UNIVERSITAT POLITÈCNICA DE CATALUNYA

Programa de Doctorat:

AUTOMÀTICA, ROBÒTICA I VISIÓ

Tesi Doctoral

Multi-Robot Cooperative Platform: a task-oriented teleoperation paradigm

Albert Hernansanz Prats

Director: Josep Amat
Ponent: Alícia Casals

Desembre de 2015

Universitat Politècnica de Catalunya
PhD Program:
Automatic Control, Robotics and Computer Vision (ARV)

Automatic Control Department (ESAII)

Research Group GRINS - Intelligent Robots and Systems

Multi-Robot Cooperative Platform:
a task-oriented teleoperation paradigm

by Albert Hernansanz Prats

Doctoral Thesis
Advisor: Josep Amat
Internal Examiner: Alícia Casals

Barcelona, December 2015

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola de Doctorat

Assessment results for the doctoral thesis

Academic year:

Full name

Doctoral programme

Structural unit in charge of the programme

Decision of the committee

In a meeting with the examination committee convened for this purpose, the doctoral candidate presented the topic of his/her doctoral thesis entitled

_____

_____.

Once the candidate had defended the thesis and answered the questions put to him/her, the examiners decided to award a mark of:

☐ UNSATISFACTORY ☐ SATISFACTORY ☐ GOOD ☐ VERY GOOD

(Full name and signature) Chairperson	(Full name and signature) Secretary

(Full name and signature) Member	(Full name and signature) Member	(Full name and signature) Member

_____, _____

The votes of the members of the examination committee were counted by the Doctoral School at the behest of the Doctoral Studies Committee of the UPC, and the result is to award the CUM LAUDE DISTINCTION:

☐ YES ☐ NO

(Full name and signature) Chair of the Standing Committee of the Doctoral School	(Full name and signature) Secretary of the Standing Committee of the Doctoral School

Barcelona, _____

International doctoral degree statement

As the secretary of the examination committee, I hereby state that the thesis was partly (at least the abstract and the conclusions) defended in a language commonly used in scientific communication in the field that is not an official language of Spain. This does not apply if the stay, report or expert is from a Spanish-speaking country.

(Nom, cognoms i signatura) Secretary of the Examination Commitee

Abstract

Multi-Robot Cooperative Platform: a task-oriented teleoperation paradigm

Albert Hernansanz Prats

This thesis proposes the study and development of a teleoperation system based on multi-robot cooperation under the task oriented teleoperation paradigm: Multi-Robot Cooperative Platform, MRCP. In standard teleoperation, the operator uses the master devices to control the remote slave robot arms. These arms reproduce the desired movements and perform the teleoperated task. With the developed work, the operator can virtually manipulate an object. MRCP automatically generates the arms orders to perform the task. The operator does not have to solve situations arising from possible restrictions that the slave arms may have. The research carried out is therefore aimed at improving the accuracy of teleoperation tasks in complex environments, particularly in the field of robot assisted minimally invasive surgery. This field requires patient safety and the workspace entails many restrictions to teleoperation. MRCP can be defined as a platform composed of several robots that cooperate automatically to perform a teleoperated task, creating a robotic system with increased capacity (workspace volume, accessibility, dexterity ...). This cooperation is based on transferring the task between robots when necessary to enable a smooth task execution. The MRCP control evaluates the suitability of each robot to continue with the ongoing task and the optimal time to execute a task transfer between the current selected robot and the best candidate to continue with the task. From the operators point of view, MRCP provides an interface that enables teleoperation though the task-oriented paradigm: operator orders are translated into task actions instead of robot orders.

This thesis is structured as follows: The first part is dedicated to review current solutions in teleoperation of complex tasks and compare them with those proposed in this research. The second part of the thesis presents and reviews in depth the different

evaluation criteria considered to determine the suitability of each robot to continue with the execution of the on-going task, considering the configuration of the robots and emphasizing the criteria of dexterity and manipulability. The study reviews the different required control algorithms to enable task-oriented telemanipulation. This proposed teleoperation paradigm is transparent to the operator.

Then, the Thesis presents and analyses several experimental results using MRCP in the field of minimally invasive surgery. These experiments study the effectiveness of MRCP in diverse tasks requiring the cooperation of two hands. The task taken as testbed and benchmark consists in a suture in minimally invasive surgery. The analysis of results is done in terms of execution time, economy of movement, quality and patient safety (potential damage produced by undesired interaction between the tools and the vital tissues of the patient). The final part proposes the implementation of different virtual aids and restrictions (guided teleoperation based on haptic visual and audio feedback, protection of restricted workspace regions, etc.) using the task oriented teleoperation paradigm. A framework is defined for implementing and applying a basic set of virtual aids and constraints within the framework of a virtual simulator for laparoscopic abdominal surgery. The set of experiments have allowed the validation of the developed work. The study has revealed the influence of virtual aids in the learning and training process of laparoscopic techniques. It has also demonstrated the improvement of learning curves, which paves the way for its implementation as a methodology for training new surgeons.

Resum

Multi-Robot Cooperative Platform: a task-oriented teleoperation paradigm

Albert Hernansanz Prats

Aquesta tesi doctoral proposa l'estudi i desenvolupament d'un sistema de teleoperació basat en la cooperació multi-robot sota el paradigma de la teleoperació orientada a tasca: Multi-Robot Cooperative Paradigm, MRCP. En la teleoperació clàssica, l'operador utilitza els telecomandaments perquè els braos robots reprodueixin els seus moviments i es realitzi la tasca desitjada. Amb el treball realitzat, l'operador pot manipular virtualment un objecte i és mitjanant el MRCP que s'adjudica a cada bra les ordres necessàries per realitzar la tasca, sense que l'operador hagi de resoldre les situacions derivades de possibles restriccions que puguin tenir els braos executors. La recerca desenvolupada està doncs orientada a millorar la teleoperació en tasques de precisió en entorns complexos i, en particular, en el camp de la cirurgia mínimament invasiva assistida per robots. Aquest camp imposa condicions de seguretat del pacient i l'espai de treball comporta moltes restriccions a la teleoperació. MRCP es pot definir com a una plataforma formada per diversos robots que cooperen de forma automàtica per dur a terme una tasca teleoperada, generant un sistema robòtic amb capacitats augmentades (volums de treball, accessibilitat, destresa, ...). La cooperació es basa en transferir la tasca entre robots a partir de determinar quin és aquell que és més adequat per continuar amb la seva execució i el moment òptim per realitzar la transferència de la tasca entre el robot actiu i el millor candidat a continuar-la. Des del punt de vista de l'operari, MRCP ofereix una interfície de teleoperació que permet la realització de la teleoperació mitjanant el paradigma d'ordres orientades a la tasca: les ordres es tradueixen en accions sobre la tasca en comptes d'estar dirigides als robots. Aquesta tesi està estructurada de la següent manera: Primerament es fa

una revisió de l'estat actual de les diverses solucions desenvolupades actualment en el camp de la teleoperació de tasques complexes, comparant-les amb les proposades en aquest treball de recerca. En el segon bloc de la tesi es presenten i s'analitzen a fons els diversos criteris per determinar la capacitat de cada robot per continuar l'execució duna tasca, segons la configuració del conjunt de robots i fent especial èmfasi en el criteri de destresa i manipulabilitat. Seguint aquest estudi, es presenten els diferents processos de control emprats per tal d'assolir la telemanipulació orientada a tasca de forma transparent a l'operari. Seguidament es presenten diversos resultats experimentals aplicant MRCP al camp de la cirurgia mínimament invasiva. En aquests experiments s'estudia l'eficcia de MRCP en diverses tasques que requereixen de la cooperació de dues mans. S'ha escollit una tasca tipus: sutura amb tècnica de cirurgia mínimament invasiva. L'anàlisi es fa en termes de temps d'execució, economia de moviment, qualitat i seguretat del pacient (potencials danys causats per la interacció no desitjada entre les eines i els teixits vitals del pacient).

Resumen

Multi-Robot Cooperative Platform: a task-oriented teleoperation paradigm

Albert Hernansanz Prats

Esta tesis doctoral propone el estudio y desarrollo de un sistema de teleoperación basado en la cooperación multi-robot bajo el paradigma de la teleoperación orientada a tarea: Multi-Robot Cooperative Paradigm, MRCP. En la teleoperación clásica, el operador utiliza los mandos para que los brazos robots reproduzcan sus movimientos y se realice la tarea deseada. Con el trabajo realizado, el operador puede manipular virtualmente un objeto y es mediante el MRCP que se adjudica a cada brazo las órdenes necesarias para realizar la tarea, sin que el operador deba resolver las situaciones derivadas de posibles restricciones que puedan tener los brazos ejecutores. La investigación desarrollada está pues orientada a mejorar la teleoperación en tareas de precisión en entornos complejos y, en particular, en el campo de la cirugía mínimamente invasiva asistida por robots. Este campo impone condiciones de seguridad del paciente y el espacio de trabajo conlleva muchas restricciones a la teleoperación. MRCP se puede definir como una plataforma formada por varios robots que cooperan de forma automática para llevar a cabo una tarea teleoperada, generando un sistema robótico con capacidades aumentadas (volmenes de trabajo, accesibilidad, destreza, ...). La cooperación se basa en transferir la tarea entre robots a partir de determinar cuál es el que es más adecuado para continuar con su ejecución y el momento óptimo para realizar la transferencia de la tarea entre el robot activo y el mejor candidato a continuarla . Desde el punto de vista del operario, MRCP ofrece una interfaz de teleoperación que permite la realización de la teleoperación mediante el paradigma de órdenes orientadas a la tarea: las órdenes se traducen en acciones sobre la tarea en vez de estar dirigidas a los robots. Esta tesis está estructurada de

la siguiente manera: Primeramente se hace una revisión del estado actual de las diversas soluciones desarrolladas actualmente en el campo de la teleoperación de tareas complejas, comparándolas con las propuestas en este trabajo de investigación. En el segundo bloque de la tesis se presentan y se analizan a fondo los diversos criterios para determinar la capacidad de cada robot para continuar la ejecución de una tarea, segn la configuración del conjunto de robots y haciendo especial nfasis en el criterio de destreza y manipulabilidad. Siguiendo este estudio, se presentan los diferentes procesos de control empleados para alcanzar la telemanipulación orientada a tarea de forma transparente al operario. Seguidamente se presentan varios resultados experimentales aplicando MRCP el campo de la cirugía mínimamente invasiva. En estos experimentos se estudia la eficacia de MRCP en diversas tareas que requieren de la cooperación de dos manos. Se ha escogido una tarea tipo: sutura con tcnica de cirugía mínimamente invasiva. El análisis se hace en trminos de tiempo de ejecución, economía de movimiento, calidad y seguridad del paciente (potenciales daos causados por la interacción no deseada entre las herramientas y los tejidos vitales del paciente). Finalmente se ha estudiado el uso de diferentes ayudas y restricciones virtuales (guiado de la teleoperación vía retorno háptico, visual o auditivo, protección de regiones del espacio de trabajo, etc) dentro del paradigma de teleoperación orientada a tarea. Se ha definiendo un marco de aplicación base e implementando un conjunto de restricciones virtuales dentro del marco de un simulador de cirugía laparoscopia abdominal. El conjunto de experimentos realizados han permitido validar el trabajo realizado. Este estudio ha permitido determinar la influencia de las ayudas virtuales en el proceso de aprendizaje de las tcnicas laparoscópicas. Se ha evidenciado una mejora en las curvas de aprendizaje y abre el camino a su implantación como metodología de entrenamiento de nuevos cirujanos.

Acknowledgments

Primer de tot m'agradaria agrair al meu director de tesi, el Prof. Josep Amat per l'oportunitat de poder desenvolupar aquesta tesi i per introduir-me dins el món de la recerca. Seguidament, vull agrair a la Prof. Alícia Casals per haver cregut en mi, deixant-me desenvolupar-me com a persona i com a professional. La força i empenta d'ambdós eś admirable i contagiosa.

De la col·laboració desinteressada de molts cirurgians i residents he pogut aprendre moltes de les coses que m'han permès evolucionar en la meva recerca. A tots, moltes gràcies per la vostra col·laboració, paciència i bona predisposició. En especial, a en Dr. Ramon Rovira amb qui hem treballat moltes hores i, espero, continuarem treballant-ne moltes més. I a la Dra. Paloma Huguet, que ha fet de nexe d'unió amb el món de la ginecologia. Tots dos heu fet un xic més entenedor aquesta complicada disciplina per a ulls d'un enginyer.

Desidero ringraziare il professor Paolo Fiorini per l'ottima accoglienza che ho ricevuto presso il laboratorio Altair di Verona. La sua professionalità e la sua esperienza mi hanno motivato e mi hanno fatto rendere il massimo durante la mia permanenza nel laboratorio. Nonostante fossi l'ultimo arrivato non mi ha mai fatto mancare quella fiducia e quel sostengo che mi hanno permesso di aggiungere il mio piccolo contributo alla ricerca. Tutti i colleghi che ho incontrato là mi hanno fatto sentire a casa fin dal primo giorni. Un ricordo speciale va a Bogdan Maris (respected professor) che fin dall'inizio si è occupato e preoccupato per me e a Lorenzo Grespan, che mi ha dato utili consigli. Infine a tutto il gruppo di ricerca VAVF (Lorenza Gasperotti, Davide Zerbato e Michele Scandola) che ha creduto nelle mie idee folli, una dopo l'altra. Davide, piu que PREGO

M'agradaria agrair als companys del laboratori que han estat al meu costat durant el decurs d'aquesta tesi i de tots els anys al laboratori. No posaré tots els noms per que de segur em deixaria alguns i seria totalment injust. De tots he aprés, amb tots he rigut i amb tots he compartit moments bons i no tant bons. M'agradaria nomenar a uns pocs que han deixat més empremta en la meva vida: Primer de tot a n'Ezio

Capelino, qui té un sentit molt pràctic de la vida. També a n'Agustín Navarro amb qui vaig tenir el privilegi de treballar. A en Joan Basomba de qui hauria d'haver aprés a treballar amb ordre i ser més humil. Per últim i molt especialment a n'Alberto Rodríguez i al Xavier Giralt amb els qui he compartit infinites hores, moments de brownies, de idees sense sentit, cafès al pati de la FME i de situacions surrealistes. Tots ells han esdevingut amics més enllà de companys.

Finalment, al que jo anomeno el consell directiu (Victor, Carles, Pep, Marc i Guillermo) juntament amb els $Q3$ de Benasque i al Marc Garcia: sense amics, no hi ha vida. I, com no, als meus pares els he d'agrair masses coses però, dins aquest context, a haver-me educat en la cultura de l'esforç i el treball. A vegades, fins i tot, massa. I a mon germà per ser una constant en la que fixar-se, aprendre i no perdre mai el nord. *Gracias Ferran, Marieta y Guillermo.*

Eva, t'ha tocat el paper de puntxar per que aquesta tesi veiés la llum. La teva disciplina i rigor científic han fet que tot el conjunt hagi millorat enormement. Tenir una matemàtica al costat és el que té. Moltes gràcies per tot, de veres.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Teleoperation, can be defined as the operation of a robot or robots guided at distance by a human operator. Such robotic systems are designed to interact with remote and/or hostile and hazardous environments (e.g. nuclear, spatial exploration, submarine, etc). Teleoperation also enables the execution of highly complex tasks in which some human intervention is required, thus, involving a human in the control and decision loop. In recent years, teleoperation has increased its presence in precision and dexterous tasks; for instance in minimally invasive surgery. Teleoperation provides some assistance for improving task execution: increase accessibility and manoeuvrability, improve operators skills (e.g. tremor reduction, motion scaling), decrease fatigue, etc. It also enables the remote intervention of a specialist (e.g. a surgeon in a concrete surgical procedure).

However, teleoperation presents several problems and challenges still to be solved. From the control point of view, time delays and their consequent control and stability problems are still an open issue. Control complexity increases in applications in which a teleoperated system is composed of multiple robots, presenting each robot a different time delay, [1–4]. Robot navigation in complex, dynamic and clustered environments represents another open research field, [5–7]. The lack of remote perception, jointly with the robot complexity and kinematic discrepancy between the operator and the robot (interaction between master and slave, number and distribution of joints in robots, etc.) causes a poor human teleoperation performance, [8,9]. Current solutions to these problems are based on improving human-robot interfaces, HRI, to facilitate a

human friendly interaction with the teleoperated system, designing new master devices and user interfaces, [10–15]. Another research field focuses the efforts on increasing human dexterity when teleoperating using virtual fixtures, [16, 17]. Examples of these synthetic aids are force feedback, motion scaling and virtual guidance.

1.0.1 Motivation

The continuous evolution of teleoperation expands its range of applicability to areas as diverse as space, underwater, maintenance or surgery. All of them present specific requirements and workspace characteristics. For instance, space robots have to cope with the lack of gravity and communication delays, whereas, the main challenges in surgical robotics are related to the requirements on dexterity, accuracy and patient safety. These increasing demands impose the development of new approaches and platforms in teleoperation.

Multi-robot systems present well known benefits, increasing the range of applicability with respect to those based on a single robot. Heavy, bulky and high volume objects can be co-manipulated by means of multiple robot systems. However, they present control challenges, increasing the control complexity: collision control on shared workspaces, synchronization issues and closed kinematic chain problems. The use of multi-robot platforms in teleoperation increases the mentioned control challenges, introducing variable time delays, require more strict collision control (no path planning in teleoperation), etc.

In spite of the progress going on in the field of autonomous robots, the increasing trend of their performance curve shows an asymptotic shape, which by now, prevents the execution of tasks that present a certain degree of uncertainness. For this reason, teleoperation tries to make up for the lack of 100% reliability of autonomous robots, bringing the intelligence, as well as the planning and improving capabilities of humans, to the location where a task takes place, even if it is an inaccessible, hazardous or hostile environment. Therefore, teleoperation does not present limitations from the point of view of intelligence (human is behind the control loop), but, instead, the

use of robot arms present kinematic limitations since teleoperated arms usually do not have the same accessibility and degrees of freedom (in number and distribution) than those of humans operating with their own hands. Not to mention, the lack of propioceptive perception compared to that of humans, even though a teleoperated system can be provided with some haptic feedback, which supplies part of this lost sensitivity.

This thesis aims to advance in the development of teleoperated systems based on multiple cooperative robots. In order to evaluate a teleoperation workstation composed of two master devices and two or more teleoperated arms, providing a normal operation with two hands, some criteria can be established as: efficiency, ergonomics and safety. A measure of efficiency can be obtained, even numerically, based on a comparative analysis of execution time. That is, for different strategies, the abilities achieved by some operators, using an assisted teleoperation workstation, can be evaluated. A second factor, ergonomics, constitutes a more subjective measure independently of efficiency. It responds to the easiness with which the user performs a task, and the potential improvements in commodity or assistance does not always result in a reduction of execution time or quality. They can, for instance, reduce tiredness, which is not a measurable parameter. And third, the achieved reliability can also be numerically evaluated by statistically assessing the number of failures, inaccuracies or dysfunctions produced during the repetitive execution of a teleoperated task.

This thesis focusses on the development of a methodology to facilitate the teleoperation of multirobot systems, proposing and developing the Multi Robot Cooperative Platform, MRCP. MRCP adapts the task oriented paradigm, generating a new teleoperation mode where diverse robots cooperate to execute a teleoperated task by offering the operator a new control interface. The task oriented paradigm is based on focusing the operator actions into the task. In standard teleoperation, the operator actions are applied to the slave robots. On the contrary, a task oriented approach applies the operator orders to the task. Let's use a telemanipulation task to illustrate the differences between both approaches. Moving the arm in a certain direction, or

opening/closing a gripper, are standard orders applied to the slave robots. Moving an object to a certain position is a task oriented order. The task oriented concept represents a higher abstraction level.

