TECHNICAL UNIVERSITY OF CATALONIA BARCELONATECH

Doctoral Programme

AUTOMATIC CONTROL, ROBOTICS AND COMPUTER VISION

PhD Dissertation

THE ROBOT NULL SPACE: NEW USES FOR NEW ROBOTIC SYSTEMS

Josep-Arnau Claret i Robert

Supervisor: Luis Basañez Villaluenga

May 2019

To my parents.

Abstract

With the new advances in the field of robotics, the kinematic complexity of robotic systems has dramatically increased, whether by the addition of more joints to standard robots, or by the appearance of new types of robots, like humanoid robots and Unmanned Air Vehicles (UAV), and new scenarios, like cooperative exploration of scenarios by groups of robots, or the interaction between robots and humans, among others. This is specially sensitive in shared tasks between a user and a machine, like in a teleoperation scenario. In this situations, a good compromise between the user input and the degrees of freedom managed by the system is critical, and a failure to properly adjust its autonomy can lead to under-performance and hazardous situations.

This dissertation proposes new approaches to address the autonomy of a robotic system through the robot redundancy, that is, the extra degrees of freedom available to solve the main task. These approaches allow to cope with the increasing complexity of the robotic systems, and to exploit them to simultaneously solve a hierarchy of multiple tasks with different levels of priority. In particular, in this work this redundancy is addressed using the robot null-space for two different robotic systems and goals.

Envisioning a world where users and humanoid robots tend to increasingly share space and tasks, in the first scenario, the redundancy of a humanoid robot is exploited to execute a novel task with low priority: the conveyance of emotional states to humans. A model to transform emotions as points in a three dimensional space to kinematic features of the robot is developed, as well as the implementation of the proposed approach in a Pepper robot. Further, the results of a user study to assess its validity are shown, its conclusions, and future directions.

In the second scenario, a novel robotic system for teleoperation is presented. This robotic system results from the integration of a mobile manipulator, an UAV and a haptic device. The user commands the end effector of the mobile manipulator using the haptic while getting real-time feedback from a camera mounted on the UAV, which takes the role of a free-flying camera. A description of this robotic system and its task hierarchy is given in the second part of the thesis, along with algorithms to coordinate the camera and the mobile manipulator in such a way that the command of the camera is transparent to the operator. Also, the null-space of the robot is exploited to improve the teleoperation experience in two directions. The first, by proposing two enhancements on the continuous inverse. The second, by using the null-space of the mobile manipulator to prevent its TCP from being occluded to the camera by the robot itself, which, to the authors knowledge, it a novel use of the robot null-space. Finally, the proposed algorithms are implemented in the Barcelona Mobile Manipulator and a Parrot AR.Drone.

Overall, new contributions on the management of the robot redundancy are presented in this dissertation, along with its implementation and validation, which expand the knowledge of this particular sub-field of robotics.

Acknowledgments

I thank Professor Luis Basañez for the opportunity to deep into the realm of robotics and accomplish this dissertation. This thesis would not have possible without his knowledge, guidance, words of encouragement at difficult times, and amusing conversations.

I would also like to thank Raúl Suárez and Jan Rosell. The first, for opening to me the doors of the IOC, many years ago, when I was still an undergraduate student, full of futuristic dreams of robots and spaceships. The second, for his invaluable support during all these years, for the trip to Lausanne, and for giving me the opportunity to contribute to the Kautham Project.

I thank Gentiane Venture for the warm welcome during my four-month internship in her lab, which resulted in the first part of the thesis. Also, to the members of the GVLab for the translations to Japanese, and attending the local participants during the user study. I cannot stress enough the impact those months in Tokyo had on me, among the temples and giant buildings of this amazing city, which memories I will treasure for the rest of my life.

This dissertation would not have been possible without the technical discussions, chats and laughs at the IOC. I want to thank Leo for always being ready to set up the hardware and software which are at the base of this work. And, also, Orestes, Carlos Aldana, Isiah, Marcos, Andrés, Henry, Gema, Néstor, Diana, José, Nacho, Abiud, Niliana, Carlos Rosales, Sergi, Emmanuel, Fernando, Carlos Rodríguez, Alexander, Paolo, Ali, Muhayyuddin, Diab, and to many others I have met during these exciting years.

Finally, thanks to my friends and my relatives, who will be pleased to know that I successfully closed this period of my life, and willing to see, as I am, which ones will come now.

And, last but not least, to my parents, for their patience, faith, and unconditional love. Among so many things, I owe them my interest in books since I was a child, which is at the root of my admiration and awe for scientists, science and its endless possibilities.

Josep-Arnau Claret i Robert. Barcelona, Spain. May 2019.

This work has been partially supported by the Spanish MINECO projects DPI2013-40882-P, DPI2014-57757-R and DPI2016-80077-R, the Spanish predoctoral grant BES-2012-054899, and the Japanese challenging exploratory research grant 15K12124.

Contents

List of Figures

List of Tables

Notation and Acronyms

Notation

$\rightarrow$	Tends to
$\in$	Belongs to
$\mathbb{R}$	All real numbers
$\mathbb{R}_{\geq 0}$	All non-negative real numbers: $\{x \in \mathbb{R} \mid x \geq 0\}$
$\mathbb{R}_{>0}$	All positive real numbers: $\{x \in \mathbb{R} \mid x > 0\}$
$SE(2)$	Special Euclidean Group in the Plane
$SE(3)$	Special Euclidean Group in the 3D Space
$SO(2)$	Rotation Group in the Plane
$SO(3)$	Rotation Group in the 3D Space
$x, X \in \mathbb{R}$	Scalar
$\mathbf{x} = [x_1 \;\; x_2 \ldots x_n]^T \in \mathbb{R}^n$	Vector
$\mathbf{X} \in \mathbb{R}^{m \times n}$	Matrix
$\lvert\cdot\rvert$	Absolute value of a real number (*i.e.*, $\lvert x\rvert = \sqrt{x^2}$)
$\lVert \cdot \rVert$	Euclidean norm of a vector (*i.e.*, $\lVert\mathbf{x}\rVert = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}$), or Induced norm of a matrix (*i.e.*, $\lVert\mathbf{A}\rVert = \max\limits_{\lVert\mathbf{x}\rVert=1} \lVert\mathbf{Ax}\rVert$)
$\boldsymbol{S}(\cdot)$	Cross product matrix
arg min f	Argument of the minimum of the function $f\,(\cdot)$
$\chi_A(\cdot)$	Indicator function of a subset A

Acronyms

BMM-I	Barcelona Mobile Manipulator I
DH	Denavit-Hartenberg
DOF	Degree of Freedom
FRI	Fast Research Interface
HRI	Human-Robot Interaction
JVG	Jerkiness - Activity - Gaze
MM	Mobile Manipulator
OBVF	Object Best View Function
PAD	Pleasure - Arousal - Dominance
SAM	Self-Assessment Manikin
SVD	Singular Value Decomposition
UAV	Unmanned Air Vehicle
UGV	Unmanned Ground Vehicle

Latin abbreviations

cf.	-confer-	see
etc.	-et cetera-	and the rest
et al.	-et alii-	and others
e.g.	-exempli gratia-	for instance
i.e.	-id est-	that is

English abbreviations

a.k.a.	also known as
s.t.	such that
w.r.t.	with respect to

Chapter 1

Introduction

It was a great thing to be a human being. It was something tremendous. Suddenly I'm conscious of a million sensations buzzing in me like bees in a hive. Gentlemen, it was a great thing.

Karel Čapek, Rossum's Universal Robots, 1920

SUMMARY: The appearance of new robots and its increasing complexity opens the way to new applications that allow multiple simultaneous tasks and take advantage of the degrees of freedom not directly controlled by the operator. This chapter outlines the main motivations underlying the use of the robot null space as a mean to cope with this complexity in two different robotic systems, and further summarizes the structure of the dissertation.

1.1 Context and motivation

Since its inception, the dream of roboticists has been to build machines that not only execute tasks similarly to humans, but even surpass them. The miniaturization of mechanical parts, together with more and more powerful electronic hardware, and the development of sophisticated algorithms, has been key to the growth of robotics in all areas of modern society. Beginning with the appearance of the Unimate robot in the sixties and the massive use of industrial robots from the next decade and on, to the current days, where autonomous vacuum cleaners have been widely adopted in our homes, and humanoid robots are step by step becoming a reality, the dream finally starts to take shape.

But in this scenario, the demands regarding what robots can actually do has been and is still growing, as more capabilities have been added to the known types of robots. Industrial

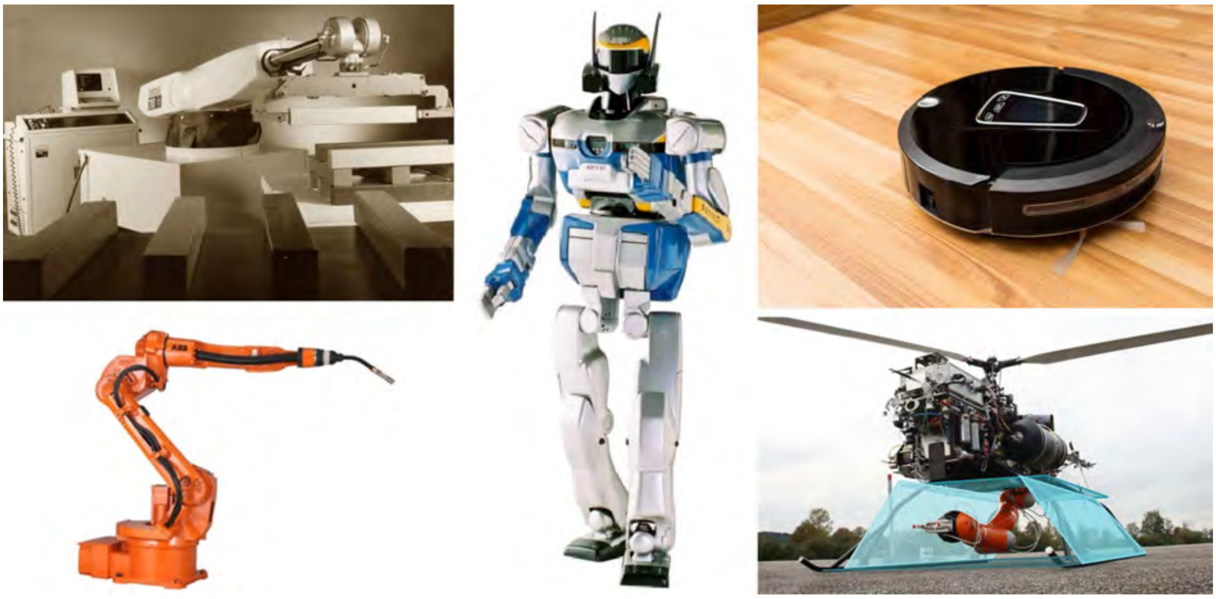


Figure 1.1 The Unimate robot, the first robot invented by George Devol (IEEE Spectrum, 2018), and a modern welding IRB 1520ID ABB industrial robot (ABB, 2018) (left figures), the HRP-3 humanoid robot at the center (Kaneko et al., 2008), and, in the right figures, a vacuum cleaner robot (Canstar Blue, 2018), and a UAV with a Kuka LWR attached as an arm with a gripper (Huber et al., 2013), they all give an idea of the diversity in robot design and capabilities.

manipulators have increased their number of joints, its speed and accuracy and, finally, taken out of their cells and become mobile. Also, new actors have appeared, like humanoid robots, autonomous vehicles and, more recently, affordable Unmanned Aerial Vehicles (UAV) (Fig. 1.1).

Thus, the well-known term *robot*, still necessary and totally in use today, has evolved to encompass a wider range of setups. It can now be seen as a particular case of a broader concept, the *robotic system*, where several robots, sensors and devices interact with each other, and even with humans, to accomplish the desired tasks. Examples of robotic systems can be seen in robots that cooperate to carry a load, multi-robot systems where a flocking formation has to be maintained between its members, in the exploration of an unknown environment by a team of robots, or in a scenario where a UAV is jointly moving with a mobile robot located on the ground, among others (Fig. 1.2).

Ultimately, whether it is due to the fact that each robot has itself become more sophisticated, or because the need has arisen to command a higher number of robots simultaneously, the overall system complexity has incremented dramatically. With it, the challenge to create new tools to address the higher number of possibilities, and to exploit them for new purposes has become apparent, as well as the need to give the user tools to command the system in order to achieve its desired goals and subgoals.

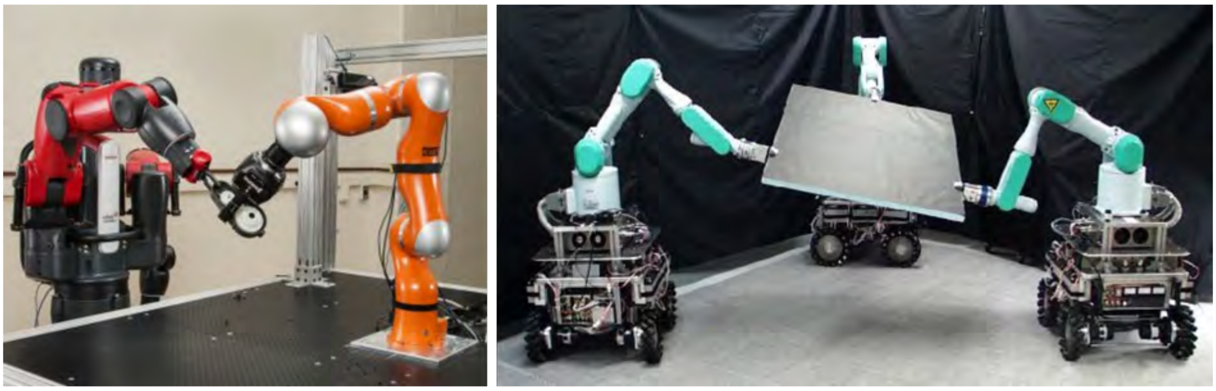


Figure 1.2 Two robots cooperating in the manipulation of an object (National Institute of Standards and Technology, 2018) (left), and a group of three mobile manipulators carrying a heavy load (Robotics @SNU, 2018) (right).

Dealing with the complexity of the system becomes particularly sensitive in situations where the user is actually part of the system. This increase in complexity can be explained by the need to design shared strategies to properly distribute the execution of the different tasks between the robots and the humans, while at the same time maintaining the task hierarchy, and avoiding collision between the different goals, that is, ensuring that the execution of a task does not negatively affects other tasks.

This is the case, for instance, of a teleoperation scenario, where the operator is controlling the robot or the robotic system to some extent in order to achieve a specific task. Teleoperated robots are advantageous in situations where it is needed to work in locations and execute tasks too dangerous for persons, like in catastrophic areas, war scenarios, or inside a nuclear plant, for example. It can also be advantageous, and even necessary, in situations where the robot can enhance the capabilities of the operator, like a surgical robot, where the robot can increase the accuracy of the motions of the surgeon, or a robot which transports a heavy load, too heavy to be carried by a human.

In comparison to fully autonomous robots, teleoperated robots can take advantage of the expertise of the operator in order to accomplish the task. The problem, though, arises due to the high complexity of the teleoperated system, which may in some situations be too much to be controlled exclusively by a single operator. Naturally, a certain degree of autonomy has to be given to the system itself, which accounts for the aspects that are not controlled by the user. This common scenario is specially significant when the operator is focused on the execution of a particular task, and the system is dedicated to one or more other tasks, whether they are secondary (reduction of the energy consumption, or increase of manipulability, for instance), thus allowing the operator to fulfill the tasks that account for the highest priority, or even by taking control over the operator commands, if a dangerous situation arises which has to be dealt with in a particular way *sine qua non*, like avoiding the collision with an obstacle.

In this regard, multiple approaches have been devised with the intent to deal with the redundancy of a robotic system, that is, when the number of degrees of freedom (DOFs) needed to achieve the main task are lower than the total number of DOFs available, and the user and the autonomous agents have to share the pool of DOFs. It is then required a specific setting for the level of autonomy of the system, topic which has been important for the research community for decades (Sheridan, 1992), and which has lead to multiple architectures with the purpose to efficiently integrate the user and the robot to fulfil the task.

1.2 Objectives

The complexity of a robotic system at the kinematic and dynamic levels can be described mathematically by using the concept of the degrees of freedom of the system. The total number of DOFs correspond to the number of possible independent displacements and rotations, and is generally equivalent to the number of joints available and actionable. Through appropriate inputs to the robot joints, whether modeled as a force (or a torque, for a revolute joint) or as velocity (linear or angular, also depending on the type of actuator), motions on the robot can be generated in order to fulfill a certain task. The degrees of freedom needed to execute the task impose the required complexity of the system. Thus, if a certain tasks needs three DOFs to transport an object to a certain position, three joints will at least be necessary to achieve the motion.

When the number of available joints of a robot is higher than the minimum necessary to accomplish the desired task, the robot is said to be redundant. Redundant robots give significant advantages over non-redundant robots, as they allow the execution of a task in different ways, the execution of multiple tasks simultaneously, or even to execute a main task along with secondary tasks at different levels of priority, thus creating a hierarchy of tasks.

The redundancy of a robot can be mathematically tackled using different approaches, like the weighted pseudoinverse (Chan and Dubey, 1995; Park et al., 2001), the augmented Jacobian (Egeland, 1987), the extended Jacobian (Klein et al., 1995) or the task priority formulations (Nakamura et al., 1987; Baerlocher and Boulic, 1998). Most of these approaches make partial or full use of the null space of the robot, which can be roughly described as the space of motions which do not affect the movements related to the main task.

While usually the main task of a robot is to place its end-effector (one or many) in a desired position with or without a specific orientation, the robot null space has been historically used to fulfill a number of secondary tasks. Some of these secondary tasks include control of multiple end-effectors (Baerlocher and Boulic, 1998), singularity avoidance (Nemec and Zlajpah, 2000), obstacle avoidance (Maciejewski and Klein, 1985), self-collision avoidance (Petrič and Žlajpah, 2011), joint constraints (Mansard et al., 2009a), manipulability optimization (Yoshikawa, 1985), center of mass positioning (Baerlocher and Boulic, 2004), compliant behavior (Peng and Adachi, 1993), reduction of energy consumption, *etc*. (Figs. 1.3).

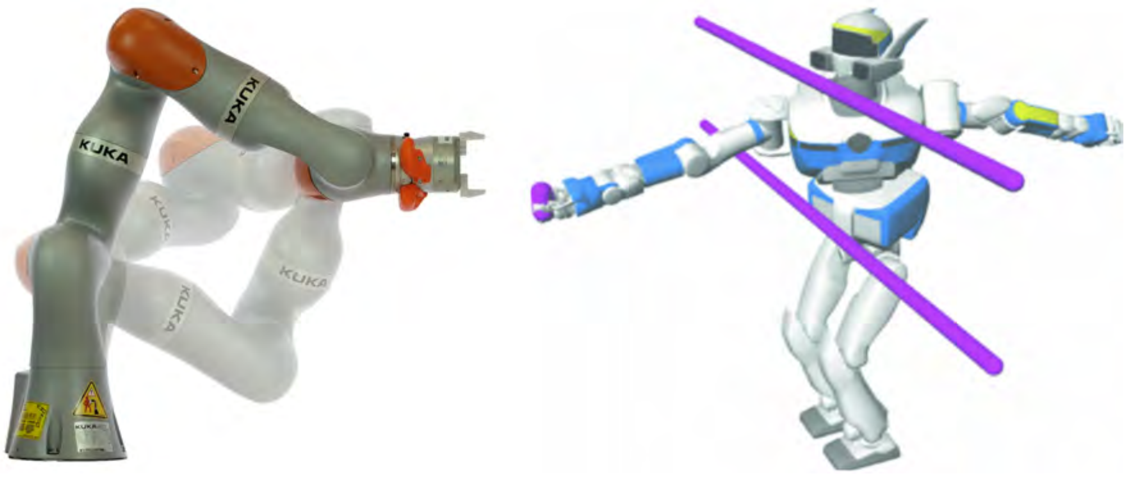


Figure 1.3 A KUKA *LBR iiwa* robot showing how the null space can be used to locate the end-effector in a specific pose with different configurations of the robot (KUKA, 2018) (left), and a simulation of an HRP-3 where a main task is executed (grasp the purple object) with a secondary tasks consisting in not colliding with the purple bars and keeping the body balanced (right).

The goal of this thesis is to further expand the knowledge of the robot null space and its application by studying its capabilities in two new applications implemented in two different robotic systems. The first robotic system is a humanoid robot designed for social purposes. The second robotic system, which is itself a novelty within the existing types of robotic systems, consists in a mobile manipulator teleoperated by an operator and a UAV with a camera attached to it. Both systems can be seen in Figs. 1.4.

As explained, the first robotic system consists on a humanoid robot. The intent is to use its null space to implement a secondary task that can improve its social skills, which, to the authors knowledge, is a task that has never been implemented as a secondary task in any kind of robot. In particular, the robot is studied to execute a main task consisting on specific gestures of its end-effectors while the null space is used for the conveyance of emotions to users as a task with lower priority.

The second robotic system is designed to allow a user to teleoperate a mobile manipulator in an unbounded scenario. The operator uses a haptic device to command the robot end-effector, which can also exert back a force to him/her, thus enhancing the teleoperation experience. The user gets visual feedback of the scenario from the camera attached to the UAV. The main purpose of this setup is to allow the operator to control the robot end-effector without the need to explicitly consider the camera, thus increasing the transparency of the system. In order to ease the teleoperation, the null space of the system is studied and exploited to ease the teleoperation of the end-effector of the mobile manipulator.

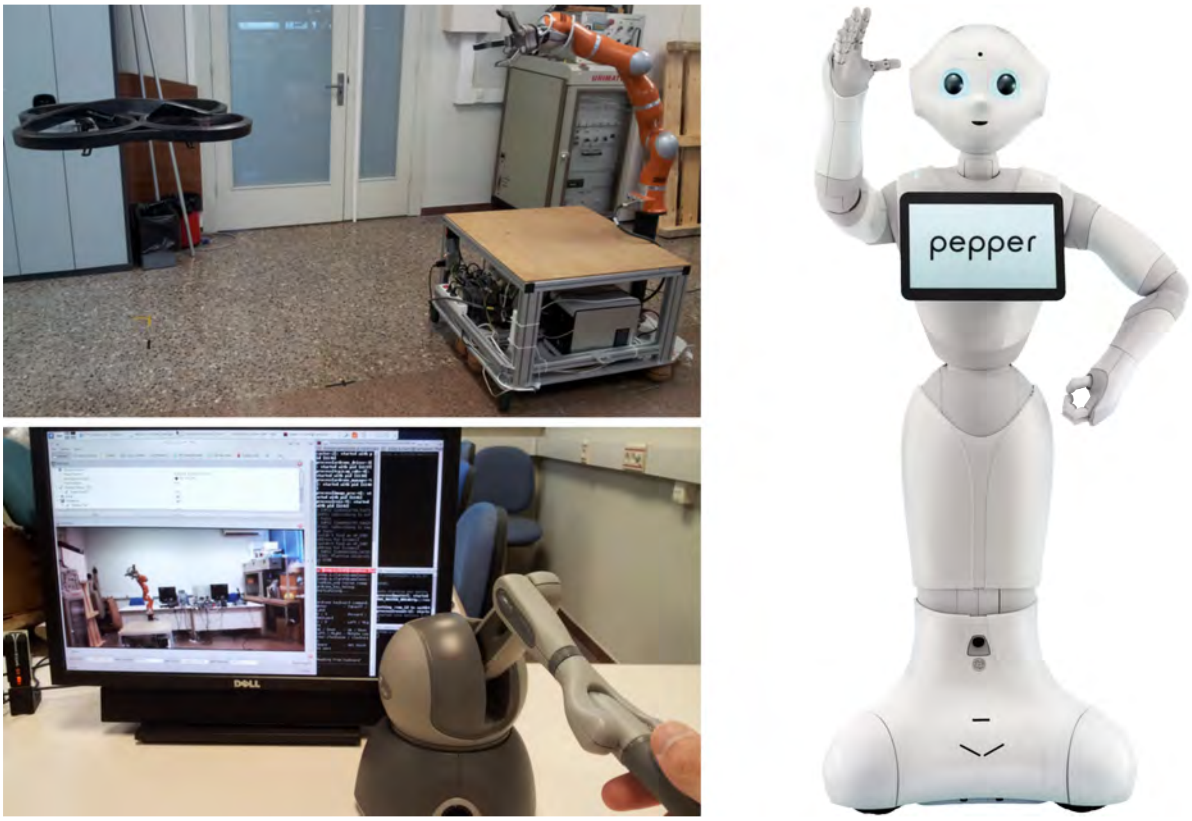


Figure 1.4 The robotic system consisting on a mobile manipulator and a free flying camera (top left), and the user perspective of the scenario as seen through the UAV camera along with the haptic device used to command the system (bottom left). The Pepper robot by SoftBank Robotics (2018), the humanoid robot used to implement the emotion conveyance task (right).

More concretely, the two main objectives addressed in this dissertation are presented below in more detail:

1. Exploiting the redundancy of a humanoid robot to convey emotions to the users as a lower priority task, while the main task is the command of one or many of its end-effectors. This main objective can be subdivided in the next subgoals:
   - To propose a general mapping from a space suited to describe emotions to a space of kinematic features appropriate for a humanoid robot.
   - To transform the proposed mapping in such a way that it can be fed to a multiple task priority system in order to execute the emotional motions on the robot as secondary tasks.
   - To implement the solution in a real robot.
   - To evaluate the proposed mapping through a user study and an analysis of the goodness of the conveyance depending on the intended emotions.

2. Introducing a novel teleoperation robotic system to command the end-effector of a mobile manipulator, using a free flying camera as the camera attached to a UAV, and a haptic device, and exploit the redundancy to ease the teleoperation experience of the user. This can be divided in a number of sub-objectives:
   - To describe the teleoperation system at a kinematic level and define the constraints that are particular to it.
   - To propose a solution to command the mobile manipulator using the haptic device while removing the need to explicitly control the free-flying camera.
   - To exploit the redundancy of the system considering that the main task is to move the end-effector, both in position and orientation, and, if possible, add secondary tasks with lower priority to increase the teleoperation performance.
   - To implement and to evaluate the proposed solution in a real platform.

1.3 Outline of the Thesis

This dissertation is structured into four sections, which can be outlined as follows:

- The first three chapters are introductory. They give the motivation and problems addressed in this dissertation (the current Chapter 1), as well as the state of the art (Chapter 2), and the corresponding theoretical background (Chapter 3).

- Part I presents the solution to the problem of conveying emotions to a user using the body motions of a humanoid robot as a secondary task. In Chapter 4 a model is presented that allows the transformation of an emotion defined in a three dimensional space to its corresponding kinematic motions of the robot, and that are ultimately fed to its null space. Finally, Chapter 5 presents a user study using a Pepper robot which evaluates the different aspects of the proposed solution.

- Part II introduces a novel robotic system for teleoperation purposes in which the operator commands a mobile manipulator while getting visual feedback of the scenario through a camera attached to a UAV. It is divided in five chapters. Chapter 6 develops a workspace mapping solution which allows the operator to command the mobile manipulator in an unbounded workspace while commanding the point of view of the camera in a natural way. The next chapter, Chapter 7, presents an enhancement on the continuous inverse, which removes potential oscillations when accounting for joint limits as a secondary task. Chapter 8 presents a solution to use the remaining degrees of freedom of the mobile manipulator to give the operator a good view of the object. Chapter 9 extends the algorithm previously presented in Chapter 6 to reduce the motion required by the UAV in order to ease the teleoperation for the user. Lastly, Chapter 10 presents the experimental validation of the proposed algorithms in a real setup with a mobile manipulator and a UAV as the free-flying camera.

- To sum up, Chapter 11 presents the conclusions of the thesis, outlining the main contributions and directions for further research.

Chapter 2

State of the Art

I visualize a time when we will be to robots what dogs are to humans, and I'm rooting for the machines.

Claude Shannon

SUMMARY: The current chapter provides the state of the art of the topics of this dissertation, first regarding the null space and the task priority formulations, next surveying the different types of robotic systems and current teleoperation systems which integrate a UAV and a mobile manipulator, and, finally, the current state of emotion conveyance using robots and the different means used to do so.

2.1 Robotic Systems

Several types of robotic systems have been devised in the robotic community, whether consisting of a single robot, a few robots cooperating, or a large number. The most relevant to this dissertation are presented below.

The simplest robotic system can be considered to be the classic industrial manipulator with six axis, which is the minimum necessary number to locate the end-effector at any position with any orientation in the workspace. Its natural extension is to increase the number of joints, thus giving the robot the capability to use its redundancy to handle more tasks simultaneously (Siciliano, 1990). Another extension are mobile manipulators, which add wheels, whether with holonomic or non-holonomic constraints, to allow the manipulator to reach positions far from its bounded workspace. The addition of wheels can be seen as adding joints to the system, thus far increasing the system complexity and its redundancy (Yamamoto and Yun, 1992; Khatib et al., 1996).

Even though several attempts can be found through history (Hum, 2018), humanoid robots are one of the last types of robots to be fully embraced by the robotics community (Sugano and Kato, 1987; Hirai et al., 1998). They are probably among the most promising. Its main advantage lies in the fact that they are designed to have the appearance and functionality of humans. They should, thus, be capable of using our tools, machines and infrastructures. Behind the utopia of a society where persons and human-shaped robots naturally interact, lies an enormous complexity which first needs to be mastered. This complexity is spread across multiple interrelated disciplines, like kinematics and dynamics (García et al., 2017; Nishiwaki et al., 2002), Human-Robot Interaction (HRI) (Goodrich and Schultz, 2007; Tapus et al., 2007), and many more.

Adding more robots to the system allows to tackle tasks that are not possible to be fulfilled by a single robot. This can be done by adding a single robot of the same type or a different type of robot (Tanner et al., 2003; Smith et al., 2012; Torres et al., 2009). In other scenarios, it may even be necessary to control a large number of robots in order to fulfill the desired goal (Dudek et al., 1993; Belta and Kumar, 2004). In any case, adding more robots dramatically extends the capabilities of the system at the cost of an increasing complexity in terms of its communication and coordination.

Interest in unmanned air vehicles has been present for decades (Howard and Kaminer, 1995), but it has been recently that they have become affordable, and an explosion in the robotics community has occurred (Goerzen et al., 2010). Works addressing the coordination of Unmanned Ground Vehicles (UGV) and UAVs can be found. For instance, Rao et al. (2003) use visual servoing as a framework to control the motion of a UGV from the camera mounted on a UAV which is flying above it. Coordinated groups of UAVs and UGVs can be found in (Tanner and Christodoulakis, 2006; Tanner, 2007), where control algorithms are proposed for reconnaissance and localization purposes; while a survey on this topic can be found in (Duan and Liu, 2010). Saska et al. (2012) and Saakes et al. (2013) both present systems in which the UAV is able to land on a UGV by the use of its vertical camera, while in the works (Wenzel et al., 2011; Bi and Duan, 2013; Hui et al., 2013) tracking of the UGV trajectory is added. Differently, Harnett et al. (2008) present a surgical robot and a UAV integrated together to support wireless, robotic telesurgery, in such a way that the UAV is used to support the wireless communication.

2.1.1 Teleoperation

One of the first practical use of robots was in a teleoperation scenario consisting of a user acting as a master and commanding a robot as a slave (Goertz, 1952). Robots in teleoperation setups have been widely studied since then due to the appearance of particular problems of this subfield, like the interaction between the robot and the operator (Nielsen et al., 2007) or delays in the communication (Nuño et al., 2017).

The vision camera is of paramount importance in a teleoperation scenario. The two most adopted camera configurations are known as *eye-in-hand*, where one or more cameras are rigidly attached to the end-effector (Loughlin and Hudson, 1983; Allen et al., 1993); and *eye-to-hand*, where the camera are fixed (Mallem et al., 1992; Kofman et al., 2005). Regarding the *eye-to-hand* configurations, early works can be found in virtual scenarios where different view points are compared and assessed through user studies (Ware and Osborne, 1990). Other works point out the advantages of automatically controlled cameras to enhance the performance of the teleoperation task (Hughes et al., 2003). Given a set of fixed cameras, a dynamic point of view that deals with the orientation can be found for robot teleoperation in the work presented by Pérez and Rosell (2011), where the active camera switches automatically to get a proper view of the remote task execution. Differently, *visual servoing* includes the camera in the control loop, automatically setting its motion as a result of the minimization of the task error (Espiau et al., 1992).

Workspace mapping algorithms map the input commands of the operator, usually from a haptic device or a joystick, to the robot workspace. Thus, they are a key component in teleoperation systems by allowing the user to operate in the workspace. The mapping is usually done at the position level, whether associating the haptic position and the robot position (*position-position* mapping), or with the robot velocity (*position-linear velocity* mapping), or by means of an hybrid mapping. This transformation has to account for the scaling and the unbound workspace for mobile robots. For instance, Conti and Khatib (2005) propose to add an imperceptible drift of the point of view on a position-position mapping to explore virtual scenarios with a haptic device, thus aiding the operator to locate the object in the centre of the image.

Few works specifically address the workspace mapping in the teleoperation of a mobile manipulator. Herdocia et al. (2012) compare three different master-slave motion coordination schemes to teleoperate a mobile manipulator with a robot arm with six degrees of freedom: workspace clutching, that is, conveniently and temporarily decoupling the motion between the slave and master in order to center the haptic in the middle of its workspace; an hybrid scheme in which a position-velocity mapping is activated only in the vicinities of the haptic workspaces; and a pure position-velocity control. Frejek and Nokleby (2013) use a position-linear mapping to command the end-effector of a mobile manipulator, while using the redundancy to handle singularities. Finally, a work that proposes a mapping to command a UAV using a haptic device can be found in (Ruesch et al., 2012).

To the author knowledge, no work in the literature can be found which studies a robotic system consisting on a mobile manipulator and a UAV, and, in particular, not in a teleoperation scenario where a camera attached to the UAV can be used to send visual feedback of the manipulator workspace to the operator. Thus, the question arises of how can such a system be exploited in order to give a proper teleoperation experience to the user, and how to take advantage of the system redundancy in order to improve this experience and fulfill the task.

2.2 Exploiting the Redundancy

In its beginnings, robots where designed as mechanical tools which could be programmed in its joints to execute a predefined task (Bejczy, 1974). But, as it was more appropriate to define the task as a trajectory of the end-effector, efforts where taken to describe the task at that level (Luh et al., 1980). In cases where the task dimension was lower than the available joints of the robot, whether the robot was redundant or not, it became possible to program more than one task (Liégeois, 1977; Klein and Huang, 1983), and several works addressed this problem and devised new tasks that could be simultaneously executed along the main task.

One approach to exploit the redundancy of a robot is to use the Weighted Jacobian (Whitney, 1969; Hollerbach and Suh, 1987; Chan and Dubey, 1995). In the Weighted Jacobian method, a weight matrix is multiplied to the Jacobian in such a way that, if there are DOFs available, an objective function can be minimized. This minimization function can be designed through proper weights, thus allowing the execution of a secondary task associated to the function. This concept is still used nowadays in combination with other methods (Yoshida et al., 2006; Cocuzza et al., 2011), though it does not by itself alone allow the addition of more than one secondary task.

Given a redundant robot, another approach taken to execute multiple tasks is the Augmented or Extended Jacobian (Egeland, 1987; Siciliano and Slotine, 1991). In this approach, new tasks are added as rows to the Jacobian matrix, thus actually imposing additional constraints to be executed along with the end-effector task. These constraints can be imposed in two ways: as motions of other end-effectors of the robot, thus fundamentally adding the rows of the Jacobians of the two or more end-effectors (Sciavicco and Siciliano, 1988), or as objective functions that need to be minimized, which derivatives with respect to the joints can be used to extend the robot Jacobian (Baillieul, 1985; Tchoń, 2006). These methods are still currently in use, usually in combination with other techniques (English and Maciejewski, 2000; Namvar and Aghili, 2005; Zanchettin and Rocco, 2012), due to the fact that collisions between the different tasks may appear which make one or many of them infeasible, and that the resulting Jacobian matrix may be singular even in configurations at which the end-effector Jacobian is full-rank.

The most widely used approach to exploit the robot redundancy is the null space of the robot Jacobian. In this method, a potential function is projected onto the kernel of the main task, that is, by only allowing to be available to the secondary task the space of joint velocities that can be executed without affecting the main one (Liégeois, 1977; Dubey et al., 1988). Interestingly, this approach guarantees the fulfillment of the first task, and so the secondary task is only executed if enough DOFs remain in the system, thus avoiding collision between tasks.

Several secondary tasks have been implemented using the Jacobian null space approach, like joint limit avoidance (Liégeois, 1977), singularity avoidance (Nemec and Zlajpah, 2000), or self-collision avoidance (Maciejewski and Klein, 1985), torque minimization (Hollerbach and Suh, 1987), simultaneous positioning of multiple end-effectors (Baerlocher and Boulic, 1998), limiting the range of the joints (Mansard et al., 2009a), manipulability optimization (Yoshikawa, 1985), center of mass positioning for humanoid robots (Baerlocher and Boulic, 2004), adding compliant behavior to the joints (Peng and Adachi, 1993), among others.

A generalization of the null space formulation can be found in the Task Priority frameworks, which, as long as there are available DOFs, potentially allow any number of secondary tasks with different levels of priority (Nakamura et al., 1987; Siciliano and Slotine, 1991; De Schutter et al., 2007). A variation is presented by Chiaverini (1997), where a task priority formulation that avoids algorithmic singularities is presented. This approach has been widely applied different types of robotic system, like manipulators, mobile manipulators (White et al., 2009) and humanoid robots (Sentis and Khatib, 2005).

One of the drawbacks concerning the null space approach is that it assumes a rigid hierarchy of tasks. This can become problematic if a modification of the priorities is needed. This can be the case, for instance, when an obstacle is getting dangerously close to the robot, that has to track a point with the end-effector as the primary task. Some works address this problem by blending together the trajectories corresponding to the different tasks, thus avoiding the discontinuities that could appear during a sudden switch between the tasks (Brock et al., 2002; Seto et al., 2005; Sugiura et al., 2007). In particular, the specific continuous activation and treatment of unilateral constraints in the hierarchy of tasks has also recently been of interest (Chan and Dubey, 1995; Marchand et al., 1996; Raunhardt and Boulic, 2007a; Mansard et al., 2009a,b), as the original null space priority approach was only able to indirectly address them by defining an appropriate objective function, and only as the task with lowest priority.

2.3 Using a Robot to Convey Emotions

Different approaches have been devised in the Human-Robot Interaction field to convey emotions to the user using a robot, whether human-like or not.

2.3.1 Voice Intonation

An interesting approach is the modulation of the voice intonation, an obvious source of emotional information in humans (Scherer, 1995). Several works take this approach as a single approach or as a complement.

For instance, a very interesting work can be found in (Pierre-Yves, 2003), which studies the most relevant features of the voice related to emotion conveyance and proposes an algorithm that manipulates the rhythm of the voice in a simple manner to convey the desired emotional tone.

Focusing in vocal prosody, that is, the elements of the speech which are characteristic of syllabus and longer units of the speed, Crumpton and Bethel (2015, 2016) review this area of linguistics to assess its potential for emotion conveyance, as well as pointing key aspects which should be taken into account during a human-robot communication scenario.

While the previous works are general and purely algorithmic, Lim et al. (2011, 2012) use the voice to enhance the emotion conveyance potential of a robot. In this work, the motions of a Nao robot are combined with the intonation of the voice to carry emotional information. Given a voice signal which conveys a predefined emotion, the proposed framework is able to map specific features of the input voice signal to kinematic features of the robot, which are used to generate new configurations in real-time, thus improving the emotion conveyance rate of the system.

The importance of the emotions elicited by robots can also be seen in (van Straten et al., 2017). With the goal to develop a robot-mediated therapy for children with autism disorder, this study analyses both the effect of the intonation of the robot voice by using two tones (monotonous and normal), and of the appearance of the robot body (whether it is more mechanical or human-like), during the interactions between the robots and autistic children. As a result, the importance of the coherence between the different emotional sources from the robot is stated.

2.3.2 Head Motion and Facial Expression

Another common approach to convey emotional states from a robot is by using its head, whether it is by using its motion or by the expression of the face.

For example, Berns and Hirth (2006) present a robot head called Roman. The Roman robot consists solely of a head, and it contains eleven servomotors which allow it to express six basic emotions and combination of them using its face. The desired facial expressions are commanded through a behaviour based control, and the mappings from the face servomotors to the desired emotions are presented, along with a user study to evaluate its goodness.

Other works, also in the animation community, build on previous studies stating the importance of the motions of the head in human-human communication (P. and WV., 1967; Sander et al., 2007; Samanta and Guha, 2017) to derive models so that the head motion can be integrated with the facial expression, the motion of the hands, and the speech, in order to put more emphasis on the emotional content. For instance, in (Busso et al., 2007a,b), a Hidden Markov Model is built to model the specific dynamics of emotional head motion sequences for the emotional categories of sadness, happiness, anger, and a neutral state.

Besides expressing emotions through the facial expression and the robot motion, the gaze is also able to carry a significant amount of emotional information, as stated in many works (Adams Jr. and Kleck, 2005; Sander et al., 2007; Rigato and Farroni, 2013). Lance and Marsella (2008) use the gaze to generate emotionally expressive head and torso movements during gaze shifts through a combination of temporal scaling and spatial transformation parameters, which transform an emotionally neutral gaze shift into an emotional one directed at a desired target.

2.3.3 Expressing Emotions through Body Motions

Conveying emotions through the body motions has been studied in the animation community for decades (Johnston and Thomas, 1981), and from the computer graphics point of view since the 90' (Unuma et al., 1995; Amaya et al., 1996). Only since the beginning of this century efforts have started to grow in the robotics community in that direction.

Kulic and Croft (2007) present the first work to measure the affective response of users to human-scale physical robot motions. The results of this work are important because they point out the fact that robot motions may create anxiety in the users, and that there is a need to design robot motions that ease the interaction between the human and the robot.

In (Saerbeck and Bartneck, 2010) a motion pattern generation over robot trajectories is presented. The proposed solution shows that by varying the amount of velocity and acceleration of the robot trajectory it is possible to transmit different emotional states to the user. The proposed solution has been implemented in a Roomba robot and further shown to be extendible to other robots.

Studies with elder people can also be found in (Nomura and Nakao, 2010), where a comparison is presented between the responses of the elder and younger people to the emotional motions of the robot. Interestingly, the results point out the dissimilar responses that can be triggered in different populations, thus making apparent the importance of biases, which can be age-related, like in this study, or due to social or cultural background. The study also shows the growth in interest on the responses of the users to the intended conveyance of emotions from the robot.

New emotional expressions are generated in (Beck et al., 2010) by interpolating between specific robot configurations. These configurations are associated to particular emotional expressions, which in turn are based on the Circumplex model of emotions (Russell, 1980). This model plays an important role, as it allows to mathematically define emotions as points in a two dimensional space, in such a way that different points correspond to different emotions.

Another approach taken by other works is to add an offset to the motions of a known gesture, whereas in terms of position and/or its derivatives. For instance, Lim et al. (2011) propose to modify the final position and the velocity of a base gesture depending on the amount and type of emotional information intended to be conveyed by a humanoid robot. In this particular study, the amount of offset varies depending on features extracted from the voice signal.

Similarly, in (Nakagawa et al., 2009) a correspondence between emotions from the Circumplex model and an offset is presented. This offset is added to the intermediate configurations and velocities of a given trajectory, in order to incorporate affective nuances to the latter, in such a way that the trajectories are parametrized by the two dimensions needed to describe the emotions of interest.

The work of Sharma et al. (2013) propose the generation of UAV trajectories that carry emotional information. The novelty of this approach is its use of the Laban Effort System as the starting point to generate the emotions. The Laban Effort System is a standard method to interpret human motions commonly used in performing arts like ballet, and an initial mapping between its features and the motions of the robot, as well as some firsts guidelines for the robotic community, are presented.

The field is capturing increasing attention and a survey can be found in (Karg et al., 2013), which covers both the automatic recognition of human affective expressions and the generation of expressive behaviours for virtual avatars and robots. Among its many points of interest, this work introduces a suitable movement categorization based on existing and accepted movement notations, as well as an important corpus of studies from psychology which help ground the discipline.

Besides the amount of research done by the robotic community to convey emotions using the body motions of the robot, the available approaches only deal with this problem in order to convey a certain emotional state as their only task. These approaches limit their usage in situations where the robot is only required to do this single task, but would fail to be useful in situations where the robot is already executing a main task and a need to convey an emotion with the body motions is detected. In that case, at most, a decision could be made in order to chose between the two tasks, but it would not be possible in general to execute both tasks simultaneously and successfully.

Chapter 3

Modeling

The robbed that smiles, steals something from the thief.

William Shakespeare

Summary: With the intent to give the reader the theoretical foundations that ground this dissertation, the current chapter provides the necessary theoretical background of task priority formulations and modeling of emotions.

3.1 The Jacobian Null Space and The Task Priority Formulations

Given a robot with joints $\boldsymbol{q} \in \mathbb{R}^n$ and a task specified by the vector $\boldsymbol{x} \in \mathbb{R}^m$, its *direct kinematics* mapping, $\boldsymbol{k}(\cdot)$, relates the values of the joint variables with the values of the desired task $\boldsymbol{x}$, which usually is the end-effector position and orientation, as

$$\boldsymbol{x} = \boldsymbol{k}(\boldsymbol{q}). \tag{3.1}$$

A robot is said to be *redundant* if the number of degrees of freedom (active joints) exceeds the number of variables needed to determine the task. In this regard, it is important to note that, as a consequence, the redundancy of the robot can only be defined with respect to a specific task. For instance, a robot with six joints can be considered redundant for a task consisting on locating the position of its end-effector, as it requires three values to set the task, thus still having three available joints to execute any other secondary task.

Many approaches to use the redundancy of the robot to solve the direct kinematics are based on the inversion of the mapping defined by the Jacobian matrix, $\boldsymbol{J} \in \mathbb{R}^{m\times n}$. This mapping relates the joint velocities, $\dot{\boldsymbol{q}}$, with the velocity of the task, $\dot{\boldsymbol{x}}$, given a joint configuration $\boldsymbol{q}$, as

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}.$$

The pseudoinverse of $\boldsymbol{J}$ can be taken prior to integrate $\dot{\boldsymbol{q}}$ to obtain the value of $\boldsymbol{q}$ that satisfies Equation 3.1

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}\dot{\boldsymbol{x}} \tag{3.2}$$

where $\boldsymbol{J}^{+}$ can be computed as $\boldsymbol{J}^{+} = \boldsymbol{J}^{T}\left(\boldsymbol{J}\boldsymbol{J}^{T}\right)^{-1}$ for non-singular configurations, or more generally using the SVD decomposition. Note that $\boldsymbol{q}$ has been dropped from $\boldsymbol{J}$ for ease of visualization, as will be from now on in this dissertation.

Expression 3.2 makes implicit use of the robot redundancy by selecting, among all possible solutions that fulfill Equation 3.1, the one that minimizes the norm of $\dot{\boldsymbol{q}}$.

A suitable joint velocity, $\dot{\boldsymbol{q}}_0$, can be projected onto the Jacobian null space matrix, $\boldsymbol{N} \in \mathbb{R}^{n\times n}$, to take more advantage of the redundancy of the robot as

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}\dot{\boldsymbol{x}} + \boldsymbol{N}\dot{\boldsymbol{q}}_0 = \boldsymbol{J}^{+}\dot{\boldsymbol{x}} + \left(\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J}\right)\dot{\boldsymbol{q}}_0$$

such that the task $\boldsymbol{x}$ will be fulfilled with highest priority and, if there any degrees of freedom left, the task associated with the input $\dot{\boldsymbol{q}}_0$ will be totally or partially fulfilled with lower priority, that is, without affecting the higher priority task.

Extending the Jacobian null space approach, the task priority formalism gives a framework to manage different tasks at different levels of priority. Given the desired highest and lowest priority tasks $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively, the vector of joint variables of the mobile manipulator, $\boldsymbol{q}$, that satisfies both tasks with the given levels of priority can be computed through the well-known formula (Siciliano and Slotine, 1991)

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{+}\dot{\boldsymbol{x}}_1 + (\boldsymbol{J}_2\boldsymbol{N}_1)^{+}\left(\dot{\boldsymbol{x}}_2 - \boldsymbol{J}_2\boldsymbol{J}_1^{+}\dot{\boldsymbol{x}}_1\right) \tag{3.3}$$

where $\boldsymbol{J}_i$ is the Jacobian of task i, such that the general velocity equation $\dot{\boldsymbol{x}}_i = \boldsymbol{J}_i\,\dot{\boldsymbol{q}}$ holds, and $\boldsymbol{N}_i = \boldsymbol{I} - \boldsymbol{J}_i^{+}\boldsymbol{J}_i$ is the projection matrix onto the null space of task i. Then, $\boldsymbol{q}$ can be computed by integrating $\dot{\boldsymbol{q}}$.

This idea can be extended to account for any number of tasks, t, with decreasing priority, by recursively applying the following formulation (Baerlocher and Boulic, 1998):

$$\begin{aligned}
\dot{\boldsymbol{q}}_1 &= \boldsymbol{J}_1^{+}\dot{\boldsymbol{x}}_1 \\
\dot{\boldsymbol{q}}_i &= \dot{\boldsymbol{q}}_{i-1} + \left(\boldsymbol{J}_i\boldsymbol{N}_{i-1}^{A}\right)^{+}\left(\dot{\boldsymbol{x}}_i - \boldsymbol{J}_i\dot{\boldsymbol{q}}_{i-1}\right) \\
\dot{\boldsymbol{q}} &= \dot{\boldsymbol{q}}_t
\end{aligned}$$

with the projection operator $\boldsymbol{N}_i^A$ also computed recursively as

$$\boldsymbol{N}_1^A = \boldsymbol{I} - \boldsymbol{J}_1^+ \boldsymbol{J}_1$$
$$\boldsymbol{N}_i^A = \boldsymbol{N}_{i-1}^A - \left(\boldsymbol{J}_i \boldsymbol{N}_{i-1}^A\right)^+ \left(\boldsymbol{J}_i \boldsymbol{N}_{i-1}^A\right).$$

In practice, the number of degrees of freedom of the robot will limit what tasks are actually executed. Thus, a redundant robot with ten joints and the goal to execute three tasks of six, six and three DOFs with different levels of priority, will only be able to execute at most the first task and solve four DOFs of the second task, while leaving the third task untouched.

3.2 Modeling Emotions

The Circumflex Model of Affect

The Circumplex Model of Affect is a model of emotions representation developed by James Russell in which emotions are represented as points in a circular space of two dimensions (Russel, 1980; Posner et al., 2005). These two dimensions are arousal (or activation) and pleasure (or valence), corresponding to the vertical and horizontal axis, respectively (Fig. 3.1), while the center of the circle represents a neutral valence and a medium level of arousal.

Unlike theories of basic emotions which claim that there is a unique and independent neural system for each emotion, the Circumplex Model of Affect proposes that affective states are the result of the combination of only two independent systems, one corresponding to the arousal and the other to the pleasure. In this regard, dimensional models of emotions as the Circumplex model consider affective experiences as a continuum of highly interrelated and often ambiguous states.

The Circumplex Model of Affect has been shown to provide a reasonable representation for findings of previous studies, and helps to explain current research and clinical findings, thus consolidating its use in the scientific community.

Moreover, its value not only lies as a mere theoretical representation of emotion, but it has been proven useful in the study of clinical psychiatry and psychology disorders, like mood and anxiety disorders (J. Larsen and Diener, 1992; Plutchik and Conte, 1997; Remington et al., 2000).

The PAD Emotional State Model

Following with a similar approach to the Circumplex Model of Affect previously explained, Mehrabian (Mehrabian, 1996) presents the Pleasure-Arousal-Dominance (PAD) Emotional State

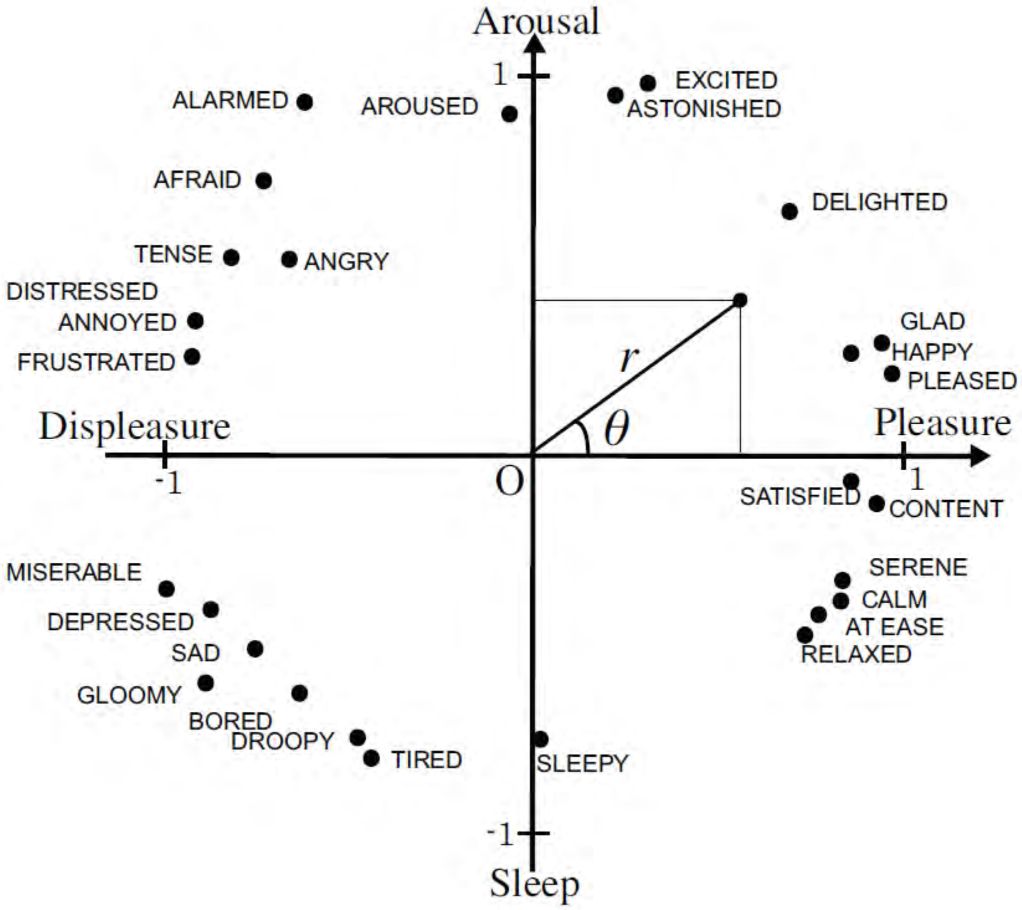


Figure 3.1 The Circumplex model.

Model, which condenses and formalises the body of knowledge of previous works. The PAD is a model for emotion representation in which each emotion can be represented using a point in a three-dimensional space such that the domain of each dimension is the interval $[-1, 1]$. The third proposed dimension, added to the previous set of two dimensions, *i.e.* arousal and pleasure, is dominance (or its opposite, submissiveness) (Fig. 3.2).

As stated in (Mehrabian, 1996), the three Pleasure-Arousal-Dominance dimensions account for *nearly* orthogonal axis. They provide the framework for a sufficiently comprehensive description of emotional states.

This claim is supported by an analysis of different studies from several authors (Osgood et al., 1975; Shaver et al., 1987), which ultimately lead them to explicitly describe different personality traits found in the literature as a linear combination of pleasure (P), arousal (A) and dominance (D). For instance, Extroversion (Eysenck and Eysenck, 1975) and Nurturance

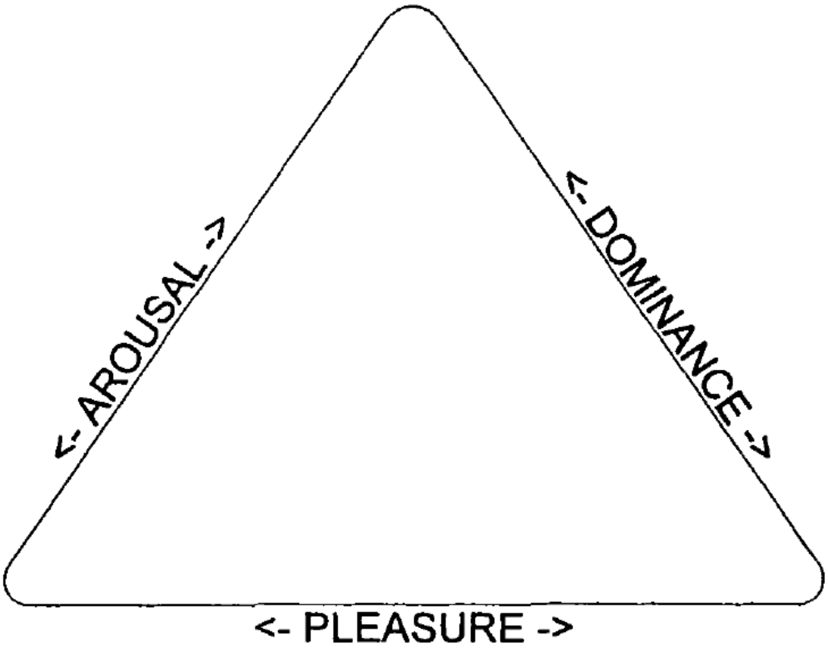


Figure 3.2 A representation of the PAD Emotional State Model.

(Wessler and Loevinger, 1969) can be decomposed as

$$\begin{aligned}\text{Extroversion} &= 0.21\ P + 0.17\ A + 0.50\ D\\ \text{Nurturance} &= 0.41\ P + 0.12\ A + 0.17\ D.\end{aligned}$$

Interestingly, the PAD three-dimensional space can be subdivided by eight octants, whether we are in the positive axis of the pleasure dimension ($+P$) or the negative ($-P$), and so on, each of which can be associated with a particular temperament, and such that opposite octants are naturally associated with opposite temperaments:

Exuberant	$(+P + A + D)$	vs.	Bored	$(-P - A - D)$
Dependent	$(+P + A - D)$	vs.	Disdainful	$(-P - A + D)$
Relaxed	$(+P - A + D)$	vs.	Anxious	$(-P + A - D)$
Docile	$(+P - A - D)$	vs.	Hostile	$(-P + A + D)$

One of the key points of the PAD space lies in its use of an underlying continuous space. This eases its use and coupling with other mathematical structures, very suited for affective and behavioural computing purposes, as can be seen in works like (Gebhard, 2005; Gunes et al., 2011).

The previously mentioned work (Gebhard, 2005) can be regarded as a paradigmatic use the

PAD space. In this work the PAD is used as the framework to generate emotions for avatar in virtual scenarios. 24 emotions are treated, ranging from more "obvious" emotions like fear and happiness, to more subtle ones like gratitude and resentment, and, notably, their corresponding pleasure, arousal and dominance values.

3.2.1 Measuring Emotions: The SAM Scale

The Self-Assessment Manikin (SAM) is a tool for measuring emotional response (Lang, 1980; Bradley and Lang, 1994), designed as an alternative to previous verbal self-report measures. It visually represents the three PAD dimensions of Mehrabian (1996), thus eliminating the problems related to verbal measures, and, due to its visual nature, allowing the participants to write their responses fast and intuitively.

It is composed of three scales, each one corresponding to a dimension of the PAD space (Fig. 3.3). In each scale, the corresponding PAD dimension is depicted with a graphic character arrayed along a continuous nine-point array: for *pleasure*, the character ranges from a smiling figure to a frowning one; for *arousal*, from a sleepy to an excited figure with the eyes wide opened; and for *dominance*, a small figure representing a feeling of being controlled to a very large figure representing an in-control mood.

The correlations between the SAM scores and the corresponding from the Mehrabian's own procedure to evaluate the PAD emotion space are impressively high for *pleasure* (0.94) and *arousal* (0.94), and slightly lower for *dominance* (0.66).

Some advantages over previous measures include:

- Due to the visual nature of the SAM scales, the ratings can be completed in few seconds.
- The short rating time allows subjects to complete multiple ratings with less fatigue.
- Both children and adults identify with the character depicted in the SAM scale and its different degrees of emotions expressed.
- The culture-free approach of the SAM scale allows it to be used in different countries and cultures.

The SAM scale can be used in a paper-and-pencil version, but there exists also electronic versions on a variety of different systems.

The guidelines to follow when carrying on a questionnaire with the SAM scale can be found detailed in (Lang et al., 2008).

A full-sized SAM scale can be found in Appendix B.

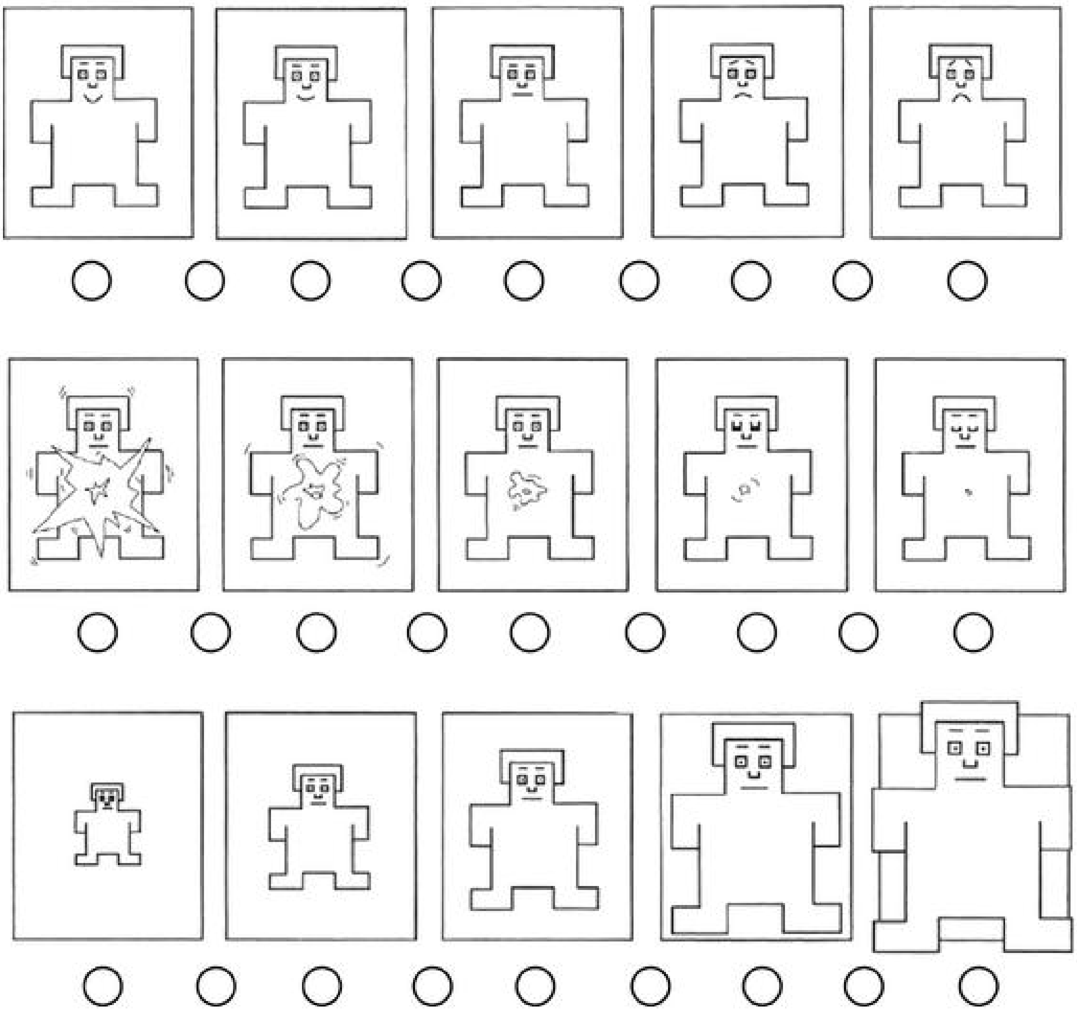
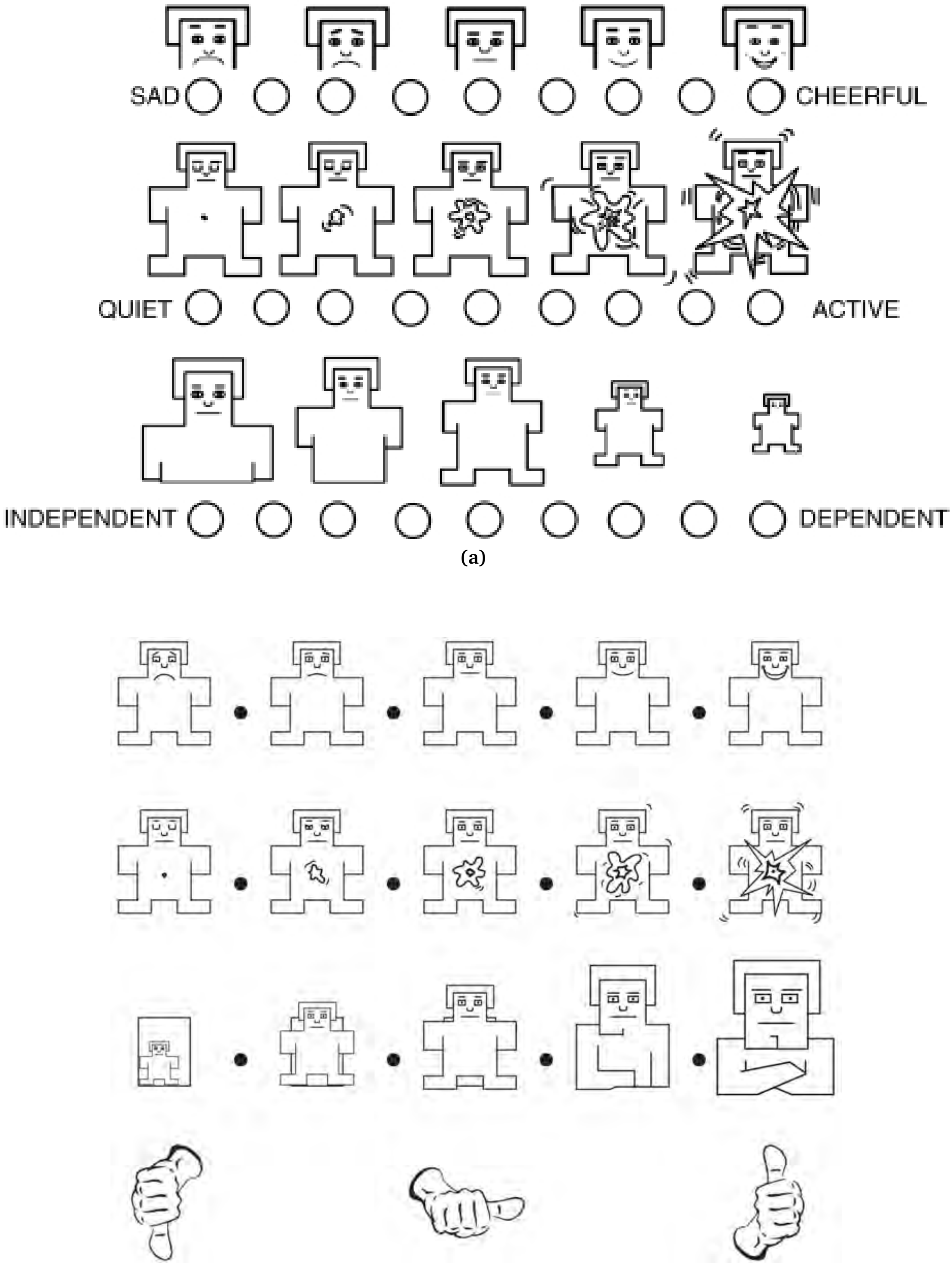


Figure 3.3 The SAM scale.

Different version of the SAM scale can be found, basically all conceiving the same information and serving the same purpose. Some examples are shown below in Fig. 3.4.

3.3 The Continuous Inverse

The continuous inverse is presented in (Mansard et al., 2009a), and it allows smooth transitions during the activation and deactivation of a task within a set of other tasks, whether they are in a transition state or not.



(a)

(b)

Figure 3.4 Other versions of the SAM scale, which can be found in the works of Schifferstein et al. (2011), (a), and Jirayucharoensak et al. (2014), (b).

The continuous inverse can be computed as

$$\boldsymbol{J}^{\oplus \boldsymbol{H}} = \sum_{P \in B(m)} \left(\prod_{i \in P} h_i \right) \prod_{i \notin P} (1 - h_i) \left(\boldsymbol{H}_P \boldsymbol{J} \right)^+ \tag{3.4}$$

where $B(m)$ are all the subsets composed of the m first integers, that is, the set of all combinations without repetitions that can be computed with the m first integers; $h_i \in [0, 1]$ are the task activation parameters, corresponding to the diagonal elements of the task activation matrix $\boldsymbol{H}$, such that when h_i is 0 the task i is deactivated, activated when it is 1, and in transition otherwise; and $\boldsymbol{H}_P$ is a diagonal matrix whose diagonal components h_{P_i} are 1 if $i \in P$ and 0 otherwise.

