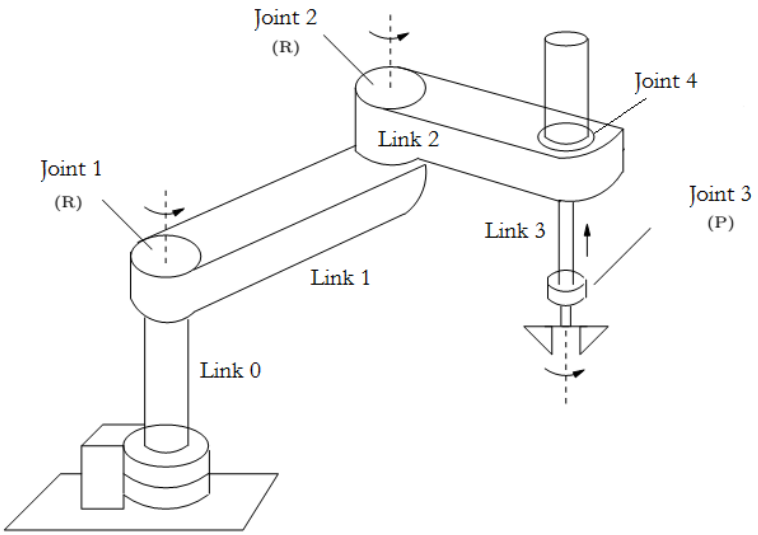
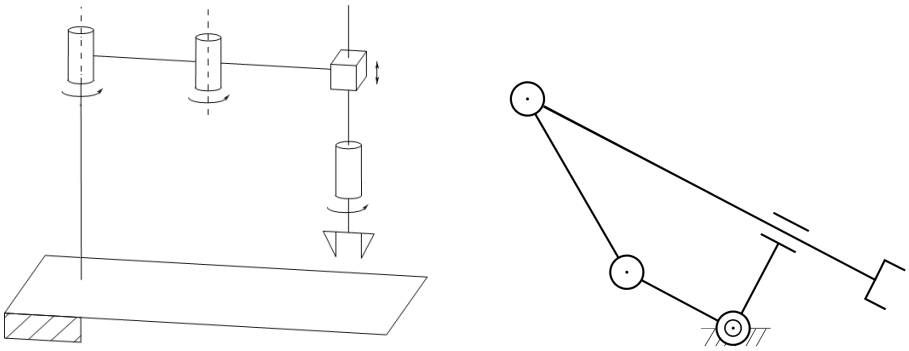


Figure 2.1: Joints and links description of a robotic manipulator. Extracted from (Leite, 2005).



(a) Open-chain manipulator (b) Closed-chain manipulator

Figure 2.2: Examples of kinematic chains. Extracted from (Siciliano et al., 2009).

For a open chain mechanism, each joint (revolute or prismatic) provides the mechanical structure with a single *degree of freedom* (DoF). The degree of freedom is defined as the number of variables necessary to determine the position of all parts of a mechanism in space. By knowing the angular or displacement position of each joint variable, it is possible to determine the end-effector position. Therefore, the main goal of Forward Kinematics is to compute the position and orientation of the manipulator end effector $\mathcal{F}_e$, which is a function of the joint variables, with respect to the base frame $\mathcal{F}_b$ (Siciliano et al., 2009).

The *forward kinematic* for a simple manipulator may be computed by analyzing the structure geometry. Then, using trigonometric relations it is possible to obtain the mapping from the base frame to the end-effector frame as a function of the joint angles. However, the problem becomes harder to solve this way for manipulators with more joints and more complex structure. The search for systematic solutions for the Forward Kinematics problem has led to some procedures such as the *Denavit-*

Hartenberg convention (Siciliano et al., 2009).

Consider an open chain manipulator composed by $n+1$ links attached by n joints. The procedure for calculating the Forward Kinematics comes from the geometrical analysis of the manipulator chain. Since each joint is a connection from one link to other, it is clear that if a relation between two consecutive links is established, then it is possible to obtain, in a recursive way, the complete description of the forward kinematics for the manipulator.

To establish this recursive solution, a coordinate frame must be attached in each link, from link 0 to link n. Then, the homogeneous transformation between the last and the first link is given by the post-multiplication of the homogeneous transformation relating two consecutive links as:

$$T_{0n}(\theta) = T_{01}(\theta_1) \cdot T_{12}(\theta_2) \cdot \ldots \cdot T_{(n-1)n}(\theta_n) \ , \tag{2.1}$$

where $\theta = [\theta_1 \cdots \theta_n]^\mathsf{T} \in \mathbb{R}^n$ is the vector containing each joint displacement $\theta_i \in \mathbb{R}$ (Siciliano et al., 2009).

Hence, the position and orientation relating the end-effector frame $\mathcal{F}_e$ and the base of the robotic system $\mathcal{F}_b$ can be obtained by:

$$T_{be}(\theta) = T_{b0} \cdot T_{0n}(\theta) \cdot T_{ne} \tag{2.2}$$

where T_{b0} and T_{ne} are constant homogeneous transformation matrices relating the first link frame $\mathcal{F}_0$ with the base frame $\mathcal{F}_b$ and the end-effector frame $\mathcal{F}_e$ with the last link frame $\mathcal{F}_n$, respectively.

Therefore, the end-effector position and orientation with respect to the manipulator base can be compactly expressed by the homogeneous transformation

$$T_{be}(\theta) = \begin{bmatrix} R_{be} & p \\ 0_{1\times 3} & 1 \end{bmatrix} \ , \tag{2.3}$$

where $\theta = [\theta_1 \cdots \theta_n]^\mathsf{T} \in \mathbb{R}^n$ is the joint variables vector, $p \in \mathbb{R}^3$ denotes the *position of the end-effector* frame $\mathcal{F}_e$ expressed in the base frame $\mathcal{F}_b$ and $R_{be} \in SO(3)$ is the rotation matrix which denotes the *orientation of the end-effector* frame with respect to the robot base frame.

2.1.1 Joint Space and Operational Space

As seen, the forward kinematics of a manipulator allows to represent the pose of the end effector with respect to a fix reference frame as a function of joint variables, through the relation (2.3). However, the use of this form of description is not computationally efficient, since the orientation matrix give a redundant description

of a frame orientation - it has nine non-independent elements and six restrictions of orthonormality.

It can be shown that just three parameters are sufficient to describe the rigid body orientation in space (Murray et al., 1994). A *representation* that uses three parameters is called *minimal representation*, such as the *Euler* angles The body orientation is expressed using three consecutive rotations – the three angles – about an axis of a moving coordinate frame (Siciliano and Khatib, 2008). However, it is possible to show that this representation always present singularity in case the first and last rotations are about the same axis. The singularity creates problems to relate the body angular velocity to the time derivative of the Euler angles representation, therefore it is not adequate for controlling robotic systems (Siciliano and Khatib, 2008).

Other convenient possibility to represent orientation is to use the unit *quaternion*, which is a non-minimal representation, but is free from singularities and more effective computationally (Hamilton, 1844; Yuan, 1988; Murray et al., 1994). Therefore, it is the representation used in this work.

The quaternion representing the orientation matrix R_{be} is defined as $q = [q_s \ q_v]^\mathsf{T} \in \mathbb{H}$ [1], where $q_s \in \mathbb{R}$ is the scalar part and $q_v \in \mathbb{R}^3$ is the vector part of the unit quaternion, subject to the constraint (Siciliano et al., 2009) :

$$\|q\| = q^\mathsf{T} q = 1 \ . \tag{2.4}$$

Thus, the manipulator configuration (pose) $x \in \mathbb{R}^m$ is described as

$$x = \begin{bmatrix} p \\ q \end{bmatrix} \ , \tag{2.5}$$

where x is expressed in the base frame $\mathcal{F}_b$.

The variable x is defined in the space where the task is performed, which is called *operational space*. The operational space has dimension m, which is the number of parameters to represent position and orientation.

During the task execution, the end effector moves in the operational space, while the joints displace accordingly to perform the trajectory. The joint variables together

[1] the symbol $\mathbb{H}$ denotes the group of unit quaternions satisfying the quaternion algebra (Wen and Kreutz-Delgado, 1991; Siciliano et al., 2009).

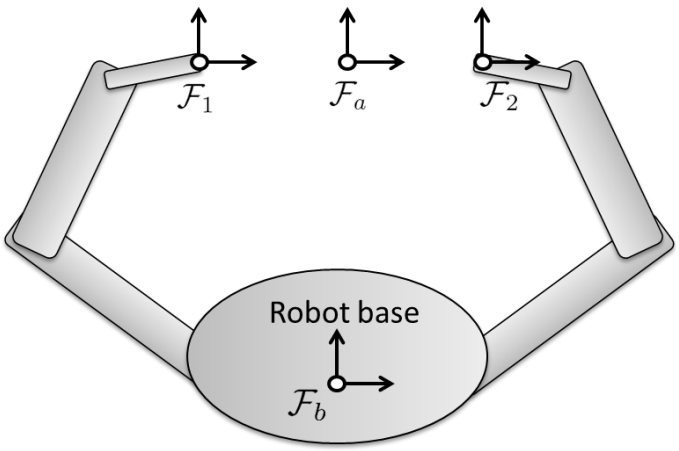


Figure 2.3: The left $\mathcal{F}_1$ and right $\mathcal{F}_2$ end-effectors frames and the robot base frame $\mathcal{F}_b$, which is the common frame for the cooperative system.

form the *joint space*, represented by the vector

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} , \tag{2.6}$$

where n – the number of joints – is the dimension of the joint space.

Finally, the operational space is connected to the joint space according to the relations:

$$p = f(\theta) , \qquad q = h(\theta) , \tag{2.7}$$

where $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^3$ is a non-linear mapping from joint space to Cartesian space and $h(\cdot) : \mathbb{R}^n \mapsto \mathbb{H}$ is a non-linear mapping from the joint space to quaternion space.

2.1.2 Absolute and Relative Configuration

When dealing with two or more manipulators, the Forward Kinematics description given by (2.5) can be applied to each manipulator individually. However, cooperative tasks require coordination of the multiple end effectors and a centralized approach to deal with the distinct endpoint configurations at all. In this context, the system is more appropriately represented by means of absolute and relative variables (Chiacchio et al., 1996).

For simplicity, consider a two-arm robot where $x_1 = [p_1 \;\; q_1]^\mathsf{T}$ and $x_2 = [p_2 \;\; q_2]^\mathsf{T}$ represent the configuration of the left and right end effectors, respectively, both expressed at a common robotic system base frame $\mathcal{F}_b$, as illustrated in Figure 2.3. The absolute frame $\mathcal{F}_a$ is defined as the frame positioned at

$$p_a = \frac{p_1 + p_2}{2} \ , \tag{2.8}$$

where $p_a \in \mathbb{R}^3$ is the *absolute position* vector, expressed in the base frame $\mathcal{F}_b$. The physical meaning of the absolute frame $\mathcal{F}_a$ is to represent the mean-value point obtained from the manipulators endpoints, which for a load-transportation task may be interpreted as the object center. However, the absolute variables solely are not enough to specify the cooperative movement since there exist infinite end-effector configurations giving the same absolute position and orientation. Indeed, when two arms move away/approximate in the same direction, $\mathcal{F}_a$ remains unchanged. Therefore, the description of the cooperative system also depends on the *relative* configuration of the manipulators.

Besides, the relative position $p_r \in \mathbb{R}^3$ is defined as

$$p_r = p_2 - p_1 \ , \tag{2.9}$$

where p_r is expressed in the base frame. It is important to note that the *desired relative position* is usually specified at the absolute frame $\mathcal{F}_a$, in order to be independent of the absolute movement. The relationship between ${}^a p_r^d$, the desired relative position expressed in the frame $\mathcal{F}_a$ and p_r^d, which is expressed in the base frame $\mathcal{F}_b$, is simply given by the rotation matrix R_{ba}.

Meanwhile, the relative orientation q_r between the two end-effector frames, expressed in frame $\mathcal{F}_1$, is given by

$${}^1 q_r = q_1^{-1} * q_2 \ , \tag{2.10}$$

where $*$ denotes the quaternion product operator, q_1^{-1} represents the conjugate of the unit quaternion representing orientation of the left end effector, q_2 is the unit quaternion representing orientation of the right end effector.

Additionally, let

$${}^1 q_r^{1/2} = \left\{ \cos\left(\frac{\theta_r}{4}\right), \sin\left(\frac{\theta_r}{4}\right) {}^1 k_r \right\} \ , \tag{2.11}$$

denote the quaternion, expressed at frame $\mathcal{F}_1$, representing half the rotation needed to align $\mathcal{F}_1$ and $\mathcal{F}_2$, where θ_r is the angle and ${}^1 k_r$ the unit axis, expressed in the end-effector frame $\mathcal{F}_1$, equivalent to the rotation $R_r = R_{12}$ (Adorno, 2011). This variable is interpreted as the mean-value rotation obtained from the manipulators orientation. When expressed at the base frame $\mathcal{F}_b$, it is used as the *absolute orientation* q_a, being defined as:

$$q_a = q_1 * {}^1 q_r^{1/2} \ , \tag{2.12}$$

which completes the cooperative variables description.

2.2 Differential Kinematics and Statics

The differential kinematics establishes a relation between the joint velocities and the corresponding end-effector linear and angular velocities. It is possible to describe this mapping by means of a matrix, called *Geometric Jacobian*, which is dependent on the manipulator configuration and is obtained by analyzing the geometry of the structure (Murray et al., 1994; Siciliano et al., 2009).

As discussed in section 2.1.1, the end-effector configuration expressed by vector $x = [p \ q]^\mathsf{T}$ is related with the joint angles vector θ by a vectorial function $[f(\cdot) \ h(\cdot)]^\mathsf{T}$. Differentiation of this function leads to a different mapping matrix, called *Analytical Jacobian*.

The Jacobian is a mapping matrix that not only characterizes the manipulator, but also allows: (a) to analyze redundancy; (b) to find singular configurations; (c) to map the forces applied at the end effector and the resulting torques at the joints; (d) to link the operational and the joint spaces and (e) to design control in operational space (Siciliano et al., 2009).

2.2.1 Geometric Jacobian

The objective of differential kinematics is to relate the joint velocities and the end-effector linear and angular velocities. It means to express the linear velocity v and the angular velocity ω as a function of joint velocities $\dot{\theta}$, according to the following relations:

$$v = J_p(\theta)\,\dot{\theta} \ , \tag{2.13}$$

where $J_p \in \mathbb{R}^{3\times n}$ is the matrix, called *position Jacobian*, mapping the joint velocities $\dot{\theta}$ to the linear velocity of the end effector ($v = \dot{p}$) and

$$\omega = J_o(\theta)\,\dot{\theta} \ , \tag{2.14}$$

where $J_o \in \mathbb{R}^{3\times n}$ is the matrix, called *orientation Jacobian*, which maps the joint velocities $\dot{\theta}$ to the angular velocity of the end effector $\omega \in \mathbb{R}^3$, expressed in the base $\mathcal{F}_b$. In a compact way, these relationships are written as

$$\mathsf{v} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} J_p(\theta) \\ J_o(\theta) \end{bmatrix} \dot{\theta} = J(\theta)\,\dot{\theta} \tag{2.15}$$

which represents the Differential Kinematics of a manipulator. Vector v is called the *Cartesian velocity vector* of the end effector, expressed in the base frame $\mathcal{F}_b$, and the

matrix $J(\theta)$ is the *Geometric Jacobian*, which depends on the joint variables. As it can be noticed, the Geometric Jacobian is a mapping from joint space to operational space, that is, $J : \Theta \mapsto \mathcal{T}$. From the analysis of the manipulator it is possible to interpret the end-effector velocity as a contribution of each joint movement (Siciliano et al., 2009).

2.2.2 Analytical Jacobian

In the last subsection, a geometric method for calculating the Jacobian was presented. This technique allows to determine the contribution of each joint velocity to the end-effector velocity v. However, if the end-effector pose is described using vector $x = [p \;\; q]^\mathsf{T}$, it is possible to compute the Jacobian by differentiation of the forward kinematics mapping.

The end-effector linear velocity $\dot{p}$ is obtained by the time derivative of the position mapping $f(\cdot)$:

$$\dot{p} = \frac{\partial f}{\partial \theta}\,\dot{\theta} = J_p(\theta)\,\dot{\theta} \;, \tag{2.16}$$

where $J_p \in \mathbb{R}^{3\times n}$ is called the *position Jacobian*.

Since in this work we consider particular case of the quaternion representation, the end-effector orientation velocity is given by the time derivative of the orientation mapping $h(\cdot)$:

$$\dot{q} = \frac{\partial h}{\partial \theta}\,\dot{\theta} = J_q(\theta)\,\dot{\theta} \;, \tag{2.17}$$

where $J_q \in \mathbb{R}^{4\times n}$ is called the *quaternion Jacobian*.

Then, the differential kinematics relation can be summarized as:

$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} J_p \\ J_q \end{bmatrix} \dot{\theta} = J_A(\theta)\dot{\theta} \;, \tag{2.18}$$

where $J_A = [\frac{\partial f}{\partial \theta} \;\; \frac{\partial h}{\partial \theta}]^\mathsf{T} \in \mathbb{R}^{7\times n}$ is called the Analytical Jacobian, which is different from the Geometric Jacobian J, since the angular velocity ω is not the same as the time derivative $\dot{q}$ of the orientation representation.

2.2.3 Representation Jacobian

In the study of Kinematics, it is useful to obtain a relationship between the Analytical and the Geometric Jacobian. For this purpose, the angular velocity ω and the quaternion derivative $\dot{q}$ must be related. Consider the evolution in time of the quaternion, that is, the quaternion propagation, with respect to inertial frame $\mathcal{F}_b$,

given by: (Wen and Kreutz-Delgado, 1991)

$$\dot{q} = \frac{1}{2} B_q(q)^\mathsf{T} \omega \ , \tag{2.19}$$

$$B_q(q) = \left[-q_v \ \vdots \ q_s \, I - \mathcal{S}(q_v) \right] \ , \tag{2.20}$$

where $B_q \in \mathbb{R}^{3\times 4}$ relates the angular velocity with the derivative of the end-effector orientation, in quaternion.

Then, the end-effector velocity v is related with the derivative of the configuration x as:

$$\mathsf{v} = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 2\, B_q(q) \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix} = J_R(q)\, \dot{x} \ , \tag{2.21}$$

where $J_R(q) \in \mathbb{R}^{6\times 7}$ is the *Representation Jacobian* for the quaternion representation.

Finally, substituting equation (2.18) in equation (2.21) and comparing with equation (2.15), leads to the relation between the Analytical and the Geometric Jacobian:

$$J(\theta) = J_R(q)\, J_A(\theta) \ . \tag{2.22}$$

2.3 Statics

The main goal of Statics is to establish a relationship between the generalized forces applied at end effector and the generalized forces applied at the joints – forces for prismatic joints, torques for revolute joints – considering the manipulator at an equilibrium configuration (Siciliano et al., 2009). The application of the *principle of virtual work* allows to obtain this relationship.

Let $\tau \in \mathbb{R}^n$ denote the vector of joint torques and $F = [f \ \ \mu]^\mathsf{T} \in \mathbb{R}^m$ the vector of end-effector *generalized forces*, composed by the linear force $f \in \mathbb{R}^3$ and by the moment $\mu \in \mathbb{R}^3$. The manipulators considered are time-invariant systems with holonomic constraints, therefore their configurations depend only on the joint variables θ – they do not depend explicitly on the time. Thus, virtual displacements coincide with elementary displacements. The elementary work associated with joint torques is

$$dW_\tau = \tau^\mathsf{T} d\theta \ . \tag{2.23}$$

For the end-effector forces, separating the contributions from force f and moment μ, the elementary work associated is

$$dW_F = f^\mathsf{T} dp + \mu^\mathsf{T} \omega \, dt \ , \tag{2.24}$$

where dp represents the linear displacement and $\omega\, dt$ the angular displacement. Considering differential kinematics, we can reevaluate relation (2.24) as

$$dW_F = f^\mathsf{T} J_p(\theta) d\theta + \mu^\mathsf{T} J_o(\theta) d\theta \tag{2.25}$$

$$= F^\mathsf{T} J(\theta) d\theta \ . \tag{2.26}$$

Since virtual and elementary displacements coincide, the corresponding virtual works are

$$dW_\tau = \tau^\mathsf{T} \delta\theta \ , \tag{2.27}$$

$$dW_F = F^\mathsf{T} J(\theta) \delta\theta \ , \tag{2.28}$$

where δ is used to represent a virtual quantity. According to the principle of virtual work, the manipulator is at static equilibrium if and only if

$$\delta W_\tau = \delta W_F \qquad \forall\ \delta\theta \ , \tag{2.29}$$

that means, for every joint displacement the difference between the virtual work of the joint torques and the virtual work of the end-effector forces must be null.

Finally, substituting (2.27) and (2.28) into (2.29), leads to

$$\tau^\mathsf{T} = F^\mathsf{T} J(\theta) \implies \tau = J^\mathsf{T}(\theta) F \ . \tag{2.30}$$

As it can be noted, the Geometric Jacobian $J(\theta)$ is responsible for the static relationship between generalized forces acting at the end effector and the joint torques of the manipulator.

2.3.1 Force Estimation based on Joint Torques

The objective of the force estimation method employed in this work is to determine the forces arising at the end effectors when a robotic system interacts with an external object, without using force sensors coupled at the tip of the end effectors. In some cases, such as robots with flexible joints, torque sensors are generally available and it is possible to use the measurement information to perform the estimation of the applied force.

The employed method decomposes the motor torque $\tau_m \in \mathbb{R}^n$ measured at the sensors coupled to the flexible joints as:

$$\tau_m = \tau_g + \tau_{din} + \tau_{ext} + \tau_d \ , \tag{2.31}$$

where $\tau_g \in \mathbb{R}^n$ is the vector of torques originated from gravity action on the manipu-

lator links, $\tau_{din} \in \mathbb{R}^n$ is the vector of dynamic torques generated by the accelerations of the structure, $\tau_{ext} \in \mathbb{R}^n$ is the vector of torques generated by the action of external elements, such as reaction from environment interaction and $\tau_d \in \mathbb{R}^n$ is the vector of torques considered as disturbance, which includes modeling errors, friction or measurement noises.

Notice that, dynamic torques τ_{din} can be calculated from the dynamic parameters of the robot. However, for the tasks of interest in this work, it may be considered negligible, because when a robot is interacting with an object or with the external environment, it is not suitable to use high speeds (Linderoth et al., 2013).

It is also worth mentioning that disturbance torques are complex to estimate and can be assumed as Gaussian white noise (Choi et al., 2012). For the sake of simplicity, in this work it is considered that this component is not relevant, implying therefore in small error in the estimation of external force.

External torques τ_{ext} are produced by the interaction of the robotic system with an object or the environment, which generates external forces F_{ext}. It is considered that these external forces are acting on the end effector, what actually happens in the case of manipulation after the grasp.

In the last section, the static relationship between generalized forces, acting at end effector, and joint torques, acting at the manipulator joints was presented. Multiplying both sides of (2.30) by the Jacobian pseudo-inverse $J(\theta)^\dagger$ matrix (Siciliano et al., 2009) leads to

$$F_{ext}^\mathsf{T} J(\theta) J(\theta)^\dagger = \tau_{ext}^\mathsf{T} J(\theta)^\dagger \ . \tag{2.32}$$

Since $J(\theta)J(\theta)^\dagger = I$ and assuming τ_{din} and τ_d are neglected, the estimation of the external force applied to the end effector can finally be obtained by knowing the measured torques and the gravitational torques:

$$F_{ext}^\mathsf{T} = \tau_{ext}^\mathsf{T} J(\theta)^\dagger = (\tau_m - \tau_g) J(\theta)^\dagger \ . \tag{2.33}$$

Notice that the gravitational torques τ_g can be estimated when the position and mass of each manipulator link is known a priori (Choi et al., 2012; Linderoth et al., 2013).

2.4 Cooperative Manipulators

A cooperative system is, in general, composed by M manipulators, each one equipped with N_i joints ($i = 1, ..., M$). In this robotic system, the position $p_i \in \mathbb{R}^3$ and orientation $R_i \in SO(3)$ of each end-effector frame $\mathcal{F}_i$ (or, equivalently $q_i \in \mathbb{H}$) is a function of the joint variables, according to Forward Kinematics (Section 2.1).

Now, consider a cooperative system manipulating a common object (Figure 2.4).

Let C be a fixed reference point of the object and $\mathcal{F}_C$ the corresponding coordinate frame with origin in C. The work done by Uchiyama and Dauchez (1988) proposed to model the system using *virtual sticks*, which are position vectors $r_i \in \mathbb{R}^3$ connecting frame $\mathcal{F}_C$ to each end-effector frame $\mathcal{F}_i$.

If the manipulated object is rigid and tightly grasped, then the virtual sticks are constant vectors when expressed at the corresponding end-effector frame $\mathcal{F}_i$. In this case, the forward kinematics of each manipulator may be determined considering the virtual stick as a part of the manipulator, that is, as a virtual link. The frame representing the tip of each virtual end-effector frame is defined as $\mathcal{F}_{S,i}$ and coincides with $\mathcal{F}_C$. Hence, the position and orientation of each virtual end effector is given by

$$p_{S,i} = p_C \ , \qquad R_{S,i} = R_C \ , \tag{2.34}$$

where p_C and R_C are respectively the position and orientation of frame $\mathcal{F}_C$, expressed in the base frame $\mathcal{F}_b$.

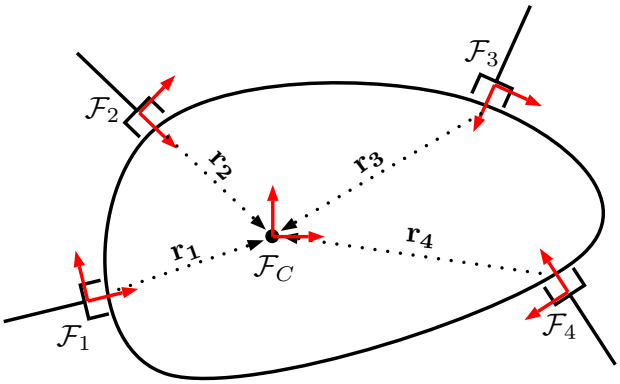


Figure 2.4: Model of multiple arm system manipulating a load.

Now, let $F_{S,i}$ be the vector of generalized forces acting at the tip of the i-th virtual stick. A relationship between the forces at the virtual and at the real end effector can be established by

$$F_{S,i} = \begin{bmatrix} I_3 & O_3 \\ -\mathcal{S}(r_i) & I_3 \end{bmatrix} F_i = W_i \, F_i \ . \tag{2.35}$$

By means of the virtual work principle, a dual relation can be derived as

$$\mathsf{v}_i = \begin{bmatrix} I_3 & \mathcal{S}(r_i) \\ O_3 & I_3 \end{bmatrix} \mathsf{v}_{S,i} = W_i^{\mathsf{T}} \, \mathsf{v}_{S,i} \ , \tag{2.36}$$

where $\mathsf{v}_{S,i}$ is the generalized velocity of the virtual stick. It is worth noticing that if the forward kinematics of each manipulator is performed including the virtual stick as a link, then $r_i = 0$ and $W_i = I_6$. This means that, in this case, the forces

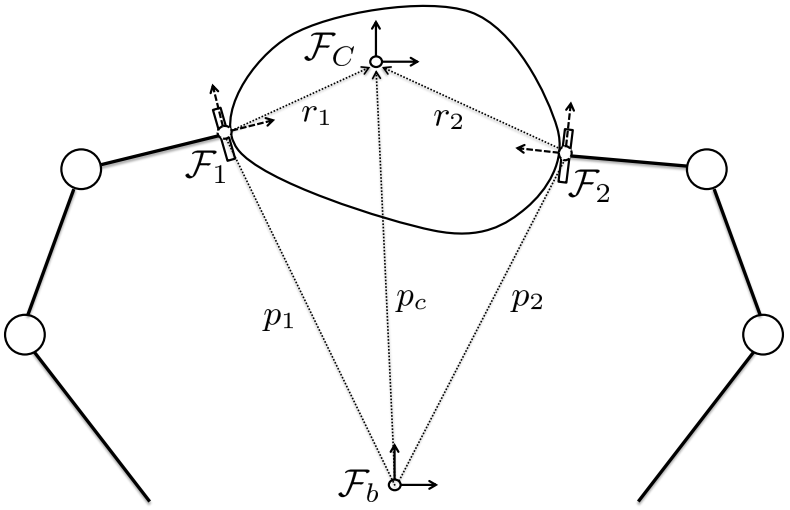


Figure 2.5: Model of a two-arm robotic system manipulating a common object. Model extracted from (Siciliano and Khatib, 2008).

and velocities at each virtual end-effector tip are coincident, which can be used to simplify the static relationship (Uchiyama and Dauchez, 1988).

Note that the validity of this model depends on non-deformation of the object. Hereafter, it is assumed that both the object and the manipulator can be considered rigid (or nearly rigid) and the grasp is tight (or nearly tight) such that the displacements between the end effectors and the contact points can be considered negligible.

Robots with two arms are quite common in industry and are suitable for manipulation tasks. Hence, in this work, we are specially interested in the case of two cooperative robots manipulating a common object. Studies considering these cases have developed more specific formulations, for instance the *Task-Space formulation* and the *Symmetric formulation*, which were conceived for modeling the cooperative operational space of a two-arm system.

2.4.1 Symmetric Formulation

The *symmetric formulation*, proposed by Uchiyama and Dauchez (1992, 1988), deals with the problem of a two-arm robotic system handling a common object (Figure 2.5). This formulation is based on kinematic and static relationships between generalized forces/velocities acting at the object and their counterparts acting at the end effectors (or virtual sticks) (Caccavale and Uchiyama, 2008).

Let $F_E \in \mathbb{R}^{6\times 1}$ be the vector of *external* generalized forces acting at the object,

expressed in the base frame $\mathcal{F}_b$, which is given by the contribution of the two arms:

$$F_E = F_{S,1} + F_{S,2} = W_1 F_1 + W_2 F_2 = W F \ , \tag{2.37}$$

where $W = [W_1 \quad W_2]$ and $F = [F_1 \quad F_2]^\mathsf{T}$. Note that matrix $W \in \mathbb{R}^{6\times 12}$ describes the geometry of the grasp and thus is called the *grasping matrix*.

Vector F_E represents the resultant force which causes movement in the object. However, when the manipulators compress the object, it is expected that internal forces arise in the object without causing any movement. Therefore, it is important to point that the inverse solution of (2.37) is given by

$$F = W^\dagger F_E + V F_I \ , \tag{2.38}$$

where F_I is the vector of *internal* generalized forces, expressed in the base frame $\mathcal{F}_b$, which represent the internal stresses acting at the object, and $V \in \mathbb{R}^{12\times 6}$ is a matrix such that its columns span the null space of W – one possible choice for V is presented at (Chiacchio et al., 1991). This solution can also be represented as:

$$F = U F_O \ , \tag{2.39}$$

where

$$U = [W^\dagger \quad V] \ , \qquad F_O = [F_E \quad F_I] \ . \tag{2.40}$$

By using the duality between generalized forces and generalized velocities, from (2.39) the following mapping is also obtained:

$$\mathsf{v}_O = U^\mathsf{T} \mathsf{v} \ , \tag{2.41}$$

where $\mathsf{v}_O = [\mathsf{v}_E^\mathsf{T} \quad \mathsf{v}_I^\mathsf{T}]^\mathsf{T}$ and $\mathsf{v} = [\mathsf{v}_1^\mathsf{T} \quad \mathsf{v}_2^\mathsf{T}]^\mathsf{T}$. Vector v_E can be interpreted as the *absolute velocity* of the object, while v_I means the *relative velocity* between frames $\mathcal{F}_1$ and $\mathcal{F}_2$, expressed in the base frame $\mathcal{F}_b$, which is null if the object is rigid and tightly grasped. Besides, p_E and $p_I \in \mathbb{R}^3$ are the absolute and the relative position variables, respectively, of the cooperative symmetric model, which are defined using the virtual sticks positions:

$$p_E = \frac{p_{S,1} + p_{S,2}}{2} \ , \qquad p_I = p_{S,2} - p_{S,1} \ . \tag{2.42}$$

The orientation in the symmetric model is given by the absolute orientation R_E and relative orientation R_I matrices, which are the rotation matrices corresponding to the quaternions defined in (2.12) and (2.10).

Finally, (2.39) and (2.41) can be used to complete the kinetostatic mappings

relating force and velocity acting at the object and the corresponding effects at the manipulators joints:

$$\tau = J_o^{\mathsf{T}} F_O \ , \tag{2.43}$$

$$\mathsf{v}_O = J_o \dot{\theta} \ , \tag{2.44}$$

where

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \ , \quad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \ , \quad J_o = U^{\mathsf{T}} J \ , \quad J = \begin{bmatrix} J_1 & O_6 \\ O_6 & J_2 \end{bmatrix} \ . \tag{2.45}$$

2.4.2 Task-Space Formulation

A different approach for modeling the two-arm manipulation problem was proposed by (Chiacchio et al., 1996; Caccavale et al., 2000), which states that the cooperative system can be described by *unique* absolute and relative variables, which can be directly computed from the pose of the end-effectors frames. Moreover, the proposed task formulation allows specification of coordinated movements of a two-manipulator system and, as a consequence, does not assume that an object is being held by the end effectors.

The specification of movement for the system using each manipulator position and orientation separately is inadequate, since coordination would be left to the user. In this context, a global description of the system should be performed, according to the relations (2.8 - 2.12).

Moreover, it can be noticed that this formulation does not depend on any assumption on the object or on the grasp (Caccavale and Uchiyama, 2008). This means the cooperative system is described by task space variables, which can be used to control the arms in a pure motion coordinated task even without any manipulated objected. Remarkably, the object/grasp may be non-rigid which represents a great difference over the symmetric formulation.