The increasing demands on teleoperation have motivated the development of a new approach for dexterous teleoperation tasks, the Multi-Robot Cooperative Platform. The combined use of multi-robot systems acting as a unique robot that increases the range of applicability of teleoperation, jointly with the use of the task oriented paradigm improving the operator skills, are the starting points of this research.

1.0.2 Objectives

This research work aims at contributing into teleoperation tasks that demand multi-robot slave stations. The proposed solution, the Multi-Robot Cooperative Platform, MRCP, is based on a task-oriented teleoperation platform. In standard teleoperation, the operator interacts with master devices to remotely guide, one by one, the slave robots. In task oriented teleoperation, the operator's orders are interpreted as task actions. MRCP proposes a new approach that combines multi-robot teleoperation with the task oriented paradigm.

Therefore, the process of task allocation in the slave side, from the orders given from a unique master by the human operator requires an automatic sequencing, deciding dynamically which should be the active robot among those in the slave multirobot system. This passes through the achievement of the following objectives, which guarantee the continuity of the task by selecting the most adequate robot at every instant.

1. **Definition of a new teleoperation approach based on the task-oriented paradigm** The improvement of operator's dexterity is a challenging open field in teleoperation. The lack of remote perception jointly with the kinematic discrepancy of most of the masters with respect to humans decreases the human manual ability. This drawback is more evident when a fine telemanipulation is required in a multi-robot environment. A proposed solution in this research work is to increase the operators dexterity by means of applying the task-oriented paradigm.

This paradigm focuses the control actions on a task rather than on the agents (slave robots in teleoperation).

2. **Development of an open control architecture to enable cooperative multi-robot teleoperation** The proposed multi-robot platform is designed to allow the execution of a wide range of teleoperation tasks. This implies the use of various types of master devices and slave units (number and type of robots). Consequently, the control architecture must be capable of controlling an open multi-robot teleoperation platform. A multi-layered control architecture has been developed, which is scalable to admit new evaluation indices (robot and task dependant) and the use of multiple robots and master devices of various types. The high level control must receive the master orders, evaluate the slave robots and compute the orders to be sent to the robot controllers.

3. **Definition of a framework to evaluate and compare robots suitability** The selection of the most adequate robot to execute a task, within a multi-robot cell, requires the study and identification of the causes that prevent them to continue with the execution of the on-going task. A classification has been established based on the nature of the cause that prevents a robot to execute a task: intrinsic, extrinsic and task dependant. A set of evaluation indices working in different spaces have been defined (Cartesian workspace and configuration space are examples of the evaluation spaces). Following with the study of robots suitability analysis, a common framework to compare different evaluation indices has been developed.

4. **Development of a set-up to test and validate the proposed platform** The development and implementation of a real set-up has accompanied all the phases of this research work with the goal to test and validate the theoretical part of the research.

   Requisites of the developed software are the implementation of the framework for evaluation and comparison of the robots suitability, including the indices themselves, the control algorithm and the action planner. The MRCP control

has been developed under a robotic simulation platform, the Robotic Proximities Query package. RPQ, developed at the beginning of this research, was designed to emulate robotic systems, generate realistic graphical representations and implement an efficient computation of distance queries in real-time.

5. **Inclusion of Virtual Fixtures into the teleoperation paradigm** The use of Virtual Fixtures, VF, to improve operators performance has been studied, obtaining promising results. VF can be defined as a set of synthetically generated aids. Virtual forces, audio guidance and motion scaling are examples of virtual fixtures. The inclusion of these virtual fixtures in the task-oriented paradigm used in MRCP seems to be a natural development step to improve the performance of human operators. Measures like execution time, quality and economy of movement have been used to test the validity of these proposed aids. A task-oriented VF paradigm has been proposed and developed.

6. **Validation of MRCP and operator performance in dexterous tasks** In order to evaluate the proposed paradigm, several tests have been developed. These tests can be classified depending on their objective: evaluate the proposed teleoperation paradigm, validate the improvement of the operators performance and study the effect of the introduction of VF in the teleoperation control loop. Several experimental set-ups have been developed to fulfil the objectives of each experiment.

1.1 Introduction to the MRCP concept

The proposed Multi-Robot Cooperative Platform for task oriented teleoperation is designed as a new approach for complex telemanipulation tasks. It is based on a single master station commanding a slave station composed of a multi-robot cooperative system. MRCP contributes to those teleoperated tasks which cannot be accomplished with a single robot due to its own limitations (joint space, singularities, etc.) or to the task requirements and workspace characteristics (accessibility, manoeuvrability,

etc.). The proposed solution is based on a multi-robot slave station where the robots act as a unique slave robot with augmented capabilities. The set of robots operate in an automatic and transparent mode to the operator, transferring the task execution from one to another when necessary. MRCP is introduced as a high level module inside the teleoperation control loop, receiving the master orders and generating the individual robot control orders, Fig.1.1. MRCP is designed for teleoperation, in which no path planning is possible. The control algorithm infers the next operator orders so as to be able to act in advance to avoid blocking the task. The uncertainness on the system inputs (desired operators commands) preclude ensuring optimal solutions (e.g. minimum number of task transfers), but the control algorithm is designed to obtain the best solution from the available information.

MRCP can be studied from the control and from the operator's point of view:

- **MRCP from the control point of view** The MRCP control must ensure an effective multi-robot cooperative teleoperation. This control is based on a sequential algorithm that determines the robots suitability to continue with the on-going task, selects the most appropriate instant of time to execute a task transfer between robots and computes the robot actions to enable the continuation of the ongoing task until the next evaluation time. The control of MRCP must accomplish two design pre-requisites: first, be computationally efficient to react to the new operator orders in real time and, second, achieve an automatic and transparent working mode with respect to the operator.

- **MRCP from the operator's point of view** MRCP is an augmented robot that allows the execution of complex tasks in complex environments. The robots cooperative behaviour must be automatic and transparent. The operator is relieved from the need to decide which robot executes each task segment and can focus the efforts on the task itself. For instance, in telemanipulation tasks, the operator does not guide a robot to generate an action on the workspace; but guides the task itself, guiding the object instead of the robot.

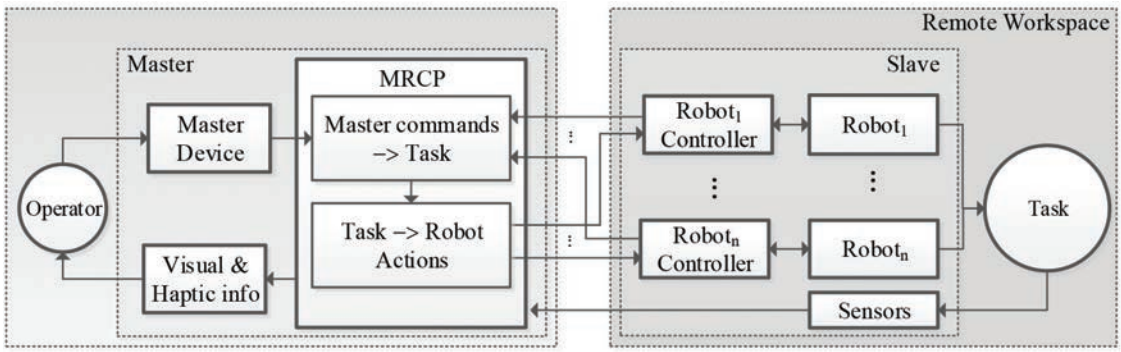


Figure 1.1: General block schema of MRCP

1.2 Advanced solutions for dexterous teleoperation

Initially, teleoperation systems where designed for particular tasks in controlled environments [18]. The continuous technological evolution of teleoperation has led to apply these systems in a wide range of task fields and environments. The continuous development has allowed the evolution of teleoperation environments, from basic telemanipulation in controlled workspaces to more challenging applications such as precise micro-surgery on deformable tissues. The growing demands of these applications require innovative systems. Therefore, research has been addressed to, on one hand, design new slave platforms composed of multiple cooperating robots or exploit the capabilities of redundant robots; and on the other hand, to develop new control paradigms.

This section reviews the state of the art of previous solutions adopted for teleoperation of complex tasks in complex environments. First, reviewing the systems based on redundant and multirobot systems, and second, exposing different contributions to teleoperation control paradigms.

1.2.1 Solutions based on robot architectures

Several solutions have been developed to increase the applicability and dexterity of robotic systems. From the architectural point of view, standard solutions are based on redundant robots, on multi-robot platforms and on the design of specific systems.

The first two solutions are exposed, pointing out the differences with MRCP.

Redundant Robots

To increase robot manoeuvrability and the operational workspace, a feasible solution is the use of redundant robots. In robotics, kinematic redundancy appears when the robot has more DoF than those required for the task. From this basic definition, several other definitions have arose, [19]. These redundant DoF allow a robot to reach a pose in the workspace in multiple configurations. The resultant multiple kinematic solutions can be used to avoid singularities [20, 21], collisions [22, 23] and to execute trajectories minimizing some parameters, like joint torques, [20]. One step further is the use of highly redundant robots. High redundancy is applied in fields as minimally invasive cardiac surgery, MICS, using snake like teleoperated robots. An example of this approach is the CardioARM, [24, 25], a poly-articulated teleoperated arm that enables single port access for beating heart surgery. Redundancy is used to adapt the robot configuration to the patients anatomy and create an access path acting as a catheter.

Flexible robots are the evolution of high redundancy. These robots are not composed of a set of joints, but of flexible materials that describe curves to position the tool tip. This feature enables the access to clustered regions of the workspace unaccessible with traditional serial manipulators. This type of robots are less invasive with the environment (e.g. surrounding tissues and organs), deforming their shape to adapt to the workspace. The STIFF-FLOP flexible robot, presented in [26], is designed for minimally invasive surgery, it needs an internal control that enables the correct tool positioning while its body is deformed by the workspace, [27].

However, depending on the task to be performed, redundant robots can be ineffective. To illustrate an application in which redundant robots do not offer a practical solution, lets consider the need to remotely manipulate an object, for its inspection, in front of a camera within a complex and dynamic environment. The visualization of the object from all the desired points of view may require large accessibility and manoeuvrability. This task can not be accomplished with only one robot due to

diverse reasons: restrictions of the own robot (workspace, singularities and joint limits), possible obstacles on the workspace, and/or occlusions of the object produced by the robotic arm, what might even require picking the object using different grasping positions. This regrasping process implies putting the object on a specific place in order to regrasp it when using a single robot, redundant or not. One such operation can significantly alter the desired trajectory, disturbance that can be avoided using MRCP. Fig.1.2 shows a MRCP composed of two robots executing one such remote manipulation. In Fig.1.2.a the robot that holds the object constitutes an occlusion within the camera field of view, whereas in Fig.1.2.b the occlusion disappears. The inspected object is manipulated from another grasping point and the task goal can be accomplished. MRCP executes automatically the task transfer between robots guaranteeing the task requisites.

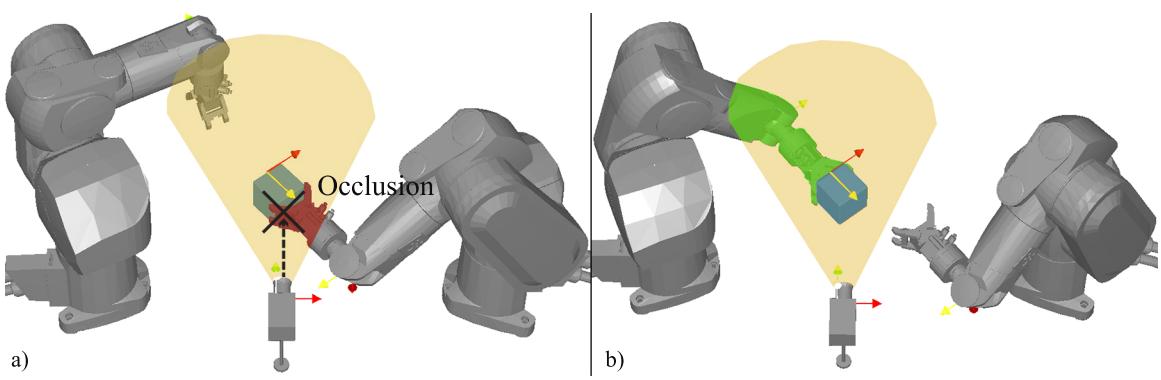


Figure 1.2: Example of visual teleinspection of an object in front of a camera.

Another illustrative example in which a redundant robot solution is inefficient consists in tasks like wrapping a large pipe with a tape. Fig.1.3 shows the trajectory required in one such task, where the pipe represents an insuperable obstacle for a unique robot, even if it is redundant.

In the proposed approach, MRCP, redundancy is based on the number of robots, thus obtaining not only a higher accessibility, but also a larger working volume. The automatic task transfer and the cooperative behaviour substitutes the robot redundancy. To illustrate the complementarity of these different approaches, let's suppose a trajectory that crosses a singularity. With redundant robots, redundancy enables

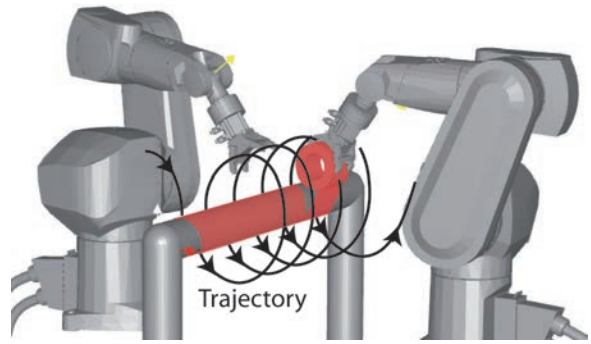


Figure 1.3: Example of a trajectory around a pipe.

obtaining these kinematic solutions that generate joint configurations that avoid singularities. With MRCP, the solution is not obtained directly from the redundancy of every single robot, but from the cooperation of all of them, resulting a redundant cooperative robotic system. This analysis can be conducted from the workspace point of view: Robot redundancy generates a dexterous workspace (a workspace in which a robot can be positioned in any orientation). MRCP achieves the same dexterous workspace by means of the cooperative behaviour of the robots conforming the system. From the point of view of this research, the dexterous workspace is reduced to a task dexterous workspace, where the robot reaches the whole workspace with all the orientations required for the task. Further information about workspaces generated by cooperative robots can be found in, [28].

Multi-Robot Systems

Multirobot systems are usually designed to carry out manipulation tasks that a single robot cannot do, [29,30]. Representative examples of these tasks are the manipulation of large, heavy or even flexible objects [31, 32]. The research in this field is mainly focused on control and path planning solutions for the closed kinematic chains formed by cooperative robots co-holding an object, [33, 34].

Concerning multi-robot path planning, an automatic multirobot path planner, presented in [35], the required motion is computed to manipulate an object from an initial point to a target point using several robots in a workspace with obstacles.

The path planner includes the possibility of transferring the manipulated object from one robot to another, thus increasing the workspace volume and avoiding possible obstacles during object transportation. The main difference between MRCP and this work relies on the lack of path planning in teleoperation. Consequently, the solutions proposed for multi-robot path planning cannot be applied in teleoperation. MRCP could represent a feasible solution when no path planning is possible.

Multirobot systems have been recently applied in bimanual manipulation, where more than one robot operate simultaneously, [36]. Unlike co-manipulation, in bi-manual manipulation each robot plays a complementary role to achieve a common objective. For instance, a robot holds a manipulated object while a second robot acts on it. The common approach is based on the division of the global task into various subtasks. Each task is assigned to one of the robots and the actions are synchronized. Examples of bi-manual manipulation can be found in [37, 38], where a robot arm holds a glass while a second robot pours liquid from a bottle; or in [39], where a motion planner for bi-manual complex insertion tasks is developed. The main difference between bimanual telemanipulation and MRCP is the complementary role of the robots that differs from one approach to the other: bimanual telemanipulation uses simultaneously two robot arms to execute the task, while MRCP (with no limitation of the number of robots composing the system) uses a single robot at a time, the most suitable one to execute the task and, when necessary, it transfers the task to another robot to ensure its completion.

Cooperative robots are also applied in teleoperation. An important part of current research, related to the proposed MRCP, focuses on Multi-Operator-Multi-Robot, MOMR, and Single-Operator-Multi-Robot, SOMR, teleoperated systems. MOMR schemas are based on architectures composed of several slave robots in which each slave robot is individually controlled by an operator [40]. Instead, SOMR architectures are composed of several robots controlled by a single operator. In [41], the control architecture of two slave robots, an arm mounted on a mobile robot, allows switching among different properties of each robot or among the robots themselves using a single master device. This SOMR proposal differs from the MRCP concept in two

basic aspects: first, in SOMR, each robot plays a different and specific role, whereas in MRCP all robots have complementary roles. A second difference refers to the knowledge about which robot is controlled at each point of time: in SOMR, the operator actively selects the robot to be controlled by the master device, whereas in MRCP the selection is made automatically and transparent to the operator. To summarize, MRCP can be classified as a SOMR from the control point of view and as a Single-Operator-Single-Robot, SOSR, from the operator's point of view.

1.2.2 Solutions based on control paradigms

Several control paradigms have been developed to carry out complex tasks. From these approaches, those having interest from the MRCP point of view are reviewed below. Again, the differences and similitudes with MRCP are pointed out.

Task Oriented Paradigm in Teleoperation

In the task oriented paradigm, the operators actions are related to the task rather than to the agents (robotic arms in teleoperation field). Following this control paradigm, the task is formulated in terms of its requirements (dexterity, pay load, task trajectory, etc) rather than on the robot actions (trajectories, grasping, workspace, etc). MRCP presents several similitudes with the task oriented paradigm: In MRCP the operator interacts with the task rather than on the robots that execute it. The control is designed to guarantee a satisfactory task execution, minimizing disturbances, in time or space, of the operators teleoperation commands.

The task oriented paradigm was introduced in teleoperation systems in works like [42]. Its authors propose a task oriented single-operator multiple-robot (SOMR) system in which a single master device is used to define actions directly related with the task to be executed. They focus the work on achieving a teleoperated system whose operator does not drive the slave arms, but uses the master to generate specific task-oriented actions. In other words, the operator modifies the task oriented variables with the master device (e.g. object trajectory in a teleinspection task). Several examples

are used to describe this approach: first, a co-manipulation of a heavy object by means of two robots. Here, as in MRCP, the master device is used to describe the manipulated object movements instead of the arms movements. Another task is based on the coordination of two individual arms when screwing an object. The variables of this task are the position and orientation, as well as the forces and torques applied. The control system then computes the coordinated actions of each robot. In [43] a task oriented teleoperation system for micro-scale manipulation tasks is proposed. Precise motions are imposed by the scale of the workspace and the tasks. To improve the operator's performance, in terms of precision and completion time, a set of predefined tasks are programmed (pick object, place object and tool movement). The operator supervises the system to ensure a correct task execution. The task oriented paradigm has been used in several other fields in robotics. In [44, 45] a task oriented control is applied to humanoid robots that execute tasks that are decomposed into subtasks. A hierarchical classification of the subtasks define which are critical and must be executed before. Following the previous work, a generalization of the proposed task oriented paradigm is presented in [46]. The task oriented paradigm is also applied in industrial robotics. In [47] the authors use a set-up composed of two redundant robotic arms controlled following this approach. Concretely, the authors base the control in the Operational Space Formulation, generating motion trajectories with dynamic consistency. These techniques have a high computational cost. Authors propose a change on the motion computation to reduce the required computing time.

MRCP is designed to enable the direct task execution with the master device following the task oriented control paradigm. The operator teleoperates the task (not the slave robots) and the control unit translates the task oriented master orders into slave robot actions. Let's use an introductory example to illustrate this control approach. The task consists in turning a multi-revolution faucet. The faucet is auto-returning, imposing a task constrain: at least one robot must be always grasping one of the faucet arms to turn it. If the faucet is ungrasped, it returns to its initial position and the task fails. The operator uses a single master device, a 6 DoF mouse, and generates task oriented orders: direction and turning velocity, Fig.1.4.