For instance, for a Jacobian, $\boldsymbol{J}$, composed of two tasks, $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$, with its corresponding activation parameters, h_1 and h_2, so that $\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_1^T & \boldsymbol{J}_2^T \end{bmatrix}^T$, the continuous inverse can be computed as

$$\boldsymbol{J}^{\oplus \boldsymbol{H}} = h_1(1 - h_2) \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{0} \end{bmatrix}^+ + (1 - h_1) h_2 \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{J}_2 \end{bmatrix}^+ + h_1 h_2 \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix}^+ . \tag{3.5}$$

Thus, the continuous inverse behaves exactly as a regular Jacobian when all the tasks are fully activated or deactivated, as can be seen by taking the limits of Eq. 3.5 for $h_i \to \{0, 1\}$. Interestingly, it also guarantees a continuous and smooth transition between different task activation states.

When dealing with multiple levels of priority, it is useful to generalize the continuous inverse to the right continuous inverse of a matrix $\boldsymbol{Q}$ activated by $\boldsymbol{W}$, $\boldsymbol{Q}^{\boldsymbol{W} \oplus}$, with $\boldsymbol{W}$ being a positive symmetric matrix whose singular values are all between 0 and 1. According to Mansard et al. (2009a), the right continuous inverse is necessary to compute the null space projection operator to obtain the joint velocity contribution of the lower priority levels. In the case of two tasks with different priority, similar to Eq. 3.3, this is

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\oplus \boldsymbol{H}} \dot{\boldsymbol{x}}_1 + \boldsymbol{J}_2^{\boldsymbol{N}_\oplus^1 \oplus} \left(\dot{\boldsymbol{x}}_2 - \boldsymbol{J}_2 \boldsymbol{J}_1^{\oplus \boldsymbol{H}} \dot{\boldsymbol{x}}_1 \right) \tag{3.6}$$

where the right continuous inverse of Q activated by W is computed as

$$\boldsymbol{Q}^{\boldsymbol{W} \oplus} = \left(\left(\boldsymbol{Q}^T \right)^{\oplus \boldsymbol{W}} \right)^T \tag{3.7}$$

with $\boldsymbol{Q}^T$ being the transpose of $\boldsymbol{Q}$, and $\boldsymbol{Q}^{\oplus \boldsymbol{W}}$, the left continuous inverse, computed as

$$\boldsymbol{Q}^{\oplus \boldsymbol{W}} = \left(\boldsymbol{U}^T \boldsymbol{Q} \right)^{\oplus \boldsymbol{S}} \boldsymbol{U}^T = \sum_{P \in B(m)} \left(\prod_{j \in P} \sigma_j \right) \prod_{j \notin P} (1 - \sigma_j) \left(\boldsymbol{H}_P \boldsymbol{U}^T \boldsymbol{Q} \right)^+ \boldsymbol{U}^T \tag{3.8}$$

with $\boldsymbol{U}$ from the SVD decomposition of $\boldsymbol{W} = \boldsymbol{U} \boldsymbol{S} \boldsymbol{U}^T$; σ_j as the j-th largest singular value of $\boldsymbol{W}$; and $\left(\boldsymbol{U}^T \boldsymbol{Q} \right)^{\oplus \boldsymbol{S}}$ being the continuous inverse of $\boldsymbol{U}^T \boldsymbol{Q}$ activated by the diagonal matrix $\boldsymbol{S}$ and computed following Eq. 3.4.

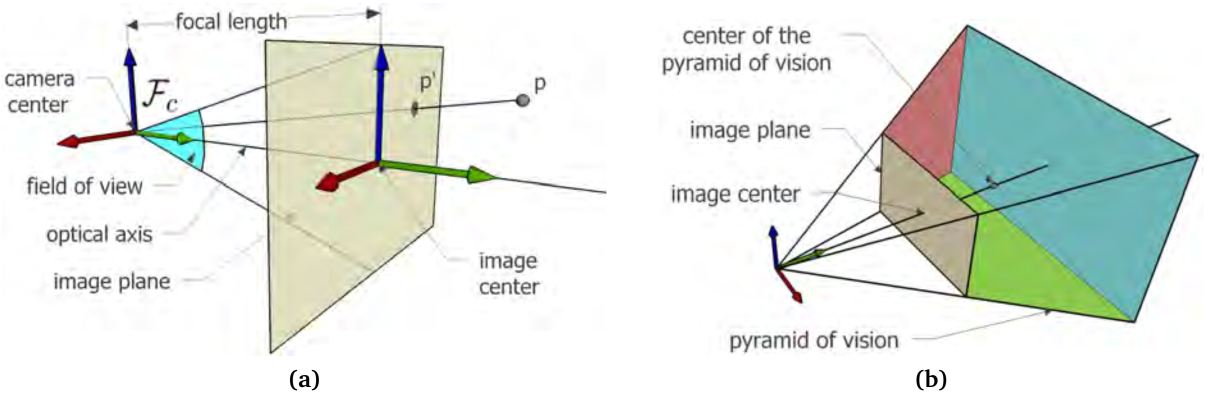


Figure 3.5 The pinhole camera model: a) the pinhole camera model coordinate system $\mathcal{F}_c$ with the camera center and the optical axis, the image center, the (vertical) field of view, and a point p projected in the image plane as p'; b) the pyramid of vision with its center.

Now the use of the continuous inverse for lower priority tasks can be generalized with

$$\boldsymbol{N}_{\oplus}^{i} = \begin{cases} \boldsymbol{I} - \boldsymbol{J}_i^{\oplus \boldsymbol{H}} \boldsymbol{J}_i & i = 1 \\ \boldsymbol{N}_{\oplus}^{i-1} - \boldsymbol{J}_i^{\boldsymbol{N}_{\oplus}^{i-1} \oplus} \boldsymbol{J}_i & \text{otherwise} \end{cases} \tag{3.9}$$

which, extending Eq. 3.6 to any level of priority

$$\dot{\boldsymbol{q}}_i = \begin{cases} \boldsymbol{J}_i^{\oplus \boldsymbol{H}} \dot{\boldsymbol{x}}_i & i = 1 \\ \dot{\boldsymbol{q}}_{i-1} + \boldsymbol{J}_i^{\boldsymbol{N}_{\oplus}^{i-1} \oplus} (\dot{\boldsymbol{x}}_i - \boldsymbol{J}_i \dot{\boldsymbol{q}}_{i-1}) & \text{otherwise} \end{cases} \tag{3.10}$$

allows the full computation of $\boldsymbol{q}$ given the set of tasks with a hierarchy.

3.4 The Pinhole Camera Model

The pinhole camera model describes the relationship between the coordinates of a 3D point and its projection onto the image plane of an ideal pinhole camera.

The pinhole camera model includes some elements that are relevant to this dissertation and can be seen in Figs. 3.5:

- A 3D orthogonal coordinate system, $\mathcal{F}_c$, with the origin at the camera aperture, called the *camera center*, and one of its axis, called the *optical or principal axis*, pointing to the viewing direction.
- *Image plane*: the plane where the 3D world is projected. It is orthogonal to the optical axis, and is located at a distance from the camera center called the *focal length*.

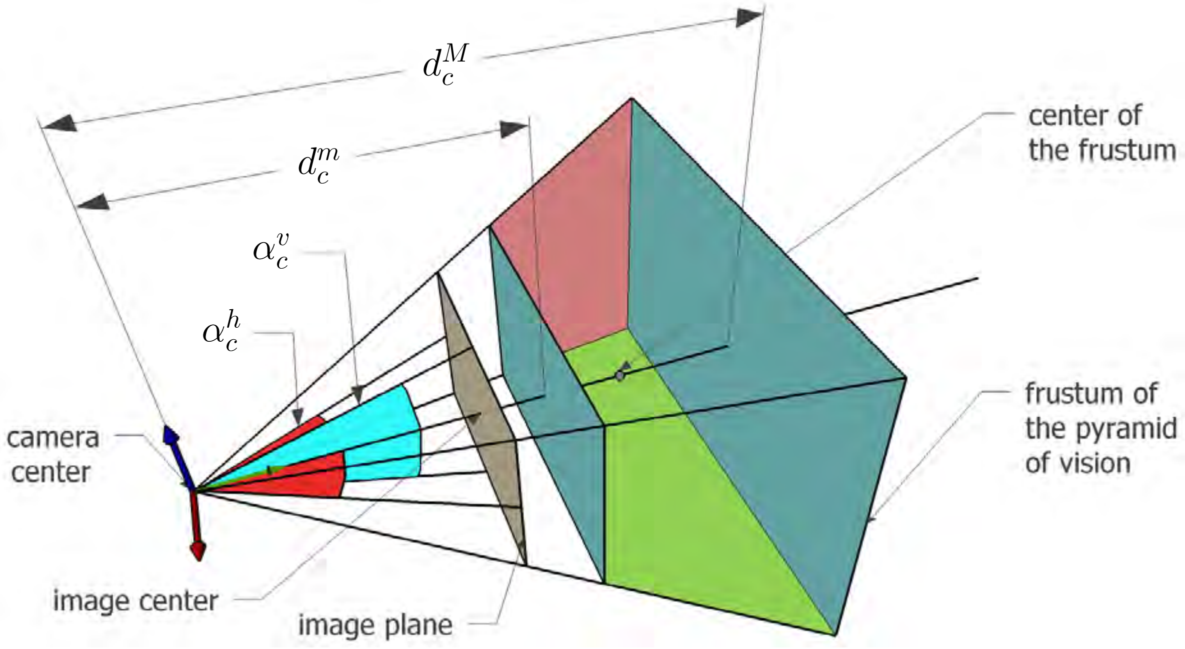


Figure 3.6 The frustum of the camera pyramid of vision defined by the distances d_c^m and d_c^M and the horizontal and vertical field of views, α_c^h and α_c^v, respectively.

- *Image center*: the intersection between the optical axis and the image plane.
- *Field of view*: measure of the extent of the observable world from the camera, computed as a solid angle with the endpoint as the camera center, and the sides starting from the endpoint to the boundaries of the image plane. It can be computed in the horizontal and vertical dimensions.
- *Pyramid of vision*: a four-sided pyramid that grows out of the camera center in the direction in which the camera is pointing (Fig. 3.5b). Its axis is the optical axis.
- *Frustum of a pyramid of vision*: a frustum is the portion of a solid that lies between one or two parallel planes cutting it.

  In particular, a frustum of a pyramid of vision V_c of the camera c can be defined by the parameters $\boldsymbol{p}_c^s, \boldsymbol{R}_c^s, \alpha_c^h, \alpha_c^v, d_c^m, d_c^M$ with:

  - $\alpha_c^h \in \mathbb{R}_{>0}$: the horizontal field of view.
  - $\alpha_c^v \in \mathbb{R}_{>0}$: the vertical field of view.
  - $d_c^m \in \mathbb{R}_{>0}$: the distance between the camera center and the face of the pyramid of vision parallel to the image plane which is closest to the camera center.
  - $d_c^M \in \mathbb{R}_{>0}$: the distance between the camera center and the face of the pyramid of vision parallel to the image plane which is furthest to the camera center.

  Fig. 3.6 depicts these concepts.

Given a point $\boldsymbol{p} = [p_x \quad p_y \quad p_z]^T \in \mathbb{R}^3$ expressed in the camera frame, its projection $\boldsymbol{p}_c \in \mathbb{R}^2$ onto the image plane can be computed as

$$\boldsymbol{p}_c = \frac{f_c}{p_z} \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

where $f_c \in \mathbb{R}_{>0}$ is the focal length.

Part I

Conveying Emotions Using the Null Space

Chapter 4

Emotion Mapping

If you look at the field of robotics today, you can say robots have been in the deepest oceans, they've been to Mars, you know? They've been all these places, but they're just now starting to come into your living room. Your living room is the final frontier for robots.

Cynthia Breazeal

SUMMARY: The main body of this chapter is devoted to presenting a mapping to transform an emotion described as a point in a three-dimensional space to a point in a space of kinematic features. Firstly, though, the humanoid robot used in this part of the thesis is introduced: the Pepper robot. Finally, the objective function that accounts for the kinematic features is shown, which can then be further used to feed a task priority algorithm.

4.1 Introduction

With industrial robots moving outside their cells, and with the advent of mobile manipulators and, even more significantly, humanoid robots, the robotics community has started to focus in the particular aspects of the new kind of interaction between humans and the artificial agent embodied in the robot. This has given birth to the so called Human-Robot Interaction (HRI) (Goodrich and Schultz, 2007; Corrales et al., 2012) and social robotics fields (Tapus et al., 2007; Breazeal, 2003; Gonzalez-Pacheco et al., 2011).

This increasing interest in HRI and social robotics (Baddoura and Venture, 2015) has naturally lead to the question of whether robots should be endowed with emotions and emphatic behaviours (Breazeal and Brooks, 2004, 2005; Salichs and Malfaz, 2006; Tapus and Mataric, 2007; Asada, 2015), both with the goal to study the key issues popping in this new found interaction, and as a mean to operate on it and enhance its outcomes in some desirable way.

In this regard, a future is devised in which humans and robots interact on a daily basis. Several studies show a link between productivity and the emotional state of the workers. For example, Oswald et al. (2009) show that happiness raises productivity, and, on the opposite side, anxiety impairs cognitive performance, which may ultimately lead to fatigue and degrade productivity (Derakshan and Eysenck, 2009). This works allow the robotic community to consider the management of the emotional state of the interaction between the human and the robot not only from a purely psychological perspective, but rather from a practical one, that is, in terms of the efficiency of the execution of the task.

Therefore, it may be of interest to program a robot such that it be capable to convey an appropriate emotional state to the user, and, as a consequence, increase the probability that he/she accomplishes his/her corresponding task. For instance, calmness could be conveyed by the robot in a situation where it was delivering a fragile load to the human. Also, a particular level of arousal could be conveyed in a scenario where a robot and a human were working cooperatively in the transport of a delicate object through a narrow passage, which is a task that could be considered to demand a certain degree of concentration. In such situations it would be desirable and even necessary to have all the possible means to induce a desired emotion to the user.

Most of the times, though, the robot may be already executing a task, which, given a particular scenario, can be considered to have the highest priority. In this situation, it can even be appealing that the robot achieves the conveyance of the desired emotion while the main task is being executed, and without degrading its performance. For example, the robot may be grasping a fragile load while approaching the user to deliver the charge.

The robot may sense that the user is in a low mood and consider that, given the fragility of the load, it is safer for the user to execute the task of carrying the load in a mood better suited for the task, thus raising the odds that the overall task is successfully carried on.

Other situations may arise where it can be interesting to convey emotions to humans. For instance, imagine a care robot that is carrying a plate and may sense that a patient is in a sad low and thus use its redundancy to convey positive emotions to him/her, which will ultimately positively affect its health (Salovey et al., 2000).

Also, it can be appropriate to be able to convey a certain emotion to increase the acceptance rate of the robot within the people it is interacting with (Hudson et al., 2016), as a mean to deal with the undesired uncanny valley effect (Mori et al., 2012). And, also, for a robot gesturing with the hands while speaking, it can be interesting to complement the emotional content of the voice with more emotional information, and thus use the redundancy to do so while not affecting the motion of the hands, which gestures are the main task.

As exposed below, and following the line of though developed in the previous paragraphs, current approaches do not allow robots to execute a task and simultaneously convey a desired emotion to users using their body motions. Neither there is, to the authors knowledge, any work that has studied the potential of the robot redundancy to convey emotions, nor, in particular, by means of the null space of the Jacobian of the robot.

This part of the current dissertation aims to make a step further in closing this gap.

4.2 The Pepper Robot

The Pepper robot is a social humanoid robot (Fig. 4.1) designed by Softbank Robotics [1]. It is composed by a main body with two arms on top of a mobile platform.

The body is divided by the waist in the torso and a lower part. The lower part can be seen as a single leg, connected to the platform with what resembles a knee. The mobile platform has three wheels and is omnidirectional. Therefore, it allows Pepper to move in any direction at any given moment without the need to do complex maneuvers. Overall, Pepper weights 28 kg and has a height of 1.2 meters, thus has the gross stature of an 8 year old child.

From a kinematic point of view the Pepper robot can be seen as a tree with three branches. These branches are the head and the two arms. All the branches share a common chain composed by the leg and the torso. The degrees of freedom of the full kinematic chain of the Pepper robot are distributed along its body as follows:

Body part	#DOFs
Head	2
Right arm	5
Right hand	1
Left arm	5
Left hand	1
Torso and leg	3
Platform	3

The total number of DOFs of Pepper adds up to 20.

For the current work, the hands are of no interest so a 18 DOFs kinematic model of the Pepper robot is used. The 18 DOFs correspond to the platform (3), leg and torso (3), head (2), right arm (5), and left arm (5). The origin of the reference frame of each arm chain is located on its wrist, and the origin of the reference frame of the head is located between the eyes (Fig. 4.1).

The particular location of the main frames of the robot can be seen in Fig. 4.1. Concretely,

[1] https://www.ald.softbankrobotics.com

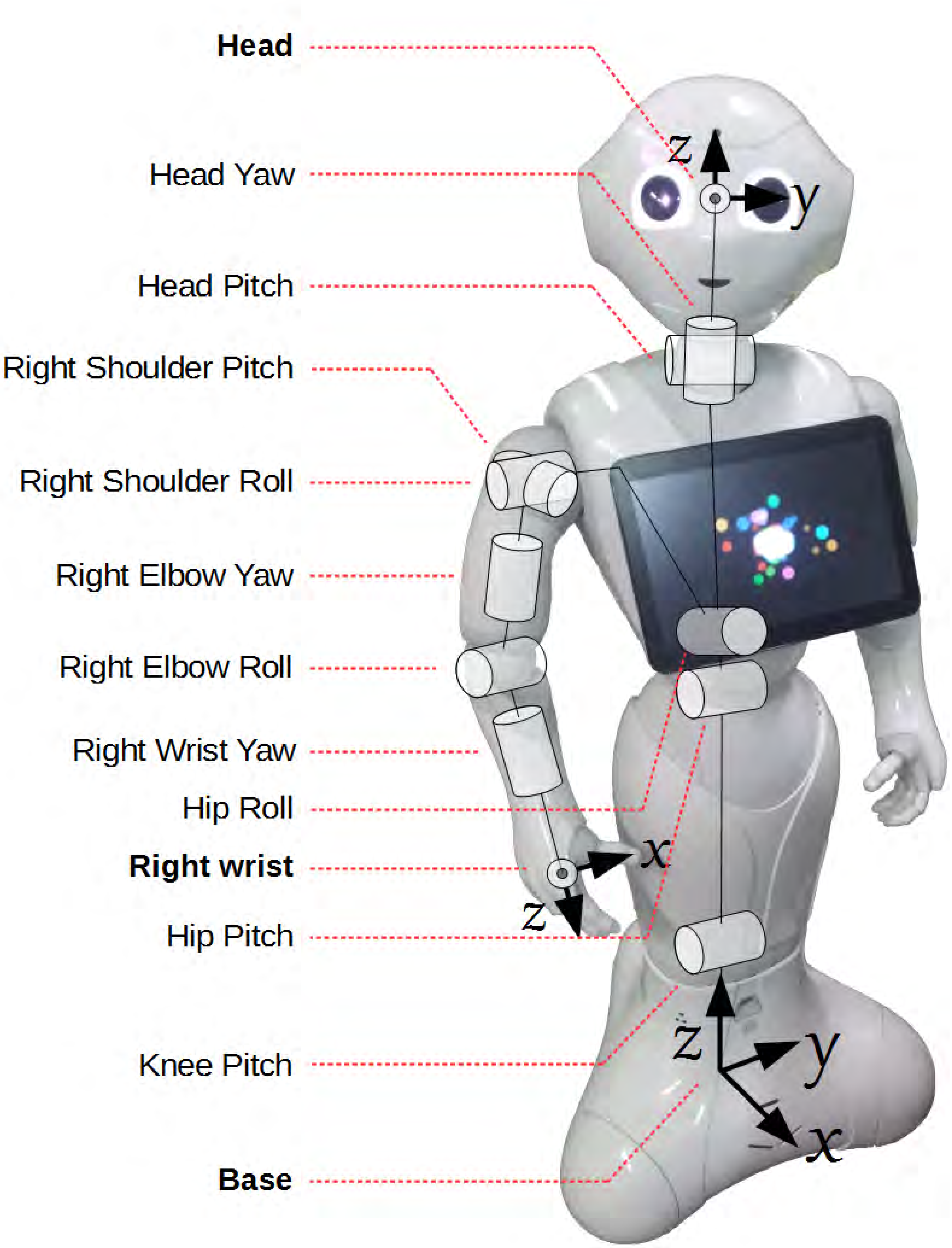


Figure 4.1 The Pepper robot.

Joint	a_i	α_i	d_i	θ_{i_0}	$\theta_{i_{min}}, \theta_{i_{max}}$
Base	0	$-\pi/2$	339	0	Fix
Knee Pitch	268	0	0	$-\pi/2$	-0.51, 0.51
Hip Pitch	79	$-\pi/2$	0	0	-1.04, 1.04
Hip Roll	0	$-\pi/2$	0	$-\pi/2$	-0.51, 0.51
Head Yaw	0	$-\pi/2$	309	$-\pi/2$	-2.09, 2.09
Head Pitch	0	0	0	0	-0.71, 0.61
Hip Roll	226	$\pi/2$	-57	0	-0.51, 0.51
R. Sh. Pitch	0	$\pi/2$	-150	$\pi/2$	-2.09, 2.09
R. Sh. Roll	0	$-\pi/2$	0	$-\pi/2$	-1.56, 0.01
R. Sh. Yaw	1	$\pi/2$	181	0	-2.09, 2.09
R. El. Roll	0	$-\pi/2$	0	0	0.01, 1.56
R. Wr.Yaw	0	0	150	0	-1.82, 1.82
Hip Roll	226	$\pi/2$	-57	0	-0.51, 0.51
L. Sh. Pitch	0	$\pi/2$	150	$\pi/2$	-2.09, 2.09
L. Sh. Roll	0	$-\pi/2$	0	$-\pi/2$	-1.56, 0.01
L. Sh. Yaw	1	$\pi/2$	181	0	-2.09, 2.09
L. El. Roll	0	$-\pi/2$	0	0	0.01, 1.56
L. Wr. Yaw	0	0	150	0	-1.82, 1.82

Table 4.1 The DH parameters of the Pepper robot. The angles are presented in radians and the lengths in millimetres. The first set of rows corresponds to the transformations of the common trunk of the kinematic tree. The second set extends the common trunk with the head transformations; the third, with the right arm; and the fourth, with the left, as in Fig. 4.1. R = Right, L = Left, Sh = Shoulder, El = Elbow, Wr = Wrist.

the end-effectors of the arms are located in its wrist, and the remaining important frames correspond to the base and the head. The rest of the joints are also shown, which correspond to the intermediate joints of the common trunk, and of the head and right arm chains. Note that the frames of the left arm and hand, as well as its joints have been omitted for ease of visualization, but they follow by symmetry.

The corresponding DH parameters of the Pepper robot have been computed from the data supplied by the manufacturer, and are shown in Table 4.1.

In order to formalize the mathematical model of the Pepper robot, let q be from now on a configuration of the robot given by

$$\boldsymbol{q} = \begin{bmatrix} x & y & \phi_z & \boldsymbol{\Theta}^T \end{bmatrix}^T \tag{4.1}$$

where:

- x, $y \in \mathbb{R}$ are the spatial coordinates of the position of the centre of the omnidirectional platform w.r.t. an inertial reference frame (see Fig. 4.1).

- $\phi_z \in SO(2)$ is the rotation angle of the platform around the Z axis of the inertial reference frame.
- $\mathbf{\Theta} = [\theta_i] \in \mathbb{R}^{15}$ is the vector of the joint values of the robot body, the *body configuration*, with i as the index of the joints in the following order: Knee Pitch, Hip Pitch, Hip Roll, Head Yaw, Head Pitch, Right Shoulder Pitch, Right Shoulder Roll, Right Elbow Yaw, Right Elbow Roll, Right Wrist Yaw, Left Shoulder Pitch, Left Shoulder Roll, Left Elbow Yaw, Left Elbow Roll and Left Wrist Yaw.

4.3 Emotion Conveyance

The main topics addressed in this part of the thesis are:

- Definition of a map from an emotional space to the kinematic features of the robot.
- Use of the Jacobian null space to execute motions on the robot in order to carry a desired emotional information as a secondary task.

In order to address these topics, a map is presented that takes emotions defined as points in a space of emotions (PAD space), and transform them to points in a space of kinematic features of the robot: extension, jerkiness, and gaze directness. Further, the map is shown to provide the robot configurations that can be fed to the null space of the robot, thus fulfilling both goals.

4.3.1 Proposed approach

Given a scenario with a user, a humanoid robot with its main task, and a desired emotion, the goal of this part of the dissertation is to execute the main task while the robot exploits its redundancy to convey the desired emotion to the user.

Concretely, in the proposed approach, an emotion is defined as a three dimensional point in the PAD emotion space. The three dimensions of this space are Pleasure, Arousal and Dominance. The coordinates in these dimensions corresponding to a particular desired emotion are transformed into the kinematic features *jerkiness*, *activity* and *gaze*, which are ultimately mapped to a continuous range of body configurations.

It is important to note that the proposed approach does not generate a single configuration of the robot for each emotion, but a whole range of continuous body configurations. Also, each emotion is not conveyed by a single body part or kinematic feature, but by the combination of the three kinematic features.

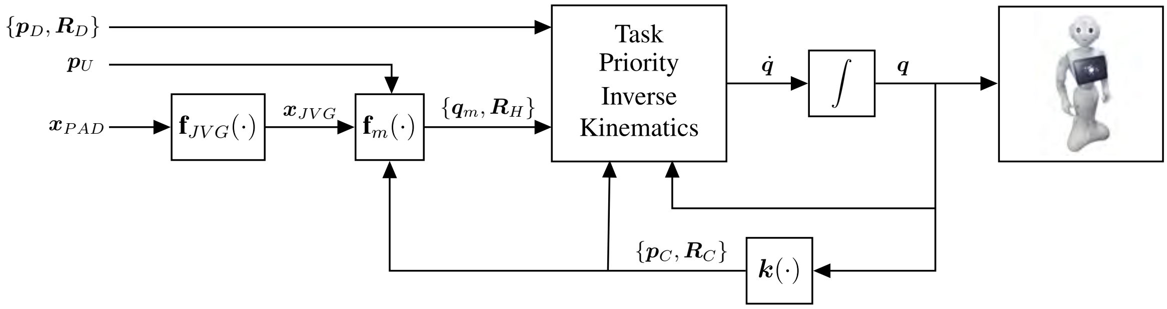


Figure 4.2 Block diagram of the emotion conveyance approach.

Finally, each one of the continuous range of body configurations becomes the input of the null space of the robot. Thus, the main task will be accomplished and the desired emotion will be conveyed by exploiting the remaining kinematic redundancy of the robot.

The diagram in Fig. 4.2 visually summarizes the proposed approach. Three variables are fed to the system:

- An emotion, represented by a point x_{PAD} in the Pleasure-Arousal-Dominance (PAD) space.
- The main task, composed by a temporal sequence of hand poses in the world frame, and represented by its position, p_D, and the orientation, R_D.
- The position of the eyes of the *user* in the world frame, p_U.

Following, the PAD point is transformed by the function $\mathbf{f}_{JVG}$ into a new space, the Jerkiness-Activity-Gaze (JVG) space, where its coordinates are mapped into the *jerkiness*, *activity* and *gaze* dimensions, thus transforming the initial emotional information to kinematic features.

Next, a new transformation, $\mathbf{f}_m$, is applied to transform the JVG point and the position of the user eyes, p_u, into a head orientation, R_H, and a robot configuration, q_m, which contain the information of the desired emotion.

Finally, the desired poses of the hand and the emotional information are used as input to the task priority inverse kinematics block. This inverse kinematics block outputs the velocity of the next configuration, which is further integrated, sent to the Pepper robot, and used to compute the current position and orientation, p_C and R_C, through the direct kinematics, $k(\cdot)$.

4.3.2 From PAD to motion features: $\mathbf{f}_{JVG}$

As explained in Section 3.2, the PAD model (Mehrabian, 1996) is used as the framework to code emotions. Several efforts have been made to develop a map between physical features of human gestures and the dimensions of the PAD model (Bernhardt and Robinson, 2007; Glowinski et al., 2011). Our work is mainly concerned with the arm and body motions (*pleasure* and *arousal*) and the robot gaze (*dominance*).

In the present approach, points in the *pleasure* and *arousal* dimensions of the PAD model are mapped to body motions similarly as in (Lim et al., 2011), but using the categorization of *jerkiness*, J, and *activity*, V, inspired by the work of Glowinski et al. (2011). Finally, by exploiting *gaze* directness, G, *dominance* can also be conveyed.

In (Glowinski et al., 2011), it is stated that, from a set of 25 motion features that convey emotions to users, 4 features retain the majority of the emotional information. These 4 features are *activity* (which captures how energetic the motion is), excursion (how is the energy distributed along the motion), extent (the openness of the arms and head) and jerkiness. Results in (Glowinski et al., 2011) show that *activity* positively correlates with arousal, as does extent to a lesser degree, hence the choice of *activity* as the carrier of the arousal dimension of the PAD space.

In the literature, it is stated that pleasure negatively correlates with *jerkiness* when arousal is high (Glowinski et al., 2011; Montepare et al., 1999). To capture this our proposed pleasure to jerkiness map is such that the less pleasure, the more *jerkiness* there is.

Gaze is a powerful mean for robots to communicate with humans in different tasks (Zheng et al., 2015). With respect to *dominance*, several works (Kleinke, 1986; Adams Jr. and Kleck, 2005; Carney et al., 2005; Lance and Marsella, 2008; Palanica and Itier, 2012; Tang and Schmeichel, 2015) point out the correlation between a direct (averted) gaze and high (low) dominance in humans. The map proposed in this work between *dominance-gaze* captures this information. Considering that the gaze of the robot can be naturally commanded through its joints, it is a straightforward mean available in a humanoid robot to convey dominance, hence its choice to convey the dominance dimension of the PAD space.

With the previous considerations, a linear map $\mathbf{f}_{JVG}$ can be defined between the points $\boldsymbol{x}_{PAD} = [P \quad A \quad D]^T \in [-1, 1]^3$ in the PAD space to the points $\boldsymbol{x}_{JVG} = [J \quad V \quad G]^T$ within the normalized domain $[0, 1]^3$ in the JVG space as

$$\mathbf{f}_{JVG} : \begin{bmatrix} P \\ A \\ D \end{bmatrix} \rightarrow \begin{bmatrix} J \\ V \\ G \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 - P \\ 1 + A \\ 1 + D \end{bmatrix} . \tag{4.2}$$

4.3.3 From JVG to the emotional configuration: $\mathbf{f}_m$

Activity

Activity, $V \in [0, 1]$, directly relates to the kinetic energy of the robot.

Let $\boldsymbol{\Theta}_m(t) = [\theta_{m_i}(t)] \in \mathbb{R}^{15}$ be the body configuration vector of the robot at time t. Given a minimum and a maximum expanded configurations, $\boldsymbol{\Theta}_0 = [\theta_{0_i}] \in \mathbb{R}^{15}$ and $\boldsymbol{\Theta}_{V_0} = [\theta_{V_{0i}}] \in \mathbb{R}^{15}$, respectively, a linear map between *activity* and the configuration space of the robot is introduced:

$$\theta_{m_i}(t) = (1 - V)\, \theta_{0_i} + V\, (\theta_{V_{0i}} + h_i \sin(\omega\, t + \varphi_i))\,. \tag{4.3}$$

The variables $\boldsymbol{h} = [h_i] \in \mathbb{R}^{15}$, with $h_i \in \mathbb{R}_{\geq 0}$, in Eq. 4.3, can be seen as an offset to $\boldsymbol{\Theta}_{V_0}$ such that when the robot is in its maximum expanded configuration it oscillates with an angular velocity ω between the configurations $\boldsymbol{\Theta}_{V_0} - \boldsymbol{h}$ and $\boldsymbol{\Theta}_{V_0} + \boldsymbol{h}$.

Differentiating $\theta_{m_i}(t)$, assuming V, h_i, ω and φ_i as constant, then

$$\dot{\theta}_{m_i}(t) = \omega\, V\, h_i \cos(\omega\, t + \varphi_i)$$

and the maximum kinetic energy K_{max_i} for joint i can be computed as

$$K_{max_i} = \max\left(\frac{1}{2}\, I_i\, \dot{\theta}_{m_i}(t)^2\right) = \frac{1}{2}\, I_i\, \omega^2\, V^2\, h_i^2 \tag{4.4}$$

where I_i is a given positive value that plays the role of the moment of inertia of joint i. From Eq. 4.4 it is straightforward to see that when *activity* is minimum (maximum), $V = 0$ ($V = 1$), the kinetic energy K_{max_i} is minimum (maximum).

Fig. 4.3 visually summarizes the *activity* category selected in this work for the Pepper robot. Four pictures can be seen in the top of the Figure: frontal and lateral views of the Pepper robot in its minimum *activity* (left pictures, $V = 0$), $\boldsymbol{\Theta}_0$; and maximum *activity* (right pictures, $V = 1$): $\boldsymbol{\Theta}_{V_0}$, the non-shaded configuration; $\boldsymbol{\Theta}_{V_0} - \boldsymbol{h}$, the shaded configuration with the lower arms; and $\boldsymbol{\Theta}_{V_0} + \boldsymbol{h}$, the shaded upper configuration with the horizontal arms.

Also, in the set of figures in the bottom row, the left figure shows the maximum directed gaze, while the other two show the extreme averted gazes; note that this is a basic example and in a real situation the head motion is added to the base emotional motion.

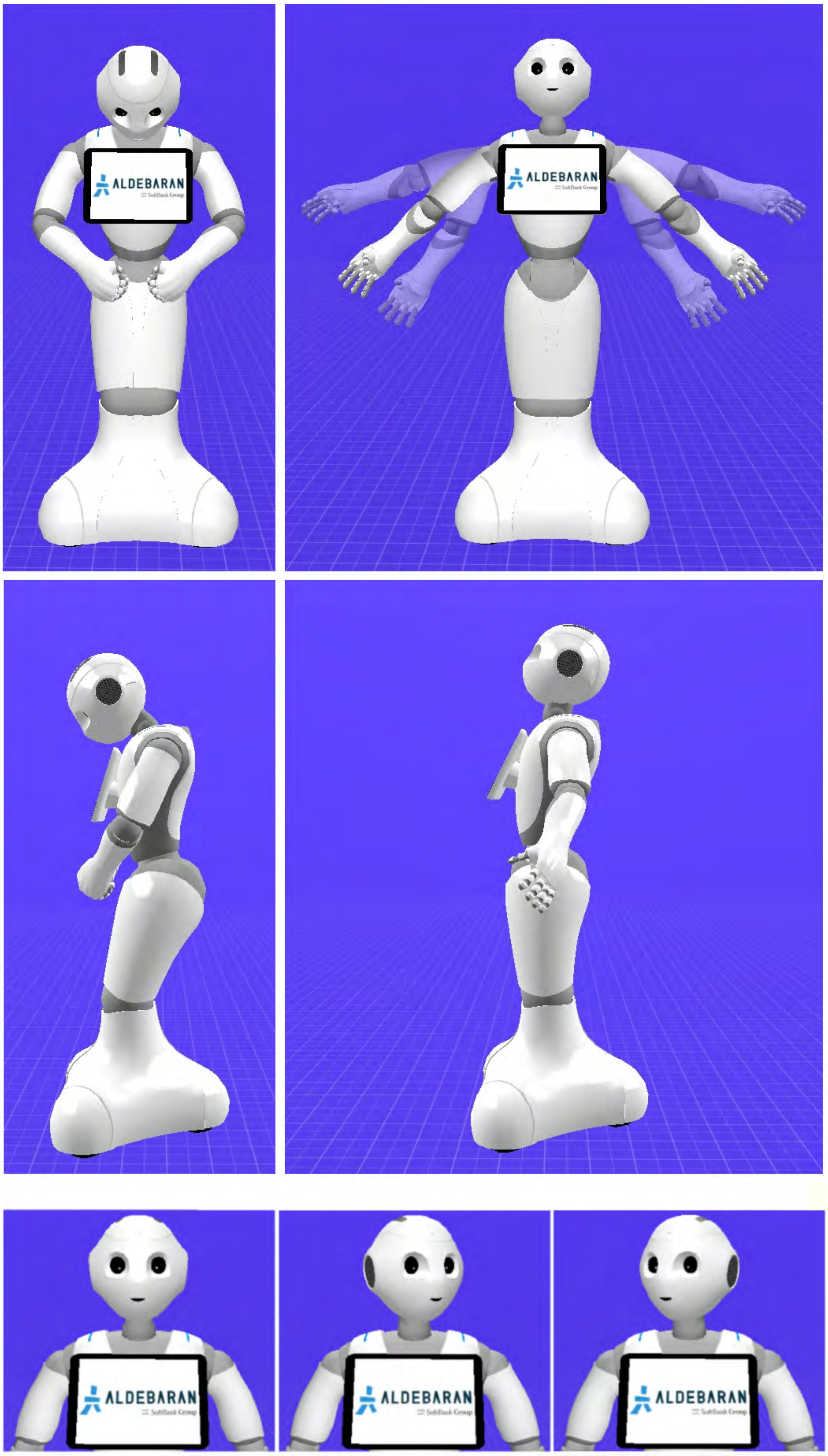


Figure 4.3 Kinematic features: *activity* (top) and *gaze* (bottom).

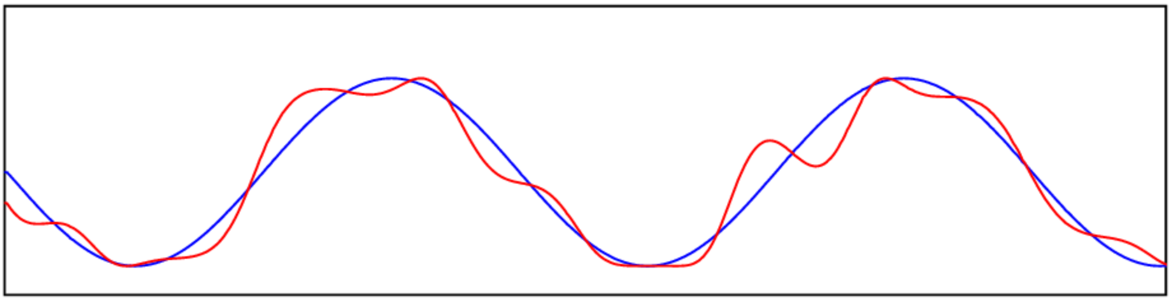


Figure 4.4 Joint base motion with (red) and without (blue) jerkiness.

Jerkiness

Jerkiness, $J \in [0, 1]$, over a base trajectory can be constructed by adding Fourier series terms in φ_i in Eq. 4.3:

$$\varphi_i(t) = J \sum_{j=1}^{n_J} \left(a_j \sin\left(\omega_j \, t\right) + b_j \cos\left(\omega_j \, t\right)\right). \tag{4.5}$$

In Fig. 4.4 a joint trajectory can be compared when the *jerkiness* is added using the above expression. In this figure, a joint base motion (blue) and the same motion with the added jerk signal (red) can be seen.

Gaze

As has been previously stated, *Gaze* directness, $G \in [0, 1]$, is an influential factor when conveying emotions, particularly in relation to the *dominance* dimension of an emotion.

In this work, the eyes are assumed to be non-actuated, so the command of the robot *gaze* is done solely through the head orientation.

In the proposed map in Eq. 4.2, the *dominance* is maximum ($D = 1$) when then *gaze* is also maximum ($G = 1$) and the orientation of the head is faced towards the user. When *dominance* is minimum ($D = 0$), the gaze is averted ($G = 0$) and the head follows the motion defined by the *activity* and *jerkiness* motion described before. Fig. 4.3 shows a direct gaze and two fully averted gazes.

The rotation matrix of the *gaze* orientation, $\boldsymbol{R}_H \in SO(3)$, is computed as the spherical linear interpolation (slerp) between two head rotation matrices: the direct *gaze*, $\boldsymbol{R}_G \in SO(3)$, and the emotional head orientation, $\boldsymbol{R}_m \in SO(3)$. $\boldsymbol{R}_G$ is the rotation matrix of the head orientation when *dominance* is maximum, that is, when the *gaze* is fully directed towards the user.

The head orientation $\boldsymbol{R}_m$ is computed from the direct kinematics with

$$\boldsymbol{q}_m = \begin{bmatrix} x & y & \phi_z & \boldsymbol{\Theta}_m^T \end{bmatrix}^T$$

obtained using Eq. 4.3. Thus, the desired head rotation, $\boldsymbol{R}_H$, is computed as

$$\boldsymbol{R}_H = \text{slerp}\left(\boldsymbol{R}_m, \boldsymbol{R}_G, G\right)$$

where the computation of $\boldsymbol{R}_G$ is straightforward given the position of the eyes of the user, $\boldsymbol{p}_U$, and the current position of the robot eyes, $\boldsymbol{p}_E$. $\boldsymbol{p}_E$ corresponds to the position vector of the head pose, and can be computed from the direct kinematics of the head chain. Then $\boldsymbol{R}_G$ is a rotation matrix computed with the X axis as the unit vector from the robot eyes to the user gazed by the robot, $\boldsymbol{p}_U - \boldsymbol{p}_E$; the Y axis, as the cross product of vector $[0 \quad 0 \quad 1]^T$ and the computed X axis, and the Z axis completes the orthonormal base.

The tracking of the orientation in $\boldsymbol{R}_H$ becomes the task for the second priority level of the inverse kinematic algorithm (Sect. 4.4).

Once $\boldsymbol{q}_m$ and $\boldsymbol{R}_H$ have been obtained, $\mathbf{f}_m$ can be defined as $\mathbf{f}_m(\boldsymbol{x}_{JVG}) = \{\boldsymbol{q}_m, \boldsymbol{R}_H\}$ and fed to the task priority module.

4.4 The Multi-Priority Inverse Kinematic Algorithm

Considering the task $\boldsymbol{x}$ of a robot and its direct kinematics $\boldsymbol{k}(\cdot)$, the robot configuration $\boldsymbol{q}$ that fulfills

$$\boldsymbol{x} = \boldsymbol{k}(\boldsymbol{q}) \tag{4.6}$$

can be obtained using to the first-order differential kinematics, $\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{x}}{\partial t}\dot{\boldsymbol{q}} = \boldsymbol{J}\dot{\boldsymbol{q}}$, by resorting to the inverse of the Jacobian, $\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}\dot{\boldsymbol{x}}$, and integrating.

When the robot is redundant $\boldsymbol{J}$ has more columns than rows and the pseudoinverse, $\boldsymbol{J}^+$, has to be used. In this situation the number of DOFs needed to execute the task is lower than the number of DOFs of the robot, and infinite solutions exist that satisfy Eq. 4.6. The general solution can be computed as

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^+\dot{\boldsymbol{x}} + \boldsymbol{N}\dot{\boldsymbol{q}}_0 = \boldsymbol{J}^+\dot{\boldsymbol{x}} + \left(\boldsymbol{I} - \boldsymbol{J}^+\boldsymbol{J}\right)\dot{\boldsymbol{q}}_0 \tag{4.7}$$

where $\boldsymbol{N}$ represents the orthogonal projection matrix in the null space of $\boldsymbol{J}$, and $\dot{\boldsymbol{q}}_0$ is an arbitrary joint-space velocity which allows to obtain different velocities of $\boldsymbol{q}$ that satisfy the desired task. Thus, a secondary task can be executed through $\dot{\boldsymbol{q}}_0$ with lower priority than task $\boldsymbol{x}$. This concept can be further extended to execute different tasks at distinct levels of priority (Siciliano and Slotine, 1991; Chiaverini, 1997; De Schutter et al., 2007; Mansard et al., 2009b).

In this work, the selected algorithm is that of Chiaverini (1997) and has been implemented as proposed by Baerlocher and Boulic (1998). The algorithm has been used to convey emotions through the priority levels two (*dominance*, through *gaze*) and three (*pleasure* and *arousal*, through *jerkiness* and *activity*, respectively) as

$$\begin{aligned}\dot{\boldsymbol{q}}_t &= \boldsymbol{J}_t^+ \boldsymbol{e}_t \\ \dot{\boldsymbol{q}}_h &= \dot{\boldsymbol{q}}_t + \boldsymbol{N}_t \boldsymbol{J}_h^+ \boldsymbol{e}_h \\ \dot{\boldsymbol{q}} &= \dot{\boldsymbol{q}}_h + \big(\boldsymbol{N}_t - (\boldsymbol{J}_h \boldsymbol{N}_t)^+ (\boldsymbol{J}_h \boldsymbol{N}_t)\big)\, \boldsymbol{e}_m\end{aligned} \tag{4.8}$$

where:

- $\boldsymbol{e}_t = \begin{bmatrix}\boldsymbol{K}_p\, \boldsymbol{\epsilon}_p^T & \boldsymbol{K}_o\, \boldsymbol{\epsilon}_o^T\end{bmatrix}^T$ is the error of the main task t with:
  - $\boldsymbol{\epsilon}_p = \boldsymbol{p}_D - \boldsymbol{p}_C$; $\boldsymbol{p}_i$, with $i \in \{D, C\}$, are the position of the robot final element of task t; and D and C, the desired and current frames, respectively.
  - $\boldsymbol{\epsilon}_o = \frac{1}{2}\left(\boldsymbol{n}_C \times \boldsymbol{n}_D + \boldsymbol{s}_C \times \boldsymbol{s}_D + \boldsymbol{a}_C \times \boldsymbol{a}_D\right)$, where $\boldsymbol{n}_i$, $\boldsymbol{s}_i$, and $\boldsymbol{a}_i$ are the columns of the rotation matrix that rotates the inertial frame into the current and desired frames C and D.
  - $\boldsymbol{K}_p$ and $\boldsymbol{K}_o$ are positive definite matrices.
- $\boldsymbol{e}_h$ is the head orientation error computed as $\boldsymbol{\epsilon}_o$.
- $\boldsymbol{e}_m$ is the emotional conveyance task error:

  $$\boldsymbol{e}_m = k_m \left(\boldsymbol{q}_m^* - \boldsymbol{q}^*\right)$$

  where $\boldsymbol{q}_i^* = \begin{bmatrix}0 & 0 & 0 & \boldsymbol{\Theta}_i^T\end{bmatrix}^T$, and $k_m \in \mathbb{R}_{>0}$.
- $\boldsymbol{J}_i$ is the Jacobian matrix of task $i \in \{t, h\}$.
- $\boldsymbol{J}^+$ is the pseudoinverse of $\boldsymbol{J}$.
- $\boldsymbol{N}_t = \boldsymbol{I} - \boldsymbol{J}_t^+ \boldsymbol{J}_t$ is the orthogonal projection operator onto the null space of task t.

A proof of the task prioritization of the proposed approach is shown in Appendix A, following the work of Chiaverini (1997) and Baerlocher and Boulic (1998).

4.5 Implementation

The algorithm has been implemented in Ubuntu 14.04, over a Intel Core i5 at 2500 GHz. The code has been written in C++ using the ROS middleware (Quigley et al., 2009). The main reason to use ROS has been to ease a future integration of the software in a higher-level framework. RViz (Rvi, 2011) has been used for visualization in the development stage, using a self-made kinematic model prior to the appearance of the official Pepper ROS node (Pep, 2015). Fig. 4.5 shows an image of Pepper as seen in RViz.

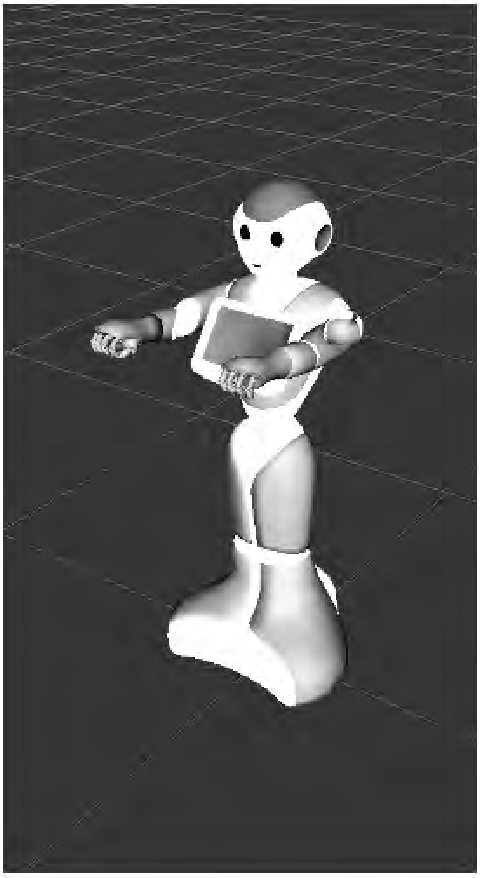


Figure 4.5 The Pepper robot in RViz.

Regarding the execution on the physical Pepper robot, the generated trajectories have been executed from a Python script loaded to Choreograph, the official software to command the Aldebaran robots.

The pseudoinverse of the Jacobian in Eq. 4.8 has been computed using the SVD decomposition. The singular values of Jacobian that were lower than a predefined threshold ($\delta < 0.001$) were set to zero in order to avoid unbounded velocities due to the appearance of singularities when inverting the Jacobian.

The frequency of the execution of the algorithms during the design phase has been 100 Hz. The samples for the trajectory execution have been fed to the Pepper robot at 4.45 Hz due to the dynamic constraints imposed by its lower level controllers. This gives rise to the necessary question of how much does the sampling frequency affect the intended jerkiness of the robot, and how to properly tune the jerkiness values to obtain the desired jerky effect.

With respect to the particular values of the parameters used in the implementation, the

Joint	Θ_{0_i}	$\Theta_{E_{0i}}$	h_i
Knee Pitch	0.50	0	0
Hip Pitch	0.50	-0.05	0
Hip Roll	0	0	0
Head Yaw	0	0	0
Head Pitch	0.64	-0.30	0
Right Shoulder Pitch	0.90	1.57	0
Right Shoulder Roll	-0.30	-0.85	0.45
Right Elbow Yaw	0	1.50	0.70
Right Elbow Roll	1.10	0.10	0.30
Right Wrist Yaw	1.00	1.15	-0.20
Left Shoulder Pitch	0.90	1.57	0
Left Shoulder Roll	0.30	0.85	0.45
Left Elbow Yaw	0	-1.50	-0.70
Left Elbow Roll	-1.10	-0.10	-0.30
Left Wrist Yaw	-1.00	-1.15	0.20

Table 4.2 Implemented values in radians.

angular velocity ω in Eq. 4.3 has been set to $\omega = 2.79$ rad/s. One term has been used in Eq. 4.5 with $n_J = 1$, $a_1 = b_1 = 0.25$ rad and $\omega_1 = 12.57$ rad/s.

Following the convention of Sect.4.2 the values Θ_{0_i}, $\Theta_{E_{0i}}$ and h_i implemented in Eq. 4.3 can be seen in Table 4.2.

4.6 Chapter Contributions

Two contributions are presented in this Chapter. The first contribution is a map that transforms emotions defined as points in a three dimensional space to kinematic features of the robot, which can be further fed to the null space of the robot.

The second contribution, which builds on the previous map, is a task priority null space approach that allows to convey emotions to users as a secondary task using the body motions of the robot while executing a primary task in the Cartesian space of the robot. To the author's knowledge, this secondary task has not been executed before in a robot.

Chapter 5

The User Study

Automatons cannot love; they can exchange their "personality packages" and hope for a fair bargain.

Erich Fromm

SUMMARY: The preliminaries and the setup on a Pepper robot of a user study to asses the emotion conveyance of the proposed approach are presented in this chapter, along with their results, the derived conclusions and some insights.

Once the mapping between the emotion space and the kinematic features of the robot has been defined, a study has been conducted to evaluate the conveyance performance of the proposed approach. The study is useful in order to analyse the dependencies between the *jerkiness*, *activity* and *gaze* directness, and the dimensions of the PAD model.

Fast responses of the participants have been required so that their responses captured the emotions they perceived on the Pepper robot, rather than what they actually *thought* they had perceived. The purpose of such an approach is to understand both the quantity and quality of the emotions that could be conveyed from the robot to the user at an unconscious level.

As explained in detail in Sect. 3.2.1, the SAM scale (Bradley and Lang, 1994) has been used to evaluate the degree of emotion conveyance of the proposed approach. The main reasons for choosing the SAM scale as a mean to measure the perceive emotions is that it eliminates the problems related to verbal measures, and, due to its visual nature, the participants can write their responses fast and intuitively, this last aspect being a key point in its choice.

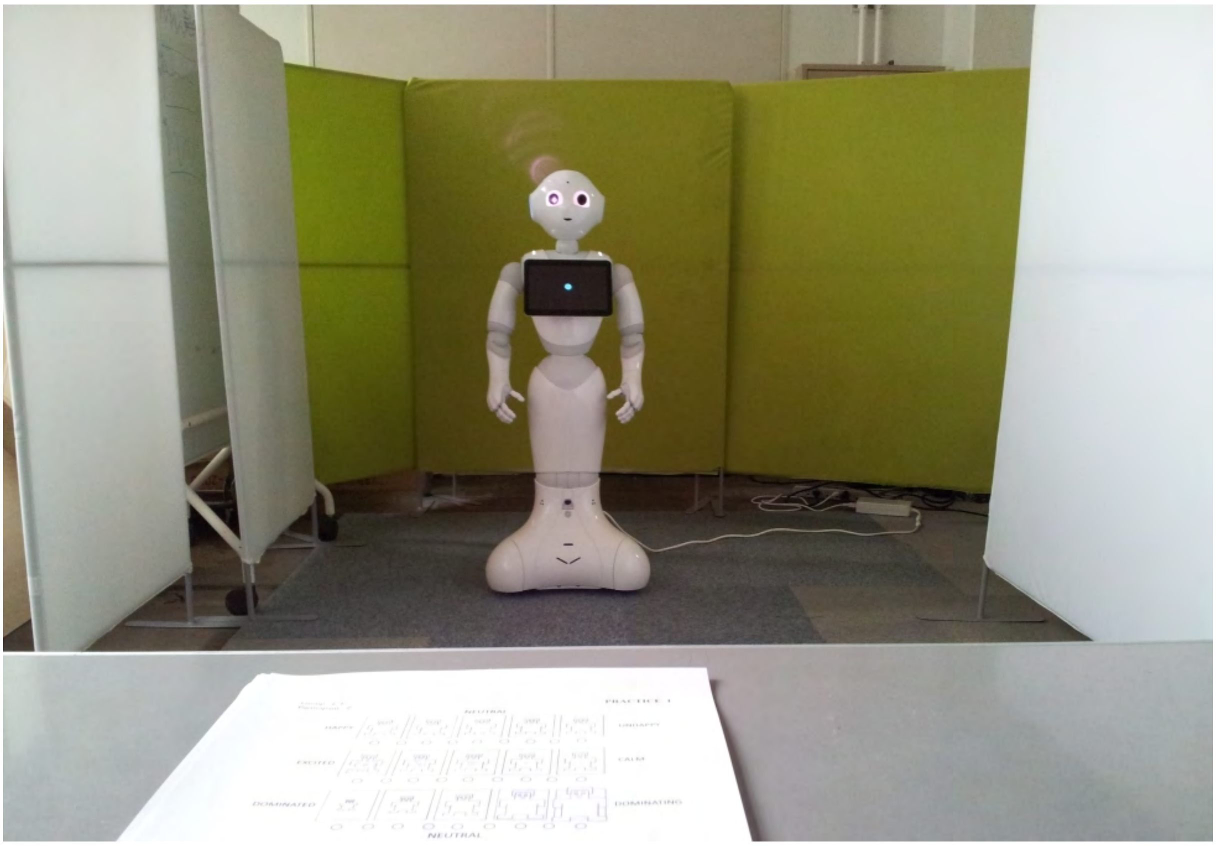


Figure 5.1 A participant's point of view during the experiment.

5.1 Description

Twelve trajectories on the Pepper robot were shown to each of the participants in the study. Right after, they evaluated the emotions that they perceived on each trajectory using the SAM scale. The duration of each motion was approximately 5 seconds. After the motion was played, each participant had 7 seconds to rate it on a paper sheet.

Fig. 5.1 shows the scenario where the experiment took place. The image depicts the view as was seen by the participants, who were 3 metres away from the Pepper robot, along with the table and the questionnaire where the answers were to be written.

There were 30 Japanese participants divided in groups of two or three persons. Among the participants, 26 were men and 4 were women, with most of them being students from the university campus. Their average age was 24.2 years, with a standard deviation of 3.7 years.

The motion chosen to be the main task in all the trajectories was a salutation movement using a hand. During this motion the platform was not moving. This motion, which in particular consisted of a waving motion executed with the right hand, constrained the six DOFs of the position and orientation of its corresponding wrist. At the same time this waving motion was being played, a movement representing an emotion on a particular point of the PAD space was played as a secondary task.

The twelve trajectories were shown in the following order:

1. A neutral motion that was not evaluated, and which purpose was to give an idea to the participants of the motion with the highest priority. In this motion, the Pepper robot only waved the right hand and the arm, without the addition of any secondary emotional motion.

2. Two random motions, that is, two motions with the waving motion as the highest priority task, but with the addition of two emotional motions each generated from random points in the PAD space. The aim of these motions was so that the participants could practice the SAM scale. As a consequence, the motions were evaluated by the participants, that is, written down on the questionnaire, but not taken as data in the analysis of the experiment.

3. Eight motions generated from a set of predefined points on the PAD space. Each point of this set of PAD points was obtained such that its P, A and D values had one of the two quasi-extreme values in each dimension: a low value L = -0.75, or a high value H = 0.75. In this way, for example, the point PAD = LHL would correspond to a point in the PAD space with P = -0.75, A = 0.75 and D = -0.75. All the possible combinations (i.e, low P, low A, low D; low P, low A, high D, *etc.*) add up to the eight motions. Six frames of different movements can be seen in Figs. 5.2. As *jerkiness* cannot be appreciated in a photography, only the two remaining dimensions, *activity* and *gaze*, can be perceived.

4. Two motions randomly chosen from a set of four well known emotions: *calm*, *sadness*, *happiness* and *fear*. The selected motions were generated from PAD values similar to (Gebhard, 2005). It is important to note that the criteria to select these particular emotions was not only that they were emotions which were expected to be familiar to the participants, but that they were characteristic of the PAD space. In this regard, Mehrabian (1996) assigns a representative label to each of the octants of the three dimensional PAD space: the octant corresponding to positive *pleasure*, *arousal* and *dominance* (+P+A+D) is labeled as Exuberant; -P-A-D, as Bored; +P-A+D, as Relaxed; -P+A-D, as Anxious, *etc*. Each of the four selected emotions in our user study belongs to a different octant: *happiness*, to the Exuberant octant; *sadness*, the Bored; *fear*, the Anxious; and *calm*, the Relaxed. Concretely, *happiness* and *sadness* correspond to opposite octants (+P+A+D and -P-A-D, respectively); as do *fear* and *calm* (-P+A-D and +P-A+D). It can now be more easily understood that the idea behind the choice of these four particular emotions was to show the participants a set of qualitatively distinguishable emotions. The evaluation of these two motions was not done using the SAM scale. Instead, the participants were asked to select which emotion between *calm*, *sadness*, *happiness* and *fear* better suited the motion they had just seen. As a fifth option, a «Do not know.» answer was also included.

Before the study took place, each participant had plenty of time to read the instructions of the experiment and to ask any question. The instructions were adapted from (Lang et al., 2008) and later translated into Japanese with the help of Japanese lab members. The average time per group of participants of each experiment was 11 minutes.

The instructions given to the participants, both in English and Japanese, as well as the questionnaire, can all be found in Appendix C.

In the video attached as a complementary material to this dissertation the full set of shown motions in the user study can be seen: the neutral waving motion, the eight predefined PAD motions, and the four motions corresponding to the four emotions. The video can be found in the link: User study video.

5.2 Results

Two main sets of results were obtained from the experiment.

The first set of results accounts for the quality of the emotion conveyance of the proposed approach. The dependencies between the features of the physical motions (*jerkiness*, *activity* and *gaze*) and the PAD perceived values are captured in the second set of results.

The underlying assumption behind the study was that the users would significantly perceive the PAD dimensions of the motions they were shown. And, as a consequence, they would perceive the corresponding emotions.

A distinction is needed from now on between the PAD values used as input to the robot in order to generate an emotional motion, and the PAD values corresponding to the responses of the participants using the SAM scales. The former will be named PAD and the latter, PAD* (or P*, A* and D*, when a need arises to express them separately).

Emotion conveyance results

Table 5.1 shows the results corresponding to the emotion selection part of the user study (the fourth step in the list in Sect. 5.1). The p-values are also presented, which were computed from a binomial distribution; as well as the 95% confidence intervals of the success rate. The success of the emotional conveyance can be evaluated by its distance above chance, which is 20%. 15 samples were obtained for each emotion.

In particular, the results show that *calm*, *happiness* and *sadness* are correctly perceived above chance. On the contrary, *fear* is sensed most of the times (80.0%) as *happiness*.

Interestingly, when wrongly perceived, *sadness* is perceived exclusively as *fear*. Finally, it is necessary to remark that, even though *calm* obtained a slightly good score, the participants were more uncertain about the conveyed emotion (40.0% of «Do not know.» responses).

Table 5.1 also presents the results of the evaluation of the SAM scale (which correspond to the third step in the Sect. 5.1 list): means (μ); p-values (p); confidence intervals (CI); and the Cohen's d effect sizes (ES). 30 samples were obtained per measure.

The results were evaluated through statistical significance towards the right bias. For instance, given a certain motion that tried to convey a high (low) *dominance* of D = H = 0.75 (D = L = -0.75), it was verified that the mean of the perceived *dominance* (μ_{D^*}) was positive (negative). The quality was analyzed using the p-value and the effect size. With respect to the effect, it can be interpreted as small if the absolute value of its effect size is less than 0.2; medium, if it is higher than 0.2 and less than 0.8; and big, when higher than 0.8. The 95% Confidence Intervals are also added to ease its interpretation.

The most straightforward conclusions that can be obtained from Table 5.1 are summarized below:

- The three dimensions are well conveyed in two PAD points: PAD = LLL and PAD = HHH. Note from its p-values that the former is far better conveyed.
- Four PAD points are well conveyed in two dimensions: LHH, HLL, HLH, HHL. The results obtained are mixed, though. For instance, *arousal* and *dominance* are better conveyed in HLL, than are *pleasure* and *dominance* in HLH.
- Two PAD points, LLH and LHL, are well conveyed in one dimension.

It can also be seen that, from the total of 8 motions, *arousal* is well conveyed in 7 motions, *dominance* in 5 motions, and *pleasure* in 4.

Overall, the interpretation of the results is not obvious, and a deeper analysis of the data is necessary in order to understand the interactions between the intended conveyed emotions and the emotions perceived by the users.

Perceived values and correlations

The second set of results are shown in Tables 5.2 and 5.3. This analysis is presented in order to shed some light on the hidden interactions between variables of the JVG space and their perceived PAD* values, which were not obvious in the previous results. The comparison is done between the JVG motion features and the perceived emotions, PAD*, as this approach allows a more intuitive visualization of the robot motions and eases the interpretation of its effect on the perception of the participant.

Table 5.1 Emotion conveyance results

	Responses (%)					p				
Emotion	C*	F*	H*	S*	⊘	C*	F*	H*	S*	⊘
Calm	**33.3**	6.7	20.0	0	40.0	**0.06**	0.83	0.35	0.96	0.02
Fear	6.7	**13.3**	80.0	0	0	0.83	**0.60**	0	0.96	0.96
Happiness	6.7	0	**73.3**	6.7	13.3	0.83	0.96	**0**	0.83	0.60
Sadness	0	33.3	0	**66.7**	0	0.96	0.06	0.96	**0**	0.96

	Confidence Interval (95%) of the Responses				
Emotion	C*	F*	H*	S*	⊘
Calm	[9.48 57.19]	[0 19.29]	[0 40.24]	[0 0]	[15.21 64.79]
Fear	[0 19.29]	[0 30.54]	[59.76 100.00]	[0 0]	[0 0]
Happiness	[0 19.29]	[0 0]	[50.95 95.71]	[0 19.29]	[0 30.54]
Sadness	[0 0]	[9.48 57.19]	[0 0]	[42.81 90.52]	[0 0]

P A D	μ_{P^*}	μ_{A^*}	μ_{D^*}	p_{P^*}	p_{A^*}	p_{D^*}	CI$_{P^*}$	CI$_{A^*}$	CI$_{D^*}$	ES$_{P^*}$	ES$_{A^*}$	ES$_{D^*}$
L L L	**-0.59**	**-0.30**	**-0.52**	0	0	0	[-0.74 -0.44]	[-0.49 -0.13]	[-0.65 -0.38]	-1.4	-0.6	-1.4
L L H	0.13	**-0.14**	-0.11	0.03	0.06	0.12	[0.01 0.25]	[-0.29 0.01]	[-0.25 0.03]	0.4	-0.3	-0.3
L H L	0.27	**0.37**	0.18	0	0	0	[0.10 0.43]	[0.22 0.52]	[0.01 0.36]	0.6	0.9	0.4
L H H	0.24	**0.20**	**0.10**	0	0	0.14	[0.11 0.38]	[0.07 0.33]	[-0.03 0.23]	0.7	0.5	0.3
H L L	-0.53	**-0.30**	**-0.49**	0	0	0	[-0.72 -0.33]	[-0.49 -0.11]	[-0.67 -0.31]	-1.0	-0.6	-1.0
H L H	**0.13**	-0.03	**-0.12**	0.04	0.71	0.06	[0.01 0.26]	[-0.21 0.14]	[-0.24 0]	0.4	-0.1	-0.4
H H L	**0.18**	**0.31**	0.02	0.07	0	0.86	[-0.01 0.38]	[0.15 0.47]	[-0.17 0.20]	0.3	0.7	0
H H H	**0.19**	**0.11**	**0.10**	0.01	0.15	0.14	[0.05 0.34]	[-0.04 0.27]	[-0.03 0.23]	0.5	0.3	0.3

Top two tables: Results of the emotion selection. 15 samples were obtained per emotion. Each row corresponds to the results regarding the conveyed emotion of the first column, *i.e.*, users perceived calm as happiness in a 20.0% of the times (first row and second column in the Responses section); the success rate of each emotion and the corresponding p-value are highlighted in bold. C*, F*, H* and S* correspond to the perceived/selected *calm*, *fear*, *happiness* and *sadness*; ⊘ is the «Do not know.» answer. Bottom table: Means of the SAM scale evaluation for each motion. Each row corresponds to the desired conveyed emotion in the PAD scale. 30 samples per emotion were gathered. Emotions that were perceived in the right direction have their means presented in bold.

a)	μ_i	p_i	CI$_{\mu_i}$	ES$_i$
P*	0	0.91	[-0.06, 0.07]	0.01
A*	0.03	0.43	[-0.04, 0.09]	0.01
D*	**-0.10**	0	[-0.17, -0.04]	-0.21

b)	r_{Ji}	r_{Vi}	r_{Gi}	$p_{r_{Ji}}$	$p_{r_{Vi}}$	$p_{r_{Gi}}$
P*	-0.02	**0.39**	**0.29**	0.61	0	0
A*	-0.01	**0.43**	0	0.54	0	0.47
D*	0.03	**0.41**	**0.22**	0.69	0	0

	CI$_{r_{Ji}}$	CI$_{r_{Vi}}$	CI$_{r_{Gi}}$
P*	[-0.11,0.15]	[0.28,0.49]	[0.17,0.40]
A*	[-0.12,0.14]	[0.32,0.53]	[-0.13,0.13]
D*	[-0.10,0.16]	[0.30,0.51]	[0.10,0.34]

Table 5.2 PAD Means and JVG to PAD* correlations: a) means (μ); p-values (p); confidence intervals (CI); and effect sizes (ES), of the PAD* responses. 240 samples per emotion have been gathered. The index i is the PAD* dimension of the corresponding row, i.e, $\mu_{A^*} = 0.03$; this notation also applies to Tables 5.3. b) general correlations. 240 samples per measure have been gathered.

As explained in Sect. 4.3.2, note from Eq. 4.2 that a high *jerkiness*, J= H, corresponds to a low *pleasure*, A= L; a high (low) *activity*, V= H (V= L), corresponds to a high (low) *arousal*, A= H (A= L); and a direct (averted) *gaze*, G= H (G= L), corresponds to a high (low) *dominance*, D= H (D= L).

Table 5.2a shows the means (μ), standard deviations (σ), and effect sizes (ES), of the PAD* values of the SAM scales rated by the participants. 240 samples per measure were gathered. The Spearman's rank correlation coefficients between the JVG variables and the user PAD responses are presented in Tables 5.2b-5.3. Table 5.2b shows the overall correlation coefficients. Tables 5.3 present the correlations for certain conditions (*i.e.* averted *gaze*, $G = L$), with bold values corresponding to statistically significant values ($p < 0.05$). For the correlations in Tables 5.3, 120 samples per measure have been gathered.