For instance, consider the task of moving a tightly grasped object. After the grasp, it is desired to keep constant the relative position p_r and orientation q_r, while the desired absolute position p_a and orientation q_a may vary according to the operator commands. Other tasks may be performed and the proper specification for each task variable should be analyzed.

It is also important to study the differential kinematics of the coordinated motion, which relates the velocities of each manipulator with the corresponding velocities of the cooperative system variables. This is specially important for the algorithmic solution of the inverse kinematics and for the analysis of the stability of the system.

The absolute linear velocity $\dot{p}_a \in \mathbb{R}^3$ of the cooperative system, expressed in the

base frame $\mathcal{F}_b$, is obtained from (2.8) as:

$$\dot{p}_a = \frac{\dot{p}_1 + \dot{p}_2}{2} \ . \tag{2.46}$$

In a similar way, it can be demonstrated (Chiacchio et al., 1996; Caccavale et al., 2000) that absolute angular velocity $\dot{\omega}_a \in \mathbb{R}^3$, expressed in the base frame $\mathcal{F}_b$, is given by:

$$\omega_a = \frac{\omega_1 + \omega_2}{2} \ . \tag{2.47}$$

Once again, the time derivative of (2.9) leads to the relative linear velocity $\dot{p}_r \in \mathbb{R}^3$:

$$\dot{p}_r = \dot{p}_2 - \dot{p}_1 \ , \tag{2.48}$$

which is also expressed in the base frame $\mathcal{F}_b$.

The relative angular velocity $\dot{\omega}_r \in \mathbb{R}^3$, expressed in the base frame $\mathcal{F}_b$, is simple defined as:

$$\omega_r = \omega_2 - \omega_1 \ . \tag{2.49}$$

Finally, by using the principle of virtual work the dual variables for the absolute and relative twists can be obtained: (Caccavale and Uchiyama, 2008)

$$f_a = f_1 + f_2 \ , \tag{2.50}$$

$$\eta_a = \eta_1 + \eta_2 \ , \tag{2.51}$$

$$f_r = \frac{f_2 - f_1}{2} \ , \tag{2.52}$$

$$\eta_r = \frac{\eta_2 - \eta_1}{2} \ , \tag{2.53}$$

where f_a and $\eta_a \in \mathbb{R}^3$ are the absolute force and absolute moment, while f_r and $\eta_r \in \mathbb{R}^3$ are the relative force and relative moment vectors.

It is important to note that the variables defined by the task-space formulation and those used in symmetric formulation are related via simple mappings. Orientation and forces always coincide, while position and moments coincide only if the forward kinematics of the manipulator considers the virtual stick as a link, that is, if $p_1 = p_{S,1}$ and $p_2 = p_{S,2}$ (Caccavale and Uchiyama, 2008).

The coordinated movement approach can be used for control systems in different ways. For example, (Caccavale et al., 2000) uses the formulation, but design individual task-space controllers. However, in this work a centralized control scheme is investigated as in (Chiacchio et al., 1996). The control of cooperative systems using this formulation will be discussed in the next chapter.

Chapter 3

Cooperative Control Strategies

In this chapter, control strategies for cooperative manipulation are discussed. The objective is to develop a simple and efficient method, which is able to achieve position tracking and force regulation and that does not depend on the dynamic parameters of the robotic system and of the object. For better comprehension, a step-by-step study is proposed.

Initially, the kinematic control of a single manipulator is described, where position, force and orientation control schemes are independently analyzed. Then, the problem is extended for the kinematic control of a cooperative system, which is modeled using the cooperative task-space formulation.

The first objective proposed for the cooperative manipulation is to achieve coordinated motion of the manipulators. Based on the approach developed in (Chiacchio et al., 1996), the differential kinematics relation is expressed in terms of the absolute and relative Jacobian, which allows to obtain the absolute and relative spatial velocities from the joint velocities. Therefore, the kinematic description of the system is not performed for each end-effector frame, but for the absolute and relative frames. In order to verify the behavior of the control scheme described, the coordinated motion is simulated and the robotic system is able to track the desired absolute and relative position and orientation of reference frames.

In the sequence, the force control problem is added in the analysis. Conflicts between the force and position objectives are presented and the classical methods for solving them – impedance control and hybrid position-force control – are discussed. These methods, however, depend on the dynamic model of the robot and, often, are based in formulations that require knowledge about the object properties.

To deal with these limitations, a new hybrid position-force kinematic control scheme is proposed for dual-arm robots, based on the hybrid kinematic control scheme developed in (Leite et al., 2010) for a single manipulator. This scheme uses the absolute and relative pose of the end effectors as well as the relative force applied by the end effectors to manipulate and maintain the grasp of an object. The

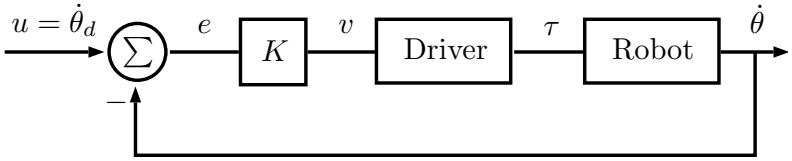


Figure 3.1: Kinematic control diagram.

stability analysis is verified using the Lyapunov Stability Theory and the numerical simulation results confirm the viability and illustrate the performance of this new control scheme.

3.1 Single-arm Kinematic Control

Consider the kinematic control problem for a robotic manipulator with n joints. It is assumed here that the robot has a high performance internal velocity control loop that directly drives the motors. Then, the robot movement can be described by:

$$\dot{\theta}_i = u_i \,, \quad i = 1, \cdots, n \,, \tag{3.1}$$

where θ_i and $\dot{\theta}_i$ are, respectively, the joint angular position and velocity; u_i is the velocity control signal applied to the i-th motor drive of the i-th joint. The kinematic control assumption is valid on the following assumptions:
(A-3.1) The forward kinematics of the manipulator is known.
(A-3.2) The dynamic effects may be neglected. This assumption is indeed feasible if the joints gear ratios are elevated and the interest tasks are smooth and do not require high speed/acceleration (Siciliano et al., 2009).

Figure 3.1 presents the block diagram for joint velocity control. The control signal v is generated by the proportional controller with a gain, K, which scales the error signal e obtained from the comparison between the desired joints position θ^d and the actual joints position θ. Then, the motor driver amplifies the signal and provide the power needed to actuate the motors, generating torques τ at the joints. Considering a positive definite high-gain matrix K is used, it implies that $e \to 0$ and $\dot{\theta} \approx u$, as expected in the kinematic control assumption.

From (2.15) and considering the kinematic control approach (3.1), the control system is given by:

$$\begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(\theta)\, u \ . \tag{3.2}$$

where $u \in \mathbb{R}^n$ is the vector of velocity control signals. A control signal $v_k(t) \in \mathbb{R}^m$ designed in Cartesian Space can be transformed in a Joint Space control signal by

using the linearizing relation:

$$u = J^{-1}(\theta)\, v_k\,, \qquad v_k = \begin{bmatrix} v_p \\ v_o \end{bmatrix}\,, \tag{3.3}$$

where $v_p, v_o \in \mathbb{R}^3$ are the Cartesian control signals, which are designed to control the position and orientation of the end effector, respectively. However, this solution only is possible if $J(\theta)$ is square and has full rank. Note that by using this solution the system can be described as:

$$\begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = \begin{bmatrix} v_p \\ v_o \end{bmatrix}\,. \tag{3.4}$$

If the manipulator is redundant, the Jacobian matrix is not square ($m < n$) and, as a consequence, not invertible. A well-known methodology for finding feasible solutions is to formulate the kinematic control as a constrained linear optimization problem (Siciliano et al., 2009). The control signal applied to the joints may be obtained as:

$$u = J^{\dagger}\, v_k\,, \tag{3.5}$$

where the matrix $J^{\dagger} = J^{\mathsf{T}}(JJ^{\mathsf{T}})^{-1} \in \mathbb{R}^{m\times n}$ is the right pseudo-inverse of J. The solution obtained (3.5) locally minimizes the norm of the joint velocities (Siciliano et al., 2009). It should be noted that the inversion of Jacobian matrix presents mathematical problems near singularities. In fact, the Jacobian determinant becomes small near singularities, so the inverse (or pseudo-inverse) matrix has large norm and thus, high velocities may appear in the joint space when using (3.5).

One solution to overcome the inversion problem in the singularity neighborhood is to use the *Damped Least-Square* (DLS) method, presented by (Nakamura and Hanafusa, 1986). Following this approach, a modified pseudo-inverse Jacobian is given by:

$$J^{\dagger} = J^{\mathsf{T}}\, \left(J\, J^{\mathsf{T}} + \lambda^2\, I \right)^{-1}\,, \tag{3.6}$$

where $\lambda \in \mathbb{R}$ is a damper factor which allows a better conditioned inversion. This factor establishes a trade-off between precision in the solution and feasibility may be chosen as:

$$\lambda = \begin{cases} 0 & , \quad w_m \geq w_0 \\ \lambda_0\,(1 - \frac{w_m}{w_0})^2 & , \quad w_m < w_0 \end{cases} \tag{3.7}$$

where $w_m = \sqrt{det\,(J\,J^{\mathsf{T}})} \in \mathbb{R}$ is called the *manipulability* of the manipulator and $\lambda_0 \in \mathbb{R}$ and $w_0 \in \mathbb{R}$ are constants chosen according to the mechanical structure of the manipulator and the maximum desired damping.

Finally, it is worth noting that other methods exist in the literature for selecting

optimal values of λ (Deo and Walker, 1992, 1995; Martinet et al., 2008) and also to solve the kinematic singularities problem as the methods present in (Wampler et al., 1986) and the filtered inverse approach (Vargas et al., 2014).

3.1.1 Position Control

Consider the position kinematic control problem for a manipulator. The control objective is to make the current end-effector position p track a desired time-varying trajectory $p^d(t)$:

$$p \to p^d(t), \quad e_p = p^d(t) - p \to 0, \tag{3.8}$$

where $e_p \in \mathbb{R}^3$ is the position error. For this objective, the applied control signal v_p may be chosen as a proportional plus feed-forward action:

$$v_p = \dot{p}^d + K_p\, e_p \tag{3.9}$$

where $K_p = k_p I \in \mathbb{R}^3$ is the proportional gain matrix. From (3.4), we have that $\dot{p} = v_p$ and, therefore, the error dynamics can be written as

$$\dot{e}_p = \dot{p}^d - \dot{p} = -K_p\, e_p \ , \tag{3.10}$$

and by choosing a positive k_p value implies that $\lim_{t\to\infty} e_p(t) = 0$.

3.1.2 Force Control

Consider the force control problem for a kinematic manipulator equipped with elastic deformation torque sensors for each joint. It is assumed that the control objective is to regulate the contact force $\hat{f}$, obtained indirectly through the procedure described in Section 2.3.1, so that it reaches a desired force f_d toward the surface contact, that is,

$$\hat{f} \to f_d, \quad e_f = f_d - \hat{f} \to 0, \tag{3.11}$$

where e_f is the force error. Considering the object elasticity, the force at the end effector f can be modeled by Hooke's law as:

$$f = k_s(p - p_s^0), \tag{3.12}$$

where p is the position of the contact point, p_s^0 is the initial position of the object surface (without contact) and k_s is the elasticity constant for the object. Since $\hat{f}$ is a estimative of f, from (3.11) and (3.12) the error equation is given by

$$\dot{e}_f = -k_s\, \dot{p} \ . \tag{3.13}$$

Considering that the force control acts on the position variables ($v_f = v_p$), then from (3.4) comes that $\dot{p} = v_f$. By using a proportional control law:

$$v_f = k_f \, e_f, \tag{3.14}$$

where k_f is the force control gain and the force error dynamics is given by $\dot{e}_f + k_s \, k_f \, e_f = 0$. By choosing $k_f > 0$ implies that $\lim_{t \to \infty} e_f(t) = 0$.

3.1.3 Orientation Control

Consider the orientation control problem for a kinematic manipulator. The control objective is to drive the current end-effector orientation matrix $R \in SO(3)$ to a desired orientation $R^d(t) \in SO(3)$:

$$R \to R^d(t), \quad R_r = R^d(t) \, R^{\mathsf{T}} \to I, \tag{3.15}$$

where $R_r \in SO(3)$ is the relative orientation matrix, that is, the orientation error expressed at inertial frame $\mathcal{F}_b$. Let $e_q = [e_{qs} \quad e_{qv}^{\mathsf{T}}]^{\mathsf{T}}$ be the unit quaternion representation for matrix R_r, such that

$$e_q = q^d(t) * q^{-1} \ , \tag{3.16}$$

where $q = [q_s \quad q_v^{\mathsf{T}}]^{\mathsf{T}}$ and $q^d(t) = [q_s^d \quad q_v^{d\,\mathsf{T}}]^{\mathsf{T}}$ are the unit quaternion representation for R and R^d, respectively. Note that $e_q = [1 \quad 0^{\mathsf{T}}]^{\mathsf{T}}$ if and only if R and R_d are aligned. Therefore, it is sufficient to define the orientation error $e_o \in \mathbb{R}^3$ as (Siciliano et al., 2009):

$$e_o = e_{qv} = q_s \, q_v^d - q_s^d \, q_v - \mathcal{S}(q_v^d) \, q_v \ . \tag{3.17}$$

From (3.4) we have that $\omega = v_o$. Thus, using a feed-forward plus proportional control law

$$v_o = \omega_d + K_o \, e_o, \tag{3.18}$$

where ω_d is the desired angular velocity and $K_o \in \mathbb{R}^{3\times3}$ a positive definite gain matrix. The orientation error dynamics is expressed as

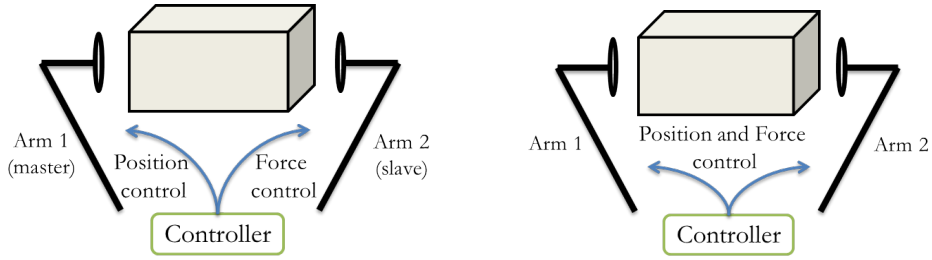
$$\tilde{\omega} + K_o \, e_o = 0 \ , \tag{3.19}$$

where $\tilde{\omega} = (\omega_d - \omega) \in \mathbb{R}^3$ is the angular velocity error of the end effector. This is a *non-linear* relation, since it contains the angular velocity error instead time-derivative of the orientation error. By using the Lyapunov stability theory, it is possible to prove that the equilibrium point $[e_{qs} \quad e_{qv}^{\mathsf{T}}]^{\mathsf{T}} = [\pm 1 \quad 0^{\mathsf{T}}]^{\mathsf{T}}$ is almost global asymptotically stable (Wen and Kreutz-Delgado, 1991; Slotine et al., 1991).

3.2 Cooperative Control

When dealing with the problem of a cooperative multi-arm system manipulating an object, it is important to model and control the absolute and relative force and motion of the arms. Other approaches for controlling force and motion were extensively studied, however suffered from theoretical or implementations problems.

For instance, the early methods were based on the *master/slave* concept. In this method, the master arm is position controlled, aiming to achieve accurate and robust tracking of position/orientation reference trajectories, and the slave arm is force controlled, designed to achieve a compliant behavior and to accept the master arm movements. As it can be expected, it is difficult to design a stable force control with fast response to the position disturbance of the master arm (Caccavale and Uchiyama, 2008).



(a) master/slave scheme. (b) non-master/slave schemes.

A posterior approach known as the *leader-follower* (Luh and Zheng, 1987) (or *coordinated control*) assigns a robot as the leader, which has independent movement, and the other robots as followers with reference motion computed via closed-chain constraints. This method also suffer from implementation issues, since the compliance of the follower arms has to be very large; the decision of leader arm and possible switch to other is also a concern.

As a consequence, more natural approaches have arisen later, considering the cooperative system as a whole and the reference motion of the object as well as the reference force applied to it as the key parameters to control the motion of all arms in the system. In this context, the symmetric formulation was an important non-master slave scheme proposed, but it is object-dependent. Meanwhile, the cooperative task-space formulation (Chiacchio et al., 1996) is a more elegant formulation, which allows straightforward control of the arms in terms of the absolute and relative variables. Therefore, we gave special attention to this method in this section, without considering force feedback yet. Later in this chapter, this formulation is extended to include force control, using the hybrid position-force method.

3.2.1 Differential Kinematics for Coordinated Motion Tasks

In Section 2.4.2, the task-space formulation was discussed. Now, the differential kinematics problem for a two-arm robotic system is treated: the objective is to compute the joint trajectories corresponding to a desired coordinated motion defined in terms of task space variables.

Consider first the case of two cooperative 6-dof arms. For each manipulator, the differential kinematic relation can be represented as:

$$\begin{bmatrix} \dot{p}_i \\ \omega_i \end{bmatrix} = J_i(\theta_i)\,\dot{\theta}_i \qquad i = 1, 2. \tag{3.20}$$

where $\theta_i \in \mathbb{R}^6$ represents the joint variables and $J_i(\theta_i) \in \mathbb{R}^{6\times 6}$ the Jacobian matrix for the i-th manipulator. The cooperative task-space formulation describes the cooperative problem in terms of absolute and relative variables. Following this formulation, a more compact and elegant way to define the relationship between joint space and task space velocities is to combine the manipulators Jacobians yielding the cooperative Jacobian (Chiacchio et al., 1996). Using the absolute task velocities definition (2.46) and (2.47), considering (3.20), yields that the absolute velocity can be obtained from the joint velocities:

$$\begin{bmatrix} \dot{p}_a \\ \omega_a \end{bmatrix} = J_a(\theta_1, \theta_2) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} , \tag{3.21}$$

where $J_a \in \mathbb{R}^{6\times 12}$ is the *absolute Jacobian*. It is straightforward to see that J_a is written in terms of each manipulator Jacobian as:

$$J_a = \begin{bmatrix} \frac{1}{2}J_1 & \frac{1}{2}J_2 \end{bmatrix} . \tag{3.22}$$

Similarly, the relative task velocities are given by:

$$\begin{bmatrix} \dot{p}_r \\ \omega_r \end{bmatrix} = J_r(\theta_1, \theta_2) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} , \tag{3.23}$$

where $J_r \in \mathbb{R}^{6\times 12}$ is the *relative Jacobian*. From (2.48) and (2.49), it is clear that:

$$J_r = \begin{bmatrix} -J_1 & J_2 \end{bmatrix} . \tag{3.24}$$

The kinematic control can be applied to both arms, so the movement of each

manipulator is described as:

$$\dot{\theta}_{1,\,i} \;=\; u_{1,\,i}\;, \quad i = 1, \cdots, n_1\;, \tag{3.25}$$

$$\dot{\theta}_{2,\,i} \;=\; u_{2,\,i}\;, \quad i = 1, \cdots, n_2\;, \tag{3.26}$$

where $\dot{\theta}_{1,\,i}$ and $\dot{\theta}_{2,\,i}$ is the i-th joint velocity of manipulator 1 and 2, respectively, and $u_{1,\,i}$ and $u_{2,\,i}$ is the control signal for velocity applied at the i-th joint motor drive of manipulator 1 and 2, respectively. Then, using $u = [u_1 \quad u_2]^{\mathsf{T}}$ as the stacked control signal, the cooperative control system is summarized as:

$$\begin{bmatrix} \dot{p}_a \\ w_a \\ \dot{p}_r \\ w_r \end{bmatrix} = \underbrace{\begin{bmatrix} J_a \\ J_r \end{bmatrix}}_{J} u\;, \tag{3.27}$$

where $J = [J_a \quad J_r]^{\mathsf{T}} \in \mathbb{R}^{12\times 12}$ is called the *cooperative Jacobian*. In order to linearize the system, an algorithmic inverse kinematics solution is constructed based in the inverse of the cooperative Jacobian:

$$u = J^{-1}(\theta_1, \theta_2)\, v_k\;, \tag{3.28}$$

where v_k is the task-space (*Cartesian*) control signal to be designed. It is assumed that v_k does not lead the manipulator to singular configurations.

Then, the Cartesian signal is design to control the absolute and relative position and orientation of the system. Since tracking of a reference trajectory is desired, an suitable control is a *proportional* plus a *feed-forward* action:

$$v_k = \begin{bmatrix} v_{pa} \\ v_{oa} \\ v_{pr} \\ v_{or} \end{bmatrix} = \begin{bmatrix} K_{pa} & & & \\ & K_{oa} & & \\ & & K_{pr} & \\ & & & K_{or} \end{bmatrix} \begin{bmatrix} e_{pa} \\ e_{oa} \\ e_{pr} \\ e_{or} \end{bmatrix} + v_d\;, \tag{3.29}$$

where v_{pa} and $v_{oa} \in \mathbb{R}^3$ are the absolute position and orientation control signals and v_{pr} and $v_{or} \in \mathbb{R}^3$ are the relative position and orientation control signals; $K_{pa}, K_{oa}, K_{pr}, K_{or} \in \mathbb{R}^{3\times 3}$ are the respective diagonal gain matrices and v_d is the feed-forward signal. Besides, the absolute position and orientation errors are given by:

$$e_{pa} \;=\; p_a^d - p_a\;, \tag{3.30}$$

$$e_{oa} \;=\; q_{a,\,s}\, q_{a,\,v}^d - q_{a,\,s}^d\, q_{a,\,v} + \mathcal{S}(q_{a,\,v}^d)\, q_{a,\,v}\;, \tag{3.31}$$

where $p_a^d \in \mathbb{R}^3$ is the desired absolute position specified in the base frame and $p_a \in \mathbb{R}^3$ is computed as in (2.8); $q_{a,\,s}$ and $q_{a,\,v}$ are the scalar and vectorial components of the quaternion for absolute orientation and $q_{a,\,s}^d$ and $q_{a,\,v}^d$ are the scalar and vectorial components, respectively of the quaternion representing the desired absolute orientation. Further, the relative position and orientation errors (e_{pr} and $e_{or} \in \mathbb{R}^3$) are given by:

$$e_{pr} = R_{ba}\,{}^a p_r^{\,d} - p_r \tag{3.32}$$

$$e_{or} = R_{b1}\left\{q_{r,\,s}\,q_{r,\,v}^d - q_{r,\,s}^d\,q_{r,\,v} + \mathcal{S}(q_{r,\,v}^d)\,q_{r,\,v}\right\}\,. \tag{3.33}$$

The desired relative position $p_r^d \in \mathbb{R}^3$ is usually expressed at the absolute frame $\mathcal{F}_a$, since in this manner the specification between the end effectors will not dependent on the orientation of $\mathcal{F}_a$. Thus, the rotation R_{ba} is used to transform ${}^a p_r^d$ to the base frame. For the same reason, the desired relative orientation q_r^d and the relative orientation error e_{or} are calculated with respect to frame $\mathcal{F}_1$, then the rotation R_{b1} expresses e_{or} at the base frame.

Furthermore, the feed-forward action is based on the derivative of the desired trajectories:

$$v_d = \begin{bmatrix} \dot{p}_a^d \\ \omega_a^d \\ R_{ba}({}^a\dot{p}_r^{\,d}) + \mathcal{S}(\omega_{ba}) R_{ba}\,{}^a p_r^{\,d} \\ R_{b1}\,{}^1\omega_r^{\,d} \end{bmatrix} \tag{3.34}$$

where ω_a^d is the desired absolute angular velocity and ${}^1\omega_r^{\,d}$ is the desired relative angular velocity specified directly in frame $\mathcal{F}_1$. Remarkably, manipulation of a common object in general requires as part of the task to keep ${}^a p_r$ and ${}^1 q_r$ constant, that is, a tight grasp. In this case, the desired motion is assigned with ${}^a\dot{p}_r^{\,d} = 0$ and ${}^1\omega_r^{\,d} = 0$.

Finally, notice that from (3.27) and (3.28), the linearized closed-loop system is given by:

$$\begin{bmatrix} \dot{p}_a \\ w_a \\ \dot{p}_r \\ w_r \end{bmatrix} = \begin{bmatrix} v_{pa} \\ v_{oa} \\ v_{pr} \\ v_{or} \end{bmatrix}\,. \tag{3.35}$$

Coordinated Motion Control using Redundant Manipulators

Now, assume that the cooperative system is composed by two 7-Dof manipulators. Since a full coordinated motion in 3D space requires 6-Dof for absolute variables plus 6-Dof for relative variables, the cooperative system (composed by 14-Dof) has 2 extra degrees of freedom.

Furthermore, the presence of redundant degrees of freedom allows to modify (3.5) by adding a term of projection into the null space of J:

$$u = J^{\dagger}\, v_k + (I - J^{\dagger} J)\, \mu \ , \tag{3.36}$$

where $\mu \in \mathbb{R}^{14}$ is an arbitrary joint velocity vector. Note that matrix $(I - J^{\dagger}J)$ projects vector μ into the null space of J. As a consequence, μ is not able to change the end-effectors linear and angular velocities, but can set the joints to other positions.

In order to take advantage of this property, μ can be specified to attend an additional constraint for the problem, as a second objective. For instance it can be used to avoid singularities, to prevent collisions with obstacles or to avoid joint limits. A typical choice for μ is:

$$\mu = k_c \left(\frac{\partial w(\theta)}{\partial \theta} \right)^{\mathsf{T}} , \tag{3.37}$$

where $k_c > 0$ is a constant and $w(\theta)$ is the secondary objective function. The solution moves along the gradient direction of function $w(\theta)$ and tries to locally *maximize* it with respect to the primary objective. When it is desired to avoid joint limit, a typical objective function is defined as:

$$w(\theta) = -\frac{1}{2n} \sum_{i=1}^{n} \left(\frac{\theta_i - \bar{\theta}_i}{\theta_{iM} - \theta_{im}} \right)^2 \tag{3.38}$$

where the range of each joint is between the maximum limit θ_{iM} and the minimum limit θ_{im}, and $\bar{\theta}_i$ is the centre of this range. Note that, in this case, it is desired to *minimize* the distance of each joint to the centre of its range, so the objective function has a minus sign in front of it.

Therefore, inverse kinematics solution (3.36) locally minimizes the norm of joint velocities and allows to generate internal movement to reconfigure the manipulators structure without affecting the end-effectors pose. This solution is used instead of (3.28) for the case of redundant manipulators.

3.2.2 Numerical simulation

In order to illustrate the theoretical performance of the cooperative coordination control scheme, the results obtained in simulation are presented. The simulation is implemented in Simulink, with the support of Robotic Toolbox, using the two-arm model with similar mechanical of the real robot used in the experiments (Baxter). The simulation parameters of control are shown in Table 3.1.

Table 3.1: Simulation parameters for cooperative coordination control.

Parameter	Value
K_{pa}	2 s^{-1}
K_{pr}	15 s^{-1}
K_{oa}	10 $rad\,s^{-1}$
K_{or}	20 $rad\,s^{-1}$

In the simulation, the absolute position control objective is to track a circular trajectory in plane XZ of the robot base frame: $x = 0.6$; $y = 0.0 + 0.2\sin(1/20\,t)$ and $z = 0.4 + 0.2\cos(1/20\,t)$.

Figure 3.3 (a) describes the absolute position and reference trajectories. The trajectory is smooth and it tracks the reference. Figure 3.3 (b) illustrates the absolute position error. After the transient, the absolute position error converges asymptotically to zero.

In order to test the relative position control, the relative reference for direction z_e is to track the sine function $z = 0.22 + 0.1\sin(1/10\,t)$ and to regulate at zero in the other directions, as shown in Figure 3.4 (a). The error, shown in Figure 3.4 (b), is acceptable and oscillates around 8 mm.

The absolute and relative orientation errors are illustrated in Figure 3.5. For a proper manipulation, priority is given for the relative orientation error, since it is more important to avoid lose of alignment. The absolute orientation error presents a small and acceptable oscillation around 0.03 rad. Meanwhile, the relative orientation error is kept close to zero.

The Cartesian control signal is demonstrated in Figure 3.6. It can be noted that the control signals are smooth and bounded. The joint velocities are also smooth and do not require any peak or unfeasible velocities from the actuators, as can be seen in Figure 3.7 and Figure 3.8.

The representation of the robot arms and the trajectory performed by the object (frame $\mathcal{F}_a$) is shown in Figure 3.9. It can be noted that, after the transient, the trajectory tracks the reference with reasonable performance.

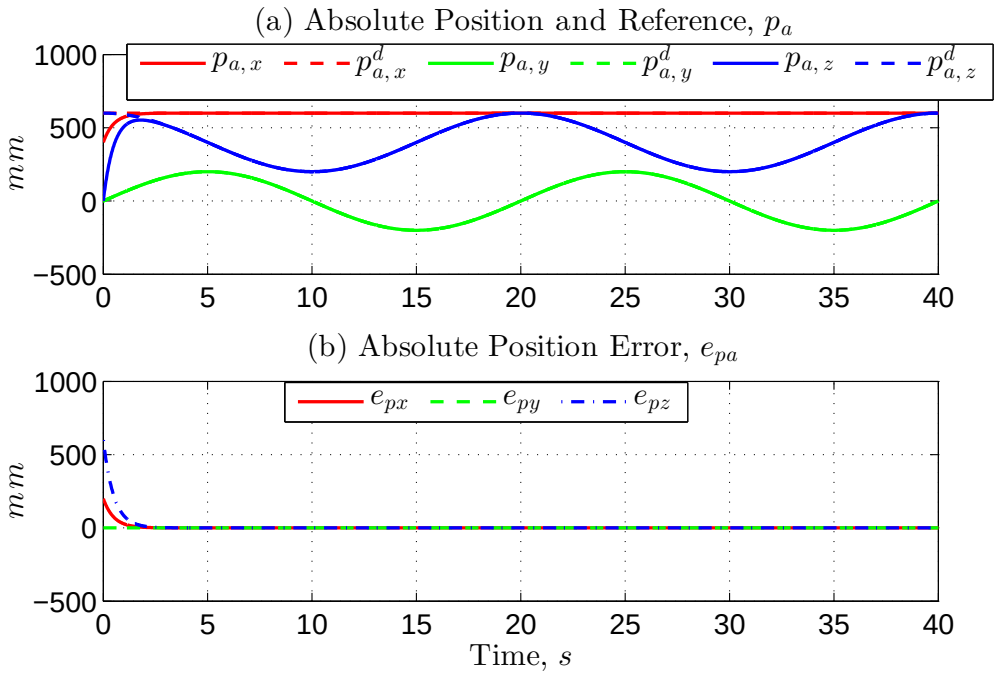


Figure 3.3: Coordinated motion: (a) absolute position and reference (b) absolute position error.

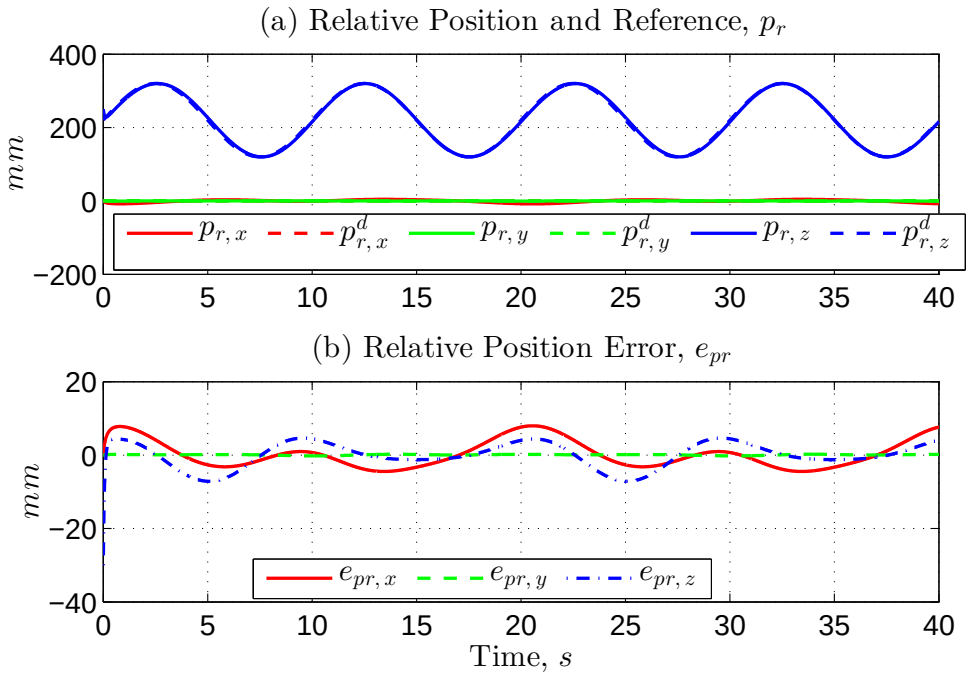


Figure 3.4: Coordinated motion: (a) relative position and reference (b) relative position error.