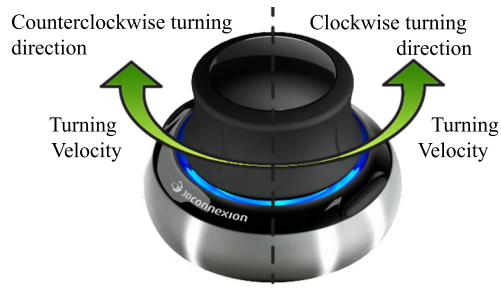


Figure 1.4: 6 DoF mouse used as master device generating task oriented orders.

This task imitates the human behaviour, in which two arms are used alternatively to turn the faucet: one arm executes the task until it reaches the limit of its workspace while the other searches the best position to continue with the task when necessary. To execute the task, the operator is provided with a master device to manage the task variables: direction of rotation and its velocity. Two 6 DoF robots (one in front of the other) conform the slave station, each equipped with a pneumatic gripper. A set of snapshots of the task execution is shown in Fig.1.5. The icons under the robots show whether the robot is grasping or not one of the three arms of the faucet.

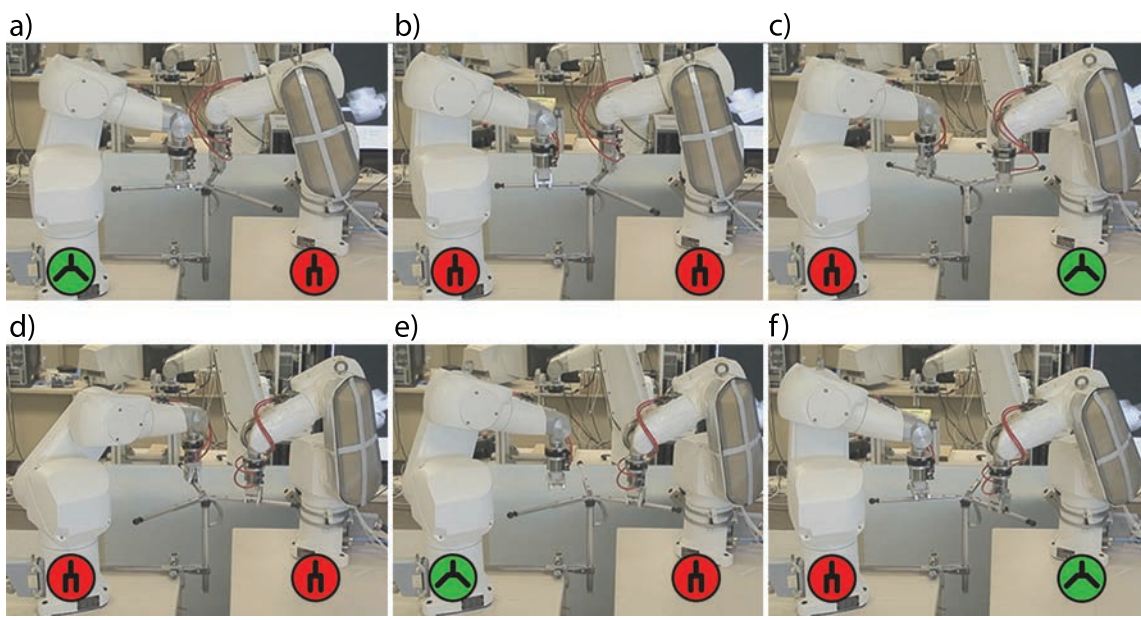


Figure 1.5: Set of snapshots of two robots alterning the telemanipulation of a multi-revolute faucet.

Competitive Multi-Robot Teleoperation

Following with the review of advanced teleoperation paradigms, another approach to be pointed out is the competitive multi-robot teleoperation paradigm. The competitive teleoperation is based on the game theory. Presented in [48], it is based on analysing those situations in which there is an interplay between the participants with similar, opposed or mixed goals.

In [49] a general overview of competitive multi-robot teleoperation is presented and an internet based teleoperation framework is proposed. In competitive teleoperation, the participants compete for mutually exclusive objectives protecting themselves and attacking the antagonists. The goal of each participant is to reach the objective or, at least, obtain the best possible result. Authors classify the taxonomy of competitive teleoperation between attack-defend taxonomy (winning the counterpart is the objective itself) and result-compared taxonomy (participants follow the same objective and the winner is the one with the best result at the end of the game). Based on these principles, a MOMR platform is developed. Authors assume a set of inherent complexity in these scenarios: unpredictable actions (a participant action depends on the task and on the others' actions) and highly unstructured and dynamic scenarios. A preliminary study is presented, in which two participants repeatedly play a game with different master devices and connection set-ups. In [50] authors present the Tele-Lightsaber, an experiment based on attack-defend competitive teleoperation. This platform is used to introduce artificial intelligence to substitute part of the human decisions. The same authors present a numerical evaluation to measure the operators Degree of Satisfaction, DoS, of the teleoperated system in the context of competitive teleoperation, [51]. DoS measures both, the accuracy of the slave robot and the observed delay between master and slave. The amount of different factors involved in the operators perception and their different nature impedes the definition of a mathematical expression to obtain the DoS. The obtained conclusions are based on experimental results. The presented results are coherent with the variation of the system performance: DoS varies in the same sense as the performance does.

Following the competitive teleoperation paradigm, in [52] the authors propose the

inclusion of a virtual repulsion force, as a safety measure in tasks with unpredictable behaviour executed in unstructured workspaces. Forces are based on an oriented cylinder bounding applied on each link of the robots. The radius of the cylinder is determined by the geometry of the links (the cylinder with minimum radius that covers the whole link) and increases with the velocity as a safety measure. The repulsion forces are modelled as mass spring models applied to the minimum distance vectors. The use of force vectors or force fields is a common solution in several robotic areas: from path planning or collision avoidance to virtual fixtures. The virtual forces are specially used in multi-robot cooperation, where more than one robot share a common workspace.

MRCP uses virtual forces in a competitive behaviour to avoid collisions between robots and obstacles and to guide the robots to the required poses. Each robot has a role, or priority, that determines whether it generates or receives a repulsive force. For instance, in Fig.1.6 a set of snapshots of a system composed of two robots with different roles (one is equipped with a driller, left on the images, and the other with a camera, right on the images) competing for a region in the workspace. The task requires a pseudo-assisted manual drilling of a bone and a continuous view of a certain region of interest, which is provided by the robot with the camera. The robot with the driller has higher priority than the one with the camera. When the manually guided robot approximates to the region of interest, it pushes the camera robot out of this region to avoid collisions. The camera is automatically reoriented to continue with the desired view. This behaviour is possible thanks to the decoupled structure of the robot (the first 3 DoF are independent from the last 3 DoF). When the robot exits the shared region, the camera robot retakes its original position guided by an attraction vector. In the mentioned set of snapshots, Fig.1.6, the force repulsion field is represented as a circle around the end effector. When the circle becomes red (larger discontinued line), the repulsion force generates a collision avoidance trajectory on the right robot. When the operator exits the shared workspace, the secondary robot recovers its initial pose in the direction of the force field. This system was developed under the research project CYCIT DPI2004-04558.

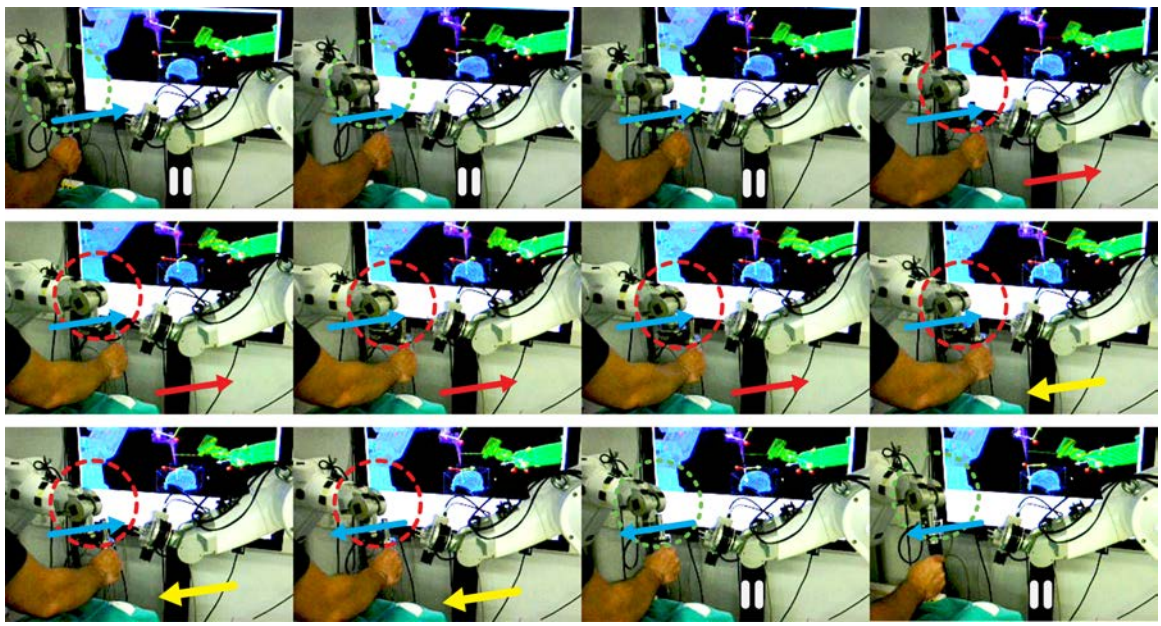


Figure 1.6: Set of snapshots of two robots with competitive behaviour.

Several similitudes can be observed between competitive teleoperation and MRCP. In MRCP there are several players, the slave robots, with a mutually exclusive goal: execute the task during the period between two consecutive sample times. To decide the winner of each game step, a set of rules are defined based on the suitability evaluation of each player. The final decision of a change between players (task transfer between robots) is decided by the need of an immediate task transfer. Using these rules and, based on the precept of minimizing the number of task transfers, the control algorithm decides if, in the current game step, a task transfer is required and which player is the winner (which slave robot executes the task).

Improving operator's dexterity in teleoperation

In teleoperation, intelligence does not impose operative barriers since there is a human behind, guiding the robot. However, the lack of propio and remote environment perception, as well as ergonomic limitations on the master device, might decrease the operator's capabilities. The use of robot arms, with less DoF than those of humans, limits dexterity and accessibility. Most teleoperated systems provide partial

visual feedback information of the workspace, focusing mainly on the area where the task is to be executed. When dealing with multirobot systems, possible collisions between robot arms or with other obstacles cannot be easily detected by the user. Several works present solutions to mitigate these problems: the use of semiautonomous control systems [53], virtual fixtures that help the user to follow a preferred path, thus improving the operator's performance, or avoid undesired slave robots configurations [54] [55]. Modifying the point of view of the remote workspace according to the task requirements constitutes another assistance to improve the operator's perception [56].

The contribution of MRCP to improve the operator's dexterity is based on the use of a task oriented paradigm. It facilitates and improves teleoperation, offering a new teleoperation interface. Current MRCP research is focused on applying virtual fixtures to the used task oriented paradigm. Following this paradigm, the virtual fixtures are not applied to the robots themselves, but to the task. Following the telemanipulation example and applying a guidance force, in MRCP the force is applied to the manipulated object and it is mapped on the master device.

1.3 Thesis Organization

This thesis is organized as follows: the second chapter reviews the most important robot's dexterity criteria, empathising those on the manipulability. The chapter also reviews the use of dexterity in several robotic fields (e.g. design, control, multi-robot systems, etc.). The third chapter develops the robot evaluation methodology, explaining how MRCP evaluates the suitability of each robot to continue with the execution of the ongoing task. The chapter describes the different criteria to evaluate the robots and proposes a methodology to enable the comparison between evaluation indices from different nature (evaluation of robots in different spaces). Once exposed the robot evaluation criteria, chapter four explains the control aspects that commands the internal behaviour of MRCP and enables task-oriented teleoperation: compare the slave robots, select the most suitable, decide when execute a task transfer and computes the robots actions. Several examples illustrate the MRCP control method-

ology. Following, the MRCP application to robot assisted minimally invasive surgery is introduced as example of dexterous teleoperation tasks in chapter five. Several experiments are explained and analysed. Finally, in chapter six the task-oriented virtual fixtures are introduced and their performance analysed with an experiment based on a virtual simulator. Conclusions and further developments are exposed in chapter seven.

Chapter 2

Robot Dexterity Evaluation

2.1 Introduction

Due to the importance of dexterity evaluators in robotics and, in particular in MRCP, this chapter introduces robot dexterity concepts and reviews the most relevant dexterity evaluation indices. Dexterity evaluation plays a central role in robot design and control. MRCP uses dexterity, among other criteria, to decide the suitability of the involved cooperative robots to perform an on-going task, as will be explained in further chapters.

Several criteria must be taken into consideration when designing a new robotic manipulator, when selecting an existing one as the optimal for a task or when planning a trajectory. One of the criteria most frequently used is robot dexterity. A generic definition of robot dexterity is *the ability of a robot to perform a task*. Robot dexterity is used in design, control, determining efficiency, evaluating task completion capacity, studying dynamic aspects, force and payload, redundancy optimization, robot placement, etc, taking in all of them a slightly different meaning. Dexterous workspace defines the subspace of the robot workspace that can be reached by the end-effector in all possible orientations.

Dexterous manipulation relies on the study of control strategies and end-effector placement for optimal grasping. Kinematic dexterity is a measure of the capability of a robot to execute a specific task in a given configuration. This last concept, which is

usually known as kinematic manipulability or simply manipulability, is used in MRCP.

Dexterity is an important factor in different robotic aspects:

Dexterity in robot design

Dexterity is used to design robot manipulators, defining aspects like optimal joint distribution, establishing link lengths or scaling actuators. In consequence, several prototyping software packages, like [57,58], include dexterity evaluation as part of the robot design phase. In [59] the design of fault tolerant robots is tackled by means of kinematic robot design that allows the recovery after a locked joint failure. This is achieved imposing the largest possible minimum singular value of the Jacobian, which is a measure of dexterity. Dexterity is also used as an optimization parameter in parallel robots, from which many examples can be found in the literature. For instance, in [60], three aspects are taken into account when designing a parallel robot: manipulability, workspace volume and kinematic optimization.

In the design of specific robots, dexterity plays an important role. The use of a dexterity criterion helps to construct robots with optimized design for the task they have been designed for. Several examples can be found in fields like space or underwater robotics. For instance, in [61,62], dexterity, in the form of dynamic manipulability, is used and adapted to include the specifications imposed by an underwater environment. Robotic minimally invasive surgery, RMIS, imposes several restrictions and requirements to be taken into account when designing a robot. Full access and high dexterity inside the whole surgical workspace are examples of these requirements. The movement constraints, imposed by the fulcrum point, must also be considered. In [63] dexterity is used to design surgical robots in terms of movability and operability. For their design, these authors have developed a simulator that uses the robot kinematics and specific anatomical patient data to obtain an accurate robot evaluation for each individual surgery. There are specific applications that demand the accomplishment of several criteria during the robot design phase. An example can be found in [64], with the design of a RMIS multi-robot system. This methodology minimizes the overall length of the robotic arms ensuring a minimum value of manipulability and

accuracy in all the workspace.

Dexterity in control and trajectory generation

Besides being a robot design criterion, dexterity is also applied in path planning. Several parameters are considered when computing a robot trajectory, which can be classified as extrinsic or intrinsic. Those extrinsic are related with the workspace where the task is executed: collision avoidance, path length, possible blockings in shared workspaces, narrow passages avoidance, etc. While those intrinsic ensure that the mechanical limitations of the robots are respected: workspace boundary, required joint efforts (in terms of velocity, acceleration or torque) or singularity avoidance. Apart from considering these limitations that may prevent a satisfactory task execution, other aspects can be included as factors to be optimized. Several works optimize robot dexterity in the process of path planning, as well as on real time trajectory execution. Examples of path planning with dexterity optimization can be found in [65], in which a trajectory generator that maximizes manipulability is presented. Their authors assume the possibility of obtaining all inverse kinematic solutions of a serial manipulator. The resulting trajectory ensures a path generated from the most dexterous robot configurations. This presented methodology is applied to an arm of a humanoid robot. Another interesting example, applied to space robotics, reduces the required joint torques of a redundant robot in the context of dexterous hand manipulation, [66]. A dynamic manipulability ellipsoid is defined as a dexterity measure to be used as minimization criterion.

In real time trajectory control, several techniques can be found that include dexterity inside the control loop. In [67] a combined method for trajectory tracking and collision avoidance in redundant robots is presented. The avoidance manipulability, which describes the ability of a robot to avoid a collision, is used to prevent collisions by means of modifying the robot configuration while continuing with the task. The capability of robot shape modification, as the authors mention, is provided by the robot redundancy. In [68], a control schema of a robotic arm acting as an assistant in RMIS is presented. The trajectory control must avoid singular configurations by

means of the computation of dexterity, expressed in terms of manipulability.

Dexterity in redundant robots

Redundant robots, defined as those robots that have more DoF than those required for the task, can use their redundancy to optimize certain criteria. Usually know as secondary task, these criteria are optimized whereas the main task is executed without any perturbation. Dexterity measures are commonly optimized, determining the robot configuration solution from those that accomplish the restrictions imposed by the main task. For instance, the use of redundancy to maximize dexterity by means of robot reconfiguration is applied in [69], where a task-oriented manipulability strategy is presented. Task oriented manipulability measures, TOMM, describe the task in terms of optimal manipulability during its execution. Robot redundancy is used to reconfigure the arm to adjust, as much as possible, its manipulability to the optimal TOMM.

Redundant DoF are also used to optimize aspects like payload, economy of movement, joint velocities, accelerations and torques. In [70] robot redundancy allows on-line obstacle avoidance. A multi-task approach is used, in which the main task is the end-effector trajectory, and collision avoidance is considered a secondary task. This second task is defined in a one dimensional space, decreasing the risk of singularities in local positions with not enough redundancy. A similar problem is solved in [71], in which the authors present a real-time control method to reconfigure redundant robots when an unexpected obstacle appears. This methodology is based on the Avoidance Manipulability index, introduced in [72], and defined as the capacity of a robot to change the shape of its links when executing a task. In [73], the concept of sub-robots generated by a redundant robot is presented and used to simplify the computation of singularities. A sub-robot is the result of generating a new robot using a subset of the original robot joints while keeping the rest blocked or, as the authors call, frozen links. To determine if a redundant robot is in a singular configuration, the following corollary is applied: a redundant robot (n DoF) is in a singularity in the performance of a m dimensional task, if and only if, all its m order sub-robots are in a

singular configuration. In hyper-redundant robots, the study of manipulability is also a key issue for design and control, [74–76]. Finalising the short review of dexterity applied to redundant robots, the work developed in [77] must be mentioned: a novel manipulability index, the Manipulability Zonotope Volume. This concept is similar to those of polytopes, presented as robot evaluation index in [78], which is used to obtain a better redundant robot evaluation.

Dexterity in multi-robot systems

In cooperative manipulation tasks, in which two or more robots simultaneously manipulate an object, the position of each arm can be computed following certain optimization criteria. Dexterity appears as one of the most used criterion. Some research works are oriented to individual dexterity optimization, with each robot maximizing its dexterity. A second approach is based on the global dexterity optimization, where the resulting configuration of each robot maximizes the dexterity of the formed closed kinematic chain.

Concerning individual dexterity optimization, in [79], the dexterous reconfiguration of each robot is used to achieve an effective coordination. Dexterity indices are used to define the required robot actions for the reconfiguration (dexterous robot reconfiguration in author's terms). In that work, force manipulability, that is complementary to dexterity or manipulability[1], is used to impose certain constrains (manipulability optimization) to the redundant robots when reconfiguring their positions. In [80], the manipulability of cooperative robots with some passive joints generating closed kinematic chains is measured. Several case studies for robot design are presented, like five bar linkage and a parallel robot following the Stewart platform construction paradigm. In [81] polytopes, which is an extension of the classical manipulability representation, are used to determine the manipulability of closed kinematic chains formed by serial manipulators that consider the dynamic constrains imposed by the own robots and the cooperative manipulation.