Tables 5.2 and 5.3 can be summarized as follows:

- There is a statistically significant bias ($p_{D^*} = 0.001$) towards non-*dominance* ($\mu_{D^*} = -0.10$).
- *Activity* correlates positively with *arousal* ($r_{VA^*} = 0.43$).
- *Jerkiness* does not correlate with any value of the SAM scale.
- *Gaze* directness positively correlates with *dominance* ($r_{GD^*} = 0.22$), highly when *activity* is low ($r_{GD^*} = 0.48$), and almost nothing when *activity* is high ($r_{GD^*} = 0.01$).

a)

Low Ext. (V = L)	r_{Ji}	r_{Gi}	$p_{r_{Ji}}$	$p_{r_{Gi}}$	CI$_{r_{Ji}}$	CI$_{r_{Gi}}$
P*	-0.03	**0.66**	0.32	0	[-0.21,0.15]	[0.55,0.75]
A*	-0.05	**0.23**	0.22	0	[-0.23,0.13]	[0.05,0.39]
D*	0	**0.48**	0.48	0	[-0.18,0.18]	[0.33,0.61]

High Ext. (V = H)	r_{Ji}	r_{Gi}	$p_{r_{Ji}}$	$p_{r_{Gi}}$	CI$_{r_{Ji}}$	CI$_{r_{Gi}}$
P*	0.07	-0.04	0.86	0.72	[-0.11,0.25]	[-0.22,0.14]
A*	0.07	**-0.23**	0.87	0	[-0.11,0.25]	[-0.39,-0.05]
D*	0.09	0.01	0.92	0.43	[-0.09,0.26]	[-0.17,0.19]

b)

Low Jrk. (J = L)	r_{Vi}	r_{Gi}	$p_{r_{Vi}}$	$p_{r_{Gi}}$	CI$_{r_{Vi}}$	CI$_{r_{Gi}}$
P*	**0.33**	**0.28**	0	0	[0.16,0.48]	[0.11,0.44]
A*	**0.36**	0.03	0	0.33	[0.19,0.51]	[-0.15,0.21]
D*	**0.36**	**0.24**	0	0	[0.19,0.51]	[0.06,0.40]

High Jrk. (J = H)	r_{Vi}	r_{Gi}	$p_{r_{Vi}}$	$p_{r_{Gi}}$	CI$_{r_{Vi}}$	CI$_{r_{Gi}}$
P*	**0.45**	**0.30**	0	0	[0.29,0.58]	[0.13,0.45]
A*	**0.51**	-0.02	0	0.62	[0.36,0.63]	[-0.20,0.16]
D*	**0.47**	**0.20**	0	0	[0.32,0.60]	[0.02,0.37]

c)

Av. Gaze (G = L)	r_{Ji}	r_{Vi}	$p_{r_{Ji}}$	$p_{r_{Vi}}$	CI$_{r_{Ji}}$	CI$_{r_{Vi}}$
P*	0.02	**0.64**	0.59	0	[-0.16,0.20]	[0.52,0.73]
A*	0.03	**0.57**	0.68	0	[-0.15,0.21]	[0.44,0.68]
D*	0.07	**0.56**	0.86	0	[-0.11,0.25]	[0.42,0.67]

Dir. Gaze (G = H)	r_{Ji}	r_{Vi}	$p_{r_{Ji}}$	$p_{r_{Vi}}$	CI$_{r_{Ji}}$	CI$_{r_{Vi}}$
P*	0.05	**0.12**	0.78	0.03	[-0.13,0.23]	[-0.06,0.29]
A*	-0.01	**0.28**	0.44	0	[-0.19,0.17]	[0.11,0.44]
D*	0.02	**0.28**	0.61	0	[-0.16,0.20]	[0.11,0.44]

Table 5.3 PAD Means and JVG to PAD* correlations: a) correlations for Low and High Activity. 120 samples per measure have been gathered. Ext = Extension. b) correlations for Low and High Jerkiness. 120 samples per measure have been gathered. Jrk = Jerkiness. c) correlations for averted and direct *gaze*. 120 samples per measure have been gathered. Av = Averted; Dir = Direct.

- When *activity* is high *gaze* negatively correlates with *arousal* ($r_{GA^*} = -0.23$). Also, when *gaze* is directed the correlation between *activity* and *pleasure*, *arousal* and *dominance* ($r_{VP^*} = 0.64$, $r_{VA^*} = 0.57$, $r_{VD^*} = 0.56$) is more than the double than when *gaze* is averted ($r_{VP^*} = 0.12$, $r_{VA^*} = 0.28$, $r_{VD^*} = 0.28$).

5.3 Discussion

The presented results point out that *happiness* and *sadness* are correctly recognized; while *calm* is slightly well perceived above chance, and *fear* is most of the times interpreted as *happiness*.

Some conclusions can be extracted from the data presented in Tables 5.2 and 5.3:

- The Pepper robot was perceived as low dominant. This can be deduced from the statistically significant bias towards non-*dominance*. This perception seems reasonable as in fact Pepper does not seem by design very dominant. As noted in the literature, the design can have a significant impact on the perception that the users have of the robot (Disalvo et al., 2002). This also implies that in order to transmit a dominant emotion (with body motion, voice, or any other mean) it would be necessary to overcome this bias (but it would also be easier to transmit low dominant emotions).
- *Activity* effectively carries the *arousal* information.
- *Jerkiness* does not convey any emotion. Three reasons can be given to explain this:
  - One reason for this might be that when there is *jerkiness* there always is the simultaneous occurrence of another more energetic motion, thus *jerkiness* may not be noticed.
  - Another, as explained in Sect. 4.5, might be that the interaction between the parameters of the jerky motion and the sample rate of the robot are not properly synchronized. This would indicate that there is a need for a formal definition of *jerkiness*, how humans perceive it, and, specially, how to artificially generate, implement and execute it on a robot.
  - The miss-conveyance of the jerky motions can also explain why *fear* is perceived as *happiness*. First, note that both *fear* and *happiness* have high arousal values, but negative and positive values in the pleasure dimension, respectively. Under the assumption that jerkiness was not perceived at all, then *fear* would have been perceived as an emotion with a higher pleasure; and that emotion is *happiness*.
- There is a positive correlation between *gaze* directness and *dominance*. More interesting, the fact that *gaze* correlates highly with a low *activity*, and barely with a high *activity*, may indicate that when the expanded movement is very energetic, and thus *activity* is high, it drags the user attention, which in turn influences all the other correlations. When *activity* is not so noticeable, other features can arise and better convey their emotions.

- The negative correlation between *gaze* and *arousal* when *activity* is high can be explained by noticing that Pepper staring towards the user diminishes the effect of the expanded arms. This may be due to the fact that the more the *gaze* is directed, then the head tends to move less, and so the overall kinematic energy of the robot decreases. Also, when *gaze* is directed, the correlation between *activity* and *pleasure*, *arousal* and *dominance* is more than the double than when the *gaze* is averted. It can be thus stated that a direct *gaze* diminishes the impact that *activity* has on the user.
- A gross evaluation in the SAM scale using the most relevant motion of the Pepper robot can be perceived. That is, when *activity* is the most noticeable (energetic) feature of a motion, all values of the SAM scale positively correlate more to *activity*: *activity* seems to drag not only *arousal*, but also *dominance* and *pleasure*. If *activity* is low, then *gaze* is the dragger. This might indicate that the user tends to create a general binary notion of the emotional state of Pepper (happy/sad, good/bad, *etc.*) and the PAD responses on the SAM scales capture this information.

The results of the user study demonstrate that emotions can be conveyed to the user as secondary tasks using the null space of a humanoid robot. It is also necessary to remark that the mapping used to transform the emotional information to the kinematic features needs to be improved in order to convey the desired emotions with more accuracy. In this regard, this work does not claim that the mapping is unique. It does not either claim that the mapping introduced in this work is the best among all possible mappings. Efforts in this direction need to be made.

Another limitation of the proposed approach is a direct consequence of the complexity of the task and the number of DOFs of the robot. For a relatively simple humanoid robot and a demanding task the dimensionality of the null space can be too low to convey the desired emotion.

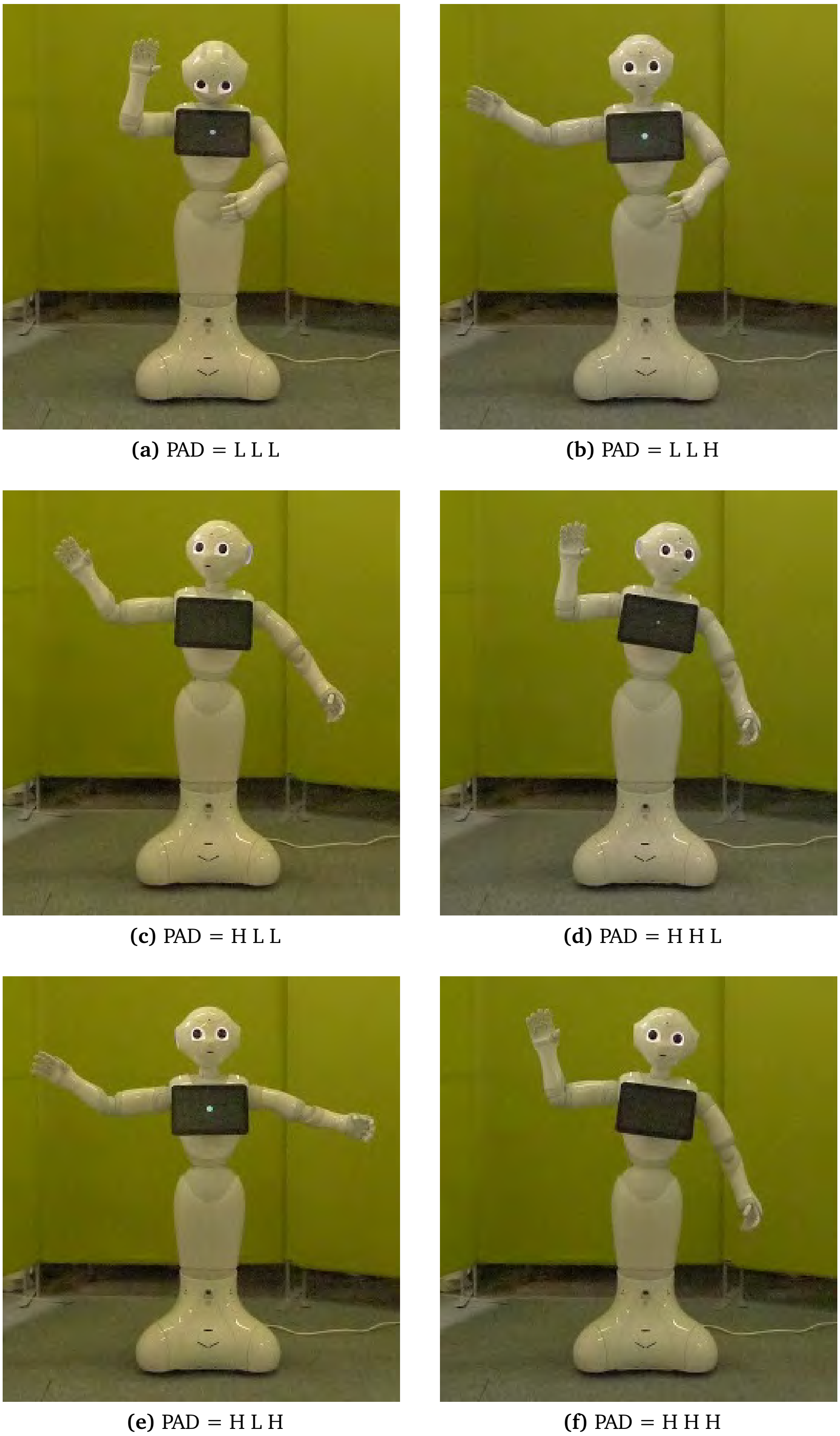
Interesting questions arising from this work:

- To what extent does the main task influence the user perception of the robot emotional state?
- Since all the participants were students from Japan, to what extent there is a cultural bias in the results?
- How much is the emotion conveyance influenced by the number of DOFs available on the robot?
- How can artificial *jerkiness* be created and properly executed by the robot? How will it be perceived by humans?
- Do more energetic motions better convey emotions? How many emotion dimensions (*pleasure*, *arousal*, *etc.*) can the robot simultaneously and successfully convey?
- To what extent does the user empathize with only the emotions conveyed by the robot body motions, rather than with motions and voice, for example?

5.4 Chapter Contributions

Summarizing, the main contributions of this Chapter are the results from a user study to assess the goodness of the emotional conveyance of the proposed approach, along with an analysis of the particular interactions between the motion features of the robot and the emotions conveyed to the participants of the user study.

Overall, this Part of the dissertation presents the first evidence that the null space of a robot can be used to convey emotions to users as a secondary task.



(a) PAD = L L L

(b) PAD = L L H

(c) PAD = H L L

(d) PAD = H H L

(e) PAD = H L H

(f) PAD = H H H

Figure 5.2 The salutation motion with different emotions.

Part II

Teleoperating a Mobile Manipulator with a UAV as visual feedback

Chapter 6

Workspace Mapping

A robot may not injure humanity, or, through inaction, allow humanity to come to harm.

Isaac Asimov, The Zeroth Law of Robotics

SUMMARY: A novel teleoperation system that allows the simultaneous and continuous command of a ground mobile manipulator and a free flying camera is presented in this chapter, along with specific tools to ease its teleoperation, and the results of a user study to assess its performance.

6.1 Introduction

Teleoperation has been a key way of steering the robots since the robotics inception (Goertz, 1952). The interest of teleoperating a robot appears in several scenarios: when the task to be executed is too dangerous for a human, like in a nuclear plant or another planet, but still too demanding to be fully automated, thus requiring a "human intelligence" to guide the robot at some level; or when the task performance by a human would be sub-optimal, and a "help" from a robot would enhance the overall performance, whether by the exertion of a higher force, or through a more accurate motion, for instance.

In some teleoperation setups, the visualization of the scene allows the user to command the robot in real-time. For example, through fixed cameras in a lab. With the advent of small and affordable unmanned air vehicles (UAV), the possibility to make use of its cameras allows them to be considered as free-flying cameras. Interestingly, this approach leads to the question of whether these free-flying cameras can be used for teleoperation purposes, and what problems arise in order to coordinate them with the teleoperated robots.

Following this line of though, this chapter presents a novel teleoperation system that allows the simultaneous and continuous command of a ground mobile manipulator (MM) and a free flying camera, implemented using a UAV, from which the operator can monitor the task execution in real-time.

The mobile manipulator will be indistinctly referenced as the (ground) mobile manipulator or the robot from now on, differentiating it from its parts: the (mobile) platform and the (arm) manipulator. The UAV will be referenced as the UAV, the drone, or the free-flying camera.

The core idea is to free as much as possible the operator from the command of the free-flying camera, so that he/she can focus on the object carried by the robot end-effector, since the transport and manipulation of this object is regarded as the ultimate goal of the teleoperation. This way, the attention of the user should be mainly set on the object, and the mobile manipulator and the camera should respond according to the user desired motions on it, posing as little demand as possible on the operator, almost as if they were not there.

In this regard, the proposed decoupled position and orientation workspace mapping allows the teleoperation of a complex robot with unbounded workspace from a single haptic device with bounded workspace.

When the operator is reaching the position and orientation boundaries of a predefined subspace inside the haptic workspace, linear and angular velocity components are respectively added to the inputs of the mobile manipulator and the free-flying camera.

A user study in a virtual environment has been conducted to evaluate the performance and the workload on the operator both before and after proper training. Analysis on the data shows that the system complexity is not an obstacle for an efficient performance.

6.2 The Teleoperation System

The robot used in this work is the Barcelona Mobile Manipulator I (BMM-I). The BMM-I consists of an omnidirectional mobile platform with an arm manipulator (see Figs. 10.1 and 6.1). From a kinematic point of view, the platform has three DOFs: two independent translations and a rotation around the vertical axis. The arm manipulator is a KUKA LWR 4+, which has seven DOFs. This makes a total of ten DOFs for the BMM-I.

The goal of this part of the dissertation is to ultimately teleoperate the BMM-I while the camera of a Parrot AR.Drone is sending the video of the scene to the operator.

Formally, the configuration of the mobile manipulator can be defined as

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{x}_p \\ \boldsymbol{q}_m \end{bmatrix} \tag{6.1}$$

with

- $\boldsymbol{x}_p = \begin{bmatrix} p_{p_x} & p_{p_y} & \phi_{p_z} \end{bmatrix}^T$ being the vector with the X and Y positions, and the orientation of the platform w.r.t. the vertical axis, $p_{p_x}, p_{p_y} \in \mathbb{R}$ and $\phi_{p_z} \in SO(2)$, respectively; and
- $\boldsymbol{q}_m \in \mathbb{R}^7$ as the vector of joint configurations of the manipulator.

6.2.1 The frames

Given an homogeneous transformation matrix

$$\boldsymbol{T}_i^j = \begin{bmatrix} \boldsymbol{R}_i^j & \boldsymbol{p}_i^j \\ \boldsymbol{0}^T & 1 \end{bmatrix}$$

composed of a rotation matrix, $\boldsymbol{R}_i^j \in SO(3)$ and a translation vector, $\boldsymbol{p}_i^j \in \mathbb{R}^3$, which expresses the frame i w.r.t. the frame j. $\boldsymbol{T}_i^j$ can also be used to represent a position vector expressed in a frame i into a new frame j.

Several frames need to be defined in this work prior to the introduction of the workspace mapping algorithm (Figs. 6.1):

- mo: the frame attached to the origin of a predefined subspace in the haptic workspace, named the *haptic subspace* from now on, which contains the regions of interest inside the haptic workspace that allow the user to trigger the different behaviors of the proposed algorithm, and which can be seen as a yellow cylinder in the top figure in Figs. 6.1.
- mi: the frame at the tip of the haptic device.
- s: the world frame at the remote scenario, a.k.a., the mobile manipulator workspace frame.
- so: the frame attached to the counterpart of the origin of the haptic subspace in the remote scenario. Concretely, the haptic subspace corresponding to the frame mo is mapped into the remote scenario, thus incorporating the operator input inside the robot workspace. This mapped subspace will be called from now on the *haptic subspace counterpart*, and can be seen as a yellow cylinder in the bottom figure in Figs. 6.1.
- si: the frame of the counterpart of the haptic tip in the remote scenario.
- c: the frame attached to the free-flying camera desired pose.

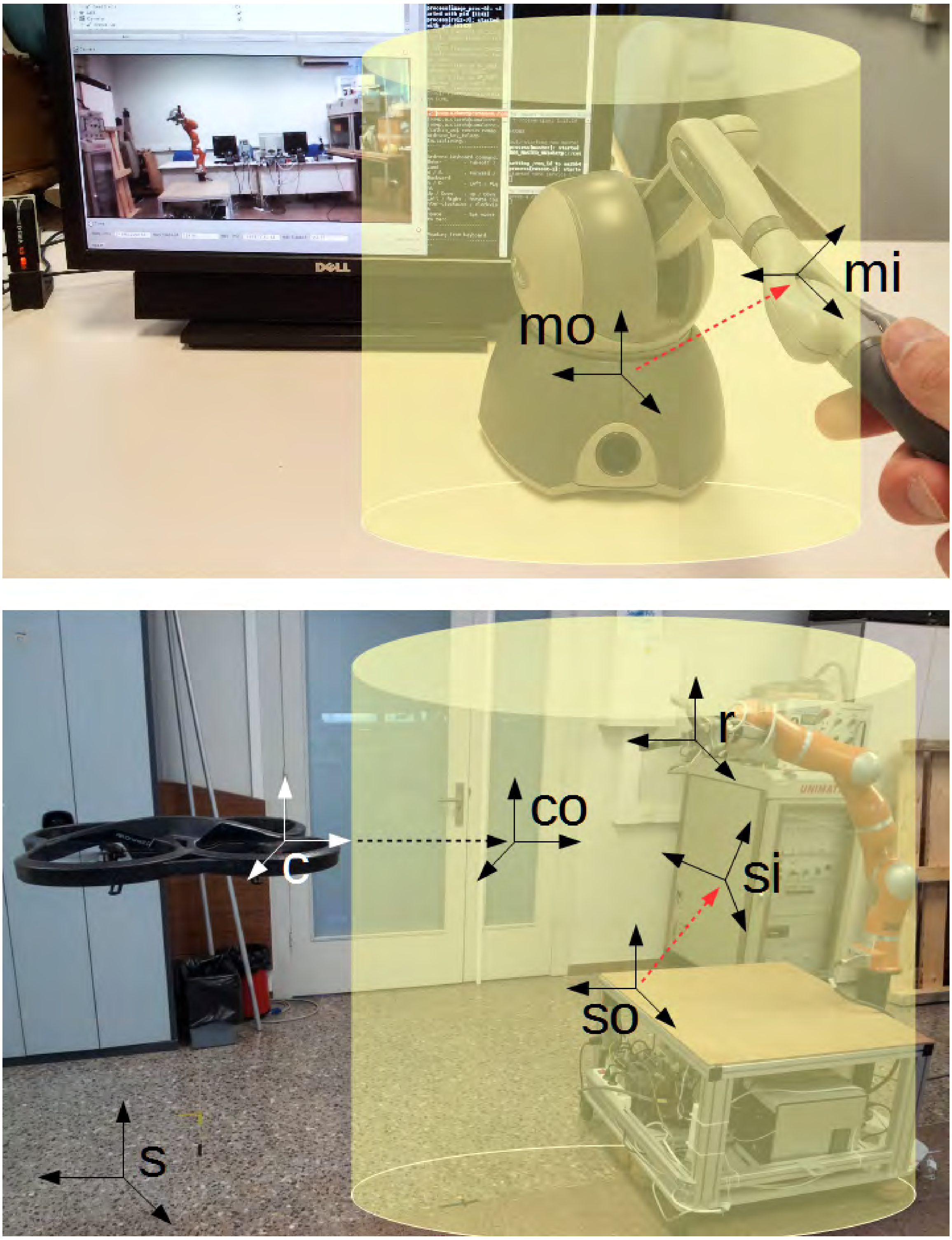


Figure 6.1 The teleoperation system. Top: the haptic, with the *haptic subspace* as a yellow cylinder, and the screen with visual feedback from the UAV. Bottom: the subspace within the haptic workspace (in yellow) is mapped into the workspace of the robot as the *haptic subspace counterpart*. Red arrows depict the user command.

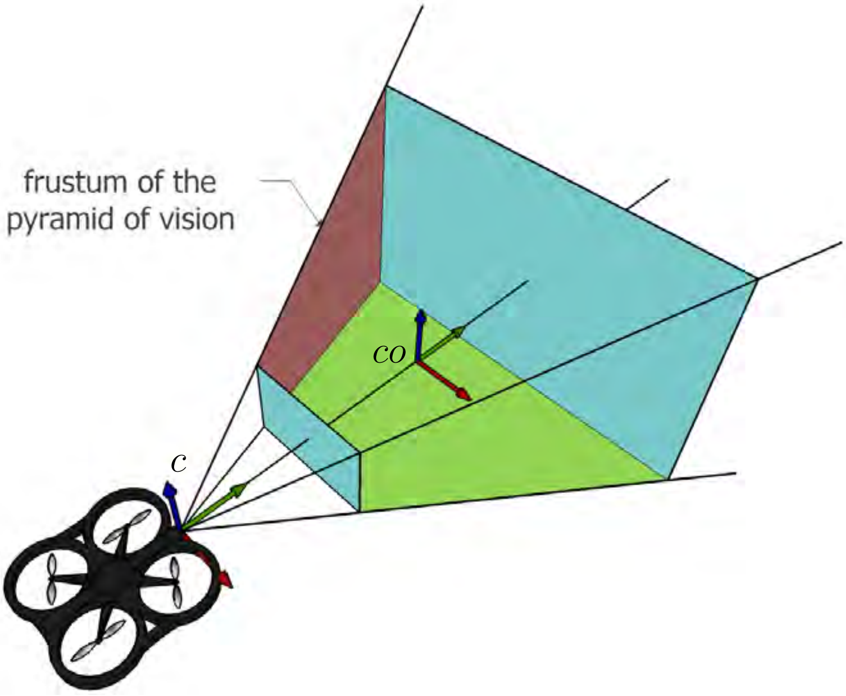


Figure 6.2 The frames of the UAV: the frame c attached to the camera, and the frame co attached to the frustum of the pyramid of vision of the camera.

- co: the frame attached to a frustum of the pyramid of vision of the camera (Sect. 3.4).

  The origin of co is located on the optical axis and at a constant distance from the image center and inside the pyramid of vision of the camera (Fig. 6.2). Note that the projection of the origin of co into the image plane is coincident with the image center of the camera.

  For convenience, the origin of co will be called the *center of the frustum*.

  From the previous definition it follows that

  $$\boldsymbol{T}_{co}^{s} = \boldsymbol{T}_{c}^{s}\,\boldsymbol{T}_{co}^{c} \tag{6.2}$$

  with $\boldsymbol{R}_{co}^{c}$ and $\boldsymbol{p}_{co}^{c}$ being a constant matrix and vector, respectively, since the frames c and co are coupled. In particular, the orientation is set to $\boldsymbol{R}_{co}^{c} = \boldsymbol{I}$.

- r: the robot end-effector frame.

During a teleoperation task, $\boldsymbol{T}_{mi}^{mo}$ contains the information of the operator command to the system through the haptic. $\boldsymbol{p}_{mi}^{mo}$ is scaled by $\boldsymbol{K}_S = \text{diag}(k_{S_x}, k_{S_y}, k_{S_z})$ as $\boldsymbol{p}_{si}^{so} = \boldsymbol{K}_S\,\boldsymbol{p}_{mi}^{mo}$. The scales need to be adjusted separately depending on the X, Y or Z dimension. This is convenient because it allows the adjustment between the haptic subspace in the haptic workspace (mo) and the haptic subspace counterpart in the MM workspace (so). $\boldsymbol{T}_{si}^{so}$ corresponds to the input of the user from the haptic in the MM workspace and is composed of a translation, $\boldsymbol{K}_S\,\boldsymbol{p}_{mi}^{mo}$, and a rotation, $\boldsymbol{R}_{mi}^{mo}$, respectively.

Given that $\boldsymbol{T}_{si}^{s} = \boldsymbol{T}_{so}^{s}\,\boldsymbol{T}_{si}^{so}$ it follows that

$$\begin{aligned} \boldsymbol{p}_{si}^{s} &= \boldsymbol{p}_{so}^{s} + \boldsymbol{R}_{so}^{s}\,\boldsymbol{K}_S\,\boldsymbol{p}_{mi}^{mo} \\ \boldsymbol{R}_{si}^{s} &= \boldsymbol{R}_{so}^{s}\,\boldsymbol{R}_{mi}^{mo}. \end{aligned} \tag{6.3}$$

From the bottom figure in Figs. 6.1 it can also be noted that

$$\boldsymbol{T}_{c}^{s} = \boldsymbol{T}_{so}^{s}\,\boldsymbol{T}_{c}^{so} \tag{6.4}$$

where $\boldsymbol{T}_{c}^{so}$ is a transformation that couples the frames c and so, and which is set to a constant transformation so that the desired position and orientation of the camera can be computed by specifying the frame so pose.

Finally, $\boldsymbol{T}_{si}^{s}$ is taken as the desired pose of the robot, and $\boldsymbol{T}_{r}^{s}$ is obtained from the computation of the mobile manipulator inverse kinematics (Sect. 6.2.2).

The proposed approach extends the solution in (Dominjon et al., 2005) to a mobile manipulator, and the technique is also applied to the orientation by using a free-flying camera.

6.2.2 Inverse kinematics

The algorithm to command the end-effector follows the work by Siciliano et al. (2008):

$$\dot{\boldsymbol{q}}_d = \boldsymbol{J}^{+} \begin{bmatrix} \dot{\boldsymbol{p}}_{si}^{s} + \boldsymbol{K}_p\,\boldsymbol{e}_p \\ \boldsymbol{\omega}_{si}^{s} + \boldsymbol{K}_o\,\boldsymbol{e}_o \end{bmatrix} \tag{6.5}$$

where

- $\boldsymbol{q}_d$ is the desired configuration of the mobile manipulator as in Eq. 6.1.
- $\boldsymbol{e}_p = \boldsymbol{d}(\boldsymbol{p}_{si}^{s} - \boldsymbol{p}_{r}^{s}, D_{max})$; with $\boldsymbol{d}(\cdot)$ such that the imposed error is bounded following the work of (Buss and Kim, 2004):

$$\boldsymbol{d}(\boldsymbol{x}, x_{max}) = \begin{cases} \frac{x_{max}}{\|\boldsymbol{x}\|}\boldsymbol{x}, & \text{if } \|\boldsymbol{x}\| > x_{max} \\ \boldsymbol{x}, & \text{otherwise} \end{cases}$$

- $\boldsymbol{e}_o = \eta_R\,\boldsymbol{\epsilon}_D - \eta_D\,\boldsymbol{\epsilon}_R - \boldsymbol{S}\,(\boldsymbol{\epsilon}_D)\,\boldsymbol{\epsilon}_R$; with η_i and $\boldsymbol{\epsilon}_i$ as the scalar and vector parts of the quaternion $Q_i = \{\eta_i, \boldsymbol{\epsilon}_i\}$, $R = \boldsymbol{R}_{r}^{s}$ and $D = \boldsymbol{R}_{si}^{s}$.

with $\boldsymbol{S}(\cdot)$ being the cross product matrix.

Once $\boldsymbol{q}_d$ is obtained by integrating $\dot{\boldsymbol{q}}_d$, and imposed to the robot, the robot end-effector pose $\boldsymbol{T}_{r}^{s}(\boldsymbol{q})$ can be computed through the direct kinematics using the current robot configuration $\boldsymbol{q}$. This algorithm allows a real-time tracking of the robot end-effector in position and orientation.

The computation of $\boldsymbol{T}_{si}^{s}$ to obtain $\boldsymbol{p}_{si}^{s}$ and $\boldsymbol{R}_{si}^{s}$ is presented in the next Section.

6.2.3 The Position Mapping

Regarding the position, when the user commands the tip of the haptic device *inside* the haptic subspace, the end-effector is commanded using a *position-position* mapping between the haptic tip position and the robot end-effector. When the boundary of the haptic subspace, roughly defined as a thick frontier containing the external workspace boundary, is reached, a *position-linear velocity* component is added to the previous *position-position*. This combination of inputs allows the user to command the end-effector of the mobile manipulator in an unbounded workspace through the bounded workspace of the haptic device.

A "cylindric bubble" is used as the haptic subspace by adapting the solution by Dominjon et al. (2005) to the workspace of mobile manipulator (Figs. 6.1 and 6.3): the Z position is commanded by a *position-position* map (Eq. 6.3) and the *position-linear velocity* is activated in the XY plane when the distance, D_V, from the haptic tip to the Z axis of the frame mo, is higher than a predefined radius R_V. Formally this can be expressed as

$$\dot{\boldsymbol{p}}^s_{so} = \left(1 - \frac{R_V}{D_V}\right) k_V\, \boldsymbol{u}^s_{si_{XY}} = \left(1 - \frac{R_V}{D_V}\right) k_V\, \boldsymbol{R}^s_{so}\, \boldsymbol{K}_S\, \boldsymbol{u}^{mo}_{mi_{XY}} \qquad \text{if } D_V > R_V \tag{6.6}$$

with $k_V > 0$; $D_V = \|\boldsymbol{p}^{mo}_{mi_{XY}}\|$; and $\boldsymbol{u}^{mo}_{mi_{XY}}$, the unit vector of $\boldsymbol{p}^{mo}_{mi_{XY}} = \begin{bmatrix} p^{mo}_{mi_x} & p^{mo}_{mi_y} & 0 \end{bmatrix}^T$. The term $(1 - R_V/D_V)$ allows for increasing velocities from zero $(D_V = R_V)$ to the maximum defined by k_V $(D_V \to \infty)$. Thus, the farther away the haptic tip is from the vertical axis of its workspace, the fastest the MM will move in the equivalent direction.

Differentiating the translation in Eq. 6.3, and using Eq. 6.6, both Equations can be merged:

$$\dot{\boldsymbol{p}}^s_{si} = \left(1 - \frac{R_V}{D_V}\right) k_V\, \boldsymbol{R}^s_{so}\, \boldsymbol{K}_S\, \boldsymbol{u}^{mo}_{mi_{XY}}\, \chi_{[R_V,\infty)}\left(D_V\right) + \boldsymbol{R}^s_{so}\, \boldsymbol{K}_S\, \dot{\boldsymbol{p}}^{mo}_{mi} \tag{6.7}$$

where $\chi_C(x)$ is the indicator function: it returns one if $x \in C$ and zero otherwise.

Additionally, in order to make the operator feel that he/she has entered in the *position-linear velocity* area, a force $\boldsymbol{f}^{mo}_{mi}$ can be applied to him by the haptic device, with the force pointing towards the haptic workspace Z axis, and proportional to the distance from the tip to the Z axis:

$$\boldsymbol{f}^{mo}_{mi} = -\left(1 - \frac{R_V}{D_V}\right) k_f\, \boldsymbol{R}^s_{so}\, \left(\boldsymbol{K}_S\, \boldsymbol{u}^{mo}_{mi_{XY}} - \boldsymbol{K}_d\, \dot{\boldsymbol{u}}^{mo}_{mi_{XY}}\right) \chi_{[R_V,\infty)}\left(D_V\right) \tag{6.8}$$

Note that a damping has been added in the previous expression to prevent the platform from moving if the user releases the haptic tip.

6.2.4 The Orientation Mapping

Differently from a fixed camera, which only allows to view the scenario from a particular orientation, the free-flying camera allows the operator to have a visual feedback when using a similar

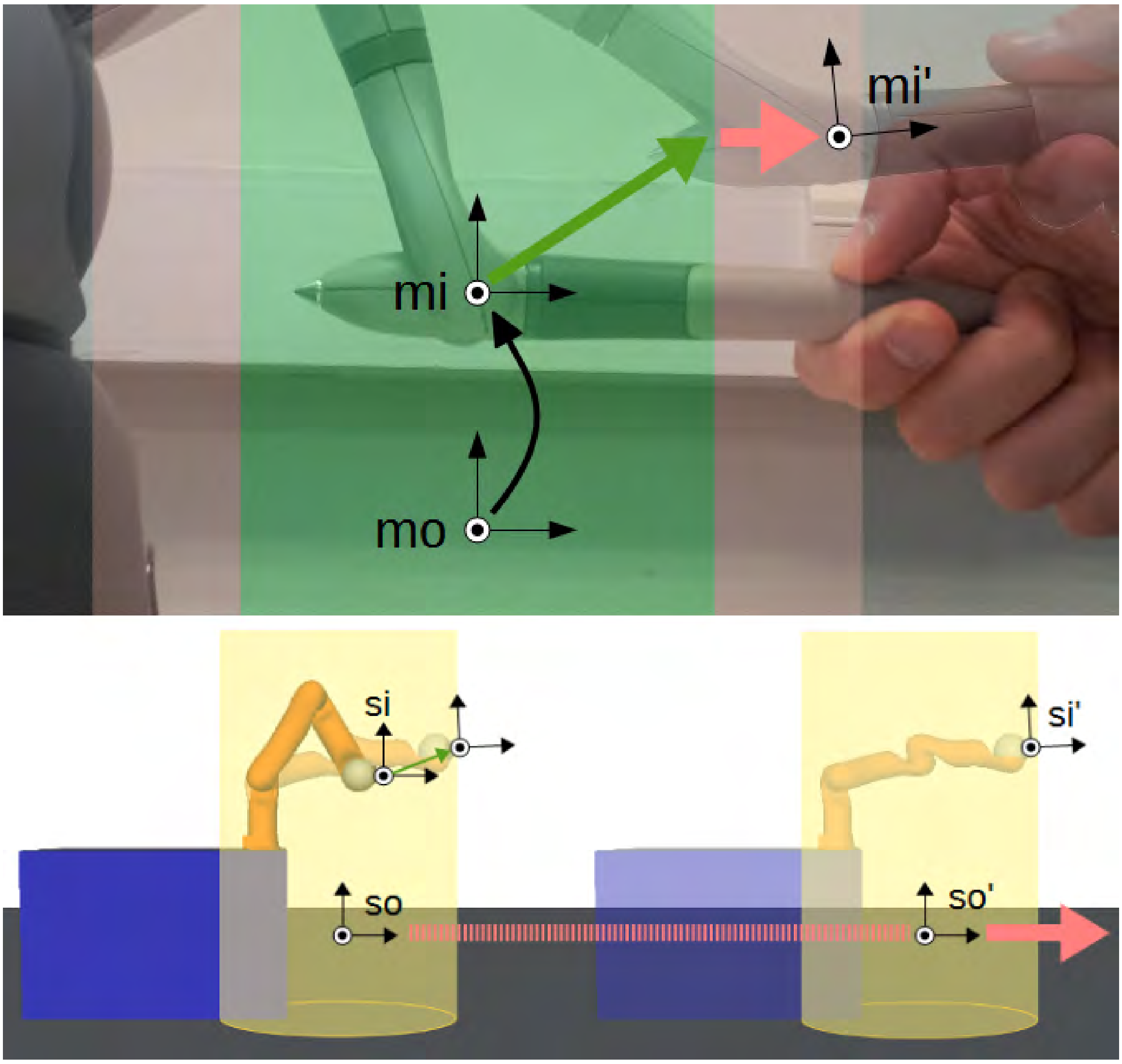


Figure 6.3 Position mapping. Top figure: in the green area the haptic position is mapped to a translation of the haptic subspace counterpart inside the mobile manipulator workspace (yellow volume in the bottom figure). Bottom figure: in the pink area (boundaries of the haptic workspace) the position is also mapped to the linear velocity and a motion on the platform is generated.

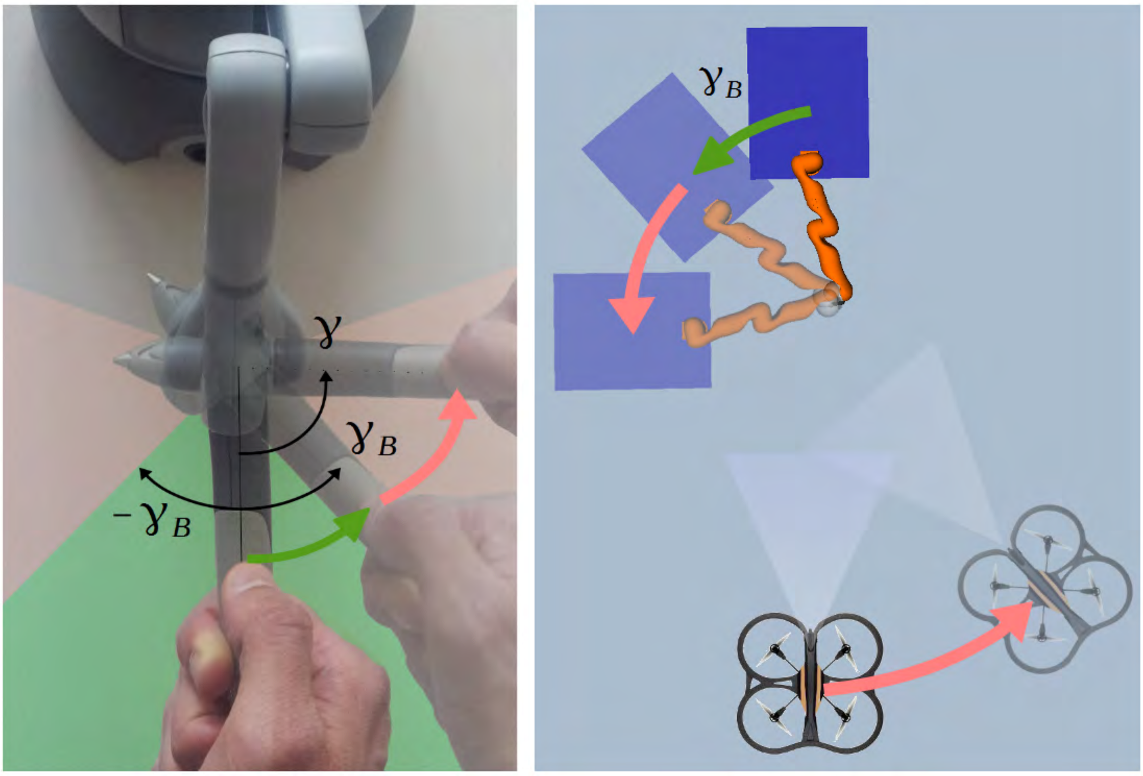


Figure 6.4 Rotation mapping. The haptic orientation around the Z axis in the green area is mapped to the camera orientation. In the pink area the orientation is also mapped to an angular velocity.

mode on the orientation to the *position-linear velocity* mode, enabling an *orientation-angular velocity* map.

The *orientation-angular velocity* control is activated when the haptic orientation surpasses a predefined boundary. This boundary can be defined using the ZYX intrinsic Euler angles as

$$\boldsymbol{R}_{mi}^{mo} = \begin{bmatrix} c_\beta c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ c_\beta s_\gamma & s_\alpha s_\beta s_\gamma + c_\alpha d_\gamma & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma \\ -s_\beta & s_\alpha c_\beta & c_\alpha c_\beta \end{bmatrix} \tag{6.9}$$

with $s_a = \sin(a)$ and $c_a = \cos(a)$.

The Euler angles can now be retrieved as $\alpha = \text{atan2}(r_{32}, r_{33})$, $\beta = \text{atan2}\left(r_{31}, \sqrt{{r_{11}}^2 + {r_{21}}^2}\right)$ and $\gamma = \text{atan2}(r_{21}, r_{11})$.

The *orientation-angular velocity* mode gets active when $|\beta| > \beta_B$ and $|\gamma| > \gamma_B$. Next, the frame $\boldsymbol{R}_{so}^{s}$ is rotated around Z_{so}^{s}, its the vertical axis, according to the rotation direction specified by the user. The angular velocity is proportional to the amount of penetration of the user into the *orientation-angular velocity* zone (Fig. 6.4, pink zone) inside the haptic subspace.

Algorithm 1: The workspace mapping algorithm.

```
input  : $\boldsymbol{T}_{mi}^{mo}$, $\boldsymbol{T}_{so}^{s}$ and $\boldsymbol{T}_{c}^{so}$
output: $\boldsymbol{T}_{si}^{s}$, $\boldsymbol{T}_{c}^{s}$
compute the position mapping:
    • $\boldsymbol{p}_{mi_{XY}}^{mo} = \begin{bmatrix} p_{mi_x}^{mo} & p_{mi_y}^{mo} & 0 \end{bmatrix}^T$
    • $D_V = \|\boldsymbol{p}_{mi_{XY}}^{mo}\|$
    • if $D_V > R_V$:
        $\boldsymbol{p}_{so}^{s} = \boldsymbol{p}_{so}^{s} + \left(1 - \frac{R_V}{D_V}\right) k_V \, \boldsymbol{R}_{so}^{s} \, \boldsymbol{K}_S \, \frac{\boldsymbol{p}_{mi_{XY}}^{mo}}{D_V} \, \Delta t$
compute the orientation mapping:
    • $\boldsymbol{R}_{mi}^{mo} = [r_{ij}]$
    • $\beta = \text{atan2}\left(r_{31}, \sqrt{{r_{11}}^2 + {r_{21}}^2}\right)$
    • $\gamma = \text{atan2}\left(r_{21}, r_{11}\right)$
    • if $|\beta| > \beta_B$ and $|\gamma| > \gamma_B$:
        $\boldsymbol{R}_{so}^{s} = \left(\boldsymbol{I} + \text{sign}(\gamma)\left(1 - \frac{\gamma_B}{|\gamma|}\right) k_R \, \boldsymbol{S}\left(\boldsymbol{n}_z\right) \, \Delta t\right) \boldsymbol{R}_{so}^{s}$
compute the next end-effector transformation:
    • $\boldsymbol{T}_{si}^{s} = \begin{bmatrix} \boldsymbol{R}_{so}^{s}\boldsymbol{R}_{mi}^{mo} & \boldsymbol{p}_{so}^{s} + \boldsymbol{R}_{so}^{s}\boldsymbol{K}_S\boldsymbol{p}_{mi}^{mo} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$
compute the next camera transformation:
    • $\boldsymbol{T}_{c}^{s} = \boldsymbol{T}_{so}^{s} \, \boldsymbol{T}_{c}^{so}$
```

By differentiating Equation 6.3, $\boldsymbol{\omega}_{si}^{s} = \boldsymbol{\omega}_{so}^{s} + \boldsymbol{R}_{so}^{s} \, \boldsymbol{\omega}_{mi}^{mo}$ is obtained, where each ω corresponds to an angular velocity.

Then, the *orientation-angular velocity* mode can be imposed with

$$\boldsymbol{\omega}_{so}^{s} = \text{sign}(\gamma) \left(1 - \frac{\gamma_B}{|\gamma|}\right) k_R \, \boldsymbol{n}_z \, \chi_{[\gamma_B, \infty)}\left(\gamma\right)$$

where $\boldsymbol{n}_z$ is the unit vector of the Z axis of the rotation matrix $\boldsymbol{R}_{so}^{s}$, that is, $\boldsymbol{n}_z = [0 \quad 0 \quad 1]^T$.

Using Eq. 6.4, $\boldsymbol{T}_{c}^{s}$ can be obtained by updating $\boldsymbol{T}_{so}^{s}$ with the computed $\boldsymbol{p}_{so}^{s}$ and $\boldsymbol{R}_{so}^{s}$.

The algorithm is summarized in Algorithm 1.

6.3 The User Study

A study has been conducted in order to evaluate:

- The dexterity in the use of the teleoperation system.
- The load of the teleoperation system on the operator.
- The progression in the learning and improvement in the teleoperation skills.

A virtual scenario has been designed where the operator has had to track a moving sphere with the BMM-I end-effector while simultaneously keeping a yellow spot on the sphere facing towards the camera (Fig. 6.5). As the sphere has been randomly and smoothly translating and rotating around the Z axis (and so, the spot orientation), a high degree of coordination between the end-effector and the view orientation of the camera is needed. An index has been obtained for each trial, which value is 0 when the performance is insufficient and 1 when perfect. The index has been computed as the mean of

$$I = \begin{cases} 0 & \text{if } \alpha > \alpha_0 \\ \left(1 - \frac{\alpha}{\alpha_0}\right) e^{-2(D-D_0)} & \text{if } \alpha \leq \alpha_0 \text{ and } D > D_0 \\ 1 - \frac{\alpha}{\alpha_0} & \text{if } \alpha \leq \alpha_0 \text{ and } D \leq D_0 \end{cases} \tag{6.10}$$

at every sample time during the experiment, and where:

- α is the absolute value of the angle between the normal on the center of the spot and the projection on the XY plane of the vector from the center of the sphere to the robot end-effector.
- α_0 is the boundary of the range of values of α where its contribution on the performance of the teleoperation is computed. If α is bigger than α_0 the teleoperation performance is considered insufficient ($I = 0$).
- D is the euclidean distance from the center of the sphere to the end-effector position of the mobile manipulator.
- D_0 is a radius that defines a sphere which center is coincident with the tracked sphere center. If the end-effector of the MM is inside this sphere ($D \leq D_0$) the performance of the teleoperation is considered perfect regarding the translation.

Fig. 6.6 depicts the parameters α and D.

The values $\alpha_0 = \frac{\pi}{2}$ rad and $D_0 = 0.4$ m have been used. The desired position of the camera w.r.t. the haptic subspace counterpart in the scenario has been set to $\boldsymbol{p}_c^{so} = [3 \quad 0 \quad 5]^T$.

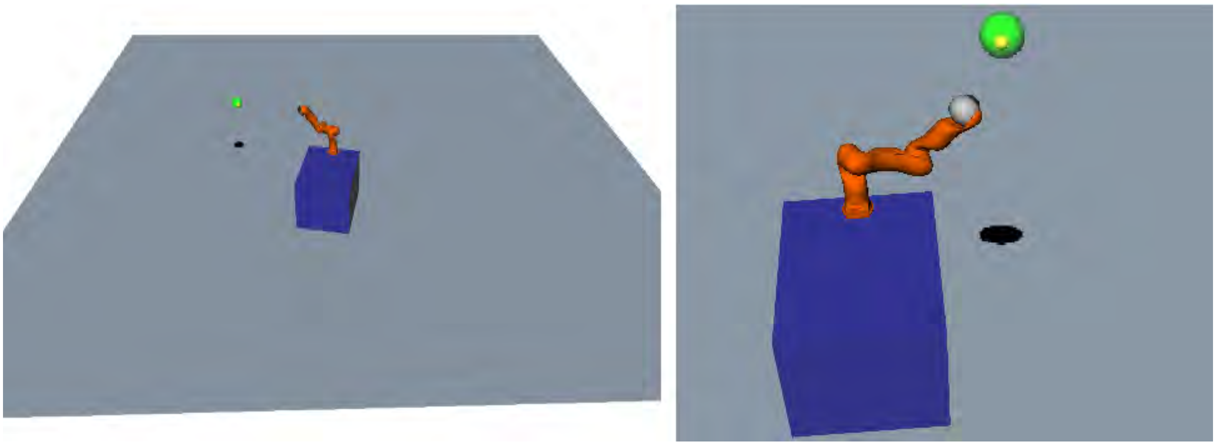


Figure 6.5 The virtual BMM-I and the sphere with the spot in the virtual scenario. A close up is shown in the right figure.

The random motion of the sphere has been achieved using a potential field. An attractive potential field has been randomly generated every 5 to 8 seconds by placing an attractive pole in the scenario, defined as a position in the XY plane and an orientation around its Z axis. A repulsive potential field has been set at the scenario limits, so that the sphere would stay inside the limits of the scenario.

The experiment has been performed with eight participants (with an age average of 27.6 years) with little or no experience with haptic devices. Four sessions have been conducted per participant in consecutive days. In each session five experiments have been executed. Each experiment lasted for 1 min. After each set of experiments the participants have completed a NASA-TLX (Hart, 2006) test to measure the perceived workload. The first session has included a description of the teleoperation system and the experiment itself, all with a duration of 20 minutes. The overall duration of the next sessions has been around 15 minutes per participant and day. Appendix D shows the questionnaire that the users had to fill after the experiment.

The teleoperation system has been tested using a PC with an Intel Core i7 at 2.80GHz and Debian OS. The software has been implemented in C++ using ROS and the Kautham Project (Rosell et al., 2014). The Phantom Omni has been used to teleoperate the virtual BMM-I.

6.3.1 Results

Two main set of results have been obtained, one regarding the dexterity in the teleoperation of the users, and the other regarding the workload on the operators during the task.

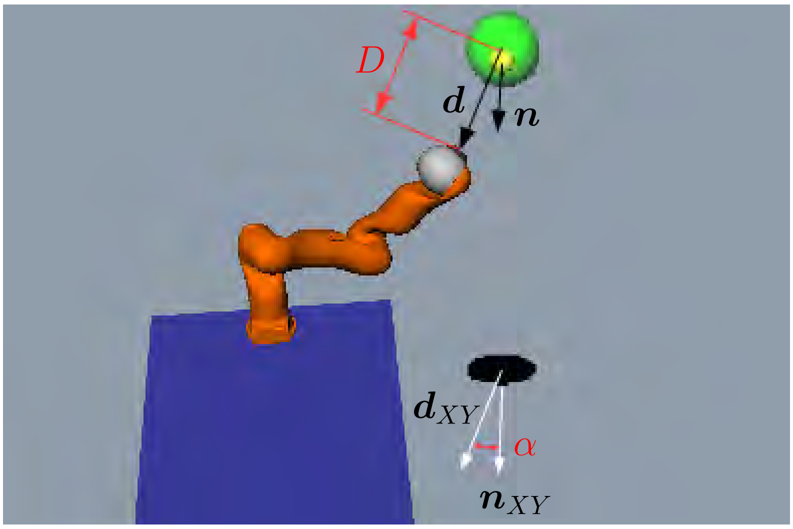


Figure 6.6 The α and D parameters of the index to evaluate the user performance. α is computed as the norm of the angle between the vectors $\boldsymbol{d}_{XY}$ and $\boldsymbol{n}_{XY}$; where $\boldsymbol{d}_{XY}$ is the projection on the floor (XY plane) of the vector, $\boldsymbol{d}$, from the center of the green sphere to the robot end-effector, and $\boldsymbol{n}_{XY}$ is the projection on the floor of the normal, $\boldsymbol{n}$, on the *tip* of yellow sphere attached to the green one. The projection (in black, as a shadow) of the green sphere on the floor can be appreciated at the right next to the platform in the image plane.

The dexterity performance scores

Regarding the dexterity level during the teleoperation, computed using Eq. 6.10, first the Saphiro-Wilk normality test (Shapiro and Wilk, 1965) has been performed on the data (Fig. 6.7). The data consisted on 40 values (5 trials per participant) per day. The data has been found to be non-normal for each day, mainly due to the skewness of the data (0.13 for the first training day and -1.63 for the fourth), the bounds on the underlying distribution and some isolated values.

Considering the operators on their first day of training as the population, and on the fourth as the sample, the null hypothesis "*the operators do not improve their teleoperation skills after four days of training in the teleoperation of the combined mobile manipulator and free-flying camera teleoperation system*" can be rejected by doing a Wilcoxon Rank Sum Test (Wilcoxon, 1945), a test suitable for non-normal data. The ranksum of the Wilcoxon test is of 937, and thus the p-value is almost zero.

The difference between two measures of the discrete distributions has been compared to have an intuitive idea of the magnitude of the increase in dexterity. The averages of the scores (41.77 for day 1, and 72.80 for day 4) are compared also against the dexterity of an expert user. Taking the mean of the score of the expert user (90.12) as an upper bound, and the lowest score (9.70) as a lower, the percentage of dexterity gain obtained can be computed as

$$\frac{72.80 - 41.77}{90.12 - 9.70} = 38.58\%.$$

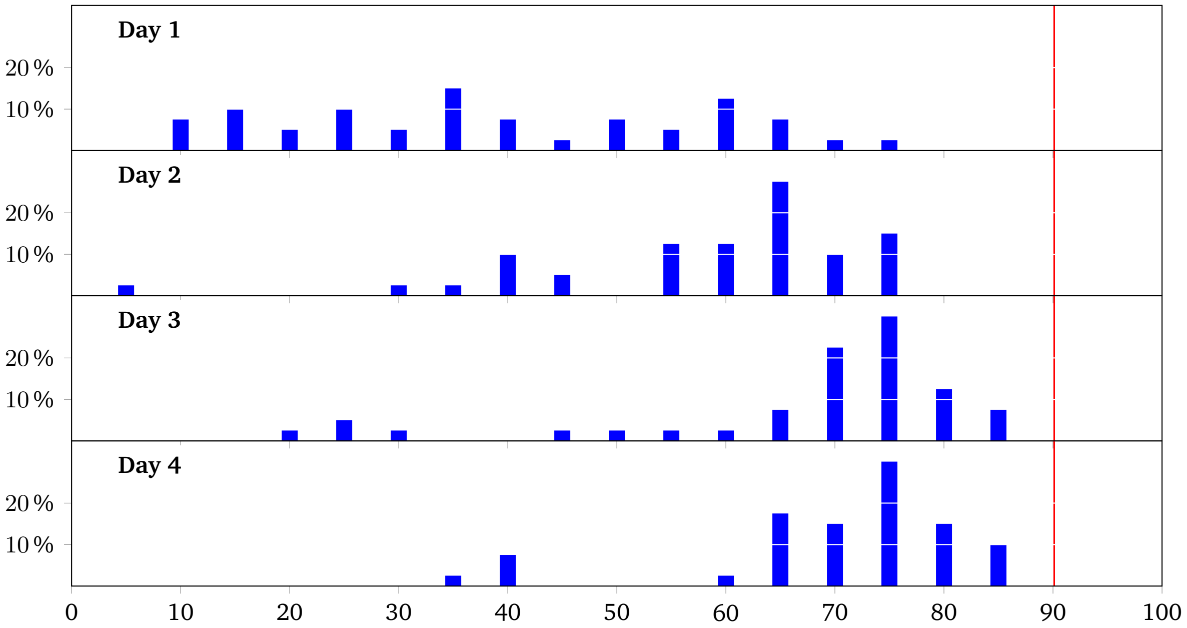


Figure 6.7 Score histograms: higher scores (horizontal axis) mean more dexterity in the teleoperation. The value 10 in the horizontal axis corresponds to scores from 10 to 15, and so on. Red vertical line: expert score. Given a particular range of scores specified by its corresponding tick in the horizontal axis, the vertical axis gives the percentage of participants whose score was within that range.

The NASA-TLX scores

The second evaluation has focused on the operator workload during the teleoperation task and the difference between untrained and trained operators (Fig. 6.8).

A Saphiro-Wilk test has been computed on the NASA-TLX scores using the data of the first and fourth training days to verify its normality. With a p-value of 0.10 the W values obtained have been 0.9079 and 0.9446 for day 1 and day 4, respectively, thus the data can be treated as normal. The NASA-TLX scores mean and standard deviation on the first day have been 52.54 (from a minimum of 0 to a maximum of 100) and 15.75, respectively. After the training, the average and standard deviations decreased to 41.79 and 6.18 (Fig. 6.9).

With this data the null hipothesis "*the workload during the teleoperation on the operators does not decrease after four days of training in the teleoperation of the combined mobile manipulator and free-flying camera teleoperation system*" can be rejected with a probability of 97.32% (Z = -1.93). The mean value of the workload is between 30.88 and 52.70 with a 95% confidence, an interval almost in its totality under the average workload of the untrained operators (52.54).

Contrary to the dexterity scores, the underlying normality of the NASA-TLX data permits its intepretation in a simple manner through the effect size (ES) (Fritz et al., 2012; Coe, 2002).

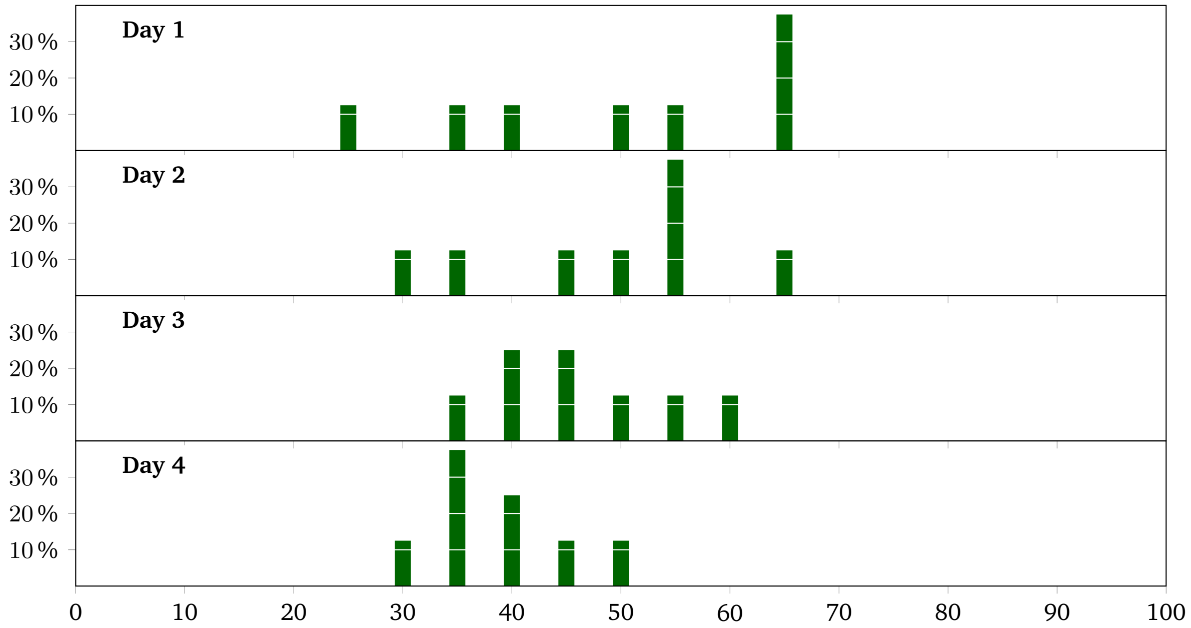


Figure 6.8 NASA-TLX histograms. A lower score corresponds to a less workload on the operator. Given a particular range of NASA-TLX scores specified by its corresponding tick in the horizontal axis, the vertical axis gives the percentage of participants whose score was within that range.

The ES eases the interpretation of the results obtained in a statistical analysis and has been widely studied in the statistics community. The ES of the difference in the workload during the teleoperation between the untrained operators and after 4 days of training is -0.91, corresponding to a 81.79% probability that the average trained user will feel less workload than a random untrained operator, and between -1.94 and 0.12 with a 95% confidence (45.15% to 97.36%, respectively). Two interpretations can light up its significance, the rank interpretation and the Common Language Effect Size (CLES) (McGraw and Wong, 1992):

- *Rank* = 18.21. In a untrained group of 100 members the rank 18 indicates that only eighteen untrained members feel less workload than the average trained operator. In an hypothetic group of 100 operators, after the training, half of the operators feel less workload than 82 of the 100 untrained operators, thus improving a 64%.
- *CLES* = 73.78%: is a 73.78% probability that a trained operator will perceive less workload than an untrained.

6.4 Chapter Contributions

The main contribution of this Chapter is the presentation of a novel teleoperation system consisting of a ground mobile manipulator, a UAV and a haptic device. The system allows a operator

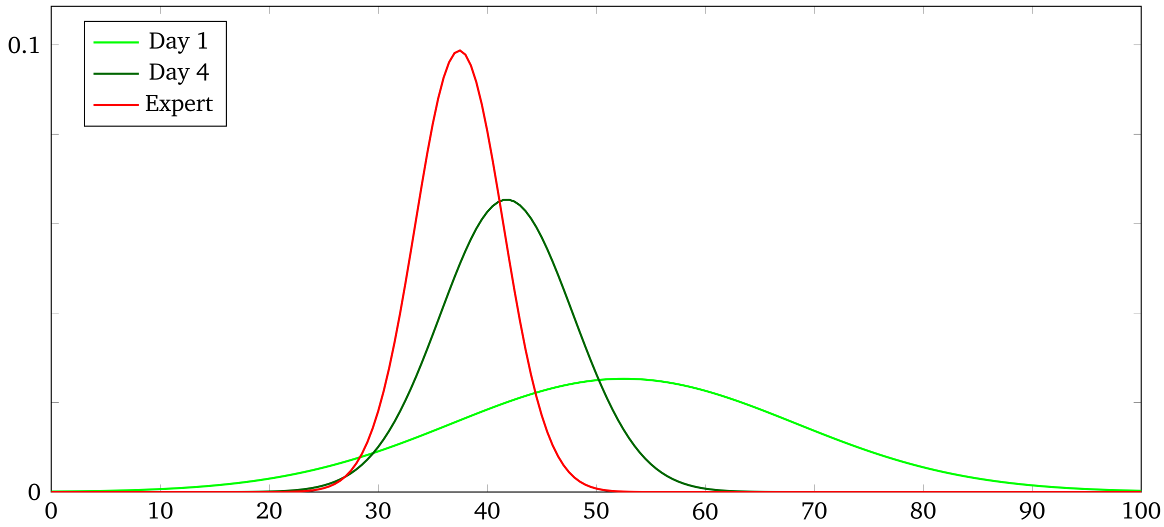


Figure 6.9 NASA-TLX normalized scores. The horizontal axis represents the NASA-TLX scores. Each curve corresponds to the normal distribution that best describes the scores of the participants in days 1, 4 and of the expert user. The improvement of the perceived workload on the participants can be appreciated as a decrease in the NASA-TLX score from day 1 to day 4, as well as in a narrowing of the gaussians.

to command the mobile manipulator end-effector using the haptic device while he/she is getting visual feedback from the UAV camera.

A workspace mapping algorithm that allows the operator to command the mobile manipulator in an unbounded workspace using the haptic device without explicitly commanding the UAV is presented. This allows him/her to focus on the teleoperation of the object attached to the mobile manipulator end-effector.

A user study has been conducted to evaluate the goodness of the proposed algorithm. The results show that training increases the performance and dexterity of the operator, and lowers his/her workload. It can be thus concluded that the proposed approach is suitable for the teleoperation of the novel robotic system.

Chapter 7

Enhancing the Continuous Inverse

The bird fights its way out of the egg. The egg is the world. Who would be born must first destroy a world. The bird flies to God. That God's name is Abraxas.

Hermann Hesse, Demian

SUMMARY: In this chapter the continuous inverse is shown to be suitable for the teleoperation robotic system to handle the transitions during the activation of different hierarchical tasks. A flaw in its behavior is shown along with the corresponding enhancement. Further, a variation is proposed to keep low the error in the lower priority levels during long transitions.

7.1 Introduction

A common situation occurs when a robot is tracking a desired pose with its end-effector and a joint reaches one of its limit. At this point, a certain policy has to be taken, whether to clamp its value or move it back away from the limit (Baerlocher and Boulic, 2004; Raunhardt and Boulic, 2007b). The task priority formulation offers an elegant mean to solve this issue, that is, to take control over the joint values, while still allowing another task to be executed. A solution is to use the task priority framework in such a way that the highest priority task corresponds to the joint limit avoidance, and so that this task is only activated when the corresponding joint is close to its limit.

But, again, the activation and deactivation of a joint limit avoidance task produces a discontinuity that is carried through the lower levels of priority, thus leading to an undesirable behaviour of the robot. This topic has recently been addressed by several authors (Mansard et al., 2009b; Petrič and Žlajpah, 2011; Dietrich et al., 2012).

The continuous inverse has been applied in different scenarios, both in classic robots and humanoids (Mansard et al., 2009b,a; Colomé and Torras, 2015; Li et al., 2012). It allows the activation and deactivation of multiple tasks in a controlled way such that discontinuities do not appear nor are carried to the lower levels of priority.

In the present work, the solution presented by Mansard et al. (2009b) has been adopted and some corrections and improvements are introduced. The proposed continuous inverse is used to manage multiple tasks with different levels of priority, in which the highest priority tasks correspond to the joint limit avoidance task.

This operator, though, is better suited for use in fast activation and deactivation of tasks, as the priority between the different task levels is not guaranteed during the transitions.

Long transitions in time can occur in situations where a task is demanding a motion of the robot end-effector that needs that a particular joint already in the vicinity of one of its limits gets even closer to it. The continuous inverse is studied when long task transitions occur in time in a robotic system with multiple tasks and different levels of priority.

This study brings to evidence several issues of the continuous inverse that can pose a significant hazard during its execution. An unboundedness on the projection operator onto the null space for lower priority levels may lead to extreme instabilities. Also, the propagation of the lower priority task errors to higher priority tasks is exposed.

A correction for both issues is presented, which enhances the continuous inverse and makes it more suitable for situations with long task transitions.

A teleoperation scenario with different tasks is presented in order to validate the proposed solutions. The tasks imposed to the robot are the joint limit avoidance, the end-effector tracking and an independent motion of the platform.

A very demanding trajectory is fed to the robot end effector and the error of each task is shown to diminish with each improvement proposed until no significant error is left. Overall, this work presents a necessary correction on the continuous inverse and a proper tool to deal with long task transitions in complex robotic systems.

7.2 Limitations of the Continuous Inverse

7.2.1 On the boundaries of the null space matrix

As stated previously, an assumption of Eq. 3.8 is that the singular values of $\boldsymbol{W}$ are real and lie inside the unit interval $[0,1]$. This is appropriate, as otherwise, $\boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}$ could lie outside the "transition zone". For example, if a value x has to change smoothly from 1 to 5, it is expected that during the transition, the intermediate value is never lower than 1 nor higher than 5, that is, $x = 1 + \lambda(5-1) \in [1,5]$, and so the activation parameter λ has to lie in the interval $[0,1]$.

Mansard et al. (2009b) use the right continuous inverse to compute the Jacobian null space projection matrix of the lower level of priority as

$$\boldsymbol{N}_\oplus^1 = \boldsymbol{I} - \boldsymbol{J}_1^{\oplus\boldsymbol{H}}\boldsymbol{J}_1 \tag{7.1}$$

for the first priority level, and

$$\boldsymbol{N}_\oplus^i = \boldsymbol{N}_\oplus^{i-1} - \boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\boldsymbol{J}_i \tag{7.2}$$

for lower priority levels. This, together with

$$\dot{\boldsymbol{q}}_i = \begin{cases} \boldsymbol{J}_i^{\oplus\boldsymbol{H}}\dot{\boldsymbol{x}}_i & i=1 \\ \boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\left(\dot{\boldsymbol{x}}_i - \boldsymbol{J}_i\boldsymbol{J}_{i-1}^{\oplus\boldsymbol{H}}\dot{\boldsymbol{x}}_{i-1}\right) & \text{otherwise} \end{cases} \tag{7.3}$$

allows, through integration, the full computation of the new configuration $\boldsymbol{q}$ given the set of tasks with its corresponding hierarchy.