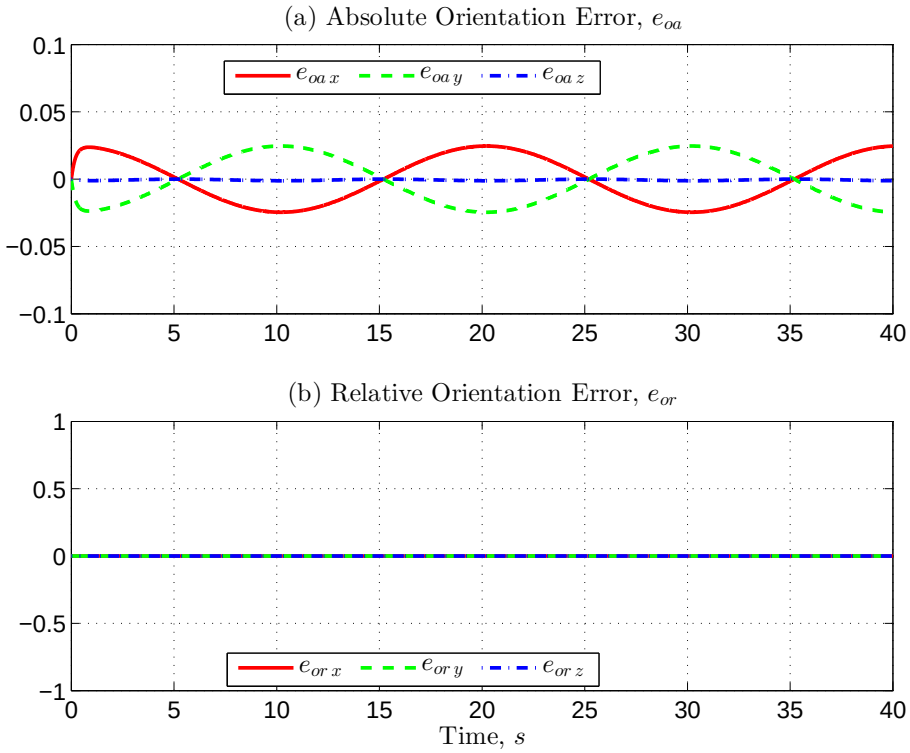


Figure 3.5: Coordinated motion: (a) absolute orientation error (b) relative orientation error.

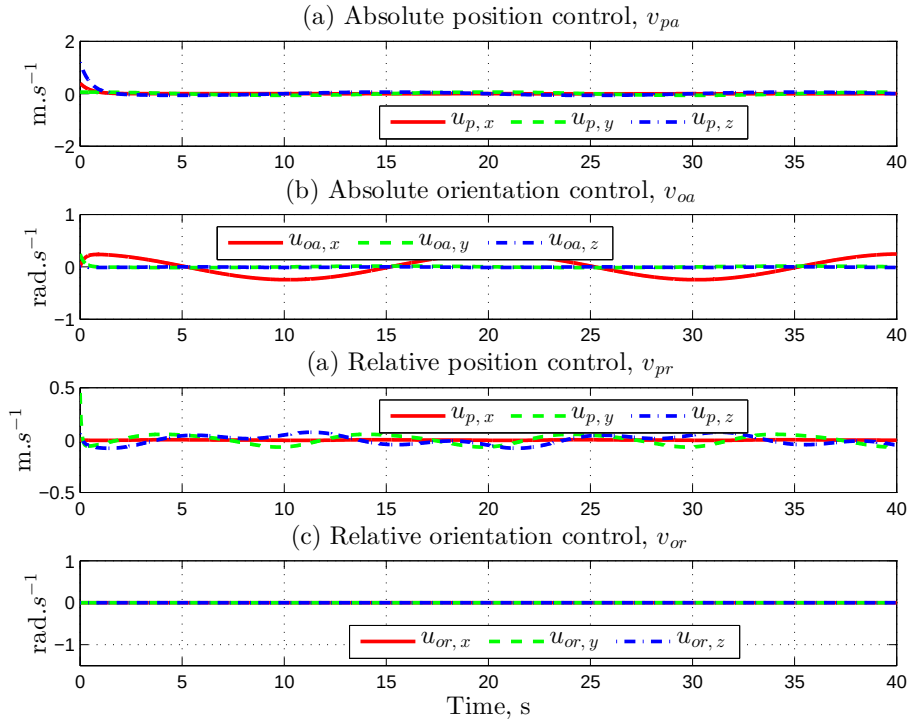


Figure 3.6: Coordinated motion: (a) absolute position control (b) relative orientation control (c) relative position control (d) relative orientation control.

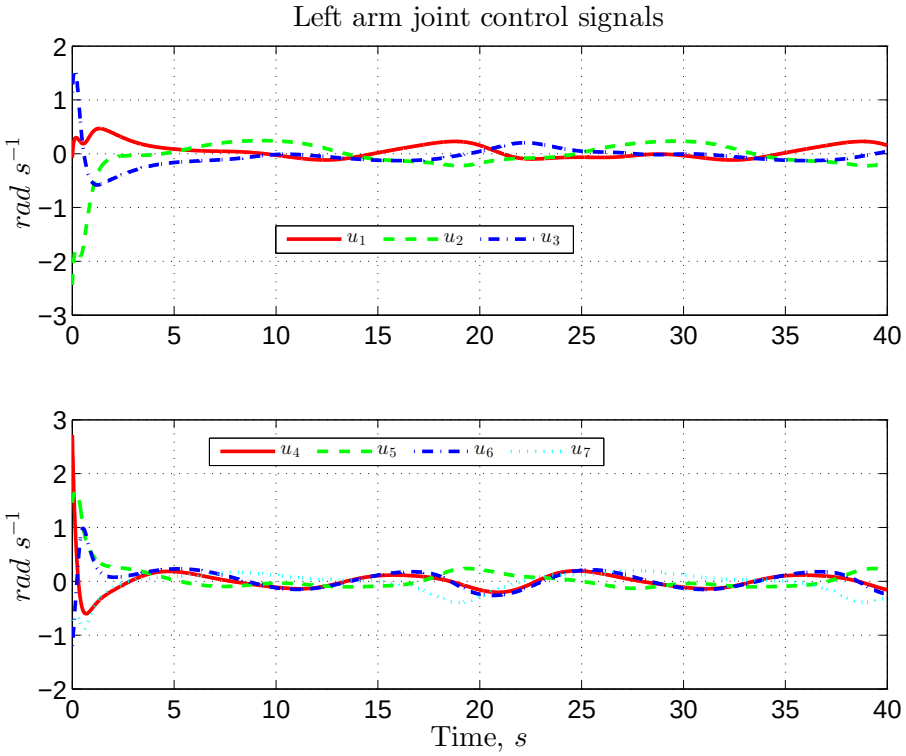


Figure 3.7: Coordinated motion: left arm joint control signals.

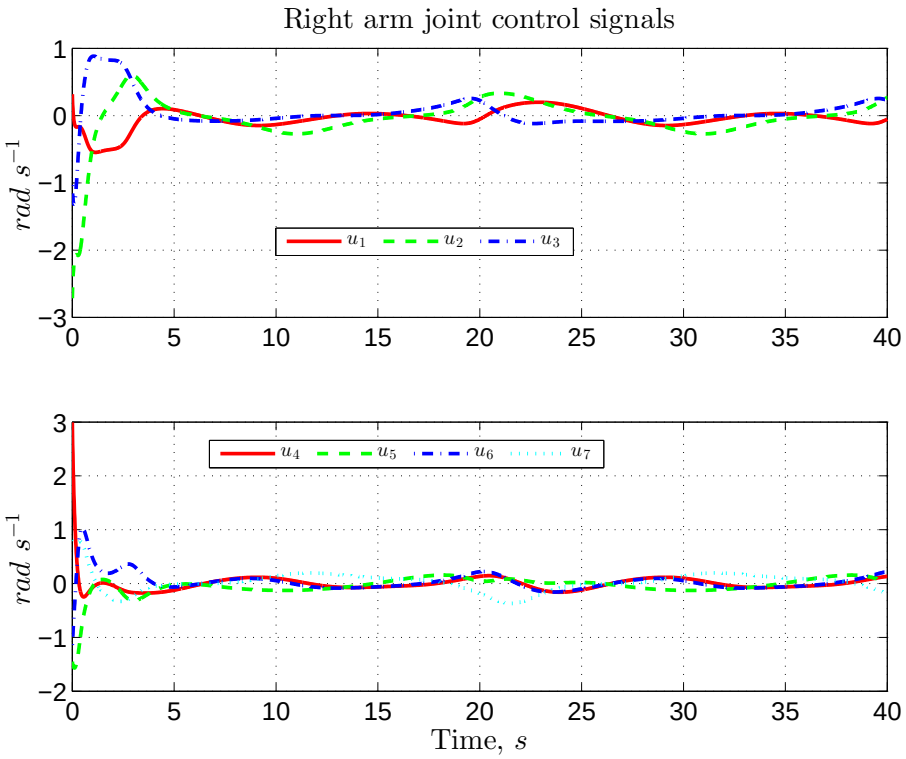


Figure 3.8: Coordinated motion: right arm joint control signals.

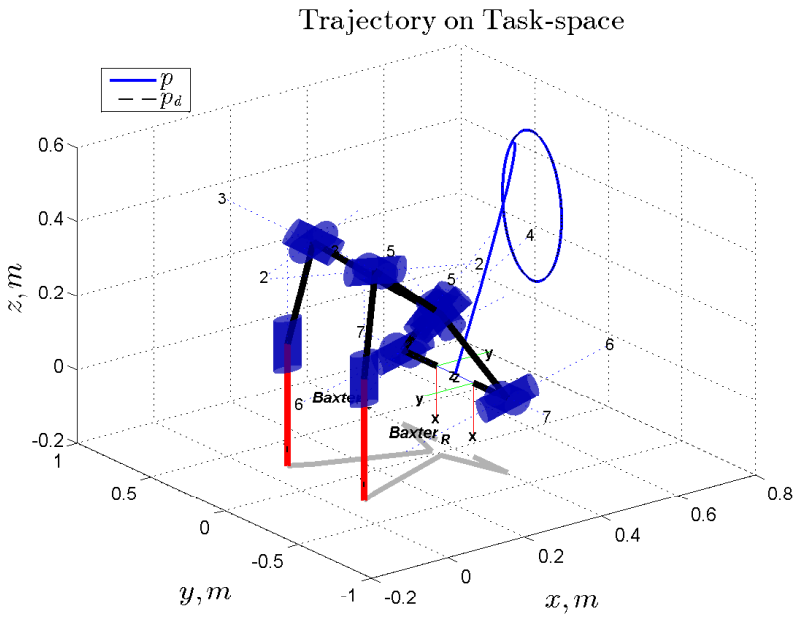


Figure 3.9: Coordinated motion: trajectory performed by manipulators (with reference).

3.3 Cooperative Control using Force Feedback

In the previous section, cooperative control was considered for coordination of the arms to achieve desired absolute and relative configurations, considering the task definition described in absolute and relative frames. Using this approach it would be possible to grasp an object of known dimensions. However, pure motion control requires accurate model of manipulator and environment, since any small error in the positioning could cause huge rise of force applied by the manipulators and consequently, unstable behavior. In this context, the use of force feedback can solve these problems, allowing the cooperative system to achieve more robust behavior when operating in unstructured environments and manipulating unknown objects.

This section presents the aspects related to force interaction between manipulators and object. First, the object, assumed to have flexibility, is modeled considering its elasticity and Hooke's law is used to determine the forces. Although the cooperative task formulation used is not dependent on any assumption of the object, the modeling described here is used in the simulation and is also useful for better physical comprehension of the grasp. Next, the contact interaction is described according to friction models.

Natural and Artificial Constraints, which are essential to understand the subspace decomposition in force and position directions, are also explained and a discussion on the interaction methods is performed, focusing on the hybrid control, a method that allows force specification when performing the manipulation task. Finally, a proposed cooperative kinematic scheme is developed, explained and simulated.

3.3.1 Object Modeling

Consider the simplified mass-spring model of an object with mass $m_o \in \mathbb{N}$, width $l_o \in \mathbb{N}$ and lateral surfaces subject to elastic deformation with constant $k_s \in \mathbb{N}$, as represented in figure 3.10. If the grasp is performed correctly, it can be assumed that the end effectors apply forces normally at the surface contact points $\mathcal{F}_{m,1}$ and $\mathcal{F}_{m,2}$.

Let $p_o \in \mathbb{R}^3$ be the position of the object, expressed at the base frame $\mathcal{F}_b$. For simplicity, we assume the forces act in the direction $\vec{z}_o$, then the forces exerted by

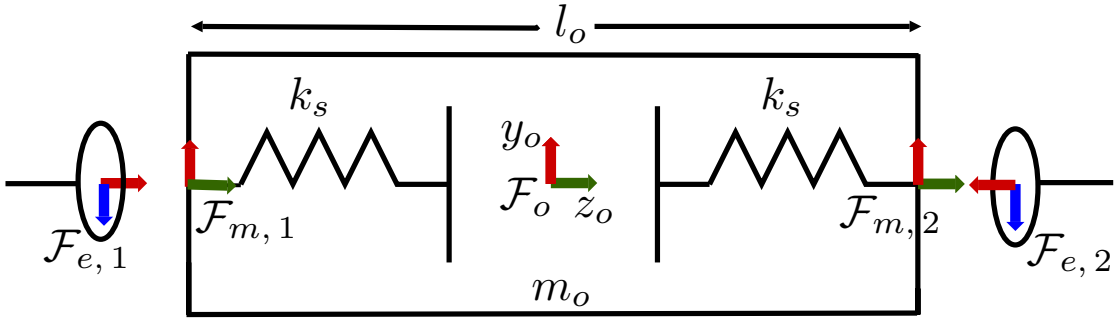


Figure 3.10: Simplified object model.

the end effectors are modeled as:

$$f_1 = \begin{cases} -k_s\left(p_o - p_1 - \frac{l_0}{2}\right), & p_o - p_1 < \frac{l_0}{2}, \\ 0, & p_o - p_1 \geq \frac{l_0}{2}, \end{cases} \tag{3.39}$$

$$f_2 = \begin{cases} k_s\left(p_2 - p_o - \frac{l_0}{2}\right), & p_2 - p_o < \frac{l_0}{2}, \\ 0, & p_2 - p_o \geq \frac{l_0}{2}. \end{cases} \tag{3.40}$$

Using this model, the relative force $f_r \in \mathbb{R}^3$ (internal force) imposed to the object can be described according to relation (2.52) as:

$$f_r = \frac{k_s}{2}\left(p_2 - p_1 - l_0\right). \tag{3.41}$$

Finally, assuming that the grasp is tight and is maintained during the manipulation task, the motion for this system in the grasp direction is given by the relation:

$$m_o\,\ddot{p}_o = f_1 + f_2 = k_s\left(p_1 + p_2 - 2\,p_o\right). \tag{3.42}$$

3.3.2 Contact Modeling

When an end effector is interacting with some surface, it is important to describe the contact model relating the forces exerted in the contact point and the total forces and moments arisen by this interaction.

For rigid-bodies, the simplest model is the *point contact without friction*. Since there is no friction, the forces can only be applied in the normal direction to the surface plane. Thus, the generalized forces exerted by the end effector can be represented as $(F)_e = [0\ 0\ f_c\ 0\ 0\ 0]^\mathsf{T}$, where $f_c \in \mathbb{R}$ is the force magnitude applied in the normal plane.

However, a more suitable model should be used in tasks that depend on the friction forces, as the grasping of a rigid object. The simplest analytical model for

point contact with friction is the *Coulomb friction model*. This experimental law establishes that the friction force magnitude in the tangent plane at the contact interface is proportional to the normal force magnitude.

Let $f_t \in \mathbb{R}$ be the friction force magnitude in the tangent plane and $f_n \in \mathbb{R}$ be the normal force magnitude. Then, the Coulomb law states that:

$$f_t \leq \mu_s \, f_n \tag{3.43}$$

where $\mu_s > 0$ is called the *static friction coefficient*. The friction coefficient depends on the two materials in contact (see table (3.2)).

Table 3.2: Static friction coefficient for some materials.

Materials	μ_s
Rubber-Solids	$1 - 4$
Rubber-Cardboard	$0.5 - 0.8$
Steel-Steel	0.78

In the limit of static contact $f_{t\,max} = \mu_s \, f_n$, that is, the maximum tangent force magnitude and it occurs just before slippage. If this limit is exceed, the surfaces in contact start to slide, and the friction is described by:

$$f_t = \mu_k \, f_n \ , \tag{3.44}$$

where μ_k is called the *kinetic friction coefficient* and is smaller than μ_s ($\mu_k \leq \mu_s$). Note that the friction force opposes the direction of motion and is independent of the speed of sliding.

Moreover, the relation (3.43) can be interpreted in terms of a friction cone. The set of forces that can be applied to a contact must remains in the cone, which is centered in the surface normal. The borders of this cone represents the maximum resultant force before slippage occurs (see Figure (3.11)). The angle formed by the cone and the normal is

$$\alpha_s = atan(\mu_s) \ , \tag{3.45}$$

where α_s represents the maximum slope of the end effector with respect to the surface, before it loses contact. For the spatial case, the friction cone forms a circular cone, defined by:

$$\sqrt{f_x^2 + f_y^2} \ \leq \ \mu_s \, f_z \ , \tag{3.46}$$

where $f = [f_x \ f_y \ f_z] \in \mathbb{R}^3$ is the force applied by the end effector and it is assumed that $f_z \geq 0$ is applied in the normal direction at the contact surface.

In the case of grasping using multiple manipulators, a common object is subject to multiple contacts. When the end effectors hold a part, the contact forces should

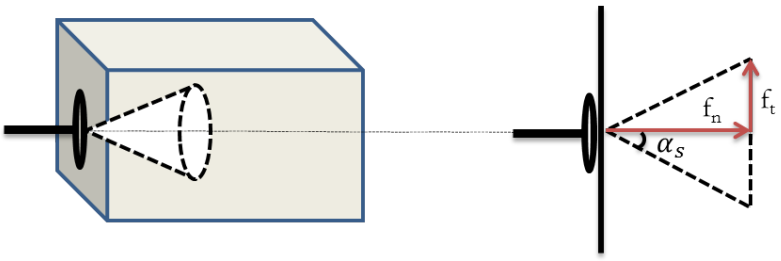


Figure 3.11: Representation of an end effector in contact with a box and the corresponding friction cone.

compensate the weight of the object. For simplicity, consider in the planar case an object, with mass equally distributed along the horizontal plane, being held by two manipulators. In this case, the equilibria is reached when:

$$m_o\, g = 2\, f_t = 2\, \mu_s\, f_n \ , \tag{3.47}$$

considering that equal forces are applied by the end effectors. Thus, the minimum force applied to guarantee no-slippage (static contact) is given by:

$$|\, f_a \,|_{min} \geq \frac{m_o\, g}{\mu_s} \ , \tag{3.48}$$

where f_a is the absolute force, as defined in (2.50).

3.3.3 Natural and Artificial Constraints

During a manipulation task, the end effector is subject to complex contact interaction with the environment, where some directions (DoF) impose force constraints - meaning that it is not possible to apply any force along a direction or any torque about an axis - and other directions impose motion constraints (position or velocity) - which means the environment does not allow translation along a direction or rotation about an axis. Such constraints are intrinsically imposed by the geometry of the task and are called *natural constraints*.

For a a rigid environment, it can be noted that along each DoF the manipulator can control only the variables that are not subject to natural constraints. Depending on the task, the control strategy may impose reference values for these variables, which are called *artificial constraints*.

Considering a compliant environment, each DoF subject to compliance behavior allows to choose the variable (force or position) suitable to be an artificial constraint, while the other variable is considered as a natural constraint (Siciliano et al., 2009). For example, if a manipulator is pushing a spring, the controller can use as reference the position of the end effector (as long a model is available) or the force applied by

it.

In order to accomplish the task goal, artificial constraints can be specified in an suitable coordinate system, eventually time-varying, named constraint frame $\mathcal{F}_S$, which is conveniently placed according to the task geometry. For instance, consider as a manipulation task the sliding of an object on a frictionless surface. The constraint frame may be attached to the surface with an axis orthogonal to the surface plane (Figure 3.12).

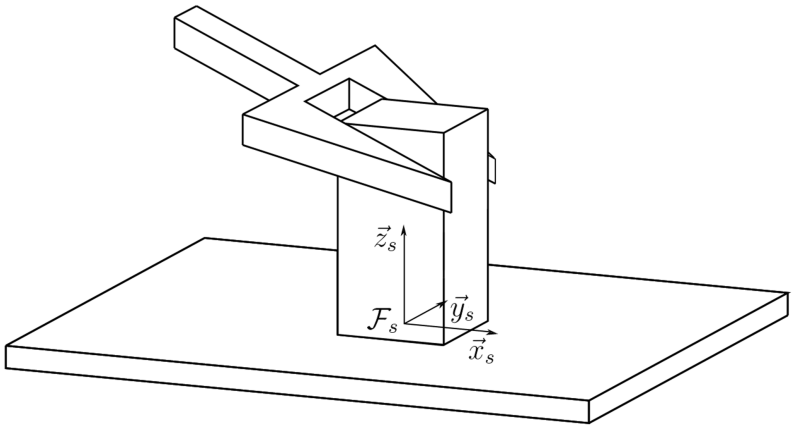


Figure 3.12: Task of manipulating an object in contact with a surface. Extracted from (Siciliano et al., 2009).

The natural constraints are imposed by the geometry of the task. In this task, the natural motion constraint reflect the impossibility to move the manipulator in z_s direction and to generate angular velocity along x_s and y_s directions. Meanwhile, the force constraints reflect the impossibility to exert forces along x_s and y_s axis or apply moment around axis z_s. As it can be noted, the two sets of constraints are complementary. An important result is that it is not possible to define position and force objectives in the same direction simultaneously (Siciliano et al., 2009).

The variables which are not subject to natural constraints, may be used to specify *artificial constraints*. In the given example, it is possible to set constraints for linear velocity along axes x_s and y_s and angular velocity along axis z_s. Likely, artificial constraints for force along z_s and moments about x_s and y_s may be specified. The set of constraints is represented in Table 3.3.

3.3.4 Force Control and Interaction Methods

One important aspect to consider when dealing with the robot-environment interaction is to solve how the robot feedback signals from force and position should be used to achieve the task objectives and requirements. The manner in which the measurement signals are used by the controller and the choice of command input signals result in different force control methods (Zeng and Hemami, 1997).

Table 3.3: Natural and artificial constraints for the example task.

Natural Constraints	Artificial Constraints
v_z	v_x
ω_x	v_y
ω_y	ω_z
f_x	f_z
f_y	μ_x
μ_z	μ_y

The classification of control algorithms is divided in two groups: indirect and direct force control methods (see Figure 3.13). Notice that the master/slave scheme is a method for coordination control of two arms and it is not classified as a force interaction method.

The first group performs force control through motion control, without using an explicit force control loop. The force control actuates as a master controller, using virtual stiffness relations to modify the reference for the position control loop, which is responsible for the final actuation. This group mainly includes Stiffness Control (Salisbury, 1980) and Impedance Control (Hogan, 1984).

On the other hand, for tasks requiring accurate control of the interaction forces exerted by the end effector, it is important to employ a strategy that allows direct specification of a desired force value. The second group uses a separate (explicit) force control loop to achieve a successful interaction control. The signals from force and position controllers are combined after, according to the specific method. Therefore, reference values can be set for force and for position objectives, separately. The main methods belonging to this category are: Hybrid Control (Raibert and Craig, 1981) and Parallel Control (Chiaverini and Sciavicco, 1993; Caccavale et al., 2013). In the following, a review of the main force interaction methods is presented (Zeng and Hemami, 1997).

Stiffness Control

Considering a single robotic arm in contact with the environment, as represented in Figure 3.14, the use of a mechanical device in the end effector usually allows better force control, since it can smooth the interaction and it can avoid high peak forces. In this context, the main idea of stiffness control, proposed by (Salisbury, 1980), is to emulate the mechanical device effects, adjusting the stiffness behavior of the robotic system as a programmable spring.

Figure 3.15 presents the diagram for Stiffness Control. Consider the position and orientation vector $x \in \mathbb{R}^m$, the force and moment vector $F \in \mathbb{R}^m$. Vector $x_s \in \mathbb{R}^m$ represents the pose of contacted environment and matrix $K_s \in \mathbb{R}^{m \times m}$ is

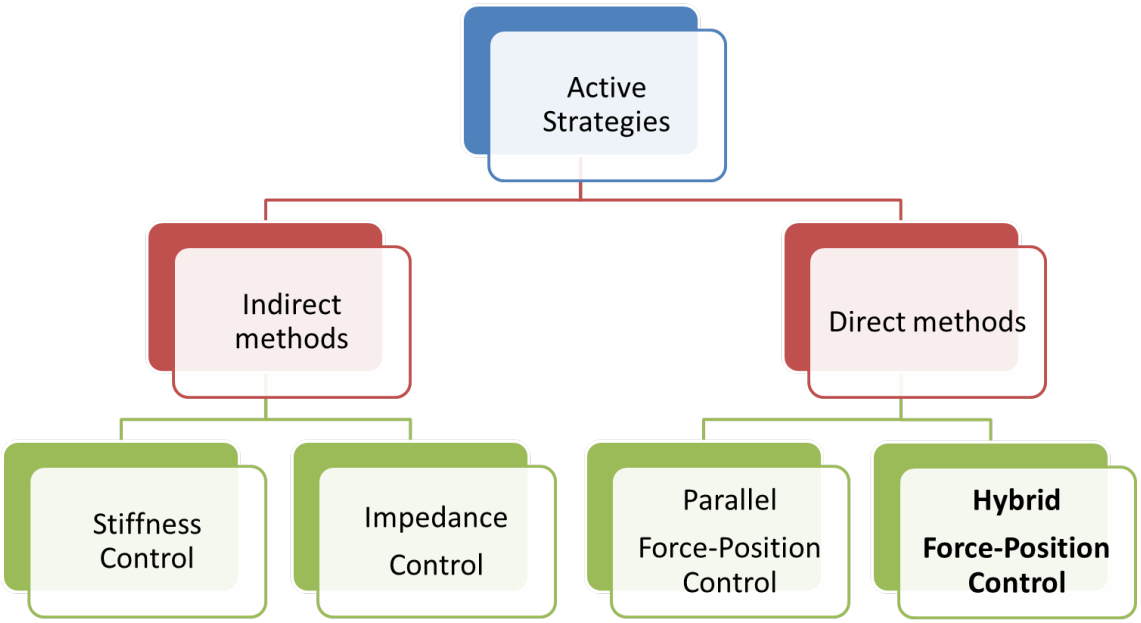


Figure 3.13: Organization of force control methods.

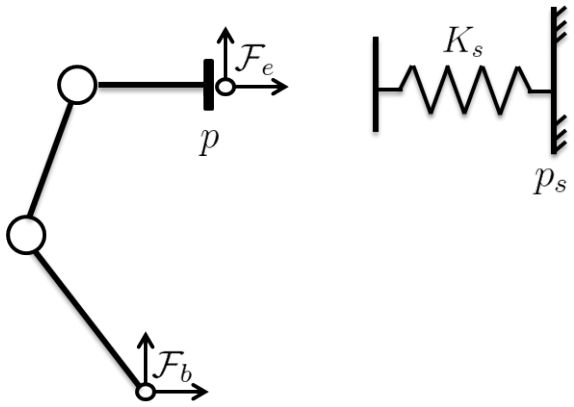


Figure 3.14: Representation of the interaction task considering an elastic surface.

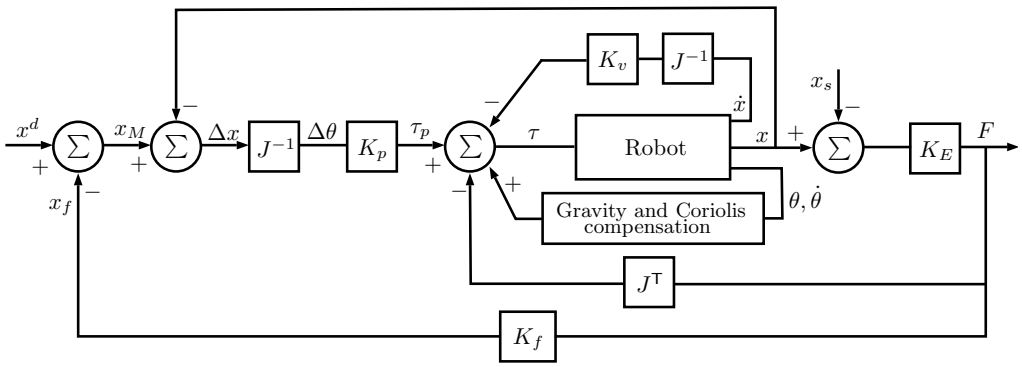


Figure 3.15: Stiffness Control.

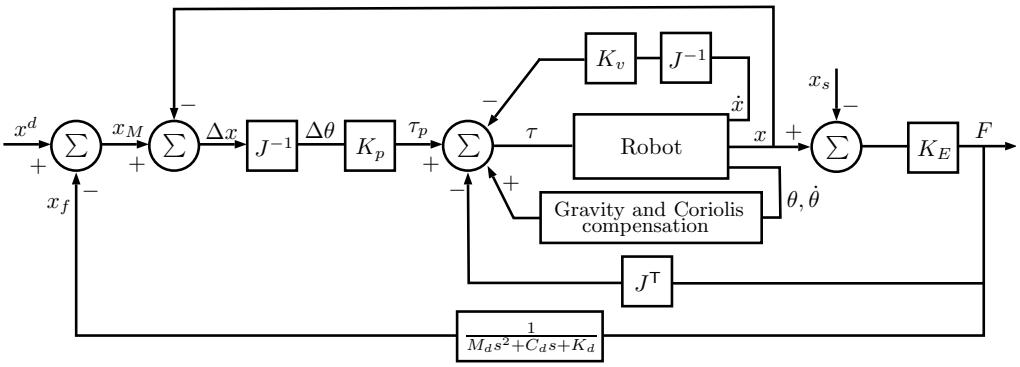


Figure 3.16: Impedance Control.

the elasticity of the environment and it can also include the elasticity of the end-effector tool. Notice that the torques generated by the contact with the environment $J^\intercal F$ and the effects of gravity and Coriolis forces are compensated.

The force vector F is used to generate an output vector x_f, which modifies the desired position vector in the task space x^d, leading to the modified desired position vector x_M. Then, the position error Δx is mapped to the joint angles deviation $\Delta\theta$ and the control signal $\tau_p \in \mathbb{R}^n$ is calculated according to the relation:

$$\tau_p = K_p \, \Delta\theta \tag{3.49}$$

where K_p is called the *stiffness matrix*. Notice that it is possible to adjust the mechanical stiffness of the manipulator by modifying the matrix K_p, which relates force and displacement.

Impedance Control

While stiffness control designs a controller to emulate a desired static relationship between force and displacements, impedance control, proposed by (Hogan, 1984), aims to achieve a desired dynamic behavior for the robotic system during interaction. The mechanical impedance Z_m represents the relationship between the force F applied and the velocity $\dot{x}$. In the frequency domain, the desired impedance behavior can be expressed as:

$$\frac{F(s)}{X(s)} = s \, Z_m(s) = M_d \, s^2 + C_d \, s + K_d \tag{3.50}$$

where M_d, C_d, K_d represent the desired constant matrices for inertia, damping and stiffness.

Figure 3.16 presents the control diagram for Impedance Control. Notice that this scheme is similar to Figure 3.15, however the dynamic behavior is used in place of the static stiffness matrix K_f.

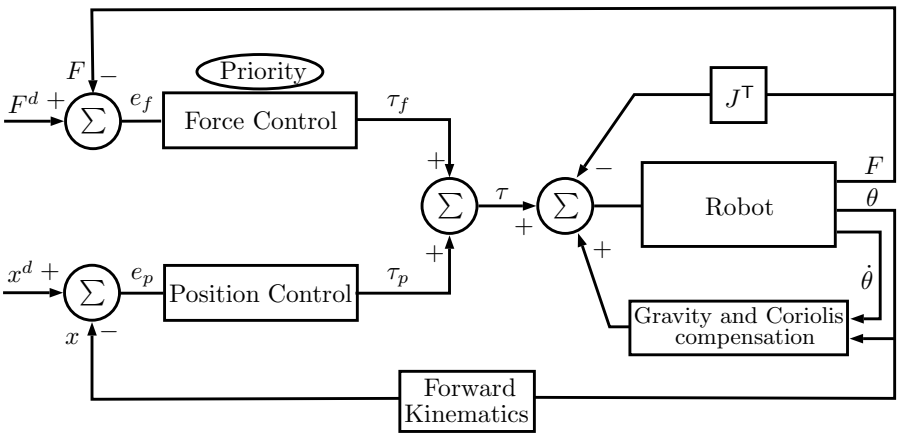


Figure 3.17: Parallel Control.

Parallel Control

Parallel control, originally proposed by (Chiaverini and Sciavicco, 1989, 1993), is a direct force control method, which uses two controllers acting in parallel to achieve the force and position goals. In order to manage the conflict between the two controllers, the strategy is to give priority to the force controller, which prevails over the position controller.

Figure 3.17 presents the parallel control scheme. The dominance of force control may be implemented using a PI force control loop in parallel with a PD position loop (Chiaverini and Sciavicco, 1993).

As an advantage, parallel control does not require detailed geometric modeling of the environment, since there is no selection of force or position controlled directions. However, the design of the controllers is not independent and it must be adjusted to ensure stable behavior.

Hybrid Force-Position Control Method

It has been previously discussed the importance of controlling the grasp by force control. At first analysis, force control and position control represent distinct objectives that might come in conflict without proper treatment.

In this context, the *Hybrid Force-Position* Method allows to combine force and position information, using the concept of *complementary orthogonal subspaces* as developed by Mason (1981) and experimentally verified by Raibert and Craig (1981).