[1]The words dexterity and manipulability are used indistinctly. The presented review uses the term proposed by the corresponding authors of each reviewed work

Concerning global manipulation, in [82] dual-arm manipulability, represented as an ellipsoid, is presented. This evaluation index is the result of the intersection of the manipulability ellipsoids of each single robot. In [83] dual-arm manipulability is redefined. The task-space oriented formulation, introduced in [84], is used to represent and solve the motion of the robots that cooperatively manipulate an object. In [85, 86] the manipulability and manoeuvrability ellipsoids are applied to two Autonomous Underwater Vehicles co-manipulating a rigid object. Several configurations and trajectories of the system are presented to illustrate how the ellipsoids vary their shapes.

Robot placement to improve dexterity

Robot placement is the process of determining the position and orientation of the robot base within the workspace. Several aspects must be taken into account when deciding the robot placement. First, the task space must lie inside the robot workspace ensuring the required end-effector reachability. Second, the task and workspace specifications must be fulfilled. An illustrative example of task specification is the restriction of the robots movements imposed by the fulcrum point in robotic MIS. Concerning workspace limitations, the obstacles present in the workspace can restrict robot mobility. The robot placement process starts with the determination of the subspace of all the candidate positions. Second, and considering the previously mentioned aspects, several optimization criteria can be used to obtain the final robot placement.

In [87] kinematic manipulability is used to optimize the robot placement. The manipulability measure, jointly with a methodology to compute the workspace boundary, is used to obtain the robot workspace given a concrete task. Once the robot workspace is fixed, the robot placement is unequivocally determined. In [88], the optimal position of an underwater robot, URV, equipped with a 7DoF redundant arm is computed, optimizing the dexterity workspace in manipulation tasks. The target points that define the task are known in advance. The dexterity of the end effector in these target points is measured using the Condition Number index, which will be described in detail later in this chapter. The positioning strategy used is based on four sequential steps: first

the robot kinematics and workspace are determined; second, the URV is placed in a position in which all the target points can be reached. Third, the robot placement is optimized and, finally, the non-existence of singular robot configurations that prevent the task execution is ensured. If a singularity is detected, the URV position is modified, avoiding these singularities.

Minimally invasive robotic surgery imposes several robot movements and reachability limitations: The end-effector, a laparoscopic tool, must be inserted and moved conditioned by the fulcrum point. This restriction, jointly with the nature of the surgery to be performed and the patient specific anatomy must be considered to ensure task completion. The correct robot placement represents an important factor to be taken into account. In [89] the optimal placement of the four arms entry points of the Raven IV surgical robotic system is studied. Again, the optimization criteria is based on the dexterous workspace maximization. Two aspects of the system are optimized: the geometric properties of the common workspace and the dexterity of each individual arm, in terms of the workspace boundary isotropy. Similar approaches can be found in [90], where the tool dexterity and the endoscopic view of the surgical regions are optimized. The resulting optimization process determines the port placement for each robot. The authors identify five different causes that prevent the robot to execute a task satisfactorily: end-effector reachability, tool orientations, collisions between tools and endoscope inside the patient, robot singularities and joint limits and, finally, robot collisions. While the first three problems are solved determining the optimal port placement, solving the two last restrictions rely on computing the optimal robot placement. In [91] the ports and robots placement of the DLR robotic surgical system (MIRO) is obtained by means of optimizing the manipulability and the accuracy measures.

Apart from the kinematic measures, another family of evaluation criteria is oriented to measure the robot dynamics. For instance, Dynamic Manipulability of Robot Manipulators; in [92], measures the relation between the joint driving force and the acceleration of the end-effector. This measure has been reformulated to better fit robot dynamics when dealing with redundant robots [93], specific robot structures, or

in the study of the movements close to singularities [94]. In [95] the Inertia Matching Ellipsoid, which combines the dynamic manipulability and the force ellipsoids, is defined and applied to optimize trajectories in industrial serial robots manipulating heavy weights, as well as, to the design of new legs of jumping robots. Dynamic measures, as well as other families of robot evaluation indices are not considered in this chapter because they are out of the scope of this work.

2.2 Dexterity Evaluation

In this subsection several dexterity evaluation indices are introduced and analysed under the point of view of the MRCP requirements. Every index is accompanied by a numerical analysis applied to several XY planes of the workspace of a Staübli RX60B 6 DoF robot, and illustrated with a graphical representation of the obtained results. The implementation of these indices has been programmed under the Robotic Proximity Queries library, RPQ, described in A.2.2. The kinematics of the used robot is shown in Fig. 2.1[2] and the Denavit-Hartenberg parameters are shown in Table 2.1. A complete kinematic analysis of the Staübli RX60B can be found in [96]. In [97] a study of manipulability of the workspace of a Puma560 and a Mitsubishy Movemaster robots is presented jointly with a new algorithm to compute robot's manipulability in order to obtain optimal trajectories.

Robot Singularities

Robot singularities can be defined as a set of subspaces inside the workspace, in which the robot looses one or more degrees of freedom. Numerically, let m be the required DoF for a task and $n-k$ the robot remaining DoF in a singularity (where n: DoF of the robot and k the number of lost DoF in that singularity), then, when $m > n-k$ the task cannot be executed. Mathematically, the origin of singularities arises from

[2]Graphics extracted from Lecture courses Chapter 4 of Robotics and NN laboratory of the Department of Cumputer Science, San Diego State University, USA. www.medusa.sdsu.edu/Robotics/CS656/Lectures/CHAP4.pdf

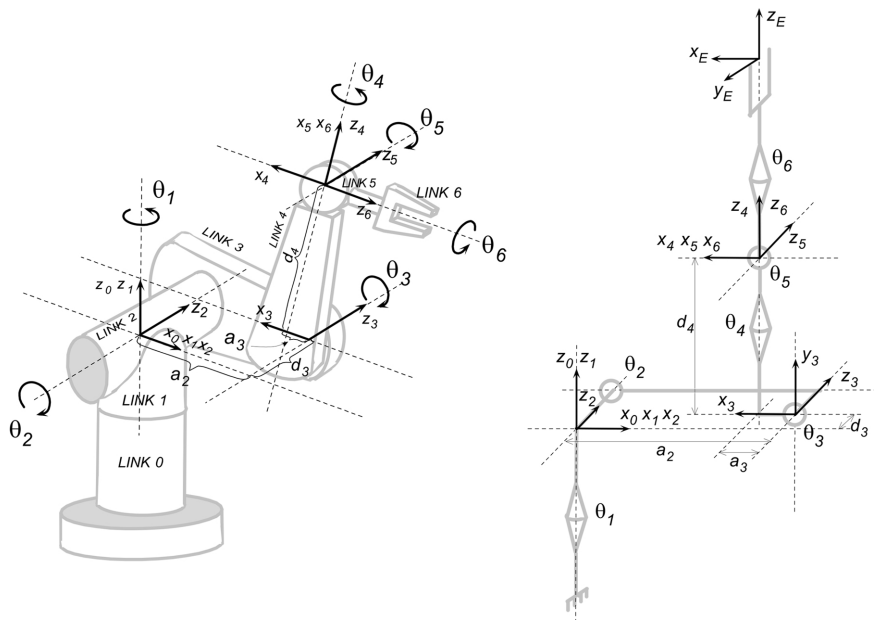


Figure 2.1: Representation of Puma560 robot kinematics, which is equivalent to the used Staübli RX60B.

the relationship between the velocities on the joint space, $\dot{\vec{\theta}}$ and the Cartesian work space, $\dot{\vec{x}}$, (2.1).

$$\dot{\vec{\theta}} = J^{-1}(\theta)\dot{\vec{x}} \tag{2.1}$$

When J becomes non invertible, J^{-1} does not exist. In other words, when J is singular, there's no possibility to map the desired $\dot{\vec{x}}$ into the joint space, $\vec{\omega}$. The robot looses one or more degrees of freedom, being impossible to move the end-effector towards certain workspace directions.

The singularities can be classified in several ways. One of the most common classification is based on the singularity relative position inside the workspace:

- Workspace boundary singularity: The robot is in a configuration in which the end-effector is in a workspace boundary. The robot reaches its boundaries when is extended to full length or when is completely retracted.

- Workspace interior singularity: This type of singularities includes all singulari-

Table 2.1: D-H parameters for Staübli RX60B

$Joint_i$	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90	0	0	$theta_2$
3	0	$a_2 = 290mm$	$d_3 = 49mm$	θ_3
4	-90	0	$d_4 = 310mm$	θ_4
5	90	0	0	θ_5
6	-90	0	0	θ_6

ties that occur out of the robot's workspace boundary. These singularities are usually produced by the alignment of, at least, two rotational joints.

The interest to compute and control robot singularities comes from several aspects:

- Lost of freedom: When the robot falls into a singularity, the number of active DoF in which the robot can move decreases. The number of DoF is determined by the rank of the Jacobian determinant.

- Workspace: The workspace can be mapped and characterized by means of the study of singularities. The robot workspace can be determined computing the workspace boundary singularities. The interior singularities are usually associated to changes in the robot configuration, requiring high joint accelerations and unexpected robot movements.

- Control: Several control approaches are used to define the trajectory of a robot. The vicinity of the robot to a singularity interferes with these types of control. For instance, when controlling the robot by means of the end-effector velocity, the ill-condition of J close to a singularity produces failures on the control system or forces the robot to generate non achievable joint velocities and accelerations.

- Mechanical constraints: When the robot is close to a singularity, the generation of end-effector movements towards the singularity requires large joint movements and accelerations.

Interesting formal reviews of the robots workspace singularities can be found in [98, 99]. In the first cited work, a part of the theoretical review, the singularities of a PPR and Scara robots types are described. Other works focus on the workspace singularities from a geometric point of view, [100], or from other algebras like the Grassmann-Cayley Algebra in [101].

Basis of Dexterity indexes

Dexterity is defined as the ability of a robot to perform a movement given a concrete configuration. Most dexterity evaluators are based on the study of the Jacobian matrix. Let's assume that a n DoF robotic manipulator operates in a m dimensional workspace, where $m \geq n$. The robot forward kinematics is defined as (2.2), where $X \in \mathbb{R}^m$ and $\theta \in \mathbb{R}^n$ are the m dimensional workspace coordinates and n dimensional joint coordinates vector respectively.

$$X = f(\theta) \tag{2.2}$$

The derivative of (2.2) gives the relationship between the joint and the Cartesian velocities of the robotic arm (2.3).

$$\dot{X} = J(\dot{\theta})\theta \tag{2.3}$$

Known as the Jacobian matrix, J (2.4), represents the linear mapping between the joint and Cartesian velocities,

$$J(\theta) = \frac{\partial f}{\partial \theta}(\theta) \in \mathbb{R}^{mxn} \tag{2.4}$$

Manipulability

Based on the Jacobian matrix analysis, the manipulability index, w, proposed in [102], is one of the most extended dexterity measures. The w index is defined as the square root of the determinant of the product between the Jacobian matrix of the manipulator, J, and its transposed, J', (2.5). This measure indicates the ability of a robot to generate a movement from a given configuration. When the robot falls into a singularity, the Jacobian loses rank, and then the manipulability index becomes zero. In other words, being $\lambda_1 \geq \lambda_2 \geq .. \geq \lambda_m$ the eigenvalues of JJ', when the robot is inside a singularity, one or more λ_i become zero.

$$w = \sqrt{det(JJ')} = \sqrt{\lambda_1 \lambda_2 .. \lambda_m} \tag{2.5}$$

Robot manipulability can be expressed as an ellipsoid in the manipulator workspace. The volume of the ellipsoid is proportional to w and denotes the ability of a manipulator to perform a movement. Once obtained the singular values, $\sigma_1 \geq \sigma_2 \geq .. \geq \sigma_m$ using the singular value decomposition technique: $J = U\Sigma V'$, w can be expressed as the product of all σ_i, 2.6. The principal axes of the manipulability ellipsoid are defined by $\sigma_i u_i$, where $i = 1..m$ and u_i are the column vectors of U. In [103, 104] real time methods for Singular Value Decomposition are presented.

$$w = \sqrt{\lambda_1 \lambda_2 .. \lambda_m} = \sigma_1 \sigma_2 .. \sigma_m \tag{2.6}$$

Geometrically, the principal ellipsoid axes are $w = \sigma_1 u_1, \sigma_2 u_2, ..., \sigma_m u_m$ where, the set of orthonormal vectors u_i define their orientation and σ_i their length. Fig.2.2 shows an illustrative example of a manipulability ellipsoid with axes $\sigma_1 u_1$, $\sigma_2 u_2$ $\sigma_3 u_3$ of a 3 DoF robot.

The manipulability index presents dependencies of order and scale, preventing the numerical use of w to directly compare two robots with different kinematics in terms of DoF or link lengths. This problem is solved in MRCP by means of a time space transformation, as explained in Chapter 6.5.3.

To solve the order dependency, a new manipulability index, M, can be obtained

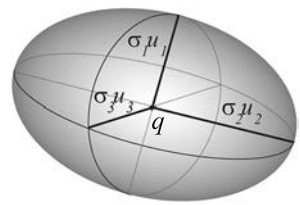


Figure 2.2: Illustrative manipulability ellipsoid example of a 3 DoF robot

using its order, m, as shown in (2.7).

$$M = \sqrt[m]{det(JJ')} \tag{2.7}$$

With M, the order dependency is solved, but dependency still remains unsolved. In [105] a possible solution is proposed, consisting on the quotient between the order independent manipulability, M, and a dimension function f_M, 2.8.

$$M_r = \frac{M}{f_M} \tag{2.8}$$

In that work the used f_M is based on the dimension of the robot arm length: the total length of the manipulator is computed as the sum of all the link lengths, $l_i = \sqrt{a_i^2 + d_i^2}$, where a_i is the ith link length and d_i its joint offset following the D-H notation.

Fig.2.3 shows the value of manipulability of the 6 DoF Staübli RX60B robot, seen from different XY planes. For better visualization, only half of the workspace is shown. The manipulability, w, is useful to detect boundary and internal singularities. w presents a smooth evolution when the robot is moving to an internal singularity but, on the contrary, it presents an abrupt gradient on the boundary vicinity. This behaviour makes w an ideal index for interior singularities detector in real time robot control. On the contrary, w is not an optimal solution for boundary detection in terms of real time control.

Derived from the original Manipulability several other indices have been developed. For instance, the Scaled manipulability ellipsoid, defined in [78], takes into account the maximum velocity achievable by each joint in the computation of the robot's

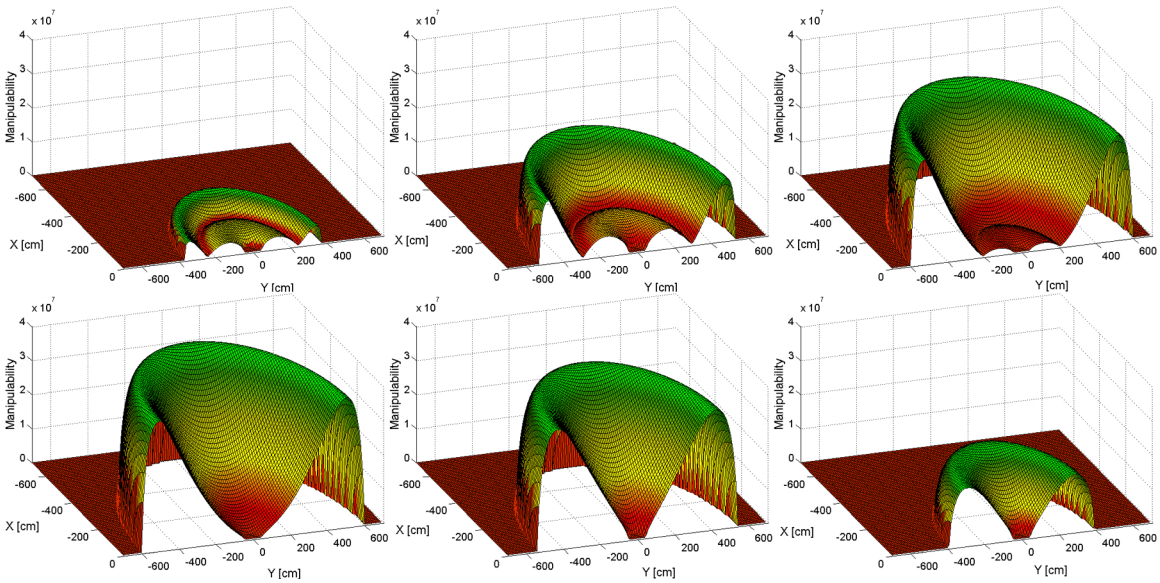


Figure 2.3: Manipulability value for a 6 DoF Staübli RX60B robot in several $X-Y$ planes (with z=0mm, z=200m, z=400mm, z=600mm, z=800mm, z=1000mm with respect to the robot base)

manipulability. In the same work, a comparison between ellipsoids and polytopes is presented. Polytopes results a better tool for expressing joint velocities. The work presented in [106] extends the manipulability index including penalization functions that modify the final robot evaluation. Examples of these penalization functions are the proximity to a joint limit, the presence of an obstacle or the use of augmented Jacobian.

Directional Manipulability

Derived from w, the directional manipulability index, ψ_u measures the robot manipulability moving from its current configuration to a concrete direction in the workspace. Given a robot configuration q and a desired movement direction expressed as a unitary vector, u, the directional manipulability is defined as,

$$\psi_u = \frac{|\dot{x}|}{|\dot{q}|} = \frac{1}{\sqrt{u'(J^\dagger)'J^\dagger)u}} \tag{2.9}$$

Assuming that J is a full rank Jacobian, the pseudoinverse $J^{\dagger}$ can be calculated as $J\dagger = J'(JJ')^{-1}$ and the directional manipulability can be rewritten as, (2.10).

$$\psi_u = \frac{|\dot{x}|}{|\dot{q}|} = \frac{1}{\sqrt{u'(JJ')^{-1}u}} \tag{2.10}$$

Geometrically, ψ_u, is the distance from the center of the ellipsoid to the surface point where the line (with direction u) intersects. Fig.2.4 shows an illustrative example of two ψ_u. In this example, $\psi_{u1} < \psi_{u2}$, indicates that the robot is more capable of generating a movement in u_1 direction than in u_2.

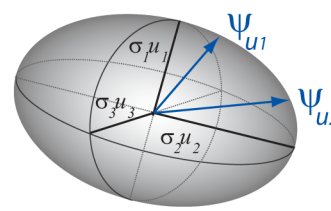


Figure 2.4: Illustrative directional manipulability example of a 3DoF robot.

Several examples of Directional Manipulability can be found in the literature. For instance, in [107], the assistant arm Manus is mounted on a mobile platform. The combined use of a robotic arm and a mobile platform generates a redundant system that opens the possibility of a non-limited X-Y dexterous workspace. The Directional Manipulability is used as a criteria to control this redundant mobile arm, defining the optimal position of the mobile platform and the robot configuration. Directional Manipulability is used as a task oriented index in a cooperative multi-robot system, [108].

This evaluation index is used in MRCP, among others, to determine the suitability of a robot to follow a trajectory. The index is reformulated to support its computation in a predicted trajectory (no path planning is possible in teleoperation), observing the mathematical uncertainness of the prediction methodologies.

Minimum Singular Value, σ_{min}

The minimum singular value of the robot Jacobian expresses the minimum workspace velocity achievable by a unit joint velocity vector. The corresponding eigenvector provides information about the most limited motion direction of the robot end-effector. When the robot is in a singular configuration, at least one on the singular values becomes zero. The singular value is efficient to determine when the Jacobian determinant is close to zero and, in consequence, when the robot is near a singularity. As the upper value of the singular value does not have a defined maximum value, the use of this index is not appropriate for control purposes.