But the singular values of $\boldsymbol{N}_\oplus^i$ may not always lie inside the unit interval. This can be demonstrated through a simple counter-example. Take $\boldsymbol{J}_1$ and $\boldsymbol{H}$ as

$$\boldsymbol{J}_1 = \begin{bmatrix} \boldsymbol{r}_{11} \\ \boldsymbol{r}_{12} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\boldsymbol{H} = \begin{bmatrix} h_1 & 0 \\ 0 & h_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

with $\boldsymbol{r}_{ij}$ being the row j of $\boldsymbol{J}_i$.

Thus, the continuous inverse, $\boldsymbol{J}_1^{\oplus\boldsymbol{H}}$, as in Eq. 3.4 results in

$$\boldsymbol{J}_1^{\oplus\boldsymbol{H}} = h_1(1-h_2)\begin{bmatrix} \boldsymbol{r}_{11} \\ \boldsymbol{0} \end{bmatrix}^+ + (1-h_1)h_2\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{r}_{12} \end{bmatrix}^+ + h_1h_2\boldsymbol{J}_1^+ = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{r}_{12}^+ \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

And now $\boldsymbol{N}_\oplus^1$ can be computed along with its SVD decomposition as

$$\boldsymbol{N}_\oplus^1 = \boldsymbol{I} - \boldsymbol{J}_1^{\oplus\boldsymbol{H}}\boldsymbol{J}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \boldsymbol{U}_1\boldsymbol{S}_1\boldsymbol{U}_1^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Given the lower priority task Jacobian, $\boldsymbol{J}_2 = [\boldsymbol{c}_{21} \quad \boldsymbol{c}_{22}]$, with $\boldsymbol{c}_{ij}$ as the column j of $\boldsymbol{J}_i$, there is everything needed to compute $\boldsymbol{J}_2^{\boldsymbol{N}^1_\oplus}$ from Eq. 3.7 and Eq. 3.8:

$$\begin{aligned}\boldsymbol{J}_2^{\boldsymbol{N}^1_\oplus \oplus} &= \left(\left(\boldsymbol{U}_1^T \boldsymbol{J}_2^T \right)^{\oplus \boldsymbol{s}_1^T} \boldsymbol{U}_1^T \right)^T = \left(\left([\boldsymbol{c}_{21} \; \boldsymbol{c}_{22}]^T \right)^{\oplus [1 \;\; 0]} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)^T \\ &= \left(\begin{bmatrix} \boldsymbol{c}_{21}^T \\ \boldsymbol{c}_{22}^T \end{bmatrix}^{\oplus [1 \;\; 0]} \right)^T = \begin{bmatrix} \boldsymbol{c}_{21}^{+T} & \boldsymbol{0} \end{bmatrix}^T = \begin{bmatrix} \boldsymbol{c}_{21}^{+} \\ \boldsymbol{0} \end{bmatrix}\end{aligned}$$

with $\boldsymbol{s}_1$ being the diagonal vector of $\boldsymbol{S}_1$. Finally, $\boldsymbol{N}^2_\oplus$ can be computed following Eq. 7.2:

$$\boldsymbol{N}^2_\oplus = \boldsymbol{N}^1_\oplus - \boldsymbol{J}_2^{\boldsymbol{N}^1_\oplus \oplus} \boldsymbol{J}_2 = \begin{bmatrix} 1 - \boldsymbol{c}_{21}^{+} \boldsymbol{c}_{21} & -\boldsymbol{c}_{21}^{+} \boldsymbol{c}_{22} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\boldsymbol{c}_{21}^{+} \boldsymbol{c}_{22} \\ 0 & 0 \end{bmatrix}$$

which largest singular value, $\sigma_1(\boldsymbol{N}^2_\oplus) = \left| \boldsymbol{c}_{21}^{+} \boldsymbol{c}_{22} \right|$, will, in general, not lie between 0 and 1.

7.2.2 On the lower priority levels

Although the continuous inverse is an elegant solution for task transitions, it is mainly designed, and its strengths better explained, to deal with fast transitions, and not for robotic system that may stay in a same transition for a long time. For instance, the task priorities cannot be guaranteed during tasks transitions, as explicitly stated in (Mansard et al., 2009a). This, although qualitatively significant, may not be a big issue for fast transitions.

But in certain circumstances, task transitions may take more time. For instance, when the velocity component of the partially activated task with higher priority is canceled by the partially activated secondary task velocity, the robot can, in theory, stay indefinitely with a task in a transition state.

To illustrate this, let's take the simplest possible robot, a single linear joint, $x \in \mathbb{R}$, and implement the continuous inverse with a joint limit activation task as the first priority level ($x_1 = -k\, x$, $k \in \mathbb{R}_{>0}$) and another task at the second priority level, with input $u_2 \in \mathbb{R}$. This system has one activation parameter which we assume in transition, that is, $h \in (0, 1)$, and that depends solely on the position of the robot, *i.e.*, $h = h(x)$. Then, following Eq. 3.6,

$$\dot{q} = -h\, k\, x + (1 - h)(u_2 + h\, k\, x) = (1 - h) u_2 - h^2\, k\, x. \qquad (7.4)$$

So it is easy to see that for a given position x and a certain degree of task activation h it is possible to find a u_2 that leaves the system motionless ($\dot{q} = 0$), *i.e.*, in the transition state, as

$$u_2 = k \frac{h^2}{1 - h} x.$$

What this actually points to is the possibility to have long lasting transitions, and thus to have long periods of time where the task hierarchy is not fulfilled, which may lead to undesirable errors due to lower priority tasks affecting the high priority tasks.

7.2.3 Example

In order to illustrate the issues explained above, the performance results of the teleoperation of a mobile manipulator are presented.

In the example, a stressful trajectory is imposed to the end-effector of the Barcelona Mobile Manipulator I (BMM-I) such that multiple joint limit avoidance tasks get active.

Thus, although the main task is the teleoperation task, that is, the end-effector tracking of a desired pose in the space, the highest priority task becomes the joint limit avoidance, only activated for a particular joint when the joint surpasses a predefined lower or upper threshold. A third task has been imposed at the third priority level: a separation between the end-effector and the center of the platform in the XY plane.

Summarizing, the three levels of priority, from highest to lowest, become:

1. Joint limit avoidance.
2. Position and orientation tracking of a desired pose for the end-effector of the mobile manipulator.
3. A desired separation in the XY plane between the BMM-I end-effector and the center of its platform.

The purpose of the first priority task is to bring the joint values to its middle values when they are close to the joint limits. As the joint configuration $\boldsymbol{q} = [q_i]$ is normalized, then $q_i \in [-1, 1]$, and its joint range middle value is 0. Therefore, the input to the task priority algorithm becomes $\dot{\boldsymbol{x}}_1 = -K_1\,\boldsymbol{q}, K_1 \in \mathbb{R}_{>0}$, which, when active, implies that $\boldsymbol{q} \to \boldsymbol{0}$.

The joint limits are taken into consideration as in (Mansard et al., 2009a). A smooth transition buffer of size $\beta \in \mathbb{R}_{>0}$ is set up so that the joint limit avoidance task is activated smoothly only in the vicinities of the limits:

$$h_i = \begin{cases} f_\beta(\beta - 1 - q_i), & -1 < q_i < -1 + \beta \\ f_\beta(\beta - 1 + q_i), & \quad 1 > q_i > \quad 1 - \beta \\ 0, & \text{otherwise} \end{cases} \tag{7.5}$$

with the smooth increasing function (Fig. 7.1), f_β, defined for $x \in [0, \beta]$ as

$$f_\beta(x) = \frac{1}{2}\left(1 + \tanh\left(\frac{\beta}{\beta - x} - \frac{\beta}{x}\right)\right). \tag{7.6}$$

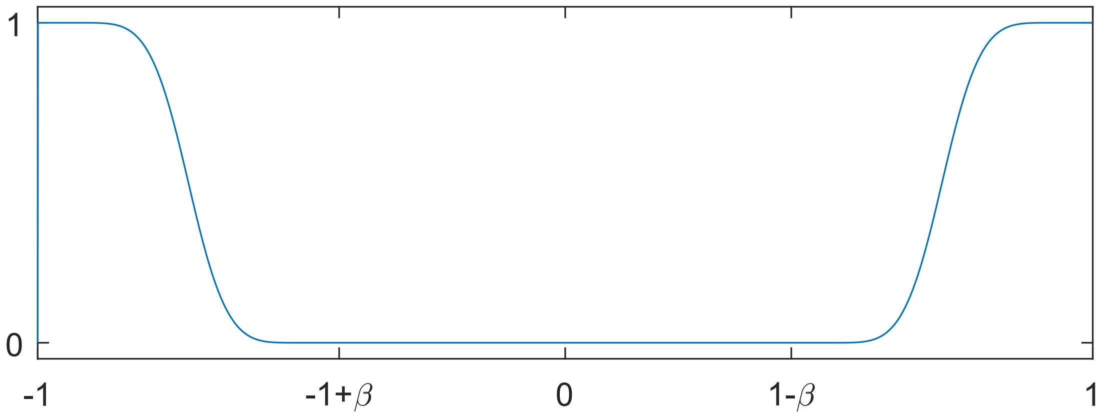


Figure 7.1 Joint smooth transition function over a normalized joint $q_i \in [-1, 1]$.

In the second priority level, a trajectory, both in position and orientation, has been imposed to the end-effector of the BMM-I (see Fig. 7.2a for the position command, and Fig. 7.2b for the orientation).

The end-effector error $\dot{\boldsymbol{x}}_2$ has been computed such that it accounts for both the position and orientation error as described in the previous Chapter (Section 6.2.2), thus obtaining

$$\dot{\boldsymbol{x}}_2 = \begin{bmatrix} \dot{\boldsymbol{p}}^s_{si} + \boldsymbol{K}_p\, \boldsymbol{e}_p \\ \boldsymbol{\omega}^s_{si} + \boldsymbol{K}_o\, \boldsymbol{e}_o \end{bmatrix} \tag{7.7}$$

where:

- $\boldsymbol{e}_p = \boldsymbol{d}(\boldsymbol{p}^s_{si} - \boldsymbol{p}^s_r, D_{max})$; with:

$$\boldsymbol{d}(\boldsymbol{x}, x_{max}) = \begin{cases} \frac{x_{max}}{\|\boldsymbol{x}\|}\boldsymbol{x}, & \text{if } \|\boldsymbol{x}\| > x_{max} \\ \boldsymbol{x}, & \text{otherwise} \end{cases}$$

- $\boldsymbol{e}_o = \eta_R\ \boldsymbol{\epsilon}_D - \eta_D\ \boldsymbol{\epsilon}_R - \boldsymbol{S}\,(\boldsymbol{\epsilon}_D)\,\boldsymbol{\epsilon}_R$; with η_i and $\boldsymbol{\epsilon}_i$ as the scalar and vector parts of the quaternion $Q_i = \{\eta_i, \boldsymbol{\epsilon}_i\}$ corresponding to the rotation matrix i, $R = \boldsymbol{R}^s_r$ and $D = \boldsymbol{R}^s_{si}$.

and $\boldsymbol{S}(\cdot)$ being the cross product matrix.

The third level of priority is dedicated to an independent movement of the platform, given by a desired separation in the XY dimensions between the end-effector and the center of the platform, $\boldsymbol{s}^p_r \in \mathbb{R}^2$:

$$\dot{\boldsymbol{x}}_3 = \boldsymbol{s}^p_r - \left(\boldsymbol{p}^s_{p_{XY}} - \boldsymbol{p}^s_{r_{XY}}\right) \tag{7.8}$$

with $\boldsymbol{s}^p_r$ being the current position of the center of the platform in the remote workspace. The values for the desired separation $\boldsymbol{s}^p_r$ can be seen in Fig. 7.2c.

Error computation

The results of the algorithm are presented below from an analysis of the values obtained during the trajectory seen in Figs. 7.2.

In particular, though, the metrics taken to evaluate the goodness of the trajectory tracking have been the errors of the trajectory, computed as the average of its norm. Thus, for task 2, the position and orientation errors, ε_{2p} and ε_{2o}, and the position error of task 3, ε_3, are computed from Eqs. 7.7 and 7.8 as

$$\varepsilon_{2p} = \frac{1}{T}\int_0^T \|\boldsymbol{e}_p\,(t)\,\|\;dt$$

$$\varepsilon_{2o} = \frac{1}{T}\int_0^T \|\boldsymbol{e}_o\,(t)\,\|\;dt$$

$$\varepsilon_{3} = \frac{1}{T}\int_0^T \|\dot{\boldsymbol{x}}_3\,(t)\,\|\;dt$$

with T the duration of the trajectory. Note that the units of ε_{2p} and ε_3 are meters, but ε_{2o} is unitless because it is computed from the quaternion components of the rotation matrices $\boldsymbol{R}_r^s$ and $\boldsymbol{R}_{si}^s$ in Eq. 7.7.

Results

In our example the parameters of the algorithm can be seen in Table 7.1.

K_1	β	K_p	D_{max}	K_o
2.0	0.1	1.0	0.5	1.0

Table 7.1 Parameters.

The duration of the trajectory has been $T = 42$ seconds.

The end-effector desired trajectory is quite complex, departing from the common practice of testing the proposed algorithms using simple cases. This common practice is useful in highlighting certain aspects of the algorithms, but may fail to expose more complex and unexpected behaviors which may ultimately lead to undesired motions and even hazardous situations.

Fig. 7.2d depicts the values of the joint limits during the trajectory, with two dashed horizontal lines at -0.9 and 0.9 representing the boundaries for the activation of the joint limit avoidance task. For instance, it can be seen that joint 2 intermittently reaches the lower joint limit avoidance task activation during roughly the first 10 seconds, and that the joint limit avoidance task are almost permanently activated for the joints 6 and 7 during the last 10 seconds of the experiment.

Fig. 7.2e shows the number of joints that have its joint limit avoidance task activated. This figure aids in the interpretation of Fig. 7.2d and eases the visualization of the algorithm behavior.

It is also important to study the singular values of $\boldsymbol{J}_2$ to check how far is the BMM-I from a singularity. Fig. 7.2f shows the lowest singular value of $\boldsymbol{J}_2$, $\sigma_6(\boldsymbol{J}_2)$. It can be seen that the lowest value of $\sigma_6(\boldsymbol{J}_2)$ is approximately 0.2 around the second 28. Thus, this is the moment during the teleoperation experiment where the robot is closest to a singularity.

Finally, Fig. 7.2g shows the norm of the position error of the end-effector tracking. Three big error peaks can be noticed, which are obviously unexpected and undesirable. Similarly, Fig. 7.2h shows the norm of the vector part of the orientation error, which should be 0 if the tracking was perfect. The cumulated error in position and orientation in this example is $\varepsilon_{2p} = 0.275$ m and $\varepsilon_{2o} = 0.205$ respectively, with the maximum position and orientation error peaks being 1 m and 1, respectively. Also, the cumulated error in the third task is $\varepsilon_3 = 0.467$ m.

It is necessary, though, to deepen into this example in order to better analyze the behavior of the inner continuous inverse algorithm. To this end, Figs. 7.3 show several results.

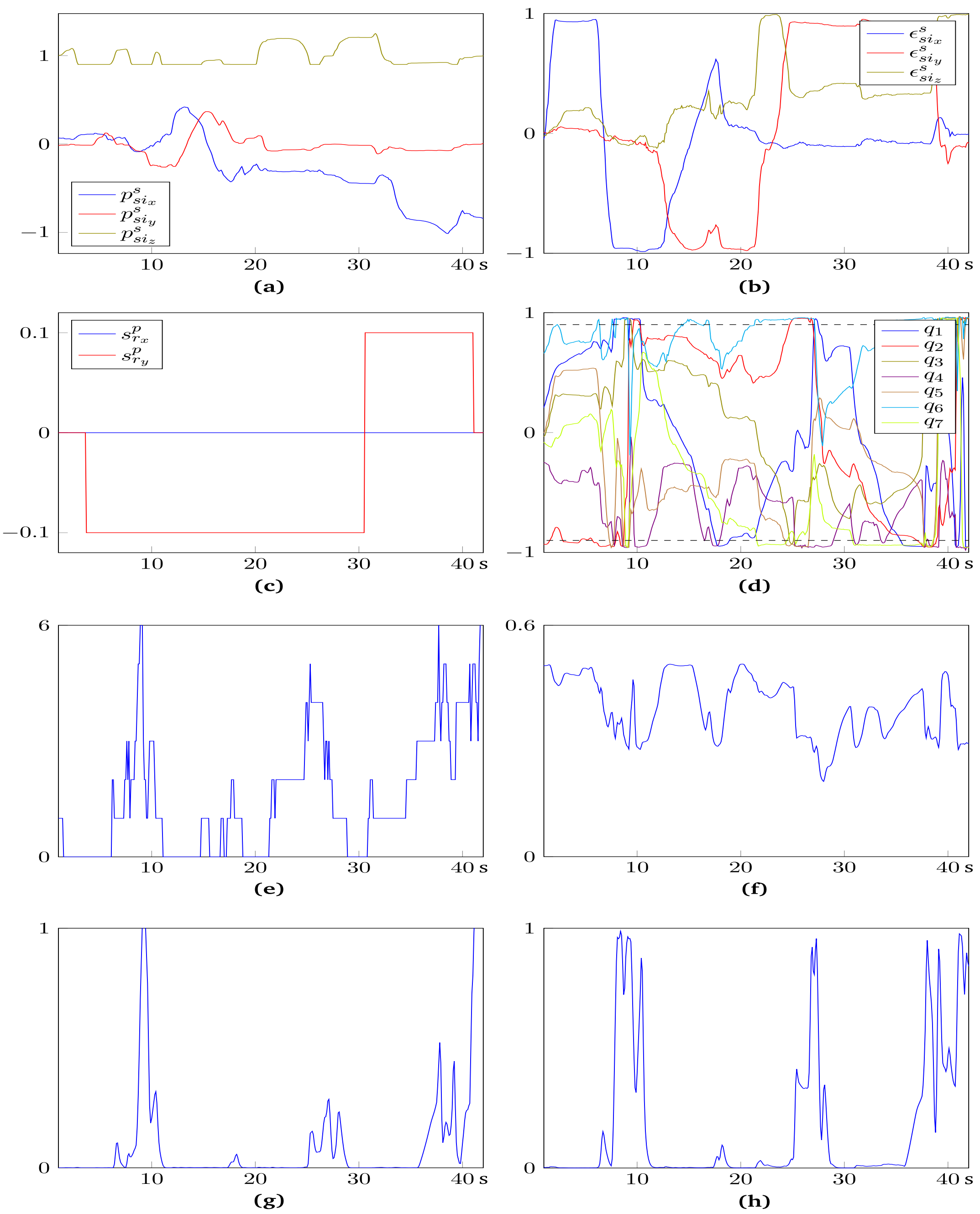
For instance, Fig. 7.3a and Fig. 7.3c show the number of partial Jacobians created in the computation of $\boldsymbol{J}_2^{\boldsymbol{N}_{\oplus}^1 \oplus}$ and $\boldsymbol{J}_3^{\boldsymbol{N}_{\oplus}^2 \oplus}$, respectively, that ultimately lead to the computation of the same number of pseudoinverses.

There is an obvious relation between the number of partial Jacobians created and the error peaks in Fig. 7.2g. This highlights the fact that the error increases with the number of activated tasks at the priority level 1. As the active task number increases, the number of partial Jacobians increases too, and so does the likelihood of being in the vicinity of a singularity.

Fig. 7.3b shows the lowest singular value from the set of partial Jacobians computed during the computation of $\boldsymbol{J}_2^{\boldsymbol{N}_{\oplus}^1 \oplus}$. Three major valleys can be seen that happen at the same time that the error peaks occur which ultimately correspond to algorithmic singularities.

With respect to the topic presented in Subsec. 7.2.1, in Fig. 7.3d it can be visually verified that the singular values of $\boldsymbol{N}_{\oplus}^2$ do not lie in the unit interval. Sometimes, not only one, but several singular values are above 1. It is also important to note that when this occurs the position errors are higher.

It is interesting to highlight that if the bound imposed to the error $\boldsymbol{e}_p$ as in (Buss and Kim, 2004) is not implemented, the unboundedness of the singular values of $\boldsymbol{N}_{\oplus}^1$ leads the robot to extreme instabilities, totally unacceptable for our needs (Figs. 7.4). The robot joint variables are shown in Fig. 7.4a to illustrate the unstable behavior. After the second 8 the system becomes clearly unstable and the position error starts to grow without bound, with a maximum peak at 40000 m (Fig. 7.4b). Finally, Fig. 7.4c shows the singular values of the null space projector at the second level of priority. Clearly, the largest singular value gets over 1 when the instabilities begin, roughly averaging a value of 2 and almost peaking at 3.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 7.2 BMM teleoperation performance with three levels of priority: joint limit avoidance (highest), TCP command by position and orientation, and platform desired velocity. From top to bottom, and right to left: a) TCP desired position, $\boldsymbol{p}^s_{si}$, in m; b) vector part of the TCP desired orientation quaternion, $\boldsymbol{\epsilon}^s_{si}$; c) desired separation between the platform and the TCP in the XY plane, $\boldsymbol{s}^p_r$, in m; d) normalized joint configuration, $\boldsymbol{q}$; e) number of active joint limit avoidance tasks; f) lowest singular value of $\boldsymbol{J}_2$; g) TCP position error, ε_{1p}, in m; h) orientation error, ε_{1o}.

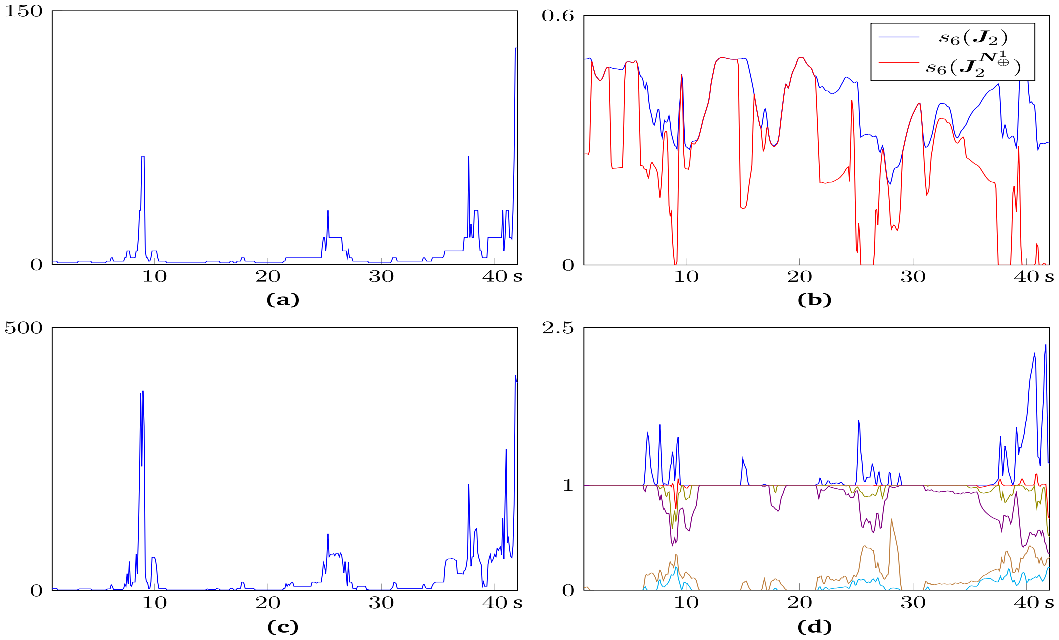


Figure 7.3 BMM teleoperation performance with three levels of priority: joint limit avoidance (highest), TCP command by position and orientation, and platform desired velocity. From top to bottom, and right to left: a) number of partial Jacobians in the priority level 2; b) lowest singular values of $\boldsymbol{J}_2$ and of the partial Jacobians at priority level 2; c) number of partial Jacobians in the priority level 3; d) singular values of $\boldsymbol{N}_\oplus^2$.

7.3 Enhancements

7.3.1 On the boundaries of the null space matrix

To keep the singular values of the projection matrix, $\boldsymbol{N}_\oplus^i$, inside the unit interval, a variation of the continuous inverse solution has been proposed and implemented in our teleoperation system.

Our proposed solution comes from the study of the eigenvalues of the partial pseudoinverses in Eq. 3.4 when this expression is used to compute the right continuous inverse (Eq. 3.7). This sheds light on the mechanism underlying the counter-example shown above.

It is interesting, though, to deepen in the continuous inverse and find out why its null space projector is not bounded to the unit circle by design.

At the end of Appendix E it can be seen that the singular values of $\boldsymbol{N}_\oplus^1$ are equal or smaller than 1 because the singular values of the partial projection operators $(\boldsymbol{H}_{P_1}\boldsymbol{J}_1)^+\boldsymbol{J}_1$ are also equal or smaller than 1.

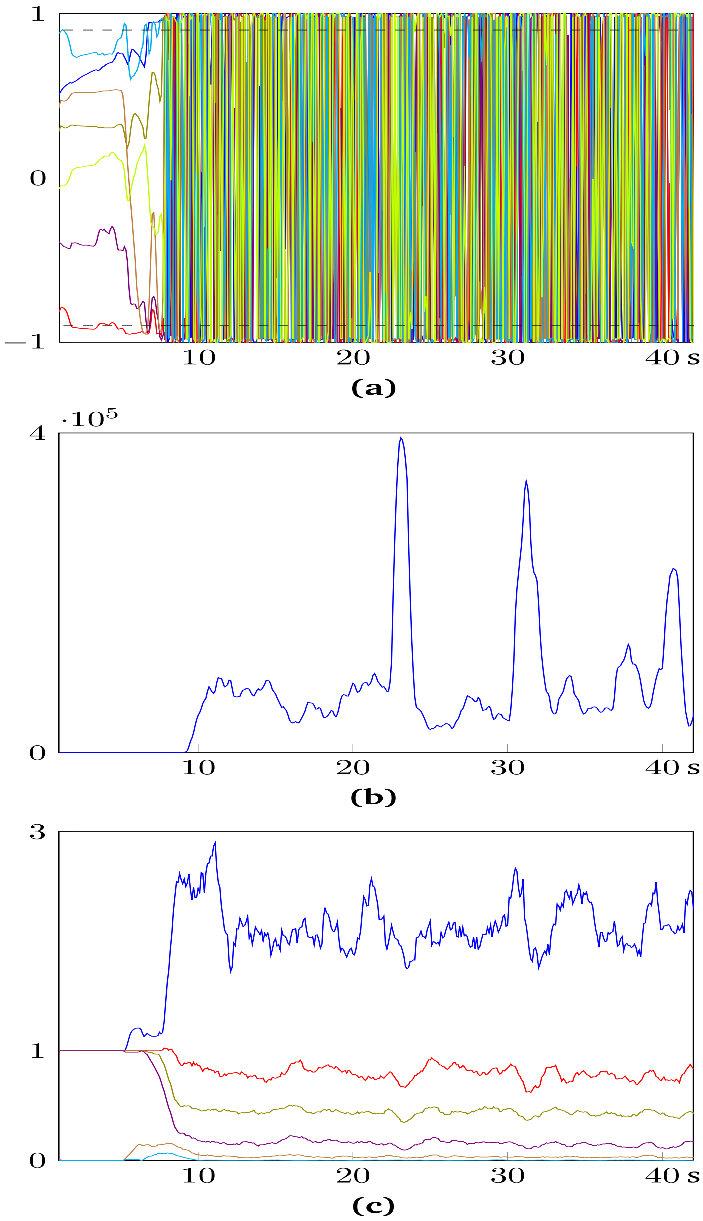


Figure 7.4 Unstable motions due to the null space projector of the lower levels not bounded to 1. From top to bottom: a) normalized joint configurations, $\boldsymbol{q}$; b) TCP position error, ε_{1p}, in m; c) singular values of $\boldsymbol{N}_{\oplus}^{2}$.

This behavior does not extend to lower levels of priority, that is, the largest singular value of $\boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\boldsymbol{J}_i$ may not be, in general, lower than 1 (as has been shown in Subsec. 7.2.1 with a simple example). Concretely, in Appendix E an upper bound for the largest singular value of $\boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\boldsymbol{J}_i$ is shown to be

$$\sigma_1\left(\boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\boldsymbol{J}_i\right) \leq \sum_{P\in B(m)} \left(\prod_{i\in P}\sigma_i\right)\prod_{i\notin P}(1-\sigma_i)\sqrt{1+\sigma_1^2\left(\left(\boldsymbol{J}_i\boldsymbol{U}_{i-1}\boldsymbol{H}_{P_i}\right)^+\boldsymbol{J}_i\boldsymbol{U}_{i-1}\right)} \tag{7.9}$$

with σ_p as be the largest singular value of the p-th addend of the sum obtained after developing the term $\boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\boldsymbol{J}_i$ in Eq. 7.2 using Eq. 3.8 and Eq. 3.7.

Even though the bound in Eq. 7.9 does not show in which cases the largest singular value will have a value greater than 1, it is consistent with this idea. Actually, it follows from the expression that the largest singular value of $\left(\boldsymbol{J}_i\boldsymbol{U}_{i-1}\boldsymbol{H}_{P_i}\right)^+\boldsymbol{J}_i\boldsymbol{U}_{i-1}$ can be guaranteed to be 1 when the task $i \in P$ in transition is decoupled from the rest of the tasks. In that case, the sum in Eq. 7.9 would be one (Mansard et al., 2009a). This will not happen in general, though.

The chosen solution to address this issue is to simply scale $\boldsymbol{N}_\oplus^i$ by its maximum value in case it is bigger than 1. Other solutions have been tested, like saturating the singular values of $\boldsymbol{N}_\oplus^i$ to 1, and scaling or saturating the particular partial terms $\boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\boldsymbol{J}_i$ in Eq. 7.2, but the best results are achieved by the simple proposed scaling.

Thus, the new projection matrix, $\boldsymbol{N}_\oplus^{*\,i}$, can be formalized as follows:

$$\boldsymbol{N}_\oplus^{*\,i} = \begin{cases} \boldsymbol{N}_\oplus^1, & i = 1 \\ \frac{1}{\sigma_1\left(\boldsymbol{N}_{t\,\oplus}^i\right)}\left(\boldsymbol{N}_{t\,\oplus}^i\right), & i > 1 \text{ and } \sigma_1\left(\boldsymbol{N}_\oplus^i\right) > 1 \\ \boldsymbol{N}_{t\,\oplus}^i, & \text{otherwise} \end{cases}$$

with $\boldsymbol{N}_{t\,\oplus}^i = \boldsymbol{N}_\oplus^{*\,i-1} - \boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{*\,i-1}\oplus}\boldsymbol{J}_i$.

The results of this correction on the projection operator of the right continuous inverse, $\boldsymbol{N}_\oplus^2$, can be seen in Figs. 7.6. Concretely, Fig. 7.6g shows the values of the singular values of $\boldsymbol{N}_\oplus^2$ and, as imposed, they are bounded to 1.

The improvement in the error performance of the algorithm is obvious, specially regarding the position (Fig. 7.6c). To put it in numbers, the cumulated position error during the teleoperation is $\varepsilon_{2p} = 0.119$ m, and the cumulated norm of the vector part of the orientation error quaternion (Fig. 7.6d), $\varepsilon_{2o} = 0.228$. The maximum error peaks in position and orientation are 0.8 m and 1, approximately, and the cumulated error in task 3 is $\varepsilon_3 = 0.425$ m.

7.3.2 On the lower priority levels

The second drawback of the continuous inverse is the fact that during the transitions the lower priority tasks can affect the higher priority tasks, which is undesirable.

There is a simple way to deal with this issue, which can be further tuned in a straightforward manner: given a certain level of priority, the key idea is to weight the contribution of the posterior levels of priority by an adaptive coefficient, k_L^i, which takes into account the impact that the latter will have in the former.

With this approach the incidence of the posterior levels of priority can be shut down by setting this variable to 0 or otherwise turned on by setting it to 1. In other words, during the transitions, the contribution of the levels of priority that negatively affect the higher ones is diminished or even discarded.

To implement this, the index of the priority level which do not accomplish their task are needed, starting from lowest to highest. If it is found that this happens at a level of priority i, the lower levels, $i+1, i+2, \ldots$, are deactivated and its contribution removed by setting $k_L^i = 0$.

This concept can be formulated to compute $\dot{\boldsymbol{q}} = \sum_{i=1}^{N} \dot{\boldsymbol{q}}_i$ for N levels of priority as in Eq. 7.3:

$$\dot{\boldsymbol{q}}_i = \begin{cases} \boldsymbol{J}_i^{\oplus \boldsymbol{H}} \dot{\boldsymbol{x}}_i, & i = 1 \\ k_L^{i-1}\, \boldsymbol{J}_i^{\boldsymbol{N}_{\oplus}^{i-1} \oplus} \left(\dot{\boldsymbol{x}}_i - \boldsymbol{J}_i \boldsymbol{J}_{i-1}^{\oplus \boldsymbol{H}} \dot{\boldsymbol{x}}_{i-1} \right), & \text{otherwise} \end{cases}$$

where, in general $k_L^{i-1} \in [0, 1]$ such that

$$k_L^i = \begin{cases} 1, & e_i < e_{min}^i \\ 0, & e_i > e_{max}^i \\ f_L^i(e_i, e_{min}^i, e_{max}^i, \frac{e_{max}^i - e_{min}^i}{2}), & \text{otherwise} \end{cases} \tag{7.10}$$

with a continuous decreasing function, $f_L^i(e_i)$, that depends on some error, e_i, of the current task.

In particular, in order to move k_L^i from 0 to 1 in a continuous way it can be used a function f_L^i similar to that of Eq. 7.5:

$$f_L^i(x, x_{min}, x_{max}, A) = \frac{1}{2}\left(1 - \tanh\left(\frac{A}{x_{max} - x} - \frac{A}{x - x_{min}}\right)\right) \tag{7.11}$$

where A is a constant that represents the slope of the function at the point $x = {}^{(x_{max}+x_{min})}\!/_{2}$ (Fig. 7.5). This function generalizes Eq. 7.6 by allowing to fully define the transition domain, not only by one of its ends through the value β.

Figs. 7.7 show the results after the implementation of the proposed solution in the example presented in Subsec.7.3.1. The values of the parameters x_{min} and x_{max} have been set to 0.001 and 0.002, respectively.

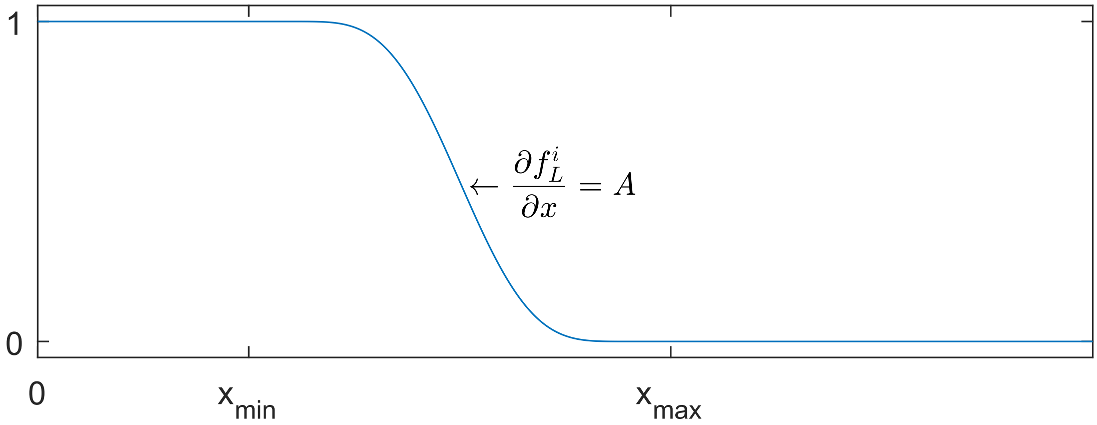


Figure 7.5 Task error smooth deactivation function, f_L^i.

There is an obvious decrease both in position and orientation (Fig. 7.7c) and Fig. 7.7d), respectively).

The maximum norm position error peak is now approximately 0.2 m, and 0.2 for the orientation. Interestingly, the number of partial pseudoinverses created has decreased significantly, with 25 pseudoinverses created instead of 60 for the second level of priority, and roughly 170 instead of 350 for the third.

Regarding the improvement of the cumulated position error during the teleoperation, it is $\varepsilon_{2p} = 0.006$ m, and the orientation error, $\varepsilon_{2o} = 0.007$, which is a 97.98 % and 96.66 % less error in position and orientation, respectively, compared to the initial example in Subsec.7.2.3.

The cumulated error in task 3 now is $\varepsilon_3 = 0.398$ m. The error has also diminished a 14.78 % from the initial 0.467 m in the example in Subsec. 7.2.3. There is still some error, though, but this is consistent with the implemented task priority hierarchy and the third task being the task with lower priority. It is also consistent with the fact that in many moments of the trajectory there is one or no DOFs left to accomplish the third task, which needs two DOFs to be fully executed.

7.4 Chapter Contributions

A flaw in the continuous inverse is presented that can lead to instabilities when dealing with more than two levels of priority. A modification of the continuous inverse operator is shown to prevent the appearance of these instabilities.

A second problem of the continuous inverse is considered: that low priority tasks affect higher priority ones during the activation and deactivation of tasks, which can lead to poor performance during long transitions. An enhancement is presented that ensures the task hierarchy.

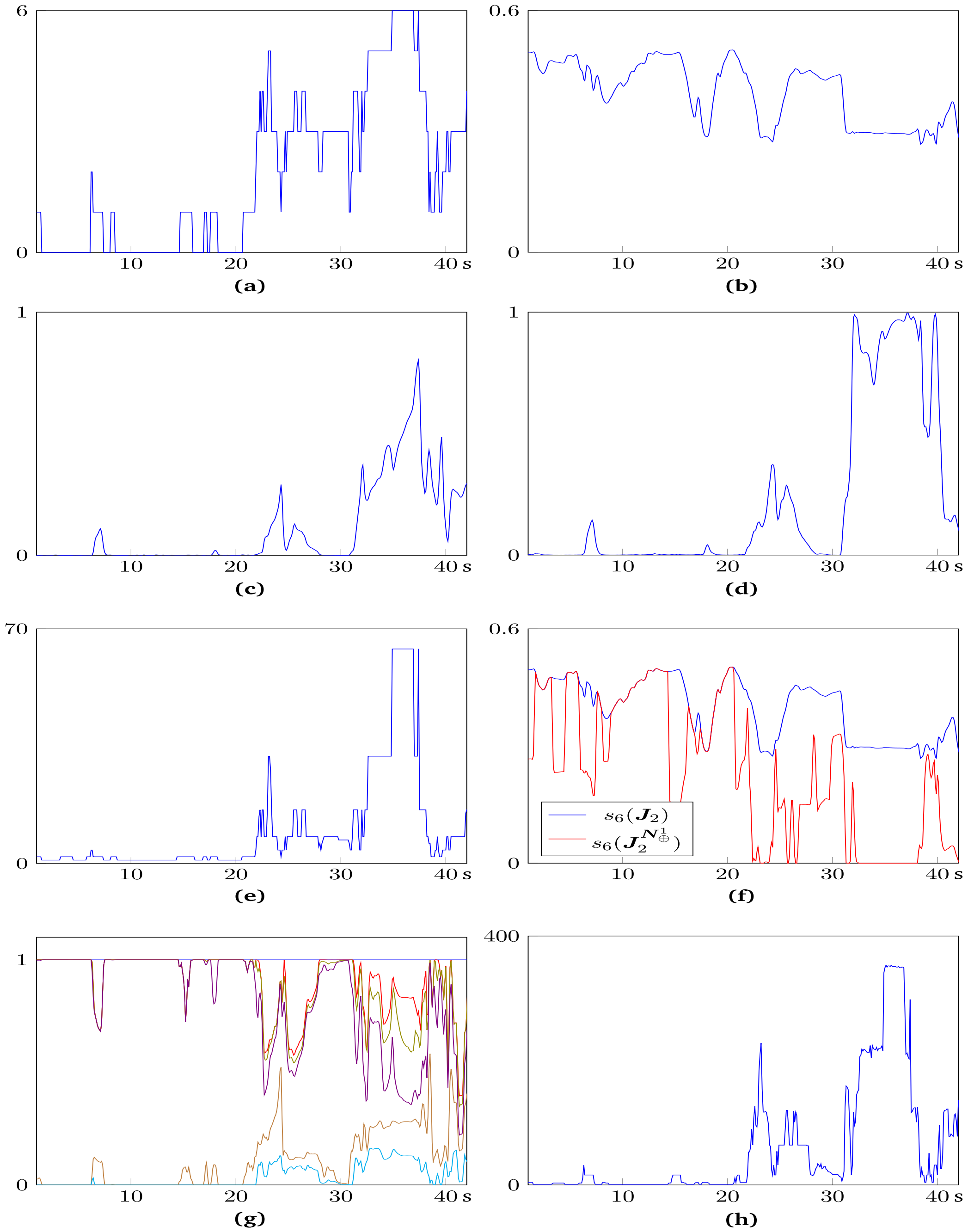


Figure 7.6 BMM teleoperation performance with $\boldsymbol{N}_{\oplus}^{2}$ scaled to bound its maximum singular value to 1. From top to bottom, and right to left: a) number of active joint limit avoidance tasks; b) lowest singular value of $\boldsymbol{J}_2$, $\sigma_6(\boldsymbol{J}_2)$; c) TCP position error, ε_{1p}, in m; d) orientation error, ε_{1o}; e) number of partial Jacobian in the priority level 2; f) lowest singular values of $\boldsymbol{J}_2$ and of the partial Jacobians at priority level 2; g) singular values of $\boldsymbol{N}_{\oplus}^{2}$; h) number of partial Jacobian in the priority level 3.

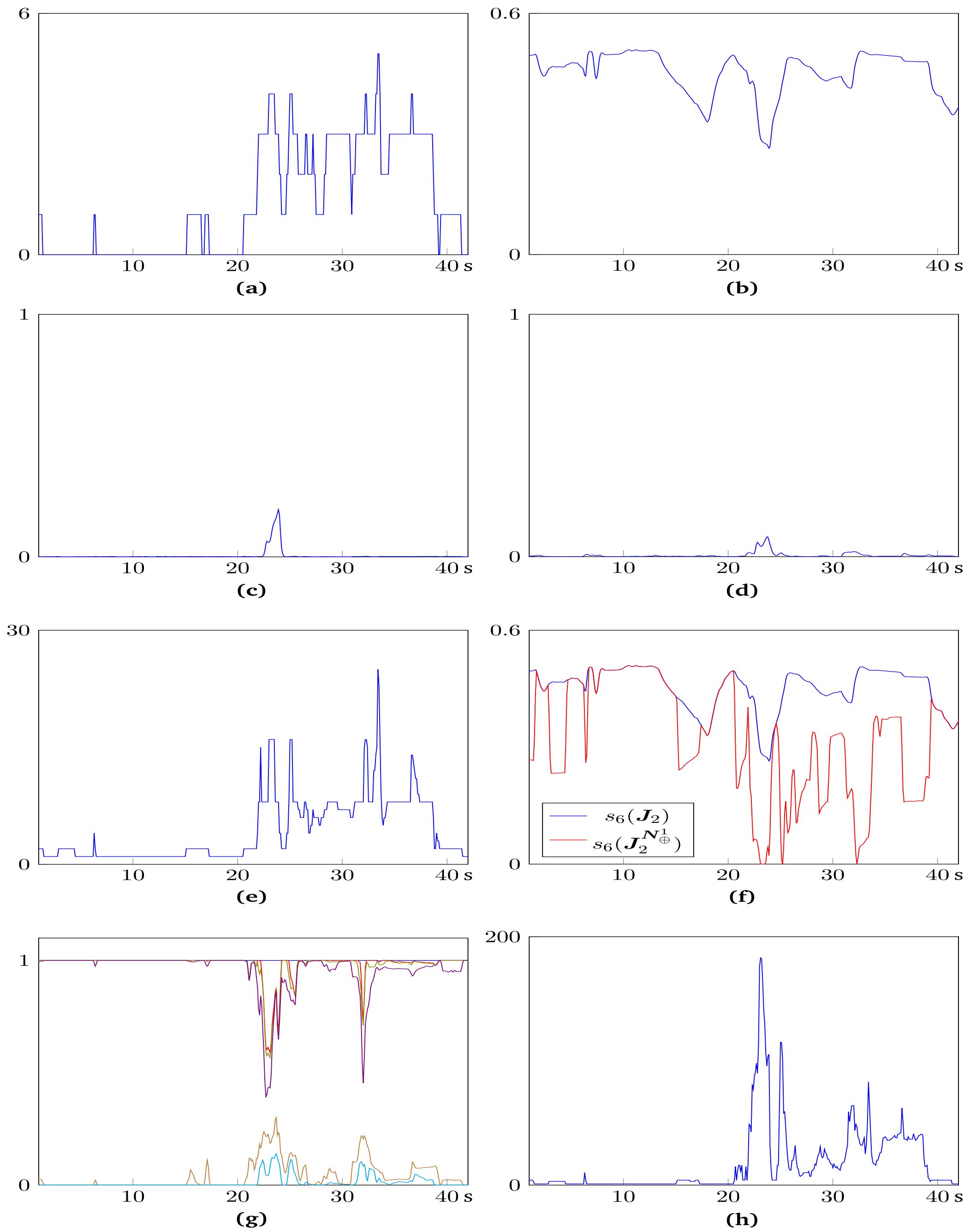


Figure 7.7 BMM teleoperation performance fulfilling the task priority hierarchy. From top to bottom, and right to left: a) number of active joint limit avoidance tasks; b) lowest singular value of $\boldsymbol{J}_2$, $\sigma_6(\boldsymbol{J}_2)$; c) TCP position error, ε_{1p}, in m; d) orientation error, ε_{1o}; e) number of partial Jacobian in the priority level 2; f) lowest singular values of $\boldsymbol{J}_2$ and of the partial Jacobians at priority level 2; g) singular values of $\boldsymbol{N}_\oplus^2$; h) number of partial Jacobian in the priority level 3.

Chapter 8

The Object Best View Function

It's the job that's never started as takes longest to finish.

The Lord of the Rings

SUMMARY: This chapter presents a method that exploits the null space of a mobile manipulator to increase the visibility of the end-effector, and thus the commanded object, by preventing the end-effector to be occluded by the body of the robot.

8.1 Introduction

One aspect of teleoperation often ignored, but which can greatly affect the teleoperation of a mobile manipulator, is the occlusion of the object at the end-effector of the robot that is being commanded by the operator.

This issue, central to the teleoperation field, can occur mainly due to two reasons. One reason is the occlusion of the commanded object by other objects of the scene. This can greatly impact the performance of a task, even leading to collisions and increasing of the workload on the operator, both physical and mental. While this is an important aspect, a solution to this problem needs to consider the full scene, which is out of the scope of this work.

The other cause that can lead to occlusions of the mobile manipulator end-effector is the robot itself, that is, its links and the platform. The robot may be in a configuration such that one or several parts of the robot are located between the camera and the end-effector, making the visibility of the commanded object more difficult.

Following the idea to free the operator from the explicit teleoperation of the robot and the free-flying camera in order to focus on the transported object, this part of the dissertation addresses the latter problem. By doing so, it is expected that situations where the robot gets between the object and the free-flying camera line of sight will be minimized, thus allowing the operator to have a better view of the object, and not disturb its focus on it.

The method that addresses the problem extends the capabilities of the teleoperation system presented in this dissertation, and, in particular, has been implemented on the Barcelona Mobile Manipulator I (BMM-I) considering that the operator is watching the scene from the free-flying camera. This method, called, from now on, the *Object Best View Function* (OBVF), or *object best view task*, or simply the *object best view*, is presented. This method prevents the object attached to the end-effector of the BMM-I to get occluded by its own platform and links. Results presented at the end of this part show the goodness of the proposed solution.

8.2 Proposed Solution

One of the leading aspects to address the aforementioned problem is its similarity with the obstacle avoidance problem.

In the classical obstacle avoidance problem, the goal is to move the robot from an initial to a final configuration in a scenario filled with objects with which the robot can collide. The challenge is to find a feasible trajectory or direction of movement that guarantees that the path or the next step is free of collisions.

The force field formulation (Khatib, 1986) assigns to each object a potential field that generates a repulsive force between the object and the robot. Given a scenario with multiple objects, the force field formulation allows the computation of a single force to be applied to the robot. This force is the result of the influence of all the potential objects in the scene. This force can be input to the robot through its null space, such that the avoidance task can be executed with a lower priority.

The main difference between the classical obstacle avoidance problem and the object best view task is that, while in the former the resultant repulsive force is applied to the robot, in the latter the repulsive force generated by each obstacle (the parts of the robot, that is, the platform or its links) has to actually be applied to the link itself or part of the robot which is occluding the end-effector.

In the OBVF, the point on the robot that is intended to stay away from the obstacles is the robot end-effector. But as the main task is to set a trajectory for the end-effector, it is not suitable to apply the resultant force to it, as there are no degrees of freedom left for more constraints. The only available choice is to apply the force to the robot itself. In particular, the force should be applied to the closest link, which is the link more likely to occlude the end-effector. This would actually make that robot link move away from the end-effector.

If the force is applied to the closest link to the end-effector, this implies that the application point of the force may change if, for example, another link gets closer to the end-effector at some other configuration. This would generate a discontinuity, which may lead to instabilities in the robot trajectory, making the algorithm infeasible.

Taking into consideration the issues exposed above, the proposed solution follows the next ideas:

- To make use of the null space of the mobile manipulator to avoid robot self-occlusions of the mobile manipulator as a secondary task.
- To pose the problem as a collision avoidance task between the robot end-effector and particular parts of the robot within an adapted force field formulation.
- To evaluate the proximity between the end-effector and the links of the robot *projected in the image plane*. By measuring the distance in the image plane between the end-effector projected in that plane and the projected links of the robot, the point of view of the operator is naturally taken into account.
- To implement the proposed solution in the BMM-I.

To compute the desired configuration velocity, and to input it to the robot null space as a secondary task, it is necessary to define a function that carries the information of the object and the camera, and that outputs the robot motion required to maximize some object view quality measure. To do so, the OBVF can be implemented using the force field formulation, in a similar way as a joint limit avoidance task.

With the force field formulation, an *escape* velocity (*i.e.*, the repulsive force in the force field formulation) in the Cartesian space is generated to further be applied to the point of the BMM-I where the axis of joints 3 and 4 intersect. The desired velocity input is designed to contain the information of the distance of all the links that can occlude the robot end-effector, and to give more importance to the closest links because those are the links more likely to occlude the end-effector.

Concretely, consider the points $\boldsymbol{p}_0$, $\boldsymbol{p}_1$, $\boldsymbol{p}_2$ and $\boldsymbol{p}_r$ of the mobile manipulator, expressed in the world reference, s, such that each point $\boldsymbol{p}_i = \begin{bmatrix} p_{i_x} & p_{i_y} & p_{i_z} \end{bmatrix}^T \in \mathbb{R}^3$, and the homogeneous transformation matrix $\boldsymbol{T}_s^c \in SE(3)$ from the world frame to the camera frame, c (Fig. 8.1). Let $\boldsymbol{a}_c$ denote the unit vector over the optical axis of the camera pointing towards the robot. Note that $\boldsymbol{a}_c$ also corresponds to the Z axis of the rotation matrix in $\boldsymbol{T}_s^c$, that is, $\boldsymbol{a}_c = \boldsymbol{T}_s^c \boldsymbol{e}_Z$ with $\boldsymbol{e}_Z = [0 \quad 0 \quad 1 \quad 0]^T$.

The first step is to decide when the OBVF needs to be activated. The OBVF is going to be activated when any of the points $\boldsymbol{p}_1$ or $\boldsymbol{p}_2$, without considering its height (Z coordinate), is in

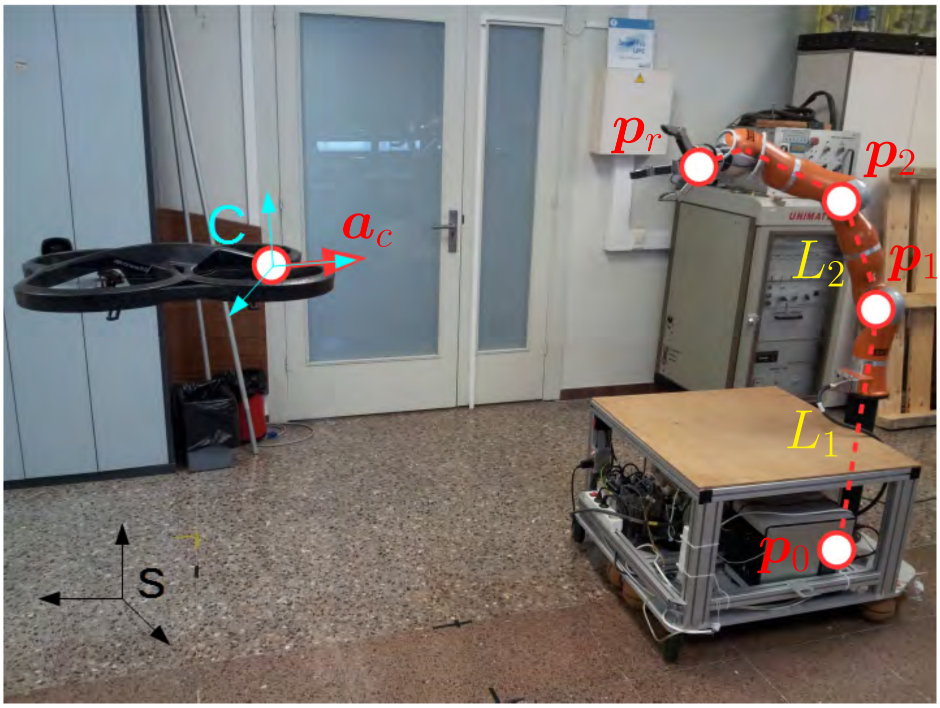


Figure 8.1 Elements of interest regarding the object best view task.

front of the point $\boldsymbol{p}_r$ as seen from the camera position, that is, from the origin of the c coordinate system.

Let $\boldsymbol{p}_{i_{XY}} = \begin{bmatrix} p_{i_x} & p_{i_y} \end{bmatrix}^T \in \mathbb{R}^2$ and $\boldsymbol{a}_{c_{XY}} = \begin{bmatrix} a_{c_x} & a_{c_y} \end{bmatrix}^T \in \mathbb{R}^2$ be the projections of the points $\boldsymbol{p}_i$ and $\boldsymbol{a}_c$ in the XY plane, then the point $\boldsymbol{p}_i$ will be in front of $\boldsymbol{p}_r$ if the next condition is satisfied

$$s_i = (\boldsymbol{p}_{i_{XY}} - \boldsymbol{p}_{r_{XY}})^T \boldsymbol{a}_{c_{XY}} > 0. \tag{8.1}$$

Thus, the OBVF is activated if any of the two s_1 or s_2 are strictly positive. Note that the activation and deactivation of the task can lead to discontinuities. This issue is addressed below.

In the case that the OBVF is activated, the next step, for convenience, is to change the points reference frame from the world to the camera, that is, $\boldsymbol{r}_i = (\boldsymbol{T}_s^c)^{-1} \boldsymbol{p}_i$ and $\boldsymbol{r}_r = (\boldsymbol{T}_s^c)^{-1} \boldsymbol{p}_r$.

Let's define the function $\mathrm{c2ph}(\boldsymbol{r}, f_c)$ that projects a point $\boldsymbol{r} = [r_x \quad r_y \quad r_z]^T \in \mathbb{R}^3$ in the camera frame to a point $\boldsymbol{c} \in \mathbb{R}^2$ in the image plane, considering a *pinhole camera model* (Section 3.4), as

$$\boldsymbol{c} = \mathrm{c2ph}(\boldsymbol{r}, f_c) = \frac{f_c}{r_z} \begin{bmatrix} r_x \\ r_y \end{bmatrix}$$

where f_c is the focal length of the camera.

In reverse order, let's also define the function $\mathrm{ph2c}(\boldsymbol{c}, z, f_c)$ that converts a projected point $\boldsymbol{c} = [c_x \quad c_y]^T \in \mathbb{R}^2$ in the image plane at the focal distance f_c into its three dimensional *unpro-*

jected point $\boldsymbol{r} \in \mathbb{R}^3$ at a distance z with

$$\boldsymbol{r} = \text{ph2c}(\boldsymbol{c}, z, f_c) = \frac{z}{f_c} \begin{bmatrix} c_x \\ c_y \\ 1 \end{bmatrix} .$$

The next step is to compute the force field exerted on each of the considered links of the mobile manipulator. The considered links can be seen in Fig. 8.1 as L_1 and L_2. L_1 is the link defined by the points $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$, while L_2 is the link defined by the points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$.

Our interest is to compute the point $\boldsymbol{r}_i^* \in \mathbb{R}^3$, in every link i, such that its projection in the image plane $\boldsymbol{c}_i^*$ is closest to the projection of the robot end-effector point also in the image plane, $\boldsymbol{c}_r$. Given $\lambda_i \in [0,1]$ any point $\boldsymbol{r}_{L_i}$ in the link L_i can be expressed as

$$\boldsymbol{r}_{L_i} = \boldsymbol{r}_{i-1} + \lambda_i \left(\boldsymbol{r}_i - \boldsymbol{r}_{i-1}\right) = \boldsymbol{r}_{i-1} + \lambda_i\, \boldsymbol{n}_i \tag{8.2}$$

with $i \in \{1, 2\}$.

This is equivalent to solve the next minimization problem:

$$\lambda_i^* = \arg\min_{\lambda_i} \|\text{c2ph}(\boldsymbol{r}_{i-1} + \lambda_i\, \boldsymbol{n}_i, f_c) - \text{c2ph}(\boldsymbol{r}_r, f_c)\| \tag{8.3}$$

in which $(\arg\min f)$ corresponds to the arguments of the minimum of the function $f(\cdot)$, that is, the points of the domain of $f(\cdot)$ at which the function values are minimized.

Eq. 8.3 is also equivalent to obtain the value λ_i^* with $i \in \{1, 2\}$ that minimizes the previous squared distance:

$$\lambda_i^* = \arg\min_{\lambda_i} \left(\left(\frac{r_{i-1_x} + \lambda_i\, n_{i_x}}{r_{i-1_z} + \lambda_i\, n_{i_z}} - \frac{r_{r_x}}{r_{r_z}} \right)^2 + \left(\frac{r_{i-1_y} + \lambda_i\, n_{i_y}}{r_{i-1_z} + \lambda_i\, n_{i_z}} - \frac{r_{r_y}}{r_{r_z}} \right)^2 \right) .$$

The solution of this minimization problem can be obtained analytically. The analytical solution can be found in Appendix F.

Note that the solutions λ_1^* and λ_2^* of the minimization problems must be in the interval $[0, 1]$. Otherwise points in the same line but outside the link would be taken into account, which is not desirable for our purposes.

Once λ_1^* and λ_2^* are obtained, the corresponding point in the link can be determined using Eq. 8.2:

$$\boldsymbol{r}_i^* = \boldsymbol{r}_{i-1} + \lambda_i^*\, \boldsymbol{n}_i. \tag{8.4}$$

Now, in order to determine the repulsive force $\boldsymbol{f}_i$ from the robot end-effector to each link both its modulus and its direction need to be computed.

The direction of the force can be expressed as the unit vector

$$\boldsymbol{u}_i = \frac{\boldsymbol{p}_i^* - \boldsymbol{p}_r}{\|\boldsymbol{p}_i^* - \boldsymbol{p}_r\|}$$

with $\boldsymbol{p}_i^* = \boldsymbol{T}_s^c\,\boldsymbol{r}_i^*$. Note that, although the force direction points from the end-effector to the point in the link, it will ultimately be the link which will be moved away from the robot end-effector.

The modulus of the force, k_i, has to be computed somewhat inversely proportional to the distance between the image of the end-effector and the link in the image plane, so as to give the teleoperator a sense of coherency between the exerted force and the proximity of the object occlusion. The distance between the end-effector and the link in the image plane is straightforward to compute:

$$d_i = \|\text{c2ph}(\boldsymbol{r}_i, f_c) - \text{c2ph}(\boldsymbol{r}_r, f_c)\|.$$

In order to compute the modulus of the force, a maximum value k_i^F of the force can be set when the distance is smaller than a certain value, d_i^m. For distances greater than d_i^M the modulus can be set to zero, *i.e.*, $k_i = 0$. The interval $[d_i^m, d_i^M]$ is selected such that a smoothing transition for the force modulus between $[0, k_i^F]$ can be implemented. k_i can be written similar to Eq. 7.6:

$$k_i = \begin{cases} 0 & \text{if } d_i > d_i^M \\ \frac{k_i^F}{2}\left(1 + \tanh\left(\frac{1}{2}\frac{d_i^M - d_i^m}{d_i^M - d_i} - \frac{1}{2}\frac{d_i^M - d_i^m}{d_i - d_i^m}\right)\right) & \text{if } d_i \in [d_i^m, d_i^M] \\ k_i^F & \text{else} \end{cases} \tag{8.5}$$

And finally the force for each link L_i can be computed as $\boldsymbol{f}_i = k_i\,\boldsymbol{u}_i$. Figure 8.2 graphically depicts the elements needed to compute each force $\boldsymbol{f}_i$.

The final force $\boldsymbol{f}$ which accounts for the two forces can be easily obtained as the sum of $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$. This ensures the continuity and smoothness in time of the force vector as both $\boldsymbol{f}_i$ are continuous and calculated using Eq. 8.5.

As stated above, the modulus of the force needs to change in a continuous way also with respect to the activation and deactivation of the OBVF. Otherwise, in a situation where the robot tip would be close to a link in the image plane, but still in front of it, if, by the command exerted by the user, the tip of the robot would get behind one of the links (making a $s_i > 0$ in Eq. 8.1) a force would suddenly appear and trigger a discontinuity in the joints.

To address this issue, a threshold s_F has been defined when using Eq. 8.1. This threshold sets up a transition zone where the force $\boldsymbol{f}$ continuously changes from zero to its appropriate value. This strategy can be implemented using a parameter k_F for determining $\boldsymbol{f} = k_F(\boldsymbol{f}_1 + \boldsymbol{f}_2)$, with

$$k_{F_i} = \begin{cases} 0 & \text{if } s_i > 0 \\ \frac{1}{2}\left(1 + \tanh\left(\frac{1}{2}\frac{s_F}{s_F - s_i} - \frac{1}{2}\frac{s_F}{s_i}\right)\right) & \text{if } s_i \in [-s_F, 0] \\ 1 & \text{else} \end{cases} \tag{8.6}$$

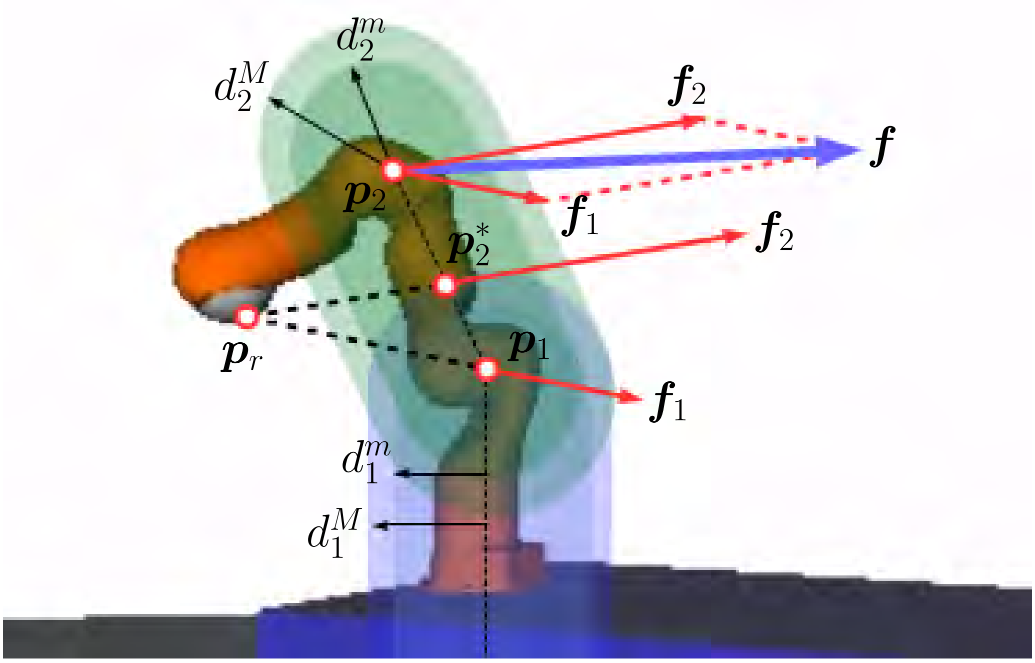


Figure 8.2 Exerted forces on the links by the proximity of the robot end-effector. Note that, in this particular case, the end-effector lies outside the area where the forces $\boldsymbol{f}_1$ or $\boldsymbol{f}_2$ are generated, which has been presented this way for ease of visualization.

and by finally setting $k_F = \max(k_{F_1}, k_{F_2})$.

The computed force $\boldsymbol{f}$ can now to be fed to the null space algorithm. To do so, it is first necessary to select the application point of the force. In order to solve the problem of having to switch between several points, which would produce undesirable discontinuities and ultimately lead to instabilities during the teleoperation task, the application point has been set to $\boldsymbol{p}_2$. Moreover, discontinuities on its derivative are also avoided with this choice.

Once the point to apply the force $\boldsymbol{f}$ has been chosen, the particular robot Jacobian in this point, $\boldsymbol{J}_{\boldsymbol{p}_2}$, can be computed.

It would be tempting to obtain the joint velocities to feed the Jacobian null space algorithm, by trying to find the appropriate $\dot{\boldsymbol{q}}$ that solves the task

$$\boldsymbol{f} = \boldsymbol{J}_{\boldsymbol{p}_2}\,\dot{\boldsymbol{q}}.$$

But note that, since this algorithm attempts to increase the distance between the robot end-effector and the links of the robot in the image plane, it doesn't matter what happens with the distance in the Cartesian space, provided that this distance, when projected in the image plane, ultimately increases. This means that it is not needed that the closest link of the robot from the end-effector follows the direction pointed by the force $\boldsymbol{f}$, which implies a three-DOFs task. It is only needed that it does so in the image plane, thus imposing a two-DOFs task.

For this reason, the Jacobian and the force need to be transformed into the camera reference frame. Next, it is necessary to only keep those directions in the camera frame which are of interest, the X and Y dimensions. Projection matrices can be defined for this purpose as

$$\boldsymbol{P}_{XY}^{J} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{P}_{XY}^{F} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and the joint velocities that execute the OBVF can be obtained by solving for $\dot{\boldsymbol{q}}$ from

$$\boldsymbol{f}_c = \boldsymbol{P}_{XY}^{F} \, (\boldsymbol{R}_s^c)^T \, \boldsymbol{f} = \boldsymbol{P}_{XY}^{J} \, \boldsymbol{M} \, \boldsymbol{J}_{\boldsymbol{p}_2} \, \dot{\boldsymbol{q}} = \boldsymbol{J}_c \, \dot{\boldsymbol{q}}$$

with $\boldsymbol{f}_c$ being the force $\boldsymbol{f}$ projected in the image plane; $\boldsymbol{J}_c$, the Jacobian of point $\boldsymbol{p}_2$ in the image plane; and with the matrix

$$\boldsymbol{M} = \begin{bmatrix} (\boldsymbol{R}_s^c)^T & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{R}_s^c)^T \end{bmatrix}$$

that rotates the velocity vectors of the Jacobian so that they are referenced in the camera frame.

Now the task can be fed as a lower priority task, for instance, to obtain the joint velocity $\dot{\boldsymbol{q}}$ that implements the OBVF as a secondary task with

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{+} \dot{\boldsymbol{x}}_1 + (\boldsymbol{J}_c \boldsymbol{N}_1)^{+} \left(\boldsymbol{f}_c - \boldsymbol{J}_c \, \boldsymbol{J}_1^{+} \dot{\boldsymbol{x}}_1 \right) .$$

with the task 1 being the tracking of a desired trajectory of the end-effector pose.

Finally, the whole procedure explained above can be summarized in Algorithm 2.

8.3 Simulation Results

A set of tests has been run in order to verify the proposed approach. The algorithm has initially been implemented in a simulation environment.

A three-level hierarchy of tasks has actually been implemented in the teleoperation robotic system both for the simulations and the real setup. The first and second levels of priority have been set as in Sect. 7.2.3: the first level of priority has been dedicated to the joint limit avoidance task, and the second, to the end-effector pose tracking of the user command in the remote scenario. The OBVF has been implemented in third level of priority. The continuous inverse has been used to account for the activation and deactivation of the joint limit avoidance task at the highest priority level.