Some issues about the validity of the orthogonal complements concept in the hybrid control theory were raised by Duffy (1990), particularly with respect to the dependence on the choice of units and dimensional inconsistency. However, the hybrid control scheme proposed in this work is free from such problems, since the formulation of the control scheme only requires the end-effector position and the

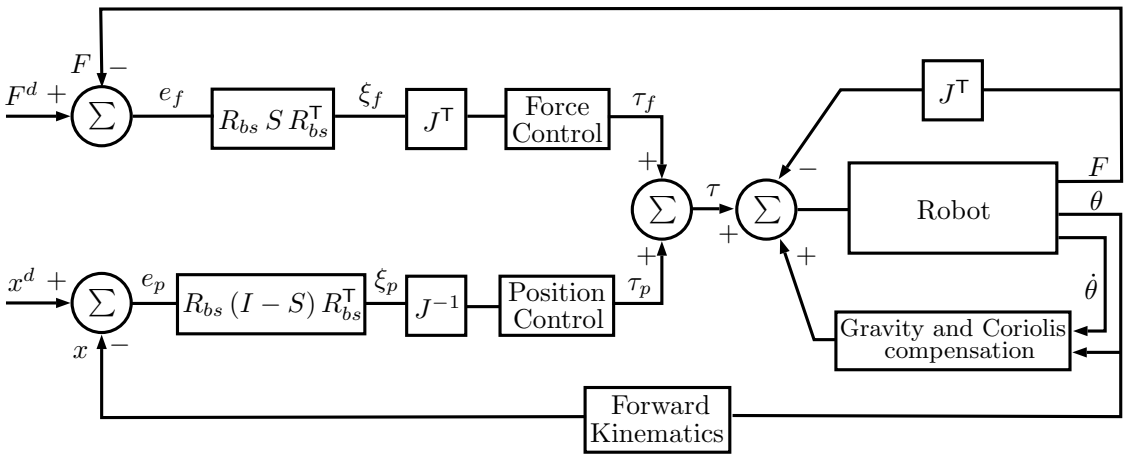


Figure 3.18: The original hybrid position-force control for a single manipulator.

interaction force between the end effector and the environment. Besides, the orientation control problem is treated separately from the position-force problem.

The main advantages of the hybrid method over the impedance method, for instance, are:

- the possibility of controlling the contact force to a desired specification.
- independent design of position and force controllers.

While Impedance Control uses force feedback to correct the position target, Hybrid Control uses two independent position and force loops, which are combined into the final hybrid signal. This explicit force control loop allows to directly specify and achieve the desired value for force.

In order to specify properly and more directly the constraints imposed by the environment, a constraint frame $\mathcal{F}_S$ must be defined, according to the task (Siciliano et al., 2009). By using the complementary selection matrices S and $S = I - S$, that is, diagonal matrices constructed with binary values, it is possible to define the degrees of freedom controlled by force or by position. Along each task direction, the proper control action is active, while the other action is ignored. This is important to avoid interference between the controllers and since the constraints are considered separately, the control laws can be designed independently. Therefore, different position and force specifications may be chosen without conflict, allowing a more efficient control.

The original scheme is presented in Figure 3.18. The position and force errors e_p and e_f are decoupled in the constraint frame $\mathcal{F}_S$, where the matrices $I - S$ and S are constant, leading to the decoupled variables ξ_p and ξ_f.

Notice also that this is a torque-controlled scheme and, as a consequence, it requires the dynamic model of the robot for compensation, expressed as $\mathcal{N}$. Besides, the end-effector orientation is considered in the position control loop and the moment

generated at the end effector is considered in the force control loop. Therefore, this scheme is not simple for implementation, since it requires moment feedback and proper treatment of the orientation control before the selection matrix is applied.

Hybrid Kinematic Position-Force Control Method

An alternative for the dynamic hybrid scheme presented is to use the hybrid kinematic position-force control, proposed in Leite et al. (2010).

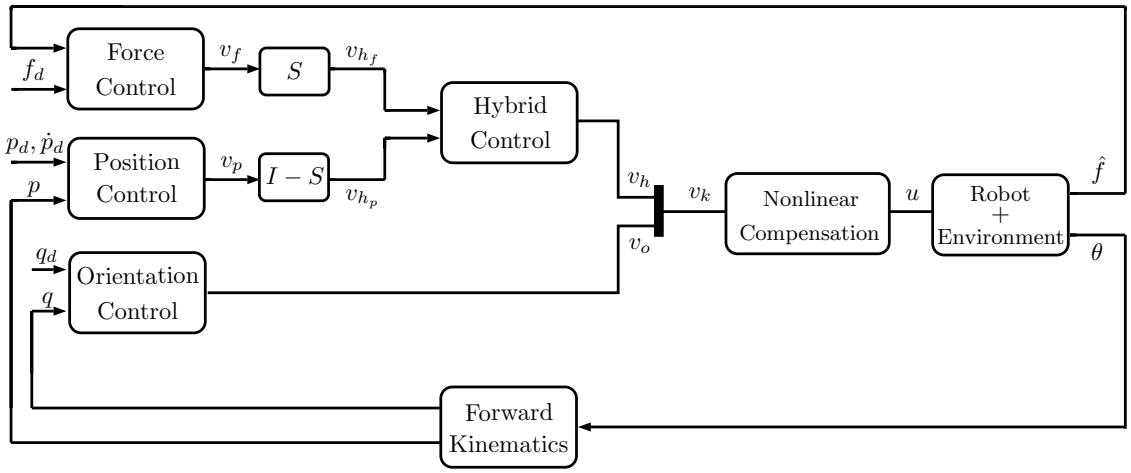


Figure 3.19: Hybrid kinematic position-force control scheme for a single manipulator.

The basic structure of the kinematic control scheme for a single manipulator is represented in Figure 3.19. Usually the position and force of the end effector are expressed in the robot base frame and must be transformed to the constraint frame $\mathcal{F}_S$ before decoupling – the force and position artificial directions are invariant when expressed in this frame. This transformation is done by the rotation matrix R_{bs}^{T}. Then, using the matrices $I - S$ and S, the control signals are decoupled in the complementary motion and force directions, respectively. Since the spaces are orthogonal, the control laws can be independently designed and added, generating the hybrid control signal:

$$v_h = v_{hp} + v_{hf} \tag{3.51}$$

where $v_{hp} \in \mathbb{R}^3$ and $v_{hf} \in \mathbb{R}^3$ are the decoupled control signals acting in the position and force subspaces, respectively. After, the hybrid control signal $v_h \in \mathbb{R}^3$ is transformed back to the base frame, through the matrix R_{bs}.

As it can be noticed, the position control signal v_p of the kinematic control law (3.3) was replaced by the hybrid control signal v_h, without loss of generality.

A contribution of this scheme is the fact that the orientation control signal $v_o \in \mathbb{R}^3$ is calculated separately, that is, it does not get involved in the subspaces

decoupling. Therefore, the kinematic control signal for the hybrid scheme can be represented as $v_k = [v_h \quad v_o]^\mathsf{T}$.

Another important advantage of this scheme is to require less information from the robotic system. Since kinematic control is used, the dynamic parameters (Inertial, Coriolis) of the robot are not required in control and there is no need of sensors to obtain the moment information, since only the end-effector force is used.

3.3.5 Dual-arm Hybrid Position-Force Kinematic Control

Consider the dual-arm manipulation problem. In this context, the control objective is to manipulate an object through the workspace while maintaining an appropriate grasp. Thus, the robotic system should be able to control the squeeze force and efficiently track a desired position and orientation trajectory for the object.

Previous work have studied this subject: (Hayati, 1986; Uchiyama et al., 1987) extended the original hybrid control for two arms, but did not model the system in terms of meaningful task variables. (Uchiyama and Dauchez, 1988) proposed a dual-arm hybrid position-force control scheme in which the task is described in terms of absolute and relative position/orientation components. Figure 3.20 presents this control scheme. Similarly to the single arm control presented in Figure 3.18 , notice that it is also a torque-controlled scheme and it presents the matrix B, responsible for transforming the errors on the orientation angles into equivalent rotation vectors. The main difference is the use of absolute and relative position vector x_O and absolute and relative force vector F_O in the controlled variables. However, they have obtained these variables using the symmetric formulation, which is object dependent.

Besides, Chiacchio et al. (1996) and Caccavale et al. (2000) have modeled the cooperative system variables directly from the end-effectors pose, nevertheless these works have not take advantage of the hybrid control. One of the main advantages of modeling using the cooperative variables is to be able to control efficiently the robotic system without relying on the knowledge of the object shape or size nor in its rigidity.

In section 3.2, we have demonstrated how cooperative control can be used to achieve kinematic position and orientation control of the dual-arm robotic system. However, in order to incorporate the force objective and considering the advantages mentioned in section 3.3.4, we now propose to implement a kinematic hybrid controller, which is less complex and also efficient. If necessary, the kinematic scheme may be used in a cascade loop to achieve dynamic control.

In order to develop this scheme, we use the basic ideas of the hybrid kinematic control scheme proposed in Leite et al. (2010) for a single manipulator and extend

its applications to multiple arms, by means of the cooperative control method, using absolute and relative variables.

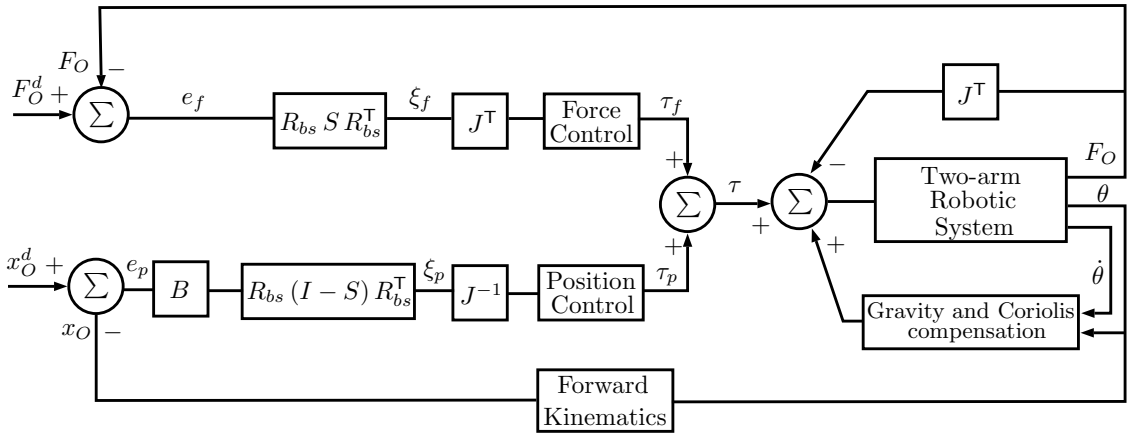


Figure 3.20: The original hybrid position-force scheme for a two arm system, considering absolute and relative variables.

At this point, it is important to understand how the hybrid scheme can be extended for the case of two manipulators. Figure (3.21) shows the constraint frame defined for the object. We assume that servo vision is not available, then in this work the absolute frame is set to be coincident with the constraint frame. Note that this frame varies with respect to the base, but it is constant with respect to the end effectors, unless contact is lost.

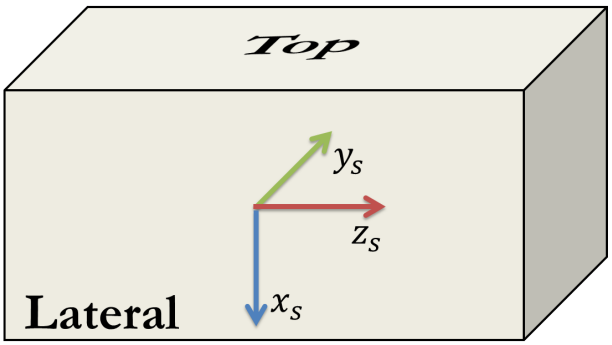


Figure 3.21: Constraint frame $\mathcal{F}_s$ represented at an example object.

In order to efficiently hold the object, it is important to control the squeeze force (or relative force f_r) along the z_s direction. The other directions x_s and y_s should be position controlled. In fact, we do not want the relative position of the end effectors to change in these directions, since this could cause deformation of the object or may lead to partial loss of contact (see Figure (3.22)). Therefore, to accomplish this task the relative position p_r and the relative force f_r should be exploited by the hybrid scheme.

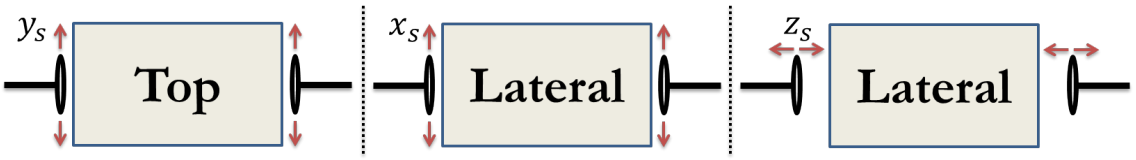


Figure 3.22: Position and force controlled variables, according to the directions.

The design of the hybrid controllers utilizes the selection matrices S and $I - S$ are given by:

$$S = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad I - S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.52}$$

which are responsible for the decoupling the control actions of force and position, respectively, acting at the constraint frame.

Using the hybrid method, it is possible to design separate force and motion controllers. In the case of object manipulation, relative force and position are in general desired to be constant and their reference values are more naturally specified in the constraint frame. Now, consider the relative force and position erros (e_{pr} and $e_{fr} \in \mathbb{R}^3$), defined as:

$$e_{pr} = R_{bs}\, {}^s p_r^d - p_r \tag{3.53}$$

$$e_{fr} = R_{bs}\, {}^s f_r^d - f_r \tag{3.54}$$

where ${}^s p_r^d$ is the desired relative position and ${}^s f_r^d$ is the desired relative force, both expressed in the constraint frame $\mathcal{F}_s$. Thus, the control laws for these variables may be constructed as:

$$v_{pr} = K_{pr}\, e_{pr} + R_{bs}\, {}^s \dot{p}_r^d \tag{3.55}$$

$$v_{fr} = K_s^{-1} \left[K_{fr}\, e_{fr} + R_{bs}\, {}^s \dot{f}_r^d \right] \tag{3.56}$$

where K_{pr} and $K_{fr} \in \mathbb{R}^{3\times3}$ are the gain matrices of relative position and force, respectively; $K_s = k_s\, I_3 \in \mathbb{R}^{3\times3}$ is the the gain matrix of the object stiffness. In general, position and force are measured in task space with respect to the base frame. However, in this frame, selection matrices are dependent on the end-effector pose. Therefore, the control signals should be decoupled in the constraint frame, where the selection matrices are invariant. Then, assuming that the Jacobian $J(\theta)$ is expressed in the base frame $\mathcal{F}_b$, the decoupled signals are transformed again to $\mathcal{F}_b$. The transformation from base to constraint frame (and vice-versa) is done by using the rotation matrix R_{bs}^{T}. Because the constraint frame is time-variant with respect to the base frame, the proposed position and force components (v_{hp} and

v_{hf}) of hybrid control law also presents an orientation-dependent term for ensuring stability (see proof in Appendix A), as discussed in (Leite et al., 2010), as follows:

$$v_{hp} = R_{bs}\,(I-S)\,R_{bs}^{\mathsf{T}}\,v_{pr} - R_{bs}\,(I-S)\,R_{bs}^{\mathsf{T}}\,2\,\mathcal{S}(\omega_b)e_{pr}\ , \tag{3.57}$$

$$v_{hf} = R_{bs}\,S\,R_{bs}^{\mathsf{T}}\,v_{fr} -\ R_{bs}\,S\,R_{bs}^{\mathsf{T}}\,K_s^{-1}\,4\,\mathcal{S}(\omega_b)\,e_{fr}\ , \tag{3.58}$$

where $\omega_b \in \mathbb{R}^3$ is the angular velocity of constraint frame $\mathcal{F}_s$ with respect to the robot base $\mathcal{F}_b$, expressed in frame $\mathcal{F}_b$. Therefore, the hybrid control signal can be described as:

$$v_{hr} = v_{hp}+v_{hf} = R_{bs}\,(I-S)\,R_{bs}^{\mathsf{T}}\,[v_{pr}-2\,\mathcal{S}(\omega_b)e_{pr}]+R_{bs}\,S\,R_{bs}^{\mathsf{T}}\,[v_{fr}-\,K_s^{-1}\,4\,\mathcal{S}(\omega_b)\,e_{fr}]\ . \tag{3.59}$$

Finally, the control signal v in (3.29) is reformulated using v_{hr} in place of v_{pr}:

$$v = \begin{bmatrix} v_{pa} \\ v_{oa} \\ v_{hr} \\ v_{or} \end{bmatrix}\ , \tag{3.60}$$

and considering the inverse kinematics (3.28) the control system is linearized to:

$$\begin{bmatrix} \dot{p}_a \\ w_a \\ \dot{p}_r \\ w_r \end{bmatrix} = \begin{bmatrix} v_{pa} \\ v_{oa} \\ v_{hr} \\ v_{or} \end{bmatrix}\ . \tag{3.61}$$

Now, consider the position and force errors $\xi_{pr} \in \mathbb{R}^3$ and $\xi_{fr} \in \mathbb{R}^3$ expressed at the base frame after the position-force decoupling are:

$$\xi_{pr} = R_{bs}\,(I-S)\,R_{bs}^{\mathsf{T}}\,e_{pr}\ , \tag{3.62}$$

$$\xi_{fr} = R_{bs}\,S\,R_{bs}^{\mathsf{T}}\,e_{fr}\ . \tag{3.63}$$

Then, the following theorem can be formulated:

Theorem 1. *(Stability of Dual-arm Hybrid Position-Force Kinematic Control) Consider the closed-loop cooperative control system described by (3.27) and (3.28) with hybrid control law (3.59) composed by relative position (3.55) and force (3.56) controllers and consider also the absolute position, absolute orientation and relative orientation controllers as described in (3.29). Assume the reference signals are bounded and consider assumptions (A-3.1) and (A-3.2). The following properties can be guaranteed:*

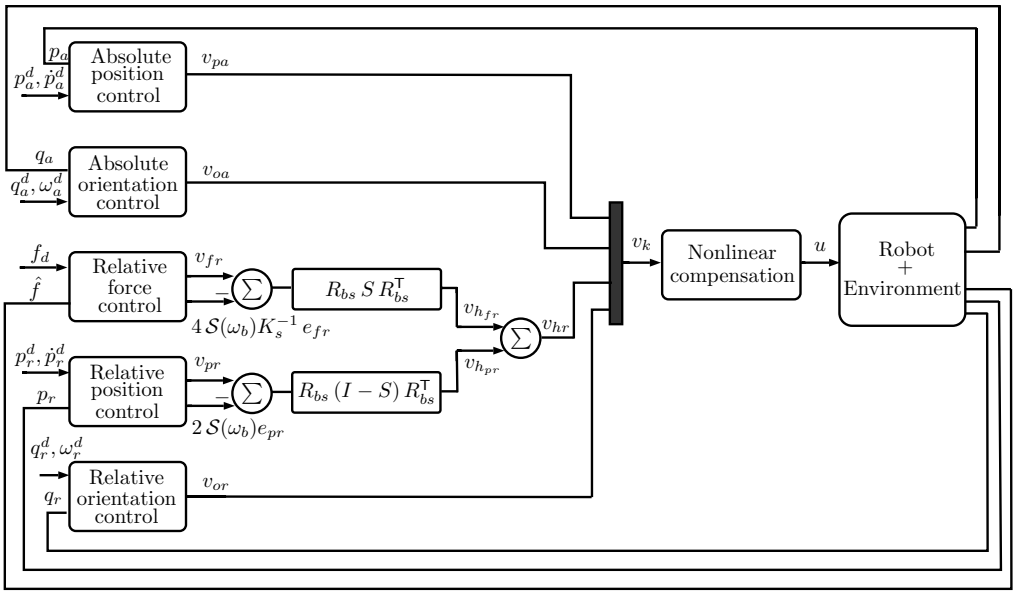


Figure 3.23: Dual-arm Hybrid position-force scheme.

(i) Every signal of the closed-loop system is bounded.
(ii) $\lim_{t\to\infty} e_{pa}(t) = 0, \lim_{t\to\infty} \xi_{pr}(t) = 0, \lim_{t\to\infty} \xi_{fr}(t) = 0, \lim_{t\to\infty} e_{qa,\,v}(t) = 0, \lim_{t\to\infty} e_{qr,\,v}(t) = 0, \lim_{t\to\infty} e_{qa,\,s}(t) = \pm 1$ *and* $\lim_{t\to\infty} e_{qr,\,s}(t) = \pm 1$
Therefore, the closed-loop system is almost-globally asymptotically stable.
Proof: *See in appendix A.*

The first contribution of this new hybrid scheme is to extend the kinematic hybrid control from single to multiple manipulators. Also, the developed scheme shows the feasibility of using, for cooperative manipulation, the hybrid position-force control only in the relative variables, which are significant for maintaining a secure grasp. The last contribution is given by the new interpretation for the constraint frame $\mathcal{F}_S$, originally developed for representing the normal direction of a surface, which now is used to represent the desired force-controlled direction on the object. While the selection matrices S and $I - S$ are constant, the constraint frame $\mathcal{F}_S$ is assumed to be time-variant and it is updated according to the current object orientation, which is estimated by variable q_a.

The main features of the hybrid scheme are: (1) Use of kinematic approach, (2) the orientation problem is time-variant and expressed using quaternions, (3) the manipulation task is 6-DoF (4) modeling does not assume the object has a known and fixed geometry – it can be flexible and (5) the control scheme is centralized and it gives priority to accomplish the cooperative task objective, rather than each manipulator positioning.

3.3.6 Numerical simulations

In order to illustrate the theoretical performance of the proposed control scheme, the results for numerical simulation are presented. The simulation is implemented in Simulink, with the support of Robotic Toolbox, using the two-arm model with similar mechanical of the real robot used in the experiments (Baxter).

Simulation Parameters

The simulation parameters of control are shown in Table 3.4. Although the scheme does not require any knowledge about the object dimension, the parameters used to simulate the flexibility behavior during the grasp are shown in Table 3.5.

Table 3.4: Simulation parameters for cooperative hybrid position-force control.

Parameter	Value
K_{pa}	1.5 s^{-1}
K_{pr}	2.25 s^{-1}
K_{oa}	1.5 $rad\,s^{-1}$
K_{or}	2.25 $rad\,s^{-1}$
K_{fr}	2.0 $m\,N^{-1}s^{-1}$

Table 3.5: Simulation parameters for a flexible object.

Parameter	Value
l_0	0.25 m
k_s	140 N/m

Tracking Circle Plane-XZ

In the numerical simulation, the position control objective is to track a circular trajectory in plane XZ of the robot base frame: $x = 0.6 + 0.2\sin(1/20\,t)$; $y = 0.0$ and $z = 0.4 + 0.2\cos(1/20\,t)$. The force control objective is to regulate the grasp to a constant reference (7 N).

Figure 3.24 (a) describes the absolute position and reference trajectories. The trajectory is smooth and it tracks the reference. Figure 3.24 (b) illustrates the absolute position error. After the transient, the absolute position error converges asymptotically to zero.

The relative position and the reference is shown in Figure 3.25 (a) and the error is also shown in Figure 3.25 (b). The position-controlled directions $p_{r,\,x}$ and $p_{r,y}$ of the relative position converges to zero, while the force-controlled direction $p_{r,\,z}$ converges to a 200 mm distance. Indeed, this is the expected value considering the

object elastic deformation parameter adopted in the simulation and the force control setpoint.

The absolute and relative orientation errors are illustrated in Figure 3.26. For a proper manipulation, priority is given for the relative orientation error, since it is more important to avoid lose of alignment. The absolute orientation error presents a small and acceptable oscillation around 0.03 rad. Meanwhile, the relative orientation error is kept close to zero.

The force applied to the object and the force error are presented in Figure 3.27. The force converges smoothly to the reference set (7 N) and the error presents small oscillation around zero, caused by the object flexible behavior.

The Cartesian control signal is demonstrated in Figure 3.28. It can be noted that the control signals are smooth and bounded. The joint velocities are also smooth and do not require any peak or unfeasible velocities from the actuators, as can be seen in Figure 3.29 and Figure 3.30.

The representation of the robot arms and the trajectory performed by the object (frame $\mathcal{F}_a$) is shown in Figure 3.31. It can be noted that, after the transient, the trajectory tracks the reference with acceptable performance.

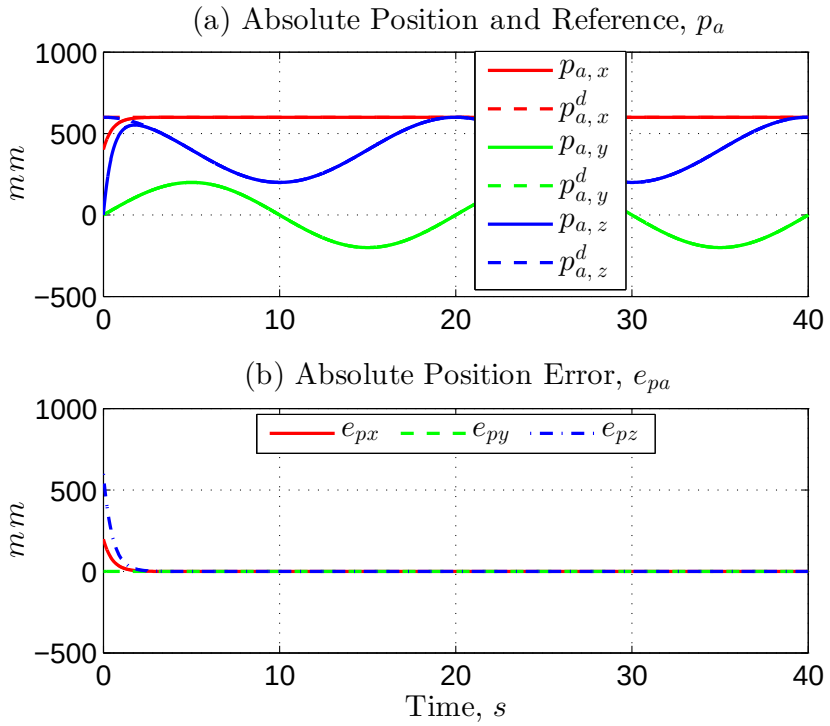


Figure 3.24: Hybrid control: (a) absolute position and reference (b) absolute position error.

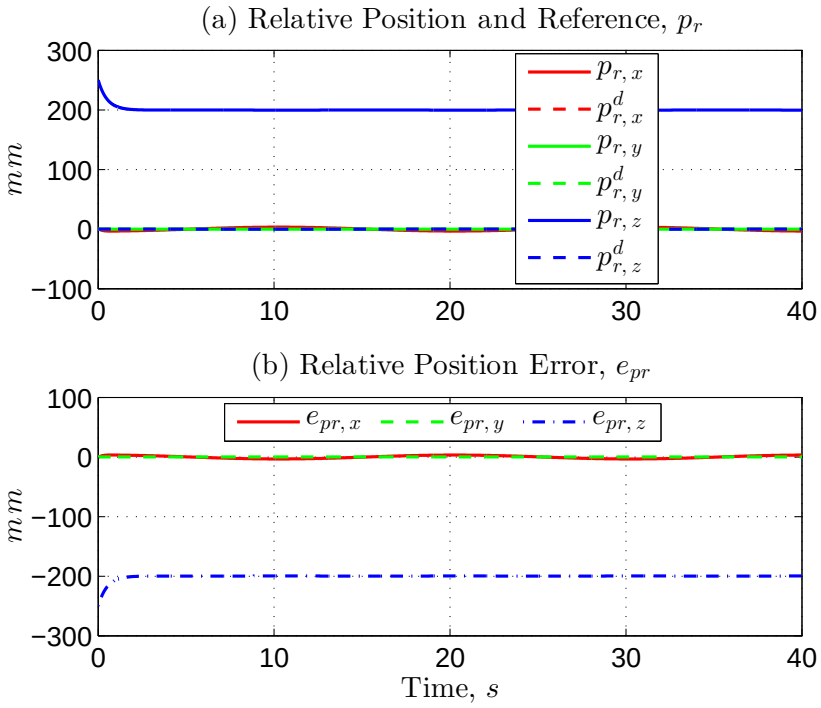


Figure 3.25: Hybrid control: (a) relative position and reference (b) relative position error.

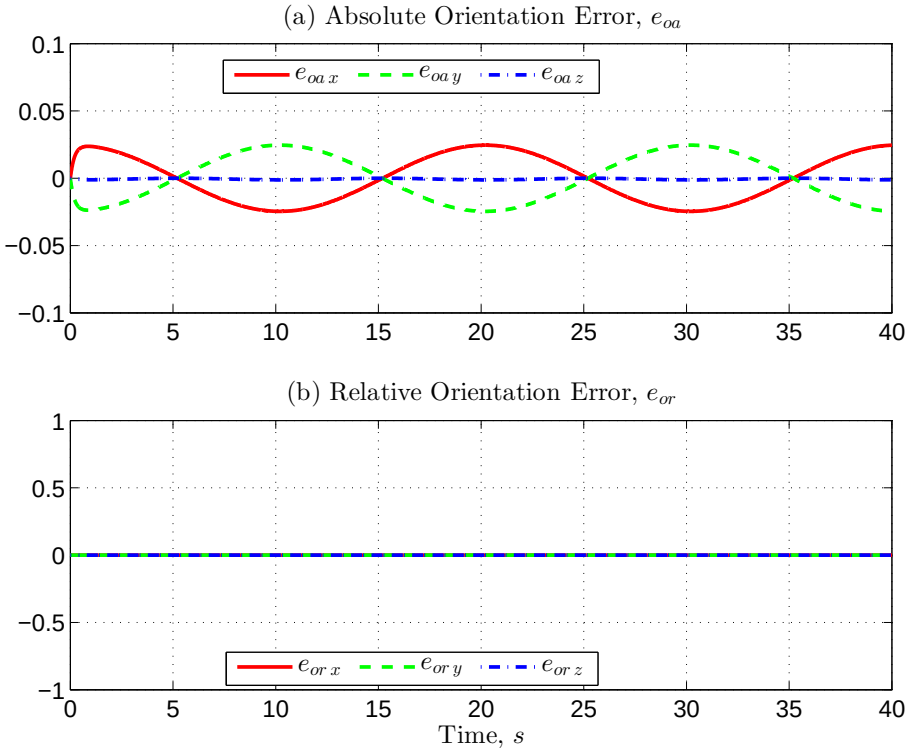


Figure 3.26: Hybrid control: (a) absolute orientation error (b) relative orientation error.

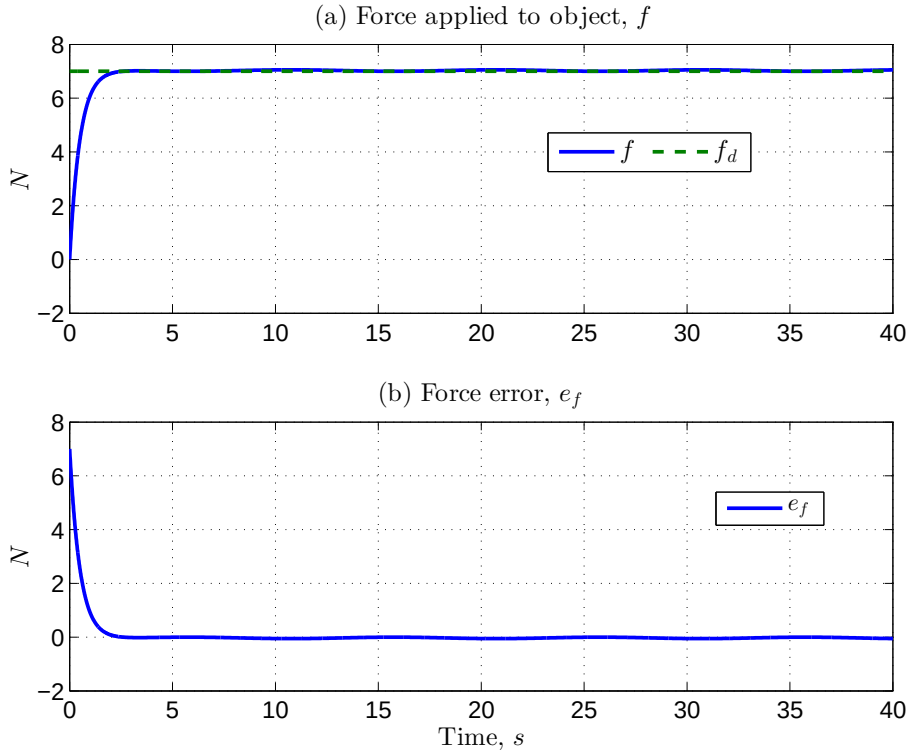


Figure 3.27: Hybrid control: (a) force applied to object (b) force error.

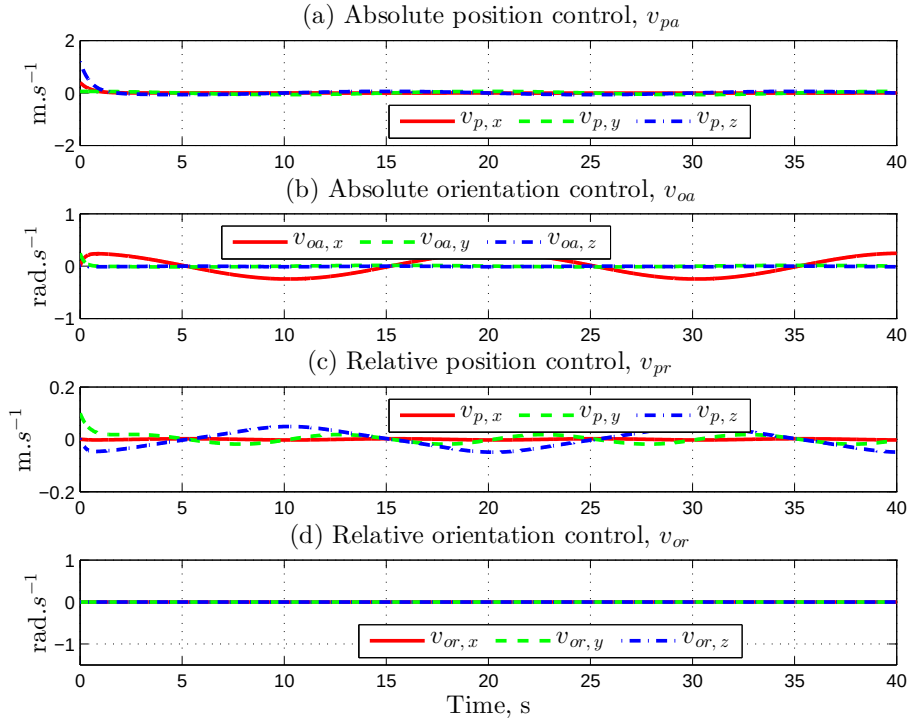


Figure 3.28: Hybrid control: (a) absolute position control (b) relative orientation control (c) relative position control (d) relative orientation control.