In Fig.2.5 the evolution of the Jacobian minimum singular value in several $X-Y$ planes is shown. The Minimum Singular Value presents a smooth gradient when the end-effector approximates to an internal singularity. On the contrary, boundary singularities cannot be always detected in advance, as shown in the first four $X-Y$ sampled planes.

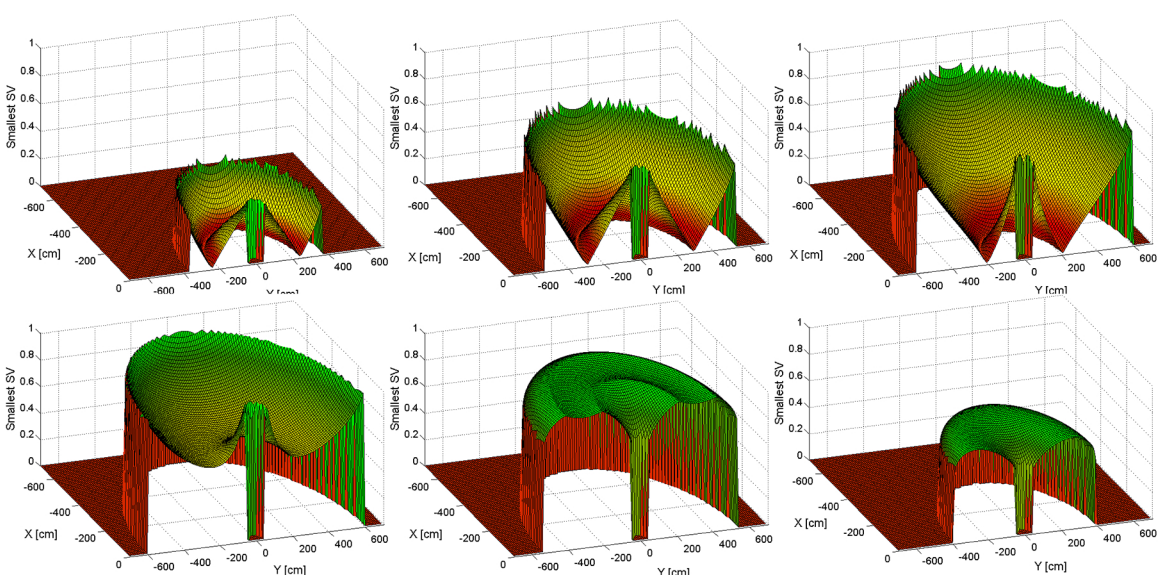


Figure 2.5: Smallest Singular Value value in several $X-Y$ planes (with z=0mm, z=200m, z=400mm, z=600mm, z=800mm, z=1000mm respect to the robot base)

Condition Number

The Condition Number is a measure of the J matrix ill-conditioning. Introduced in [109] as a measure of the distance of a robot to a singularity, the Condition Number is the relation between the largest and the smallest singular values of J, (2.11).

$$CN = \frac{\sigma_{max}}{\sigma_{min}} \tag{2.11}$$

The Condition Number measures the Jacobian matrix ill-conditioning. When the manipulability is optimal (equal capability of generating movement in all directions), all its singular values are equal; $\sigma_{max} = \sigma_{min} \rightarrow CN = 1$.

Positive aspects are, first, CN is scale independent and, second, it can be used as a kinematic accuracy measure. On the contrary, CN presents a drawback: no analytical expression, as a function of joint angles, can be obtained. From the MRCP point of view, CN cannot be used as a dexterity measure because the high range of CN is not well fitted: $Range_{CN} = [1, \infty)$. Fig.2.6 shows the CN value for several $X - Y$ planes for the studied StaubliRX60B robot. In the first three sampled planes the non fitted value of CN can be observed. Neither the boundary nor the internal singularities can be detected in advance, preventing CN to be used for control purposes.

In [110] the CN is used to obtain a global performance index: Global Conditioning Index, GCI, which uses the maximum achievable CN in all the reachable workspace to normalize the value of the local CN obtained when the robot is evaluated in a certain position.

Reciprocal Condition Number

The RCN index, derived from CN and designed to solve its non closed upper range problem, is the result of the inverse of the CN, (2.12). Mathematically, RCN is the ratio between the minimum and maximum singular values. The RCN range is well fitted, $Range_{RCN} = [0, 1]$, being $RCN = 1$ when the robot presents better manipulability, $\sigma_{min} = \sigma_{max}$ and $RCN = 0$ when the robot is in a singularity, $\sigma_{min} = 0$.

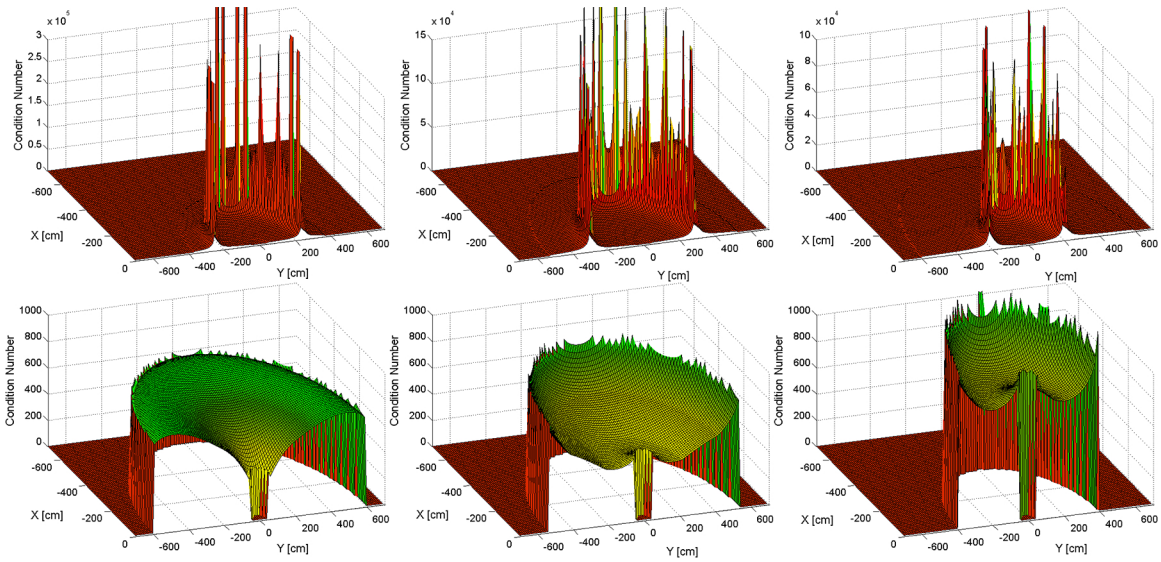


Figure 2.6: Condition Number value in several $X-Y$ planes (with z=0mm, z=200m, z=400mm, z=600mm, z=800mm, z=1000mm respect to the robot base)

$$RCN = \frac{1}{CN} = \frac{\sigma_{min}}{\sigma_{max}} \tag{2.12}$$

The behaviour of RCN is similar to the Minimum Singular Value: the internal singularities can be detected in advance but, on the contrary, the boundary ones cannot always be detected.

Trace of J

The trace of the Jacobian matrix, Ψ, is defined as the arithmetic mean of the JJ' eigenvalues, (2.13).

$$\Psi = \frac{trace(JJ')}{m} = \frac{\sum^{m} \lambda_i}{m} \tag{2.13}$$

The trace of the Jacobian matrix, Ψ, can be used to detect internal singularities, but it cannot be used as a generic singularity detection index because it can not detect in advance boundary singularities. This index has been included in this review because it is used as a part of other dexterity indexes, like Isotropy. As shown in Fig.

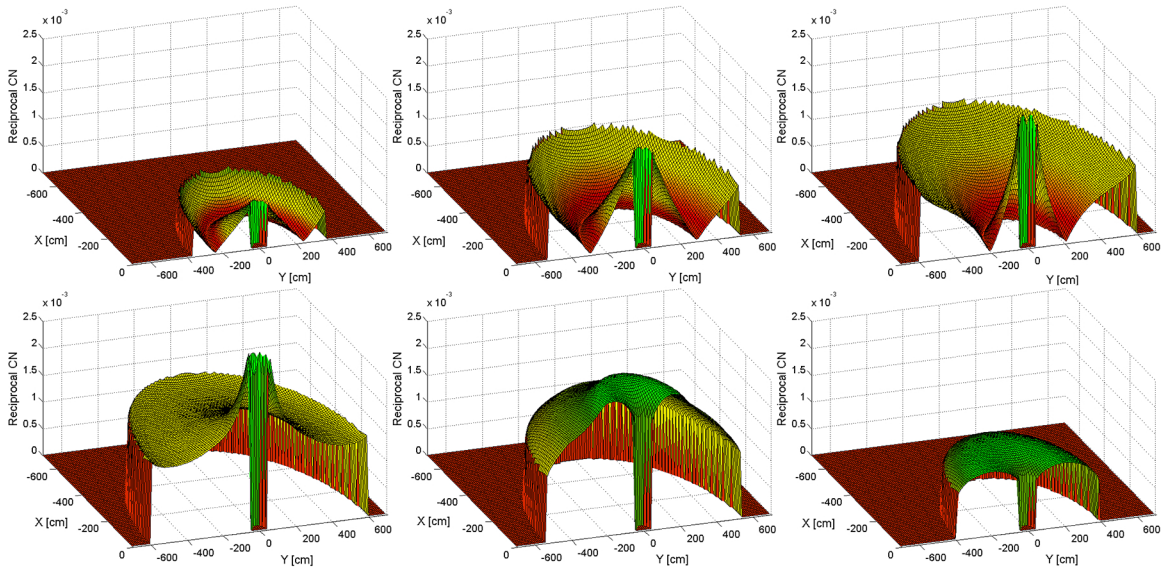


Figure 2.7: Reciprocal Condition Number value in several $X-Y$ planes (with z=0mm, z=200m, z=400mm, z=600mm, z=800mm, z=1000mm respect to the robot base)

2.8, where the values of Ψ for several $X-Y$ planes can be observed, only the internal singularities are detected in advance: $\Psi \to 0$.

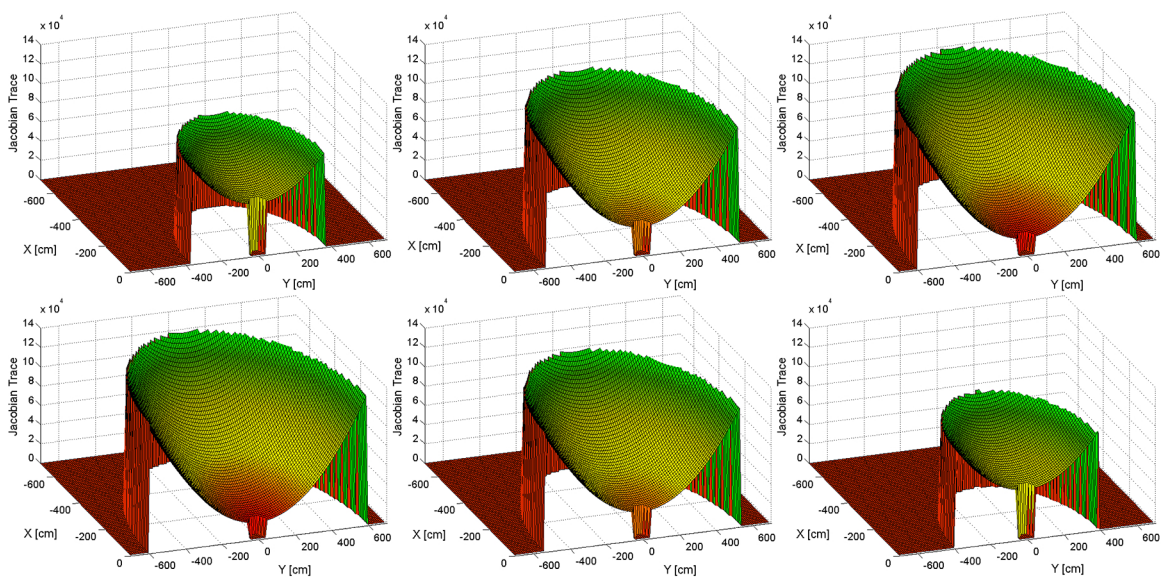


Figure 2.8: Trace value in several $X-Y$ planes (with z=0mm, z=200m, z=400mm, z=600mm, z=800mm, z=1000mm respect to the robot base)

Isotropy

Isotropy can be defined as the degree of uniformity of the robot to generate movements in all directions from a concrete pose. This property is used by [105] as a measure of robot dexterity: the degree of isotropy in the Jacobian matrix denotes the ability of a robot to perform a movement in all directions. The isotropy measure is the ratio between the geometric mean, M and the arithmetic mean, Ψ of the eigenvalues of JJ' in a m dimensional workspace, (2.14).

$$\Delta = \frac{M}{\Psi} = \frac{(det(JJ'))^{1/m}}{trace(JJ')/m} \quad \text{, where } M \leq \Psi \Rightarrow \Delta \in [0,1] \tag{2.14}$$

The isotropy measure presents two advantages as an evaluation index. First, its value is well fitted ($M \leq \Psi$), being $\Delta = 1$ when the robot is in the configuration with optimal dexterity (JJ' is isotropic, $w = \Psi$) and $\Delta = 0$ when the robot is in a singularity (JJ' losses rank and, consequently, $w = 0$). Second, the measure of isotropy is scale independent since both components, M and Ψ have dimension of $length^2$.

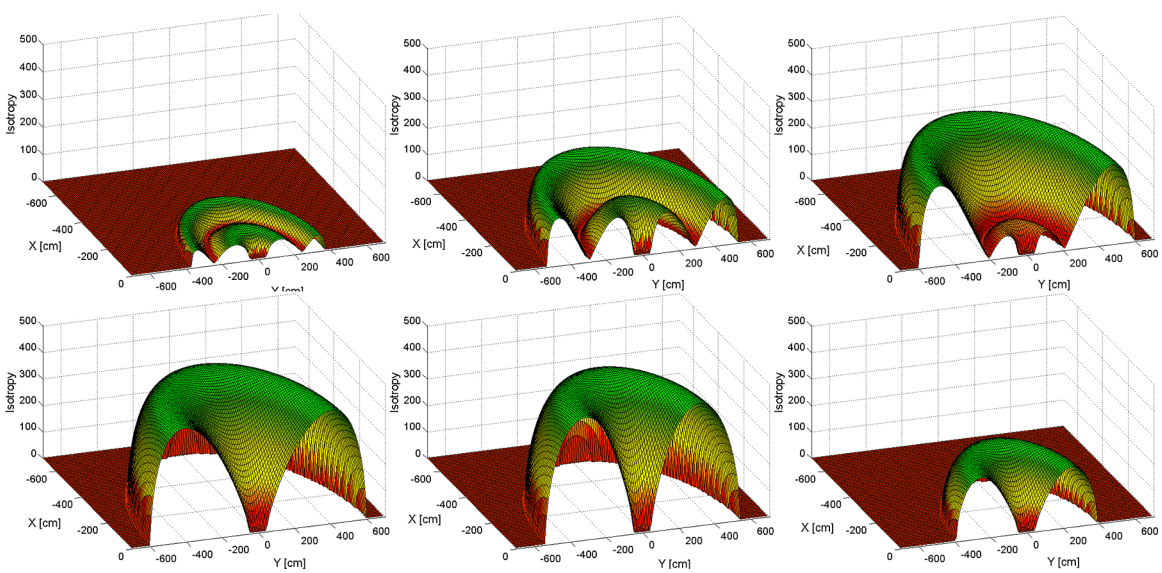


Figure 2.9: Isotropy value in several $X-Y$ planes (with z=0mm, z=200m, z=400mm, z=600mm, z=800mm, z=1000mm respect to the robot base)

Unlike Manipulability and Reciprocal Condition Number, Isotropy presents smooth gradients in all types of singularities: internal and boundary. This property makes this evaluation index a feasible candidate to be used for real time robot control.

Global Evaluation Indices

The dexterity indices reviewed above can be classified as local (position dependent evaluators). A different approach to evaluate robots dexterity is the use of global evaluators. These global indices use the whole workspace information to determine the dexterity of robots in their current configuration with respect to the best evaluation value around the workspace. There's not a better evaluation approach; the decision about which method (local or global) must be used depend on the evaluation purposes.

As the global indexes generate relative values normalized with respect to some maximized criteria inside their workspace, instead of generating absolute values (with or without physical meaning), they can be used to compare different robots only when they present similar characteristics in terms of kinematics, dynamic capabilities, ... This limitation affects the MRCP performance, preventing its use when the robots forming the slave station are different.

One of the most used global evaluators is the Global Isotropy Index, GII. The GII, proposed in [111], is the global version of the Reciprocal Condition Number and is used as a design index. For instance, GII is used by the authors in [112] to design the mechanism of a haptic device (a haptic pen). Fig2.10 shows the GII workspace in several $X-Y$ planes. In [110] the Global Conditioning Index, GCI is used as a performance measure for robot control. GCI is defined as the ratio between the inverse of the Jacobian condition number integrated over the reachable workspace and the volume of the workspace, (2.15), where η is the Global Conditioning Index. CN is the Condition Number of a concrete robot configuration.

$$\eta = \frac{\int_w CN^{-1} \partial w}{\int_w \partial w} \tag{2.15}$$

With GCI the workspace can be mapped and used as a guide for trajectory gen-

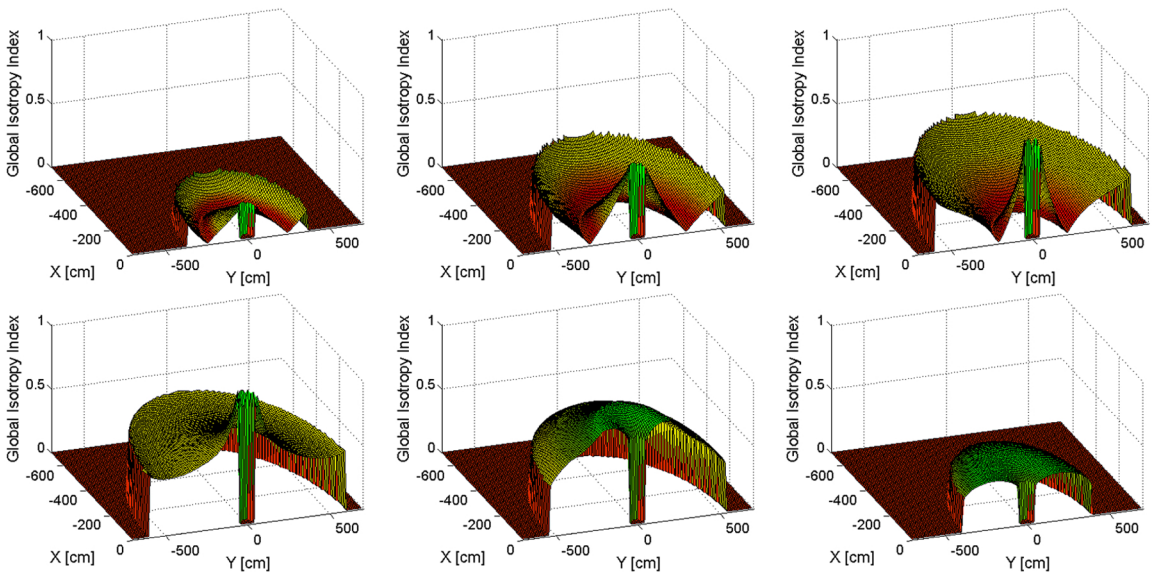


Figure 2.10: Global Isotropy Index value in several $X-Y$ planes (with z=0mm, z=200m, z=400mm, z=600mm, z=800mm, z=1000mm respect to the robot base)

eration. Examples of the use of GCI and other global evaluation indices, like Global Velocity Index, can be found in [113] and [114].