The simulation performed shows the response of the proposed algorithm in three teleoperation scenarios in which the primary task is the command of the end-effector by the operator. In this scenarios the situations to verify the object best view task are:

Algorithm 2: The object best view function algorithm.

```
input  : $\boldsymbol{p}_0, \boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_r, \boldsymbol{T}_s^c$
output: $\boldsymbol{f}_c, \boldsymbol{J}_c$
get the optical axis $\boldsymbol{a}_c = \boldsymbol{T}_s^c\,\boldsymbol{e}_Z$
compute $s_1 = (\boldsymbol{p}_{1_{XY}} - \boldsymbol{p}_{r_{XY}})^T \boldsymbol{a}_{c_{XY}}$ and
        $s_2 = (\boldsymbol{p}_{2_{XY}} - \boldsymbol{p}_{r_{XY}})^T \boldsymbol{a}_{c_{XY}}$
if $s_1 > 0$ or $s_2 > 0$ then
    change the reference frame to the camera: $\boldsymbol{r}_i = (\boldsymbol{T}_s^c)^{-1}\,\boldsymbol{p}_i$ and $\boldsymbol{r}_r = (\boldsymbol{T}_s^c)^{-1}\,\boldsymbol{p}_r$
    compute the closest point on the links to the end-effector in the image:
        • $\lambda_i^* = \arg\min_{\lambda_i} \|\mathrm{c2ph}(\boldsymbol{r}_{i-1} + \lambda_i\,\boldsymbol{n}_i, f_c) - \mathrm{c2ph}(\boldsymbol{r}_r, f_c)\|$
        • $\boldsymbol{r}_i^* = \boldsymbol{r}_{i-1} + \lambda_i^*\,\boldsymbol{n}_i$
        • $\boldsymbol{p}_i^* = \boldsymbol{T}_s^c\,\boldsymbol{r}_i^*$
    compute the repulsion forces $\boldsymbol{f}_i$ with $i = 1, 2$:
        • $\boldsymbol{u}_i = \frac{\boldsymbol{p}_i^* - \boldsymbol{p}_r}{\|\boldsymbol{p}_i^* - \boldsymbol{p}_r\|}$
        • $d_i = \|\mathrm{c2ph}(\boldsymbol{r}_i, f_c) - \mathrm{c2ph}(\boldsymbol{r}_r, f_c)\|$
        • $k_i = \begin{cases} 0 & \text{if } d_i > d_i^M \\ \frac{k_i^F}{2}\left(1 + \tanh\left(\frac{1}{2}\frac{d_i^M - d_i^m}{d_i^M - d_i} - \frac{1}{2}\frac{d_i^M - d_i^m}{d_i - d_i^m}\right)\right) & \text{if } d_i \in [d_i^m, d_i^M] \\ k_i^F & \text{else} \end{cases}$
        • $\boldsymbol{f}_i = k_i\,\boldsymbol{u}_i$
    compute the total repulsion force:
        • for each $i = 1, 2$: $k_{F_i} = \begin{cases} 0 & \text{if } s_i > 0 \\ \frac{1}{2}\left(1 + \tanh\left(\frac{1}{2}\frac{s_F}{s_F - s_i} - \frac{1}{2}\frac{s_F}{s_i}\right)\right) & \text{if } s_i \in [-s_F, 0] \\ 1 & \text{else} \end{cases}$
        • $k_F = \max(k_{F_1}, k_{F_2})$
        • $\boldsymbol{f} = k_F(\boldsymbol{f}_1 + \boldsymbol{f}_2)$
    $\boldsymbol{f}_c = \boldsymbol{P}_{XY}^F\,(\boldsymbol{R}_s^c)^T\,\boldsymbol{f}$
    $\boldsymbol{J}_c = \boldsymbol{P}_{XY}^J\,\boldsymbol{M}\,\boldsymbol{J}_{\boldsymbol{p}_2}$
else
    $\boldsymbol{f}_c = \boldsymbol{0}$
    $\boldsymbol{J}_c = \boldsymbol{0}$
end
```

1. With manipulator of the the BMM-I standing in its vertical configuration, the operator moves the end-effector at the *back* of the robot, occluding the end-effector by the robot links.
2. The operator commands the robot end-effector in a straight motion from right to left and parallel to the camera plane, such that the robot end-effector gets behind the robot parts during the motion.
3. The BMM-I is set in a configuration such that its end-effector, as seen in the image plane, is located behind the links of the robot. Next, the OBVF is activated.

The values of the parameters tunned for this simulation can be seen Table 8.1.

f_c	d_1^m	d_1^M	k_{F_1}	d_2^m	d_2^M	k_{F_2}	s_F
1.0	0.10	0.15	1.5	0.05	0.10	1.0	0.1

Table 8.1 Simulation parameters.

Since the parameters d_1^m, d_1^M, d_2^m and d_2^M are distances, higher values would imply that the effect of the end-effector on the links would be exerted from further distances. On the other hand, small values would imply that the end-effector would approach too much the links before the latter would move away from it. The chosen values present a good balance between these two aspects.

The values of k_{F_1} and k_{F_2} have been chosen after experimentation so that the effect of the end-effector on the links would not be too abrupt.

The parameter $s_F = 0.1$ corresponds to an angle of roughly 84.2 degrees between the vectors $(\boldsymbol{p}_{i_{XY}} - \boldsymbol{p}_{r_{XY}})$ and $\boldsymbol{a}_{c_{XY}}$. That means that the activation / deactivation zone of the OBVF corresponds to almost 6 degrees (as it is activated between 90 to 84.2 degrees).

Simulation 1

The first simulation consists in moving the BMM-I from the vertical configuration to a configuration with the arm extended horizontally and such that the end-effector be hidden behind the links of the robot from the operator perspective.

The sequence of images in Fig. 8.3 shows the motion without the OBVF activated, and it can be clearly seen how the end-effector is occluded by the robot.

In Fig. 8.4 the same motion is shown but with the function activated. It can be clearly seen that now the BMM-I does a self-motion such that it prevents the occlusion to occur.

Finally, Fig. 8.5a shows the modulus of the force $\boldsymbol{f}$ exerted during the motion, and Fig. 8.5b, the minimum distances d_1 and d_2 when the OBVF is active and not. It can be seen that the modulus $\|\boldsymbol{f}\|$ increases when the distance between the robot end-effector and the links decreases,

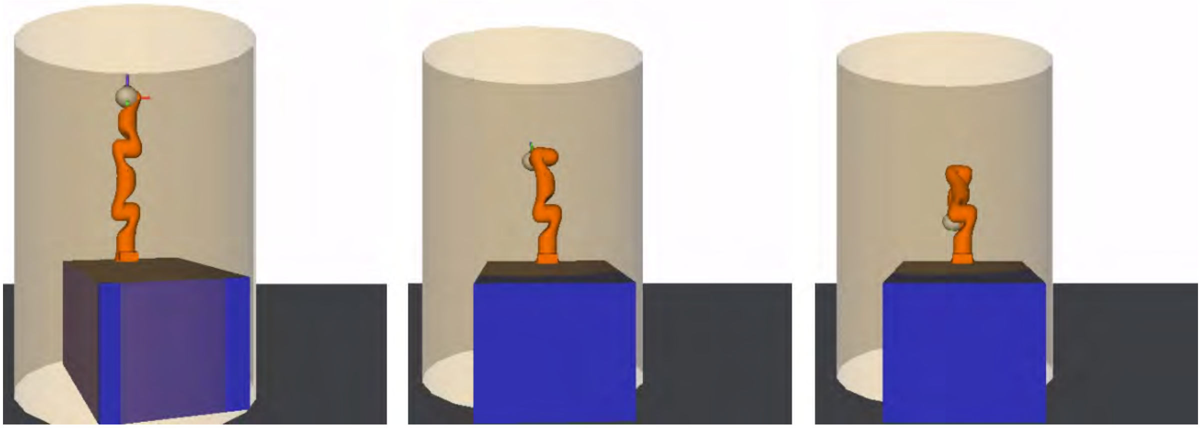


Figure 8.3 Object best view function simulation 1 without activation. From left to right, the motion from the vertical to the somehow horizontal BMM-I configuration.

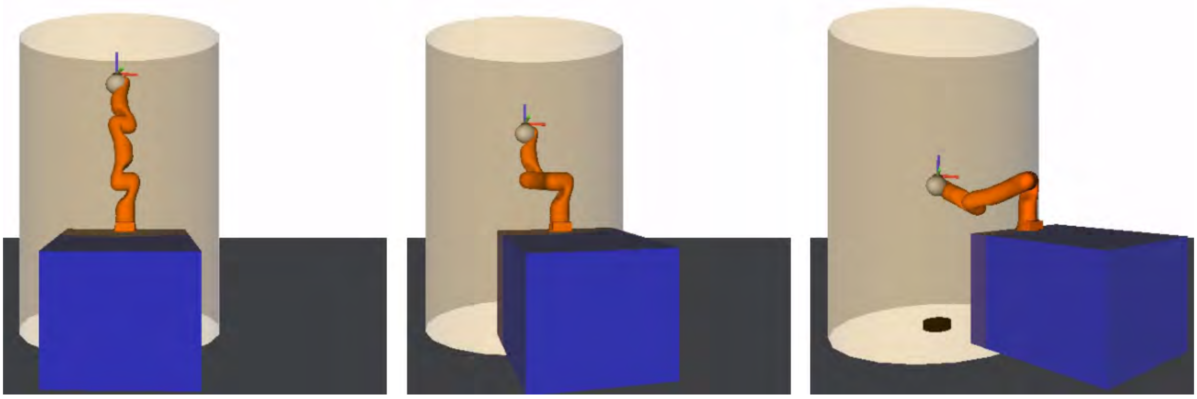


Figure 8.4 Object best view function simulation 1 with activation. From left to right, the motion from the vertical to the horizontal BMM-I configuration with the OBVF active.

peaking at the lowest distance (blue line), and then stabilizing. It can also be appreciated how the distance remains far from zero when the OBVF is active (blue line in the right image), while when the task is not active the distance between the robot end-effector and the robot links becomes almost zero.

A video with this type of motion can be found on the link *Simulation 1*.

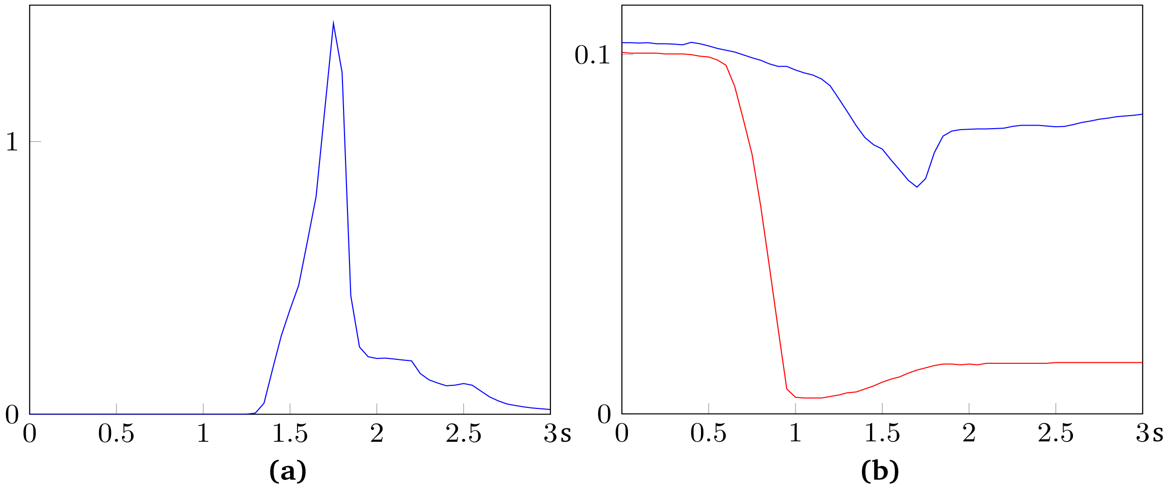


Figure 8.5 Object best view function simulation 1: a) modulus of the force $\boldsymbol{f}$ when the OBVF is active; b) minimum distances between d_1 and d_2, *i.e.*, $\min(d_1, d_2)$, with the OBVF active (blue) and not active (red).

Simulation 2

In the second simulation, the operator commands the robot end-effector to move from right to left parallel to the image plane and such that, during the first part of the motion, the end-effector is at the back of the platform with respect to the camera. This required motion is solved by the inverse kinematics by first executing a general rotation motion around the Z axis of the platform in order to let the robot end-effector locate at the left side of the platform, as if it was to pull the platform towards the desired motion to the left side of the scenario.

The particular motion arranged in the scenario of this simulation has been set such that the rotation moves the end-effector behind the robot becoming occluded by the robot. To prevent the mentioned occlusion, the object best view approach has been activated.

The sequences of images in Fig. 8.6 and Fig. 8.7 show the motion of this second simulation without and with the OBVF activated, respectively. When the object best view task is not active an occlusion occurs during the teleoperation (third image in Fig. 8.6), which does not happen when the task is active.

Fig. 8.8b shows the minimum distances d_1 and d_2 when the OBVF is active for the current simulation. From this image, it is clear that the minimum distance becomes zero when the function is not active (corresponding to the third image in Fig. 8.6), which is undesirable, since it prevents the operator from clearly seeing the object. When the function is active, the distance decreases until the force exerted by the OBVF is strong enough to prevent the robot end-effector to pass behind the robot links. Fig. 8.8a shows the force $\boldsymbol{f}$ exerted by the robot end-effector against the links. This force increases at the beginning as soon as the distance starts to diminish,

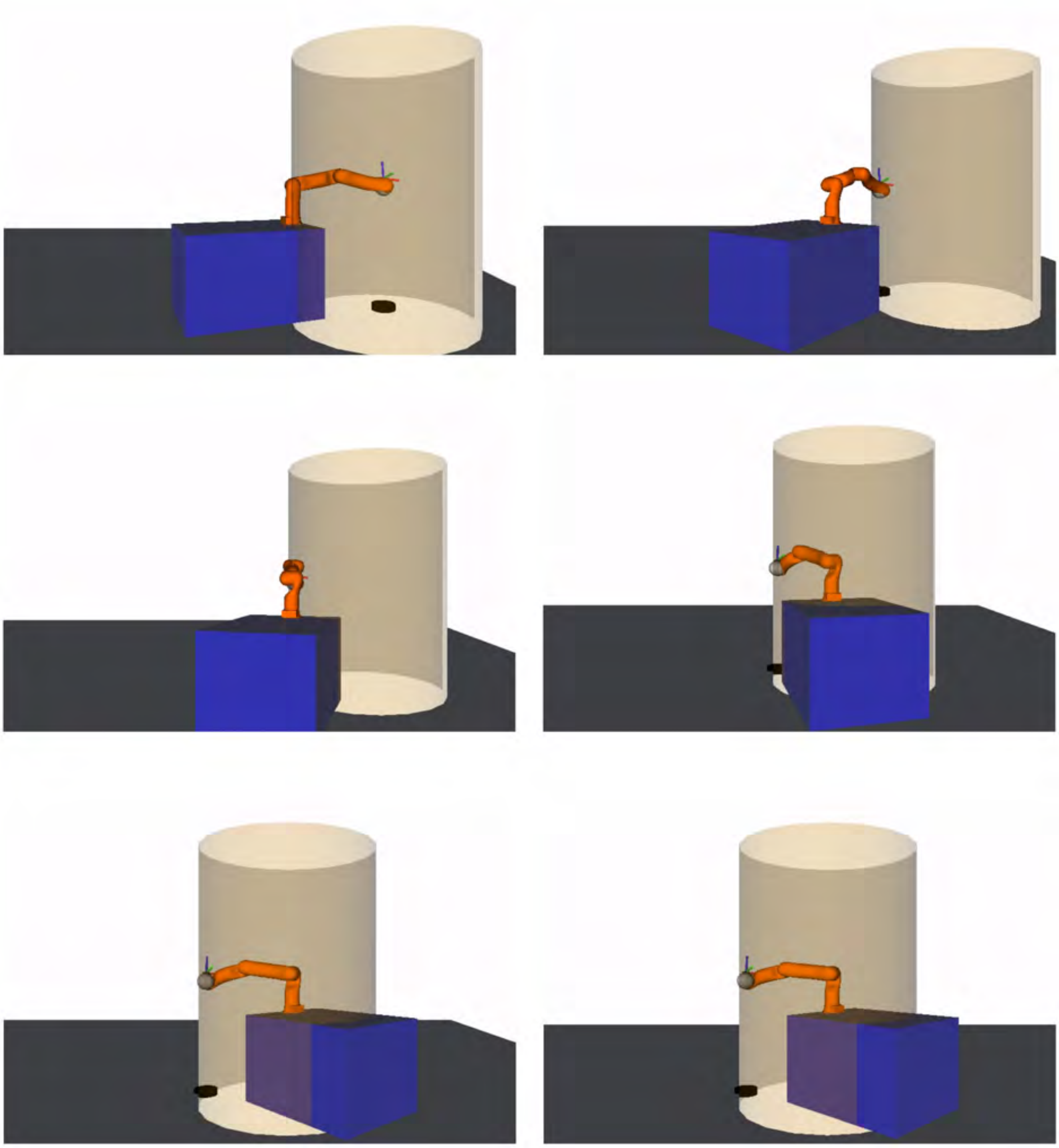


Figure 8.6 Object best view function simulation 2 without activation. From left to right and top to bottom, a motion parallel to the image plane is shown that takes the end-effector behind the robot when the OBVF is not activated. Note that in the third image (second row, first column) the robot end-effector is occluded. The last two figures are similar because the mobile manipulator is only translating to the left, as if pulled by its end-effector.

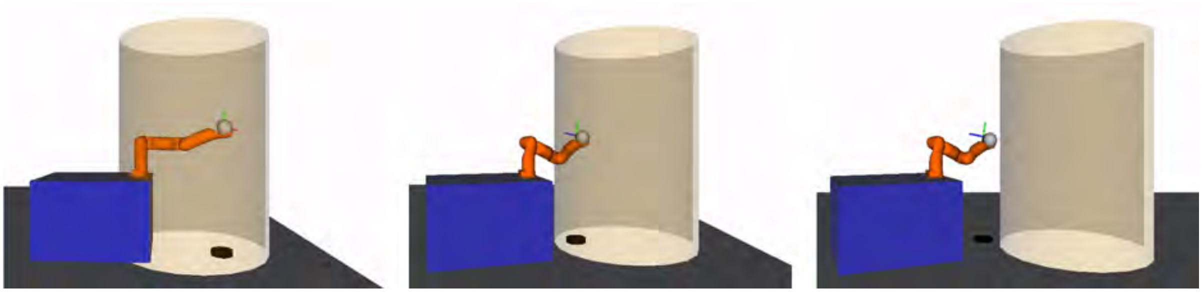


Figure 8.7 Object best view function simulation 2 with activation. From left to right, the motion parallel to the image plane is shown such that the end-effector moves behind the robot with the OBVF activated. The robot end-effector is never occluded and the primary task is fulfilled, that is, the mobile manipulator end-effector is translated to its left.

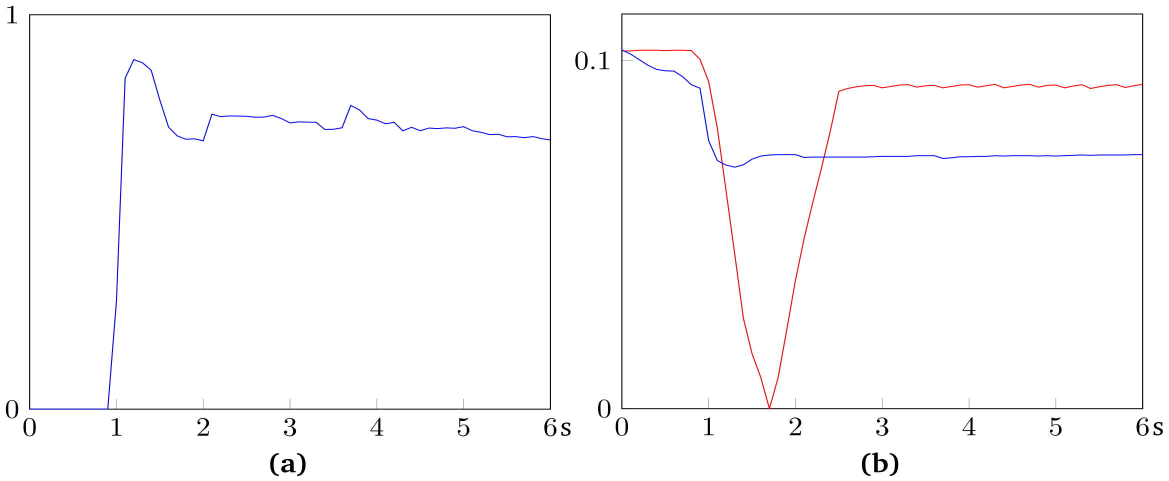


Figure 8.8 Object best view function simulation 2: a) modulus of the force $\boldsymbol{f}$ when the OBVF is active; b) minimum of the distances d_1 and d_2 with the OBVF active (blue) and not active (red).

peaks and finally stabilizes.

Note that, differently from the first simulation, the force exerted does not become zero after it peaks. This is because the commanded trajectory is such that its end-effector is always pushed against its links, so that the force $\boldsymbol{f}$ is never *allowed* to diminish. On the contrary, in the first simulation, the desired trajectory for the end-effector is such that the OBVF exerts a force that move the robot links fully away from the end-effector, making then force $\boldsymbol{f}$ equal to zero.

A video with this type of motion can be found on the link *Simulation 2*.

Simulation 3

The last simulation consists on the *escape* of the BMM-I from a configuration in which, from the operator point of view, the end-effector is very close to the links (left image in Figs. 8.9). Note that the links 2 and 3 are much closer to the end-effector in the image plane than the first link.

Suddenly, the OBVF gets activated, roughly at the second 0.5. Being the scenario such that the links 2 and 3 are occluding the robot end-effector, it is expected that the influence of the force $\boldsymbol{f}_2$ will be stronger than $\boldsymbol{f}_1$, and so the joints of the robot will rotate in a way that the links 2 and 3 will move away from the end-effector before the platform starts to move.

Figs. 8.9 shows the evolution of the robot configuration once the OBVF is activated, showing the expected behavior and freeing the end-effector from being occluded.

Similar to the first two simulations, Figs. 8.10 show the modulus of the exerted force $\boldsymbol{f}$ and the minimum distance between the end-effector and the links of the robot. This force makes the mentioned distance increase until around the value 0.08 m.

Note the shape of the force in Fig. 8.10a at the beginning, the force $\boldsymbol{f}$ is almost 1.5 N, but since the object best function is not active until the second 0.5, it does not generate a motion on the robot nor an increase of the distance between the robot end-effector and the links. Once the task gets activated, the distance starts to increase, and the force, to decrease. At this point, a high negative slope of the force is followed by a time interval with a constant force at around 0.9 N, followed again by a high negative slope until the force becomes almost null.

This *chair* shape is explained by considering that, at the beginning, the force $\boldsymbol{f}$ contains both the influence of the force $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$, and that, as the robot end-effector is very close to the link 2, the contribution of $\boldsymbol{f}_2$ is maximum, *i.e.*, $\boldsymbol{f}_2 = k_2\,\boldsymbol{u}_2$. Once the OBVF gets active the force starts to move the links 1 and 2 away from the robot end-effector. During the first high negative slope phase the force $\boldsymbol{f}_2$ is constant, as the distance between the end-effector and the link 2 is still very small ($d_2 < d_i^m$ as in Eq. 8.6), but $\boldsymbol{f}_1$ is decreasing ($d_1 \in [d_1^m, d_1^M]$). Once $\boldsymbol{f}_1$ becomes null ($d_1 > d_1^M$), the only contribution of $\boldsymbol{f}$ is $\boldsymbol{f}_2$ (and thus, $\|\boldsymbol{f}\| = \|\boldsymbol{f}_2\| = \|k_2\,\boldsymbol{u}_2\| = k_2$) so that for around 0.3 seconds the modulus of the force is constant and the distance between the end-effector and the robot links continues to increase (in particular with link 2). Finally, d_2 becomes such that $\boldsymbol{f}_2$ starts to diminish ($d_2 \in [d_2^m, d_2^M]$), and ultimately becomes null ($d_2 > d_2^M$).

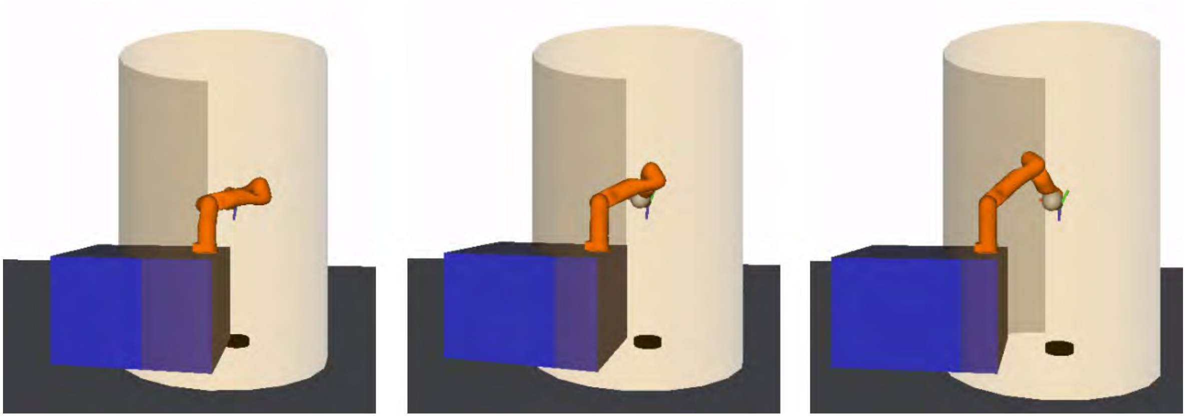


Figure 8.9 Object best view function simulation 3. The activation of the OBVF from an initial configuration with the end-effector very close to the links of the robot in the image plane.

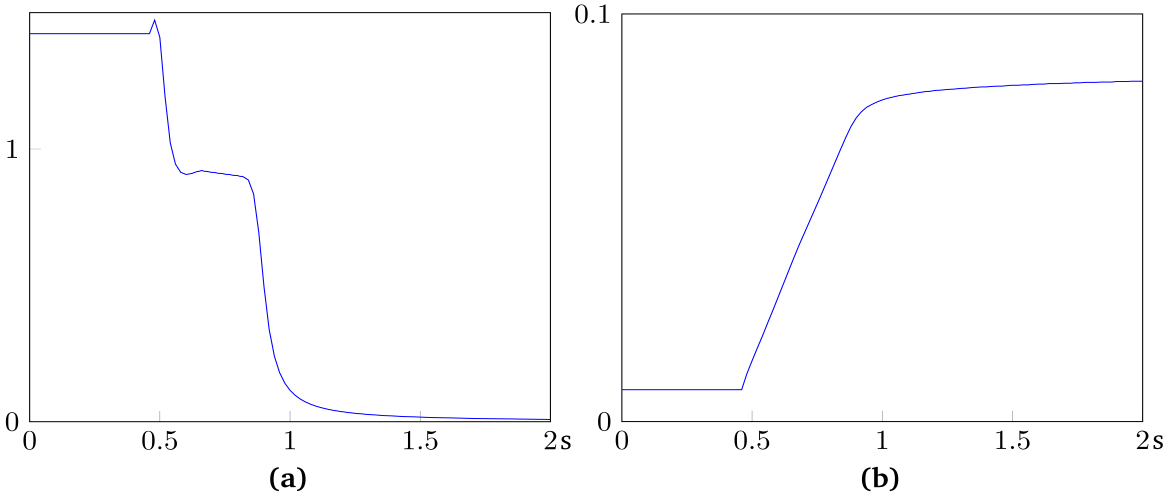


Figure 8.10 Object best view function simulation 3: a) modulus of the force $\boldsymbol{f}$; b) minimum of the distances d_1 and d_2.

Conclusions

The three proposed scenarios to test the OBVF have been successfully dealt with. In all three cases the proposed solution has prevented the robot end-effector from being occluded by the robot parts, which is the goal of this part of this dissertation. Thus, the simulations show that the approach presented previously is convenient to prevent the end-effector occlusions by the robot.

The proposed approach provides the user with a tool to teleoperate a mobile manipulator carrying an object in such a way that the object is always kept at direct view of the operator.

8.4 Chapter Contributions

A novel use of the robot null space is presented. The proposed approach prevents occlusions of the mobile manipulator end-effector, as seen by the operator, that may occur due to parts of the robot intersecting the line of sight between the end-effector and the free-flying camera. To the author's knowledge, this is the first time that such task is implemented as a secondary task using the robot null space.

Chapter 9

The Drone Track Function

The thief left it behind:
the moon
at the window.

Ryokan

Summary: This chapter presents an enhancement of the previously presented workspace mapping algorithm to command the mobile manipulator and the free-flying camera with a single haptic device. The proposed enhancement frees the camera from unnecessary motions by only triggering movements when the robot image is about to leave the image plane of the camera. The development of the algorithm is presented along with some simulations.

9.1 Introduction

The approach presented in Chapter 6 to coordinate the mobile manipulator and the free-flying camera allows the operator to command the robot without explicitly considering the motion of the camera. This is appropriate, since the operator has limited attentional and physical (hands) resources, and a need arises to free him from as much demand as possible.

The previously proposed approach, though, is quite demanding of the drone. For instance, a task may require the mobile manipulator to execute a *gross motion*, that is, a motion that translates the robot end-effector to a position outside of the manipulator workspace. For these kind of motions, a translation of the platform is needed to bring the manipulator at the final end-effector position, and, each time the platform is moving, the camera moves too.

Concretely, recalling the Algorithm 1 in Chapter 6, the motions of the free-flying camera are computed in relation to the motions of the haptic subspace counterpart in the remote scenario. When the user locates the haptic tip near the boundaries of the haptic workspace, a linear motion is triggered on the haptic subspace counterpart, which provokes a motion of the camera that keeps the origin of the haptic subspace counterpart in the middle of the camera field of view. This in turn implies that the robot will stay in the central area of the image plane at all time.

This is in contrast with tasks that demand *fine motions* of the mobile manipulator, that is, translations in which the goal end-effector pose is always inside the workspace of the manipulator, and a motion of the platform is not necessary for the robot to reach it. In these situations, the algorithm presented in Chapter 6 does not trigger any camera motion.

But the activation of the camera translations during gross motions of the mobile manipulator may also be unnecessary in some cases, like in a situation where the final end-effector pose is only a few centimeters outside of the manipulator workspace. In that case, a small translation of the platform will be executed, which will in turn trigger a small translation of the camera. During such tasks, though, the robot will in general remain at any moment inside the camera field of view. So there would not be a real necessity of moving the camera, in terms of easing the teleoperation experience.

In particular, it can be noted that, regarding the performance of the teleoperation, it is only strictly required to trigger the camera motion when the mobile manipulator is about to leave the camera field of view, in order to allow the operator to see the robot at any time.

For instance, let us consider a task where the robot needs to grasp an object on a table, with the final goal to move it to another table that lies outside the reachable workspace of its manipulator, but still inside the camera field of view. The operator should be able command the robot to the second table without moving the camera.

This approach for the coordination between the mobile manipulator and the camera would also ease the teleoperation experience by lowering the motion of the camera, stressing less the operator, and demanding less energy from the UAV.

Summarizing, the aim of this chapter is to present an algorithm that allows this type of coordination between the different parts of the teleoperation robotic system.

The free-flying camera should only translate when not doing so would render insufficient the view of the mobile manipulator from the user perspective, whether because the robot is about to leave the camera field of view, or because it is too close or too far to the image plane, which would also make the teleoperation infeasible.

9.2 Proposed Solution

9.2.1 Requirements

The proposed approach to only command the free-flying camera when the robot is about to disappear from the camera field of view is presented below.

In order to achieve this behavior, some requirements need to be fulfilled:

1. The mobile manipulator should be able to move freely inside the camera field of view without triggering a motion of the free-flying camera. This would avoid unnecessary motions for the UAV during the teleoperation.

   Specifically, the robot should be able to move freely inside a frustum of the pyramid of vision of the camera (Sect. 3.4). The frustum, a volume bounded by the camera field of view, also considers if the robot is neither too close or far away from the camera.

2. The free-flying camera should start moving when the robot reaches the boundaries of the camera frustum, in such a way that the operator could always see the robot through the camera.

   The core idea is that, during gross motions of the robot, the camera motion is such that the image center tracks the robot image only when the mobile manipulator is about to leave the frustum of the camera pyramid of vision. The UAV should then "follow" the robot across the scenario until this gross motion is over.

   Concretely, when this movement of the camera is initiated, the camera motion should be such that the robot image would approach the image center, that is, roughly, that some measure of the offset between the image center and the robot image should approach to zero.

3. The camera should rotate when the haptic tip reaches the rotation workspace boundary of the haptic device as in the algorithm of Section 6.2.4.

4. If the robot image is at an arbitrary position in the camera image plane and the rotation motion of the camera around the robot is activated (by satisfying requirement 3) the rotation should not modify the offset between the robot image and the image center. In other words, the relative position of the robot with respect to the camera should remain constant, and so would the position of the robot in the operator screen.

   This requirement should only be met whenever requirement 2 is not active.

An algorithm that satisfies all these requirements is proposed below.

9.2.2 Kinematic model

Prior to the development of the algorithm, it is advisable to recall some concepts and to introduce some new ones:

- As detailed in Section 6.2, a frame so is attached to the haptic subspace counterpart in the remote scenario. The shape of this subspace is a cylinder (bottom figure in Figs. 6.1).
- A frame, co, is attached to a frustum of the pyramid of vision of the camera (Section 3.4), as explained in Section 6.2, such that the frames c and co are coupled, that is, the frame co is fixed w.r.t. the frame c.
- A *pinhole camera model* is used.

  The frame c, as explained Section 6.2, coincides with the coordinate system $\mathcal{F}_c$ of the pinhole camera model (Section 3.4).
- The pose of the UAV is determined by a point in a three dimensional space and a rotation angle around its vertical axis. Then the total number of DOFs of the UAV amounts to 4: 3 DOFs of the 3D position, and 1 DOF regarding the rotation around the Z axis.

  Since the camera is attached to the UAV, the camera position and orientation w.r.t. to the world frame, s, of the remote scenario can be described by a 4-dimensional vector

  $$\boldsymbol{x}_c^s = \begin{bmatrix} p_{c_x}^s & p_{c_y}^s & p_{c_z}^s & \phi_{c_z}^s \end{bmatrix}^T \tag{9.1}$$

  and its velocity

  $$\dot{\boldsymbol{x}}_c^s = \begin{bmatrix} \dot{p}_{c_x}^s & \dot{p}_{c_y}^s & \dot{p}_{c_z}^s & \omega_{c_z}^s \end{bmatrix}^T \tag{9.2}$$

  with $\boldsymbol{\omega}_c^s = \dot{\boldsymbol{\phi}}_c^s$. The following vectors may be used from now on:

  $$\begin{aligned} \boldsymbol{p}_c^s &= \begin{bmatrix} p_{c_x}^s & p_{c_y}^s & p_{c_z}^s \end{bmatrix}^T & \qquad \boldsymbol{\phi}_c^s &= \begin{bmatrix} 0 & 0 & \phi_{c_z}^s \end{bmatrix}^T \\ \dot{\boldsymbol{p}}_c^s &= \begin{bmatrix} \dot{p}_{c_x}^s & \dot{p}_{c_y}^s & \dot{p}_{c_z}^s \end{bmatrix}^T & \qquad \boldsymbol{\omega}_c^s &= \begin{bmatrix} 0 & 0 & \omega_{c_z}^s \end{bmatrix}^T . \end{aligned}$$
- The UAV is commanded by imposing a desired linear velocity and a desired angular velocity around the Z axis.

  Thus, the command input to the UAV is the 4-dimensional vector

  $$\boldsymbol{u}_c^s = [u_{c_x}^s \quad u_{c_y}^s \quad u_{c_z}^s \quad u_{c_\phi}^s]^T,$$

  where $u_{c_x}^s$, $u_{c_y}^s$ and $u_{c_z}^s$ are the X, Y and Z components of the desired linear velocity; and $u_{c_\phi}^s$, the desired Z axis angular velocity.

9.2.3 Subtasks

It is necessary to formalize the requirements presented in Section 9.2.1 in order to extract the particular subtasks that the will be entrusted to the free-flying camera:

1. Requirement 2 implies that, when the robot is about to leave the frustum of the camera, the camera has to move in order to bring the projection of the robot in the image plane close to the image center.

   One way to address this requirement is by noting that the location of the manipulator workspace is similar to the haptic subspace counterpart location. Roughly, the robot actually tracks the haptic subspace counterpart, since the desired end-effector frame is always inside the haptic subspace counterpart (Fig. 6.3).

   Thus, by imposing that the center of the frustum converges to the center of the haptic subspace counterpart, the projection of the origin of the haptic subspace counterpart frame can be made to converge towards the center of the image plane, and, as a consequence, the mobile manipulator will converge too. This can be formalized through a subtask that implies

   $$\boldsymbol{p}^{s}_{co} \rightarrow \boldsymbol{p}^{s}_{so}, \tag{9.3}$$

   with $\boldsymbol{p}^{s}_{co}$, the center of the frustum, which is coupled to the camera; and $\boldsymbol{p}^{s}_{so}$, the frame origin of the haptic subspace counterpart in the remote scenario.

   Now, considering

   $$\boldsymbol{R}^{s}_{so}\,\boldsymbol{p}^{so}_{co} = \boldsymbol{p}^{s}_{co} - \boldsymbol{p}^{s}_{so}, \tag{9.4}$$

   Eq. 9.3 can also be stated as

   $$\boldsymbol{p}^{so}_{co} \rightarrow \boldsymbol{0}, \tag{9.5}$$

   which will be more convenient to develop the proposed solution.

   Actually, the Z dimension (height) of $\boldsymbol{p}^{so}_{co}$ is of no interest when commanding the drone. In this regard, Fig. 9.1 depicts the projection of $\boldsymbol{p}^{so}_{co}$ in the XY plane, $\boldsymbol{p}^{so}_{co_{XY}}$, which is the vector that it is intended to lead to zero. But, in order to ease the visualization of the equations, from now on, the *XY* subindex will be dropped and $\boldsymbol{p}^{so}_{co}$ will be used to represent $\boldsymbol{p}^{so}_{co_{XY}}$.

2. Requirement 3 implies that the camera has to rotate around the haptic subspace counterpart whenever the orientation of the haptic tip frame reaches the boundaries of the haptic orientation workspace.

   This requirement can be stated as $\boldsymbol{\phi}^{s}_{c} = \boldsymbol{\phi}^{s}_{so}$ or $\boldsymbol{\phi}^{s}_{co} = \boldsymbol{\phi}^{s}_{so}$, since the frames c and co have the same orientation. This is equivalent to impose that the camera orientation tracks the orientation of the haptic subspace counterpart frame:

   $$\boldsymbol{\phi}^{s}_{co} \rightarrow \boldsymbol{\phi}^{s}_{so}. \tag{9.6}$$

   Recalling that the only relevant orientation is due to rotations around the Z axis, the previous relation can be further simplified to

   $$\phi^{s}_{co_z} \rightarrow \phi^{s}_{so_z}. \tag{9.7}$$

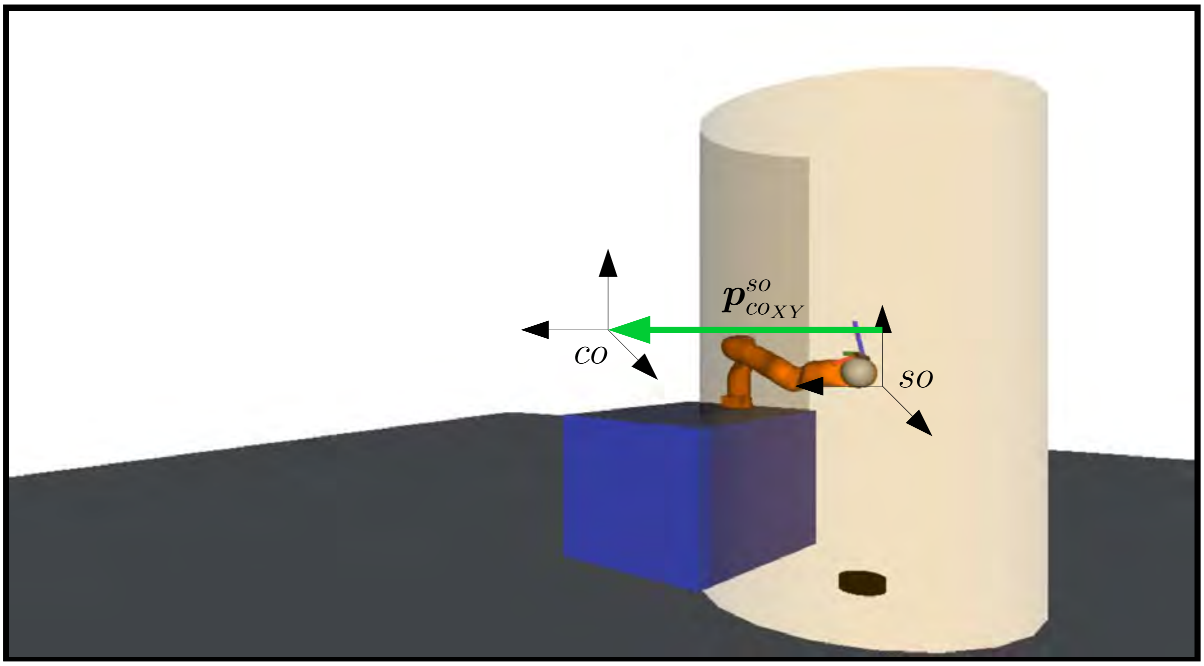


Figure 9.1 The XY projection of the offset between the camera pyramid of vision frustum and the haptic subspace counterpart centers (green line), co and so, respectively, as seen from the camera.

3. When a rotation of the camera is taking place, and the robot is inside the camera frustum, requirement 4 imposes that the offset between the mobile manipulator and the camera should be kept constant.

   This can be formalized by imposing that, once the camera motion is triggered, the vector between the haptic subspace counterpart center and the center of the frustum does not change. Let $\boldsymbol{p}^{so}_{co_{NR}}$ be this offset at the moment of the activation of the rotation. The task can be defined as

$$\boldsymbol{p}^{so}_{co} \to \boldsymbol{p}^{so}_{co_{NR}}. \tag{9.8}$$

Subtasks hierarchy

Two levels of priority can be extracted from an analysis of the previous requirements and subtasks.

But first, it is important to note that subtask 2 has to be always active, whether subtasks 1 or 3 can be or not active.

Then, note that subtask 3 can only be activated when there is a rotation, that is, when subtask 2 gets activated. Thus, whenever active, subtasks 2 and 3 have to be active simultaneously. This clearly defines a level of priority composed of these subtasks.

But subtask 3 can only be imposed when requirement 2 is not met, that is, when subtask 1 is not active. This implies that subtask 3 has a lower priority than subtask 1.

At the same time, since subtask 2 has to be always active, this implies another level of priority composed of subtasks 1 and 2. This subtasks need to have the highest level of priority.

Formalizing these ideas, two sets of subtasks at different levels of priority can be defined. Each subtask set will be called from now on a *priority task* or simply a *task*. Now task 1 and task 2 can be presented through their respective errors

$$\boldsymbol{\epsilon}_1 = \begin{bmatrix} \boldsymbol{p}^{so}_{co_{d1}} - \boldsymbol{p}^{so}_{co} \\ \phi^s_{so_z} - \phi^s_{c_z} \end{bmatrix} \qquad \boldsymbol{\epsilon}_2 = \begin{bmatrix} \boldsymbol{p}^{so}_{co_{d2}} - \boldsymbol{p}^{so}_{co} \\ \phi^s_{so_z} - \phi^s_{c_z} \end{bmatrix} \tag{9.9}$$

with the desired offsets $\boldsymbol{p}^{so}_{co_{d1}} = \boldsymbol{0}$ and $\boldsymbol{p}^{so}_{co_{d2}} = \boldsymbol{p}^{so}_{co_{NR}}$, and task 1 having the highest priority.

9.2.4 Algorithm Development

Task 1

As stated previously, the requirement with highest priority imposes Eq. 9.5. This condition demands 3 DOFs. Also, requirement 3 imposes Eq. 9.7, which constraints 1 additional DOF, thus amounting to a total of 4 DOFs for task 1.

The error that accounts for both requirements is

$$\boldsymbol{\epsilon}_1 = \begin{bmatrix} \boldsymbol{p}^{so}_{co_{d1}} - \boldsymbol{p}^{so}_{co} \\ \phi^s_{so_z} - \phi^s_{c_z} \end{bmatrix} \tag{9.10}$$

with the desired offset $\boldsymbol{p}^{so}_{co_{d1}} = \boldsymbol{0}$.

Differentiating Eq. 9.10 with respect to time,

$$\dot{\boldsymbol{\epsilon}}_1 = \begin{bmatrix} \dot{\boldsymbol{p}}^{so}_{co_{d1}} - \dot{\boldsymbol{p}}^{so}_{co} \\ \omega^s_{so_z} - \omega^s_{co_z} \end{bmatrix} \tag{9.11}$$

and from Eq. 9.4

$$\boldsymbol{p}^{so}_{co} = \left(\boldsymbol{R}^s_{so}\right)^T \left(\boldsymbol{p}^s_{co} - \boldsymbol{p}^s_{so}\right) \tag{9.12}$$

and differentiating it is obtained

$$\begin{aligned} \dot{\boldsymbol{p}}^{so}_{co} &= \left(\boldsymbol{S}\left(\boldsymbol{\omega}^s_{so}\right)\boldsymbol{R}^s_{so}\right)^T \left(\boldsymbol{p}^s_{co} - \boldsymbol{p}^s_{so}\right) + \left(\boldsymbol{R}^s_{so}\right)^T \left(\dot{\boldsymbol{p}}^s_{co} - \dot{\boldsymbol{p}}^s_{so}\right) \\ &= \left(\boldsymbol{R}^s_{so}\right)^T \left(\boldsymbol{S}\left(\boldsymbol{\omega}^s_{so}\right)^T \left(\boldsymbol{p}^s_{co} - \boldsymbol{p}^s_{so}\right) + \dot{\boldsymbol{p}}^s_{co} - \dot{\boldsymbol{p}}^s_{so}\right) \\ &= \left(\boldsymbol{R}^s_{so}\right)^T \left(-\boldsymbol{S}\left(\boldsymbol{\omega}^s_{so}\right)\left(\boldsymbol{p}^s_{co} - \boldsymbol{p}^s_{so}\right) + \dot{\boldsymbol{p}}^s_{co} - \dot{\boldsymbol{p}}^s_{so}\right) \\ &= \left(\boldsymbol{R}^s_{so}\right)^T \left(-\boldsymbol{S}\left(\boldsymbol{\omega}^s_{so}\right)\boldsymbol{R}^s_{so}\boldsymbol{p}^{so}_{co} + \dot{\boldsymbol{p}}^s_{co} - \dot{\boldsymbol{p}}^s_{so}\right), \end{aligned} \tag{9.13}$$

then, substituting $\dot{\boldsymbol{p}}^{so}_{co}$ into Eq. 9.11,

$$\begin{aligned} \dot{\boldsymbol{\epsilon}}_1 &= \begin{bmatrix} \dot{\boldsymbol{p}}^{so}_{co_{d1}} - \left(\boldsymbol{R}^s_{so}\right)^T \left(-\boldsymbol{S}\left(\boldsymbol{\omega}^s_{so}\right) \boldsymbol{R}^s_{so} \boldsymbol{p}^{so}_{co} + \dot{\boldsymbol{p}}^s_{co} - \dot{\boldsymbol{p}}^s_{so}\right) \\ \omega^s_{so_z} - \omega^s_{co_z} \end{bmatrix} \\ &= \begin{bmatrix} \dot{\boldsymbol{p}}^{so}_{co_{d1}} \\ 0 \end{bmatrix} + \begin{bmatrix} \left(\boldsymbol{R}^s_{so}\right)^T & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}^s_{so} + \boldsymbol{S}\left(\boldsymbol{\omega}^s_{so}\right) \boldsymbol{R}^s_{so} \boldsymbol{p}^{so}_{co} - \dot{\boldsymbol{p}}^s_{co} \\ \omega^s_{so_z} - \omega^s_{co_z} \end{bmatrix} \end{aligned} \tag{9.14}$$

Now recalling that only the rotation around the Z axis, $\boldsymbol{n}_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$, is considered, then

$$\boldsymbol{\omega}_i = \omega_{i_z} \boldsymbol{n}_z \tag{9.15}$$

and so

$$\boldsymbol{S}\left(\boldsymbol{\omega}_i\right) = \boldsymbol{S}\left(\omega_{i_z} \boldsymbol{n}_z\right) = \omega_{i_z} \boldsymbol{S}\left(\boldsymbol{n}_z\right) = \omega_{i_z} \boldsymbol{S}_z \tag{9.16}$$

with

$$\boldsymbol{S}_z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{9.17}$$

Eq. 9.16 can be substituted in Eq. 9.14 to obtain

$$\begin{aligned} \dot{\boldsymbol{\epsilon}}_1 &= \begin{bmatrix} \dot{\boldsymbol{p}}^{so}_{co_{d1}} \\ 0 \end{bmatrix} + \begin{bmatrix} \left(\boldsymbol{R}^s_{so}\right)^T & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \left(\begin{bmatrix} \dot{\boldsymbol{p}}^s_{so} + \boldsymbol{S}_z \boldsymbol{R}^s_{so} \boldsymbol{p}^{so}_{co} \omega^s_{so_z} \\ \omega^s_{so_z} \end{bmatrix} - \begin{bmatrix} \dot{\boldsymbol{p}}^s_{co} \\ \omega^s_{co_z} \end{bmatrix} \right) \\ &= \begin{bmatrix} \dot{\boldsymbol{p}}^{so}_{co_{d1}} \\ 0 \end{bmatrix} + \begin{bmatrix} \left(\boldsymbol{R}^s_{so}\right)^T & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_z \boldsymbol{R}^s_{so} \boldsymbol{p}^{so}_{co} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}^s_{so} \\ \omega^s_{so_z} \end{bmatrix} - \begin{bmatrix} \dot{\boldsymbol{p}}^s_{co} \\ \omega^s_{co_z} \end{bmatrix} \right). \end{aligned} \tag{9.18}$$

Now, differentiating

$$\begin{bmatrix} \boldsymbol{p}^s_{co} \\ \phi^s_{co_z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}^s_c + \boldsymbol{R}^s_c \boldsymbol{p}^c_{co} \\ \phi^s_{c_z} + \phi^c_{co_z} \end{bmatrix}, \tag{9.19}$$

it gives rise to

$$\begin{aligned} \dot{\boldsymbol{x}}^s_{co} = \begin{bmatrix} \dot{\boldsymbol{p}}^s_{co} \\ \omega^s_{co_z} \end{bmatrix} &= \begin{bmatrix} \dot{\boldsymbol{p}}^s_c + \boldsymbol{S}\left(\boldsymbol{\omega}^s_c\right) \boldsymbol{R}^s_c \boldsymbol{p}^c_{co} + \boldsymbol{R}^s_c \dot{\boldsymbol{p}}^c_{co} \\ \omega^s_{c_z} + \omega^c_{co_z} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{p}}^s_c + \boldsymbol{S}_z \boldsymbol{R}^s_c \boldsymbol{p}^c_{co} \omega^s_{c_z} \\ \omega^s_{c_z} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_z \boldsymbol{R}^s_c \boldsymbol{p}^c_{co} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}^s_c \\ \omega^s_{c_z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_z \boldsymbol{R}^s_c \boldsymbol{p}^c_{co} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \dot{\boldsymbol{x}}^s_c \end{aligned} \tag{9.20}$$

since the coupling between the camera and the center of the frustum is constant ($\dot{\boldsymbol{p}}^c_{co} = \boldsymbol{0}$ and $\omega^c_{co_z} = 0$), and as $\boldsymbol{p}^{so}_{co_{d1}}$ is constant ($\dot{\boldsymbol{p}}^{so}_{co_{d1}} = \boldsymbol{0}$), and substituting to Eq. 9.18:

$$\dot{\boldsymbol{\epsilon}}_1 = \begin{bmatrix} \left(\boldsymbol{R}^s_{so}\right)^T & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_z \boldsymbol{R}^s_{so} \boldsymbol{p}^{so}_{co} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \dot{\boldsymbol{x}}^s_{so} - \begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_z \boldsymbol{R}^s_c \boldsymbol{p}^c_{co} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \dot{\boldsymbol{x}}^s_c \right). \tag{9.21}$$

By imposing a desired velocity $\boldsymbol{u}_1$ to the camera input corresponding to task 1

$$\begin{aligned}\boldsymbol{u}_1 = \dot{\boldsymbol{x}}_c^s &= \begin{bmatrix}\boldsymbol{I} & \boldsymbol{S}_z\,\boldsymbol{R}_c^s\,\boldsymbol{p}_{co}^c\\ \boldsymbol{0}^T & 1\end{bmatrix}^{-1}\left(\begin{bmatrix}\boldsymbol{I} & \boldsymbol{S}_z\,\boldsymbol{R}_{so}^s\,\boldsymbol{p}_{co}^{so}\\ \boldsymbol{0}^T & 1\end{bmatrix}\dot{\boldsymbol{x}}_{so}^s + \begin{bmatrix}(\boldsymbol{R}_{so}^s)^T & \boldsymbol{0}\\ \boldsymbol{0}^T & 1\end{bmatrix}^{-1}\boldsymbol{K}_1\boldsymbol{\epsilon}_1\right)\\ &= \begin{bmatrix}\boldsymbol{I} & -\boldsymbol{S}_z\,\boldsymbol{R}_c^s\,\boldsymbol{p}_{co}^c\\ \boldsymbol{0}^T & 1\end{bmatrix}\left(\begin{bmatrix}\boldsymbol{I} & \boldsymbol{S}_z\,\boldsymbol{R}_{so}^s\,\boldsymbol{p}_{co}^{so}\\ \boldsymbol{0}^T & 1\end{bmatrix}\dot{\boldsymbol{x}}_{so}^s + \begin{bmatrix}\boldsymbol{R}_{so}^s & \boldsymbol{0}\\ \boldsymbol{0}^T & 1\end{bmatrix}\boldsymbol{K}_1\boldsymbol{\epsilon}_1\right)\\ &= \boldsymbol{J}_1^{-1}\,(\boldsymbol{a}_1 + \boldsymbol{R}_1\,\boldsymbol{K}_1\,\boldsymbol{\epsilon}_1) = \boldsymbol{J}_1^{-1}\,\dot{\boldsymbol{x}}_1\end{aligned} \tag{9.22}$$

with $\boldsymbol{K}_1$ being a 4×4 diagonal positive matrix, and

$$\boldsymbol{J}_1 = \begin{bmatrix}\boldsymbol{I} & -\boldsymbol{S}_z\,\boldsymbol{R}_c^s\,\boldsymbol{p}_{co}^c\\ \boldsymbol{0}^T & 1\end{bmatrix} \qquad \boldsymbol{a}_1 = \begin{bmatrix}\boldsymbol{I} & \boldsymbol{S}_z\,\boldsymbol{R}_{so}^s\,\boldsymbol{p}_{co}^{so}\\ \boldsymbol{0}^T & 1\end{bmatrix}\dot{\boldsymbol{x}}_{so}^s \qquad \boldsymbol{R}_1 = \begin{bmatrix}\boldsymbol{R}_{so}^s & \boldsymbol{0}\\ \boldsymbol{0}^T & 1\end{bmatrix} \tag{9.23}$$

the error is made to decrease with $\dot{\boldsymbol{\epsilon}}_1 = -\boldsymbol{K}_1\,\boldsymbol{\epsilon}_1$.

Note that the matrix

$$\boldsymbol{J}_1 = \begin{bmatrix}\boldsymbol{I} & -\boldsymbol{S}_z\,\boldsymbol{R}_c^s\,\boldsymbol{p}_{co}^c\\ \boldsymbol{0}^T & 1\end{bmatrix}$$

is actually the Jacobian of the camera with respect to the task 1, since it relates the camera input $\boldsymbol{u}_1$ (its desired velocity) with the task space vector $\dot{\boldsymbol{x}}_1$.

Task 2

The second task is very similar to task 1:

$$\boldsymbol{\epsilon}_2 = \begin{bmatrix}\boldsymbol{p}_{co_{d2}}^{so} - \boldsymbol{p}_{co}^{so}\\ \phi_{so_z}^s - \phi_{c_z}^s\end{bmatrix} \tag{9.24}$$

with the desired offset $\boldsymbol{p}_{co_{d2}}^{so} = \boldsymbol{p}_{co_{NR}}^{so}$.

Concretely, the requirement 4 states that once the camera rotation around the robot is activated, the offset $\boldsymbol{p}_{co}^{so}$ should remain constant.

Let $\boldsymbol{p}_{co_{NR}}^{so}$ be the offset at the moment of the activation of the rotation motion. The assignment of $\boldsymbol{p}_{co_{NR}}^{so}$ can be formalized as

$$\boldsymbol{p}_{co_{NR}}^{so} = \begin{cases}\boldsymbol{p}_{co_{NR}}^{so} & \text{if } |\omega_{so_z}^s| > 0\\ \boldsymbol{p}_{co}^{so} & \text{else}\end{cases} \tag{9.25}$$

that is, $\boldsymbol{p}_{co_{NR}}^{so}$ is left unchanged when the camera is rotating, and so it will take the last value of $\boldsymbol{p}_{co}^{so}$ previous to the activation of the rotation.

The error of task 2 leads to a similar expression for the desired velocity of the camera:

$$\begin{aligned}\boldsymbol{u}_2 &= \begin{bmatrix}\boldsymbol{I} & -\boldsymbol{S}_z\,\boldsymbol{R}_c^s\,\boldsymbol{p}_{co}^c\\ \boldsymbol{0}^T & 1\end{bmatrix}\left(\begin{bmatrix}\boldsymbol{I} & \boldsymbol{S}_z\,\boldsymbol{R}_{so}^s\,\boldsymbol{p}_{co}^{so}\\ \boldsymbol{0}^T & 1\end{bmatrix}\dot{\boldsymbol{x}}_{so}^s + \begin{bmatrix}\boldsymbol{R}_{so}^s & \boldsymbol{0}\\ \boldsymbol{0}^T & 1\end{bmatrix}\boldsymbol{K}_2\boldsymbol{\epsilon}_2\right)\\ &= \boldsymbol{J}_2^{-1}\,(\boldsymbol{a}_2 + \boldsymbol{R}_2\,\boldsymbol{K}_2\,\boldsymbol{\epsilon}_2) = \boldsymbol{J}_2^{-1}\,\dot{\boldsymbol{x}}_2.\end{aligned} \tag{9.26}$$

It is straightforward to see that

$$\boldsymbol{u}_2 = \boldsymbol{J}_2^{-1}\,\dot{\boldsymbol{x}}_2 \quad = \boldsymbol{J}_2^{-1}\,(\boldsymbol{a}_2 + \boldsymbol{R}_2\,\boldsymbol{K}_2\,\boldsymbol{\epsilon}_2) = \boldsymbol{J}_1^{-1}\,(\boldsymbol{a}_1 + \boldsymbol{R}_1\,\boldsymbol{K}_2\,\boldsymbol{\epsilon}_2) = \boldsymbol{J}_1^{-1}\,\dot{\boldsymbol{x}}_2. \tag{9.27}$$

Merging both tasks

Now the priority hierarchy between tasks 1 and 2 needs to be considered, in order to obtain a single input for the free-flying camera $\boldsymbol{u}_c^s$.

Considering that task 1 and task 2 each need 4 DOFs to be executed, which is exactly the number of DOFs of the free-flying camera, it will not be possible to execute both tasks simultaneously, and not even to partially execute the second task if the first task is being executed. This implies that once the task 1 is being executed, the dimension of its null space will be zero, so it will only be possible to execute the task 2 when task 1 is not active, in which case the task 2 will be fully executed.

Thus, one possible input for the camera to account for both tasks can be:

$$\begin{aligned}
\boldsymbol{u}_c^s &= h_1\,\boldsymbol{u}_1 + (1-h_1)\;h_2\,\boldsymbol{u}_2 = h_1\,\boldsymbol{J}_1^{-1}\,\dot{\boldsymbol{x}}_1 + (1-h_1)\;h_2\,\boldsymbol{J}_1^{-1}\,\dot{\boldsymbol{x}}_2 \\
&= \boldsymbol{J}_1^{-1}\,(h_1\,\dot{\boldsymbol{x}}_1 + (1-h_1)\;h_2\,\dot{\boldsymbol{x}}_2) \\
&= \boldsymbol{J}_1^{-1}\,(h_1\;(\boldsymbol{a}_1 + \boldsymbol{R}_1\,\boldsymbol{K}_1\,\boldsymbol{\epsilon}_1) + (1-h_1)\;h_2\;(\boldsymbol{a}_1 + \boldsymbol{R}_1\,\boldsymbol{K}_2\,\boldsymbol{\epsilon}_2)) \\
&= \boldsymbol{J}_1^{-1}\,((h_1 + (1-h_1)\;h_2)\,\boldsymbol{a}_1 + h_1\,\boldsymbol{R}_1\,\boldsymbol{K}_1\,\boldsymbol{\epsilon}_1 + (1-h_1)\;h_2\,\boldsymbol{R}_1\,\boldsymbol{K}_2\,\boldsymbol{\epsilon}_2)
\end{aligned} \tag{9.28}$$

with the task activation parameters $h_1, h_2 \in \{0,1\}$. The conditions under which h_1 and h_2 are 0 or 1 will be developed below.

By imposing $\boldsymbol{K}_1 = \boldsymbol{K}_2 = \boldsymbol{K}$ it is possible to merge both errors into a single error by first obtaining

$$\boldsymbol{u}_c^s = \boldsymbol{J}_1^{-1}\,((h_1 + (1-h_1)\;h_2)\,\boldsymbol{a}_1 + \boldsymbol{R}_1\,\boldsymbol{K}\,(h_1\,\boldsymbol{\epsilon}_1 + (1-h_1)\;h_2\,\boldsymbol{\epsilon}_2))\,, \tag{9.29}$$

and next, through an analysis of the errors

$$\boldsymbol{\epsilon}_1 = \begin{bmatrix} \boldsymbol{0} - \boldsymbol{p}_{co}^{so} \\ \phi_{so_z}^s - \phi_{c_z}^s \end{bmatrix} \qquad\qquad \boldsymbol{\epsilon}_2 = \begin{bmatrix} \boldsymbol{p}_{co_{NR}}^{so} - \boldsymbol{p}_{co}^{so} \\ \phi_{so_z}^s - \phi_{c_z}^s \end{bmatrix},$$

it can be noted that they can both be unified into a single error

$$\boldsymbol{\epsilon}_h = \begin{bmatrix} (1-h_1)\,\boldsymbol{p}_{co_{NR}}^{so} - \boldsymbol{p}_{co}^{so} \\ \phi_{so_z}^s - \phi_{c_z}^s \end{bmatrix}, \tag{9.30}$$

since $\boldsymbol{\epsilon}_h\,(h_1 = 1) = \boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_h\,(h_1 = 0) = \boldsymbol{\epsilon}_2$, and so

$$\begin{aligned}
\boldsymbol{u}_c^s\,(h_1 = 1) &= \boldsymbol{J}_1^{-1}\,(\boldsymbol{a}_1 + \boldsymbol{R}_1\,\boldsymbol{K}\,\boldsymbol{\epsilon}_1) \\
\boldsymbol{u}_c^s\,(h_1 = 0) &= \boldsymbol{J}_1^{-1}\,(h_2\,\boldsymbol{a}_1 + h_2\,\boldsymbol{R}_1\,\boldsymbol{K}\,\boldsymbol{\epsilon}_2) = h_2\,\boldsymbol{J}_2^{-1}\,\dot{\boldsymbol{x}}_2,
\end{aligned} \tag{9.31}$$

which leads Eq. 9.29 to become

$$\begin{aligned} \boldsymbol{u}_c^s &= \boldsymbol{J}_1^{-1}\left(\left(h_1 + (1-h_1)\ h_2\right) \boldsymbol{a}_1 + \boldsymbol{R}_1\, \boldsymbol{K} \left(h_1\, \boldsymbol{\epsilon}_h + (1-h_1)\ h_2\, \boldsymbol{\epsilon}_h\right)\right) \\ &= \left(h_1 + (1-h_1)\ h_2\right) \boldsymbol{J}_1^{-1} \left(\boldsymbol{a}_1 + \boldsymbol{R}_1\, \boldsymbol{K}\, \boldsymbol{\epsilon}_h\right). \end{aligned} \tag{9.32}$$

Expanding the previous formula, the final expression for $\boldsymbol{u}_c^s$ is

$$\boldsymbol{u}_c^s = P_h \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{S}_z\, \boldsymbol{R}_c^s\, \boldsymbol{p}_{co}^c \\ \boldsymbol{0}^T & 1 \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_z\, \boldsymbol{R}_{so}^s\, \boldsymbol{p}_{co}^{so} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \dot{\boldsymbol{x}}_{so}^s + \begin{bmatrix} \boldsymbol{R}_{so}^s & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \boldsymbol{K} \begin{bmatrix} (1-h_1)\, \boldsymbol{p}_{co_{NR}}^{so} - \boldsymbol{p}_{co}^{so} \\ \phi_{so_z}^s - \phi_{c_z}^s \end{bmatrix} \right) \tag{9.33}$$

with $P_h = h_1 + (1-h_1)\ h_2$ and $\boldsymbol{K}$ a 4×4 diagonal positive matrix.

Finally, the expression for $\boldsymbol{p}_{co_{NR}}^{so}$ in Eq. 9.25 needs to also account for both tasks and their priority hierarchy. In particular, if the camera is rotating but task 1 is active, $\boldsymbol{p}_{co_{NR}}^{so}$ has to be updated. Equivalently, $\boldsymbol{p}_{co_{NR}}^{so}$ must remain constant if the camera is *only* rotating. With this considerations, Eq. 9.25 can be extended to:

$$\boldsymbol{p}_{co_{NR}}^{so} = \begin{cases} \boldsymbol{p}_{co}^{so} & \text{if } h_1 = 1 \text{ or } \left|\omega_{so_z}^s\right| = 0 \\ \boldsymbol{p}_{co_{NR}}^{so} & \text{else} \end{cases} \tag{9.34}$$

Using the continuous inverse

Equation 9.33 can also be obtained by explicitly using the continuous inverse with the two levels of priority. The development is trivial, since task 1 takes the 4 DOFs available in the system, not leaving any remaining DOFs in its null space. Only when task 1 is not active, should task 2 take full control.

In general, the continuous inverse with two levels of priority can be used to compute the joints of a robot as

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\oplus \boldsymbol{H}} \dot{\boldsymbol{x}}_1 + \boldsymbol{J}_2^{\boldsymbol{N}_\oplus^1 \oplus} \left(\dot{\boldsymbol{x}}_2 - \boldsymbol{J}_2 \boldsymbol{J}_1^{\oplus \boldsymbol{H}} \dot{\boldsymbol{x}}_1\right).$$

To adjust this expression to the UAV, take $\dot{\boldsymbol{q}} = \boldsymbol{u}_c^s$, since the input of the UAV is its desired velocity.

Also, task 1 is only activated by one parameter, $\boldsymbol{H} = h_1$. These remarks lead the previous expression to

$$\boldsymbol{u}_c^s = \boldsymbol{J}_1^{\oplus h_1} \dot{\boldsymbol{x}}_1 + \boldsymbol{J}_2^{\boldsymbol{N}_\oplus^1 \oplus} \left(\dot{\boldsymbol{x}}_2 - \boldsymbol{J}_2 \boldsymbol{J}_1^{\oplus h_1} \dot{\boldsymbol{x}}_1\right), \tag{9.35}$$

which can be further developed with

$$\boldsymbol{J}_1^{\oplus \boldsymbol{H}} = \boldsymbol{J}_1^{\oplus h_1} = h_1 \boldsymbol{J}_1^{+} = h_1 \boldsymbol{J}_1^{-1} \tag{9.36}$$

and so

$$\boldsymbol{N}_\oplus^1 = \boldsymbol{I} - \boldsymbol{J}_1^{\oplus \boldsymbol{H}} \boldsymbol{J}_1 = \boldsymbol{I} - \boldsymbol{J}_1^{\oplus h_1} \boldsymbol{J}_1 = \boldsymbol{I} - h_1 \boldsymbol{J}_1^{-1} \boldsymbol{J}_1 = (1-h_1)\boldsymbol{I}, \tag{9.37}$$

which, using Eq. 3.7 and 3.8, allows to solve $\boldsymbol{J}_2^{\boldsymbol{N}_\oplus^1 \oplus}$ as

$$\begin{aligned}\boldsymbol{J}_2^{\boldsymbol{N}_\oplus^1 \oplus} &= \left(\left(\boldsymbol{J}_2^T\right)^{\oplus \boldsymbol{N}_\oplus^1}\right)^T = \left(\left(\boldsymbol{U}_1^T \boldsymbol{J}_2^T\right)^{\oplus \boldsymbol{S}_1} \boldsymbol{U}_1^T\right)^T = \left(\left(\boldsymbol{J}_2^T\right)^{\oplus (1-h_1)}\right)^T \\ &= \left(\left((1-h_1)\left(\boldsymbol{J}_2^T\right)^+\right)\right)^T = (1-h_1)\,\boldsymbol{J}_2^+ = (1-h_1)\,\boldsymbol{J}_1^+ = (1-h_1)\,\boldsymbol{J}_1^{-1}\end{aligned} \tag{9.38}$$

recalling that $\boldsymbol{U}_1$ and $\boldsymbol{S}_1$ are obtained from the SVD decomposition of $\boldsymbol{N}_\oplus^1 = \boldsymbol{U}_1 \boldsymbol{S}_1 \boldsymbol{U}_1{}^T$ and, since $\boldsymbol{N}_\oplus^1$ is a diagonal matrix with all elements equal to $(1-h_1)$, $\boldsymbol{U}_1 = \boldsymbol{I}$ and $\boldsymbol{S}_1 = (1-h_1)\boldsymbol{I}$.