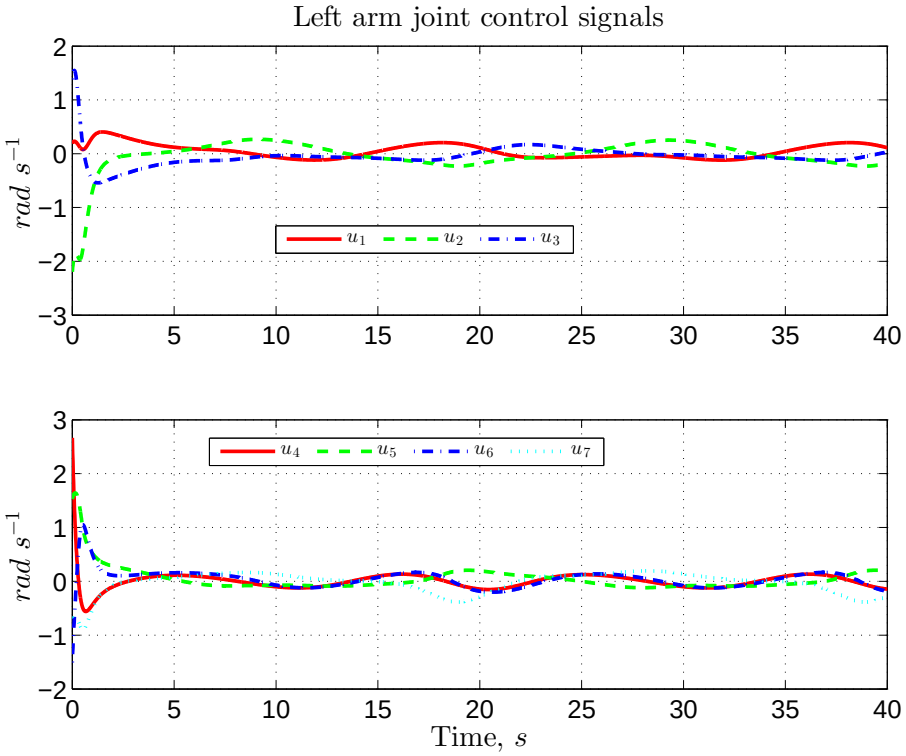


Figure 3.29: Hybrid control: left arm joint control signals.

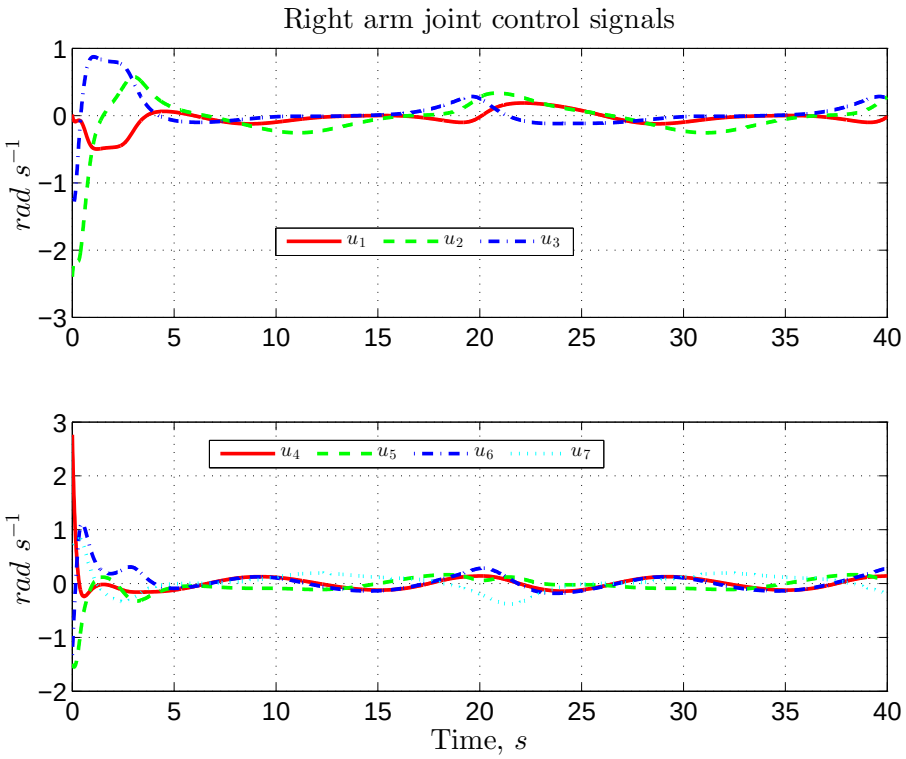


Figure 3.30: Hybrid control: right arm joint control signals.

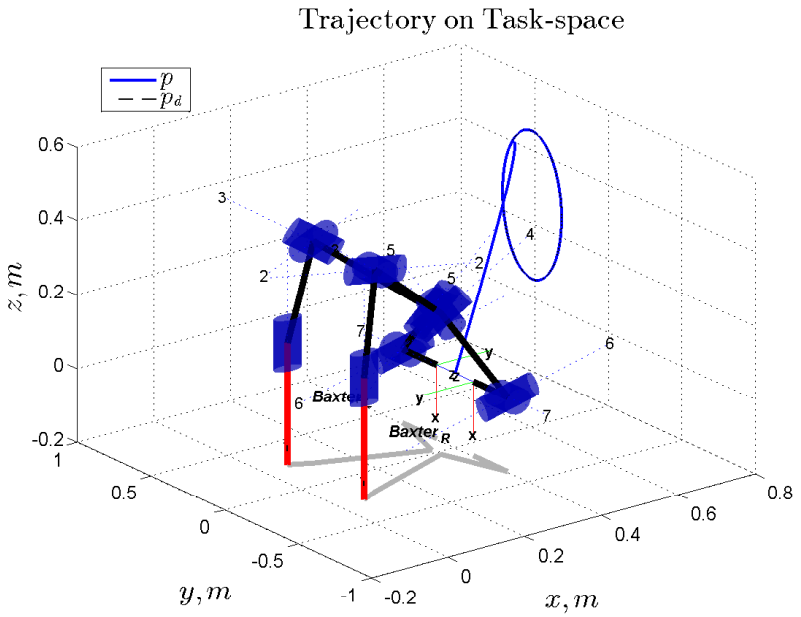


Figure 3.31: Hybrid control: trajectory performed by manipulators (with reference).

3.4 Conclusion

In this chapter, control strategies for cooperative manipulation are discussed. Initially, the kinematic control of a single manipulator is studied, where position, force and orientation control schemes are independently analyzed. Then, the problem is extended for the kinematic control of a cooperative system, which is modeled using the cooperative task-space formulation.

For the sake of testing and better comprehension before implementation in the robot, a numerical simulation is performed using a two-arm model with similar mechanical structure of the real robot used in the experiments. At this step, the objective proposed for the cooperative manipulation was to achieve coordinated motion of the manipulators and the proposed task involved tracking of trajectories designed for the absolute and relative frames. The results confirm the feasibility of the cooperative task-space formulation.

In the sequence, force control is considered for the cooperative manipulation. As part of the analysis, modeling of an generic object is described and a analysis of the contact forces and friction which happen during manipulator-object interaction is studied. Next, conflicts between the force and position objectives are considered and the characteristics and differences of the most-known interaction methods – impedance control and hybrid position-force control – are discussed.

The studies led to a new cooperative hybrid position-force control scheme for dual-arm robots. The scheme is purely kinematic and does not depend on the dynamic model of the robot, neither of the object properties. Additionally, the constraint frame is analyzed in order to determine the subspace decomposition, which is required by a hybrid controller, and guarantee its implementation feasibility.

The stability analysis is verified using the Lyapunov Stability Theory. Finally, a new numerical simulation is performed for the proposed control scheme, including the object behavior, and the results corroborate the develop method.

Chapter 4

Experimental Setup

In this chapter, the experimental setup, including the Baxter robot, is described. A real manipulation task is proposed and verified. For better comprehension, the task is conceptually subdivided in two main steps: the autonomous grasp of an object and the hybrid position-force control for cooperative manipulation (Figure 4.1).

Considering a scenario where the robot is used to manipulate remote objects, it is important that the robotic system is prepared to perform automated tasks. Indeed, the robot ability to grasp autonomously an object may lead to a safer, less costly and less complex manipulation task. Therefore, this chapter presents as part of the experimental setup, a methodology for complete identification and grasping of an unknown object using dual-arm robots.

After the grasp closure is performed, the proposed cooperative hybrid control scheme is activated. On the purpose of obtaining the experimental results, different predefined trajectories are tested for the object position. The results presented are commented and confirm the feasibility and effectiveness of the method (Section 4.3).

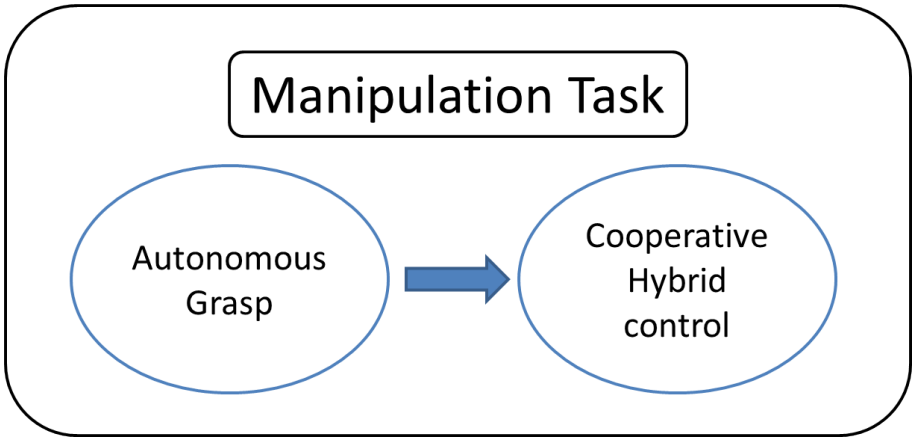


Figure 4.1: The manipulation task proposed has two main steps.

4.1 Problem Formulation

In this section, the problem formulation for the manipulation task is described (Figure 4.2). A dual-arm robot composed by two manipulators with torque sensors in the joints and two coupled cameras (one in each arm) is used – Baxter robot.

The task of interest is to grasp an object and move it according to a reference trajectory defined by an operator in a remote local.

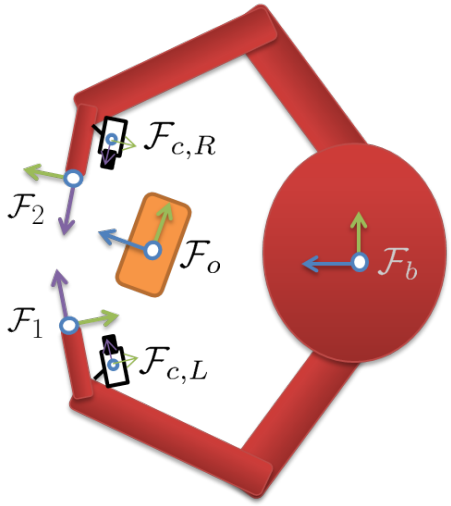


Figure 4.2: Experimental setup: robot, cameras and object.

The force is controlled autonomously by a force control loop. The desired contact force f_d is determined experimentally and it can not exceed the limits to avoid that the contact is lost ($f_d > f_{min}$) or that the object is damaged ($f_d < f_{max}$). Also, in the experiments there are no force sensors attached to the robot end effectors and the estimated contact force $\hat{f}$ is obtained from the direct measurement of torque sensors at each joint (Section 2.3.1).

Therefore, the proposed task involves many challenges as: (i) control the pose of an object manipulated by the two robotic arms; (ii) control the contact force indirectly from the torques in the joints; (iii) visually identify the initial pose of the object to perform the procedure grip safely and efficiently.

4.1.1 Baxter Robot Description - Hardware and Software

The robotic system is composed by a Baxter^{TM} Robot (Rethink Robotics) with 7-DoF, where each arm is endowed by an interchangeable end effector and a video camera for recognizing objects, parts and the workspace (Guizzo and Ackerman, 2012).

Baxter also has a 360 degrees sonar array and a front camera mounted in its head for human presence detection. Each arm has torque, velocity and position sensing

in each joint. Other integrated sensors are 3-axis accelerometers and infrared range finders in each wrist.

The joints are manufactured with Series Elastic Actuators (Pratt and Williamson, 1995; De Luca and Book, 2008), where the joint flexibility is about $843\,Nm\,rad^{-1}$ for the shoulder and elbow joints and $250\,Nm\,rad^{-1}$ for the wrist joints. The robot positional accuracy is about 5 mm for the whole workspace, however it is about 0.5 mm for a limited envelope according to the manufacturer. The maximum joint speed are $2.0\,rad\,s^{-1}$ for the shoulder and elbow joints and $4.0\,rad\,s^{-1}$ for the wrist joints. The maximum payload including the end effector is about 2.2 Kg.

Based on the manufacturer Unified Robot Description Format (URDF) model, we have created a simplified representation of the robot arm (Figure 4.3), where the links lengths L_i are expressed in meters and the robot joints J_i are of revolution type. From this model, we can obtain the standard parameters of the Denavit-Hartenberg convention for the robot arm (Siciliano et al., 2009).

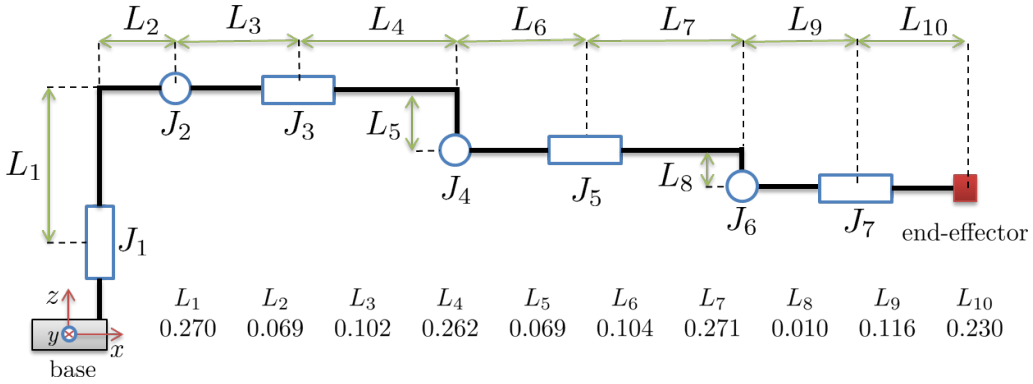


Figure 4.3: Links and joints representation for the Baxter arm.

For the manipulation task, we have coupled a tool to each end effector. The tool is a rubber disk, which intentionally increases the friction coefficient and protects the end effectors against impacts.

Due to limited bandwidth capabilities of the Baxter cameras and to avoid processing overload in the robot hardware, a Logitech HD Webcam C270 is directly connected to the computer by a USB port. The image distortion coefficients (radial and tangential) k, as well as the focal length f, the principal point coordinates c_x, c_y and the skew coefficient α_s, are determined using a conventional calibration method provided by the OpenCV library. The calibrated camera parameters are presented in table 4.1.

In the experiments, we have used a Baxter Research Robot. The main difference compared to the Baxter industrial model is related to the software. The research model is able to perform tasks according to the algorithms developed using the open-source Robot Operating System (ROS), which is code platform with libraries

Table 4.1: Camera parameters

Resolution	FPS	f_α	c_x	c_y
1280×800	20	402.3	669.6	404.3

and tools available to assist in the development of applications for robotic systems (Quigley et al., 2009).

The robot communicates via ROS messages with a local computer. This computer is responsible for the main data processing – it executes the visual identification algorithm and the control loop calculations.

The Baxter robot has a 3rd Gen Intel Core i7-3770 3.4 GHz processor with 4 GB DDR3 RAM memory and 128 GB Solid State Drive. During the experiments, the local computer used is a Macbook air mid-2012, with an Intel Dual Core i5 1.8 GHz processor, 4 GB DDR3 RAM memory and 128 GB Solid State Drive.

The software routines created in this work were developed in Python language. The control algorithm is executed with 50 Hz frequency (maximum loop time is 20 ms). After the control loop calculations are performed, a message is sent to the robot via ROS topic.

The robot receives the commands using the *Joint Control Listeners*, which store the commands to be retrieved asynchronously by the *RealTime Motor Control Loop* (the highest priority process on Baxter PC) every 1 ms (1 KHz frequency). The velocity command is then sent to each Joint Control Board (JCB) and executed within the same 1 ms timeslot. These layers are not accessible for user modification.

4.1.2 Object description

The physical characteristics of the object are considered to be not known a priori, but its width can be determined during the grasping procedure. For simplicity, it is also assumed that the dimensions of the object not too small so that it can be picked by a claw, but it is sufficiently large so that it can be carried by two robots manipulators with known load capacity. Besides, the object must be placed inside the robot workspace, in such a way that it is possible for the end effectors to grasp it.

The object utilized in the experiments is a cardboard box with dimensions $0.325\,m \times 0.215\,m \times 0.14\,m$ ($L \times W \times H$). Additionally, two tags are attached to the largest lateral faces.

4.1.3 Validation of the force estimation

In this work, we propose to obtain the contact force information using the data available from the robot torque sensors, as discussed in Section 2.3.1.

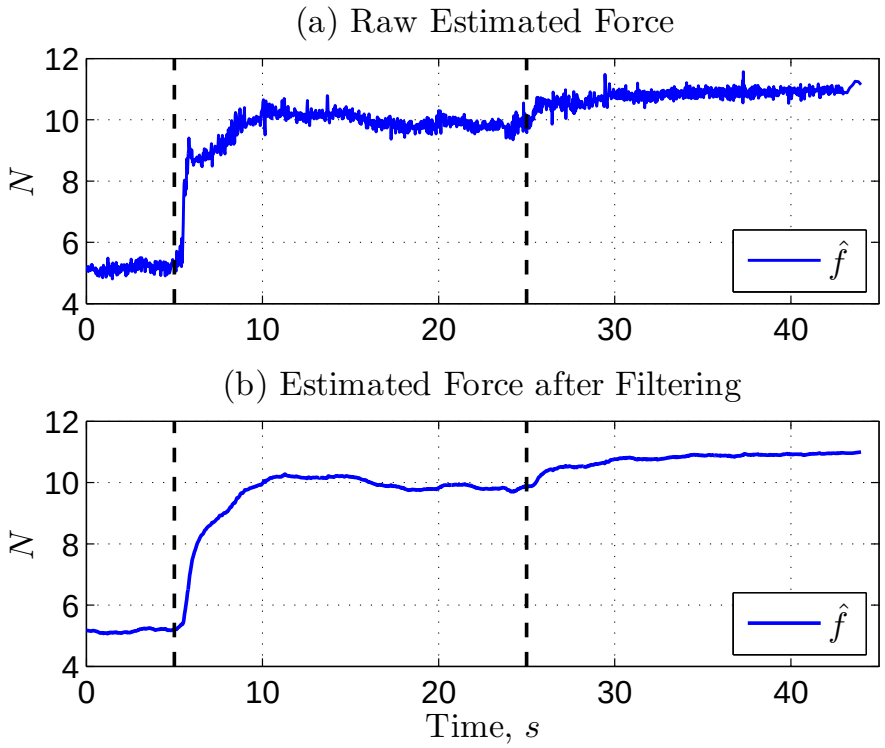
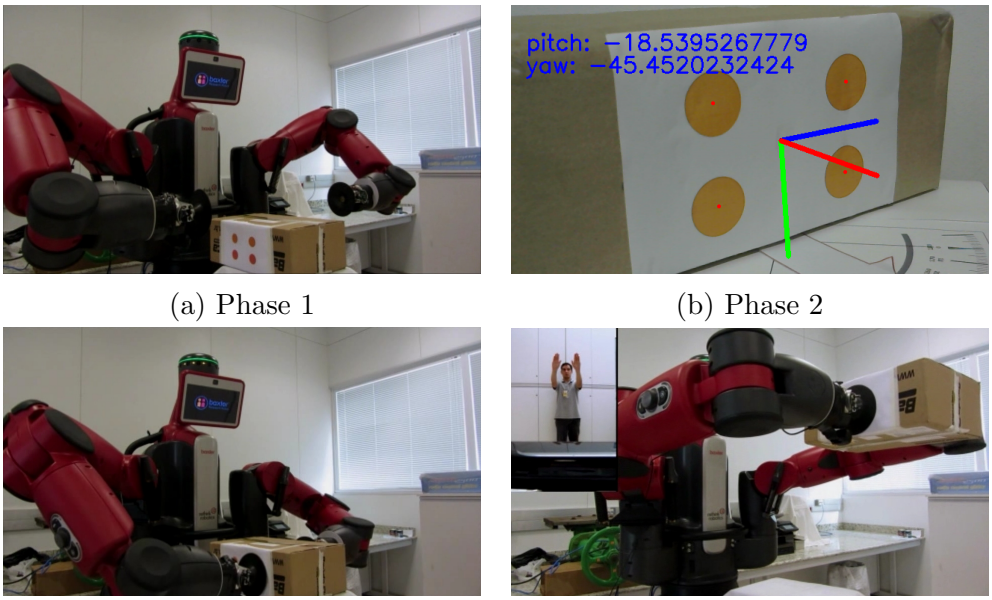


Figure 4.4: Experiment for force estimation (a) using data obtained from the torque sensors and (b) after a low-pass filter.

In order to validate the estimation, an experimental verification is performed. Baxter provides an estimative of the gravitational torques based on the robot kinematic model. Thus, to ensure correct estimation of the gravitational torques, the robot is leveled with respect to the ground, using the robot built-in bubble level. Next, the torque sensors are calibrated according to the manufacturer procedure (Rethink provides an automatic calibration algorithm).

In the beginning of the experiment, the end effector is positioned pointing upside. Then, two reference weights are added to the top of the end effector at different moments: at $t = 5$ s, the first weight $w_1 = 0.522$ Kg is added and at $t = 25$ s, the second weight $w_2 = 0.103$ Kg is joined to the first weight. The data obtained from the torque sensors is used in the estimation algorithm and the force obtained is presented in Figure 4.4 (a), where $\hat{f}$ is the force estimation in z_e direction, which is the information required by the force control loop.

As can be noticed, the sensors noise is significant - the amplitude is about 0.6 N. In order to reduce noise, a digital low-pass filter, with transfer function $H(z) = 0.95/(z - 0.05)$ is used. The resultant filtered estimation is presented in Figure 4.4 (b).



(a) Phase 1 (b) Phase 2

(c) Phase 3 (d) Phase 4

Figure 4.5: Phases of the proposed manipulation task.

4.1.4 Sequence of operation

The manipulation task is divided in four phases (see Figure 4.5):

- **Phase 1 - Initial Positioning** : Using a kinematic position control loop for each manipulator, the two arms are positioned in predefined pose and the camera attached to the manipulator is able to visualize the markers. If needed, the operator can adjust the end-effectors position.

- **Phase 2 - Pose Identification** : Using the algorithm described in section 4.2.2, the relative position and orientation of the object TAG and the camera is obtained.

- **Phase 3 - Approximation and Object Grasp** : Using again a kinematic position control loop for each manipulator, the end effectors are repositioned close to the grab location, which is autonomously determined using the visual markers center position (Section 4.2.5). Then, using the kinematic control for each manipulator, the reference is gradually incremented in the grasp direction ($\vec{h}_{12}$), until the force detected reaches a specified value.

- **Phase 4 - Hybrid Position-Force Control Loop** : The Hybrid Cooperative Control is turned on and the control scheme regulates the force and tracks the desired position for the object, which is chosen as predefined circular trajectories. The force measure used in the force control loop is the estimated contact force $\hat{f}$ obtained from the direct measurement of torque sensors at each joint.

4.2 Methodology for dual-arm grasp of objects

The importance of autonomous grasp has motivated the development of a methodology for dual-arm grasp. For example, if the robotic system is used to manipulate remote objects, some common problems as communications delays and wrong perception of the operator may introduce difficulties for grasping the object. Besides, using force feedback devices, such as haptic controllers, increase the costs of the teleoperated system. Hence, autonomous grasp allows the operator to focus on the main activity: to control the desired trajectories for the manipulation task.

Despite the benefits, autonomous grasp is still a challenging problem and subject of research (Saxena et al., 2008). Different technologies can be used to identify an unknown object and obtain the main information needed. For instance, the use of laser, structured light systems or sonar may provide rich information, but can increase costs and the data treatment may add complexity to the problem (Lippiello et al., 2013). On the other hand, the use of monocular cameras offer a good cost/performance and usually are already integrated in the robotic system – if not, they can be easily installed. The development of open-source libraries to treat image information, in the last years, is also an attractive motivation for the use robot vision.

Usually, there are two main phases in the process of grasping an unknown object: the visual identification of the object pose and the grasp planning.

Several techniques can be employed for visual identification of the pose (Yoshimi and Allen, 1994; Collet et al., 2009; Wang et al., 2010). Some methods are based on processing images of the environment and extract natural features (Rice et al., 2006). For instance, the volumetric scene reconstruction (Dyer, 2001) and the surface scene reconstruction approaches (Xu and Prince, 1998). It is worth noting that volumetric methods, since a spatial model of the object is obtained, are more suitable for collision avoidance, while surface methods are more adequate for grasp planning (Lippiello et al., 2013). Other possibilities for object reconstruction depends on the use of several kinds of sensors, as the vision-based structured light systems (Ribo and Brandner, 2005).

However, object reconstruction based on naturally-occurring features is a costly problem in computation. In many cases, the use of artificial features – also known as fiducial markers – rather than unconstrained images can greatly simplify the pose identification problem. Indeed, this approach improves the runtime performance and the reliability in object localization (Rice et al., 2006).

In this work, we focus on objects with simple geometry. Some assumptions on the object are:

(A-4.1) It has locally parallel faces.

(A-4.2) It is convenient to attach tags in the grasping points.

For instance, the object could be a box or a large ball. After marking the grasping point with tags, a visual fiducial system can provide relative position and orientation of a tag, which can be composed by multiple markers, with respect to the camera. (Olson, 2011).

The choice of the tag system is an interesting topic of research. Many 2D bar code style tags have been proposed, such as Cantag (Rice et al., 2006), ARTag (Fiala, 2010) and AprilTag (Olson, 2011); these tags are useful not only for pose identification but also for object discrimination, providing unique markers. Other approaches, as the work Faria et al. (2015) choose light markers, which could be adapted to the object (valves, in this case) in an industrial environment without trouble and offer the advantage of being robust to changes in the environment lightning conditions. However, since the manipulation task proposed here deals with moving objects, problems such as powering the markers would arise.

In this case, the use of passive markers, which do not require electrical power and can be made with planar patterns for convenient printing and mounting (Fiala, 2010), is more suitable. Besides, studies have shown that circular tags offer better location and pose accuracy (Rice et al., 2006) and low maximum error of centroid location due to digitization (Bose and Amir, 1990). Therefore, in this work, the methodology for visual identification of the object pose proposed in (Faria et al., 2015) is used, except that now a tag composed of colored circular markers is chosen in place of the light markers (Figure 4.6).

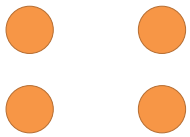


Figure 4.6: Tag composed of four circular markers.

It is worth noting that even with the modern tags, feature extraction under different lighting conditions, surface geometries, and textures is still a challenging and error prone problem. In the following, the use of key features to obtain the relative camera-object pose is described.

4.2.1 Object identification using key features

The objective of the identification algorithm treated here is to find the relative position and orientation of the object with respect to the camera, that is, describe the pose of the frame $\mathcal{F}_m$, where the markers are placed, with respect to the camera frame $\mathcal{F}_c$.

The input for the system is the image (either a static image or a video stream) containing the tag attached to an 3D object, where the 3D object coordinates projected into the camera frame create the 2D image coordinates (see Figure 4.7). These 3D-2D correspondences are also referred as the *homography* between the known position of the points in the tag frame to the measured pixels in the image frame (Petersen, 2008*a*).

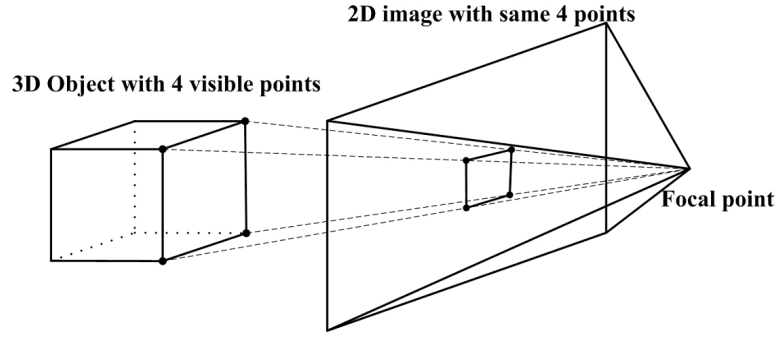


Figure 4.7: Perspective problem using four feature points. Extracted from (Petersen, 2008*a*)

Fischler and Bolles (1981) have used the term *Perspective n-Point problem* or simply PnP problem, for this type of problem using n feature points from the tags.

To solve the PnP problem, a minimum number of correspondences are required. The P3P problem is the smallest set of feature points ($n = 3$) that yields a finite number of solutions (Gao et al., 2003). Fischler and Bolles (1981) noticed that there are at most four possible solutions to the P3P (with three non-collinear points) equation system.

In theory, the pose of a calibrated camera can be uniquely determined from a minimum of four coplanar and not collinear points (Schweighofer and Pinz, 2006). In practice, Oberkampf et al. (1996) have shown that instead of a single solution, the P4P problem always presents two acceptable solutions (two local minima), which are mirror images with respect to a plane parallel to the image plane. They also developed a solution for this pose ambiguity: the algorithm POSIT maintains two alternative solutions and it uses a quality measure to decide for the better one. However, this method produces acceptable solution only when the camera distance is very large compared to the depth in the scene and, to cope with this limitation, Schweighofer and Pinz (2006) developed a new algorithm for a unique and robust solution of the P4P problem.

Many other algorithms have been developed for solving and optimizing the PnP problem. The most relevant are RANSAC (Fischler and Bolles, 1981), Direct Linear Transformation (DLT) (Abdel-Aziz et al., 2015) and Efficient Perspective-n-Point Camera Pose Estimation (EPnP) (Lepetit et al., 2009).

4.2.2 Image Processing

The objective of the image processing described, in this subsection, is to obtain a list of markers given an input image (either a static image or a frame from a video stream) (Fiala, 2010). The precise and efficient identification of these key points in the image is an important step for an adequate pose estimation.

From the original image (Figure 4.8) each step is described in the following:

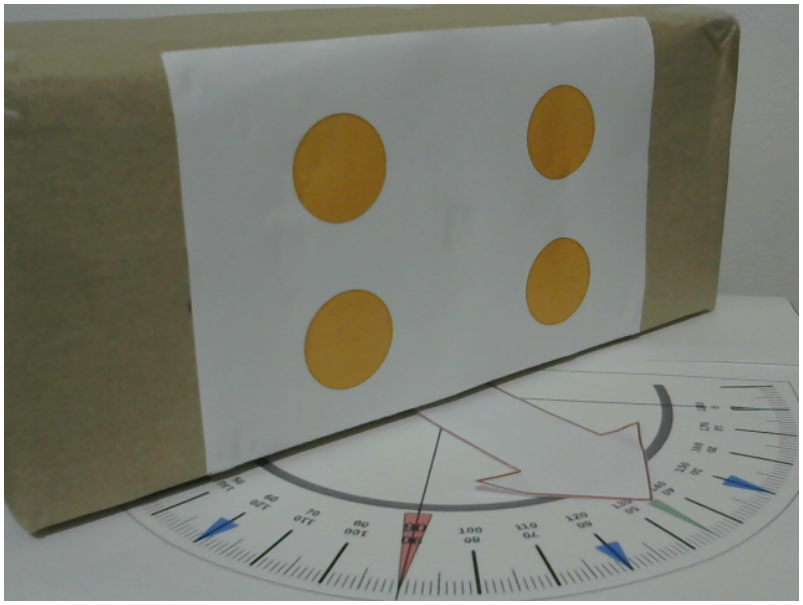


Figure 4.8: Original image of an object with markers, oriented in a 45° angle.

Step 1 - HSV conversion: In this step, each recorded image frame is converted from the traditional RGB color space to the HSV space, which represents the image information in three channels: Hue, Saturation and Value. The HSV representation is more suitable to perform color separation (Cheng et al., 2001).

The Hue channel is defined by the dominant wavelength of each color. The Value channel refers to the lightness/darkness of a color and it also indicates the quantity of light reflected. The saturation channel refers to the brilliance and intensity of a color.

Step 2 - Mask creation and bitwise conjuction: By selecting limited ranges for each channel, a specific color can be isolated. The Hue channel is more sensible to color identification. In order to identify the orange color of the markers used here, a Hue range $13 - 26$ was defined experimentally.