For real time control strategies, the use of global evaluation workspace techniques is not a feasible solution. In these control techniques, the robot path is defined using the map information and, consequently, the described path improves certain criteria. This approach requires a previous off-line map computation. Unfortunately, these mapping approaches have a drawback: the high complexity maps generated when working in high dimensional workspaces, prevent its real-time applicability.

Task Oriented

The Task Oriented manipulability indices arise from the need of obtaining an accurate evaluation of a robot in a determined task. These indices are also used as a criteria to design robots for a concrete task. In these cases, obtaining a measure related with a task is more effective than a generic one. Most of these measures are based on manipulability and are commonly known as Task Oriented Manipulability Measures, TOMM. Although the specific robots are out of the scope of this work, some examples

are mentioned next, providing a complete review of TOMM. In underwater robotics, some examples can be found, as in [115], where task oriented force manipulability, TOFMM, and ellipsoid, TOFME, are presented. In the field of bioinspired robotics, several examples can be cited. To mention one such example, in [116] the locomotion system of a cricket is reproduced. To evaluate the jumping behavior, a TOMM based on velocity and a TOMM based on force are proposed. An interesting TOMM measure from the MRCP point of view is the Desired Manipulability Ellipsoid, proposed in [82, 117]. The manipulability ellipsoid provides two types of information: a scalar evaluation of a robot manipulability (proportional to the ellipsoid volume) and directional information obtained from its shape. Using this second value, the authors propose to stablish the most convenient ellipsoid shape at each task step. The major axis of the ellipsoid must coincide with the trajectory direction. When there is not a preferred movement direction, all the ellipsoid axes must be equal (indicating equal manipulability robot capabilities in all directions). On the contrary, when the robot must follow a certain movement direction, the ellipsoid must change its shape increasing the length in that direction. The use of optimal manipulability ellipsoid is illustrated with a pick and place task, Fig.2.11. In the first picture of the sequence of snapshots, Fig.2.11.a, there is no preferred movement direction and, in consequence, the ellipsoid has a spherical shape (all major axis have the same lenght). In the next step, Fig.2.11.b, the manipulated object must be moved along the y axis: the ellipsoid has deformed its shape, increasing the y major axis. In the third step, Fig.2.11.c, the ellipsoid retakes the spherical shape indicating non-preferred direction. Finally, Fig.2.11.d, the preferred direction z deforms the ellipsoid in the respective z principal axis.

This TOMM measure is obtained using the closeness between the desired and the real ellipsoid. Two different metrics are proposed to measure the closeness: the intersection volume and the shape discrepancy. In redundant robots, the maximization of ellipsoids closeness can be used as a criterion to stablish the robot joint configuration.

The use of TOMM in MRCP presents some positive aspects, but also some drawbacks. Regarding the positive aspects, measuring the closeness of the robots can be

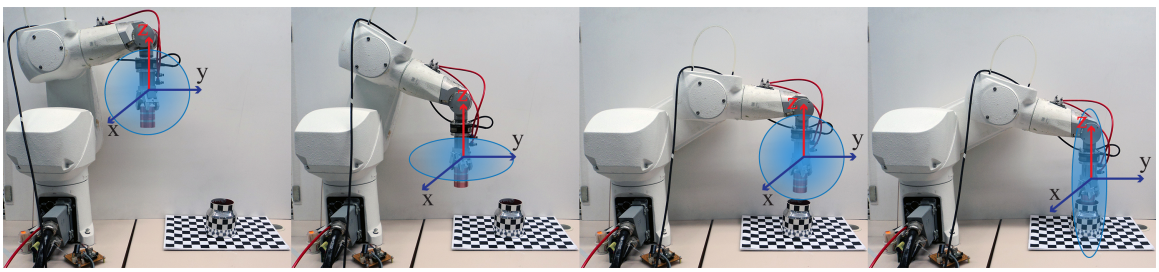


Figure 2.11: Desired Manipulability Ellipsoid example on a pick and place task using a 6 DoF robot

used to determine the robot availability. However, TOMM does not offer such an absolute measure of robot availability. Consequently, a diminution of the ellipsoids closeness cannot be imputed to a shape change or a volume decrease.

Manipulability of multi-robot systems

Manipulability indices are also applied in multi-robot systems, where the robots form a closed kinematic chain. Several examples can be mentioned, like the manipulability generated by different robots generating a closed kinematic chain, the optimal configuration and the design of each finger in multi-fingered end-effectors, etc. Again, in multi-robot systems the manipulability can be used for both, the design and the control of robots. An illustrative work to understand the manipulability ellipsoid and the force ellipsoids (internal forces, external forces, etc) formed by closed kinematic chains can be found in [118].

In [119] the velocity and force ellipsoids (in the form of active, passive and internal forces) for a closed kinematic chain are presented. A five joints chain is used to illustrate several examples of the presented ellipsoids. In [120–122], a geometric based approach to detect singular configurations and compute manipulability for multi-robot systems is presented. Authors use the Riemann geometry formulation to describe the closed chain in study. The singularities are classified into configuration space singularities (joint limits), actuator singularities and end-effector singularities. An example of each case is presented as well as several examples of the resulting manipulability ellipsoid of the closed chain.

Task oriented manipulability is applied in cooperative robotics. In [123], task-oriented manipulability measure, TOMM, is extended to task-oriented dual-arm manipulability measure, TODAMM. The formulation of TODAMM is based on a function of the derivative of each arm trajectory and the relative freedom between their respective movements. The cooperation between two robots is divided in tight cooperation (both robots holding a single rigid object and no relative movement freedom between them) and loose cooperation (each robot executes a independent subtask with no motion constrains between them). The dual-arm manipulability measure, DAMM, for tight and loose cooperation is defined as the maximum volume of intersection between the manipulability ellipsoid of each robot. The combination of the TOMM and DAMM concepts, define TODAMM, as the intersection volume between the optimal task ellipsoid and the ellipsoid generated by the dual-arm system.

Chapter 3

MRCP Robot Suitability Evaluation

3.1 Introduction

This chapter reviews the methodology and the indices designed to evaluate the suitability of each slave robot composing the MRCP all along the execution of a task. The evaluation of each robot suitability has to provide a complete adequacy estimation of every slave robot, taking into account all the aspects that can impede a robot to continue with the execution of the ongoing teleoperation task.

The suitability evaluation criteria have been defined considering:

- The different causes that can prevent the operative robot from continuing the execution of a task.
- The suitability of the rest of robots to replace the operative robot in the execution of the task.
- The uncertainness produced by the lack of path planning in teleoperated tasks.
- The computational cost required to achieve real time operation.

The causes that prevent a robot from continuing the execution of a task can be classified, under the MRCP point of view, in: intrinsic, extrinsic and task dependant. For each of them, several indices have been defined following the above premises.

- Intrinsic evaluators: Measures related to intrinsic robot aspects that define their capability to execute the desired task. Joint Limits and kinematic based indices are examples of intrinsic evaluators.

- Extrinsic evaluators: Measures related to the interaction between the robots and their workspace. The estimation of collision risk between robots and obstacles is an example of extrinsic evaluator.

- Task oriented evaluators: Measures that are designed to accomplish with the specifications or restrictions imposed by the task. Visibility constrains in teleinspection or maximum tool orientation in the fulcrum point in robotic minimally invasive surgery are examples of task dependant evaluators.

3.2 Robot Suitability Evaluation specifications in MRCP

Robot Suitability Evaluation is the first step of the MRCP control schema. Every robot is evaluated from a set of indices that cover the different aspects that determine their suitability. As input, each robot evaluation relies on data coming from the master device, the robot controllers and the workspace sensors. The resulting robot evaluation is used by the Suitability Robot Selection Module. Fig. 3.1 shows the integration of this evaluation process in MRCP, where $REval_i$ is the evaluation module of the ith robot using k different evaluation indices.

The implementation of the indices is conditioned by the context in which they are applied. Several specifications are imposed to the evaluation indices to enable their use in MRCP. These specifications should be taken into account when designing a new evaluation index or to adapt an existing one to the MRCP control.

The first requisite refers to the need to achieve real time control preventing potential delays, which decrease the operator performance and destabilize the system. To mitigate its effects, complex control strategies are required. Delays are consequence

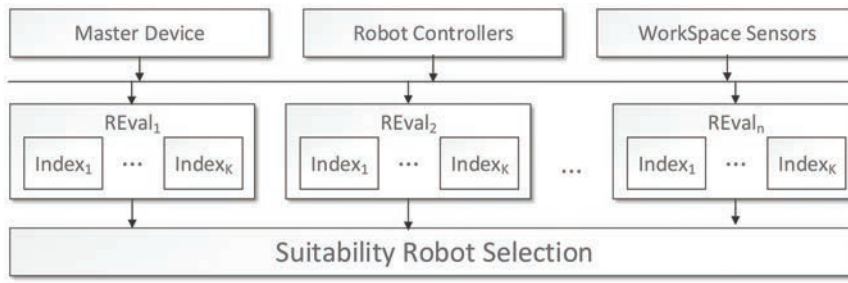


Figure 3.1: Schema of the Robot Evaluation Index inside the MRCP control architecture

of, basically, two aspects: first, communications delays (due to long distances between master and slaves, inefficient communication protocols, network saturation or instability) and, second, complex control schemas with high computational costs. In MRCP the communication delay is out of the research scope and is assumed to be null. Concerning control delays, the problem has been solved designing a real time control based on an efficient computational schema.

Another issue is the lack of path planning in teleoperation. When a path can be planned, robot evaluation indices can be applied to obtain the robot suitability along this path. These indices could also be used to determine the optimal path. In teleoperation there is no possibility to know in advance the robot's path. The adopted solution is to adapt these indices to a short term trajectory prediction. Prediction minimizes the potential disturbances that might affect a teleoperated task. Disturbances are produced by time delays or undesired trajectory modifications. The evaluation indices are adapted to predict the future path so as to be able to foresee task evolution, and thus, decide future actions in advance. These indices do not only evaluate the robots in their current configuration, but also in several future positions of the predicted trajectory. This methodology observes the uncertainness associated to each predicted position.

A final point to be considered is the indices heterogeneity, since multiple evaluation indices of different nature are used. Therefore, a common metrics must be established to compare them. Every index generates a numerical robot evaluation value in its

own space. Consequently, when several evaluation indices are used, they generate non directly comparable results. This problem also appears when an evaluation index is scale or order dependant, like, for instance, most of the dexterity indices based on the study of the robots Jacobian matrix, as introduced in [105].

The MRCP robot evaluation indices have been designed considering the need of real time control and trajectory prediction. The non comparativeness of indices is solved in posterior control steps by transforming them to the time domain. With this transformation, all robots and their associated indices can be directly compared, as will be explained in Section 4.5.2. To achieve real time, the MRCP indices present closed form solutions and sampled based solutions.

3.3 Extension of robot suitability evaluation to the predicted trajectory

In the MRCP context arises the need of estimating the evolution of the robot evaluation indices along a predicted trajectory, with the aim of obtaining the robot availability in the immediate future. Consequently, the robot evaluation indices must be adapted to deal with a sequence of predicted robot positions. Working with a predicted trajectory entails to deal with the uncertainness associated to each predicted point. The methodology for estimating the future robot evolution is based on a) generating the trajectory prediction from the previous observed points, b) computing the uncertainness associated to each predicted point and, finally, c) evaluating the robot along that trajectory. Depending on the trajectory prediction methodology, the uncertainness will be computed and expressed in a different manner.

MRCP does not impose a concrete trajectory prediction methodology and is responsibility of the user to select the most appropriated, taking into account the task characteristics and the real time requirements. The MRCP requirements with respect to the trajectory predictor depends only on the predicted points representation: each point must be expressed as a probability distribution function with mean

on the own predicted point and its associated uncertainness. In normal distribution functions, each predicted point is described by its mean and standard deviation: $\Omega_{i+k} = (p_{i+k}, d_{i+k})$ for the kth predicted point.

3.3.1 Case study: Polynomial Models generated with the Least Squares Curve Fitting technique

This case study presents the use of polynomial models to fit the previous observed points and predict the new ones. Several polynomial fitting techniques can be included in this methodology like least squares and all its variants (weighted least squares, LOSS,...), Lagrangian and Newton interpolating polynomials, splines, etc, [124]. The presented fitting technique is the least squares curve fitting which offers a good balance between computational costs and accuracy.

To obtain the fitting polynomial in the current instant of time $t = i$, the sequence of the last k robot trajectory observed points, $p_{i-k}, p_{i-k-1}, ..., p_{i-1}, p_i \in R^n$, are required. Given the observed points, the trajectory can be modelled as a polynomial with degree, usually, no greater than three to avoid over-fitting. The polynomial is obtained, in this case study, applying the least squares fitting methodology, (3.1).

$$p = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_l x^l \tag{3.1}$$

Once obtained, the fitting polynomial is used to generate the predicted points in which the robot will be evaluated. (3.2) is the generic form of the second order polynomial that generates all the predicted points p_o.

$$\hat{p}_o = \beta_o + \beta_1 x_o + \beta_2 x_o^2 \tag{3.2}$$

A quantifiable uncertainness is associated to every predicted point p_o formulated as a confidence interval of $1 - \alpha$, centred in p_o and with limits defined as, (3.3).

$$\hat{p}_o \in \left(p_o \pm t_{\alpha/2} \hat{S}_R \sqrt{1 + v_{hh}} \right) \tag{3.3}$$

where $\hat{S}_R^2$ is the estimated variance, (3.4).

$$\hat{S}_R = \frac{\sum e_i^2}{n-(k+1)} \tag{3.4}$$

and

$$v_{hh} = x_h'(X'X)x_h \quad ; \quad \frac{1}{n} \leq v_{hh} \leq 1 \tag{3.5}$$

and X is the data matrix. The columns of X are the independent variables plus an additional first column formed by ones, (3.6).

$$X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \cdots & \cdots \\ 1 & x_n \end{pmatrix} \quad ; \quad X'X = \begin{pmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix} \tag{3.6}$$

The probability distribution inside the confidence interval is distributed as a normal centred in $\hat{p}_o$, (3.7), [125].

$$p_o \sim N\left(\hat{p}_o, \sigma\frac{\sqrt{\hat{n}_h+1}}{\sqrt{\hat{n}_h}}\right) \tag{3.7}$$

where $\hat{n}$ corresponds to the equivalent number of observations to estimate the mean predicted value, $E[\hat{y}_o] = m_o$

$$\hat{n} = \frac{1}{v_{hh}} \tag{3.8}$$

Fig. 3.2 shows an illustrative example of a polynomial fitting, the extrapolation curve and the associated confidence intervals. In the same figure, a predicted point p_o with its confidence interval $[CI_{Sup}, CI_{Inf}]$ and the associated probability distribution function $N(p_o, d_o)$ is shown.

The extension of robot evaluation to predicted trajectories using polynomial fitting techniques is based on the concept of mathematical expectation, weighting each value

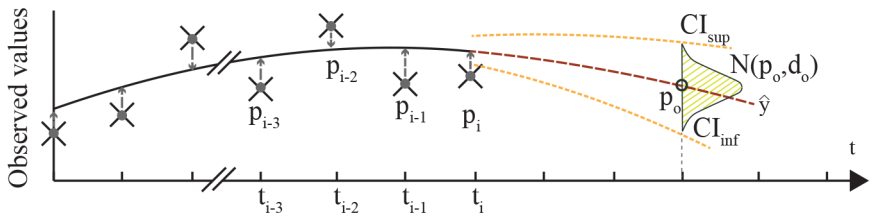


Figure 3.2: Example of a polynomial fitting and the extrapolation curve. On the same picture: a single predicted point p_o with its confidence interval and probability distribution function.

inside the confidence interval with its occurrence probability. Let $y = f(x)$ be a function of x, the mathematical expectation or expected value of y is the weighted sum of all $f(x)$ values, using as weight the occurrence probability of each $f(x_i)$, (3.9).

$$E(y) = E[f(x)] = \sum_{-\infty}^{\infty} f(x)P(x) \tag{3.9}$$

Based on the mathematical expectation and, observing the resulting expression for each predicted point (3.7), the evaluation index measures the index value of the robot in this interval. The obtained $1 - \alpha$ confidence level, taking into account the probability distribution of each value inside this interval, (3.10).

$$E(\hat{p}_o) = \int_{CI_{Inf}}^{CI_{Sup}} f(\hat{p}_o)P(\hat{p}_o = p_o)dp \tag{3.10}$$

where the integration limits CI_{Sup} and CI_{Inf} are the confidence limits, defined as (3.11).

$$\left.\begin{aligned} CI_{Sup} &= +t_{\alpha/2}\hat{S}_R\sqrt{1+v_{hh}} \\ CI_{Inf} &= -t_{\alpha/2}\hat{S}_R\sqrt{1+v_{hh}} \end{aligned}\right\} \tag{3.11}$$

3.4 Intrinsic Evaluators

The intrinsic evaluators measure parameters of the own robot, like kinematic or dynamic limitations, joint limits, etc. In this research, two families of intrinsic evaluators have been used: those that are mapped on the joint space and those mapped on the workspace. The former are related with the static join properties (joint limits) and the dynamics (maximum achievable joint velocities and accelerations); whereas the latter are related to kinematic aspects of the robot, specially focused on robot singularities and their implications in control and teleoperation execution.

3.4.1 Robot Joint Evaluation, RJ

The Robot Joint Limits index, RJ, is designed to evaluate the robot in the joint space. Several works have been developed to control robots in joint space. Examples can be found in [126], where the kinematic constraints imposed by workspace obstacles are represented in the joint space. The advantage of this method is the possibility of describing the obstacles as a set of parametric equations. In [127] the robots joint space is used for constrained motion planning, which is useful for many real world tasks (open a drawer or a door, holding an object, etc). Two space sampling methodologies are presented which allow the planing of motions with constraints as, for instance, opening a door while avoiding an obstacle. The presented motion planning is designed for redundant robots. [128] presents a methodology to overcome joint limitations in terms of joint range, velocity and acceleration limits by working in motion joint space. An iterative method ensures a motion plan that respects joint limitations.

The Joint Limits index, RJ, is designed to evaluate the risk of a robot to reach a joint limit. The measure is extended to fulfil the MRCP requirements, computing the joint limit risk during the execution of a predicted trajectory. The risk is measured in terms of the minimum distance between the current position of each joint, θ_i, with respect to its limits, $[\theta_{i,min}, \theta_{i,max}]$. The use of distance as an evaluation index does not provide a realistic estimation of the joint limit: a better evaluation is obtained pointing at the instant when distance starts to represent a significant risk in the

proximity of each joint limit. With this aim, a continuous and smooth modulation function is applied to the joint distance computation, $dist(\theta_i)$, generating the $J_{eval,i}$ index for every θ_i joint, (3.12).

$$J_{eval,i} = (1 - \cos(\text{dist}(\theta_i)^{1/b_i})\pi)/2 \ , \ \forall b_i \geq 1 \text{ and } i = 1..n \text{ joints} \tag{3.12}$$

where b_i is used as modulation parameter and $\text{dist}(\theta_i)$ is the minimum Euclidean distance between the current position of each joint and its limits $[\theta_{min}, \theta_{max}]$, (3.13).

$$\text{dist}(\theta_i) = \frac{\min(|\theta_i - \theta_{max}|, |\theta_i - \theta_{min}|)}{|\theta_{max} - \theta_{min}|/2} \tag{3.13}$$

where min is the minimum function.

Fig.3.3 shows the modulation function J_{eval} using different values of the modulation parameter b, $(1 \leq b \leq 6)$, for a rotational joint θ.