Substituting Eqs. 9.36 and 9.38 into Eq. 9.35 leads to:

$$\begin{aligned}\boldsymbol{u}_c^s &= h_1 \boldsymbol{J}_1^{-1}\dot{\boldsymbol{x}}_1 + (1-h_1)\,\boldsymbol{J}_1^{-1}\left(\dot{\boldsymbol{x}}_2 - \boldsymbol{J}_2 h_1 \boldsymbol{J}_1^{-1}\dot{\boldsymbol{x}}_1\right) \\ &= h_1 \boldsymbol{J}_1^{-1}\dot{\boldsymbol{x}}_1 + (1-h_1)\,\boldsymbol{J}_1^{-1}\left(\dot{\boldsymbol{x}}_2 - h_1 \boldsymbol{J}_1 \boldsymbol{J}_1^{-1}\dot{\boldsymbol{x}}_1\right) \\ &= h_1 \boldsymbol{J}_1^{-1}\dot{\boldsymbol{x}}_1 + (1-h_1)\,\boldsymbol{J}_1^{-1}\left(\dot{\boldsymbol{x}}_2 - h_1 \dot{\boldsymbol{x}}_1\right).\end{aligned} \tag{9.39}$$

The activation of task 2 can be imposed in the previous Equation through the activation parameter h_2 as

$$\begin{aligned}\boldsymbol{u}_c^s &= h_1 \boldsymbol{J}_1^{-1}\dot{\boldsymbol{x}}_1 + (1-h_1)\,h_2 \boldsymbol{J}_1^{-1}\left(\dot{\boldsymbol{x}}_2 - h_1 \dot{\boldsymbol{x}}_1\right) \\ &= \boldsymbol{J}_1^{-1}\left(h_1 \dot{\boldsymbol{x}}_1 + (1-h_1)\,h_2\left(\dot{\boldsymbol{x}}_2 - h_1 \dot{\boldsymbol{x}}_1\right)\right)\end{aligned} \tag{9.40}$$

which, expanding the vectors $\dot{\boldsymbol{x}}_1$ and $\dot{\boldsymbol{x}}_2$ and unifying the errors as previously done to a single error $\boldsymbol{\epsilon}_h$ such that

$$\begin{aligned}\dot{\boldsymbol{x}}_1 &= \boldsymbol{a}_1 + \boldsymbol{R}_1\,\boldsymbol{K}_1\,\boldsymbol{\epsilon}_1 = \boldsymbol{a}_1 + \boldsymbol{R}_1\,\boldsymbol{K}\,\boldsymbol{\epsilon}_h \\ \dot{\boldsymbol{x}}_2 &= \boldsymbol{a}_2 + \boldsymbol{R}_2\,\boldsymbol{K}_2\,\boldsymbol{\epsilon}_2 = \boldsymbol{a}_1 + \boldsymbol{R}_1\,\boldsymbol{K}_2\,\boldsymbol{\epsilon}_2 = \boldsymbol{a}_1 + \boldsymbol{R}_1\,\boldsymbol{K}\,\boldsymbol{\epsilon}_h = \dot{\boldsymbol{x}}_1\end{aligned} \tag{9.41}$$

leads to

$$\begin{aligned}\boldsymbol{u}_c^s &= \boldsymbol{J}_1^{-1}\left(h_1 \dot{\boldsymbol{x}}_1 + (1-h_1)\,h_2\left(\dot{\boldsymbol{x}}_1 - h_1 \dot{\boldsymbol{x}}_1\right)\right) \\ &= \left(h_1 + (1-h_1)^2\,h_2\right)\boldsymbol{J}_1^{-1}\left(\boldsymbol{a}_1 + \boldsymbol{R}_1\,\boldsymbol{K}_1\,\boldsymbol{\epsilon}_1\right).\end{aligned} \tag{9.42}$$

The square over $(1-h_1)$ can be removed without affecting the performance of the algorithm, since h_1 can only take the values 0 or 1, and Eq. 9.32 is finally obtained.

The activation parameters

What are the criteria to set the values of the activation parameters h_1 and h_2?

Task 1 needs to be considered with highest priority when the robot is about to leave the frustum of the camera pyramid of vision.

Also, a metric that considers how far a three dimensional point is from being outside of the volume contained within the frustum is needed. Formally, this metric can be defined as the distance $d_V(\cdot)$ between a given point $\boldsymbol{p}$ and the frustum V_c of a pyramid of vision defined by the frame c, as

$$d_V(V_c, \boldsymbol{p}) = \begin{cases} 0 & \text{if } \boldsymbol{p} \notin V_c \\ \min\left(\|\boldsymbol{p}_{V_c} - \boldsymbol{p}\|\right) & \text{else} \end{cases} \tag{9.43}$$

with $\boldsymbol{p}_{V_c} \in V_c$.

In particular, the minimum distance between a point $\boldsymbol{p}$ and a point $\boldsymbol{p}_{V_c}$ of the pyramid of vision frustum V_c, $\min\left(\|\boldsymbol{p}_{V_c} - \boldsymbol{p}\|\right)$, can be computed as

$$\begin{aligned}\min\left(\|\boldsymbol{p}_{V_c} - \boldsymbol{p}\|\right) = \min\Bigg(&\left|d_c^M - p_z^c\right|, \left|d_c^m - p_z^c\right|, \|\boldsymbol{p}_{XZ}^c\| \sin\left(\frac{\alpha_c^h}{2} - \left|\text{atan}\left(\frac{p_x^c}{p_z^c}\right)\right|\right), \\ &\|\boldsymbol{p}_{YZ}^c\| \sin\left(\frac{\alpha_c^v}{2} - \left|\text{atan}\left(\frac{p_y^c}{p_z^c}\right)\right|\right)\Bigg)\end{aligned} \tag{9.44}$$

considering a camera frame c such that its optical axis corresponds to the Z axis; the vector $\boldsymbol{p}^c = \left(\boldsymbol{R}_c^s\right)^T \left(\boldsymbol{p} - \boldsymbol{p}_c^s\right)$ being the point $\boldsymbol{p}$ expressed in the camera reference frame; and with $\boldsymbol{p}_{XZ}^c$ and $\boldsymbol{p}_{YZ}^c$ as the projections of point $\boldsymbol{p}^c$ in the XZ and YZ planes of the camera reference frame, respectively.

With this considerations, task 1 has to be activated when the robot is outside the frustum of the camera pyramid of vision. The end-effector is considered to be the first point to leave the frustum, since it is the point *pulled* by the operator through the haptic device. Thus, h_1 will be set to 1 when $\boldsymbol{p}_{si}^s \notin V_c$.

The deactivation of task 1 needs to consider both that the robot is inside the frustum, and that the convergence $\boldsymbol{p}_{co}^{so} \rightarrow \boldsymbol{0}$ has been achieved, which in turn implies that $\boldsymbol{p}_{si}^s \in V_c$. Also note that h_1 has to be set to 0 not only when $\boldsymbol{p}_{co}^{so} = \boldsymbol{0}$, since this can casually occur at any moment that the operator is commanding the robot inside the frustum, but when the convergence of task 1 is taking place, that is, $h_1 = 1$. Finally, as stated in requirement 2, the camera should stop tracking the robot only when the latter stops moving, that is, when $\dot{\boldsymbol{p}}_{so}^s = \boldsymbol{0}$.

The previous ideas can be formalized as

$$h_1 = \begin{cases} 0 & \text{if } \left(t = 0 \text{ and } \boldsymbol{p}_{si}^s \in V_c\right) \text{ or } \left(h_1 = 1 \text{ and } \|\boldsymbol{p}_{co}^{so}\| = 0 \text{ and } \|\dot{\boldsymbol{p}}_{so}^s\| = 0\right) \\ 1 & \text{else if } \boldsymbol{p}_{si}^s \notin V_c \end{cases} \tag{9.45}$$

where $(t = 0$ and $\boldsymbol{p}_{si}^s \in V_c)$ accounts for the initial position of the robot by setting $h_1 = 0$ at time $t = 0$ the robot is inside the pyramid of vision of the camera, that is, $\boldsymbol{p}_{si}^s \in V_c$.

The definition of the activation parameter associated to task 2 is straightforward. As stated in the requirement 4, task 2 has to be activated when requirement 3 is met, which occurs when the camera starts rotating around the robot. This can be formalized as

$$h_2 = \begin{cases} 1 & \text{if } \left|\omega_{so_z}^s\right| > 0 \\ 0 & \text{else} \end{cases} \tag{9.46}$$

Finally, Eq. 9.34 can be updated with this definition of h_2 to

$$\boldsymbol{p}_{co_{NR}}^{so} = \begin{cases} \boldsymbol{p}_{co}^{so} & \text{if } h_1 = 1 \text{ or } h_2 = 0 \\ \boldsymbol{p}_{co_{NR}}^{so} & \text{else} \end{cases} \tag{9.47}$$

9.2.5 Summary

Summarizing, the expression to compute the camera input is

$$\boldsymbol{u}_c^s = P_h \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{S}_z\, \boldsymbol{R}_c^s\, \boldsymbol{p}_{co}^c \\ \boldsymbol{0}^T & 1 \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_z\, \boldsymbol{R}_{so}^s\, \boldsymbol{p}_{co}^{so} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \dot{\boldsymbol{x}}_{so}^s + \begin{bmatrix} \boldsymbol{R}_{so}^s & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \boldsymbol{K} \begin{bmatrix} (1-h_1)\, \boldsymbol{p}_{co_{NR}}^{so} - \boldsymbol{p}_{co}^{so} \\ \phi_{so_z}^s - \phi_{c_z}^s \end{bmatrix} \right)$$

where

$$h_1 = \begin{cases} 0 & \text{if } (t = 0 \text{ and } \boldsymbol{p}_{si}^s \in V_c) \text{ or } (h_1 = 1 \text{ and } \|\boldsymbol{p}_{co}^{so}\| = 0 \text{ and } \|\dot{\boldsymbol{p}}_{so}^s\| = 0) \\ 1 & \text{else if } \boldsymbol{p}_{si}^s \notin V_c \end{cases}$$

$$h_2 = \begin{cases} 1 & \text{if } \left|\omega_{so_z}^s\right| > 0 \\ 0 & \text{else} \end{cases}$$

$$\boldsymbol{p}_{co_{NR}}^{so} = \begin{cases} \boldsymbol{p}_{co}^{so} & \text{if } h_1 = 1 \text{ or } h_2 = 0 \\ \boldsymbol{p}_{co_{NR}}^{so} & \text{else} \end{cases}$$

with:

- $h_1, h_2 \in \{0, 1\}$;
- $P_h = h_1 + (1 - h_1)\ h_2$;
- $\boldsymbol{K} \in \mathbb{R}^{4\times 4}$, a positive diagonal matrix;
- $\{\boldsymbol{p}_c^s, \boldsymbol{R}_c^s, \alpha_c^h, \alpha_c^v, d_c^m, d_c^M\}$, the parameters that define the frustum V_c of the pyramid of vision corresponding to the camera frame c.

9.3 Simulations

A set of simulations has been carried on in order to evaluate the solution presented in this Chapter. The simulations are related to the requirements presented above, with the intent to give a better idea of the contribution of each of them. The trajectories have not been preprogrammed, but using the haptic to command, in real-time, the pose of the robot end-effector.

The parameters used in the simulations are $\boldsymbol{K} = 0.5\boldsymbol{I}$, $\alpha_c^h = 68$ arc degrees, $\alpha_c^v = 52$ arc degrees, $d_c^m = 1.5$ m and $d_c^M = 4.0$ m. In practice, small thresholds have been set for the modulus $\left|\omega_{so_z}^s\right|$, $\|\boldsymbol{p}_{co}^{so}\|$ and $\|\dot{\boldsymbol{p}}_{so}^s\|$ in Eqs. 9.25, 9.45 and 9.46: 0.001 rad/s, 0.05 m and 0.001 m/s, respectively.

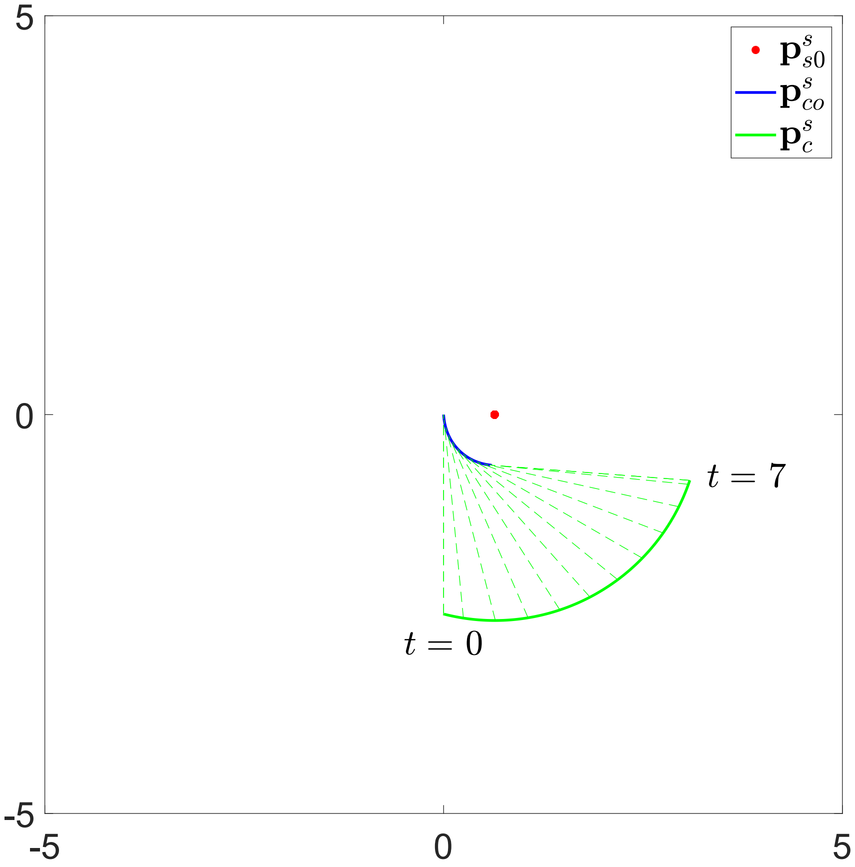


Figure 9.2 Simulation 1. The XY plot in meters of a 90 degrees counter-clockwise rotation trajectory of the camera. Concretely, the values of $\boldsymbol{p}^s_{s0}$, $\boldsymbol{p}^s_{c0}$ and $\boldsymbol{p}^s_c$ in the XY plane along the trajectory are shown, starting with the initial state ($t = 0$) to the end ($t = 7$). The blue curve corresponds to the positions of the center of the frustum along the trajectory; while the green, to the camera position; and the red dot, to the origin of the haptic subspace counterpart. The green dashed lines show the vector $\boldsymbol{p}^c_{c0}$ sampled at equal time intervals, which correspond to the optical axis of the camera, thus giving a sense of what the user is seeing through the camera at each sampled time. Note that $\|\boldsymbol{p}^{so}_{co}\|$ remains constant (the distance between the points in the blue arch and the red point).

Simulation 1

To show how the proposed algorithm behaves with respect to the requirements 3 and 4, the first simulation consists of a motion of the haptic device that triggers a counter clockwise rotation of the camera around the robot for almost 90 degrees during approximately 7 seconds.

The robot image is not initially located at the center of the camera image, but with an offset $\boldsymbol{p}^{so}_{co}$ with a norm of approximately 0.60 m. The rotation starts at the second 1 (Fig. 9.3h) and task 2 gets activated (Fig. 9.3b), and stops after the second 6.

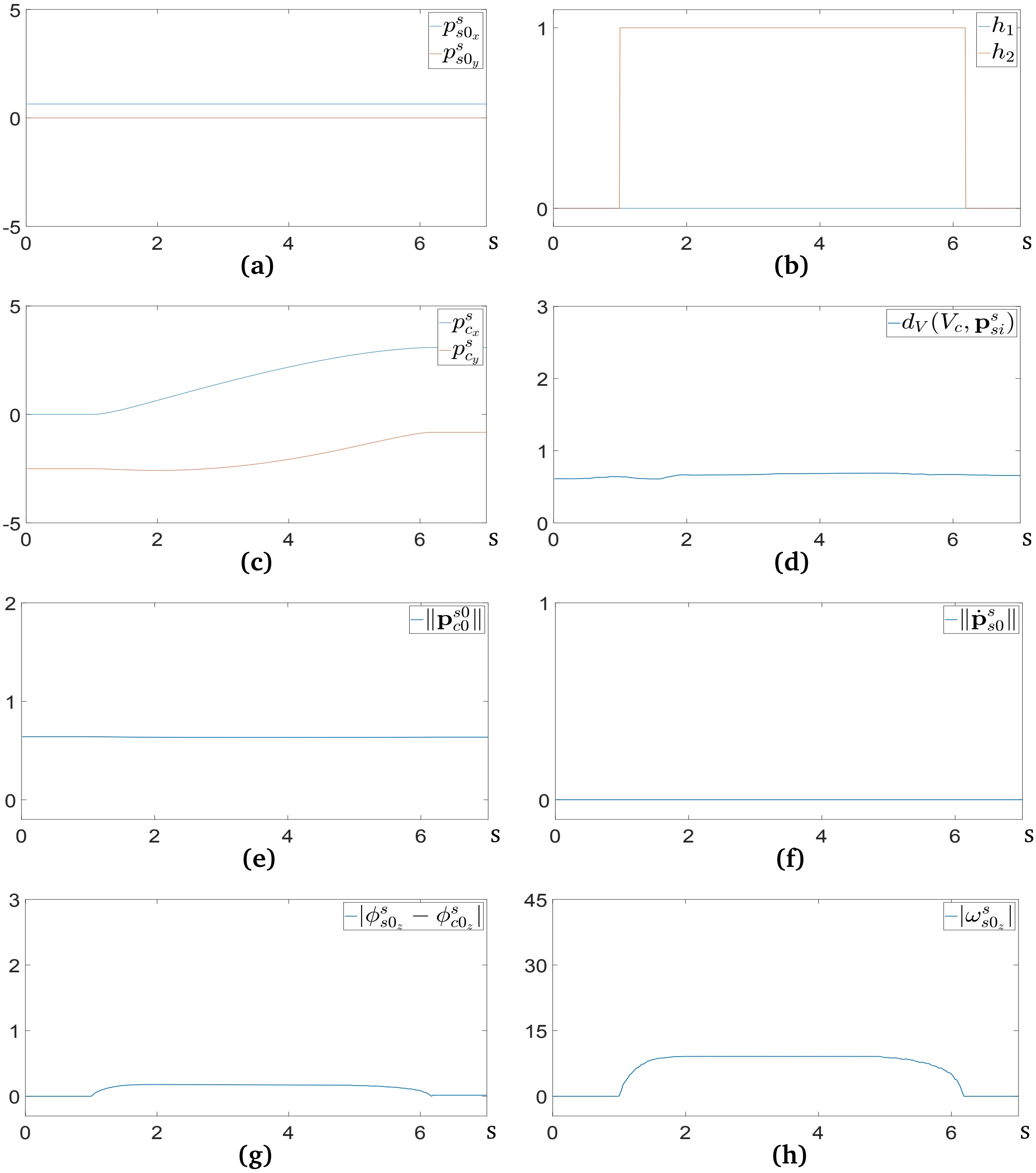


Figure 9.3 Simulation 1 plots: a) origin of the haptic subspace counterpart frame in m; b) activation parameters; c) origin of the camera frame in m; d) distance between the robot TCP and the pyramid of vision frustum in m; e) modulus of the offset between the origin of the haptic subspace counterpart and the frustum center in m; f) linear velocity of the origin of the haptic subspace counterpart frame in m/s; g) modulus of the orientation difference between the haptic subspace counterpart and the pyramid of vision frustum frames in degrees; h) modulus of the angular velocity of the haptic subspace counterpart frame in degrees/s.

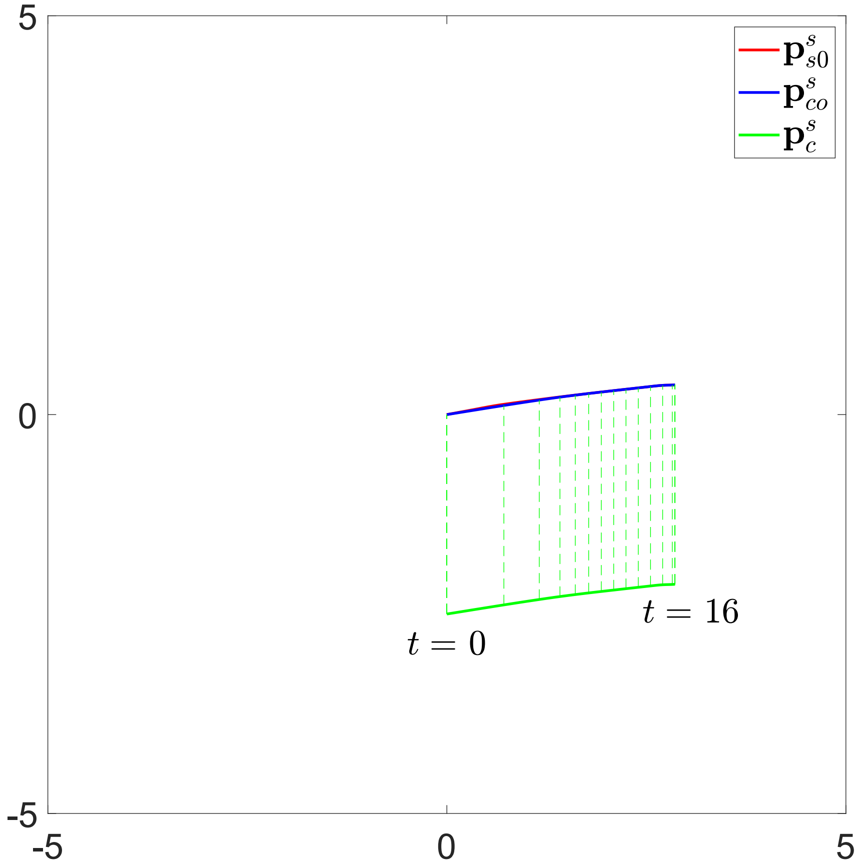


Figure 9.4 Simulation 2. The trajectory in the XY plane of the points $\boldsymbol{p}_{s0}^{s}$, $\boldsymbol{p}_{c0}^{s}$ and $\boldsymbol{p}_{c}^{s}$ in meters: a translation to the right. The green dashed lines show the vector $\boldsymbol{p}_{c0}^{c}$ sampled at equal time intervals, which correspond to the optical axis of the camera.

During the rotation, since the robot always remains inside the frustum of the pyramid of vision, $\boldsymbol{p}_{co}^{so}$ must remain constant. This is shown in Figs. 9.2 and 9.3. In particular, Fig. 9.3e shows the norm of $\boldsymbol{p}_{co}^{so}$, which remains almost constant. Similarly, the norm of the orientation error remains bounded while the camera is rotating (Fig. 9.3g).

Simulation 2

The activation of the camera motion when the robot is about to leave the frustum of the camera pyramid of vision is addressed through requirement 2.

In this simulation, the robot translates from the center of the scenario to the right beginning at time 3 seconds and finishing at 12.5 seconds (Fig. 9.4).

Figure 9.5d shows the distance between the robot end-effector and the frustum of the pyramid of vision. This distance is computed as the minimum distance between the robot end-effector and each of the vertical faces of the frustum with respect to the camera frame (Eq. 9.43).

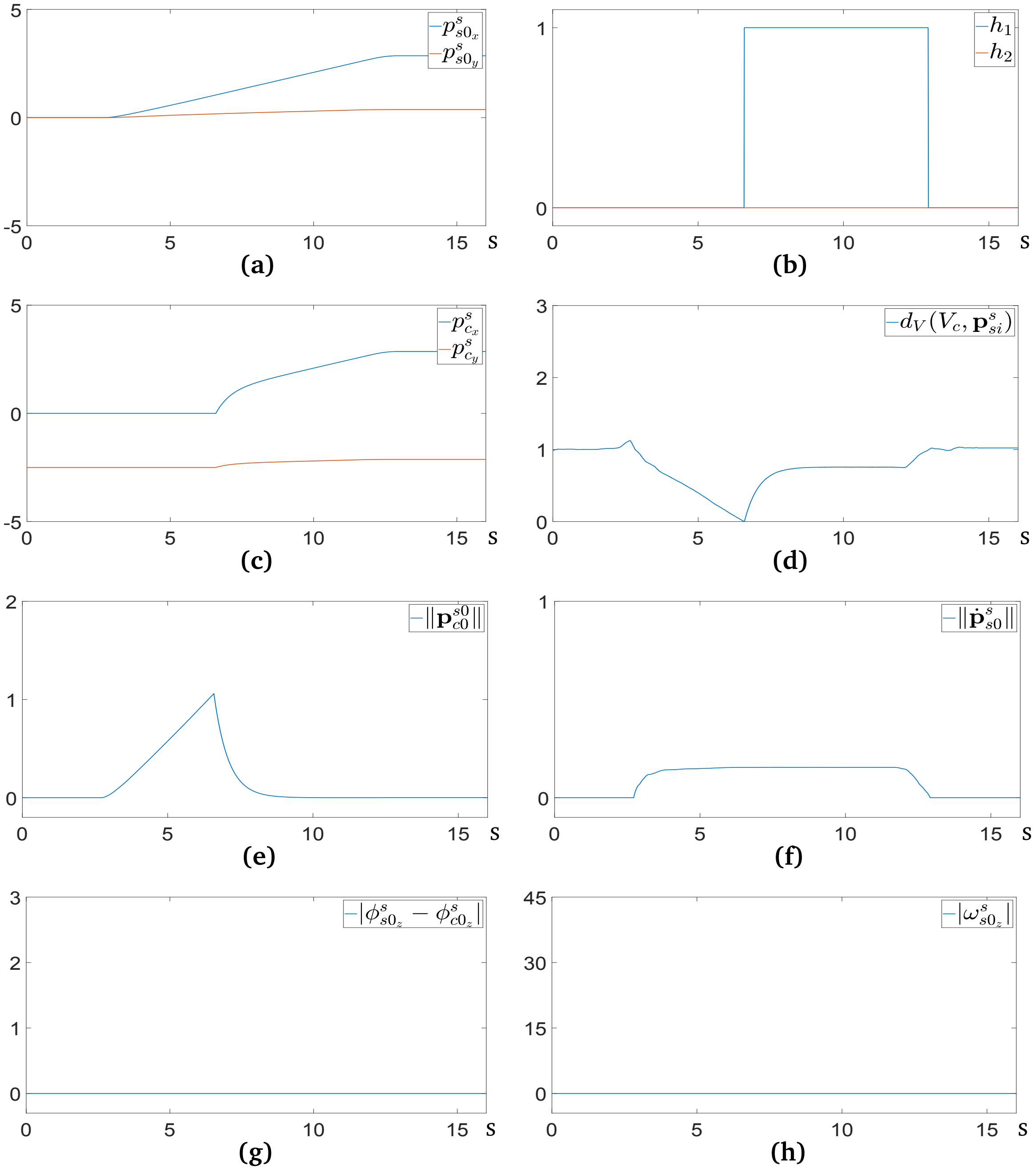


Figure 9.5 Simulation 2 plots: a) origin of the haptic subspace counterpart frame in m; b) activation parameters; c) origin of the camera frame in m; d) distance between the robot TCP and the pyramid of vision frustum in m; e) modulus of the offset between the origin of the haptic subspace counterpart and the frustum center in m; f) linear velocity of the origin of the haptic subspace counterpart frame in m/s; g) modulus of the orientation difference between the haptic subspace counterpart and the pyramid of vision frustum frames in degrees; h) modulus of the angular velocity of the haptic subspace counterpart frame in degrees/s.

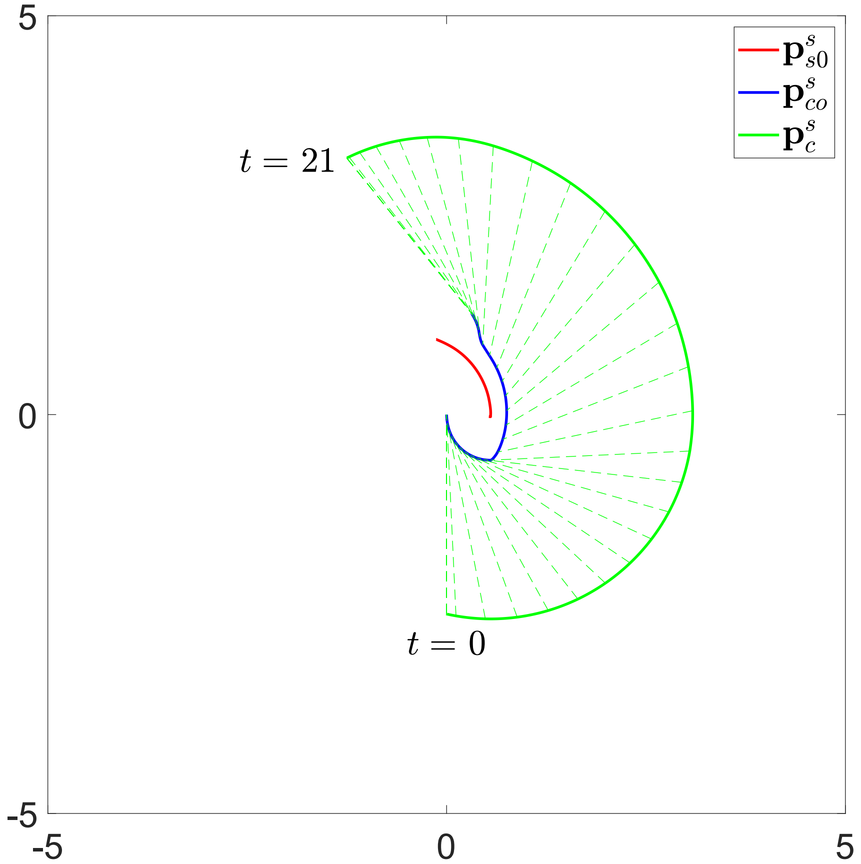


Figure 9.6 Simulation 3. The rotation followed by a translation of the camera in the XY plane. The counter-clockwise rotation can be appreciated, and how the curvature diminishes due to the translation component of the trajectory and the resulting non-null linear velocity part of the camera input. Axis units are meters. The trajectory in the XY plane of the points $\boldsymbol{p}^s_{s0}$, $\boldsymbol{p}^s_{c0}$ and $\boldsymbol{p}^s_{c}$ along the trajectory is shown. The green dashed lines show the vector $\boldsymbol{p}^c_{c0}$ sampled at equal time intervals, which correspond to the optical axis of the camera.

Figure 9.5e shows the distance between the origin of the haptic subspace counterpart frame and the center of the frustum. This distance peaks at around 6.5 seconds, at which moment the robot end-effector reaches the boundaries of the frustum (Fig. 9.5d), and the camera starts to move after h_1 is activated (Fig. 9.5b). The norm of the offset becomes almost zero at time 9 seconds, thus fulfilling the task.

The motion of the camera ends when the robot stops moving at time 12,5 seconds (Fig. 9.5f), and h_1 is set back to zero (Fig. 9.5b).

Simulation 3

The third simulation consists of a motion that begins with a rotation of the camera as in Simulation 1, and then a linear velocity like the one shown in Simulation 2 is added. Figs. 9.6

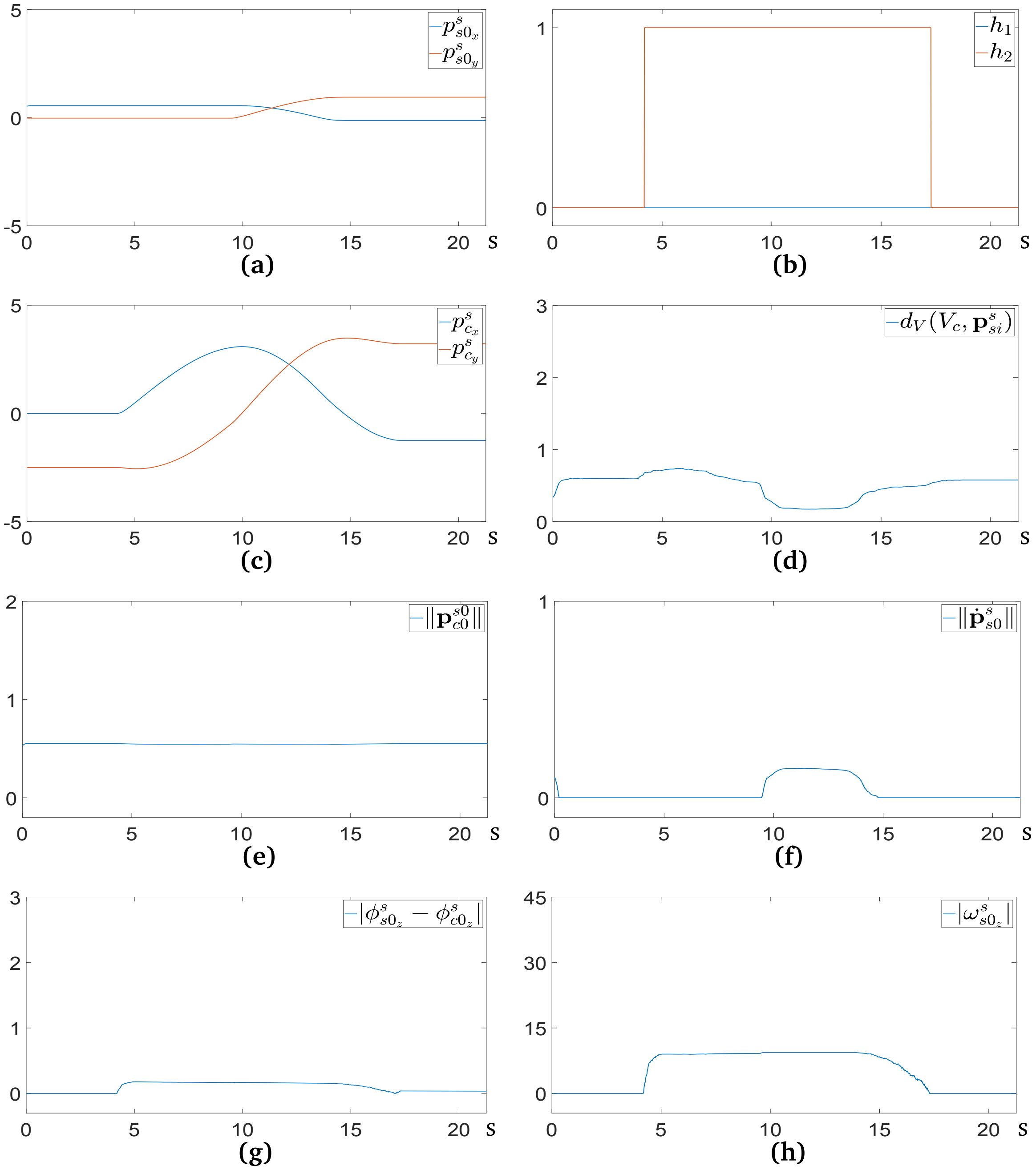


Figure 9.7 Simulation 3 plots: a) origin of the haptic subspace counterpart frame in m; b) activation parameters; c) origin of the camera frame in m; d) distance between the robot TCP and the pyramid of vision frustum in m; e) modulus of the offset between the origin of the haptic subspace counterpart and the frustum center in m; f) linear velocity of the origin of the haptic subspace counterpart frame in m/s; g) modulus of the orientation difference between the haptic subspace counterpart and the pyramid of vision frustum frames in degrees; h) modulus of the angular velocity of the haptic subspace counterpart frame in degrees/s.

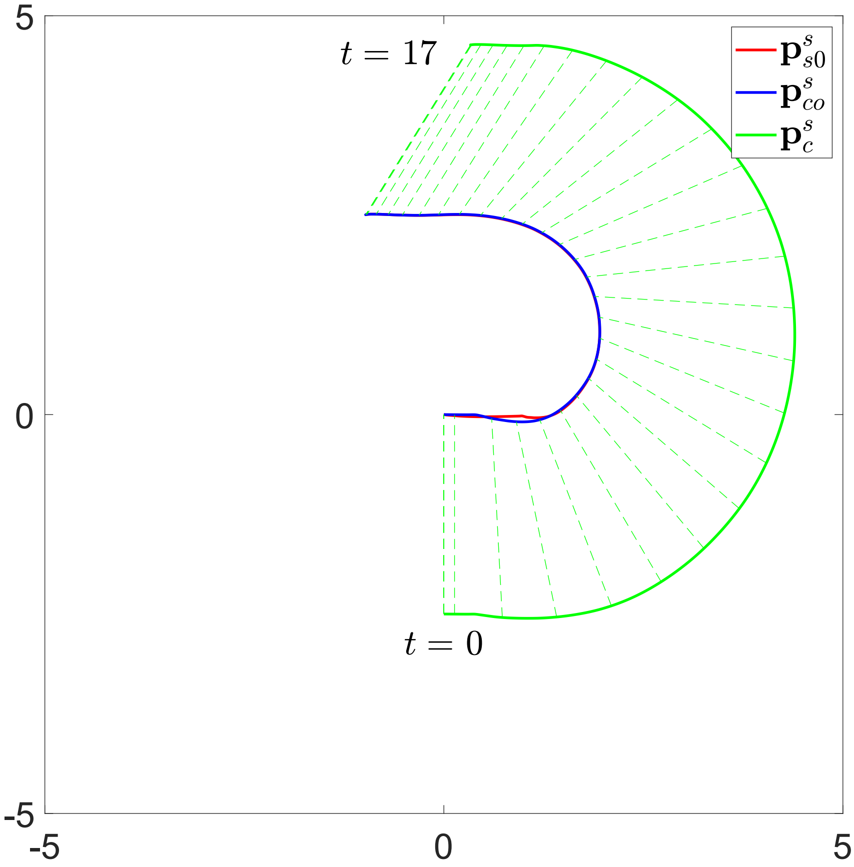


Figure 9.8 Simulation 4. The robot trajectory in meters of the translation followed by a counter-clockwise rotation in the XY plane. Concretely, the trajectories in the XY plane of the points $\boldsymbol{p}^s_{s0}$, $\boldsymbol{p}^s_{c0}$ and $\boldsymbol{p}^s_c$ along the trajectory are shown. The green dashed lines show the vector $\boldsymbol{p}^c_{c0}$ sampled at equal time intervals, which correspond to the optical axis of the camera.

and 9.7 show its results.

The rotation of the camera starts when h_2 is activated at time 4 s (Fig. 9.7b) in response to the rotation of the haptic subspace counterpart (Fig. 9.7h), which stops at 18 seconds. In this regard, it is very similar to Simulation 1. In between, the translation motion is triggered by the user approximately at time 9 seconds, and finishes almost at time 15 seconds (Fig. 9.7a and 9.7f).

Even though the linear motion is added at time 9 seconds and lasts for almost 6 seconds, the $\boldsymbol{p}^{so}_{co}$ offset norm is kept the same (Fig. 9.7e) and the rotation error, bounded (Fig. 9.7g).

Simulation 4

The fourth simulation is similar to the third one, but instead of beginning with a rotation motion and later adding a linear motion, the reverse is done: at the beginning, a linear motion

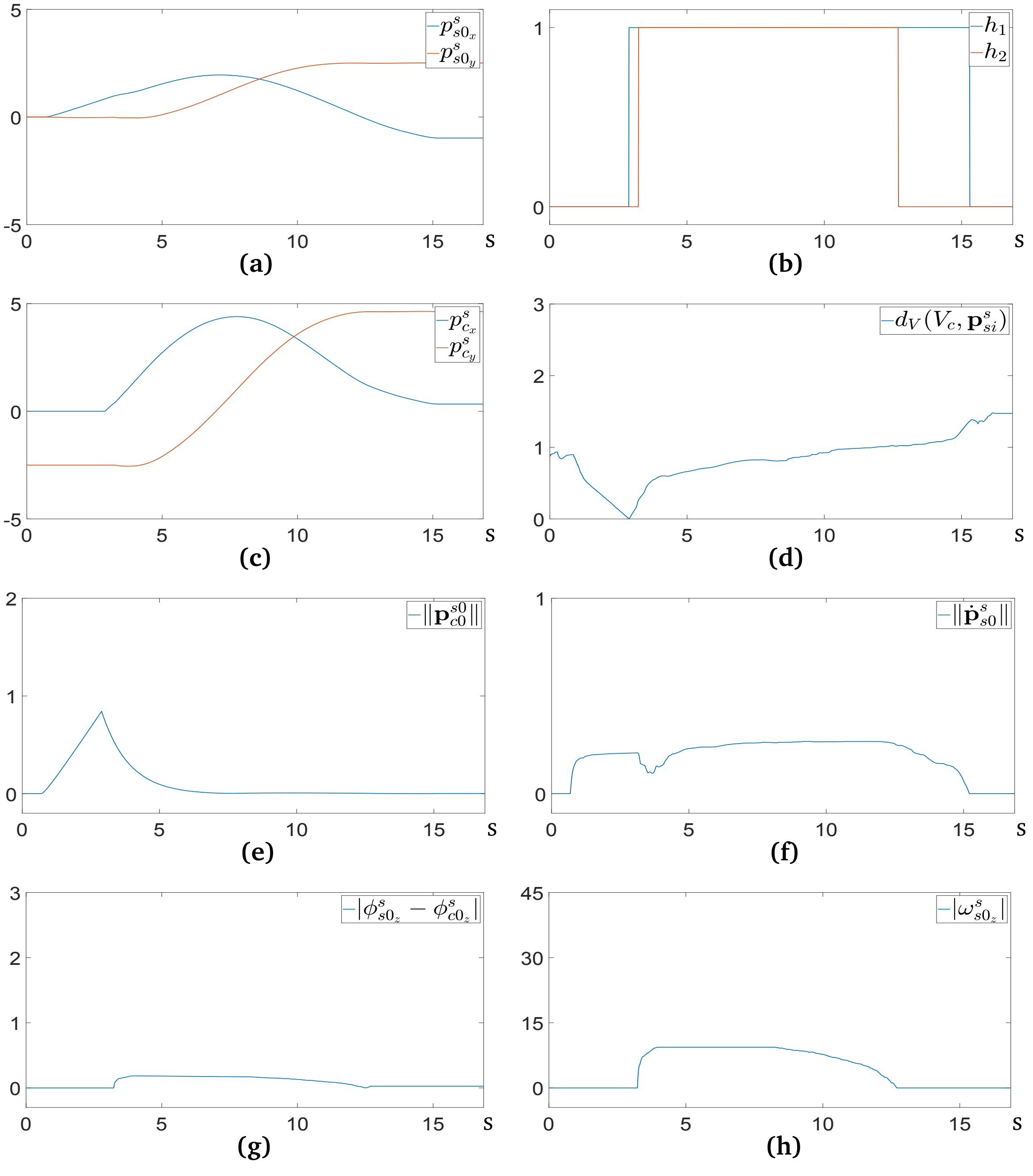


Figure 9.9 Simulation 4 plots: a) origin of the haptic subspace counterpart frame in m; b) activation parameters; c) origin of the camera frame in m; d) distance between the robot TCP and the pyramid of vision frustum in m; e) modulus of the offset between the origin of the haptic subspace counterpart and the frustum center in m; f) linear velocity of the origin of the haptic subspace counterpart frame in m/s; g) modulus of the orientation difference between the haptic subspace counterpart and the pyramid of vision frustum frames in degrees; h) modulus of the angular velocity of the haptic subspace counterpart frame in degrees/s.

is executed that activates the movement of the camera, as in Simulation 2, and, afterwards, a rotation of around 180 degrees is imposed on the camera, as in Simulation 1. Plots of this combined motion can be seen in Figs. 9.8 and 9.9.

Concretely, Fig. 9.9d shows that at time around 3 seconds the robot end-effector reaches the boundaries of the frustum of the camera pyramid of vision. At that moment, task 1 becomes activated (h_1 is set to 1, Fig. 9.9b).

Half a second later, a rotation of the haptic subspace counterpart (Fig. 9.9h) triggers the activation of task 2 (h_2 is set to 1, Fig. 9.9b), and the orientation error is kept bounded during the trajectory (Fig. 9.9g).

Similarly, the offset between the robot and the camera peaks at 3 seconds and starts to diminish until it becomes zero at approximately 7.5 seconds, as expected (Fig. 9.9e).

Task 2 is deactivated at time 12.5 seconds, when the rotation of the haptic subspace counterpart stops (Fig. 9.9h). Three seconds later the linear motion of the robot also stops (Fig. 9.9f), and task 1 is deactivated.

Videos

Two videos with motions similar to the ones of the Simulations have been recorded.

A first video with a set of motions corresponding to the ones in Simulations 1 and 3 can be found in the link: *Simulations 1 and 3*. A first rotation of the camera is shown until the second 15, where the offset $\|\boldsymbol{p}_{co}^{so}\|$ is almost zero and the mobile manipulator is always in the centre of the field of view of the camera. A second rotation of the camera is shown until the second 25, where the offset $\|\boldsymbol{p}_{co}^{so}\|$ is significant and shown to remain constant during all the motion. These two last motions correspond to the motion in Simulation 1. Finally, a third rotation is shown corresponding to Simulation 3. In this last motion a linear motion is added to the robot during the rotation, and the offset also remains constant.

A second video with a set of motions corresponding to the ones in Simulations 2 and 4 can be found in the link: *Simulations 2 and 4*. In particular, three motions of the type of motions of Simulation 2 are shown until the second 57: a first translation that approaches the mobile manipulator to the right boundary of the field of view is shown until the second 17; a second translation that approaches the mobile manipulator to the furthest face of the frustum of the camera pyramid of vision is shown until the second 33; and a third translation that triggers the convergence of the camera towards the haptic subspace counterpart while the robot is still moving, such that the tracking of the camera finishes when the robot motion stops. Finally a motion similar to the Simulation 4 is shown.

9.4 Chapter Contributions

This Chapter presents an enhancement of the algorithm developed in Chapter 6. The modified algorithm only activates the motions of the camera when the robot approaches the field of view boundaries of the free-flying camera, in such a way that the mobile manipulator always remains inside the camera field of view.

Chapter 10

Experimental Validation

Las cosas se hacen haciéndolas.
Anonymous.

Summary: In this chapter, the experimental validation on real robots of the teleoperation robotic system developed in this part of the dissertation is presented. Its main components are detailed, along with an overview of their integration into their robotic system.

10.1 Introduction

The algorithms proposed in the second part of this dissertation have been so far implemented in a simulation environment. Concretely, the simulations have been mainly executed in the Kautham Project (Rosell et al., 2014) and, in order to test particular aspects of the algorithms, in Matlab (2018). Occasionally, Maple (2017) has also been used to manipulate equations.

But an effort has been made to move the developments presented in this part of the thesis to real robots. This has been itself a significant challenge, specially considering that the proposed work includes the handling of a set of robots that, to the author's knowledge, has not been jointly coordinated before.

The current Chapter presents this implementation. First, the robots are introduced. Next, the approach and the tools to integrate the robots are presented, with a detailed explanation of its main modules and its interconnections. Following, the Kautham Project and its use as a simulation environment is explained. The final Section explains a set of videos where several experiments in real setups are shown.

10.2 The robots

10.2.1 The Barcelona Mobile Manipulator I

The mobile manipulator (MM) used to implement the algorithms proposed in this dissertation is the Barcelona Mobile Manipulator I (BMM-I), developed at the IOC.

The BMM-I consists of an omnidirectional mobile platform with an arm manipulator on top of the platform (Fig. 10.1).

The platform is composed by a main body of 138 kg and $1\times0.78\times0.708$ m^3. It has three DOFs: two independent translations and a rotation around the vertical axis, which can be commanded without constraints thanks to its omnidirectional wheels (Fig. 10.2). More mechanical and kinematic details of the platform can be found in (Clos and Martínez, 2007).

The arm manipulator used is a KUKA LWR 4+ (Bischoff et al., 2010). It is a redundant serial robot of seven DOFs, with the third joint being the redundant joint (if locked when its joint value is zero, the manipulator becomes a classical six joint industrial robot). Its maximum length is approximately 1.9 meters. It weights approximately 15 kg and is able to carry a load of 7 kg at full speed, and 15 kg at low speed. Adding to the previous three DOFs of the platform, the BMM-I has a total of ten degrees of freedom.

The platform is controlled by a PC with four Intel Core i5 CPUs running at 3.1GHz on a real-time Linux Xenomai environment. The PC runs an Orocos node on top of an EtherCat software which is used to control the wheels with the intent to guarantee its real-time control at low level. This node also allows the platform to be commanded by velocity inputs at the platform level, that is, by defining its forward and side velocities, and its angular velocity around its vertical axis. This translation-rotation input has been used to command the platform in this work. Over the Orocos node, a ROS node has been developed to communicate with the rest of the system components.

The command of the KUKA LWR is done through the Fast Research Interface library (FRI), an interface supplied by the manufacturer (Schreiber et al., 2010) which gives direct low-level real-time access to the KUKA robot controller. A driver developed at Stanford has been used to communicate with the FRI (LWR, 2014). The FRI allows several command modes: a joint position control, an Cartesian position control of the end-effector, both controls with impedance, and joint torque control. In this work, the position joint control has been used.

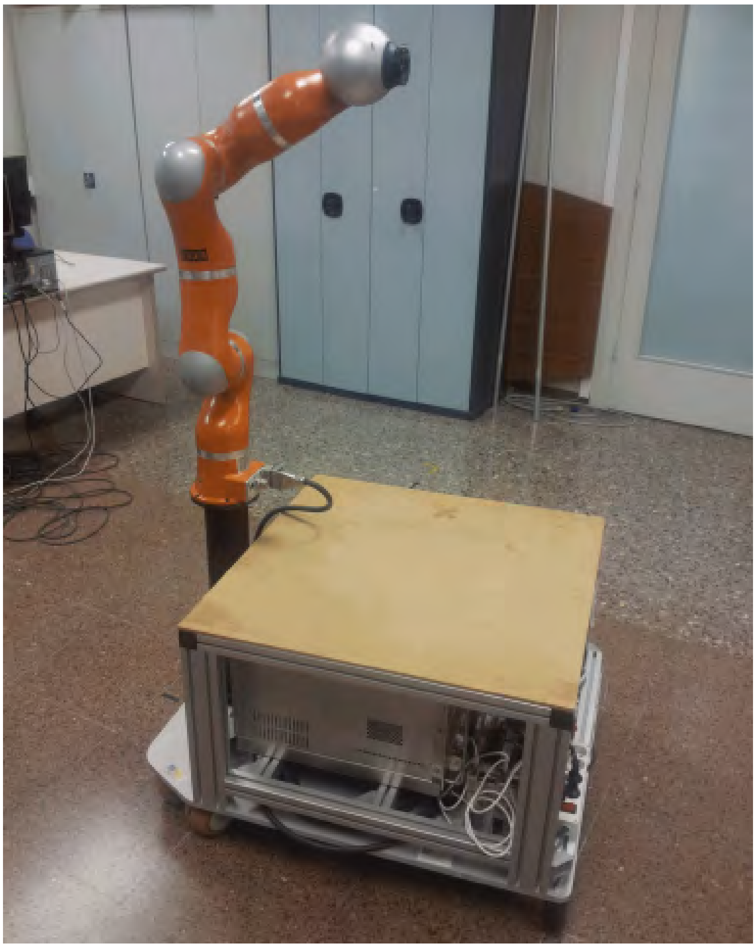
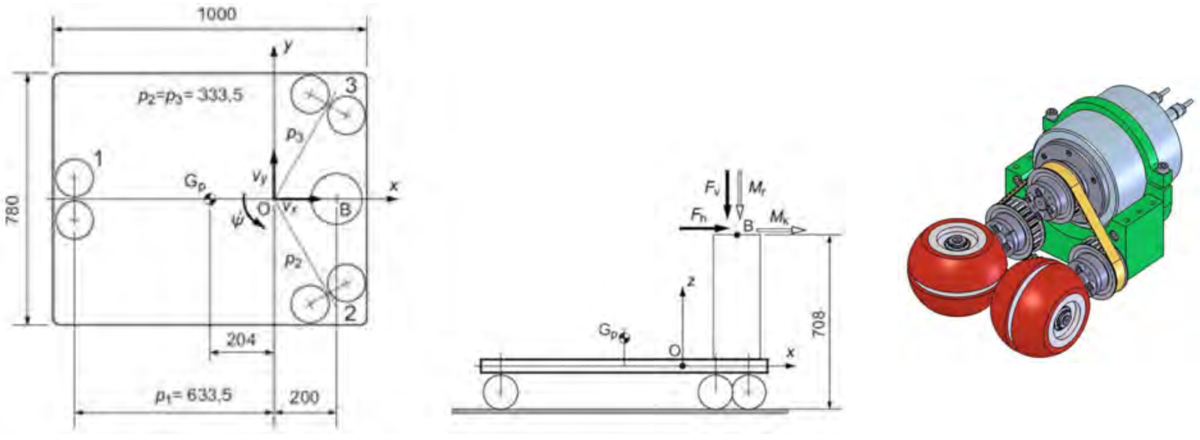


Figure 10.1 The Barcelona Mobile Manipulator.



(a) **(b)**

Figure 10.2 a) Schematic XY and XZ views of the BMM-I; b) a motor and its wheels.

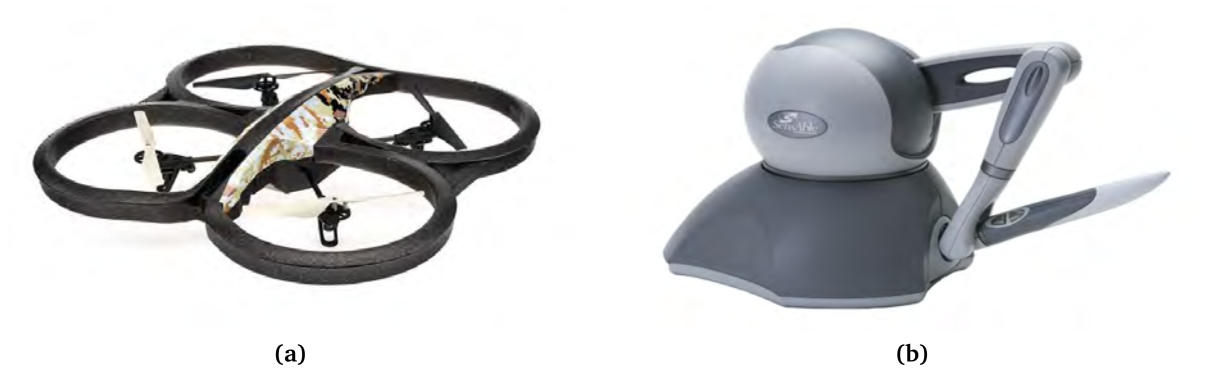


Figure 10.3 The Parrot Ardrone (Par, 2010) and the Phantom Omni haptic (Sensable, 2019).

10.2.2 The Parrot AR.Drone

The unmanned air vehicle (UAV) used in this work as the free-flying camera is the Parrot AR.Drone 2.0 (Par, 2010) (Fig. 10.3a). It is a low-cost UAV of roughly 0.5 x 0.5 meters and a weight of 420 or 380 grams, depending if it wears its outdoor hull or not.

It relies on a Linux OS over an ARM preprocessor at 1 GHz to compute its odometry through a measurement set consisting of a gyroscope, an accelerometer, a magnetometer, a pressure sensor, and a vertical camera.

It has another on-board camera at the front: a 720p HD camera which records at 30 fps. This camera allows the user to command the UAV, usually from a smartphone or a tablet, and is the camera used in this work to give feedback to the operator.

The Parrot AR.Drone communicates with an external device through Wi-Fi, acting as a router and activating a LAN network that the user has to connect to. In this work, the command of the UAV has been done through an external PC. A ROS node has been used to communicate with the UAV (Par, 2011), which allows to retrieve its odometry, get the images recorded from both cameras, do the take-off and the landing, and send linear velocity inputs (X, Y, Z translations, and the rotation around the Z axis) to the UAV.

10.2.3 The Phantom Omni

A Phantom Omni has been used as the haptic device to command the mobile manipulator.

The Phantom Omni is a portable haptic device developed by Sensable Technologies and commercialized by Geomagic, that evolved from the work of Massie and Salisbury (1994) (Fig. 10.3b). It has six sensed DOFs, and the first three DOFs are also actuated, thus allowing to provide force feedback to the user. The Phantom Omni workspace dimensions are

16 cm x 12 cm x 7 cm. It is equipped with two buttons.

The integration of the Phantom Omni is done through the Open Haptics library, which serves as the driver. This driver is encapsulated inside a ROS node (Pha, 2012) that allows the reading of the joints positions; to set torques in its first three joints, or set forces in its tip referenced in the Cartesian space; and the sensing of the two buttons.

10.3 Integration

Several components are needed to implement the teleoperation robotic system. These components include the drivers to communicate and control the devices and robots; but also components dedicated to intermediate calculations, for instance, of the algorithms presented in the previous chapters; and components responsible for the coordination of the different parts of the system.

The programming language used for the implementation has been C++. The code has been parametrized such that, by setting a single variable to *true* or *false*, it can be chosen if the robots will be moved in the simulation environment or in a real scenario.

Since the system is composed of multiple robots, and the components are executed in different machines, a communication layer is necessary. In that sense, ROS (Quigley et al., 2009) has been chosen as the middleware to communicate the components. Figure 10.4 shows a schema with the components, which are explained in detail below:

- **Haptic Controller**

  This component allows the reading of the current values of the Phantom Omni haptic device, and the input of new desired values. It is a wrapper around the Open Haptics Phantom Omni driver (Pha, 2012).

  It contains an asynchronous thread responsible of reading the joint values, the tip position and orientation w.r.t. the base, its velocity, and the state of the buttons. This node also allows to impose desired torques on the first three joints or desired Cartesian forces at the tip of the Phantom Omni.

- **Haptic Manager**

  This component adds some extra functionalities to the Phantom Omni by extending the *Phantom Omni Controller* node. One of this functions is the possibility to lock the omni in its current configuration when both buttons are simultaneously pressed.

  This node also handles the initial state of the Phantom Omni when the robotic system is initialized. It does so by locking the tip of the haptic device at a predefined initial pose, ready to be unlocked and used by the operator.

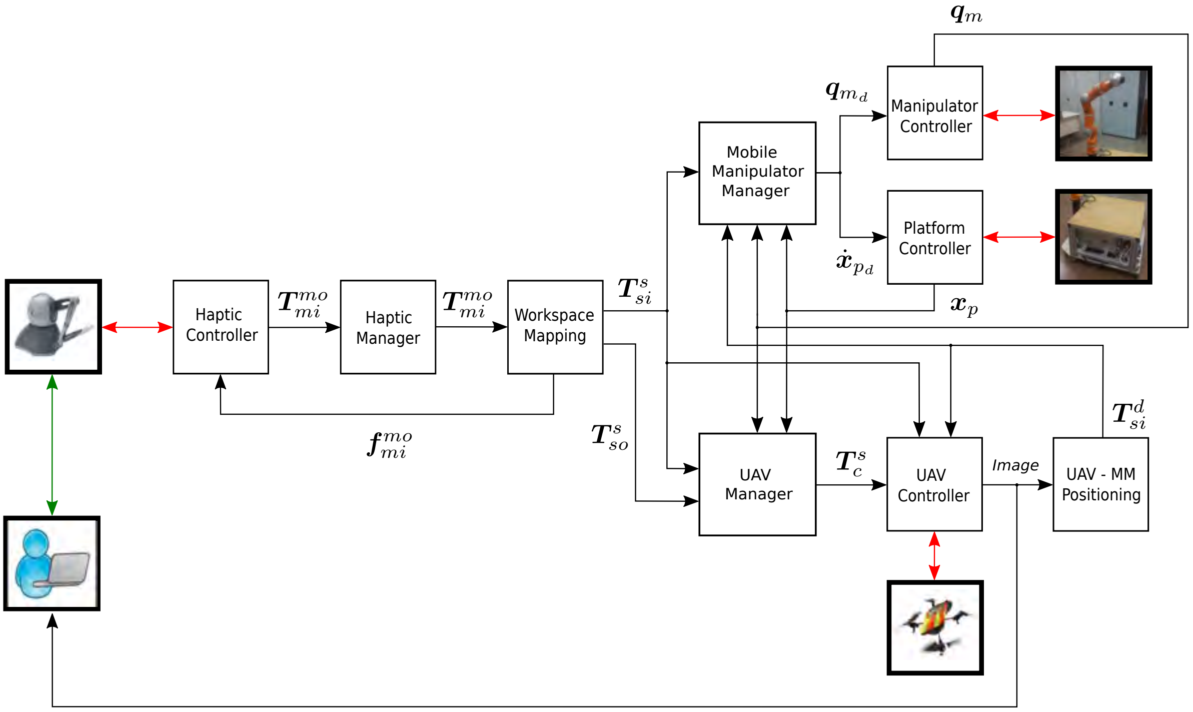


Figure 10.4 The components of the teleoperation robotic system. The communications between the devices and their driver components are depicted in red; and the input of the user on the haptic device, in green. x_{p_d} and q_{m_d} correspond to the desired configuration vectors of the platform and the manipulator, x_p and q_m, respectively, as explained in Sect. 6.2.

- **Workspace Mapping**

  This node contains the algorithms presented in Chapter 6, that is, the computations to transform the input from the user, *i.e.*, a position and orientation of the Phantom Omni tip frame in the haptic device workspace, T_{mi}^{mo}, to the haptic subspace counterpart pose in the remote scenario, T_{so}^{s} in Eq. 6.6, and the mobile manipulator end-effector desired pose in the scenario, T_{si}^{s} in Eq. 6.7.

  This component also computes the force f_{mi}^{mo} (Eq. 6.8) that alerts the user that the tip of the haptic device is operating near the boundaries of the haptic position workspace, thus triggering a gross motion of the mobile manipulator, and further sends this signal to the Phantom Omni driver.

- **Mobile Manipulator Manager**

  This is the main BMM-I node. It takes the desired end-effector pose, T_{si}^{s}; computes the next desired BMM-I state q_m and x_p; and shares them out to their corresponding drivers.

  The computation of the continuous inverse task priority formulation with two levels of priority, explained in Section 7.2.3, is contained in this node. In the first (highest) priority level, a joint limit avoidance task has been implemented, and, in the second, the tracking of the desired end-effector pose. This node also contains the logic to compute the OBVF presented in Chapter 8, which can be activated at the third level of priority.

It also contains initialization logic: the node waits for the first user input through the haptic device and the first robot joint values, moves the robot to a predefined initial configuration, publishes its state, and waits for the UAV to be flying and ready for the teleoperation.

This node also contains security features, like boundaries for the BMM-I workspace, and for the manipulator end-effector workspace so that it does not collide with the platform.

- **Platform Controller**

  This component is used to command the BMM-I platform. It runs in the PC in the BMM-I. It is a wrapper to a library that transforms the desired linear and angular velocities sent as inputs to the BMM-I, $\dot{\boldsymbol{x}}_{p_d}$, to the velocities for the wheel motors.

- **Manipulator Controller**

  This component sends the desired KUKA LWR joint values, $\boldsymbol{q}_{m_d}$, to the FRI library, which in turns sends it to the KUKA LWR controller. It runs in the PC of the BMM-I.

  Once initialized, it sends the current configuration of the KUKA LWR, $\boldsymbol{q}_m$, to other nodes so that the teleoperation can start.

- **Unmanned Air Vehicle Manager**

  The main purpose of this node is to compute the next desired camera pose, $\boldsymbol{T}_c^s$, as detailed in Chapter 9, given $\boldsymbol{T}_{so}^s$, $\boldsymbol{T}_{si}^s$ and the current configuration of the robot $\boldsymbol{q} = [\boldsymbol{x}_p \quad \boldsymbol{q}_m]^T$.

  It also contains the initialization procedure: it waits for the first user input through the haptic and for the BMM-I to be in its initialization pose. It also triggers the UAV taking-off, waits until the mobile manipulator is detected in its images, and starts the teleoperation.

- **Unmanned Air Vehicle Controller**

  This node contains the logic to coordinate the UAV with the other nodes of the system. It communicates with the *ardrone_autonomy* node (Par, 2011), which is a ROS driver for the Parrot AR-Drone 1.0 and 2.0 based on the official AR-Drone SDK (version 2.0.1).

  This node coordinates the UAV take-off and landing, and waits for the first MM detection in the UAV camera images before being ready to send new inputs for the UAV.