In the algorithm, first a HSV mask is created using the desired range. Then, a bitwise conjunction is performed between the original image array and the HSV mask. Figure 4.9 shows the result of this operation.

Figure 4.9: Result of bitwise operation with HSV mask, applied to the original image.

Step 3 - Gaussian Filtering: The image obtained from the bitwise conjunction is usually intensely affected by noise. Because of that, a Gaussian Filter is applied to smooth the borders and attenuate noise before the contours identification. The result is shown in Figure 4.10. This filtering is also important to prevent outliers in the next steps.

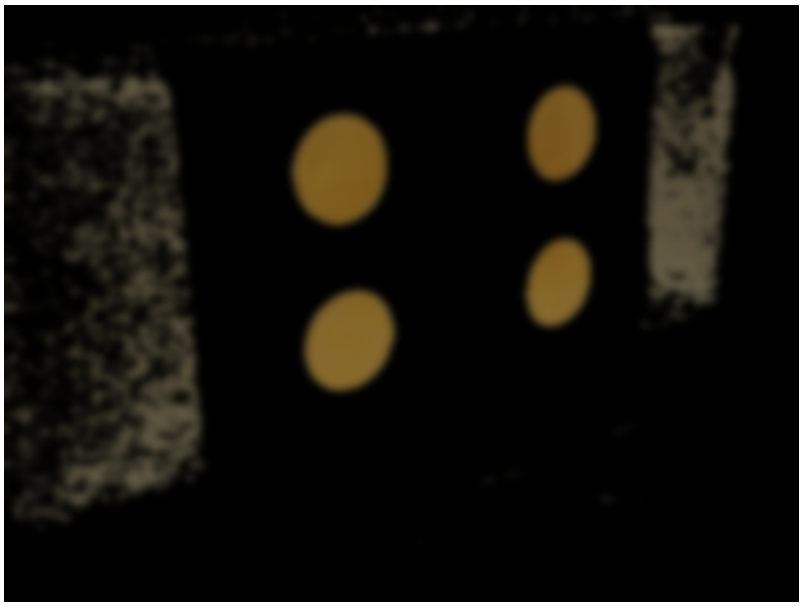


Figure 4.10: Smoother image after the Gaussian filter.

Step 4 - Canny Filtering: In this step, the objective is to remove all visual information from the image that does not belong to the markers contours, using the well-known Canny filter (Canny, 1986).

This filter aims to find the intensity gradient of the image, applying a pair of vertical and horizontal gradient convolution masks. Then, the intensity and direction of the 2D gradient is calculated. After, it determines whether a pixel belongs to the contour, performing a non-maximum suppression and hysteresis thresholding method. The lower and upper values of the gray scale thresholds are chosen as $\mu_1 = 100\ pixel$ and $\mu_2 = 200\ pixel$ respectively, for a robust contour detection. The result is presented in figure 4.11. As it can be seen, the image still presents some outliers, however none in a circular format.

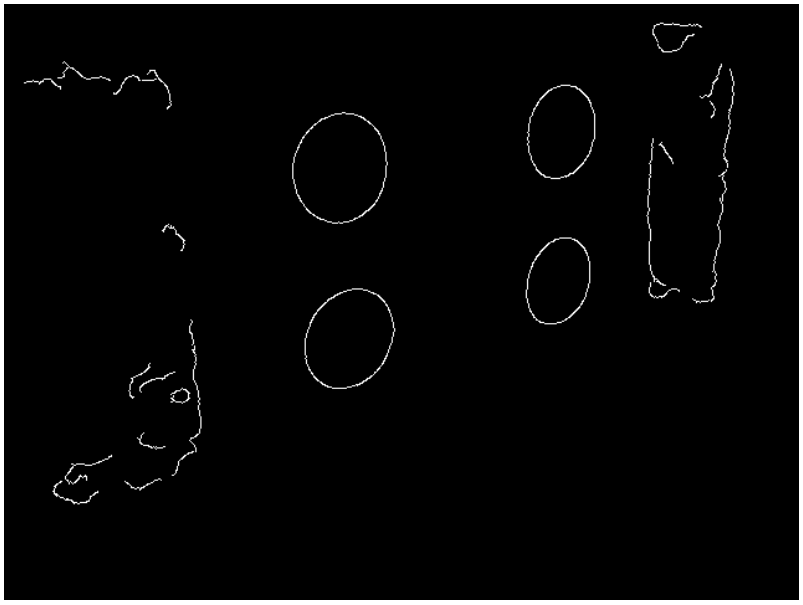


Figure 4.11: Contours detection.

Step 5 - Contour Selection and Centroid Coordinates Estimation: In order to find the centroid coordinates for each marker in the image, a range of feasible radius and minimum values for the solidity condition is defined. The solidity condition is defined as the ratio of contour area to its convex hull area, where a value close to the unity implies in a circular edge. However, in general, the detection algorithm finds more than one contour representing the same circle, which generates a number of candidate centroid coordinates. To overcome this multiple solutions problem, we propose to calculate the mean value of the coordinates for all centroids of the same circle in order to determine only one centroid for each. The final result is presented in Figure 4.12, where the markers coordinates are illustrated.

4.2.3 Pose Estimation Algorithms

After image processing, the image coordinates are obtained and the P4P problem needs to be solved. The process is illustrated in figure 4.13.

In general, the algorithms PnP problems require as input information:

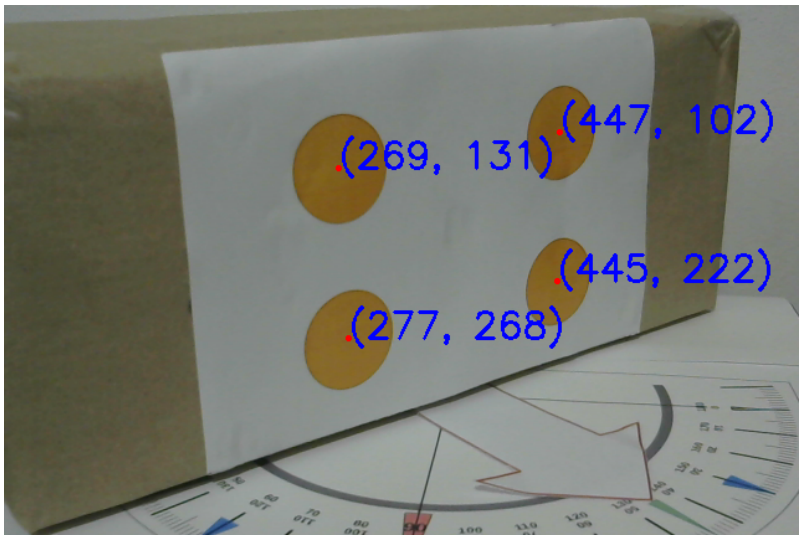


Figure 4.12: Markers coordinates are successfully obtained.

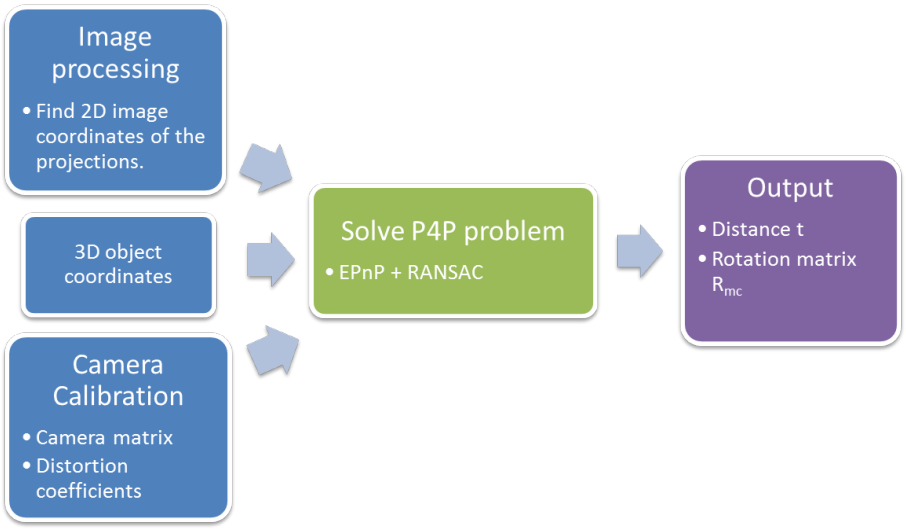


Figure 4.13: Input/Output model for the P4P problem.

1. the set of object points,
2. their corresponding image projections,
3. the camera matrix,
4. the distortion coefficients,

providing as output:

1. the rotation matrix R_{mc} and
2. the distance vector r_{mc} between the camera and the markers.

The camera matrix and the distortion coefficients are obtained in the camera calibration process.

In this work, the P4P problem is solved using an iterative method based on Levenberg-Marquardt optimization (Marquardt, 1963) and available in OpenCV library. In this method, the algorithm finds a pose that minimizes the reprojection error, which is the sum of squared distances between the image coordinates and the object coordinates. The image output is presented in Figure 4.14, where the axis are plotted for easy visualization.

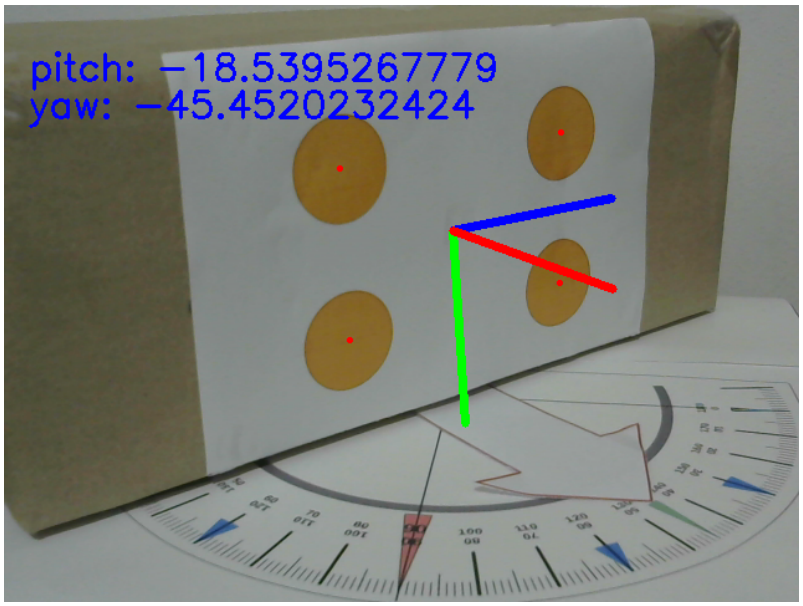


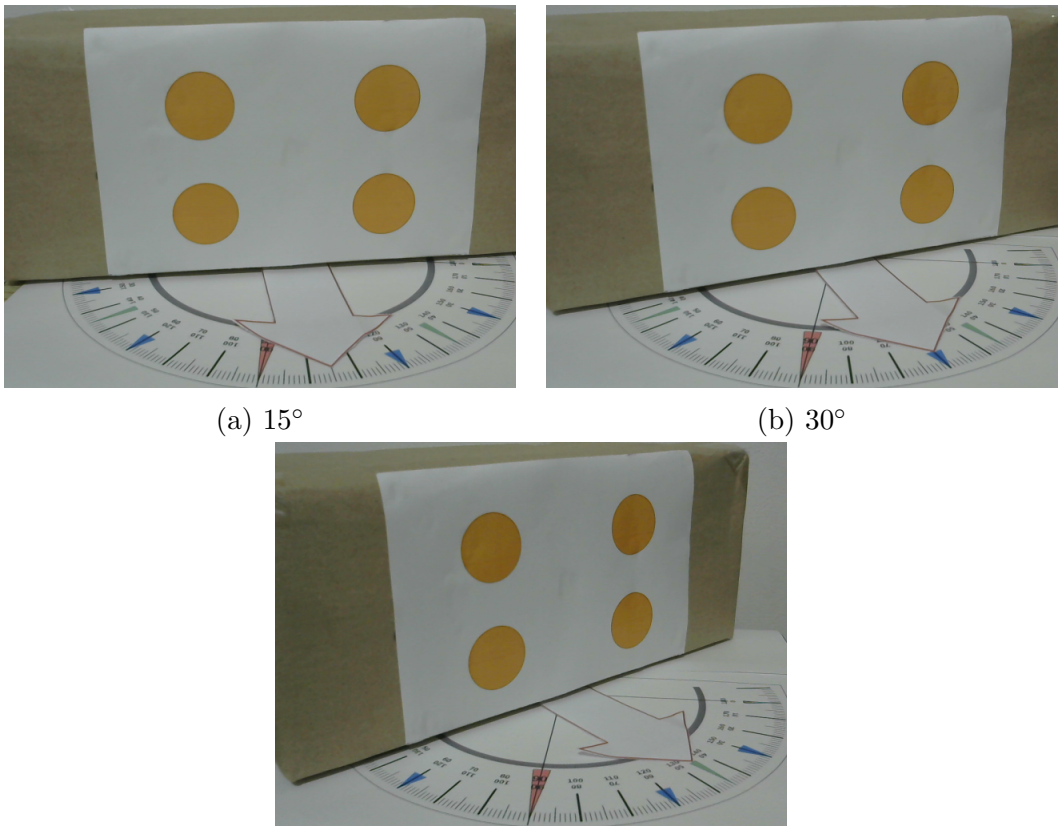
Figure 4.14: Output image and TAG frame.

From the rotation matrix R_{mc}, the vector $\vec{h}_o$, normal to the TAG plane and representing the direction of the object, is obtained. Also, let α be the *yaw* angle, that is, the direction the object is pointing to, with respect to the robot base. The yaw angle can be calculated as $\alpha = \arccos(\vec{h}_o \cdot \vec{x}_b) - \pi/2$ and is useful in the manipulators alignment process.

4.2.4 Experiments for Validation

In this subsection, the experiment for validation of the pose estimation methodology is described and its results are presented.

For illustration, the object is disposed in three different configurations, using a protractor for better precision (Figure 4.15). Next, the algorithm is run: for each object pose, 100 images are taken and its correspondent estimations are obtained. The results are displayed in (Figure 4.16). Also, the statistics for the tests are shown in Table 4.2.



(a) 15° (b) 30°

(c) 45°

Figure 4.15: Vision Experiment: object different configurations.

Table 4.2: Statistics of visual identification experiments.

	15°	30°	45°
mean	15.80	30.43	45.61
std. dev	0.93	0.71	0.59
max	18.17	32.03	47.33
min	13.78	28.75	44.31

4.2.5 Alignment of the Manipulators

After the visual identification, a planning algorithm uses the information obtained to drive the manipulators to the markers position.

The problem of alignment consists in determine a proper position for the manipulators before the grasping and to align them with the direction of the object $\vec{h}_o$ (see Figure 4.17).

Assume that the object lies in the plane x_b-y_b of coordinate system of the base $\mathcal{F}_b$ (pitch = 0°) and its initial pose is uncertain. Let $p_{c,1}$ and $p_{c,2} \in \mathbb{R}^3$ be the positions the cameras attached to the left and right end effectors, respectively. Also, consider $p_{m,1}$ and $p_{m,2} \in \mathbb{R}^3$ the positions of the visual markers fixed on the left side and the right side of the object, respectively.

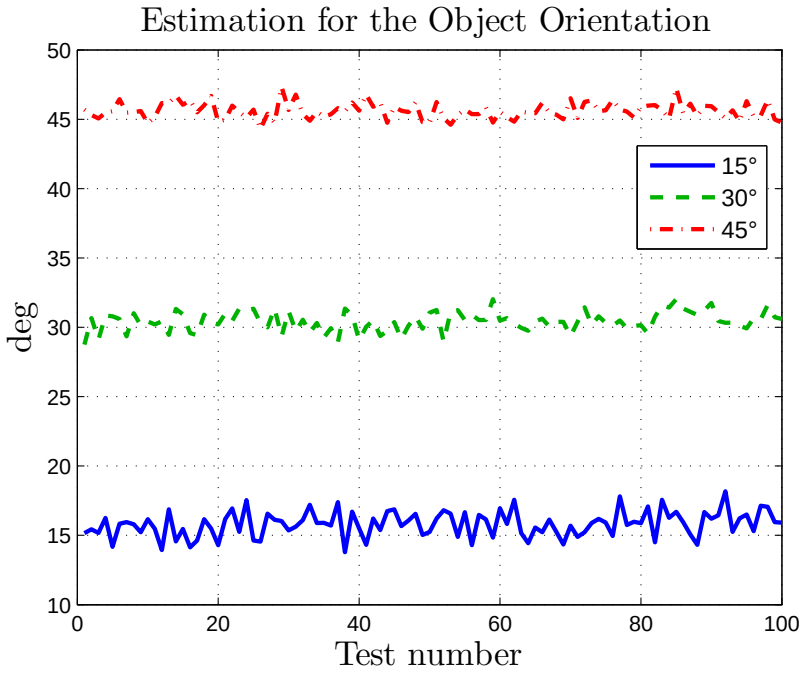


Figure 4.16: Results for the vision experiment.

The position of the cameras with respect to the end effectors $p_{ec,\,i}$ is constant and known: therefore, the camera position with respect to the robot base $p_{c,\,i}$ can be obtained using Forward Kinematics. Besides, the relative distance of the each marker to the correspondent camera $r_{cm,\,i}$ is obtained from the visual estimation algorithm. Hence, the position of each marker $p_{m,\,i}$ with respect to the base is:

$$p_{m,\,i} = p_{c,\,i} + r_{cm,\,i}, \qquad i = \{1, 2\}\,. \tag{4.1}$$

The markers position is chosen as the target contact point for the grasp. Additionally, the manipulators should be positioned closer to the contact points keeping a distance $d \in \mathbb{R}$, experimentally defined, from them. Therefore, the desired positions for each end effector $p_{d,\,i}$, before the grasp, are calculated as:

$$p_{d,\,i} = p_{m,\,i} + d\,h_o, \qquad i = \{1, 2\}\,. \tag{4.2}$$

It is worth mentioning that if an estimative of the object width l is available, the alignment can be performed using just one camera. In this case, the position of the other marker is estimated as $p_{m,\,2} = p_{m,\,1} + l\,h_o$.

If the end effectors are aligned in the initial direction $\vec{y}_b$, with orientation $R_{0,\,i}$, then the desired orientation for the correct alignment with the object can be obtained by a rotation of α (the yaw angle) around axis $\vec{x}_e$:

$$R_{d,\,i} = R_x(\alpha)\,R_{0,\,i} \qquad i = \{1, 2\}\,. \tag{4.3}$$

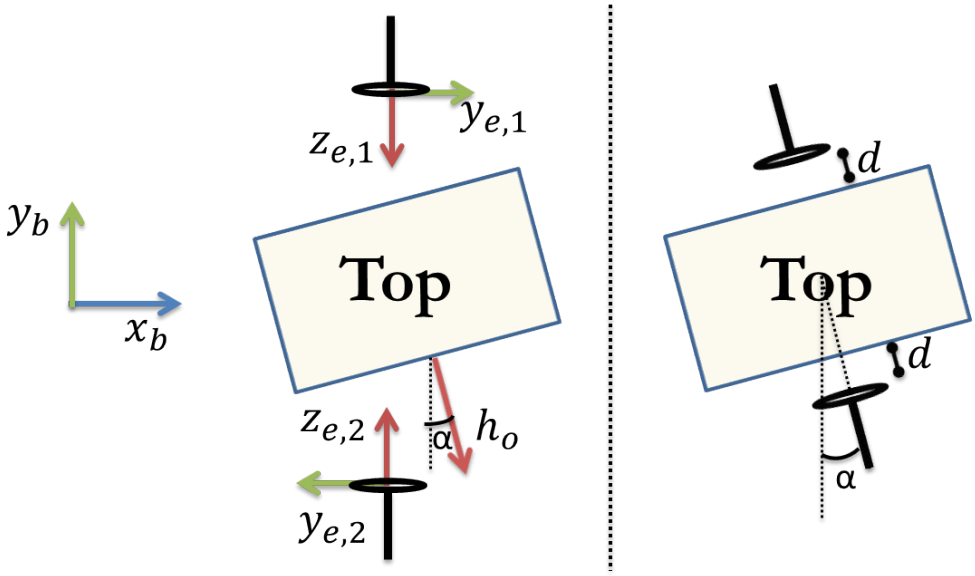


Figure 4.17: Alignment problem.

Finally, after this initial positioning procedure, force control is applied in the direction of the unit vector $\vec{h}_o$ until a minimum contact force f_{min} is reached.

4.3 Experiments and Results

In this section, the experimental results for the Cooperative Hybrid Relative Position-Force control (Section 3.3.5), obtained with the Baxter robot, are presented. Initially, the grasp is performed using the vision methodology described. The results presented in this section are obtained after the hybrid control is enabled.

In order to illustrate the control performance, two different planned trajectories for the end-effectors absolute position are specified. First, the desired trajectory is a circle defined in plane YZ. In the sequence, the next desired trajectory is a circle defined in plane XY. Both circular trajectories have radius $r_c = 0,2\,m$ and angular frequency $\omega_c = \frac{\pi}{5}\,rad\,s^{-1}$.

While the position control loop tracks the absolute position and keeps the relative position null, the force control loop regulates the contact force, that is, the relative force, applied by the manipulators, to 10 N. The contact force is estimated using the joint torque measurements. Also, the absolute force *setpoint* is set to 0 N, since both manipulators should apply the same force.

The experiments were design to avoid kinematic singularities, then the trajectories do not lead the manipulators to the limit of the robot workspace.

The control parameters utilized in the experiments were determined by the simulation presented in Section 3.3.6 and are presented at table 4.3.

In the experiments presented in this section, we have used a local computer to

communicate with the robot. Another possibility is to create a teleoperated system. For further details, a proposed system architecture for teleoperation is described in Appendix B.

Table 4.3: Experimental control parameters for cooperative hybrid position-force control.

Parameter	Value
K_{pa}	1,5 s^{-1}
K_{pr}	2,25 s^{-1}
K_{oa}	1,5 $rad\,s^{-1}$
K_{or}	2,25 $rad\,s^{-1}$
K_{fr}	2,0 $m\,N^{-1}s^{-1}$

Tracking a circular reference - plane YZ

The experimental results described here include the data obtained before the change of reference, which occurs around 10 s.

Figure 4.18 (a) describes the absolute position and reference trajectories. The trajectory is smooth and it tracks the reference. Figure 4.18 (b) illustrates the absolute position error. After the transient, the absolute position error converges asymptotically to a region around 10 mm. The result is acceptable considering that the joints flexibility effects were not modeled in the control scheme and also considering that this flexibility degrades the accuracy of position control (De Luca and Book, 2008).

The relative position and the reference is shown in Figure 4.19 (a) and the error is also shown in Figure 4.19 (b). The position-controlled directions $p_{r,\,x}$ and $p_{r,y}$ of the relative position converges to a region around zero, while the force-controlled direction $p_{r,\,z}$ converges to a constant distance. Indeed, this is the expected considering the distance depends on the force control action.

The absolute and relative orientation errors are illustrated in Figure 4.20. Both absolute and relative orientation error presents a small and acceptable oscillation around 0.02 rad.

The filtered estimated force applied to the object and the force error are presented in Figure 4.21. The force reaches the reference (10 N) and the error presents relatively big oscillation (2 N) around zero. This occurs probably due to the presence of flexibility on the object and expected errors in the force estimation method, which may be originated from noise, friction and vibration as discussed in section 2.3.1. However, in the experiments, the error has not influenced in the manipulation task and the object has not lost contact. It is worth noting that the force

control performance tends to degrade and oscillate more when the object stiffness is decreased. Then, better performance would be expected for a rigid object.

The Cartesian control signal is demonstrated in Figure 4.22. It can be noted that the control signals are smooth and bounded. The joint velocities are also smooth and do not require any peak or unfeasible velocities from the actuators, as can be seen in Figure 4.23 and Figure 4.24.

The representation of the robot arms and the trajectory performed by the object (frame $\mathcal{F}_a$) is shown in Figure 4.25. It can be noted that, after the transient, the trajectory tracks the reference with acceptable performance.

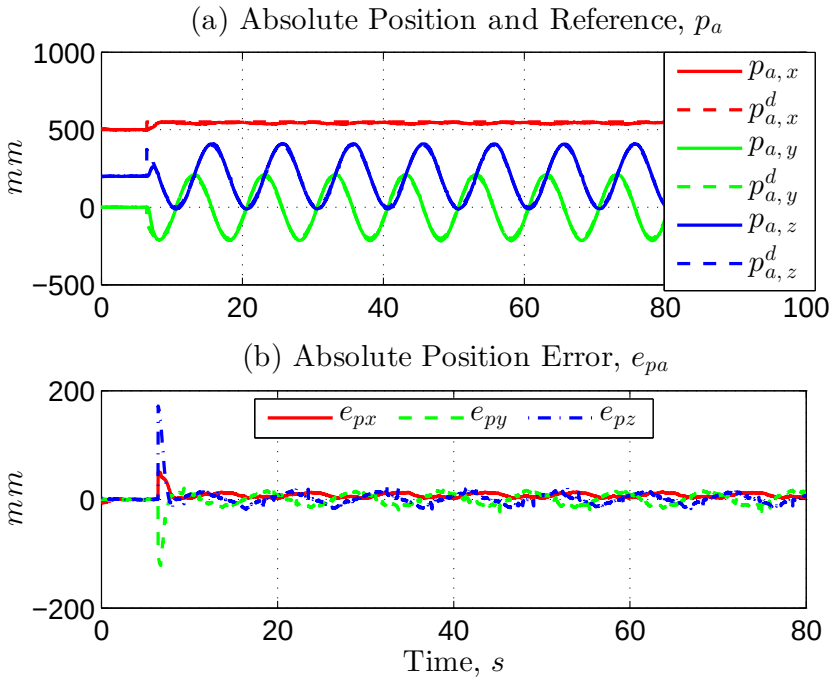


Figure 4.18: Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: (a) absolute position and reference (b) absolute position error.

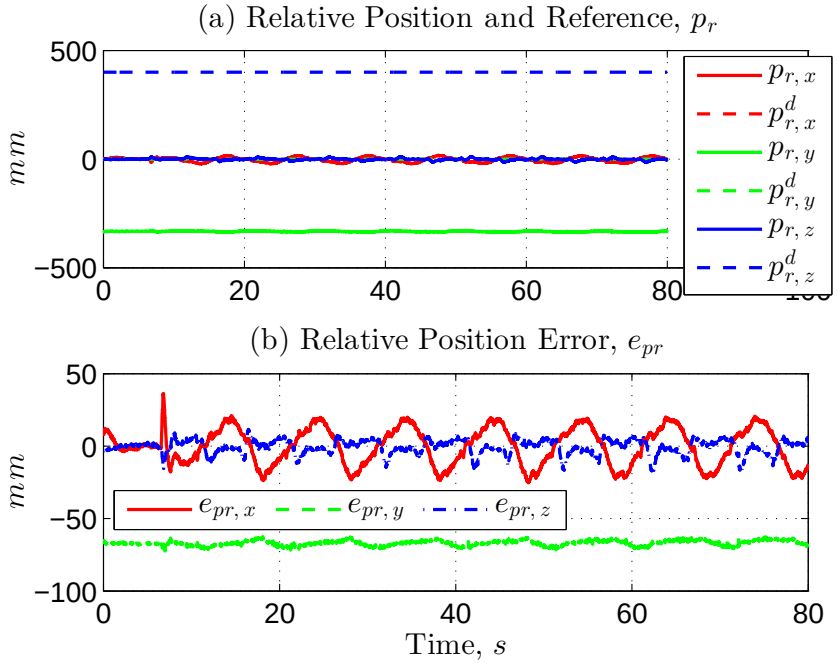


Figure 4.19: Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: (a) relative position and reference (b) relative position error.

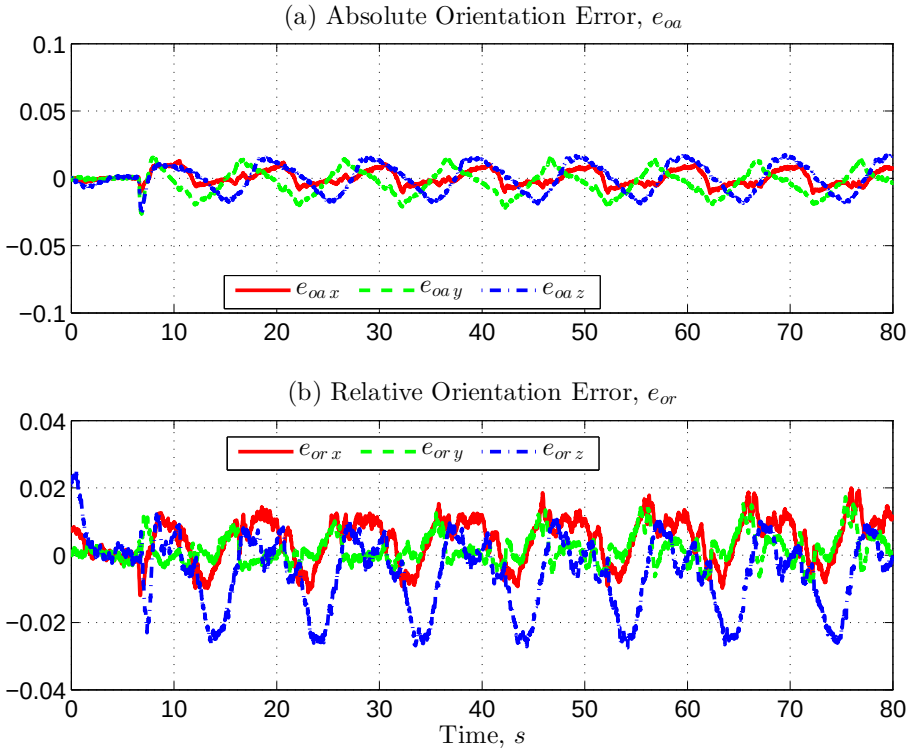


Figure 4.20: Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: (a) absolute orientation error (b) relative orientation error.

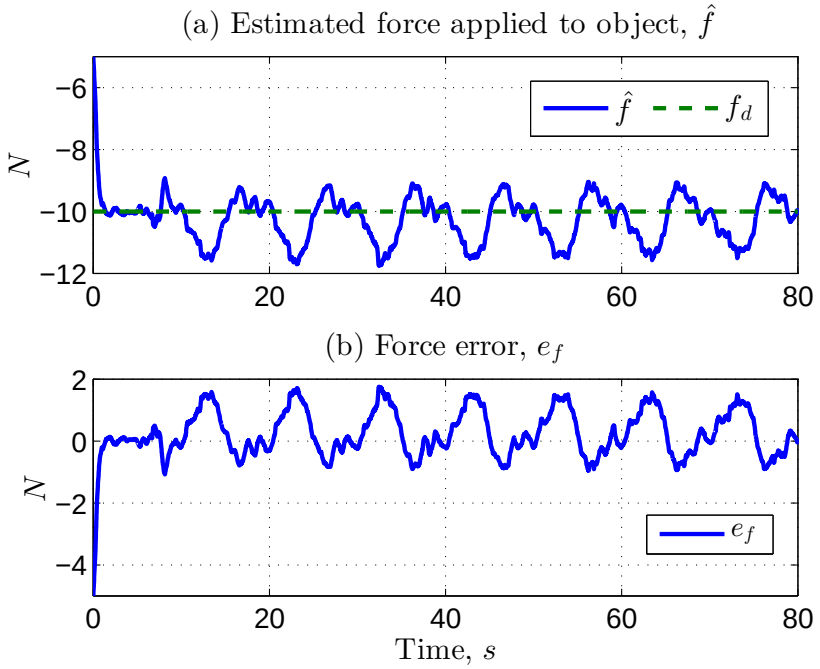


Figure 4.21: Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: (a) force applied to object (b) force error.

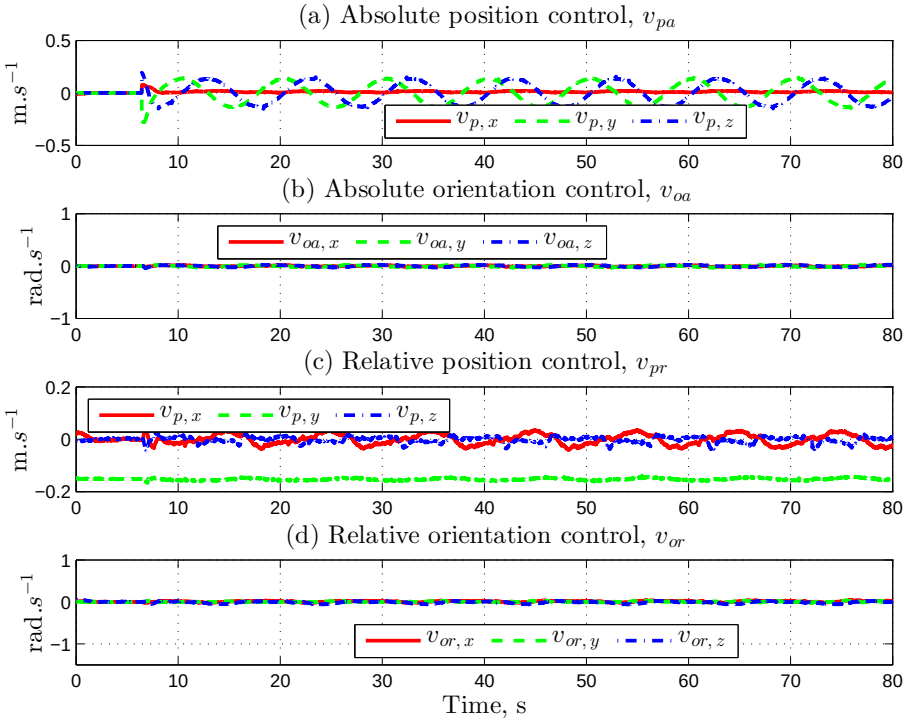


Figure 4.22: Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: (a) absolute position control (b) relative orientation control (c) relative position control (d) relative orientation control.