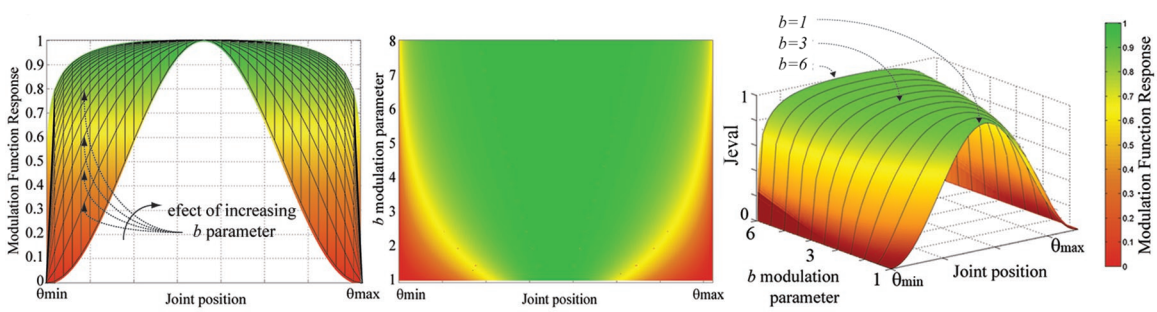


Figure 3.3: Effect of b parameter in the modulation of the Joint Limit index in a rotational joint. Range of the joint: $[\theta_{min}, \theta_{max}]$

Fig3.4 and Fig. 3.5 show, from different points of view, several modulated robot joint evaluations of a Staübli RX60B in the plane $z = -400mm$ using several b modulation parameter values.

For a better robot joint evaluation, J_{eval} must reflect not only the distance to a joint limit (in its modulated expression), but also its approaching velocity. The modulation parameter b can be dynamically adjusted to reflect this velocity. Numerically, b varies its value inversely proportional to the ratio between the current joint velocity, $\dot{\theta}_i$, and the maximum joint velocity, $\dot{\theta}_{i,max}$, as shown in (3.14).

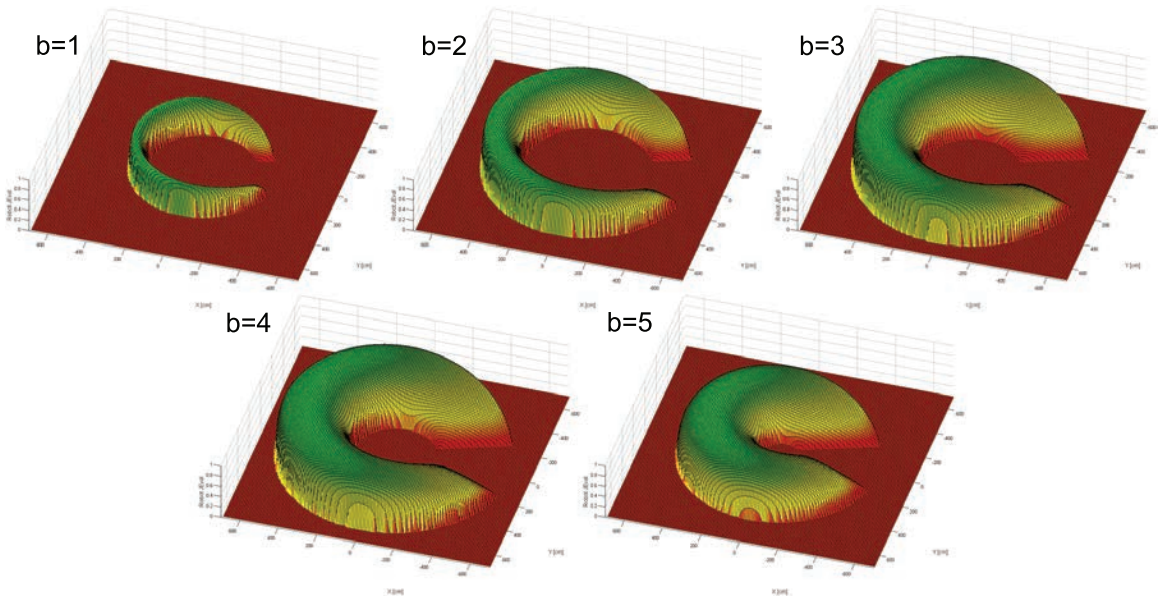


Figure 3.4: Modulation of the Joint Limit index in a rotational joint. Range of the joint: $[\theta_{min}, \theta_{max}]$

$$b_i = (b_{max} - b_{min})\frac{\dot{\theta}_i}{\dot{\theta}_{i,max}} + b_{min} \ , \ \forall i = 1..n \text{ joints} \tag{3.14}$$

Once computed the J_{eval} for all robot joints, the final robot joint evaluation is determined by the joint that presents higher risk to reach a joint limit (worst value). This value is set using the minimum of all the J_{eval}, previously calculated, (3.15).

$$RJ = \min(J_{eval,i}) \ , \ \forall i = 1..n \text{ joints} \tag{3.15}$$

Fulfilling the MRCP requirements, the J_{eval} index is extended to evaluate a robot during the execution of a predicted trajectory. To obtain a realistic estimation along the trajectory, the uncertainness associated to each predicted point, $\Omega(p_i, d_i)$, is reflected by means of reducing the valid joint ranges. To obtain the estimated J_{eval} at every predicted point, two aspects must be computed (at each predicted point): the robot joint position and their new reduced joint ranges, (3.16).

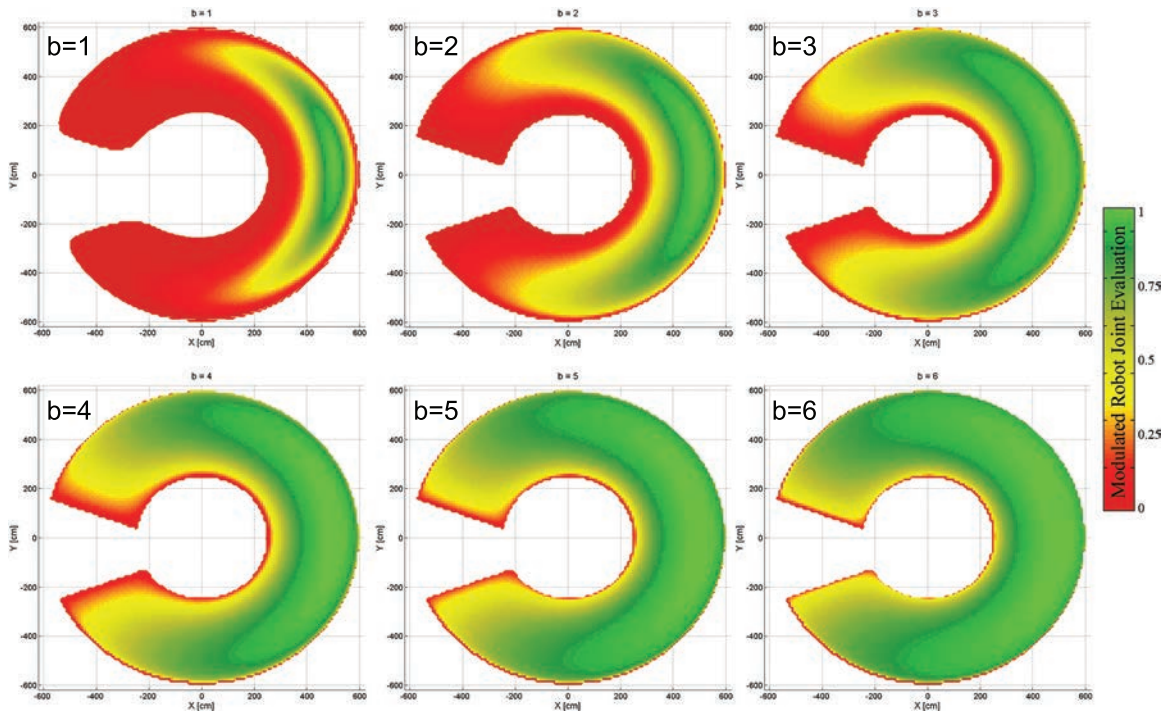


Figure 3.5: Top view of the modulation of the Joint Limit index in a rotational joint. Range of the joint: $[\theta_{min}, \theta_{max}]$ and $b = [1, .., 6]$

$$\begin{aligned} &\Omega(p_{i+k}, d_{i+k}) \rightarrow (\vec{\theta}_{i+k}, [\vec{\theta}_{min,i+k}, \vec{\theta}_{max,i-m}]) \\ &\forall k = 0..K \text{ predicted points, and} \\ &i = \text{current evaluation time} \end{aligned} \tag{3.16}$$

The corresponding joint position, $\vec{\theta}_{i+k}$, for each predicted point is computed applying the inverse kinematics to the corresponding p_{i+k}: $p_{i+k} \rightarrow \vec{\theta}_{i+k}$. To fix the new joint ranges, the inverse kinematics applied to the probability distribution function does not offer a computationally feasible solution. An alternative is the use of the observed errors between the current robot joint configuration and the estimated values at previous trajectory predictions.

For a better understanding of the used computational methodology, a new nota-

tion is proposed: $p_{i,k}$ where i is the ith previous trajectory prediction (i evaluation times before the current evaluation) and k denotes the kth point of the predicted trajectory. Following this notation, $\{p_{0,0}, \Omega(p_{0,1}, d_{0,1}), ..., \Omega(p_{0,K}, d_{0,K})\}$ represents the predicted trajectory formed by K points and generated from the current robot pose $p_{0,0}$ and $\{p_{i,0}, \Omega(p_{i,1}, d_{i,1}), ..., \Omega(p_{i,K}, d_{i,K})\}$, the ith previous predicted trajectory generated from the robot pose in the ith previous evaluation time, $p_{i,0}$.

The new joint limits of the kth predicted point of the current trajectory prediction, $p_{0,k}$, $[\theta_{max,k}, \theta_{min,k}]$ are obtained as shown in (3.17).

$$\left.\begin{aligned}\vec{\theta}_{min,k} &= \vec{\theta}_{min} - \vec{e}_k \\ \vec{\theta}_{max,k} &= \vec{\theta}_{max} - \vec{e}_k\end{aligned}\right\} \text{ where } \vec{e}_k = \vec{\theta}_{i,0} - \vec{\theta}_{i-k,k} \tag{3.17}$$

Supposing the robot position at the current instant of time ($t = i$) $p_{i,0}$ and, generating trajectory predictions composed of two points ($p_{i,1}, p_{i,2}$), the new limits are established as, (3.18).

$$\begin{aligned}
&p_{i,0} \rightarrow \vec{\theta}_{i,0}, [\vec{\theta}_{i,min}, \vec{\theta}_{max}] \\
&p_{i,1} \rightarrow \vec{\theta}_{i,1}, [\vec{\theta}_{i,min} - \vec{e}_1, \vec{\theta}_{i,max} - \vec{e}_1],\ \vec{e}_1 = \vec{\theta}_{i,0} - \vec{\theta}_{i-1,1} \\
&p_{i,2} \rightarrow \vec{\theta}_{i,2}, [\vec{\theta}_{i,min} - \vec{e}_2, \vec{\theta}_{i,max} - \vec{e}_2],\ \vec{e}_2 = \vec{\theta}_{i,0} - \vec{\theta}_{i-2,2}
\end{aligned} \tag{3.18}$$

Once the new joint ranges are fixed and J_{eval} for all the predicted points are computed, the evaluation of each joint along the predicted trajectory, $\overline{J}_{eval}$, can be computed. $\overline{J}_{eval}$ is obtained as the weighted sum of all the J_{eval} along the K point predicted trajectory, as shown in (3.19), where the weights w_k balance the contribution of each J_{eval_k} and K is the total amount of generated prediction points.

$$\overline{J}_{eval} = \frac{\sum_{t=i}^{K}(w_t J_{eval_t})}{\sum_{t=i}^{K}(J_{eval_t})},\ w_k = \frac{1}{k+1} \tag{3.19}$$

Finally, the complete robot joint evaluation during a predicted trajectory, RJ

index, is fixed using the same methodology used by a single robot pose: the robot joint that is closer to one of its limits during the predicted trajectory. This value is set using the minimum of all the $\overline{J}_{eval}$, previously calculated, (3.20).

$$RJ = \min(\overline{J}_{eval,i}) \; \forall i = 1..n \text{ joints} \tag{3.20}$$

Fig.3.6 illustrates an example of the RJ index evolution of two robots during the execution of a telemanipulation task consisting in rotating an object. The figure shows a sequence of snapshots of the robots in different configurations along the task and the corresponding robot Joint limits. The object has two predefined grasping positions and the robots involved in the task (simulated Staubli Rx60B) have 6 DoF with the imposed joint limits shown in Table 3.4.1.

Table 3.1: Imposed Robot Joint Limits

Joint Number	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
JLim 1 [rad]	-2.79	-3.80	1.66	-3.05	0.09	-3.05
JLim2 [rad]	2.79	1.65	3.92	3.05	2.01	3.05

3.4.2 Anisotropic Dexterity Index, Θ

The Anisotropic Dexterity Index Θ, based on the directional manipulability ψ_u, measures the dexterity of a robot along a trajectory. As previously explained, ψ_u index evaluates the capacity of a robot to generate movement from its current configuration in a concrete direction. MRCP requirements impose that the index must evaluate a robot not only in a single joint configuration, but during the execution of a predicted trajectory, and it must be performed in real-time. The Anisotropic Dexterity Index, Θ, is designed to fulfil these requirements.

In order to illustrate the dexterity evaluation of a robot following a trajectory, Fig.(3.7) shows the evolution of the manipulability ellipsoid (reduced to the three po-

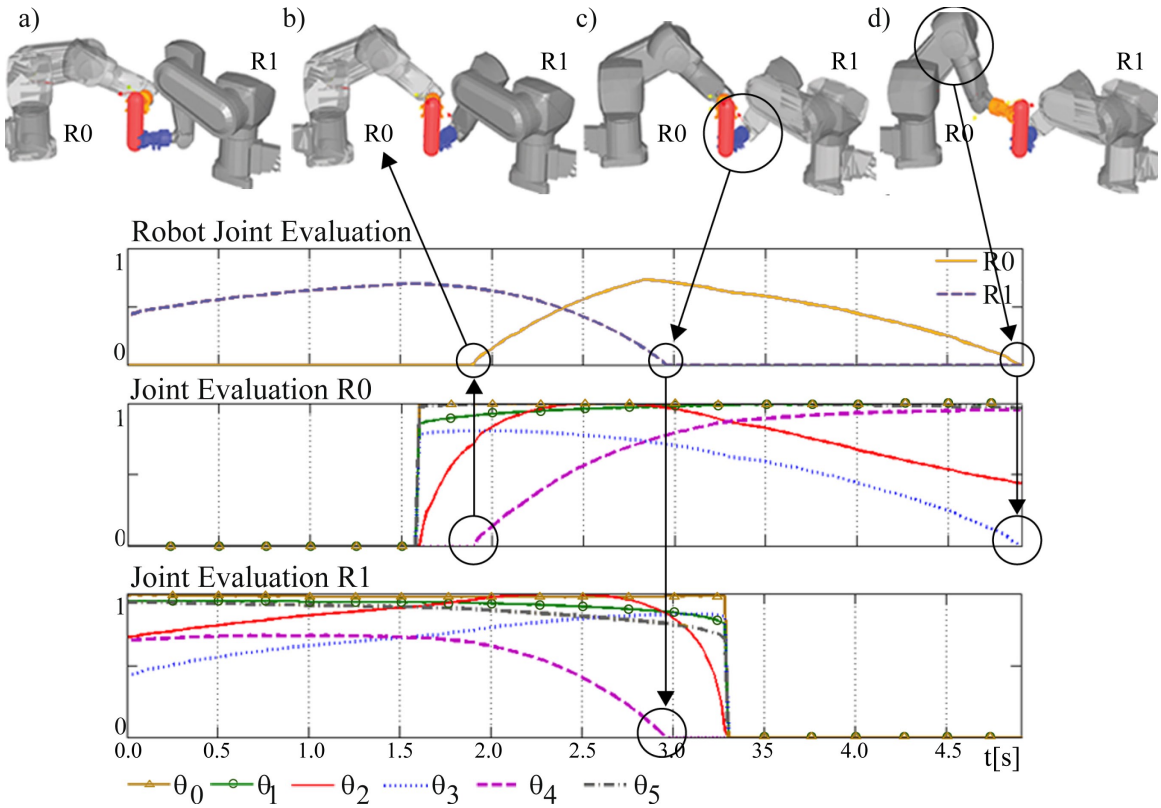


Figure 3.6: RJ index for two robots along the execution of a teleoperated task. Upper graphic shows the evolution of RJ_0 and RJ_1. Middle and lower graphics show the $\overline{J}_{eval}$ in every joint for both robots.

sition components (x, y, z)) during the execution of a linear trajectory. The closer the robot is to a singularity, the smaller becomes the ellipsoid. The ellipsoid degenerates when the robot falls into a singularity, loosing one or more dimensions.

The explanation of Θ index will be conducted as follows: first, the directional manipulability index, ψ_u, on which the Θ index is based, is shortly reviewed. Then, the ϕ^i_{i+1} index, which computes the directional manipulability from a single point to a predicted point is explained. This index is extended to compute the dexterity value when the origin point is also a predicted point: Φ^i_{i+1}. Finally, the anisotropic dexterity index, Θ, which computes the directional dexterity of a robot following a predicted trajectory is computed from these previous indexes.

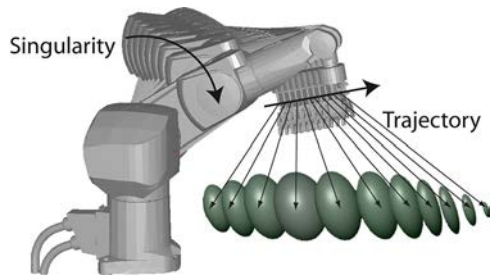


Figure 3.7: Robot executing a linear trajectory and the evolution of the manipulability ellipsoid.

Directional Manipulability index, ψ_u: $p_i \rightarrow p_{i+1}$

The directional manipulability index, ψ_u expresses the manipulability of a robot moving along a specific direction, (3.21), where q is the robot configuration, u is the desired direction and $\dot{x}$ the robot velocity vector. Geometrically, ψ_u represents the distance from the center of the manipulability ellipsoid to the point where u intersects with the ellipsoid surface.

$$\psi_u = \frac{|\dot{x}|}{|\dot{q}|} = \frac{1}{\sqrt{u'(JJ')^{-1}u}} \tag{3.21}$$

ϕ^i_{i+1} evaluation index: $p_i \rightarrow \Omega(p_{i+1}, d_{i+1})$

Next step to obtain the Θ index is to compute the directional manipulability in a range of directions with a common origin: the current robot pose, p_i. This range is the result of the uncertainness associated to every predicted point (a probability distribution function, $\Omega(p_{i+1}, d_{i+1})$).

Let p_i be the current robot pose and $\Omega(p_{i+1}, d_{i+1})$ the next predicted robot pose expressed as a normal probability distribution function, then ϕ^i_{i+1} represents the directional manipulability in all possible movement directions. The range of all possible directions $u_{i,i+1}$ is determined fixing the origin in p_i and, as destination, all those points that intersect with $\Omega(p_{i+1}, d_{i+1})$. A geometric analogy can be established in $\mathbb{R}^3$ where all directions generate a cone with the apex in p_i and a degenerated base with

ellipsoid shape in the Π plane (plane containing the ellipse resulting from the external intersection between the probability distribution functions and the u vectors), Fig.3.8.

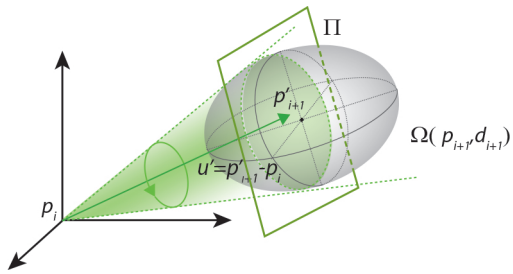


Figure 3.8: Cone generated from the current robot position (illustration reduced to $\mathbb{R}^3$), p_i and the probability distribution function generated around the predicted point $p_{i+1} : \Omega(p_{i+1}, d_{i+1})$

In order to weight every possible movement direction u, each ψ_u is weighted using its probability of occurrence, $P(u)$, (3.22).

$$\phi_{i+1}^{i} = \int \psi_u P(u) du \tag{3.22}$$

The physical analogy of the ϕ_{i+1}^{i} index is the mass generated by all ψ_u inside the robot manipulability ellipsoid in p_i, $M = V\delta$. The density of this mass is function of the associated occurrence probability, $P(u)$, of every ψ_u; which can be interpreted as the line segment inside the distribution probability of p_{i+1}. Fig. 3.9 illustrates the geometrical meaning of the calculus of the index ϕ_{i+1}^{i} in $\mathbb{R}^3$.