  The new inputs for the UAV, $\boldsymbol{u}_d$, are generated from the current pose of the end-effector w.r.t. the UAV frame, $\boldsymbol{T}_{si}^d$, and the desired camera pose defined by $\boldsymbol{T}_c^s$, that allows the computation of the desired end-effector pose w.r.t. the UAV frame as

  $$\boldsymbol{T}_{si_d}^d = \boldsymbol{T}_c^d \, (\boldsymbol{T}_c^s)^{-1} \, \boldsymbol{T}_{si}^s \tag{10.1}$$

  with the frame d being a frame attached to the UAV, and $\boldsymbol{T}_c^d$ being the constant transformation between the UAV frame and the camera frame. This allows the computation of the new desired UAV velocity:

  $$\boldsymbol{u}_d = -\begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}_z \, \boldsymbol{p}_{si}^d \\ \boldsymbol{0}^T & 1 \end{bmatrix} \boldsymbol{K}_d \, \boldsymbol{\varepsilon}_{si}^d - \boldsymbol{K}_m \dot{\boldsymbol{x}}_d \tag{10.2}$$

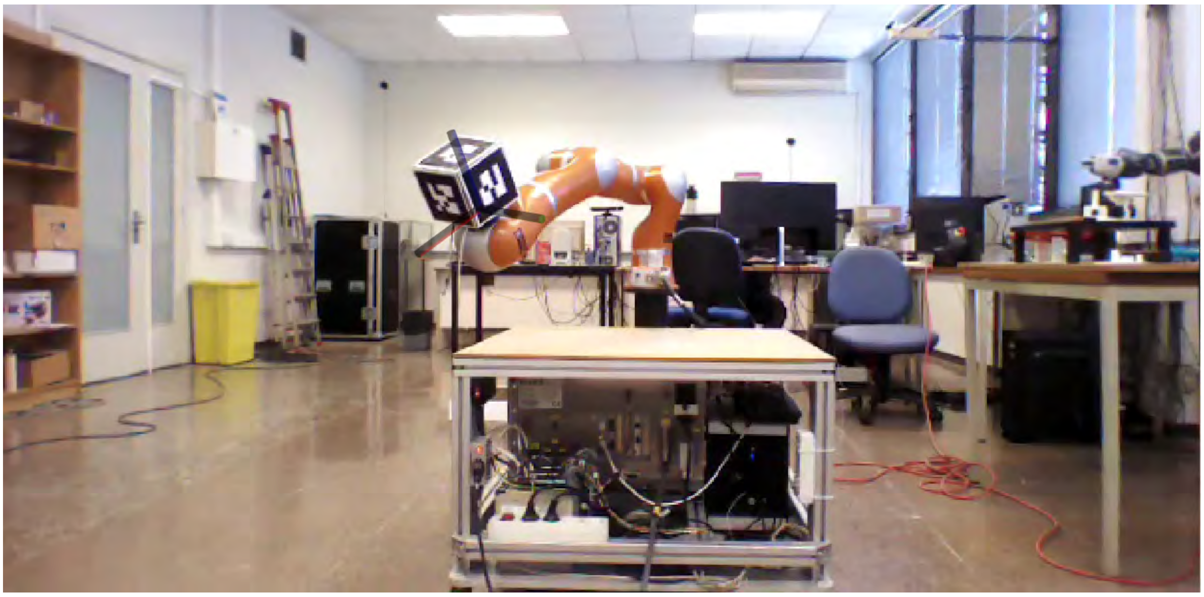


Figure 10.5 Alvar tags at the robot end-effector and frame $\boldsymbol{T}^{c}_{si}$.

with:

- $\boldsymbol{\varepsilon}^{d}_{si} = \begin{bmatrix} \varepsilon^{d}_{si_x} & \varepsilon^{d}_{si_y} & \varepsilon^{d}_{si_z} & \varepsilon^{d}_{si_\theta} \end{bmatrix}^{T}$: the error between the current and the desired poses of the transformation between the UAV and the mobile manipulator end-effector frames in the X, Y, Z dimensions and the rotation around the Z axis (θ).
- $\dot{\boldsymbol{x}}_{d} = \begin{bmatrix} \dot{p}_{d_x} & \dot{p}_{d_y} & \dot{p}_{d_z} & \omega_{d_\theta} \end{bmatrix}^{T}$: the current velocity of the UAV expressed in the frame d, used to add a damping to the expression to compute $\boldsymbol{u}_{d}$.
- $\boldsymbol{K}_{d}$, $\boldsymbol{K}_{m}$: two positive diagonal constant matrices.
- $\boldsymbol{S}_{z}$: the cross product matrix evaluated at the Z axis unit vector,

$$\boldsymbol{S}_{z} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

- **Unmanned Air Vehicle - Mobile Manipulator Positioning**

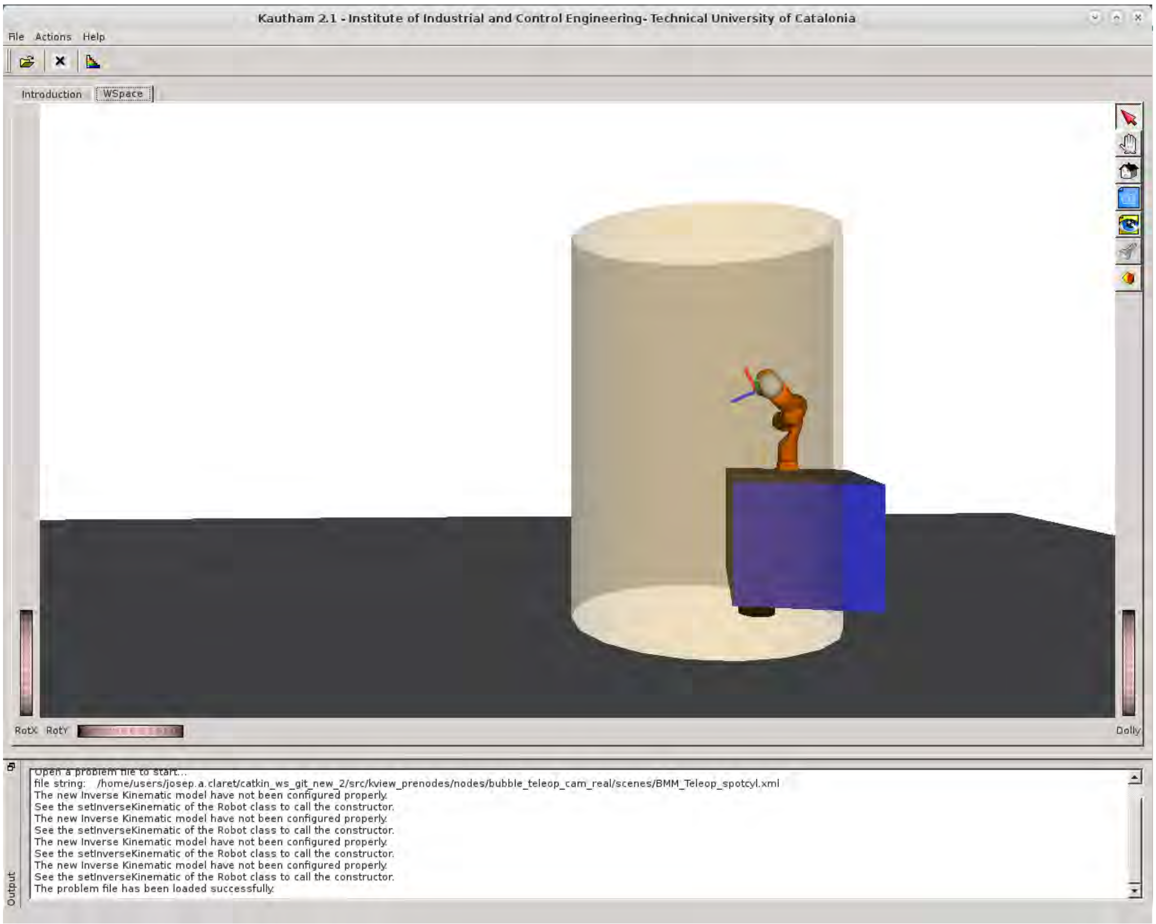
  This node determines the relative pose between the UAV and the mobile manipulator, and publishes this information to the whole system through the matrix $\boldsymbol{T}^{d}_{si}$.

  This relative positioning is done through a set of marks located at a know position w.r.t. the robot. For testing purposes, they have been located at the robot end-effector.

  During the teleoperation, these marks are recognized and located w.r.t. the camera, thus obtaining the transformation $\boldsymbol{T}^{c}_{si}$. It is then straightforward to compute

$$\boldsymbol{T}^{d}_{si} = \boldsymbol{T}^{d}_{c}\boldsymbol{T}^{c}_{si}. \tag{10.3}$$

  The marks are arranged in a bundle of tags (Fig. 10.5) using the Alvar library (VTT, 2016), which can be located through a camera using the *ar_track_alvar* ROS node (Alv, 2012).


Kautham 2.1 - Institute of Industrial and Control Engineering- Technical University of Catalonia
File
Actions
Help
Introduction
WSpace
RotX
RotY
Dolly
Output

```
Open a problem file to start...
file string:   /home/users/josep.a.claret/catkin_ws_git_new_2/src/kview_prenodes/nodes/bubble_teleop_cam_real/scenes/BMM_Teleop_spotcyl.xml
The new Inverse Kinematic model have not been configured properly.
See the setInverseKinematic of the Robot class to call the constructor.
The new Inverse Kinematic model have not been configured properly.
See the setInverseKinematic of the Robot class to call the constructor.
The new Inverse Kinematic model have not been configured properly.
See the setInverseKinematic of the Robot class to call the constructor.
The new Inverse Kinematic model have not been configured properly.
See the setInverseKinematic of the Robot class to call the constructor.
The problem file has been loaded successfully.
```

Figure 10.6 A view of the Kautham Project with the simulated BMM-I.

10.4 Simulation environment: The Kautham Project

The Kautham Project has been the simulation environment tool used (Fig. 10.6). It is a software tool written in C++ for teaching and research in robot motion planning. Along with its own planners, it uses planners provided by the Open Motion Planning Library (Şucan et al., 2012). It allows scenarios with multiple robots, from simple two-DOFs linear robots to long serial-chain robots. It implements collision checking between robots and objects in the scene and the visualization of the collision objects, and an inverse kinematics module for a predefined set of robots. It can be executed as a ROS node.

The Kautham software has been used here as a ROS node, easing the integration with the rest of the components of the robotic system. It has also been modified in order to add two functionalities relevant to this work:

- Allowing real-time feeding of external signals through ROS topics, since by default the

Kautham software works with ROS services.

- Dynamic position of the camera in the scenario. This is a critical aspect in this work, since the main camera of the Kautham is the UAV camera.

10.5 Experimentation

A set of four experiments with the real setup have been done in order to test the robotic system introduced in this dissertation.

10.5.1 BMM-I Inverse Kinematics Test

A video of the first experiment can be found on the link *BMM-I Inverse kinematics*. This video shows the behavior of the BMM-I when a desired pose is imposed to its end-effector, both in position and orientation (Fig. 10.7). The desired position can be seen as a red sphere in the simulation environment. The inverse kinematics algorithm executed during this motion to obtain the joint values corresponds to Eq. 6.5.

During the first 10 seconds, the motion of the BMM-I can be seen in the simulation environment, and, after, both the simulated and the real BMM-I are shown. Note that until the second 18 the platform is locked and only the manipulator is moving and tracking the desired pose. From that moment, the platform is also moving, and the 10 DOFs of the system are used to solve the inverse kinematics.

10.5.2 Object Best View Function Experiment 1

Two experiments have been executed to test the Object Best View Function (OBVF) presented in Chapter 8.

The first video corresponds to a the real setup with a motion corresponding to the Simulation 1. It can be found on the link *Best-view Simulation 1 real setup*. In this motion, a trajectory is imposed to the end-effector such that it gets behind the robot links w.r.t. the camera, thus getting occluded.

The video is divided in four areas (Fig. 10.8): the left figures correspond to the motions without activation of the OBVF, and the right figures show the motions with the OBVF activation; while the top figures correspond to the motions in the simulation environments, and, the bottom figures, in the real setup. Also note that the free-flying camera is fixed, since the motion of the camera does not affect the performance of the null space motions.

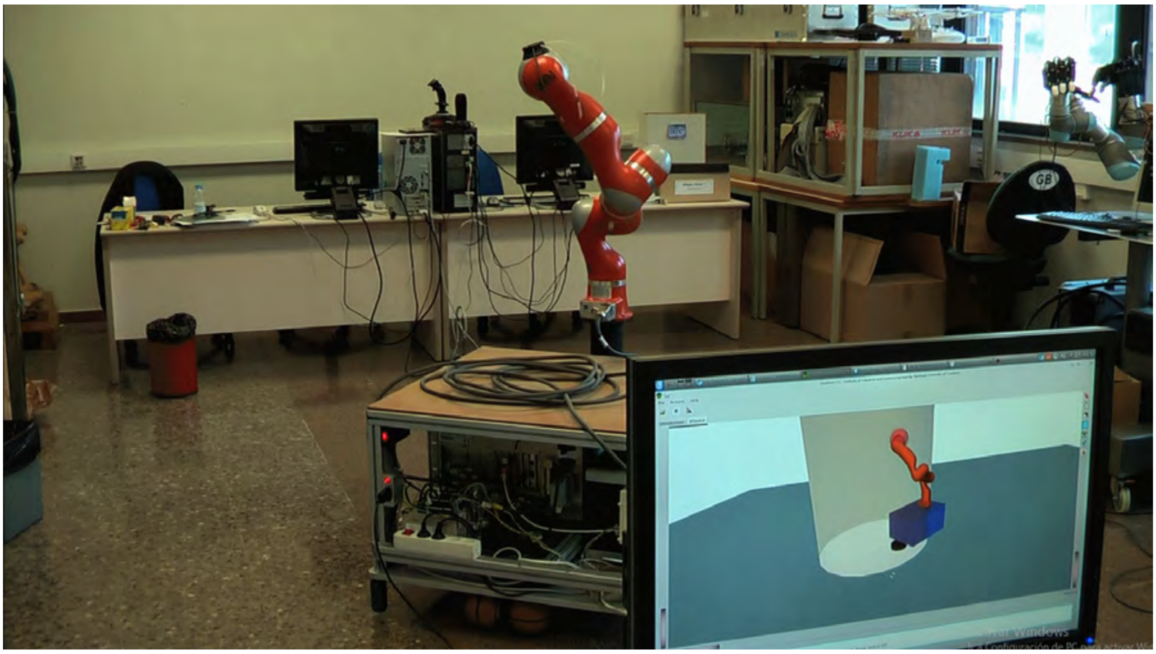


Figure 10.7 Snapshot of the BMM-I inverse kinematics test.

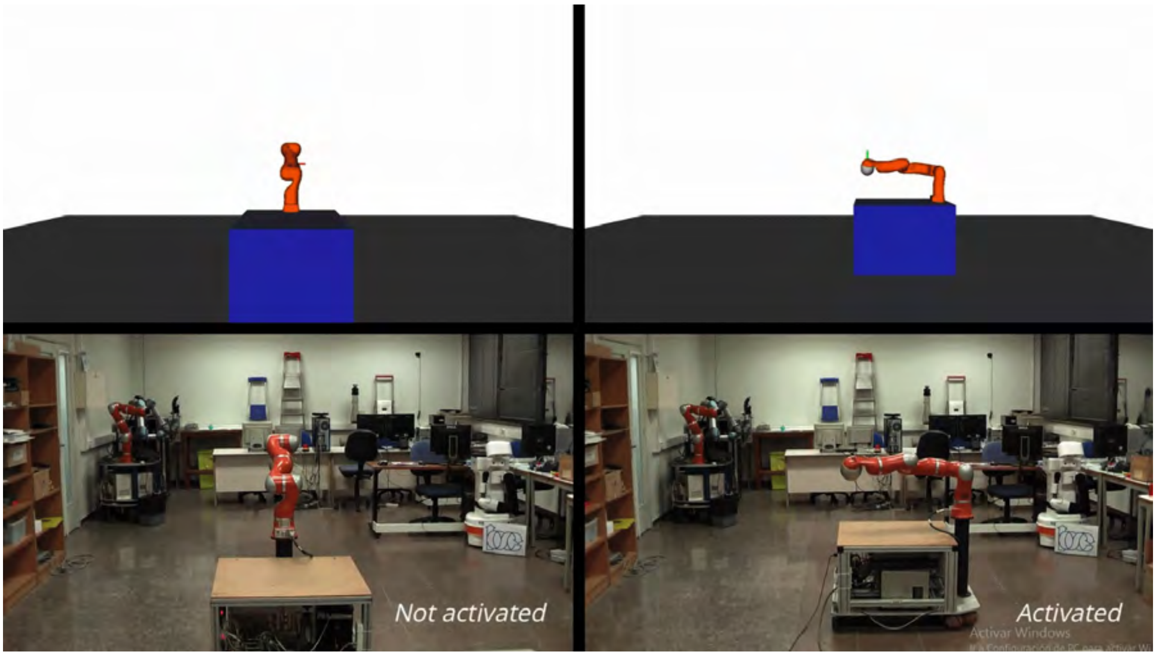


Figure 10.8 Snapshot of the OBVF experiment 1. The left figures correspond to the motions without activation of the OBVF, and the right figures, with the OBVF activated. The top figures correspond to the simulation environment, and, the bottom figures, in the real setup.

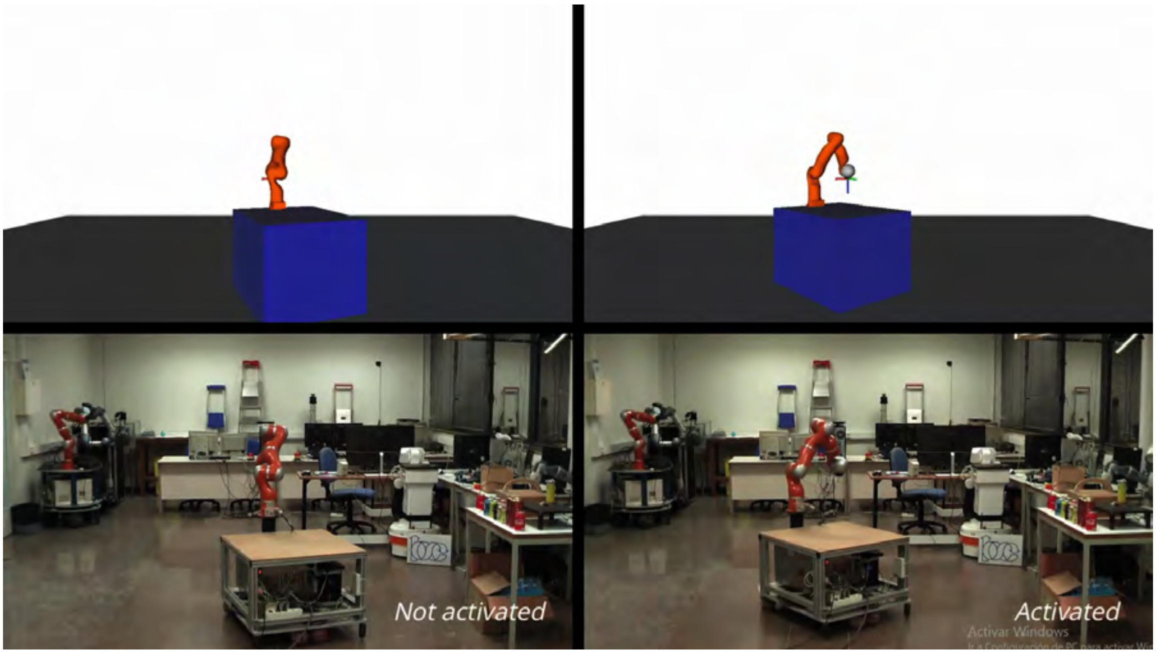


Figure 10.9 Snapshot of the OBVF experiment 2. The left figures correspond to the motions without activation of the OBVF, and the right figures, with the OBVF activated. The top figures correspond to the simulation environment, and, the bottom figures, in the real setup.

10.5.3 Object Best View Function Experiment 2

A video with the testing of the OBVF in the real setup and with a motion corresponding to the Simulation 2 can be found on the link *Best-view Simulation 2 real setup*.

The motion shown in this motion is a translation of the BMM-I from right to left such that the end-effector gets occluded by the links of the robot.

Like the previous one, the video is divided in four areas (Fig. 10.9): the left figures correspond to the motions without activation of the OBVF, and the right figures show the motions with the OBVF activation; while the top figures correspond to the motions in the simulation environments, and, the bottom figures, in the real setup. The free-flying camera is also fixed.

10.5.4 Drone Track Experiment

A video of an experiment where the whole robotic system is shown can be downloaded from the link *BMM-I Teleoperation*.

In this experiment, the BMM-I is about to leave the frustum of the camera pyramid of vision in two occasions: first, by approaching the left face of the frustum, and, after, the furthest face

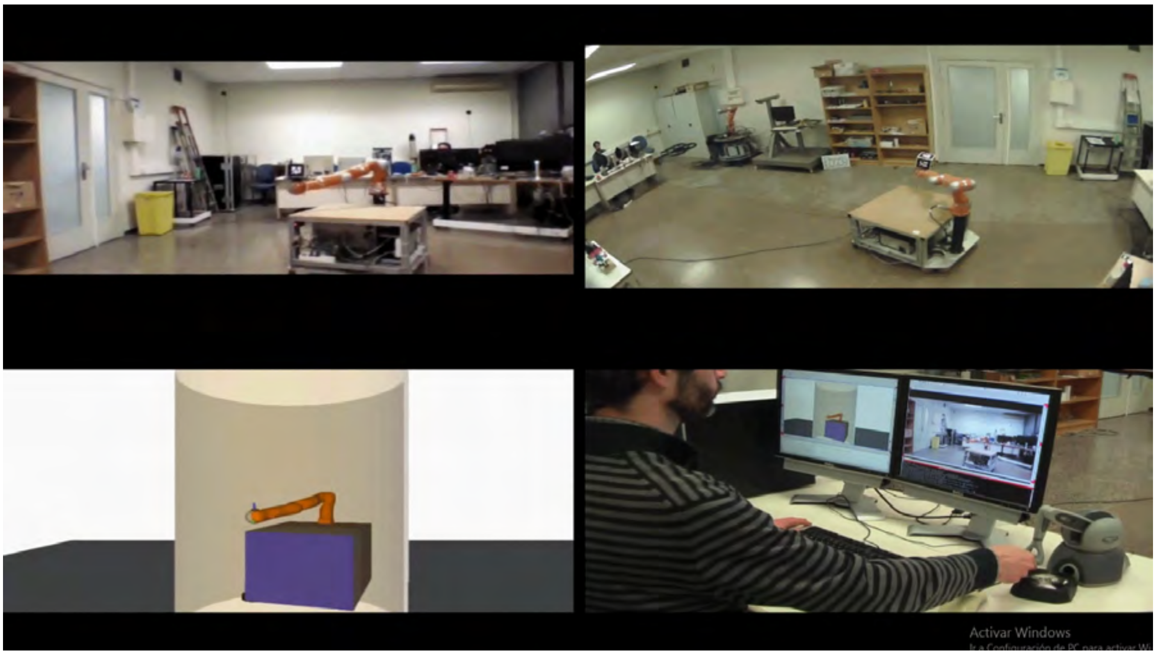


Figure 10.10 Snapshot of the BMM-I teleoperation experiment using the image of the camera UAV as visual feedback. Top left: the image from the UAV camera which is the image seen by the operator. Bottom left: the motion in the simulation environment. Top right: a general view of the workspace. Bottom right: the user executing the motion using the haptic device while observing the image from the free-flying camera.

of the frustum w.r.t. the camera. In both cases, a motion is triggered on the UAV such that the projection of the mobile manipulator in the camera image plane is located at its center.

The video shows the recordings of four cameras (Fig. 10.10): the image from the UAV camera which is the image seen by the operator (top left area), the motion in the simulation environment (bottom left), a general view of the workspace (top right), and the user executing the motion using the haptic device while observing the image from the free-flying camera (bottom right).

The first translation of the robot to the left starts roughly after 9 seconds and the end-effector reaches the boundaries of the pyramid of vision at second 18. At that moment, the camera motion is triggered, and the UAV moves in order to locate the robot in the middle of the field of view, which takes place at second 23.

A second translation of the BMM-I is triggered by the user at second 32. In this new motion, the robot moves away from the camera in the direction of the optical axis for around 9 seconds, at which moment, again, the convergence of the UAV towards the origin of the haptic subspace in the remote scenario is activated, finishing at second 46.

Chapter 11

Conclusions

Caminante, no hay camino,
se hace camino al andar.

Antonio Machado

SUMMARY: This chapter presents both the general and particular conclusions of the present dissertation, along with the future work and the related list of publications.

11.1 Contributions

The main general contributions of this thesis are the following:

- A robotic system which, to the author's knowledge, is proposed and studied for the first time. This new robotic system, whose main application is real-time teleoperation, consists of a mobile manipulator, a haptic device and a UAV equiped with a free-flying camera. The kinematic analysis of this particular robotic system has been presented with a study of its viability and its implementation.
- Two new uses of the robot null space that allow the robot to execute a primary task and use the remaining degrees of freedom of the system to execute secondary tasks have been proposed:
  - The first use of the null space permits the conveyance of emotional information from a humanoid robot to a user, allowing the robot to execute a main task while exploiting its redundancy to convey the desired emotions.

  - The second proposed null space algorithm has been implemented on the haptic-UAV-mobile manipulator robotic system presented in the current dissertation and it minimizes the occlusions of the robot end-effector to the operator, making easy the system teleoperation.

Regarding the use of the redundancy of a humanoid robot to convey emotions to the user, which corresponds to the Part I of this dissertation, the particular contributions are:

- The first approach, to the author's knowledge, for a humanoid robot to simultaneously execute a kinematic task and convey emotions to the users with its body motions. To do so, a task priority null space approach that conveys emotions to the users as a secondary task has been developed. The approach allows the specification of a desired primary task in the Cartesian space and the use of the remaining DOFs to convey a desired emotion to the user through the robot body motions.

- A novel map that transforms emotional information to kinematic features of the robot. The proposed model builds up on previous studies in which the emotions are defined as points in a *pleasure-arousal-dominance* space. Further, the model proposes a transformation from this emotion space to the kinematic features *jerkiness*, *activity*, and *gaze* directness. Ultimately, the presented map is a necessary step that allows the robot programmer to feed the null space of the robot with the desired emotion state.

- An analysis of the interactions between motion features of the robot and the emotions conveyed to the user. For this analysis, a user study has been conducted along with an evaluation of the interactions between the robot emotional kinematic features and the *pleasure-arousal-dominance* dimensions of the emotions perceived by the user. The results indicate that the null space is a promising mean to convey emotions, and that the users do perceive emotions on the robot even when the robot is executing a primary task. Particularly, the next conclusions can be drawn from the study:

  - *Happiness* and *sadness* can be successfully conveyed.
  - There exists a positive correlation between *activity* and *arousal*.
  - There is a non-statistically significant perception of *jerkiness*.
  - The *dominance* dimension is well conveyed when *activity* is low and *gaze* is directed towards the user.
  - The Pepper robot is perceived as non dominant.

  Thus, this dissertation presents the first evidence that the null space is a promising mean to convey the emotions to the users as a lower priority task.

- One of the first works using a Pepper robot, although the proposed approach is not only valid for this robot, but easily extendible to other humanoid robots.

Concerning the teleoperation robotic system presented in Part II consisting of a haptic device, a mobile manipulator and a free-flying camera on a UAV, the particular contributions are:

- A novel teleoperation system that permits the simultaneous command of a ground mobile manipulator and a free flying camera using a single haptic device.
- A workspace mapping algorithm to command a mobile manipulator in an unbounded workspace using a single haptic device. The proposed solution allows the automatic coordination of the UAV camera motion (both with translation and rotation) with the mobile manipulator, which in turn allows the operator to focus on the teleoperation of the robot through the haptic while allowing the algorithm to deal with the free-flying camera motion, and while keeping the robot in its point of view. The results of a user study show that training increases the operator performance and its dexterity in the teleoperation over time, while also lowering the workload demand on the operator, thus making the proposed teleoperation approach suitable for its purpose.
- An enhancement of the null space projector of the continuous inverse for lower priority tasks. This enhancement ensures the stability of the algorithm when using tasks with priority level three or lower.
- A second adaptation of the continuous inverse that ensures that lower priority tasks do not affect the tasks of the higher priority for long tasks transitions. This allows the algorithm to comply with one of its most important objectives, by solving one of its main drawbacks, that is, the impossibility to guarantee the predefined priorities during transitions. The proposed solution does so while fulfilling the criteria for continuous tasks transitions.
- A novel use of the robot null space to prevent occlusions of the end-effector by the robot itself as seen by a camera, that is, to avoid that the structure of the robot intersects the line of sight between the end-effector and the free-flying camera.
- An algorithm that activates the motion of the camera in a way that guarantees that the robot is always within the field of view of the camera, but only triggering its motion when the robot is approaching the boundaries of the field of view.

11.2 Future work

During the realization of this doctoral work, several interesting directions to follow in future research have emerged.

With respect to the emotion conveyance part of this thesis, further extensions of the proposed approach aim to close the gap between the current implementation presented in this work, and the implementation in real world applications. In this regard, new studies on other robots with more complex main tasks, like manipulation of an object, or carrying a load cooperatively with a user, are interesting future experiments.

It would be interesting to develop the user study in other countries in order to measure the cultural bias that might be present in the current results. This would allow to assess whether it would be necessary to have different mappings from emotions to kinematics features depending on the cultural background of the user interacting with the robot.

The adaptation of the proposed approach to industrial robots can also be an interesting research topic. A more efficient criteria for parameter tuning is desirable, too, and more user studies are necessary to assess the usability and comfortability of the user when he or she is co-working or cohabitating with a robot. Future studies might evaluate the usability of the framework for users with intellectual disabilities or autism.

Future work should also include a deeper understanding on the role of *jerkiness* and a more complex map between the PAD space and the kinematic features, which could be addressed using a learning approach. In this regard, it could also be interesting to create a dynamic mapping, which could increase in complexity (for example, incorporating more kinematic features) as the number of available redundant DOFs increases.

Finally, a very appealing work that would condense all the presented contributions and bring them to a higher level would be to not only convey emotions to the user but rather to influence the emotional state of the user in a controlled manner. To do so, a first step would be to measure the user emotional state and, given the desired emotion for the user, execute the corresponding motions. A feedback loop (probably visual or auditive) would be required to update the emotional state of the user, so as to modify the desired emotion to convey, in what could be seen as user *emotion control* by a robot.

Regarding the teleoperation system consisting of a mobile manipulator, a UAV and a haptic device, future work contemplates taking into consideration the zoom of the camera, which would add more DOFs to the system. This would allow the camera to approach or move away from the robot while adjusting the zoom to fake as if the UAV was not moving. This extra DOF could then be used, for instance, for obstacle avoidance while still keeping the camera focused on the robot.

Another problem to be addressed as future work could be to improve the computation of the relative pose between the UAV and the mobile manipulator through the free-flying camera images in the *UAV-MM Positioning* component of the robotic system. One way to address this would be to compare a three dimensional model of the mobile manipulator with the image, and use the known joint configuration of the manipulator to infer the relative transformation.

Finally, it would also be interesting to generalize the OBVF algorithm to prevent self-occlusions of the end-effector by the robot to other type of robots, not only to mobile manipulators with a seven-DOFs arm.

11.3 Publications

The following list contains the published and submitted papers generated during the course of this thesis.

Journals

- Claret, J.A., Venture, G. and Basañez, L. Exploiting the robot kinematic redundancy for emotion conveyance to humans as a lower priority task. *International Journal of Social Robotics*, volume 9(2), pages 277-292, January 2017. ISSN 1875-4805.
  DOI 10.1007/s12369-016-0387-2.
- Zaplana, I.; Claret, J.A.; Basañez, L. Análisis Cinemático de Robots Manipuladores Redundantes: Aplicación a los Robots Kuka LWR 4+ y ABB Yumi. *Revista Iberoamericana de Automática e Informática industrial*, [S.l.], November 2017. ISSN 1697-7920.
  DOI 10.4995/riai.2017.8822.
- Claret, J.A. and Basañez, L. Teleoperating a mobile manipulator with a free-flying camera: challenges and solutions for a new robotic system. *Journal of Field Robotics. Special Issue on Future Challenges and Opportunities in Vision-based Drone Navigation*, May 2019. *Submitted*

Book Chapters

- Dubois, M., Claret, J.A., Basañez, L. and Venture, G. Influence of Emotional Motions in Human-Robot Interactions, *2016 International Symposium on Experimental Robotics*, pages 799-808, 2017, Springer International Publishing. ISBN 978-3-319-50115-4. DOI 10.1007/978-3-319-50115-4_69.

Conferences

- Claret, J.A., Zaplana, I. and Basañez, L. Teleoperating a mobile manipulator and a free-flying camera from a single haptic device, *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Lausanne, Switzerland, pages 291-296, October 2016. DOI 10.1109/SSRR.2016.7784318.
- Claret, J.A. and Basañez, L. Using an UAV to guide the teleoperation of a mobile manipulator, *XXXVIII Jornadas de Automática*, Gijón, Spain, September 2017, Universidad de Oviedo. ISBN: 978-84-16664-74-0, pages 694-700.

- Claret, J.A. and Basañez, L. Teleoperating a mobile manipulator with a free-flying camera without robot-own occlusions, *XL Jornadas de Automática*, Ferrol, Spain, September 2019, Universidade da Coruña. *Submitted*

Appendix A

The Emotional Conveyance Algorithm: Task Priority Proof

Following the work of Chiaverini (1997) and Baerlocher and Boulic (1998), a proof of the task prioritization of the proposed solution, that is, that the execution of the lower priority tasks does not affect the execution of the higher priority tasks, is shown below.

The next identity will be used:

$$\boldsymbol{A}\,\boldsymbol{A}^{+}\,\boldsymbol{A} = \boldsymbol{A} \tag{A.1}$$

where $\boldsymbol{A}^{+}$ is the pseudoinverse of $\boldsymbol{A}$.

Given an idempotent matrix $\boldsymbol{B}$, that is, $\boldsymbol{B} = \boldsymbol{B}^2$; and Hermitian, $\boldsymbol{B} = \boldsymbol{B}^*$ in general, with $\boldsymbol{B}^*$ the conjugate of $\boldsymbol{B}$, and, in particular for this proof, $\boldsymbol{B} = \boldsymbol{B}^T$ then

$$\boldsymbol{B}\,(\boldsymbol{AB})^{+} = (\boldsymbol{AB})^{+}. \tag{A.2}$$

Given a matrix $\boldsymbol{C}$, then $\boldsymbol{D} = \boldsymbol{I} - \boldsymbol{C}^{+}\boldsymbol{C}$ is the orthogonal projector onto the kernel of $\boldsymbol{C}$, thus idempotent and Hermitian. So, in light of Eq. A.2,

$$(\boldsymbol{AD})^{+} = \boldsymbol{D}(\boldsymbol{AD})^{+}. \tag{A.3}$$

Given a task $\boldsymbol{x}_i$ of the robot defined as a function of its configuration $\boldsymbol{q}$ through its direct kinematics

$$\boldsymbol{x}_i = \boldsymbol{k}_i(\boldsymbol{q}),$$

differentiating and applying the chain rule

$$\dot{\boldsymbol{x}}_i = \frac{\partial \boldsymbol{x}_i}{\partial t} = \frac{\partial \boldsymbol{k}_i(\boldsymbol{q})}{\partial t} = \frac{\partial \boldsymbol{k}_i(\boldsymbol{q})}{\partial \boldsymbol{q}}\frac{\partial \boldsymbol{q}}{\partial t} = \boldsymbol{J}_i\,\dot{\boldsymbol{q}},$$

a mapping between the velocity of task i and the joint velocities is obtained.

Similarly, the differential mapping of the main task t becomes $\dot{\boldsymbol{x}}_t = \boldsymbol{J}_t\,\dot{\boldsymbol{q}}$. The joint velocities $\dot{\boldsymbol{q}}$ as defined in Eq. 4.8 are

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_t^{+}\boldsymbol{e}_t + \boldsymbol{N}_t\,\boldsymbol{J}_h^{+}\boldsymbol{e}_h + \big(\boldsymbol{N}_t - (\boldsymbol{J}_h\boldsymbol{N}_t)^{+}(\boldsymbol{J}_h\boldsymbol{N}_t)\big)\,\boldsymbol{e}_m, \tag{A.4}$$

thus, by substituting Eq. A.4 in $\dot{\boldsymbol{x}}_t = \boldsymbol{J}_t\,\dot{\boldsymbol{q}}$, it is obtained

$$\dot{\boldsymbol{x}}_t = \boldsymbol{J}_t\,\left(\boldsymbol{J}_t^{+}\boldsymbol{e}_t + \boldsymbol{N}_t\,\boldsymbol{J}_h^{+}\boldsymbol{e}_h + \big(\boldsymbol{N}_t - (\boldsymbol{J}_h\boldsymbol{N}_t)^{+}(\boldsymbol{J}_h\boldsymbol{N}_t)\big)\,\boldsymbol{e}_m\right).$$

Using Eq. A.3 in $(\boldsymbol{J}_h\boldsymbol{N}_t)^{+}$ and rearranging terms the expression can be transformed to

$$\dot{\boldsymbol{x}}_t = \boldsymbol{J}_t\,\boldsymbol{J}_t^{+}\boldsymbol{e}_t + \boldsymbol{J}_t\,\boldsymbol{N}_t\,\left(\boldsymbol{J}_h^{+}\boldsymbol{e}_h + \big(\boldsymbol{I} - (\boldsymbol{J}_h\boldsymbol{N}_t)^{+}(\boldsymbol{J}_h\boldsymbol{N}_t)\big)\,\boldsymbol{e}_m\right)$$

and now, using Eq. A.1 in $\boldsymbol{J}_t\,\boldsymbol{N}_t$ as

$$\boldsymbol{J}_t\,\boldsymbol{N}_t = \boldsymbol{J}_t\left(\boldsymbol{I} - \boldsymbol{J}_t^{+}\boldsymbol{J}_t\right) = \boldsymbol{J}_t - \boldsymbol{J}_t\boldsymbol{J}_t^{+}\boldsymbol{J}_t = \boldsymbol{J}_t - \boldsymbol{J}_t = \boldsymbol{0},$$

becomes $\dot{\boldsymbol{x}}_t = \boldsymbol{J}_t\,\boldsymbol{J}_t^{+}\boldsymbol{e}_t$. This expression shows that the emotional tasks h and m do not affect the execution of the main task t.

Similarly for the task h corresponding to the *gaze*:

$$\begin{aligned}\dot{\boldsymbol{x}}_h &= \boldsymbol{J}_h\,\left(\boldsymbol{J}_t^{+}\boldsymbol{e}_t + \boldsymbol{N}_t\,\boldsymbol{J}_h^{+}\boldsymbol{e}_h + \big(\boldsymbol{N}_t - (\boldsymbol{J}_h\boldsymbol{N}_t)^{+}(\boldsymbol{J}_h\boldsymbol{N}_t)\big)\,\boldsymbol{e}_m\right)\\ &= \boldsymbol{J}_h\,\boldsymbol{J}_t^{+}\boldsymbol{e}_t + \boldsymbol{J}_h\,\boldsymbol{N}_t\,\boldsymbol{J}_h^{+}\boldsymbol{e}_h + \boldsymbol{J}_h\,\big(\boldsymbol{N}_t - (\boldsymbol{J}_h\boldsymbol{N}_t)^{+}(\boldsymbol{J}_h\boldsymbol{N}_t)\big)\,\boldsymbol{e}_m\end{aligned}$$

which, using Eqs. A.2 and A.1,

$$\begin{aligned}&\boldsymbol{J}_h\left(\boldsymbol{N}_t - (\boldsymbol{J}_h\boldsymbol{N}_t)^{+}(\boldsymbol{J}_h\boldsymbol{N}_t)\right)\\ &= \boldsymbol{J}_h\left(\boldsymbol{N}_t - \boldsymbol{N}_t(\boldsymbol{J}_h\boldsymbol{N}_t)^{+}(\boldsymbol{J}_h\boldsymbol{N}_t)\right)\\ &= \boldsymbol{J}_h\boldsymbol{N}_t - (\boldsymbol{J}_h\boldsymbol{N}_t)(\boldsymbol{J}_h\boldsymbol{N}_t)^{+}(\boldsymbol{J}_h\boldsymbol{N}_t)\\ &= \boldsymbol{J}_h\boldsymbol{N}_t - \boldsymbol{J}_h\boldsymbol{N}_t = \boldsymbol{0},\end{aligned}$$

becomes $\dot{\boldsymbol{x}}_h = \boldsymbol{J}_h\,\boldsymbol{J}_t^{+}\boldsymbol{e}_t + \boldsymbol{J}_h\,\boldsymbol{N}_t\,\boldsymbol{J}_h^{+}\boldsymbol{e}_h$. In this expression it can be seen that the second priority task h is only affected by the higher priority task t. This completes the proof.

Appendix B

The SAM Scale

The Self-Assessment Manikin (SAM) is a tool for measuring emotional responses (Lang, 1980; Bradley and Lang, 1994). It is based in a model that represents emotions as points in a three dimensional space: the Pleasure, Arousal and Dominance (PAD) space.

Each dimension is depicted with a graphic character arrayed along a continuous nine-point array: for pleasure, the character ranges from a smiling figure to a frowning figure; for arousal, from sleepy to excited; and for dominance, a small figure representing a feeling of being controlled to a very large figure representing an in-control mood. Given an experiment with emotional content, the user can mark the point in each dimension array that best matches the level of pleasure, arousal and dominance that he/she has perceived. The particular emotion felt during the experiment can then be measured from the user responses.

A full-sized SAM scale can be found in the next page.

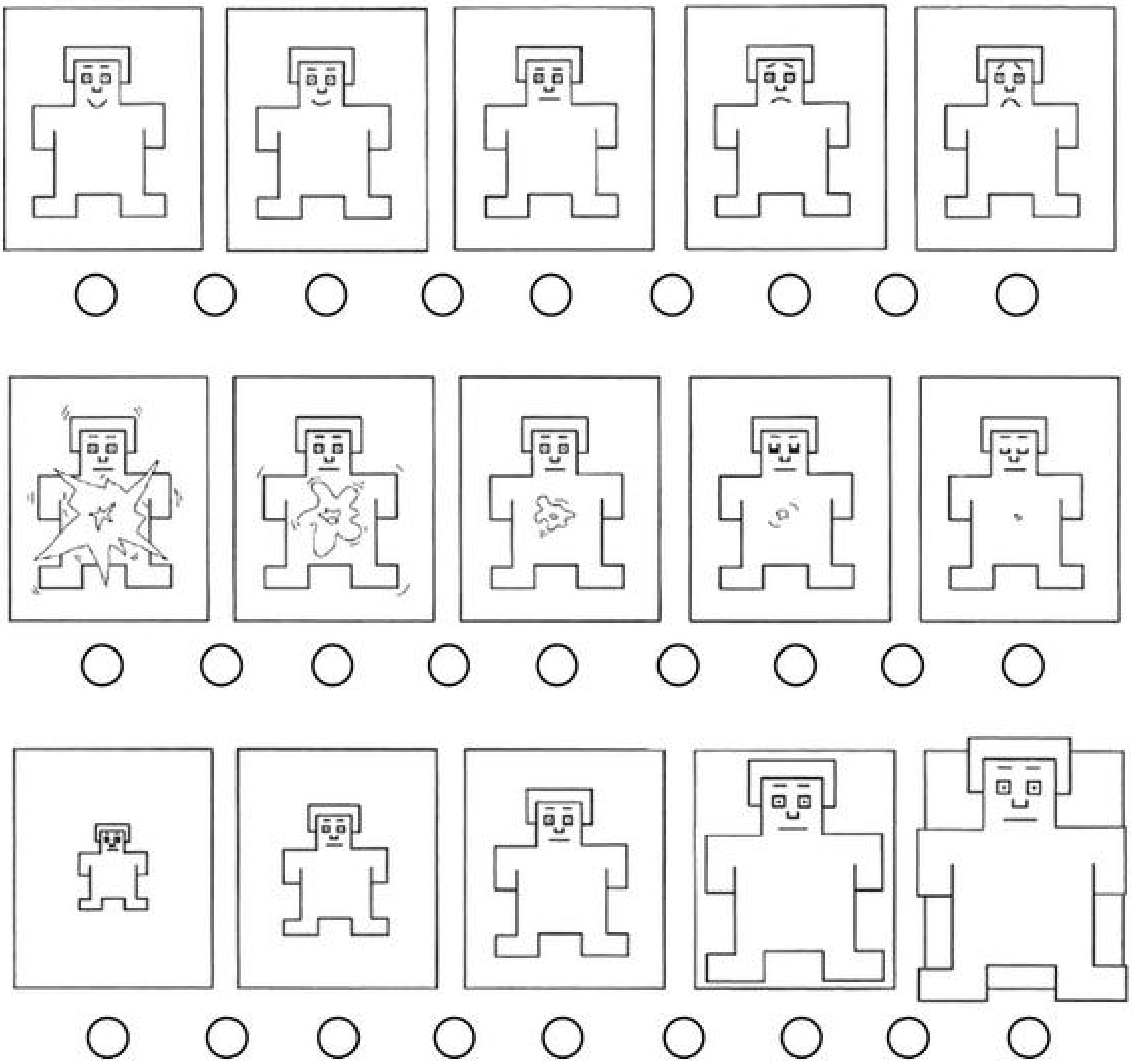


Figure B.1 The full-sized SAM scale.

Appendix C

The Emotional Conveyance User Study: Documentation

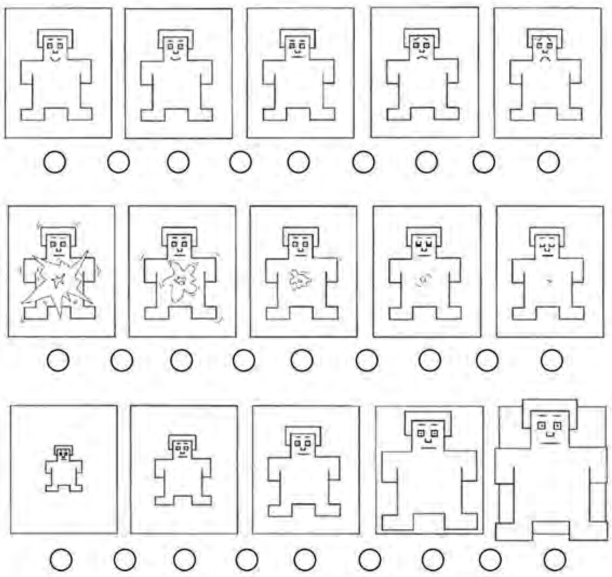
The documents given the the users of the study as explained in Chapter 5 are included in this appendix in the following order:

- Instructions given to the participants prior to the experiment in English (1 page).
- Instructions given to the participants prior to the experiment in Japanese (1 page).
- Questionnaire to be filled by the participants after the experiment (6 pages).

Pepper's emotional motions: a user study

Instructions:

We thank you for coming today and appreciate your participation in this experiment. In this study, we are interested in how people understand the emotions displayed by the Pepper robot through its motions. For about the next 5 minutes, you will be looking at Pepper doing different hello motions, and you will be rating each motion in terms of what you think Pepper was feeling during the motion. There are no right or wrong answers, so simply respond as honestly as you can.



In the picture above you can see a SAM scale. Each SAM scale consists on 3 sets of 5 figures along a continuum. The SAM scale shows three different kinds of feelings: Happy vs. Unhappy, Excited vs. Calm, and Controlled vs. In-control.

The first SAM scale is the happy-unhappy scale. At the left end of the happy vs. unhappy scale, you think Pepper was feeling completely happy, pleased, satisfied, hopeful. The right end of the scale is when Pepper was feeling completely unhappy, annoyed, unsatisfied, melancholic, despaired, bored. The figures allow you to describe feelings of pleasure by placing an "X" over the circle under the corresponding figure. If you think Pepper was feeling completely neutral, place an "X" over the figure in the middle. If the feeling of pleasure or displeasure falls between two of the pictures, then place an "X" in the circle between the figures.

The excited vs. calm dimension is the second scale. At the left end Pepper was feeling stimulated, excited, frenzied, jittery, wide-awake, aroused. At the right end Pepper was feeling completely relaxed, calm, sluggish, dull, sleepy, unaroused. You can proceed similarly as in the happy-unhappy scale.

The third scale is the dimension of controlled vs. in-control. At the left end Pepper was feeling completely controlled, influenced, cared-for, awed, submissive, guided. At the other extreme, Pepper was feeling completely controlling, influential, in control, important, dominant, autonomous. Note that when the figure is big, Pepper feels important and influential, and that it will be very small when Pepper feels controlled and guided.

Please rate HOW YOU THINK PEPPER WAS FEELING WHILE YOU WATCHED THE ROBOT MOTION.

We are interested in your own personal ratings of the motions. Therefore, please don't make any comments which might influence the ratings that other people make.

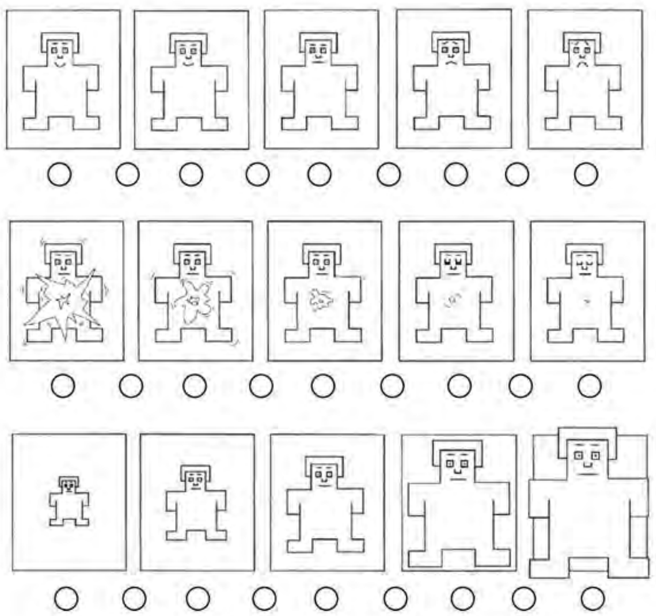
The procedure will be as follows: First you will be shown a neutral motion. The next 2 motions will be samples to practice the rating on the SAM scale. After, you will be shown 8 motions of 5 seconds and after each motion you will have 5 seconds to rate it using one SAM scale per motion. After the SAM ratings, 2 more motions will be shown, and for each motion you will have to chose which emotion, from a list of 5 options, suits better the emotion you felt while watching Pepper.

Are there any questions before we begin?

Pepper's emotional motions: a user study

Instructions:

この度は実験にご協力いただき有難うございます．私たちはペッパーの動作を通して表現される感情を人間がどのように理解するかを調査しています．約５分間，いくつかの種類のペッパーの手を振る動作を見ていただきます．そして，ペッパーを見ていただきながら，それぞれの動作中にペッパーが何を感じていたかについて評価していただきます．正解や間違いといったものはないので，可能な限り正直に回答をお願いします．



上記の図はSAMスケールというものです．このSAMスケールは５つの絵が横に並び３組用意されています．また，このSAMスケールはそれぞれ喜びと悲しみ，興奮と沈静，高揚か抑鬱の３種類の感情を表しています．

１つめのSAMスケールは喜びと悲しみについてです．このスケールの左端にいくほど，ペッパーは喜び，嬉しさ，満足，希望を感じていることになります．また，このスケールの右端に行くほど，ペッパーは悲しみ，不幸，いらつき，不満，憂鬱などを感じていることになります．ペッパーが感じた感情に一番近い絵の下の丸に”X”を書いてください．もし，ペッパーが何も感じていないと思ったら，”X”は真ん中の絵の下の丸に書いてください．もし，ペッパーの感情が２つの絵に該当すると感じたら，２つの絵の間の丸に”X”を書いてください．

２つめのスケールは興奮と冷静についてです．このスケールの左端にいくほど，ペッパーは興奮，活気，熱狂を感じていることになります．このスケールの右端にいくほど，ペッパーは冷静，落ち着き，穏やか，眠さを感じていることになります．チェックの方法については幸せと不幸に関するスケールと同じように行ってください．

３つめのスケールは気分のうきしずみについてです．このスケールの左端にいくほど，ペッパーは抑鬱，不自由，畏怖，服従などを感じていることになります．このスケールの右端にいくほど，ペッパーは高揚，自由，高尚，尊大などを感じていることになります．絵が大きくなるにつれて，ペッパーは尊大や奨励など優位な感情を持ち，小さくなるにつれてペッパーは抑鬱や服従など劣位な感情を持つこととなります．

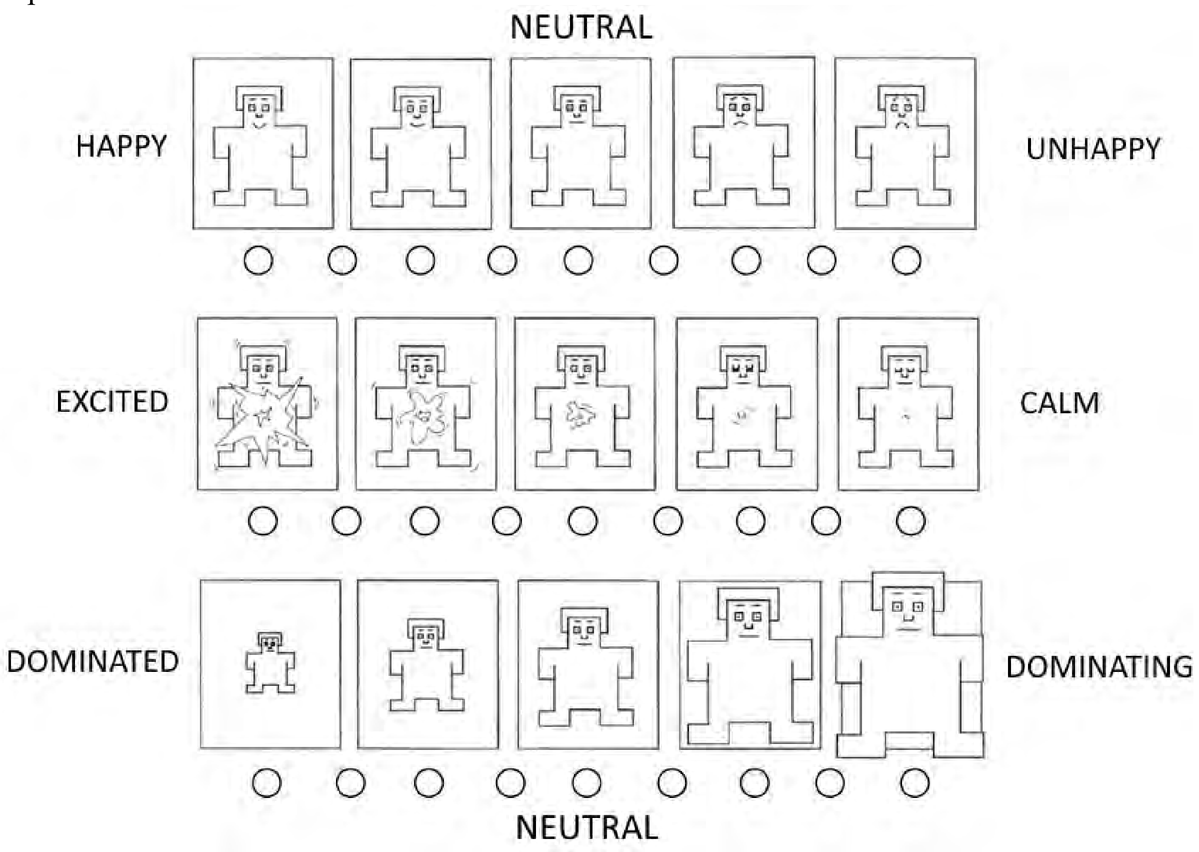
ペッパーの動作中に．ペッパーがどのような感情を抱いたと思うかの評価をお願いします．私たちはあなたの動作の評価に興味があります．また，実験中は他の方の回答に影響をあたえるようなコメントは言わないようお願いします．

実験の流れは以下のようになります．まずペッパーの自然な動作を見ていただきます．そして，次の２つの動作はSAMスケールによる評価の練習のための例として行います．その後５秒の動作を８つ見ていただき，それぞれの動作に対するSAMスケールによる評価を５秒でお願いします．SAMスケールによる評価が終わった後，さらに２つの動作を見ていただき，それぞれの動作についてリストにある５つの感情からどれがもっとも当てはまるか選んでください．

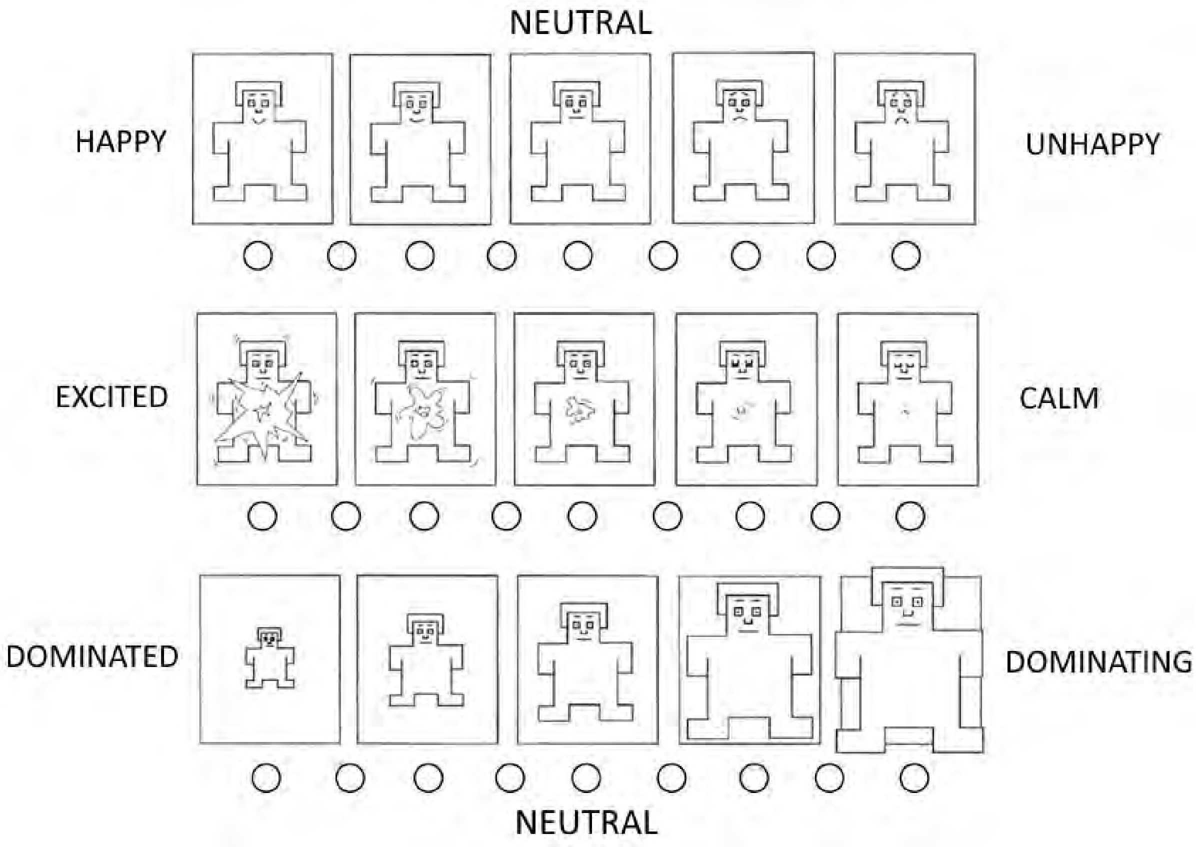
質問がある場合は実験前にお願いします．

Group:
Participant:

PRACTICE: 1

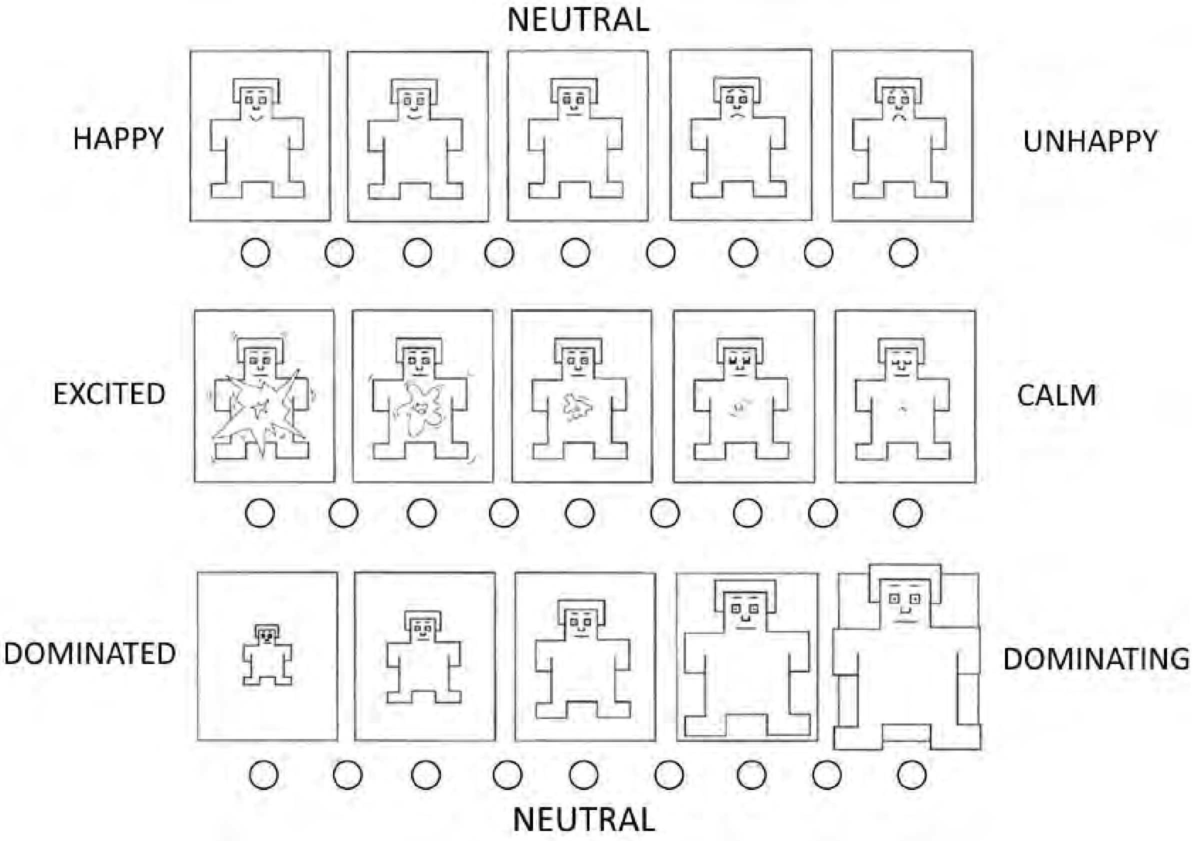


PRACTICE: 2

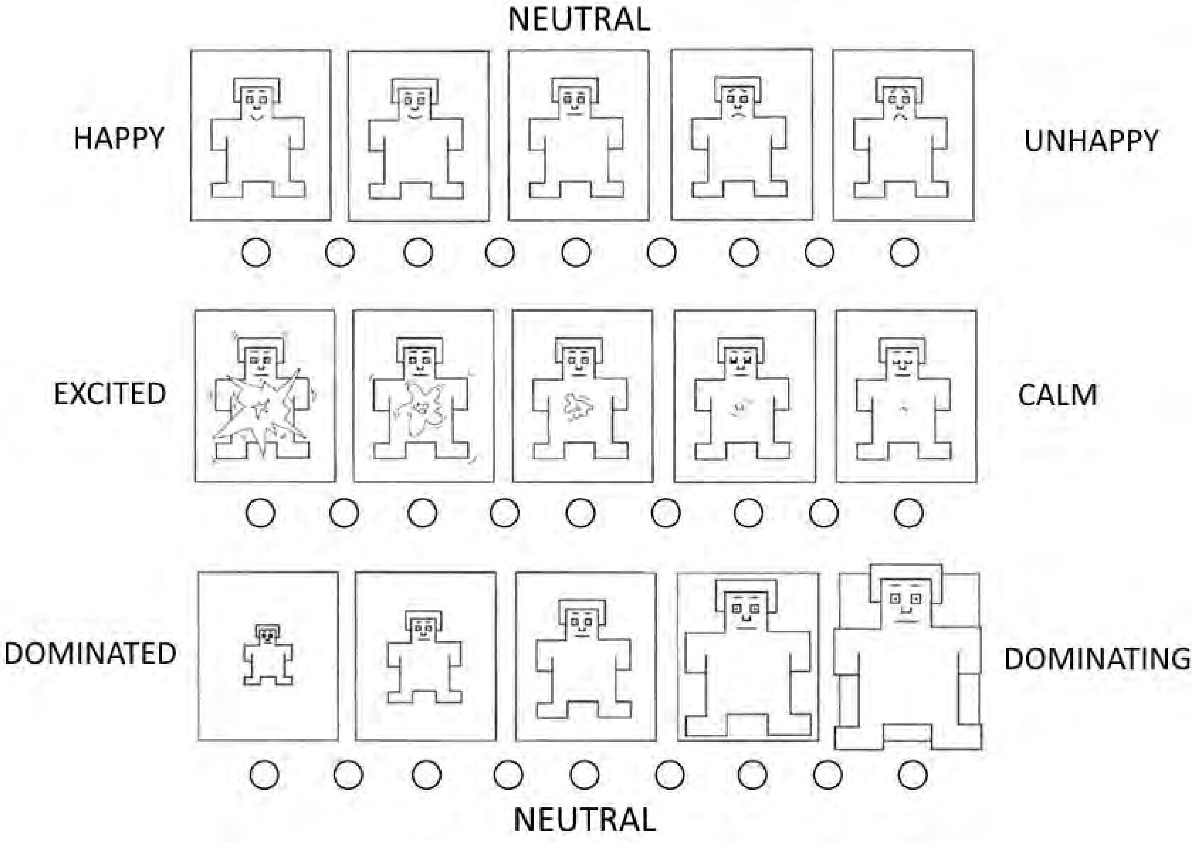


Group:
Participant:

MOTION: 1



MOTION: 2

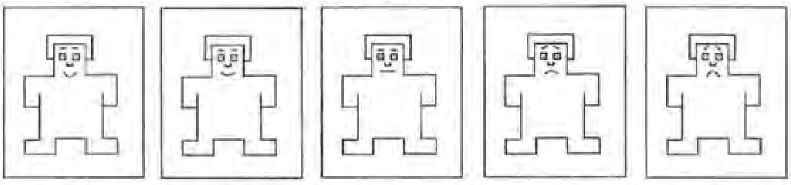


Group:
Participant:

MOTION: 3

NEUTRAL

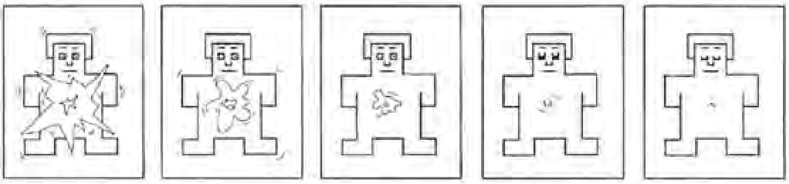
HAPPY



UNHAPPY

○ ○ ○ ○ ○ ○ ○ ○ ○

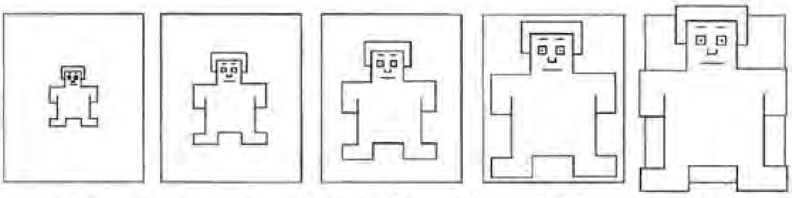
EXCITED



CALM

○ ○ ○ ○ ○ ○ ○ ○ ○

DOMINATED



DOMINATING

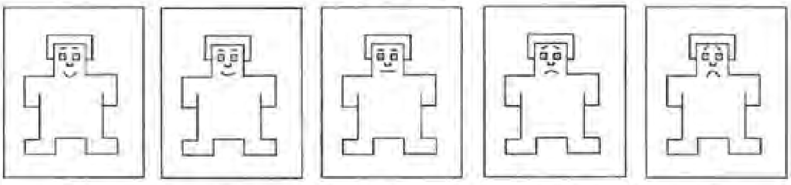
○ ○ ○ ○ ○ ○ ○ ○ ○

NEUTRAL

MOTION: 4

NEUTRAL

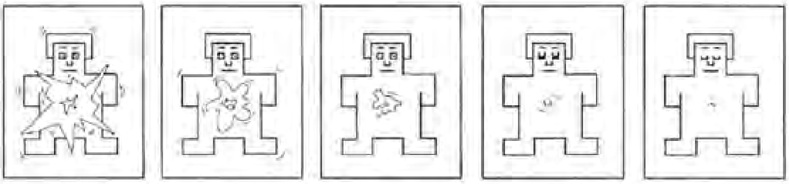
HAPPY



UNHAPPY

○ ○ ○ ○ ○ ○ ○ ○ ○

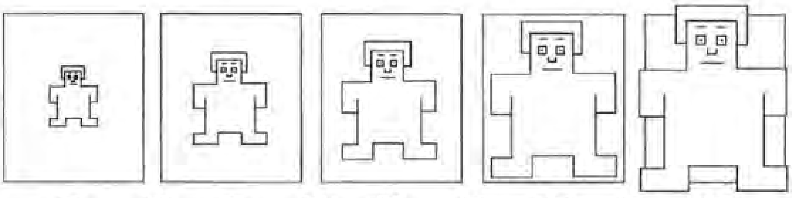
EXCITED



CALM

○ ○ ○ ○ ○ ○ ○ ○ ○

DOMINATED



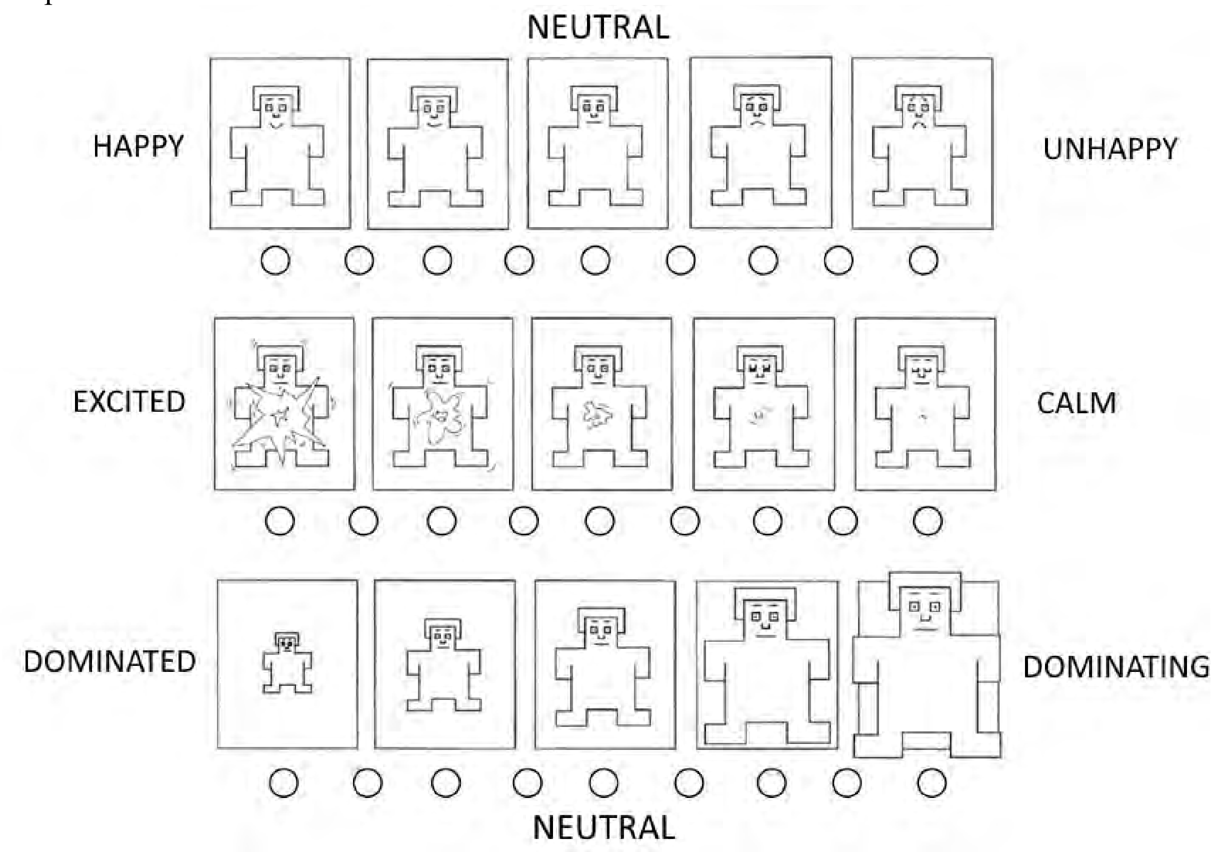
DOMINATING

○ ○ ○ ○ ○ ○ ○ ○ ○

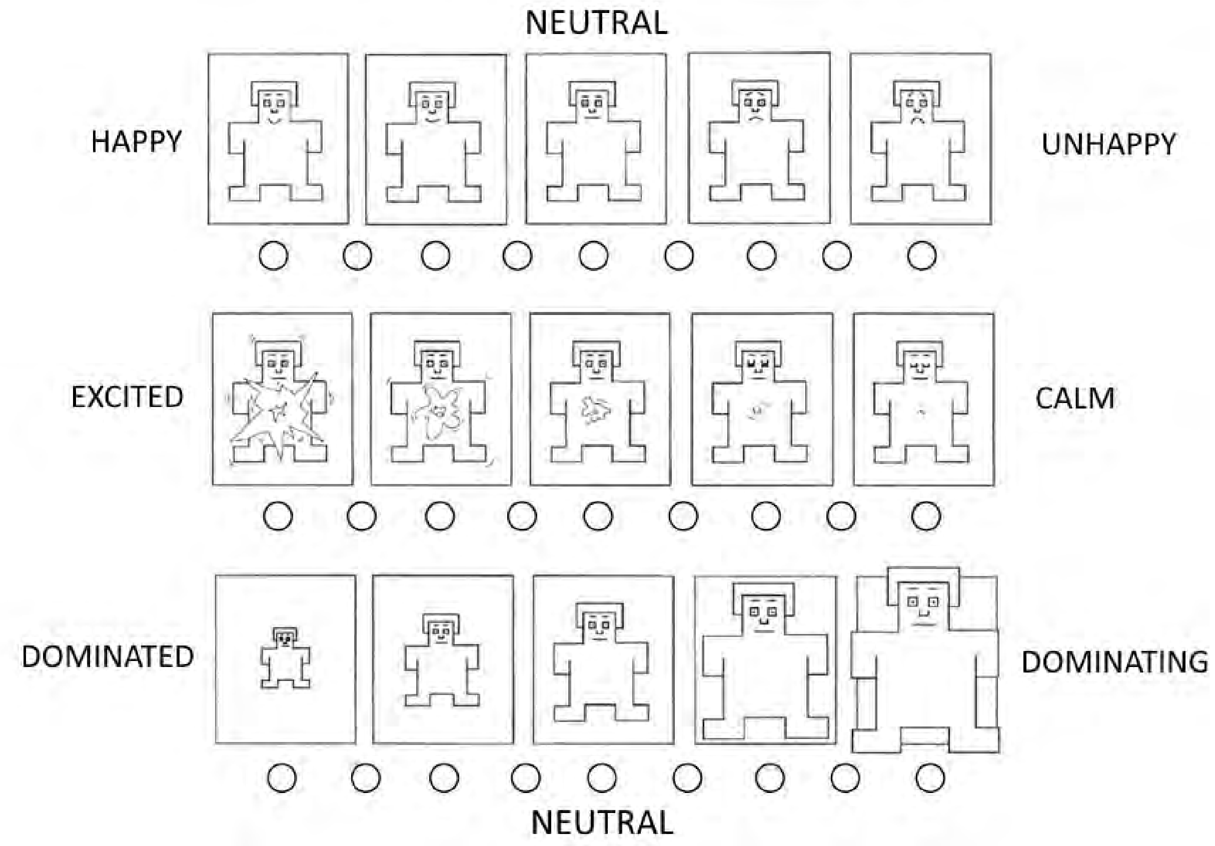
NEUTRAL

Group:
Participant:

MOTION: 5



MOTION: 6

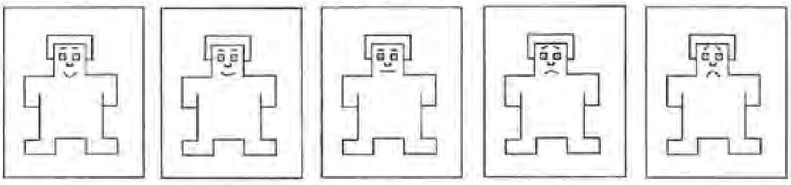


Group:
Participant:

MOTION: 7

NEUTRAL

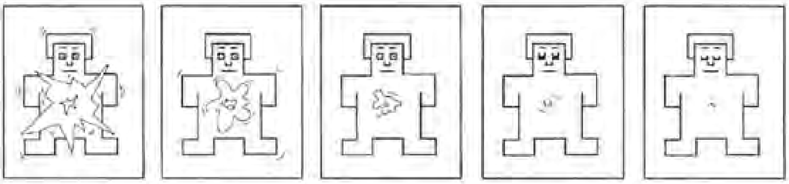
HAPPY



UNHAPPY

○ ○ ○ ○ ○ ○ ○ ○ ○

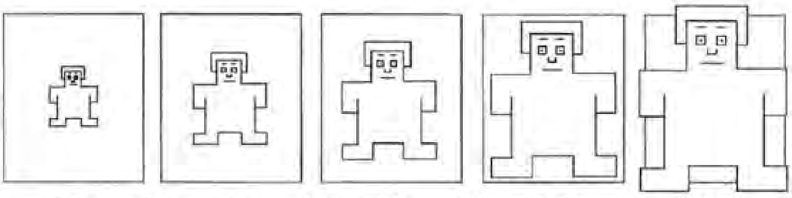
EXCITED



CALM

○ ○ ○ ○ ○ ○ ○ ○ ○

DOMINATED



DOMINATING

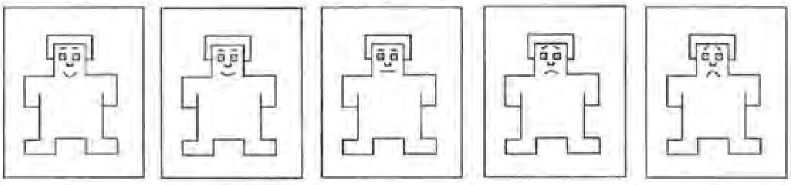
○ ○ ○ ○ ○ ○ ○ ○ ○

NEUTRAL

MOTION: 8

NEUTRAL

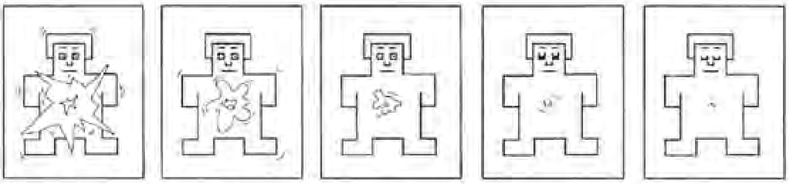
HAPPY



UNHAPPY

○ ○ ○ ○ ○ ○ ○ ○ ○

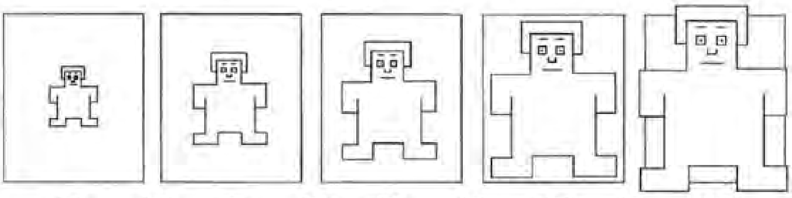
EXCITED



CALM

○ ○ ○ ○ ○ ○ ○ ○ ○

DOMINATED



DOMINATING

○ ○ ○ ○ ○ ○ ○ ○ ○

NEUTRAL

Group:

Participant:

EMOTION SELECTION

Please, put a circle on the emotion that you think better describes what Pepper was feeling.