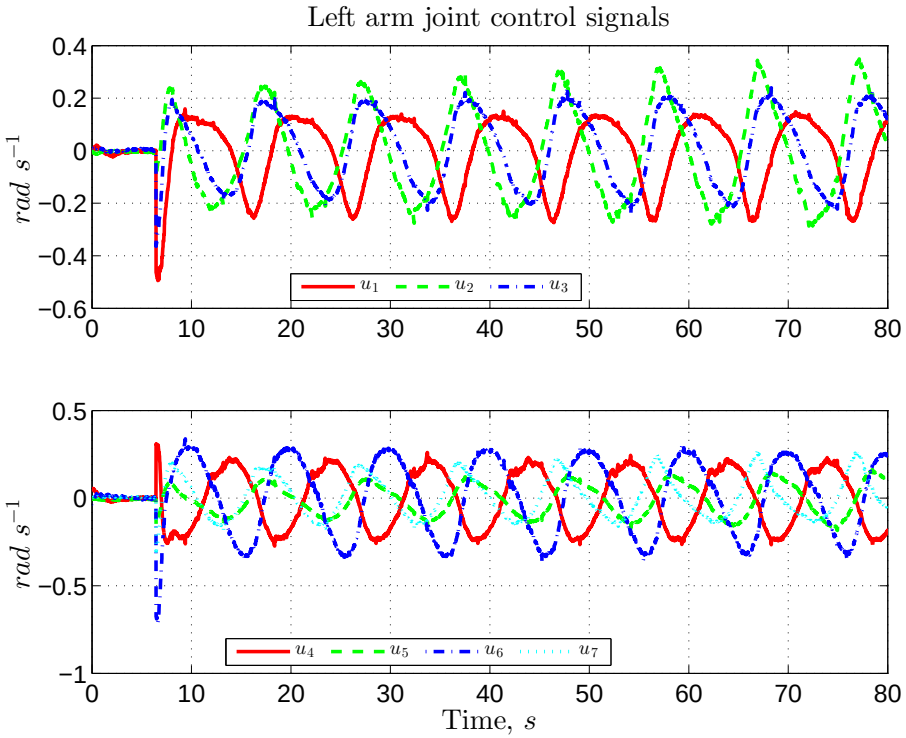


Figure 4.23: Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: left arm joint control signals.

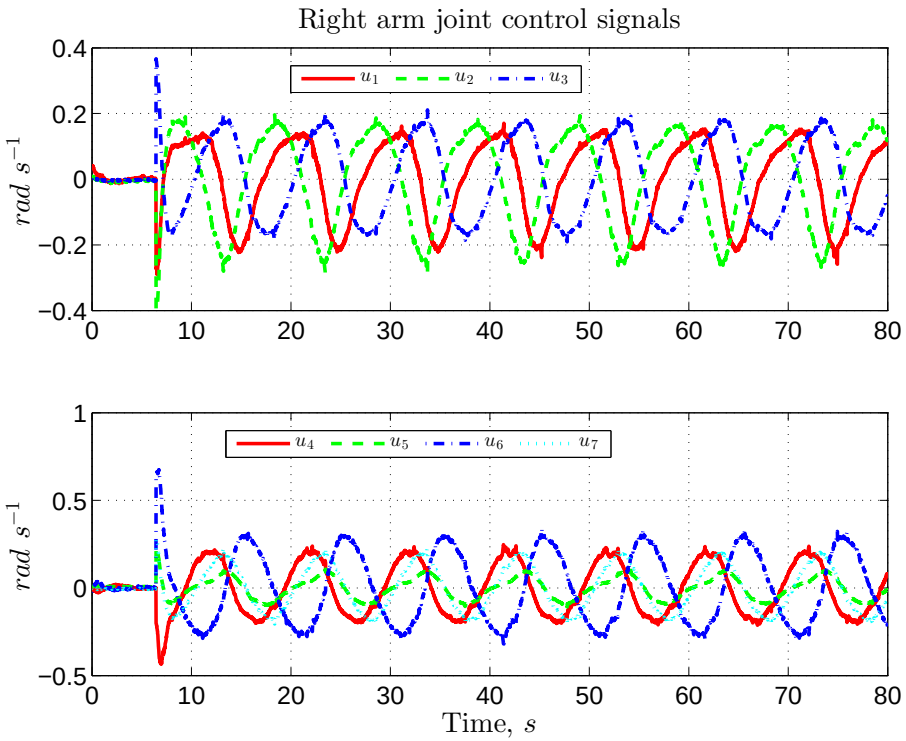


Figure 4.24: Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: right arm joint control signals.

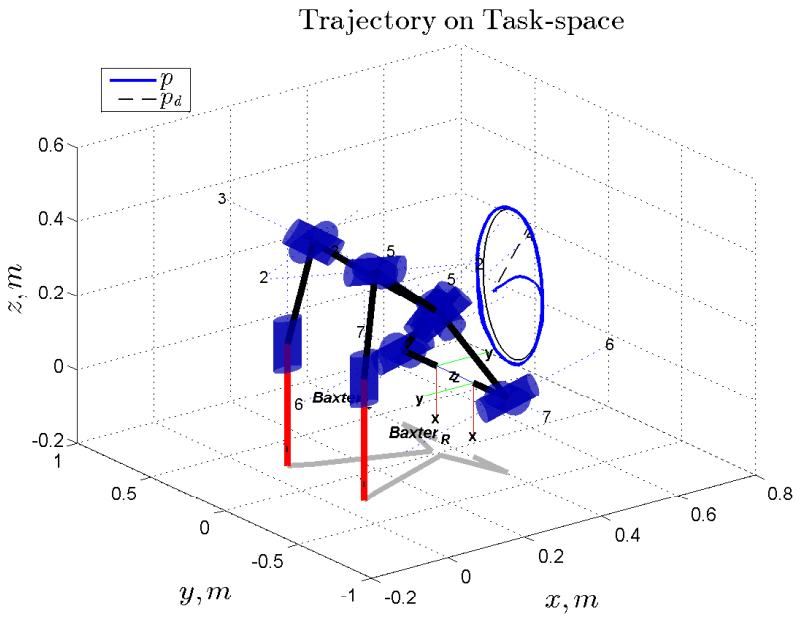


Figure 4.25: Experimental Results - Hybrid Relative Position-Force Control tracking YZ plane: trajectory performed by manipulators (with reference).

Tracking a circular reference - plane XY

The experiments performed here are similar to the last subsection, but the absolute position reference is modified to track a trajectory in XY plane, with respect to the robot base.

Figure 4.26 (a) describes the absolute position and reference trajectories and Figure 4.26 (b) illustrates the absolute position error. The relative position and the reference is shown in Figure 4.27 (a) and the error is also shown in Figure 4.27 (b). The absolute and relative orientation errors are illustrated in Figure 4.28. The filtered estimated force applied to the object and the force error are presented in Figure 4.29. The Cartesian control signal is demonstrated in Figure 4.30. The joint velocities are illustrated in Figure 4.31 and Figure 4.32. The representation of the robot arms and the trajectory performed by the object (frame $\mathcal{F}_a$) is shown in Figure 4.33.

The results show that the manipulation task can also be performed in the XY plane with similar performance. As expected, the absolute position variable is responsible for the object positioning and it is able to modify the position without any restriction, even for movements occurring in the force-controlled direction, since the force control only acts in the relative position variable.

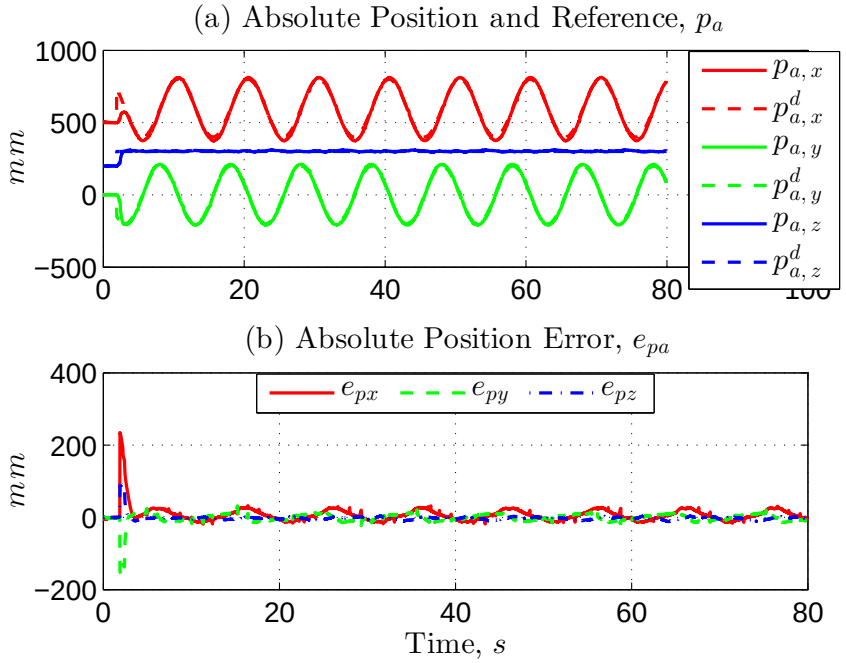


Figure 4.26: Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: (a) absolute position and reference (b) absolute position error.

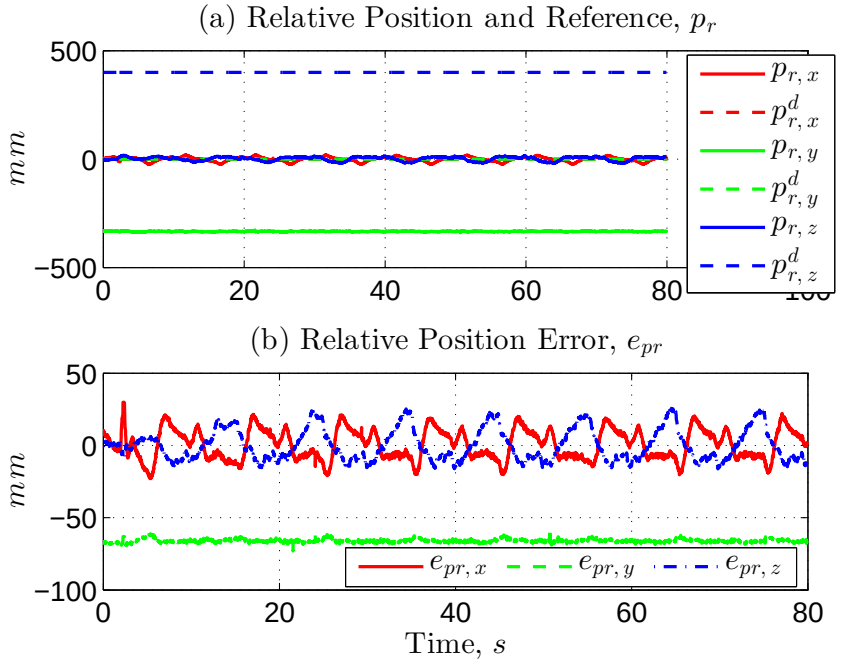


Figure 4.27: Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: (a) relative position and reference (b) relative position error.

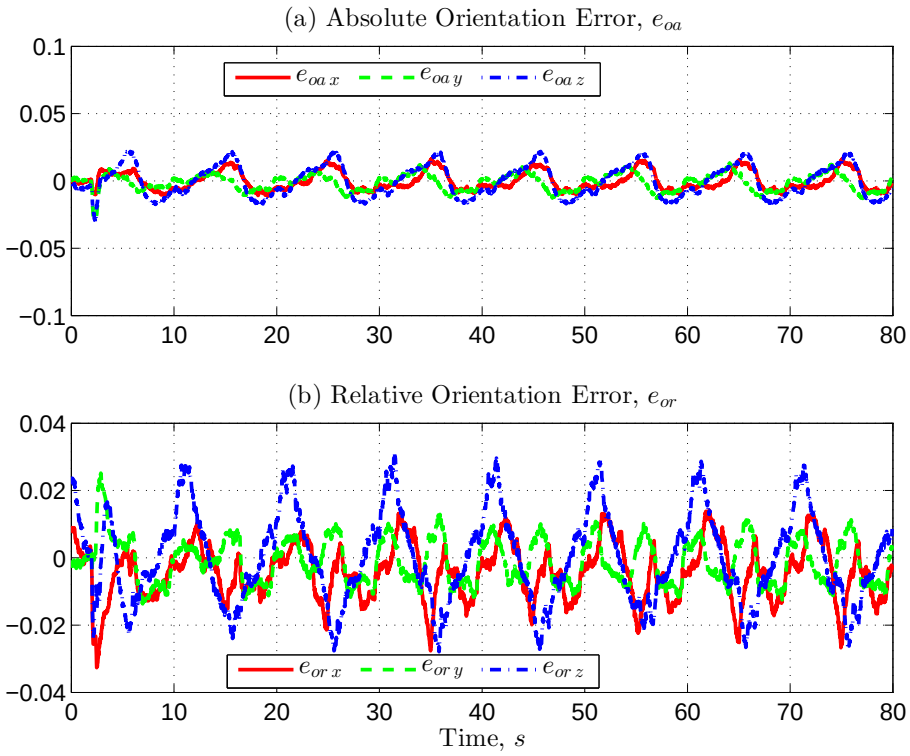


Figure 4.28: Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: (a) absolute orientation error (b) relative orientation error.

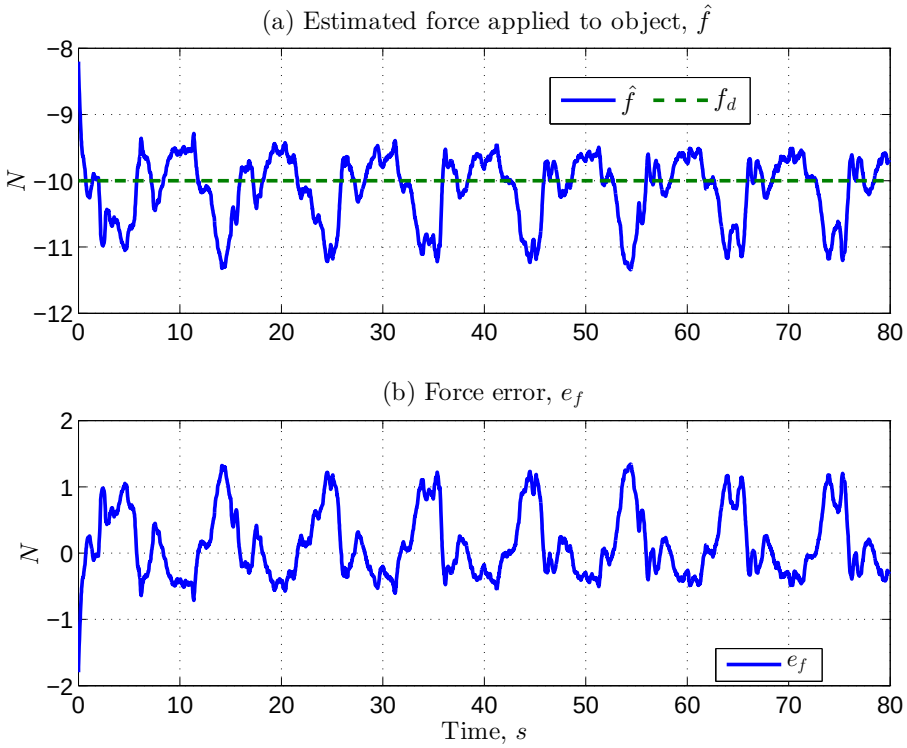


Figure 4.29: Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: (a) force applied to object (b) force error.

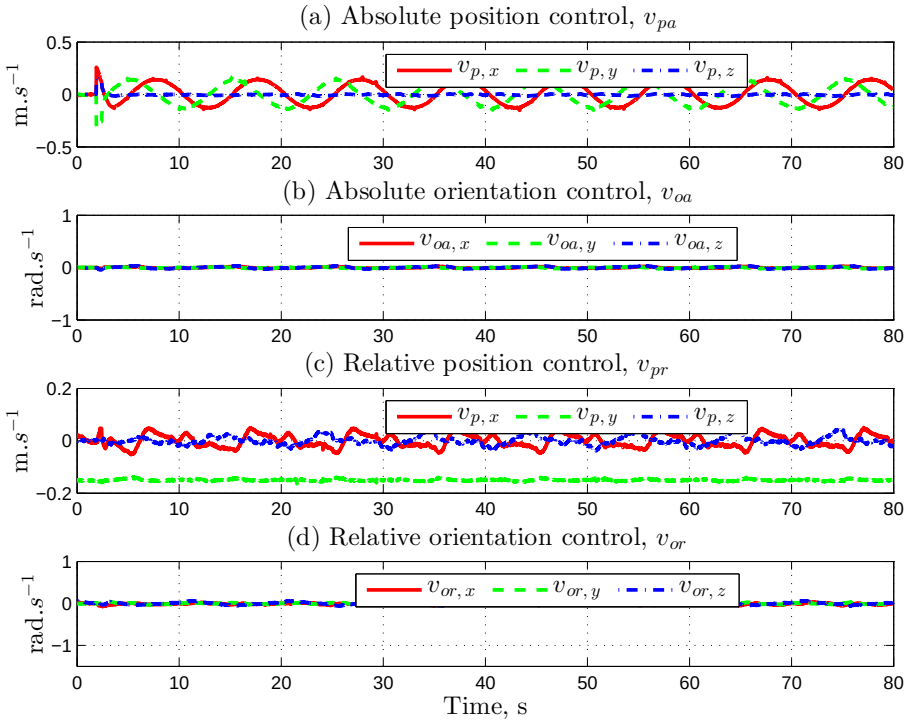


Figure 4.30: Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: (a) absolute position control (b) relative orientation control (c) relative position control (d) relative orientation control.

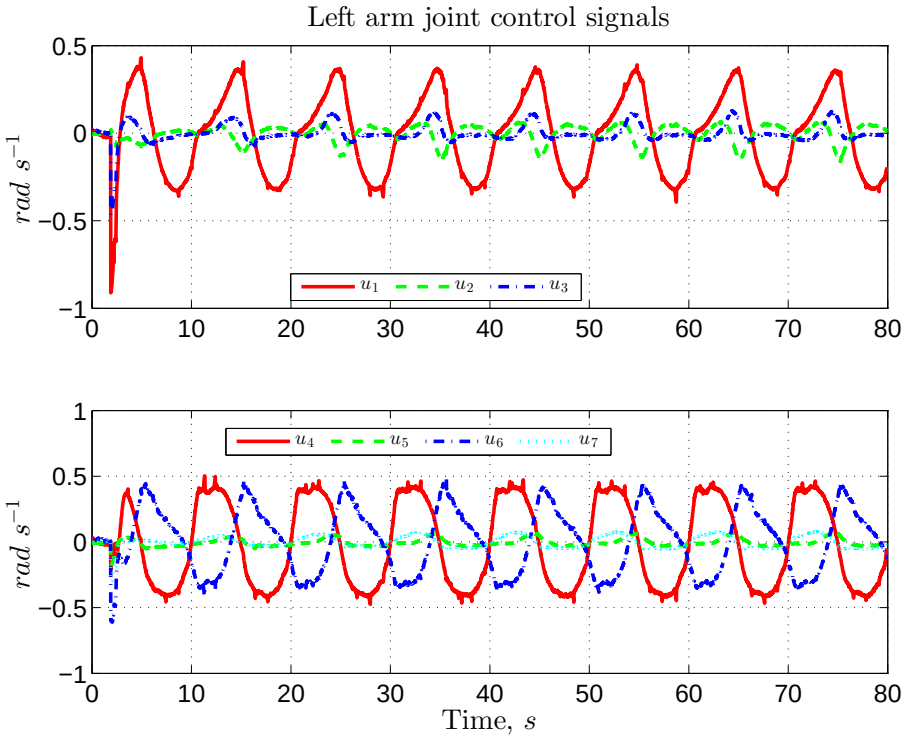


Figure 4.31: Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: left arm joint control signals.

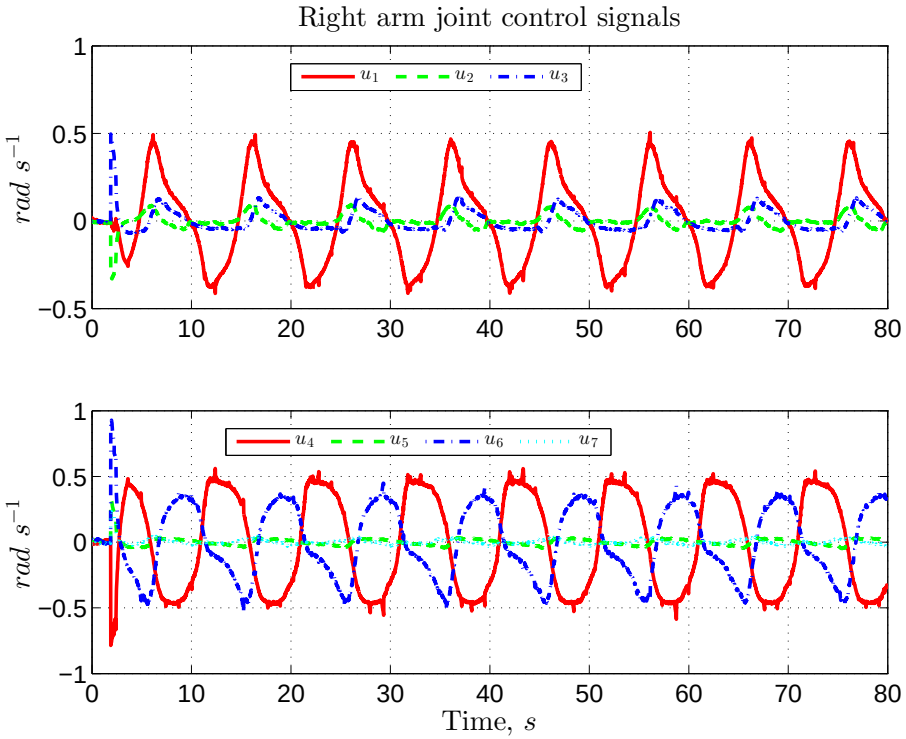


Figure 4.32: Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: right arm joint control signals.

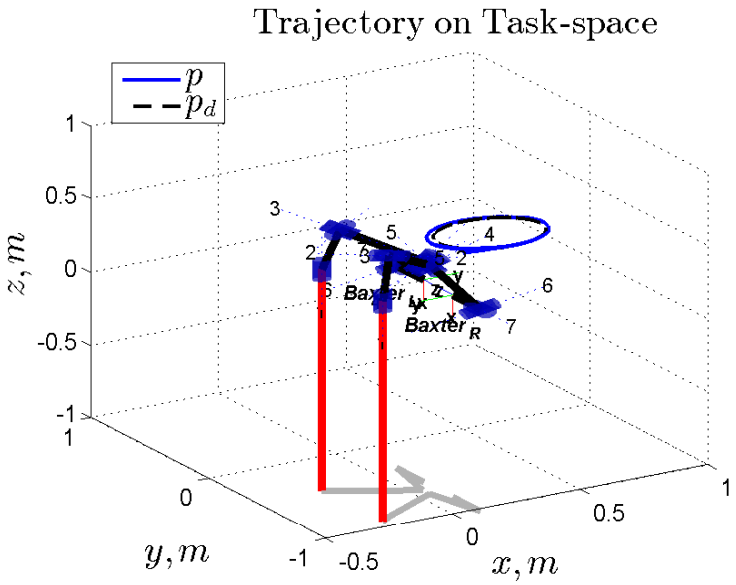


Figure 4.33: Experimental Results - Hybrid Relative Position-Force Control tracking XY plane: trajectory performed by manipulators (with reference).

4.4 Conclusion

In this chapter, an experimental setup is proposed and verified. The problem formulation is explained and information about the robotic system used in the experiments is presented (Section 4.1.1).

A methodology for autonomous grasp based on visual information is developed and explained. The methodology requires a monocular camera, which offers good cost/performance to solve the P4P problem. The experiment performed using an object with visual markers illustrate the feasibility of the method to estimate the pose of objects.

After the grasp closure is performed, the cooperative control is activated. Then, the experimental results, using the Baxter robot, are obtained for the proposed cooperative hybrid scheme and confirm the feasibility and effectiveness of the method.

Finally, the results show, as expected, that the manipulation task can be performed with acceptable performance even for movements occurring in the force-controlled direction, since the force control only acts in the relative position variable.

Chapter 5

Concluding Remarks and Perspectives

5.1 Conclusions

In this work, a hybrid position-force kinematic control scheme for cooperative manipulation was proposed, based on the hybrid kinematic control scheme developed in Leite et al. (2010) for single manipulators. The scheme proposed here considered the problem of bi-manual cooperative manipulation, using the hybrid position-force method, which is a direct force control method, that is, the force is controlled via a separate (explicit) force loop. This strategy allows direct specification of a desired force value.

Additionally, the orientation control is based on the unit quaternion representation, which is free from singularities and computationally more efficient. The stability analysis was presented, considering the position, force and orientation loops, as part of the hybrid control scheme. The analysis uses the Lyapunov stability theory and it takes advantage of the complementary orthogonal spaces, which allows an independent design of the control loops.

The robotic system for cooperative manipulation is modeled based on the *cooperative task space formulation*, proposed in (Chiacchio et al., 1996; Caccavale et al., 2000). This formulation defines the cooperative task, in terms of absolute and relative variables, which can be computed from the end-effector pose.

With the objective of not using force sensors, which are expensive and require customization of the end effector, the contact forces acting at the end effectors are estimated based on the joint torques information, which are measured using built-in sensors positioned in each robot joint. The results confirm the feasibility of the proposed estimation for object manipulation. However, for tasks that require a high accurate force objective, it should be considered installing force/torque sensors at

the end effector.

In the experimental setup, the proposed manipulation task requires the capability of the robotic system to identify the pose of an object. Based on the ideas proposed in Faria et al. (2015), an alternative procedure for visual estimation of the 3D pose of objects using a single monocular camera is developed. In this work, colored visual markers are used to determine the pose of an object located in the robot workspace.

In the sequence, a grasping algorithm, which allows the robot arms to grasp the object autonomously, is proposed. This algorithm can be used, for example, during teleoperation to assist the operator, which can focus on other part of the process.

The feasibility and effectiveness of the proposed control method is verified through numerical simulation (Matlab) and experiments in a real robotic system (Baxter robot), which uses the Robot Operating System (ROS). The experimental results obtained were presented and certify the performance of the cooperative hybrid control algorithm proposed.

5.2 Future Work

In order to continue the development and to promote the research discussed in this work, some suggestions of future work are presented:

- Implement the proposed method in a robotic system endowed with multiple manipulators;
- Install force sensors at each end effector and compare the performance with the results presented in this work;
- Include the joint flexibility dynamic modeling, considering the cooperative description of the system. If the joint flexibility effect is considered in the modeling, the system order becomes higher than usual, which increases the complexity of the control design;
- Include an external camera to measure and obtain an accurate estimate of the object pose. The camera or eventually multi-camera system would be included in the environment where the manipulation is to be performed;
- Investigate the feasibility to use other types of fiducial markers. It is important to increase the robustness of the pose estimation algorithm to image disturbances caused by the usual presence of smoke and steam in many industrial environments;
- Implement an obstacle avoidance and detection algorithm enabling the robot to operate in unstructured environments.

Bibliography

Abdel-Aziz, Y., Karara, H. and Hauck, M. (2015), 'Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry', *Photogrammetric Engineering & Remote Sensing* **81**(2), 103–107.

Adorno, B. V. (2011), Two-arm manipulation: From manipulators to enhanced human-robot collaboration, PhD thesis, Université Montpellier II-Sciences et Techniques du Languedoc.

Adorno, B. V., Fraisse, P. and Druon, S. (2010), Dual position control strategies using the cooperative dual task-space framework, *in* 'Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on', IEEE, pp. 3955–3960.

Aghili, F. (2011), Self-tuning cooperative control of manipulators with position/orientation uncertainties in the closed-kinematic loop, *in* '2011 IEEE/RSJ International Conference on Intelligent Robots and Systems', IEEE, pp. 4187–4193.

Bonitz, R. G. and Hsia, T. C. (1996), 'Internal force-based impedance control for cooperating manipulators', *Robotics and Automation, IEEE Transactions on* **12**(1), 78–89.

Bonitz, R. and Hsia, T. (1994), 'Force decomposition in cooperating manipulators using the theory of metric spaces and generalized inverses', *Proceedings of the 1994 IEEE International Conference on Robotics and Automation* .

Bose, C. B. and Amir, I. (1990), 'Design of fiducials for accurate registration using machine vision', *IEEE Transactions on Pattern Analysis & Machine Intelligence* (12), 1196–1200.

Caccavale, F., Chiacchio, P. and Chiaverini, S. (2000), 'Task-space regulation of cooperative manipulators', *Automatica* **36**(6), 879–887.

Caccavale, F., Chiacchio, P., Marino, A. and Villani, L. (2008), 'Six-dof impedance control of dual-arm cooperative manipulators', *IEEE/ASME Trans. Mech.* **13**(5), 576–586.

Caccavale, F., Lippiello, V., Muscio, G., Pierri, F., Ruggiero, F. and Villani, L. (2013), 'Grasp planning and parallel control of a redundant dual-arm/hand manipulation system', *Robotica* **31**(07), 1169–1194.

Caccavale, F. and Uchiyama, M. (2008), Cooperative manipulators, *in* B. Siciliano and O. Khatib, eds, 'Springer Handbook of Robotics', Springer Berlin Heidelberg, pp. 701–718.

Canny, J. (1986), 'A computational approach to edge detection', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (6), 679–698.

Cheng, H.-D., Jiang, X., Sun, Y. and Wang, J. (2001), 'Color image segmentation: advances and prospects', *Pattern recognition* **34**(12), 2259–2281.

Chiacchio, P., Chiaverini, S., Sciavicco, L. and Siciliano, B. (1991), 'Global task space manipulability ellipsoids for multiple-arm systems', *Robotics and Automation, IEEE Transactions on* **7**(5), 678–685.

Chiacchio, P., Chiaverini, S. and Siciliano, B. (1993), Task-oriented kinematic control of two cooperative 6-dof manipulators, *in* 'American Control Conference, 1993', IEEE, pp. 336–340.

Chiacchio, P., Chiaverini, S. and Siciliano, B. (1996), 'Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system', *Journal of Dynamic Systems, Measurement, and Control* **118**(4), 691–697.

Chiaverini, S. and Sciavicco, L. (1989), 'Force/position control of manipulators in task space with dominance in force', *Proc. IFAC Rob. Cont* pp. 137–143.

Chiaverini, S. and Sciavicco, L. (1993), 'The parallel approach to force/position control of robotic manipulators', *IEEE Trans. Rob. Aut.* **9**(4), 361–373.

Chiaverini, S., Siciliano, B. and Villani, L. (1994), 'Force/position regulation of compliant robot manipulators', *IEEE Transactions on Automatic Control* **39**(3), 647–652.

Choi, J., Kang, S. et al. (2012), External force estimation using joint torque sensors for a robot manipulator, *in* 'Proc. IEEE Int. Conf. Rob. Aut.', pp. 4507–4512.

Collet, A., Berenson, D., Srinivasa, S. S. and Ferguson, D. (2009), Object recognition and full pose registration from a single image for robotic manipulation, *in* 'Robotics and Automation, 2009. ICRA'09. IEEE International Conference on', IEEE, pp. 48–55.

Dauchez, P., Delebarre, X., Bouffard, Y. and Degoulange, E. (1991), Task modeling and force control for a two-arm robot, *in* 'Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on', IEEE, pp. 1702–1707.

Dauchez, P., Fournier, A. and Jourdan, R. (1989), 'Hybrid control of a two-arm robot for complex tasks', *Robotics and Autonomous Systems* **5**(4), 323–332.

Dauchez, P. and Uchiyama, M. (1987), Kinematic formulation for two force-controlled cooperating robots, *in* 'ICAR Int. Conf. on Advanced Robotics, Versailles, France'.

De Luca, A. and Book, W. (2008), Robots with flexible elements, *in* B. Siciliano and O. Khatib, eds, 'Springer Handbook of Robotics', Springer Berlin Heidelberg, pp. 287–319.

De Santis, A., Siciliano, B., De Luca, A. and Bicchi, A. (2008), 'An atlas of physical human-robot interaction', *Mechanism and Machine Theory* **43**(3), 253–270.

Deo, A. S. and Walker, I. D. (1992), Robot subtask performance with singularity robustness using optimal damped least-squares, *in* 'Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on', IEEE, pp. 434–441.

Deo, A. S. and Walker, I. D. (1995), 'Overview of damped least-squares methods for inverse kinematics of robot manipulators', *Journal of Intelligent and Robotic Systems* **14**(1), 43–68.

Doulgeri, Z. and Karayiannidis, Y. (2007), Force/position tracking of a robot in compliant contact with unknown stiffness and surface kinematics, *in* 'Proc. IEEE Int. Conf. Rob. Aut.', pp. 4190–4195.

Duffy, J. (1990), 'The fallacy of modern hybrid control theory that is based on orthogonal complements of twist and wrench spaces', *Journal of Robotic Systems* **7**(2), 139–144.

Dyer, C. R. (2001), Volumetric scene reconstruction from multiple views, *in* 'Foundations of image understanding', Springer, pp. 469–489.

Enríquez, V. L. and Alejo, V. G. (2015), 'Hybrid force-position control for manipulators under transitions free to constrained motion', *International Journal of Mechanical Engineering and Robotics Research* **4**(4), 319.

Erhart, S. and Hirche, S. (2014), 'Model and analysis of the interaction dynamics in cooperative manipulation tasks'.