Motion 1

- Calm
- Fear
- Happyness
- Sadness
- None of the above emotions

Motion 2

- Calm
- Fear
- Happyness
- Sadness
- None of the above emotions

Appendix D

The NASA-TLX

The NASA Task Load Index (NASA-TLX) is a subjective, multidimensional assessment tool that rates the perceived workload in order to assess a task, system, team's effectiveness or other aspects of performance (Hart, 2006).

It consists of two parts. In the first part the total workload is divided into six subjective subscales:

- Mental Demand
- Physical Demand
- Temporal Demand
- Performance
- Effort
- Frustration

In the second part, the subjects compare these subscales pairwise based on their perceived importance in order to individually weight them. The number of times each subscale is chosen is the weighted score, which is multiplied by the scale score for each dimension and divided by 15 to get the overall task load index.

The paper-and-pencil version of the NASA-TLX can be found in the next pages, with each page corresponding to the first and second part of the questionnaire, respectively.

Subject ID: ______________ Task ID: ____________________

RATING SHEET

MENTAL DEMAND

Low High

PHYSICAL DEMAND

Low High

TEMPORAL DEMAND

Low High

PERFORMANCE

Good Poor

EFFORT

Low High

FRUSTRATION

Low High

RATING SHEET

MENTAL DEMAND

Low High

PHYSICAL DEMAND

Low High

TEMPORAL DEMAND

Low High

PERFORMANCE

Good Poor

EFFORT

Low High

FRUSTRATION

Low High

Effort or Performance	Temporal Demand or Frustration	Frustration or Effort	Performance or Mental Demand
Temporal Demand or Effort	Physical Demand or Frustration	Performance or Temporal Demand	Mental Demand or Effort
Performance or Frustration	Physical Demand or Temporal Demand	Mental Demand or Physical Demand	Effort or Physical Demand
Physical Demand or Performance	Temporal Demand or Mental Demand	Frustration or Mental Demand	

Appendix E

Upper Bound in the Continuous Inverse Singular Values

With the aim to study the singular values of $\boldsymbol{N}_\oplus^i$, a development that leads to an upper bound of the largest singular value of

$$\boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\boldsymbol{J}_i,$$

is shown in this appendix.

Given Eq. 7.2

$$\boldsymbol{N}_\oplus^i = \boldsymbol{N}_\oplus^{i-1} - \boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\boldsymbol{J}_i,$$

the second term of the right member can be developed to

$$\begin{aligned}
\boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\,\boldsymbol{J}_i &= \left(\left(\boldsymbol{J}_i^T\right)^{\oplus\boldsymbol{N}_\oplus^{i-1}}\right)^T \boldsymbol{J}_i = \left(\left(\boldsymbol{U}_{i-1}^T\boldsymbol{J}_i^T\right)^{\oplus\boldsymbol{S}_{i-1}}\ \boldsymbol{U}_{i-1}^T\right)^T \boldsymbol{J}_i \\
&= \boldsymbol{U}_{i-1} \sum_{P\in B(m)} k_P \left(\left(\boldsymbol{H}_{P_i}\,\boldsymbol{U}_{i-1}^T\boldsymbol{J}_i^T\right)^+\right)^T \boldsymbol{J}_i \\
&= \boldsymbol{U}_{i-1} \sum_{P\in B(m)} k_P \left(\boldsymbol{J}_i\,\boldsymbol{U}_{i-1}\,\boldsymbol{H}_{P_i}\right)^+ \boldsymbol{J}_i
\end{aligned} \tag{E.1}$$

making use of the SVD decomposition of $\boldsymbol{N}_\oplus^{i-1} = \boldsymbol{U}_{i-1}\boldsymbol{S}_{i-1}\boldsymbol{U}_{i-1}^T$, assuming that $\boldsymbol{N}_\oplus^{i-1}$ is a positive symmetric matrix, where $k_P = \prod_{j\in P}\sigma_j^{i-1}\prod_{j\notin P}(1-\sigma_j^{i-1})$ as in Eq. 3.8, and that σ_k^{i-1} is the k-th largest singular value of $\boldsymbol{N}_\oplus^{i-1}$.

Given $\boldsymbol{A} = \sum \boldsymbol{A}_i$, it is well known that $\sigma_1(\boldsymbol{A}) = \|\boldsymbol{A}\| \leq \sum \|\boldsymbol{A}_i\|$ and $\|\boldsymbol{R}\,\boldsymbol{A}\| = \|\boldsymbol{A}\|$, where $\boldsymbol{R}$ has all singular value 1, $\sigma_1(\boldsymbol{A})$ is the largest singular value of a matrix $\boldsymbol{A}$, and $\|\boldsymbol{A}\|$, the spectral norm of $\boldsymbol{A}$. Thus, following from Eq. E.1,

$$\begin{aligned}\sigma_1\left(\boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\,\boldsymbol{J}_i\right) &= \left\|\boldsymbol{U}_{i-1}\sum_{P\in B(m)} k_P\left(\boldsymbol{J}_i\,\boldsymbol{U}_{i-1}\,\boldsymbol{H}_{P_i}\right)^+\boldsymbol{J}_i\right\| \\ &= \left\|\sum_{P\in B(m)} k_P\left(\boldsymbol{J}_i\,\boldsymbol{U}_{i-1}\,\boldsymbol{H}_{P_i}\right)^+\boldsymbol{J}_i\,\boldsymbol{U}_{i-1}\right\| \\ &\leq \sum_{P\in B(m)} k_P\left\|\left(\boldsymbol{J}_i\,\boldsymbol{U}_{i-1}\,\boldsymbol{H}_{P_i}\right)^+\boldsymbol{J}_i\,\boldsymbol{U}_{i-1}\right\|.\end{aligned} \tag{E.2}$$

Considering $\boldsymbol{J}_i\,\boldsymbol{U}_{i-1} = \boldsymbol{J}_i^{\boldsymbol{U}}$, it is always possible to find a permutation matrix $\boldsymbol{P}_C^T$ that sorts the columns of $\boldsymbol{J}_i^{\boldsymbol{U}}\boldsymbol{H}_{P_i}$ such that its null columns are all on the right side of the new permuted matrix $\boldsymbol{J}_i^{\boldsymbol{UC}}\boldsymbol{H}_{P_i}^C$, that is, $\boldsymbol{J}_i^{\boldsymbol{U}}\boldsymbol{H}_{P_i}\boldsymbol{P}_C^T = \boldsymbol{J}_i^{\boldsymbol{UC}}\boldsymbol{H}_{P_i}^C = \begin{bmatrix}\boldsymbol{J}_1^{\boldsymbol{UC}} & \boldsymbol{0}\end{bmatrix}$, and also $\boldsymbol{J}_i^{\boldsymbol{U}}\boldsymbol{P}_C^T = \boldsymbol{J}_i^{\boldsymbol{UC}} = \begin{bmatrix}\boldsymbol{J}_1^{\boldsymbol{UC}} & \boldsymbol{J}_2^{\boldsymbol{UC}}\end{bmatrix}$, such that $\boldsymbol{J}_1^{\boldsymbol{UC}} \in \mathbb{R}^{m\times n_1}$ and $\boldsymbol{J}_2^{\boldsymbol{UC}} \in \mathbb{R}^{m\times n_2}$. Then Eq. E.2 results in

$$\begin{aligned}\sigma_1\left(\boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\,\boldsymbol{J}_i\right) &\leq \sum_{P\in B(m)} k_P\left\|\left(\boldsymbol{J}_i\,\boldsymbol{U}_{i-1}\,\boldsymbol{H}_{P_i}\right)^+\boldsymbol{J}_i\,\boldsymbol{U}_{i-1}\right\| \\ &= \sum_{P\in B(m)} k_P\left\|\boldsymbol{P}_C^T\left(\boldsymbol{J}_i^{\boldsymbol{UC}}\,\boldsymbol{H}_{P_i}^C\right)^+\boldsymbol{J}_i^{\boldsymbol{UC}}\,\boldsymbol{P}_C\right\| \\ &= \sum_{P\in B(m)} k_P\left\|\left(\boldsymbol{J}_i^{\boldsymbol{UC}}\,\boldsymbol{H}_{P_i}^C\right)^+\boldsymbol{J}_i^{\boldsymbol{UC}}\right\|.\end{aligned} \tag{E.3}$$

Assigning $\boldsymbol{M}_P$ to the matrix inside the latter norm, developing it and further dropping the superscript $\boldsymbol{UC}$ for ease of readability leads to

$$\begin{aligned}\|\boldsymbol{M}_P\| &= \left\|\begin{bmatrix}\boldsymbol{J}_i^{\boldsymbol{UC}} & \boldsymbol{H}_{P_i}^C\end{bmatrix}^+\boldsymbol{J}_i^{\boldsymbol{UC}}\right\| = \left\|\begin{bmatrix}\boldsymbol{J}_1^{\boldsymbol{UC}} & \boldsymbol{0}\end{bmatrix}^+\begin{bmatrix}\boldsymbol{J}_1^{\boldsymbol{UC}} & \boldsymbol{J}_2^{\boldsymbol{UC}}\end{bmatrix}\right\| \\ &= \left\|\begin{bmatrix}\boldsymbol{J}_1^+ \\ \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{J}_1 & \boldsymbol{J}_2\end{bmatrix}\right\| = \left\|\begin{bmatrix}\boldsymbol{J}_1^+\,\boldsymbol{J}_1 & \boldsymbol{J}_1^+\,\boldsymbol{J}_2 \\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\right\|.\end{aligned} \tag{E.4}$$

This norm corresponds to the largest singular value of its matrix and, thus, to the square root of its largest eigenvalue:

$$\sqrt{\lambda_1\left(\boldsymbol{M}_P^T\boldsymbol{M}_P\right)} = \sqrt{\lambda_1\begin{bmatrix}\boldsymbol{J}_1^+\boldsymbol{J}_1 & \boldsymbol{J}_1^+\boldsymbol{J}_2 \\ (\boldsymbol{J}_1^+\boldsymbol{J}_2)^T & (\boldsymbol{J}_1^+\boldsymbol{J}_2)^T\boldsymbol{J}_1^+\boldsymbol{J}_2\end{bmatrix}}. \tag{E.5}$$

On the assumption that the largest eigenvalue of Eq. E.5 is bigger than 1, this value can be obtained after solving the characteristic equation as

$$\det\left(\boldsymbol{J}_1^+\boldsymbol{J}_1 - \lambda\boldsymbol{I}\right)\cdot\det\left((\boldsymbol{J}_1^+\boldsymbol{J}_2)^T\boldsymbol{J}_1^+\boldsymbol{J}_2 - \lambda\boldsymbol{I} - (\boldsymbol{J}_1^+\boldsymbol{J}_2)^T(\boldsymbol{J}_1^+\boldsymbol{J}_1 - \lambda\boldsymbol{I})^{-1}\boldsymbol{J}_1^+\boldsymbol{J}_2\right) = 0. \tag{E.6}$$

Taking $\boldsymbol{J}_1 \in \mathbb{R}^{m \times n_1}$, if $m \geq n_1$ $\boldsymbol{J}_1^+ \boldsymbol{J}_1$ will have a unique singular value of 1 with multiplicity n_1. If $m < n_1$ $\boldsymbol{J}_1^+ \boldsymbol{J}_1$ will have two singular values: 1, with multiplicity n_1, and 0, with multiplicity $n_1 - m$. Now, considering the SVD of $\boldsymbol{J}_1 = \boldsymbol{U}_1 \boldsymbol{S}_1 \boldsymbol{V}_1^T$, it can be seen that $\boldsymbol{J}_1^+ \boldsymbol{J}_1 = \boldsymbol{V}_1 \boldsymbol{S}_1^+ \boldsymbol{U}_1^T \boldsymbol{U}_1 \boldsymbol{S}_1 \boldsymbol{V}_1^T = \boldsymbol{V}_1 \boldsymbol{S}_1^+ \boldsymbol{S}_1 \boldsymbol{V}_1^T$, in which the components of the diagonal matrix $\boldsymbol{S}_1^+ \boldsymbol{S}_1$ will be 0 or 1. Thus, $\boldsymbol{J}_1^+ \boldsymbol{J}_1 - \lambda \boldsymbol{I} = \boldsymbol{V}_1 \boldsymbol{S}_1^+ \boldsymbol{S}_1 \boldsymbol{V}_1^T - \lambda \boldsymbol{I} = \boldsymbol{V}_1 \boldsymbol{S}_1^+ \boldsymbol{S}_1 \boldsymbol{V}_1^T - \lambda \boldsymbol{V}_1 \boldsymbol{V}_1^T = \boldsymbol{V}_1 (\boldsymbol{S}_1^+ \boldsymbol{S}_1 - \lambda \boldsymbol{I}) \boldsymbol{V}_1^T$ and so, for the case $m < n_1$, the determinant $\det\left(\boldsymbol{J}_1^+ \boldsymbol{J}_1 - \lambda \boldsymbol{I}\right) = \det\left(\boldsymbol{S}_1^+ \boldsymbol{S}_1 - \lambda \boldsymbol{I}\right) = (1-\lambda)^{n_1} \lambda^{n_1 - m}$.

From the SVD decomposition of $\boldsymbol{J}_2 = \boldsymbol{U}_2 \boldsymbol{S}_2 \boldsymbol{V}_2^T$ the second term in expression inside the latter determinant in Eq. E.6 can be developed as

$$\begin{aligned}
&(\boldsymbol{J}_1^+ \boldsymbol{J}_2)^T (\boldsymbol{J}_1^+ \boldsymbol{J}_1 - \lambda \boldsymbol{I})^{-1} \boldsymbol{J}_1^+ \boldsymbol{J}_2 \\
&= \boldsymbol{V}_2 \boldsymbol{S}_2 \boldsymbol{U}_2^T \boldsymbol{U}_1 \boldsymbol{S}_1^+ \boldsymbol{V}_1^T \left(\boldsymbol{V}_1 \left(\boldsymbol{S}_1^+ \boldsymbol{S}_1 - \lambda \boldsymbol{I}\right) \boldsymbol{V}_1^T\right)^{-1} \boldsymbol{V}_1 \boldsymbol{S}_1^+ \boldsymbol{U}_1^T \boldsymbol{U}_2 \boldsymbol{S}_2 \boldsymbol{V}_2^T \\
&= \boldsymbol{V}_2 \boldsymbol{S}_2 \boldsymbol{U}_2^T \boldsymbol{U}_1 \left(\boldsymbol{S}_1 \left(\boldsymbol{S}_1^+ \boldsymbol{S}_1 - \lambda \boldsymbol{I}\right) \boldsymbol{S}_1\right)^{-1} \boldsymbol{U}_1^T \boldsymbol{U}_2 \boldsymbol{S}_2 \boldsymbol{V}_2^T \\
&= \boldsymbol{V}_2 \boldsymbol{S}_2 \boldsymbol{U}_2^T \boldsymbol{U}_1 \left(\boldsymbol{S}_1 \boldsymbol{S}_1 \left(1 - \lambda\right)\right)^{-1} \boldsymbol{U}_1^T \boldsymbol{U}_2 \boldsymbol{S}_2 \boldsymbol{V}_2^T \\
&= \frac{1}{1-\lambda} \boldsymbol{V}_2 \boldsymbol{S}_2 \boldsymbol{U}_2^T \boldsymbol{U}_1 \boldsymbol{S}_1^+ \boldsymbol{V}_1^T \boldsymbol{V}_1 \boldsymbol{S}_1^+ \boldsymbol{U}_1^T \boldsymbol{U}_2 \boldsymbol{S}_2 \boldsymbol{V}_2^T \\
&= \frac{1}{1-\lambda} \left(\boldsymbol{J}_1^+ \boldsymbol{J}_2\right)^T \boldsymbol{J}_1^+ \boldsymbol{J}_2.
\end{aligned} \tag{E.7}$$

And now the expression in the second determinant in Eq. E.6 can be transformed to

$$\begin{aligned}
&(\boldsymbol{J}_1^+ \boldsymbol{J}_2)^T \boldsymbol{J}_1^+ \boldsymbol{J}_2 - \lambda \boldsymbol{I} - \left(\boldsymbol{J}_1^+ \boldsymbol{J}_2\right)^T \left(\boldsymbol{J}_1^+ \boldsymbol{J}_1 - \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{J}_1^+ \boldsymbol{J}_2 \\
&= -\frac{\lambda}{1-\lambda} \left(\left(1-\lambda\right) \boldsymbol{I} + \left(\boldsymbol{J}_1^+ \boldsymbol{J}_2\right)^T \boldsymbol{J}_1^+ \boldsymbol{J}_2\right).
\end{aligned} \tag{E.8}$$

Thus, taking $\boldsymbol{J}_2 \in \mathbb{R}^{m \times n_2}$, the left side of Eq. E.6 can be rearranged to

$$\begin{aligned}
&(1-\lambda)^{n_1} \lambda^{n_1 - m} \det\left(-\frac{\lambda}{1-\lambda} \left((1-\lambda) \boldsymbol{I} + (\boldsymbol{J}_1^+ \boldsymbol{J}_2)^T \boldsymbol{J}_1^+ \boldsymbol{J}_2\right)\right) \\
&= (-1)^{n_2} \left(1-\lambda\right)^{n_1 - n_2} \lambda^{n_1 + n_2 - m} \det\left(\boldsymbol{I} + \left(\boldsymbol{J}_1^+ \boldsymbol{J}_2\right)^T \boldsymbol{J}_1^+ \boldsymbol{J}_2 - \lambda \boldsymbol{I}\right).
\end{aligned} \tag{E.9}$$

And accounting that the matrix in the last determinant is symmetric then $\left(\boldsymbol{J}_1^+ \boldsymbol{J}_2\right)^T \boldsymbol{J}_1^+ \boldsymbol{J}_2 = \boldsymbol{U}_{12} \boldsymbol{S}_{12} \boldsymbol{U}_{12}^T$ so

$$\begin{aligned}
&\det\left(\boldsymbol{I} + \left(\boldsymbol{J}_1^+ \boldsymbol{J}_2\right)^T \boldsymbol{J}_1^+ \boldsymbol{J}_2 - \lambda \boldsymbol{I}\right) \\
&= \det\left(\boldsymbol{U}_{12} \left(I + \boldsymbol{S}_{12} - \lambda \boldsymbol{I}\right) \boldsymbol{U}_{12}^T\right) \\
&= \det\left(\boldsymbol{I} + \boldsymbol{S}_{12} - \lambda \boldsymbol{I}\right).
\end{aligned} \tag{E.10}$$

which shows that the singular values of $\boldsymbol{I} + \left(\boldsymbol{J}_1^+ \boldsymbol{J}_2\right)^T \boldsymbol{J}_1^+ \boldsymbol{J}_2$ solve the characteristic equation in Eq. E.6. This in turn allows the computation of the singular values of $\boldsymbol{M}_P$ (Eq. E.4), which are 0, 1 and $\sqrt{1 + \sigma_1^2(\boldsymbol{J}_1^+ \boldsymbol{J}_2)}$.

So it is easy to see that the largest singular value of $\boldsymbol{M}_P$ is $\sigma_p = \sqrt{1+\sigma_1^2(\boldsymbol{J}_1^+\boldsymbol{J}_2)}$, which, in general, will be greater that 1, and only equal to 1 when the tasks corresponding to the Jacobians $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$ are decoupled. σ_p poses an upper bound to the largest singular value of the expression $\sigma_1\left(\boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\,\boldsymbol{J}_i\right)$ in Eq. E.2:

$$\begin{aligned}\sigma_1\left(\boldsymbol{J}_i^{\boldsymbol{N}_\oplus^{i-1}\oplus}\,\boldsymbol{J}_i\right) &\leq \sum_{P\in B(m)} k_P \left\|(\boldsymbol{J}_i\,\boldsymbol{U}_{i-1}\,\boldsymbol{H}_{P_i})^+\,\boldsymbol{J}_i\,\boldsymbol{U}_{i-1}\right\| \\ &= \sum_{P\in B(m)} k_P \sqrt{1+\sigma_1^2\left((\boldsymbol{J}_i\,\boldsymbol{U}_{i-1}\,\boldsymbol{H}_{P_i})^+\,\boldsymbol{J}_i\,\boldsymbol{U}_{i-1}\right)}.\end{aligned} \tag{E.11}$$

On the contrary, if instead of the right continuous inverse, the continuous inverse is analyzed in a similar way, the upper bound for its null space projector can be guaranteed to be at most 1. From the projector operator onto the null space of the first level, $\boldsymbol{N}_\oplus^1$ in Eq. 7.1 it is straightforward to see that

$$\left\|\sum_{P\in B(m)} k_P\,(\boldsymbol{H}_{\boldsymbol{P}_1}\,\boldsymbol{J}_1)^+\,\boldsymbol{J}_1\right\| \leq \sum_{P\in B(m)} k_P \left\|(\boldsymbol{H}_{\boldsymbol{P}_1}\,\boldsymbol{J}_1)^+\,\boldsymbol{J}_1\right\|. \tag{E.12}$$

Following as in Eq. E.3, a permutation matrix can be found that sorts the rows of $\boldsymbol{H}_P\boldsymbol{J}_1$ so that the last row of the permuted matrix is a null vector. And thus, naming the permuted matrix inside the latter norm as $\boldsymbol{M}_P$, then, as in Eq. E.4,

$$\|\boldsymbol{M}_P\| = \left\|\begin{bmatrix}\boldsymbol{J}_{11}^+ & \mathbf{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{J}_{11}\\ \boldsymbol{J}_{12}\end{bmatrix}\right\| = \left\|\boldsymbol{J}_{11}^+\boldsymbol{J}_{11}\right\|, \tag{E.13}$$

which is 1. Since $\sum_{P\in B(m)} k_P$ adds up 1 if at least one of the h_i is 1 (Mansard et al., 2009a), this proves that the singular values of $\boldsymbol{N}_\oplus^1$ lie in the unit interval.

Appendix F

Minimum distance between the images of a link and a point

Given a link L defined by its two end points $\boldsymbol{r}^M \in \mathbb{R}^3$ and $\boldsymbol{r}^m \in \mathbb{R}^3$ such that any point $\boldsymbol{r}_L \in \mathbb{R}^3$ in the link can be computed $\boldsymbol{r}_L = \boldsymbol{r}^m + \lambda\, \boldsymbol{n}$ with $\lambda \in [0, 1]$ and $\boldsymbol{n} = \boldsymbol{r}^M - \boldsymbol{r}^m$, and given a third point $\boldsymbol{r} \in \mathbb{R}^3$, the aim is to find the value of λ^* that minimizes the distance $D \in \mathbb{R}_{>0}$ between the images of the link L and the point $\boldsymbol{r}$ in the image plane of a pinhole camera model.

The solution to this minimization problem can be expressed as

$$\lambda^* = \arg\min_{\lambda} (D) = \arg\min_{\lambda} \left(D^2\right) = \arg\min_{\lambda} \left(\left(\frac{r_x^m + \lambda\, n_x}{r_z^m + \lambda\, n_z} - \frac{r_x}{r_z} \right)^2 + \left(\frac{r_y^m + \lambda\, n_y}{r_z^m + \lambda\, n_z} - \frac{r_y}{r_z} \right)^2 \right)$$

with $\boldsymbol{r}^m = \begin{bmatrix} r_x^m & r_y^m & r_z^m \end{bmatrix}^T$, $\boldsymbol{r} = \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T$ and $\boldsymbol{n} = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}^T$.

By first differentiating with respect to λ and then equating the expression to zero, the following equation is obtained:

$$\begin{aligned}\frac{\partial D^2}{\partial \lambda} = 2 \left(\frac{\lambda\, n_x + r_x^m}{\lambda\, n_z + r_z^m} - \frac{r_x}{r_z} \right) \left(\frac{n_x}{\lambda\, n_z + r_z^m} - \frac{(\lambda\, n_x + r_x^m)\, n_z}{(\lambda\, n_z + r_z^m)^2} \right) + \\ + 2 \left(\frac{\lambda\, n_y + r_y^m}{\lambda\, n_z + r_z^m} - \frac{r_y}{r_z} \right) \left(\frac{n_y}{\lambda\, n_z + r_z^m} - \frac{\left(\lambda\, n_y + r_y^m\right) n_z}{(\lambda\, n_z + r_z^m)^2} \right) = 0.\end{aligned}$$

Solving this equation, $\lambda^* = {}^{A}\!/_{B}$ is obtained with

$$\begin{aligned} A &= (r_x^m)^2 r_z\, n_z - r_x^m\, r_z^m\, r_x\, n_z - r_x^m\, r_z^m\, r_z\, n_x + (r_y^m)^2 r_z\, n_z - r_y^m\, r_z^m\, r_y\, n_z - r_y^m\, r_z^m\, r_z\, n_y \\ &\quad + (r_z^m)^2 r_x\, n_x + (r_z^m)^2 r_y\, n_y \\ B &= r_x^m\, r_x\, (n_z)^2 - r_x^m\, r_z\, n_x\, n_z + r_y^m\, r_y\, (n_z)^2 - r_y^m\, r_z\, n_y\, n_z - r_z^m\, r_x\, n_x\, n_z - r_z^m\, r_y\, n_y\, n_z \\ &\quad + r_z^m\, r_z\, (n_x)^2 + r_z^m\, r_z\, (n_y)^2. \end{aligned}$$

Bibliography

The Parrot Ardrone 2.0. https://www.parrot.com/global/drones/parrot-ardrone-20-elite-edition/, 2010.

RViz ROS node. http://wiki.ros.org/rviz, 2011.

The Parrot Ardrone ROS node. https://github.com/AutonomyLab/ardrone_autonomy/, 2011.

The Alvar AR tag tracking ROS node. http://wiki.ros.org/ar_track_alvar/, 2012.

The Phantom Omni ROS node. http://wiki.ros.org/phantom_omni/, 2012.

Fast Research Interface Library. https://cs.stanford.edu/people/tkr/fri/html/, 2014.

Pepper ROS node. http://wiki.ros.org/pepper_robot, 2015.

Humanoid Robot, Wikipedia. https://en.wikipedia.org/wiki/Humanoid_robot, 2018.

ABB. https://new.abb.com/products/robotics/industrial-robots/irb-1520id, 2018.

ADAMS JR., B. and KLECK, R. Effects of direct and averted gaze on the perception of facially communicated emotion. *Emotion*, vol. 5(1), pages 3–11, 2005. ISSN 1528-3542.

ALLEN, P. K., TIMCENKO, A., YOSHIMI, B. and MICHELMAN, P. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Transactions on Robotics and Automation*, vol. 9(2), pages 152–165, 1993. ISSN 1042-296X.

AMAYA, K., BRUDERLIN, A. and CALVERT, T. Emotion from motion. In *Proceedings of the Conference on Graphics Interface '96*, GI '96, pages 222–229. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 1996. ISBN 0-9695338-5-3.

ASADA, M. Towards artificial empathy. *International Journal of Social Robotics*, vol. 7(1), pages 19–33, 2015. ISSN 1875-4791.

BADDOURA, R. and VENTURE, G. This robot is sociable: Close-up on the gestures and measured motion of a human responding to a proactive robot. *International Journal of Social Robotics*, vol. 7(4), pages 489–496, 2015. ISSN 1875-4791.

BAERLOCHER, P. and BOULIC, R. Task-priority formulations for the kinematic control of highly redundant articulated structures. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 1, pages 323–329 vol.1. 1998.

BAERLOCHER, P. and BOULIC, R. An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *Vis Comput*, vol. 20(6), pages 402–417, 2004. ISSN 0178-2789.

BAILLIEUL, J. Kinematic programming alternatives for redundant manipulators. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, pages 722–728. 1985.

BECK, A., HIOLLE, A., MAZEL, A. and CAÑAMERO, L. Interpretation of emotional body language displayed by robots. In *Proceedings of the 3rd International Workshop on Affective Interaction in Natural Environments*, AFFINE '10, pages 37–42. ACM, New York, NY, USA, 2010. ISBN 978-1-4503-0170-1.

BEJCZY, A. K. *Robot arm dynamics and control*, vol. 33-669 of *JPL technical memorandum*. Jet Propulsion Laboratory, 1974.

BELTA, C. and KUMAR, V. Abstraction and control for groups of robots. *IEEE Transactions on robotics*, vol. 20(5), pages 865–875, 2004.

BERNHARDT, D. and ROBINSON, P. Detecting affect from non-stylised body motions. In *Proceedings of the 2Nd International Conference on Affective Computing and Intelligent Interaction*, ACII '07, pages 59–70. Springer-Verlag, Berlin, Heidelberg, 2007. ISBN 978-3-540-74888-5.

BERNS, K. and HIRTH, J. Control of facial expressions of the humanoid robot head roman. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3119–3124. 2006.

BI, Y. and DUAN, H. Implementation of autonomous visual tracking and landing for a low-cost quadrotor. *Optik-International Journal for Light and Electron Optics*, vol. 124(18), pages 3296–3300, 2013. ISSN 0030-4026.

BISCHOFF, R., KURTH, J., SCHREIBER, G., KOEPPE, R., ALBU-SCHAEFFER, A., BEYER, A., EIBERGER, O., HADDADIN, S., STEMMER, A., GRUNWALD, G. and HIRZINGER, G. The kuka-dlr lightweight robot arm - a new reference platform for robotics research and manufacturing. In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pages 1–8. 2010.

BRADLEY, M. M. and LANG, P. J. Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, vol. 25(1), pages 49–59, 1994. ISSN 0005-7916.

BREAZEAL, C. Emotion and sociable humanoid robots. *International Journal of Human-Computer Studies*, vol. 59(1-2), pages 119–155, 2003. ISSN 1071-5819.

BREAZEAL, C. and BROOKS, R. Robot emotion: A functional perspective. *MIT Media Lab*, 2004.

BREAZEAL, C. and BROOKS, R. Robot Emotion: A Functional Perspective. In *Who Needs Emotions?: The brain meets the robot* (edited by J.-M. Fellous and M. A. Arbib). Oxford Scholarship Online, 2005.

BROCK, O., KHATIB, O. and VIJI, S. Task-consistent obstacle avoidance and motion behavior for mobile manipulation. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 1, pages 388–393 vol.1. 2002.

BUSS, S. R. and KIM, J.-S. Selectively damped least squares for inverse kinematics. *Journal of Graphics Tools*, vol. 10, pages 37–49, 2004. ISSN 1086-7651.

BUSSO, C., DENG, Z., GRIMM, M., NEUMANN, U. and NARAYANAN, S. Rigid head motion in expressive speech animation: Analysis and synthesis. *IEEE Speech Audio Process*, vol. 15(3), pages 1075–1086, 2007a. ISSN 1558-7916.

BUSSO, C., DENG, Z., NEUMANN, U. and NARAYANAN, S. S. Learning expressive human-like head motion sequences from speech. In *Data-Driven 3D Facial Animations* (edited by Z. Deng), pages 113–131. Springer-Verlag Press, 2007b.

CANSTAR BLUE. `http://hairsjdi.org/best-robot-vacuum-for-pet-hair-2017-australia/`, 2018.

CARNEY, D. R., HALL, J. A. and LEBEAU, L. S. Beliefs about the nonverbal expression of social power. *Journal of Nonverbal Behavior*, vol. 29(2), pages 105–123, 2005. ISSN 1573-3653.

CHAN, T. F. and DUBEY, R. A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Trans Robot Autom*, vol. 11(2), pages 286–292, 1995. ISSN 1042-296X.

CHIAVERINI, S. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Trans Robot Autom*, vol. 13(3), pages 398–410, 1997. ISSN 1042-296X.

CLOS, D. and MARTÍNEZ, J. Plataforma Mòbil amb Rodes Esfèriques per al Robot "Lightweight Robot 4" de Kuka Roboter. Technical report, Institute of Industrial and Control Engineering - Technical University of Catalonia, 2007.

COCUZZA, S., PRETTO, I. and DEBEI, S. Novel reaction control techniques for redundant space manipulators: Theory and simulated microgravity tests. *Acta Astronautica*, vol. 68(11-12), pages 1712–1721, 2011. ISSN 0094-5765.

COE, R. It's the effect size, stupid: What effect size is and why it is important. Education-line, 2002.

COLOMÉ, A. and TORRAS, C. Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements. *IEEE/ASME Transactions on Mechatronics*, vol. 20(2), pages 944–955, 2015. ISSN 1083-4435.

CONTI, F. and KHATIB, O. Spanning large workspaces using small haptic devices. In *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, pages 183–188. 2005.

CORRALES, J. A., GÓMEZ, G. J. G., TORRES, F. and PERDEREAU, V. Cooperative tasks between humans and robots in industrial environments. *International Journal of Advanced Robotic Systems*, vol. 9(3), page 94, 2012. ISSN 1729-8814.

CRUMPTON, J. and BETHEL, C. L. Validation of vocal prosody modifications to communicate emotion in robot speech. In *2015 International Conference on Collaboration Technologies and Systems (CTS)*, pages 39–46. 2015.

CRUMPTON, J. and BETHEL, C. L. A survey of using vocal prosody to convey emotion in robot speech. *International Journal of Social Robotics*, vol. 8(2), pages 271–285, 2016. ISSN 1875-4805.

DE SCHUTTER, J., DE LAET, T., RUTGEERTS, J., DECRÉ, W., SMITS, R., AERTBELIËN, E., CLAES, K. and BRUYNINCKX, H. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *The International Journal of Robotics Research*, vol. 26(5), pages 433–455, 2007. ISSN 0278-3649.

DERAKSHAN, N. and EYSENCK, M. W. Anxiety, processing efficiency, and cognitive performance: New developments from attentional control theory. *European Psychologist*, vol. 14(2), pages 168–176, 2009. ISSN 1016-9040.

DIETRICH, A., ALBU-SCHÄFFER, A. and HIRZINGER, G. On continuous null space projections for torque-based, hierarchical, multi-objective manipulation. In *2012 IEEE International Conference on Robotics and Automation*, pages 2978–2985. 2012. ISSN 1050-4729.

DISALVO, C. F., GEMPERLE, F., FORLIZZI, J. and KIESLER, S. All robots are not created equal: The design and perception of humanoid robot heads. In *in Proceedings of the DIS Conference*, pages 321–326. ACM Press, 2002.

DOMINJON, L., LECUYER, A., BURKHARDT, J. M., ANDRADE-BARROSO, G. and RICHIR, S. The "bubble" technique: interacting with large virtual environments using haptic devices with limited workspace. In *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, pages 639–640. 2005.

DUAN, H. and LIU, S. Unmanned air/ground vehicles heterogeneous cooperative techniques: Current status and prospects. *Science China Technological Sciences*, vol. 53(5), pages 1349–1355, 2010.

DUBEY, R. V., EULER, J. A. and BABCOCK, S. M. An efficient gradient projection optimization scheme for a seven-degree-of-freedom redundant robot with spherical wrist. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 28–36 vol.1. 1988.

DUDEK, G., JENKIN, M., MILIOS, E. and WILKES, D. A taxonomy for swarm robots. In *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on*, vol. 1, pages 441–447. IEEE, 1993.

EGELAND, O. Task-space tracking with redundant manipulators. *IEEE Journal on Robotics and Automation*, vol. 3(5), pages 471–475, 1987. ISSN 0882-4967.

ENGLISH, J. D. and MACIEJEWSKI, A. A. On the implementation of velocity control for kinematically redundant manipulators. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 30(3), pages 233–237, 2000. ISSN 1083-4427.

ESPIAU, B., CHAUMETTE, F. and RIVES, P. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, vol. 8(3), pages 313–326, 1992. ISSN 1042-296X.

EYSENCK, H. J. and EYSENCK, S. B. G. *Manual of the Eysenck Personality Questionnaire (junior and adult)*. Hodder and Stoughton, 1975.

FREJEK, M. and NOKLEBY, S. B. A methodology for tele-operating mobile manipulators with an emphasis on operator ease of use. *Robotica*, vol. 31, pages 331–344, 2013. ISSN 1469-8668.

FRITZ, C. O., MORRIS, P. E. and RICHLER, J. J. Effect size estimates: Current use, calculations, and interpretation. *Journal of Experimental Psychology : General*, vol. 141(1), pages 2–18, 2012. ISSN 1939-2222.

GARCÍA, N., SUÁREZ, R. and ROSELL, J. Task-dependent synergies for motion planning of an anthropomorphic dual-arm system. *IEEE Transactions on Robotics*, vol. 33(3), pages 756–764, 2017. ISSN 1552-3098.

GEBHARD, P. Alma: A layered model of affect. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, pages 29–36. ACM, New York, NY, USA, 2005. ISBN 1-59593-093-0.

GLOWINSKI, D., DAEL, N., CAMURRI, A., VOLPE, G., MORTILLARO, M. and SCHERER, K. Toward a minimal representation of affective gestures. *IEEE Trans Affect Comput*, vol. 2(2), pages 106–118, 2011. ISSN 1949-3045.

GOERTZ, R. C. A force reflecting positional servo mechanism. *Nucleonics*, vol. 10, pages 43–45, 1952. ISSN 0096-6207.

GOERZEN, C., KONG, Z. and METTLER, B. A survey of motion planning algorithms from the perspective of autonomous uav guidance. *Journal of Intelligent and Robotic Systems*, vol. 57(1-4), page 65, 2010. ISSN 0921-0296.

GONZALEZ-PACHECO, V., RAMEY, A., ALONSO-MARTIN, F., CASTRO-GONZALEZ, A. and SALICHS, M. A. Maggie: A social robot as a gaming platform. *International Journal of Social Robotics*, vol. 3(4), pages 371–381, 2011. ISSN 1875-4805.

GOODRICH, M. A. and SCHULTZ, A. C. Human-robot interaction: a survey. *Foundations and trends in human-computer interaction*, vol. 1(3), pages 203–275, 2007. ISSN 1551-3963.

GUNES, H., SCHULLER, B., PANTIC, M. and COWIE, R. Emotion representation, analysis and synthesis in continuous space: A survey. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 827–834. IEEE, 2011.

HARNETT, B. M., DOARN, C. R., ROSEN, J., HANNAFORD, B. and BRODERICK, T. J. Evaluation of unmanned airborne vehicles and mobile robotic telesurgery in an extreme environment. *Telemedicine and e-Health*, vol. 14(6), pages 539–544, 2008. ISSN 1530-5627.

HART, S. G. Nasa-Task Load Index (Nasa-TLX); 20 Years Later. In *Human Factors and Ergonomics Society Annual Meeting*, vol. 50. 2006. ISBN 9780945289296.

HERDOCIA, A. H., SHADEMAN, A. and JAGERSAND, M. Unimodal asymmetric interface for teleoperation of mobile manipulators: A user study. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5214–5219. 2012. ISSN 2153-0858.

HIRAI, K., HIROSE, M., HAIKAWA, Y. and TAKENAKA, T. The development of honda humanoid robot. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2, pages 1321–1326. IEEE, 1998.

HOLLERBACH, J. and SUH, K. Redundancy resolution of manipulators through torque optimization. *IEEE Journal on Robotics and Automation*, vol. 3(4), pages 308–316, 1987. ISSN 0882-4967.

HOWARD, R. M. and KAMINER, I. Survey of unmanned air vehicles. In *American Control Conference, Proceedings of the 1995*, vol. 5, pages 2950–2953. IEEE, 1995.

HUBER, F., KONDAK, K., KRIEGER, K., SOMMER, D., SCHWARZBACH, M., LAIACKER, M., KOSSYK, I., PARUSEL, S., HADDADIN, S. and ALBU-SCHÄFFER, A. First analysis and experiments in aerial manipulation using fully actuated redundant robot arm. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3452–3457. IEEE, 2013.

HUDSON, J., ORVISKA, M. and HUNADY, J. People's attitudes to robots in caring for the elderly. *International Journal of Social Robotics*, pages 1–12, 2016. ISSN 1875-4805.

HUGHES, S., MANOJLOVICH, J., LEWIS, M. and GENNARI, J. Camera control and decoupled motion for teleoperation. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 2, pages 1339–1344. IEEE, 2003.

HUI, C., YOUSHENG, C., XIAOKUN, L. and SHING, W. W. Autonomous takeoff, tracking and landing of a uav on a moving ugv using onboard monocular vision. In *Control Conference (CCC), 2013 32nd Chinese*, pages 5895–5901. IEEE, 2013.

IEEE SPECTRUM. `https://spectrum.ieee.org/automaton/robotics/industrial-robots/george-devol-a-life-devoted-to-invention-and-robots`, 2018.

J. LARSEN, R. and DIENER, E. Promises and problems with the circumplex model of emotion. In *Review of Personality and Social Psychology: Emotion*, vol. 13. 1992.

JIRAYUCHAROENSAK, S., PAN-NGUM, S. and ISRASENA, P. Eeg-based emotion recognition using deep learning network with principal component based covariate shift adaptation. *The Scientific World Journal*, vol. 2014, 2014.

JOHNSTON, O. and THOMAS, F. *The Illusion of Life: Disney Animation*. Abbeville Press, New York, 1981. ISBN 0-89659-233-2.

Kaneko, K., Harada, K., Kanehiro, F., Miyamori, G. and Akachi, K. Humanoid robot hrp-3. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2471–2478. IEEE, 2008.

Karg, M., Samadani, A.-A., Gorbet, R., Kuhnlenz, K., Hoey, J. and Kulic, D. Body movements for affective expression: A survey of automatic recognition and generation. *IEEE Trans Affect Comput*, vol. 4(4), pages 341–359, 2013. ISSN 1949-3045.

Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, vol. 5(1), pages 90–98, 1986. ISSN 0278-3649.

Khatib, O., Yokoi, K., Chang, K., Ruspini, D., Holmberg, R. and Casal, A. Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation. In *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, vol. 2, pages 546–553. IEEE, 1996.

Klein, C. A., Chu-Jenq, C. and Ahmed, S. A new formulation of the extended jacobian method and its use in mapping algorithmic singularities for kinematically redundant manipulators. *IEEE Transactions on Robotics and Automation*, vol. 11(1), pages 50–55, 1995. ISSN 1042-296X.

Klein, C. A. and Huang, C. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13(2), pages 245–250, 1983. ISSN 0018-9472.

Kleinke, C. L. Gaze and Eye Contact: A Research Review. *Psychol Bull*, vol. 100(1), pages 78–100, 1986.

Kofman, J., Wu, X., Luu, T. J. and Verma, S. Teleoperation of a robot manipulator using a vision-based human-robot interface. *IEEE Transactions on Industrial Electronics*, vol. 52(5), pages 1206–1219, 2005. ISSN 0278-0046.

KUKA.
`https://www.kuka.com/en-de/products/robot-systems/industrial-robots/lbr-iiwa`, 2018.

Kulic, D. and Croft, E. Physiological and subjective responses to articulated robot motion. *Robotica*, vol. 25, pages 13–27, 2007. ISSN 1469-8668.

Lance, B. J. and Marsella, S. C. A model of gaze for the purpose of emotional expression in virtual embodied agents. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '08, pages 199–206. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2008. ISBN 978-0-9817381-0-9.

Lang, P. J. Behavioral treatment and bio-behavioral assessment: Computer applications. In *Technology in Mental Health Care Delivery Systems* (edited by J. B. Sidowski, J. H. Johnson and T. A. Williams), pages 119–137. Ablex Pub. Corp., 1980.

LANG, P. J., BRADLEY, M. M. and CUTHBERT, B. N. International affective picture system (IAPS): Affective ratings of pictures and instruction manual. Technical Report A-8, The Center for Research in Psychophysiology, University of Florida, Gainesville, FL, 2008.

LI, T., KERMORGANT, O. and KRUPA, A. Maintaining visibility constraints during tele-echography with ultrasound visual servoing. In *2012 IEEE International Conference on Robotics and Automation*, pages 4856–4861. 2012. ISSN 1050-4729.

LIÉGEOIS, A. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Trans Syst, Man, Cybern, Syst*, vol. 7(12), pages 868–871, 1977. ISSN 0018-9472.

LIM, A., OGATA, T. and OKUNO, H. Converting emotional voice to motion for robot telepresence. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 472–479. 2011. ISSN 2164-0572.

LIM, A., OGATA, T. and OKUNO, H. G. Towards expressive musical robots: a cross-modal framework for emotional gesture, voice and music. *EURASIP J Audio, Speech and Music Process*, vol. 2012, page 3, 2012. ISSN 1687-4722.

LOUGHLIN, C. and HUDSON, E. Eye in hand robot vision scores over fixed camera. *Sensor Review*, vol. 3(1), pages 23–26, 1983. ISSN 0260-2288.

LUH, J., WALKER, M. and PAUL, R. Resolved-acceleration control of mechanical manipulators. *IEEE Transactions on Automatic Control*, vol. 25(3), pages 468–474, 1980. ISSN 0018-9286.

MACIEJEWSKI, A. A. and KLEIN, C. A. Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *The International Journal of Robotics Research*, vol. 4(3), pages 109–117, 1985. ISSN 0278-3649.

MALLEM, M., CHAVAND, F. and COLLE, E. Computer-assisted visual perception in teleoperated robotics. *Robotica*, vol. 10(2), pages 93–103, 1992. ISSN 1469-8668.

MANSARD, N., KHATIB, O. and KHEDDAR, A. A unified approach to integrate unilateral constraints in the stack of tasks. *IEEE Transactions on Robotics*, vol. 25(3), pages 670–685, 2009a. ISSN 1552-3098.

MANSARD, N., STASSE, O., EVRARD, P. and KHEDDAR, A. A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–6. 2009b.

MAPLE. *version 2015.1*. Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario, 2017.

MARCHAND, E., CHAUMETTE, F. and RIZZO, A. Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS '96*, vol. 3, pages 1083–1090 vol.3. 1996.

Massie, T. H. and Salisbury, J. K. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME Dynamic Systems and Control Division*, pages 295–301. 1994.

Matlab. *version 9.4.0 (R2018a)*. The MathWorks Inc., Natick, Massachusetts, 2018.

McGraw, K. O. and Wong, S. A common language effect size statistic. *Psychological Bulletin*, vol. 111(2), page 361, 1992. ISSN 0033-2909.

Mehrabian, A. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in Temperament. *Curr Psychol*, vol. 14(4), pages 261–292, 1996. ISSN 0737-8262.

Montepare, J., Koff, E., Zaitchik, D. and Albert, M. The use of body movements and gestures as cues to emotions in younger and older adults. *Journal of Nonverbal Behavior*, vol. 23(2), pages 133–152, 1999. ISSN 1573-3653.

Mori, M., MacDorman, K. F. and Kageki, N. The uncanny valley [from the field]. *IEEE Robotics & Automation Magazine*, vol. 19(2), pages 98–100, 2012. ISSN 1070-9932.

Nakagawa, K., Shinozawa, K., Ishiguro, H., Akimoto, T. and Hagita, N. Motion modification method to control affective nuances for robots. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5003–5008. 2009.

Nakamura, Y., Hanafusa, H. and Yoshikawa, T. Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, vol. 6(2), pages 3–15, 1987. ISSN 0278-3649.

Namvar, M. and Aghili, F. Adaptive force-motion control of coordinated robots interacting with geometrically unknown environments. *IEEE Transactions on Robotics*, vol. 21(4), pages 678–694, 2005. ISSN 1552-3098.

National Institute of Standards and Technology. `https://www.nist.gov/programs-projects/performance-collaborative-robot-systems`, 2018.

Nemec, B. and Zlajpah, L. Null space velocity control with dynamically consistent pseudo-inverse. *Robotica*, vol. 18(5), pages 513–518, 2000. ISSN 0263-5747.

Nielsen, C. W., Goodrich, M. A. and Ricks, R. W. Ecological interfaces for improving mobile robot teleoperation. *IEEE Transactions on Robotics*, vol. 23(5), pages 927–941, 2007. ISSN 1552-3098.

Nishiwaki, K., Kagami, S., Kuniyoshi, Y., Inaba, M. and Inoue, H. Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired zmp. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, pages 2684–2689. IEEE, 2002.

Nomura, T. and Nakao, A. Comparison on identification of affective body motions by robots between elder people and university students: A case study in japan. *International Journal of Social Robotics*, vol. 2(2), pages 147–157, 2010. ISSN 1875-4791.

Nuño, E., Aldana, C. I. and Basañez, L. Task space consensus in networks of heterogeneous and uncertain robotic systems with variable time-delays. *International Journal of Adaptive Control and Signal Processing*, vol. 31(6), pages 917–937, 2017. ISSN 1099-1115.

Osgood, C. E., May, W. H. and Miron, M. S. *Cross-cultural universals of affective meaning*, vol. 1. University of Illinois Press, 1975.

Oswald, A., Proto, E. and Sgroi, D. Happiness and productivity. *IZA Discussion Papers*, vol. 4645, 2009. ISSN 2365-9793.

P., E. and WV., F. Head and body cues in the judgment of emotion: a reformulation. *Percept Mot Skills*, 1967. ISSN 0031-5125.

Palanica, A. and Itier, R. Attention capture by direct gaze is robust to context and task demands. *Journal of Nonverbal Behavior*, vol. 36(2), pages 123–134, 2012. ISSN 1573-3653.

Park, J., Choi, Y., Chung, W. K. and Youm, Y. Multiple tasks kinematics using weighted pseudo-inverse for kinematically redundant manipulators. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 4, pages 4041–4047 vol.4. 2001. ISSN 1050-4729.

Peng, Z. . and Adachi, N. Compliant motion control of kinematically redundant manipulators. *IEEE Transactions on Robotics and Automation*, vol. 9(6), pages 831–836, 1993. ISSN 1042-296X.

Pérez, A. and Rosell, J. An assisted re-synchronization method for robotic teleoperated tasks. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 886–891. 2011. ISSN 1050-4729.

Petrič, T. and Žlajpah, L. Smooth transition between tasks on a kinematic control level: Application to self collision avoidance for two kuka lwr robots. In *2011 IEEE International Conference on Robotics and Biomimetics*, pages 162–167. 2011.

Pierre-Yves, O. The production and recognition of emotions in speech: features and algorithms. *International Journal of Human-Computer Studies*, vol. 59(1?2), pages 157 – 183, 2003. ISSN 1071-5819.

Plutchik, R. E. and Conte, H. R. *Circumplex models of personality and emotions*. American Psychological Association, 1997.

Posner, J., Russell, J. A. and Peterson, B. S. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology*, vol. 17(3), pages 715–734, 2005. ISSN 0954-5794.

Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R. and Ng, A. Y. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*. 2009.

RAO, R., KUMAR, V. and TAYLOR, C. Visual servoing of a UGV from a UAV using differential flatness. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1, pages 743–748 vol.1. 2003.

RAUNHARDT, D. and BOULIC, R. Progressive clamping. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4414–4419. IEEE, 2007a.

RAUNHARDT, D. and BOULIC, R. Progressive clamping. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 4414–4419. 2007b. ISSN 1050-4729.

REMINGTON, N. A., FABRIGAR, L. R. and VISSER, P. S. Reexamining the circumplex model of affect. *Journal of personality and social psychology*, vol. 79(2), page 286, 2000. ISSN 0022-3514.

RIGATO, S. and FARRONI, T. The role of gaze in the processing of emotional facial expressions. *Emotion Review*, 2013. ISSN 1754-0739.

ROBOTICS @SNU. `http://roboticslab.snu.ac.kr/m/html/sub02_04_03.php`, 2018.

ROSELL, J., PÉREZ, A., AKBARI, A., MUHAYYUDDIN, PALOMO, L. and GARCÍA, N. The kautham project: A teaching and research tool for robot motion planning. In *IEEE International Conference on Emerging Technologies and Factory Automation*, pages 1–8. Institute of Electrical and Electronics Engineers (IEEE), 2014.

RUESCH, A., MERSHA, A. Y., STRAMIGIOLI, S. and CARLONI, R. Kinetic scrolling-based position mapping for haptic teleoperation of unmanned aerial vehicles. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3116–3121. 2012. ISSN 1050-4729.

RUSSEL, J. A circumplex model of affect. *J. Personality and Social Psychology*, vol. 39, pages 1161–78, 1980. ISSN 0022-3514.

RUSSELL, J. A. A circumplex model of affect. *Journal of Personality and Social Psychology*, vol. 39, pages 1161–1178, 1980. ISSN 0022-3514.

SAAKES, D., CHOUDHARY, V., SAKAMOTO, D., INAMI, M. and LGARASHI, T. A teleoperating interface for ground vehicles using autonomous flying cameras. In *Artificial Reality and Telexistence (ICAT), 2013 23rd International Conference on*, pages 13–19. 2013.

SAERBECK, M. and BARTNECK, C. Perception of affect elicited by robot motion. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pages 53–60. 2010.

SALICHS, M. and MALFAZ, M. Using emotions on autonomous agents. the role of happiness, sadness and fear. *Proceedings of AISB'06: Adaptation in Artificial and Biological Systems*, vol. 1, 2006.

SALOVEY, P., ROTHMAN, A. J., DETWEILER, J. B. and STEWARD, W. T. Emotional states and physical health. *American Psychologist*, vol. 55(1), pages 110–121, 2000. ISSN 0003-066X.

SAMANTA, A. and GUHA, T. On the role of head motion in affective expression. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2886–2890. 2017.

SANDER, D., GRANDJEAN, D., KAISER, S., WEHRLE, T. and SCHERER, K. R. Interaction effects of perceived gaze direction and dynamic facial expression: Evidence for appraisal theories of emotion. *European Journal of Cognitive Psychology*, vol. 19(3), pages 470–480, 2007. ISSN 0954-1446.

SASKA, M., KRAJNIK, T. and PFEUCIL, L. Cooperative μ UAV-UGV autonomous indoor surveillance. In *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, pages 1–6. 2012.

SCHERER, K. R. Expression of emotion in voice and music. *Journal of Voice*, vol. 9(3), pages 235–248, 1995. ISSN 0892-1997.

SCHIFFERSTEIN, H. N. J., TALKE, K. S. S. and OUDSHOORN, D.-J. Can ambient scent enhance the nightlife experience? *Chemosensory Perception*, vol. 4(1-2), pages 55–64, 2011. ISSN 1936-5802.

SCHREIBER, G., STEMMER, A. and BISCHOFF, R. The fast research interface for the kuka lightweight robot. 2010.

SCIAVICCO, L. and SICILIANO, B. A solution algorithm to the inverse kinematic problem for redundant manipulators. *IEEE Journal on Robotics and Automation*, vol. 4(4), pages 403–410, 1988. ISSN 0882-4967.

SENSABLE. `http://www.delfthapticslab.nl/device/phantom-omni/`, 2019.

SENTIS, L. and KHATIB, O. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, vol. 2(4), pages 505–518, 2005. ISSN 0219-8436.

SETO, F., KOSUGE, K. and HIRATA, Y. Self-collision avoidance motion control for human robot cooperation system using robe. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3143–3148. 2005. ISSN 2153-0858.

SHAPIRO, S. S. and WILK, M. B. An analysis of variance test for normality (complete samples). *Biometrika*, vol. 52(3/4), pages pp. 591–611, 1965. ISSN 0006-3444.

SHARMA, M., HILDEBRANDT, D., NEWMAN, G., YOUNG, J. E. and ESKICIOGLU, R. Communicating affect via flight path exploring use of the laban effort system for designing affective locomotion paths. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 293–300. 2013. ISSN 2167-2121.

SHAVER, P., SCHWARTZ, J., KIRSON, D. and O'CONNOR, C. Emotion knowledge: further exploration of a prototype approach. *Journal of personality and social psychology*, vol. 52(6), page 1061, 1987. ISSN 0022-3514.

SHERIDAN, T. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, 1992. ISBN 9780262193160.

SICILIANO, B. Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, vol. 3(3), pages 201–212, 1990. ISSN 0921-0296.

Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 1846286417, 9781846286414.

Siciliano, B. and Slotine, J.-J. A general framework for managing multiple tasks in highly redundant robotic systems. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, vol. 2, pages 1211–1216. 1991.

Smith, C., Karayiannidis, Y., Nalpantidis, L., Gratal, X., Qi, P., Dimarogonas, D. V. and Kragic, D. Dual arm manipulation?a survey. *Robotics and Autonomous systems*, vol. 60(10), pages 1340–1353, 2012. ISSN 0921-8890.

SoftBank Robotics. `https://www.softbankrobotics.com/emea/en/robots/pepper`, 2018.

van Straten, C. L., Smeekens, I., Barakova, E., Glennon, J., Buitelaar, J. and Chen, A. Effects of robots' intonation and bodily appearance on robot-mediated communicative treatment outcomes for children with autism spectrum disorder. *Personal and Ubiquitous Computing*, 2017. ISSN 1617-4917.

Şucan, I. A., Moll, M. and Kavraki, L. E. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, vol. 19(4), pages 72–82, 2012. ISSN 1070-9932. `http://ompl.kavrakilab.org`.

Sugano, S. and Kato, I. Wabot-2: Autonomous robot with dexterous finger-arm–finger-arm coordination control in keyboard performance. In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, pages 90–97. IEEE, 1987.

Sugiura, H., Gienger, M., Janssen, H. and Goerick, C. Real-time collision avoidance with whole body motion control for humanoid robots. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2053–2058. 2007. ISSN 2153-0858.

Tang, D. and Schmeichel, B. J. Look me in the eye: Manipulated eye gaze affects dominance mindsets. *Journal of Nonverbal Behavior*, vol. 39(2), pages 181–194, 2015. ISSN 1573-3653.

Tanner, H. G. Switched UAV-UGV cooperation scheme for target detection. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3457–3462. 2007. ISSN 1050-4729.

Tanner, H. G. and Christodoulakis, D. K. Cooperation between aerial and ground vehicle groups for reconnaissance missions. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 5918–5923. 2006. ISSN 0191-2216.

Tanner, H. G., Loizou, S. G. and Kyriakopoulos, K. J. Nonholonomic navigation and control of cooperating mobile manipulators. *IEEE Transactions on robotics and automation*, vol. 19(1), pages 53–64, 2003.

Tapus, A., Mataric, M. and Scasselati, B. Socially assistive robotics [grand challenges of robotics]. *IEEE Robot Autom Mag*, vol. 14(1), pages 35–42, 2007. ISSN 1070-9932.

TAPUS, A. and MATARIC, M. J. Emulating empathy in socially assistive robotics. In *In Proceedings of the AAAI Spring Symposium on Multidisciplinary Collaboration for Socially Assistive Robotics*. 2007.

TCHOŃ, K. Repeatable, extended jacobian inverse kinematics algorithm for mobile manipulators. *Systems & Control Letters*, vol. 55(2), pages 87 – 93, 2006. ISSN 0167-6911.

TORRES, F., PUENTE, S. and DÍAZ, C. Automatic cooperative disassembly robotic system: Task planner to distribute tasks among robots. *Control Engineering Practice*, vol. 17(1), pages 112 – 121, 2009. ISSN 0967-0661.

UNUMA, M., ANJYO, K. and TAKEUCHI, R. Fourier principles for emotion-based human figure animation. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 91–96. ACM, New York, NY, USA, 1995. ISBN 0-89791-701-4.

VTT. `http://virtual.vtt.fi/virtual/proj2/multimedia/index.html`, 2016.

WARE, C. and OSBORNE, S. Exploration and virtual camera control in virtual three dimensional environments. *SIGGRAPH Comput. Graph.*, vol. 24(2), pages 175–183, 1990. ISSN 0097-8930.

WENZEL, K. E., MASSELLI, A. and ZELL, A. Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle. *J. Intell. Robotics Syst.*, vol. 61(1-4), pages 221–238, 2011. ISSN 0921-0296.

WESSLER, R. and LOEVINGER, J. Reviews: Jackson, d. n. personality research form. goshen, n. y.: Research psychologists press, 1967. *American Educational Research Journal*, vol. 6(2), pages 302–306, 1969.

WHITE, G., BHATT, R., TANG, C. P. and KROVI, V. Experimental evaluation of dynamic redundancy resolution in a nonholonomic wheeled mobile manipulator. *IEEE/ASME Trans Mechatronics*, vol. 14(3), pages 349–357, 2009. ISSN 1083-4435.

WHITNEY, D. E. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, vol. 10(2), pages 47–53, 1969. ISSN 0536-1540.

WILCOXON, F. Individual comparisons by ranking methods. *Biometrics Bulletin*, vol. 1(6), pages pp. 80–83, 1945. ISSN 0099-4987.

YAMAMOTO, Y. and YUN, X. Coordinating locomotion and manipulation of a mobile manipulator. In *Decision and Control, 1992., Proceedings of the 31st IEEE Conference on*, pages 2643–2648. IEEE, 1992.

YOSHIDA, E., KANOUN, O., ESTEVES, C., LAUMOND, J.-P. and YOKOI, K. Task-driven support polygon reshaping for humanoids. In *Humanoids*, pages 208–213. 2006.

YOSHIKAWA, T. Dynamic manipulability of robot manipulators. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, pages 1033–1038. 1985.

ZANCHETTIN, A. M. and ROCCO, P. A general user-oriented framework for holonomic redundancy resolution in robotic manipulators using task augmentation. *IEEE Transactions on Robotics*, vol. 28(2), pages 514–521, 2012. ISSN 1552-3098.

ZHENG, M., MOON, A., CROFT, E. A. and MENG, M. Q.-H. Impacts of robot head gaze on robot-to-human handovers. *International Journal of Social Robotics*, vol. 7(5), pages 783–798, 2015. ISSN 1875-4805.