Erhart, S., Sieber, D. and Hirche, S. (2013), An impedance-based control architecture for multi-robot cooperative dual-arm mobile manipulation, *in* '2013 IEEE/RSJ International Conference on Intelligent Robots and Systems', IEEE, pp. 315–322.

Faria, R. O., Kucharczak, F., Freitas, G. M., Leite, A. C., Lizarralde, F., Galassi, M. and From, P. J. (2015), A methodology for autonomous robotic manipulation of valves using visual sensing, *in* 'Proc. 2nd IFAC Works. Aut. Cont. Offshore O&G Produc.', pp. 228–234.

Farooq, M. and Wang, D. B. (2008), 'Hybrid force/position control scheme for flexible joint robot with friction between and the end-effector and the environment', *International Journal of Engineering Science* **46**, 1266–1278.

Fiala, M. (2010), 'Designing highly reliable fiducial markers', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **32**(7), 1317–1324.

Fischler, M. A. and Bolles, R. C. (1981), 'Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography', *Communications of the ACM* **24**(6), 381–395.

Fisher, W. D. and Mujtaba, M. S. (1992), 'Hybrid position/force control: a correct formulation', *The International Journal of Robotics Research* **11**(4), 299–311.

Freitas, G. M., Leite, A. C. and Lizarralde, F. (2011), 'Kinematic control of constrained robotic systems', *Sba: Controle & Automação Sociedade Brasileira de Automatica* **22**(6), 559–572.

Gao, X.-S., Hou, X.-R., Tang, J. and Cheng, H.-F. (2003), 'Complete Solution Classification for the Perspective-three-point Problem', *IEEE Trans. on Pattern Anal. and Mach. Intell.* **25**(8), 930–943.

Guizzo, E. and Ackerman, E. (2012), How Rethink Robotics Built Its New Baxter Robot Worker. IEEE Spectrum.

Hagn, U., Konietschke, R., Tobergte, A., Nickl, M., Jörg, S., Kübler, B., Passig, G., Gröger, M., Fröhlich, F., Seibold, U. et al. (2010), 'Dlr mirosurge: a versatile system for research in endoscopic telesurgery', *International journal of computer assisted radiology and surgery* **5**(2), 183–193.

Hamilton, W. R. (1844), On a new species of imaginary quantities connected with a theory of quaternions, *in* 'Proceedings of the Royal Irish Academy', Vol. 2, pp. 424–434.

Hattori, A., Suzuki, N., Hashizume, M., Akahoshi, T., Konishi, K., Yamaguchi, S., Shimada, M. and Hayashibe, M. (2002), 'A robotic surgery system (da vinci) with image guided function–system architecture and cholecystectomy application.', *Studies in health technology and informatics* **94**, 110–116.

Hayati, S. (1986), Hybrid position/force control of multi-arm cooperating robots, *in* 'Proc. IEEE Int. Conf. Rob. Aut.', Vol. 3, pp. 82–89.

Herrmann, G., Jalani, J., Mahyuddin, M. N., Khan, S. G. and Melhuish, C. (2014), 'Robotic hand posture and compliant grasping control using operational space and integral sliding mode control', *Robotica* pp. 1–23.

Hogan, N. (1984), Impedance control: An approach to manipulation, *in* 'American Control Conference, 1984', IEEE, pp. 304–313.

Iborra, A., Pastor, J., Alvarez, B., Fernandez, C. and Merono, J. (2003), 'Robots in radioactive environments', *IEEE Robot Autom Mag* **10**(4), 12–22.

Jafari, A. and Ryu, J.-H. (2016), 'Independent force and position control for cooperating manipulators handling an unknown object and interacting with an unknown environment', *Journal of the Franklin Institute* **353**(4), 857–875.

Kruse, D., Wen, J. and Radke, R. (2015), 'A sensor-based dual-arm tele-robotic system', *IEEE Trans. Autom. Science and Eng.* **12**(1), 4–18.

Le, D. P., Choi, J. and Kang, S. (2013), External force estimation using joint torque sensors and its application to impedance control of a robot manipulator, *in* 'Control, Automation and Systems (ICCAS), 2013 13th International Conference on', IEEE, pp. 1794–1798.

Leite, A. C. (2005), Controle híbrido de força e visão de um manipulador robótico sobre superfícies desconhecidas, Master's thesis, COPPE/UFRJ.

Leite, A. C. (2011), Servovisão Adaptativa e Controle de Força para Robôs Manipuladores com Cinemática e Dinâmica Incertas Interagindo com Ambientes não-estruturados, PhD thesis, COPPE/UFRJ.

Leite, A. C., Lizarralde, F. and Hsu, L. (2009), 'Hybrid adaptive vision-force control for robot manipulators interacting with unknown surfaces', *The International Journal of Robotics Research* **28**(7), 911–926.

Leite, A. C., Lizarralde, F. and Hsu, L. (2010), A cascaded-based hybrid position-force control for robot manipulators with nonnegligible dynamics, *in* 'Proc. American Contr. Conf.', pp. 5260–5265.

Lepetit, V., Moreno-Noguer, F. and Fua, P. (2009), 'Epnp: An accurate o (n) solution to the pnp problem', *International journal of computer vision* **81**(2), 155–166.

Linderoth, M., Stolt, A., Robertsson, A. and Johansson, R. (2013), Robotic force estimation using motor torques and modeling of low velocity friction disturbances, *in* 'Proc. IEEE/RSJ Int. Conf. Intell. Rob. Syst.', pp. 3550–3556.

Lippiello, V., Ruggiero, F., Siciliano, B. and Villani, L. (2013), 'Visual grasp planning for unknown objects using a multifingered robotic hand', *Mechatronics, IEEE/ASME Transactions on* **18**(3), 1050–1059.

Liu, Y.-H. and Arimoto, S. (1998), 'Decentralized adaptive and nonadaptive position/force controllers for redundant manipulators in cooperations', *The International Journal of Robotics Research* **17**, 232–247.

Lizarralde, F. and Wen, J. T. (1996), 'Attitude control without angular velocity measurement: A passivity approach', *Automatic Control, IEEE Transactions on* **41**(3), 468–472.

Luh, J. and Zheng, Y. (1987), 'Constrained relations between two coordinated industrial robots for motion control', *The International journal of robotics research* **6**(3), 60–70.

Marquardt, D. W. (1963), 'An algorithm for least-squares estimation of nonlinear parameters', *Journal of the Society for Industrial & Applied Mathematics* **11**(2), 431–441.

Martinet, P., Lee, S., Kim, H. et al. (2008), 'Damped least square based genetic algorithm with ggaussian distribution of damping factor for singularity-robust inverse kinematics', *Journal of Mechanical Science and Technology* **22**(7), 1330–1338.

Mason, M. T. (1981), 'Compliance and force control for computer controlled manipulators', *IEEE Trans. Syst., Man and Cyber.* **11**(6), 418–432.

McClamroch, N. (1986), 'Singular systems of differential equations as dynamic models for constrained robot systems', *Proceedings. 1986 IEEE International Conference on Robotics and Automation* .

Miller, A. T., Knoop, S., Christensen, H. I. and Allen, P. K. (2003), Automatic grasp planning using shape primitives, *in* 'Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on', Vol. 2, IEEE, pp. 1824–1829.

Mukai, T., Hirano, S., Nakashima, H., Kato, Y., Sakaida, Y., Guo, S. and Hosoe, S. (2010), Development of a nursing-care assistant robot riba that can lift a human in its arms, *in* 'Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on', IEEE, pp. 5996–6001.

Murray, R. M., Li, Z. and Sastry, S. S. (1994), *A mathematical introduction to robotic manipulation*, CRC Press.

Nakamura, Y. and Hanafusa, H. (1986), 'Inverse kinematic solutions with singularity robustness for robot manipulator control', *Journal of dynamic systems, measurement, and control* **108**(3), 163–171.

Nakano, E., Ozaki, S., Ishida, T. and Kato, I. (1974), Cooperational control of the anthropomorphous manipulator melarm, *in* 'Proc. 4th Int. Symp. Industrial Robots', pp. 251–260.

Oberkampf, D., DeMenthon, D. F. and Davis, L. S. (1996), 'Iterative pose estimation using coplanar feature points', *Comp. Vis. Image Understand.* **63**(3), 495–511.

Olson, E. (2011), Apriltag: A robust and flexible visual fiducial system, *in* 'Robotics and Automation (ICRA), 2011 IEEE International Conference on', IEEE, pp. 3400–3407.

Panwar, V., Kumar, N., Sukavanam, N. and Borm, J.-H. (2012), 'Adaptive neural controller for cooperative multiple robot manipulator system manipulating a single rigid object', *Applied Soft Computing* **12**(1), 216–227.

Petersen, T. (2008*a*), 'A comparison of 2d-3d pose estimation methods', *Master's thesis, Aalborg University-Institute for Media Technology Computer Vision and Graphics, Lautrupvang* **15**, 2750.

Petersen, T. (2008*b*), 'A comparison of 2d-3d pose estimation methods', *Aalborg University-Institute for Media Technology Computer vision and graphics. Aalborg University* .

Pratt, G. A. and Williamson, M. M. (1995), Series elastic actuators, *in* 'Proc. IEEE/RSJ Int. Conf. Intell. Rob. Syst.', Vol. 1, pp. 399–406.

Pratt, G. and Manzo, J. (2013), 'The darpa robotics challenge [competitions]', *Robotics & Automation Magazine, IEEE* **20**(2), 10–12.

Qian, K., Song, A., Bao, J. and Zhang, H. (2012), 'Small teleoperated robot for nuclear radiation and chemical leak detection', *International Journal of Advanced Robotic Systems* **9**.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. and Ng, A. (2009), ROS: An Open-source Robot Operating System, *in* 'ICRA Workshop on Open Source Software'.

Raibert, M. H. and Craig, J. J. (1981), 'Hybrid position/force control of manipulators', *Journal of Dynamic Systems, Measurement, and Control* **103**(2), 126–133.

Ribo, M. and Brandner, M. (2005), State of the art on vision-based structured light systems for 3d measurements, *in* 'Robotic Sensors: Robotic and Sensor Environments, 2005. International Workshop on', IEEE, pp. 2–6.

Rice, A. C., Beresford, A. R. and Harle, R. K. (2006), Cantag: an open source software toolkit for designing and deploying marker-based vision systems, *in* 'Pervasive Computing and Communications, 2006. PerCom 2006. Fourth Annual IEEE International Conference on', IEEE, pp. 10–21.

Sadeghian, H., Ficuciello, F., Villani, L. and Keshmiri, M. (2012), 'Global impedance control of dual-arm manipulation for safe interaction', *IFAC Proceedings Volumes* **45**(22), 767–772.

Salisbury, J. K. (1980), Active stiffness control of a manipulator in cartesian coordinates, *in* 'Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on', IEEE, pp. 95–100.

Saxena, A., Driemeyer, J. and Ng, A. Y. (2008), 'Robotic grasping of novel objects using vision', *The International Journal of Robotics Research* **27**(2), 157–173.

Schneider, S. and Cannon, R.H., J. (1992), 'Object impedance control for cooperative manipulation: theory and experimental results', *IEEE Trans. Rob. Autom.* **8**(3), 383–394.

Schweighofer, G. and Pinz, A. (2006), 'Robust pose estimation from a planar target', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **28**(12), 2024–2030.

Siciliano, B. and Khatib, O. (2008), *Springer Handbook of Robotics*, Springer Science & Business Media.

Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G. (2009), *Robotics: Modeling, Planning and Control*, Springer Verlag London.

Sieber, D., Musić, S. and Hirche, S. (2015), Multi-robot manipulation controlled by a human with haptic feedback, *in* 'Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on', IEEE, pp. 2440–2446.

Slotine, J.-J. E., Li, W. et al. (1991), *Applied nonlinear control*, Vol. 199, Prentice-Hall Englewood Cliffs, NJ.

Smith, C., Karayiannidis, Y., Nalpantidis, L., Gratal, X., Qi, P., Dimarogonas, D. V. and Kragic, D. (2012), 'Dual arm manipulation: a survey', *Robotics and Autonomous systems* **60**(10), 1340–1353.

Soares, B. F. F., da Silva, F. S., Nigri, I., Mello, C. B. and Meggiolaro, M. A. (2008), 'Master-slave servo-bilateral control of direct drive electrical manipulators'.

Stolt, A., Linderoth, M., Robertsson, A. and Johansson, R. (2012), Force controlled robotic assembly without a force sensor, *in* 'Robotics and Automation (ICRA), 2012 IEEE International Conference on', IEEE, pp. 1538–1543.

Suda, R., Kosuge, K. and Kakuya, H. (2003), Object-impedance-based cooperative handling of object by mobile robot helper and human using visual and force information, *in* 'Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings. 2003 IEEE/ASME International Conference on', Vol. 1, IEEE, pp. 592–597.

Sujan, V. A. and Meggiolaro, M. A. (2004), 'Model predictive disturbance rejection during cooperative mobile robot assembly tasks', *Journal of the Brazilian Society of Mechanical Sciences and Engineering* **26**, 260 – 268.

Sun, D. and Mills, J. K. (2002), 'Manipulating rigid payloads with multiple robots using compliant grippers', *Mechatronics, IEEE/ASME Transactions on* **7**(1), 23–34.

Tarn, T., Bejczy, A. and Yun, X. (1988), 'New nonlinear control algorithms for multiple robot arms', *IEEE Transactions on Aerospace and Electronic Systems* **24**(5), 571–583.

Taylor, R. and Stoianovici, D. (2003), 'Medical robotics in computer-integrated surgery', *Robotics and Automation, IEEE Transactions on* **19**(5), 765–781.

Tinos, R., Terra, M. and Ishihara, J. (2006), 'Motion and force control of cooperative robotic manipulators with passive joints', *IEEE Trans. Cont. Syst. Tech.* **14**(4), 725–734.

Uchiyama, M. and Dauchez, P. (1987), Statics, kinematics, and hybrid control scheme for a two arm robot, *in* 'Proceedings of the 9th IASTED International Symposium Robotics and Automation', pp. 28–32.

Uchiyama, M. and Dauchez, P. (1988), A symmetric hybrid position/force control scheme for the coordination of two robots, *in* 'IEEE Int. Conf. Rob. Aut.', pp. 350–356.

Uchiyama, M. and Dauchez, P. (1992), 'Symmetric kinematic formulation and non-master/slave coordinated control of two-arm robots', *Advanced Robotics* **7**(4), 361–383.

Uchiyama, M., Iwasawa, N. and Hakomori, K. (1987), 'Hybrid position/force control for coordination of a two-arm robot', *Proceedings. 1987 IEEE International Conference on Robotics and Automation* .

Vargas, L. V., Leite, A. C. and Costa, R. R. (2014), Overcoming kinematic singularities with the filtered inverse approach, *in* 'Proc. 19th IFAC World Congress', pp. 8496–8502.

Villani, L. and De Schutter, J. (2008), Force control, *in* 'Springer handbook of robotics', Springer, pp. 161–185.

Wahrburg, A., Zeiss, S., Matthias, B. and Ding, H. (2014), Contact force estimation for robotic assembly using motor torques, *in* '2014 IEEE International Conference on Automation Science and Engineering (CASE)', IEEE, pp. 1252–1257.

Wampler, C. W. et al. (1986), 'Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods', *Systems, Man and Cybernetics, IEEE Transactions on* **16**(1), 93–101.

Wang, Y., Lang, H. and De Silva, C. W. (2010), 'A hybrid visual servo controller for robust grasping by wheeled mobile robots', *Mechatronics, IEEE/ASME Transactions on* **15**(5), 757–769.

Wen, J.-Y. and Kreutz-Delgado, K. (1991), 'The attitude control problem', *Automatic Control, IEEE Transactions on* **36**(10), 1148–1162.

Xu, C. and Prince, J. L. (1998), 'Snakes, shapes, and gradient vector flow', *Image Processing, IEEE Transactions on* **7**(3), 359–369.

Yamano, M., Kim, J.-S., Konno, A. and Uchiyama, M. (2004), 'Cooperative control of a 3d dual-flexible-arm robot', *Journal of Intelligent and Robotic Systems* **39**(1), 1–15.

Yamano, M., Kim, J.-S. and Uchiyama, M. (1998), Hybrid position/force control of two cooperative flexible manipulators working in 3d space, *in* 'Proc. IEEE Int. Conf. Rob. Aut.', Vol. 2, pp. 1110–1115.

Ye, S., Suzuki, K., Suzuki, Y., Ishikawa, M. and Shimojo, M. (2013), Robust robotic grasping using ir net-structure proximity sensor to handle objects with unknown position and attitude, *in* 'Robotics and Automation (ICRA), 2013 IEEE International Conference on', IEEE, pp. 3271–3278.

Yoshimi, B. H. and Allen, P. K. (1994), Visual control of grasping and manipulation tasks, *in* 'Multisensor Fusion and Integration for Intelligent Systems, 1994. IEEE International Conference on MFI'94.', IEEE, pp. 575–582.

Yuan, J. S. (1988), 'Closed-loop manipulator control using quaternion feedback', *IEEE Journal of Robotics and Automation* **4**(4), 434–440.

Zeng, G. and Hemami, A. (1997), 'An overview of robot force control', *Robotica* **15**(05), 473–482.

Appendix A

Proof of Theorem 1: Dual-arm Hybrid Relative Position-Force Control

For the closed-loop stability analysis of the cooperative hybrid relative position-force scheme, the following Lyapunov function is proposed:

$$2\,V_h(e_{pa},\xi_{pr},\xi_{fr},e_{qa},e_{qr}) = e_{pa}^{\mathsf{T}}\,e_{pa} + \xi_{pr}^{\mathsf{T}}\,\xi_{pr} + \xi_{fr}^{\mathsf{T}}\,\xi_{fr} + (e_{qa,\,s}\,-\,1)^2 + e_{qa,\,v}^{\mathsf{T}}\,e_{qa,\,v} + (e_{qr,\,s}\,-\,1)^2 + e_{qr,\,v}^{\mathsf{T}}\,e_{qr,\,v}\;. \quad \text{(A.1)}$$

The time derivative of the decoupled position and force errors (3.62) and (3.63) is given by:

$$\dot{\xi}_{pr} = \dot{R}_{bs}\,(I-S)\,R_{bs}^{\mathsf{T}}\,e_{pr} + R_{bs}\,(I-S)\,[(\dot{R_{bs}^{\mathsf{T}}})\,e_{pr} + R_{bs}^{\mathsf{T}}\,\dot{e}_{pr}]\;, \quad \text{(A.2)}$$

$$\dot{\xi}_{fr} = \dot{R}_{bs}\,S\,R_{bs}^{\mathsf{T}}\,e_{fr} + R_{bs}\,S\,[(\dot{R_{bs}^{\mathsf{T}}})\,e_{fr} + R_{bs}^{\mathsf{T}}\,\dot{e}_{fr}]\;. \quad \text{(A.3)}$$

Remind that $\dot{R}_{bs} = \mathcal{S}(\omega_b)\,R_{bs}$ or equivalently $\dot{R}_{bs} = R_{bs}\,\mathcal{S}(\omega_s)$, where vector ω denotes the angular velocity of frame $\mathcal{F}_s$ with respect to frame $\mathcal{F}_b$. The notation ω_s and ω_b means the vector is expressed in the frame $\mathcal{F}_s$ or frame $\mathcal{F}_b$, respectively. Also, knowing that $\dot{R}_{sb} = (\dot{R_{bs}^{\mathsf{T}}}) = -\mathcal{S}(\omega_s)\,R_{bs}^{\mathsf{T}}$, then it can be concluded that $(\dot{R_{bs}^{\mathsf{T}}}) = (\dot{R}_{bs})^{\mathsf{T}}$. Since the selection matrices are diagonal, then the first and second terms of equations are equivalent.

Besides, from (3.61) and (3.41) the relations $\dot{p}_r = v_{hr}$ and $\dot{f}_r = \frac{1}{2}\,K_s\,\dot{p}_r$ are obtained and thus the time derivative of the relative errors are $\dot{e}_{pr} = \dot{p}_r^d - v_{hr}$ and $\dot{e}_{fr} = \dot{f}_r^d - (\frac{1}{2}\,K_s\,\dot{p}_r)$, respectively. Therefore, the error dynamics (A.2)-(A.3) can be

rewritten as:

$$\dot{\xi}_{pr} = R_{bs}\,(I-S)\,R_{bs}^{\mathsf{T}}\,[-2\,\mathcal{S}(\omega_b)\,e_{pr} + \dot{p}_r^d - v_{hr}]\;, \tag{A.4}$$

$$\dot{\xi}_{fr} = R_{bs}\,S\,R_{bs}^{\mathsf{T}}\,[-2\,\mathcal{S}(\omega_b)\,e_{fr} + \dot{f}_r^d - \frac{1}{2}\,K_s\,v_{hr}]\;. \tag{A.5}$$

From the quaternion error propagation (2.19) (Lizarralde and Wen, 1996), the quaternion error dynamics is given by

$$\dot{e}_q = \frac{1}{2}\,J_q(e_q)\,\tilde{\omega}\,, \qquad J_q(e_q) = \begin{bmatrix} -e_{qv}^{\mathsf{T}} \\ e_{qs}I - \mathcal{S}(e_{qv}) \end{bmatrix}\;, \tag{A.6}$$

where $\tilde{\omega} = \omega^d - \omega$. Then, the time derivative of V_h is obtained:

$$\begin{aligned} \dot{V}_h &= e_{pa}^{\mathsf{T}}\,\dot{e}_{pa} + \xi_{pr}^{\mathsf{T}}\,\dot{\xi}_{pr} + \xi_{fr}^{\mathsf{T}}\,\dot{\xi}_{fr} + e_{qa,\,v}^{\mathsf{T}}(\omega_a^d - \omega_a) + e_{qr,\,v}^{\mathsf{T}}(\omega_r^d - \omega_r) \\ &= e_{pa}^{\mathsf{T}}\,[\dot{p}_a^d - v_{pa}] + \xi_{pr}^{\mathsf{T}}\,R_{bs}\,(I-S)\,R_{bs}^{\mathsf{T}}\,[-2\,\mathcal{S}(\omega_b)\,e_{pr} + \dot{p}_r^d - v_{hr}] + \\ &\quad \xi_{fr}^{\mathsf{T}}\,R_{bs}\,S\,R_{bs}^{\mathsf{T}}\,[-2\,\mathcal{S}(\omega_b)\,e_{fr} + \dot{f}_r^d - \frac{1}{2}\,K_s\,v_{hr}] + e_{qa,\,v}^{\mathsf{T}}\,\tilde{\omega}_a + e_{qr,\,v}^{\mathsf{T}}\,\tilde{\omega}_r\;. \end{aligned} \tag{A.7}$$

Substituting the hybrid control law v_{hr} given by (3.59) into and using the following properties of the orthogonal subspaces (Mason, 1981) we obtain

$$(I-S)\,S = S\,(I-S) = 0, \quad (I-S)\,(I-S) = (I-S), \quad S\,S = S, \tag{A.8}$$

it is possible to obtain:

$$\begin{aligned} \dot{V}_h = e_{pa}^{\mathsf{T}}\,[\dot{p}_a^d - v_{pa}] + \xi_{pr}^{\mathsf{T}}\,R_{bs}\,(I-S)\,R_{bs}^{\mathsf{T}}\,[\dot{p}_r^d - v_{pr}] + \\ \xi_{fr}^{\mathsf{T}}\,R_{bs}\,S\,R_{bs}^{\mathsf{T}}\,[\dot{f}_r^d - \frac{1}{2}\,K_s\,v_{fr}] + e_{qa,\,v}^{\mathsf{T}}\,\tilde{\omega}_a + e_{qr,\,v}^{\mathsf{T}}\,\tilde{\omega}_r\;. \end{aligned} \tag{A.9}$$

Then, using the absolute and relative position control laws v_{pa} and v_{pr}, given by (3.29) and (3.55), the relative force control law v_{fr} given by (3.56), as well as the absolute and relative orientation control laws ω_a and ω_r given by and (3.19), the following result is obtained:

$$\dot{V}_h = -e_{pa}^{\mathsf{T}}\,K_{pa}\,e_{pa} - \xi_{pr}^{\mathsf{T}}\,K_{pr}\,\xi_{pr} - \frac{1}{2}\xi_{fr}^{\mathsf{T}}\,K_{fr}\,\xi_{fr} - e_{qa,\,v}^{\mathsf{T}}\,K_{oa}\,e_{qa,\,v} - e_{qr,\,v}^{\mathsf{T}}K_{or}\,e_{qr,\,v} \leq 0\;.$$

Considering the tracking problem, the error dynamics is non-autonomous and thus the Barbalat's lemma has to be applied to analyze the convergence and stability properties of the system.

In order to apply Barbalat's Lemma for stability analysis (Slotine et al., 1991), function $\dot{V}_h(t)$ has to be *uniformly continuous* in addition to be negative or zero. A sufficient condition for a differentiable function to be uniformly continuous is that

its derivative be bounded. By analyzing

$$\ddot{V}_h = -2\, e_{pa}^{\mathsf{T}}\, K_{pa}\, \dot{e}_{pa} - 2\, \xi_{pr}^{\mathsf{T}}\, K_{pr}\, \dot{\xi}_{pr} - \xi_{fr}^{\mathsf{T}}\, K_{fr}\, \dot{\xi}_{fr} - 2\, e_{qa,\,v}^{\mathsf{T}}\, K_{oa}\, \dot{e}_{qa,\,v} - 2\, e_{qr,\,v}^{\mathsf{T}}\, K_{or}\, \dot{e}_{qr,\,v}$$

it is sufficient to demonstrate that the errors and its derivatives are bounded.

Since $\dot{V}_h(t) \leq 0$, this implies that $V_h(t) \leq V_h(0)$ and, therefore that $e_{pa}, \xi_{pr}, \xi_{fr}, e_{qa,\,v}$ and $e_{qr,\,v}$ are bounded. Taking into account (3.62) and (3.63), signals e_{pr} and e_{fr} also satisfy the boundedness property.

Under the assumption that p_a^d, p_r^d, f_r^d and their respective time derivatives $\dot{p}_a^d, \dot{p}_r^d, \dot{f}_r^d$ are bounded, it is possible to conclude from (3.29), (3.55) and (3.56) that v_{pa}, v_{pr}, v_{fr} as well as v_{hr} are also bounded. From the errors time derivative equations $\dot{e}_{pa} = \dot{p}_a^d - v_{pa}$, (A.4) and (A.5), it implies that $\dot{e}_{pa}, \dot{\xi}_{pr}, \dot{\xi}_{fr}$ are bounded. For orientation terms, assuming that ω_a^d, ω_r^d are bounded and knowing that $J_q, \tilde{\omega}_a, \tilde{\omega}_r$ are also bounded, then $\dot{e}_{qa,\,v}, \dot{e}_{qr,\,v}$ are also bounded. This shows that $\ddot{V}$ is bounded and, hence, $\dot{V}$ is uniformly continuous.

Since $V_h(t) > 0$ and $\dot{V}_h(t) \leq 0$, $\lim_{t\to\infty} V_h(t) = V_\infty > 0$ exists. Finally, using the Barbalat's Lemma implies that: $\lim_{t\to\infty} e_{pa}(t) = 0$, $\lim_{t\to\infty} \xi_{pr}(t) = 0$, $\lim_{t\to\infty} \xi_{fr}(t) = 0$, $\lim_{t\to\infty} e_{qa,\,v}(t) = 0$, $\lim_{t\to\infty} e_{qr,\,v}(t) = 0$ and, as $\|q\| = 1$, $\lim_{t\to\infty} e_{qa,\,s}(t) = \pm 1$, $\lim_{t\to\infty} e_{qr,\,s}(t) = \pm 1$, which proves that the overall closed-loop control system is Almost-Globally Asymptotically Stable (AGAS). The term AGAS is used to indicate that the domain of attraction is the entire state space, except for a set of measure zero (Wen and Kreutz-Delgado, 1991).

Appendix B

Architecture for Teleoperation using Natural User Interface

Considering a teleoperated application, we propose a system architecture composed by a local and a remote environment, as represented in Figure B.1. In the local environment, the robot communicates via ROS messages with a local pc. This local pc is responsible for the main data processing – it executes the visual identification algorithm and the control loop calculations. The local pc is able to communicate over Internet, via ROS messages, to a remote pc, which contains a GUI (Graphic User Interface) and is responsible for obtaining the *setpoints* for the object motion.

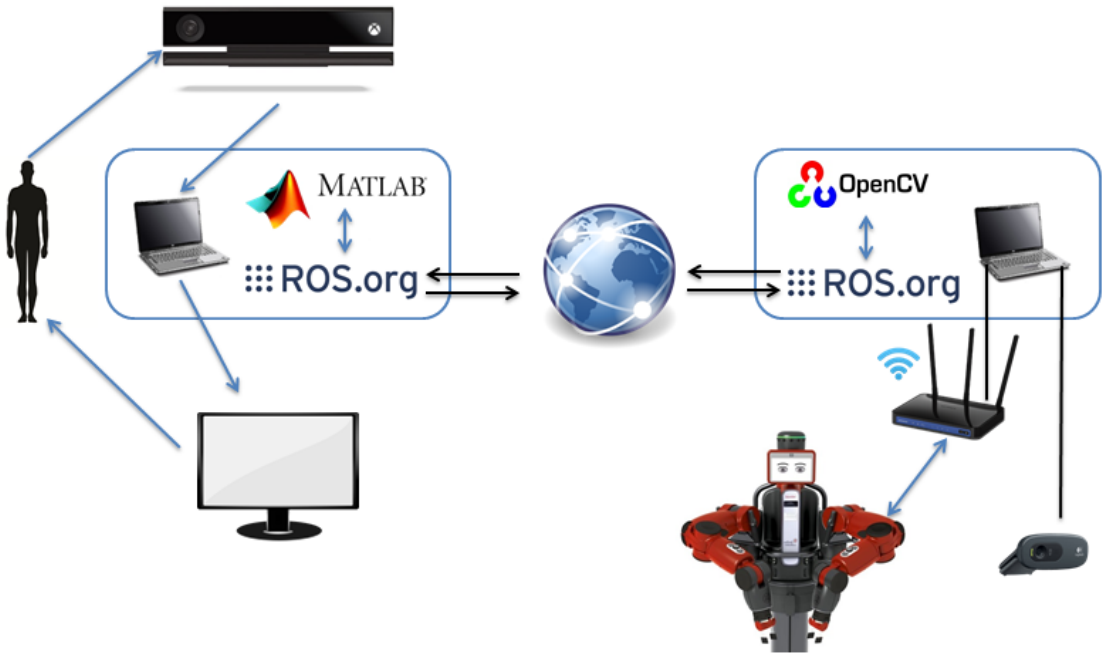


Figure B.1: System architecture

B.1 Human Machine Interface

The operator can choose the desired motion to be a preloaded trajectory or may define the trajectory through a mathematical equation in the GUI. Another possibility is to use a natural interface device, such as the *Microsoft Kinect*. Thus, the operator can conduct the desired motion of the object by moving his arms.

The Microsoft Kinect uses one RGB camera (640 × 480 px @ 30 Hz) and one infrared (IR) depth sensor (640 × 480 px @ 30 Hz), which provide full-body 3D motion capture. The Kinect also counts with a Software Development Kit (SDK) and there are libraries implemented in C++ and Matlab.

First, the Matlab software obtains the visual information from the cameras. Second, a skeleton algorithm identifies the main joints and links of the operator body and creates a matrix of Joint Coordinates (see Figure B.2). Next, the left and right wrist position are compared to the body center and are used to calculate the desired position for the box. Also, the vector connecting the wrists is normalized and its orientation defines the desired orientation for the object.

Since the operator range is different from the robotic manipulators range, a scaling is performed.

Finally, a message containing the desired position and orientation is sent to the local pc via ROS, which adjusts the position control loop *setpoints*. In the next iteration, the Matlab loop restarts, performs the calculation for the new operator position and, at the end, another message in sent again.

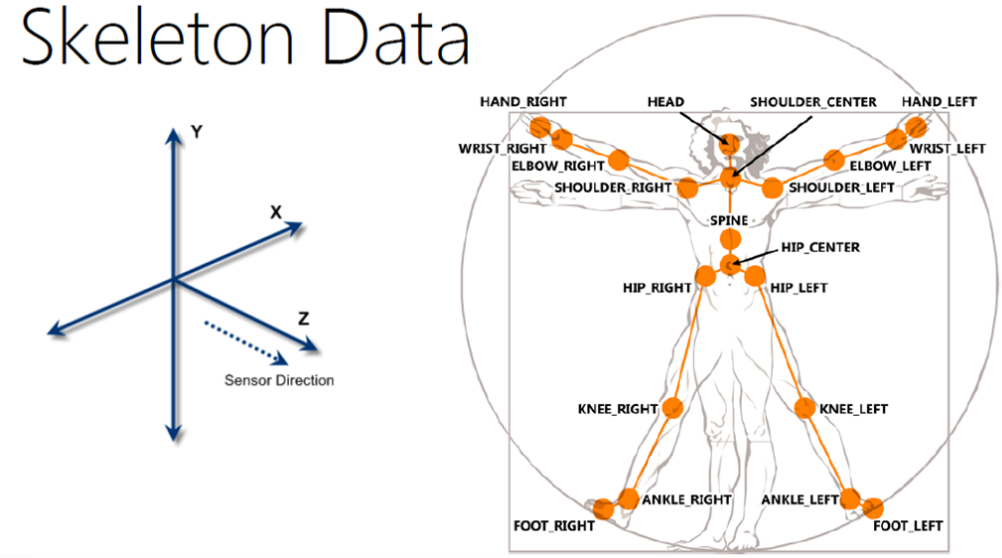


Figure B.2: Kinect skeleton model. Extracted from Microsoft website.