To compute ϕ_{i+1}^{i} two different approaches can be used. The first is based on obtaining a numerical closed form solution with low computational cost. The second solution is based on discretizing the space and obtaining a set of sample directions. The development of both solutions are based on decoupled kinematic robots, where the position and orientation can be determined independently, generating two $\mathbb{R}^3$ subspaces and simplifying the computation and the explanation without loosing generality.

The closed form solution uses the physical analogy to compute the mass contained

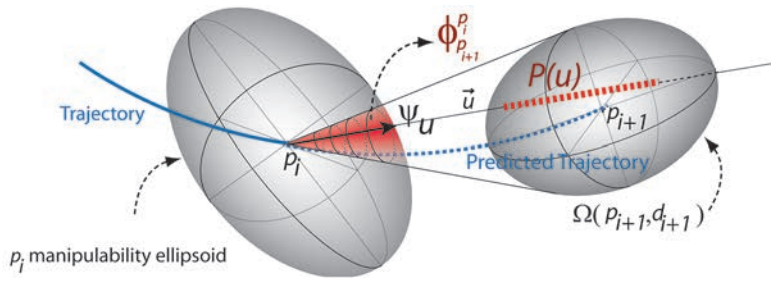


Figure 3.9: Representation of the directional manipulability index, ψ_u, its associated occurrence probability, $P(u)$, the probability distribution function, $\Omega(p_{i+1}, d_{i+1})$, and the anisotropic dexterity index, ϕ^{pi}_{pi+1}, from p_i to p_{i+1}.

inside a volume with variable density. The mass is defined as $M = V\delta$, where the volume, V, is the accumulated length of all the segments with origin in p_i and direction $u = p'_{i+1} - p_i$ where $p'_{i+1} \in \Omega(p_{i+1}, d_{i+1})$ and density δ determined by the success probability of every $P(p'_{i+1})$, (3.23).

$$M = V\delta = \int_{x0}^{x1}\int_{y0}^{y1}\int_{z0}^{z1} \psi((u_x, u_y, u_z))\delta(u_x, u_y, u_z)dzdydx \tag{3.23}$$

Several drawbacks prevent the use of the closed form solution. The directional manipulability is based on the Jacobian which, at the same time, depends on the robot joint values for each robot configuration (pose in the Cartesian space). Consequently, no closed form solution can be found.

The second approach is based on sampling the probability distribution function: for each sample, k, the respective $\psi_u(k)$ and $P(u, k)$ are obtained. Finally, once all K samples are obtained, the average of all ϕ^i_{i+1} is computed, (3.24).

$$\overline{\phi}^i_{i+1} = \frac{\sum_{k=1}^{K}(\psi^i_{i+1}(k)P(p_{i+1,k}))}{k\sum_{k=1}^{K}P(p_{i+1,k})} \tag{3.24}$$

The sampling method does not depend on the probability distribution function. However, two aspects must be pointed out: first, depending on the number of samples, the computational costs can be higher than the closed form solution. Second,

the impossibility of ensuring the inexistence of a singularity inside Ω. This second drawback can be overcome by fixing a threshold to the minimum acceptable singular value, σ_{min}, to each sample. The σ_{min} value decreases when approaching a singularity, becoming zero at singular points.

Φ^i_{i+1} evaluation index: $\Omega(p_i, d_i) \rightarrow \Omega(p_{i+1}, d_{i+1})$

The use of the ϕ index is extended to evaluate robot dexterity when the origin point is also a predicted point, $\Omega(p_i, d_i)$, generating a new index: Φ^i_{i+1}. This index is the result of all the directional manipulability indices, ψ^i_{i+1} mean, within its distribution $\Omega(p_1, d_i)$, (3.25), where $p_i \in \Omega$, is the origin point and Ω is the probability distribution around the predicted point.

$$\Phi^i_{i+1} = \int_{\Omega} \phi^i_{i+1} P(p_k) d\Omega \tag{3.25}$$

Again, to compute Φ^i_{i+1}, two solutions can be applied: the numerical closed form solution and the sampling method. As demonstrated on the ϕ calculus, the Jacobian matrix of the robot prevents the use of the closed form. Consequently, the sampling solution is used. Its mathematical expression,(3.26), depends on M, the total number of sampled points on $\Omega(p_i, d_i)$.

$$\overline{\Phi}^i_{i+1} = \frac{\sum_{k=1}^{M} (\Phi^i_{i+1}(k) P(p_k))}{M \sum_{k=1}^{M} P(p_k)} \tag{3.26}$$

Fig.3.11 shows a set of snapshots of a 6 DoF robot following a vertical trajectory, $p_{i+1} = p_i + \Delta_z$ and their associated manipulability ellipsoids. This trajectory forces the robot to pass close to three different singularities. The first two ($t \approx 1'4s$, Fig.3.11.c and $t \approx 2'5s$, Fig.3.11.d) are produced by the alignment of joints 3 and 5, whereas the last one ($t \approx 3'6s$, Fig.3.11.g) is produced by a boundary singularity. In the same figure, three different Θ indices are shown. Each index corresponds to the robot evaluation as if, from the current pose, the robot moves towards the directions: $u_{i+1} = p_i + \Delta_x$, $u_{i+1} = p_i + \Delta_y$ and $u_{i+1} = p_i + \Delta_z$. The last graphic shows the robot

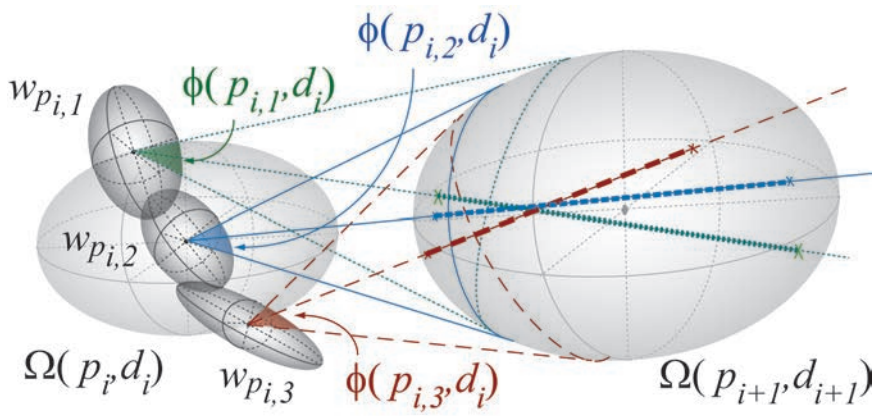


Figure 3.10: Representation of the directional manipulability index, ψ_u in $\mathbb{R}^3$, its associated occurrence probability, $P(u)$, the probability distribution function, $\Omega(p_{i+1}, d_{i+1})$, and the anisotropic dexterity index, ϕ^i_{i+1}, from p_i to p_{i+1}.

joints positions along the trajectory. Observing the evolution of θ_3 and θ_5 the two first singularities can be detected. These singularities generate an abrupt change on the joints positions: the robot changes the arm configuration (no-flip $\rightarrow$ flip $\rightarrow$ no-flip), requiring unachievable joint accelerations.

3.5 Extrinsic Evaluators: Collision Risk

The previous indices evaluate intrinsic robot parameters. In order to obtain a complete estimation of the robots suitability, extrinsic factors like the risk of collision are also required.

The robots that compose MRCP share part of or all their workspace, together with other moving obstacles, generating a dynamic and complex workspace that requires a strict collision detection and an efficient obstacle avoidance control strategy. Such control should not only avoid physical damages caused by a collision, but also minimize disturbances that could affect the teleoperated task, providing an estimation of the risk that a robot collides.

Classical collision avoidance approaches use distance as a unique parameter to

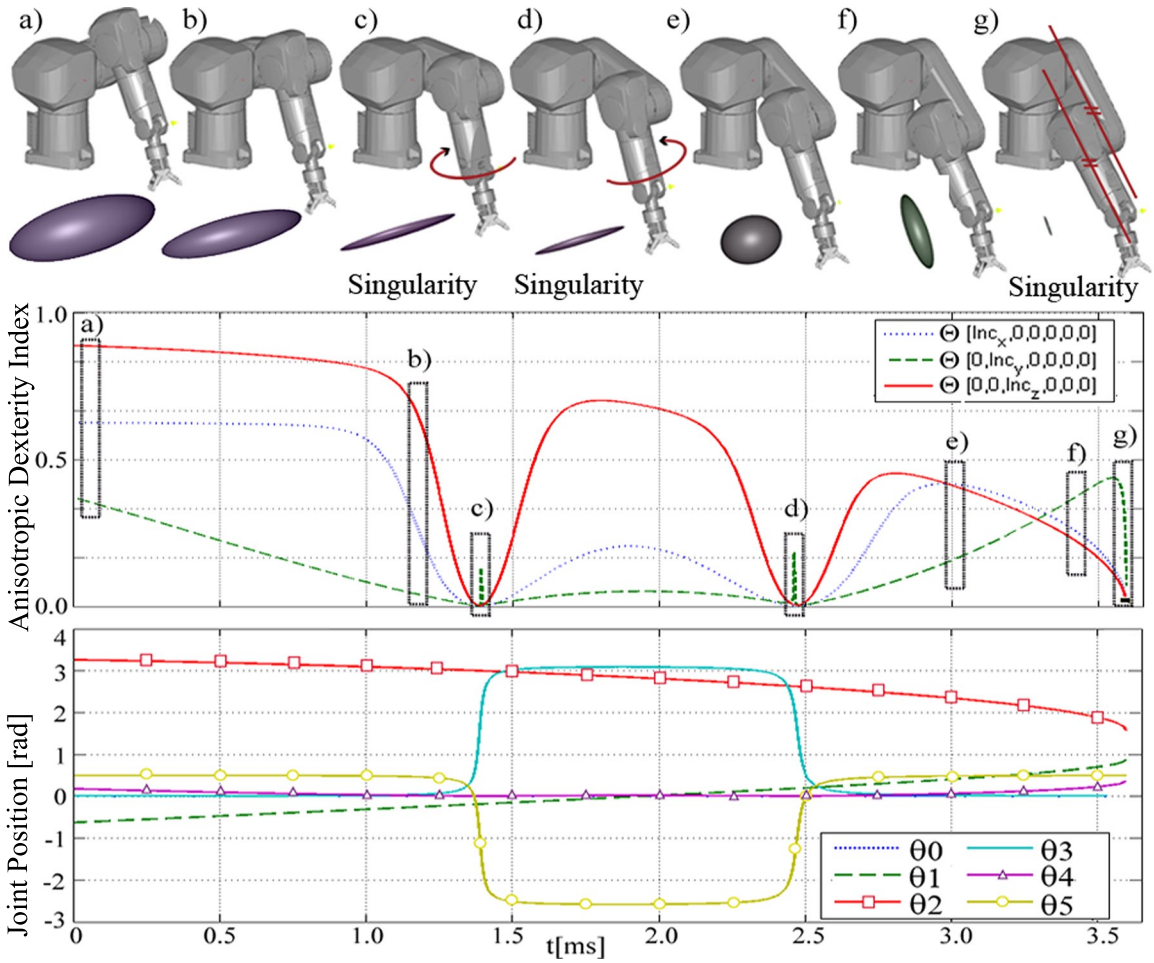


Figure 3.11: Top: Set of snapshots of a robot executing a trajectory and the associated manipulability ellipsoids. Middle: Θ index of the robot in three different trajectory directions. bottom: Evolution of the joint positions.

determine the risk of impact between two objects. The use of uniquely distance measures does not guarantee a realistic collision risk evaluation, the relative direction of the movement between the objects and their estimated impact time must also be included into the collision risk estimation. In [129], the collision risk is established based on the relative velocity between two objects and their maximum achievable acceleration, which determine their reaction capability to avoid a collision. Based on the same principle, here, the Collision Risk index, CR, is based on the estimated

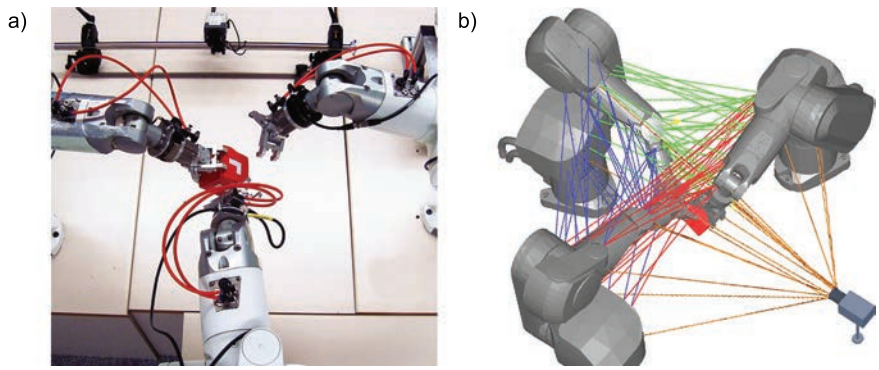


Figure 3.12: Visualization of the minimum distance computed between three robots and an external obstacle.

impact time of the robots when executing a task, as well as on the time required for a task transfer.

Other approaches are based on the Avoidance Manipulability of redundant manipulators, which indicates the ability of a robot to avoid a collision. In [71] two ways of computing the collision risk are analysed: the sum of avoidance matrix singular values and the sum of the Avoidance Manipulability ellipsoid volume. The work concludes that the second method results more efficient than the first one.

The estimated collision time of each robot, $timp'_{Ri}$, is determined obtaining the minimum distance vectors from each robot link, Li, to all the obstacles in its workspace, treating the rest of robots as obstacles. These distances are computed all along the predicted trajectory. Using these distances, all the estimated impact times of the links, $timp'_{Ri,Lj}$, are calculated. Then, $timp'_{Ri}$ is determined using the minimum of all its links estimated impact times, $timp'_{Ri} = min(timp'_{Ri,L1}, ...timp'_{Ri,LN})$ for an N links robot. To smooth the evolution of minimum distance vectors, as well as to reduce the computation complexity, the robot links are recovered with smooth convex hulls. Fig.3.12 illustrates the minimum computed distance between three robots cooperating in a teleoperation application in front of an external obstacle (in this case a camera for the object visual inspection).

Fig.3.13 presents different views of the original 3D models of a Staubli RX60B

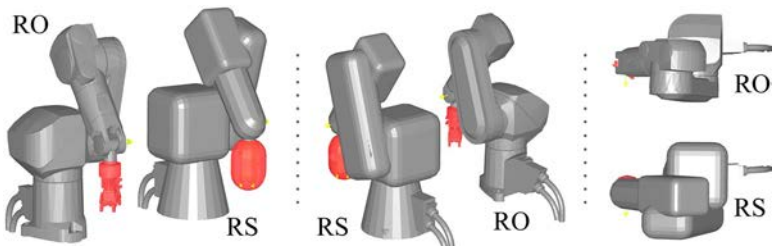


Figure 3.13: Several views of the original 3D model of a Staubli RX60B, RO, and the applied convex hull, RS.

robot, RO, and the convex hull applied to the links, RS. As can be seen the models complexity is drastically reduced, decreasing the computation time.

Finally, CR is determined by the quotient between the remaining time before an estimated collision occurs for each robot, r, and the maximum of these times for all the available robots, (3.27). The minimum distance vectors are computed using the RPQ library, described in A.2.2.

$$CR_r = \frac{timp'_r}{\max(timp'_1, ..., timp'_R)} \ , \ \forall r = 1..R \tag{3.27}$$

3.6 Task Oriented Evaluators

The indices explained above evaluate intrinsic and extrinsic robot aspects. Depending on the task requirements, these indices could not provide a full adequacy estimation of a robot to execute a specific task. In these cases, the design of a task oriented evaluation index provides an optimal solution. Thanks to the modularity of the MRCP control architecture and the common evaluation metrics (transformation into time domain of all evaluation indices), the introduction of new evaluation indices is feasible and do not affect the general control schema.

This section presents diverse examples of these indices. Each example points out the reasons why intrinsic and extrinsic evaluation indices are not enough to estimate the robots adequacy. The first example, a remote teleinspection, must ensure the

visibility of a manipulated object. The second one, a robotic assisted minimally invasive surgery, imposes restrictions on the robot movements. The last example, a co-manipulation of a long object, restricts the force and torque limits of the robots and/or the manipulated objects.

3.6.1 Teleinspection task: Visibility Index

Derived from CR index, the Visibility Index is designed to automatically detect and avoid robot arms occlusions in teleinspection. Let's suppose a remote visual teleinspection of an object in front of a camera. The operator guides the inspected object with a single master device. The MRCP control must compute the robot movements to enable this teleinspection and to decide which robot is the most adequate at every evaluation time. To provide all desired views of the object, several aspects must be observed: intrinsic evaluators prevent the robots to get blocked (due to joint limits or singularities) and those extrinsic prevent collisions between them and with the camera or other workspace obstacles. These evaluation indices enable a smooth teleoperation with automatic task transfers. However, a gap occurs if only these indices are used: occlusions in front of the camera. A new task oriented evaluation index must be introduced to prevent these occlusions: the Visibility Index.

The Visibility Index is based on minimum distance computation. The field of view of the used camera is represented as a solid cone with apex on the camera and the base on the minimum convex hull, in the form of a circle, of the 2D image of the object obtained by the current camera view. While the robots do not intersect with the cone, the complete visibility of the object is ensured.

To avoid occlusions, the robots must transfer the object when necessary. The Visibility Index is used to prevent occlusions and to decide the robot trajectories and the most appropriated grasping poses. Fig. 3.14 illustrates two occlusions and their new grasping poses (providing full visibility). The field of view of the camera is also represented.

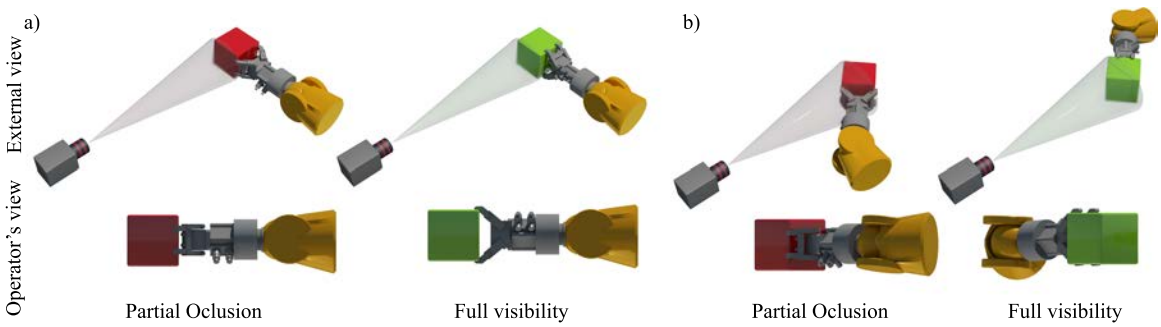


Figure 3.14: Representation of two occlusions and feasible grasping solutions for a full visibility

3.6.2 RMIS task: Tool Orientation Cone

The Tool Orientation Cone virtual constraint is designed for RMIS. It is situated on the entrance point of each laparoscopic tool and its objective is to prevent damages on the abdominal wall of patients. This index defines a maximum orientation cone in each fulcrum point and controls that the robots tools do not overlap its limits. With this limitation, excessive forces and torques are avoided, minimizing hematomas and abdominal wall tears. In manual surgery, the surgeon feels these forces and can minimize damages produced to the patient. In robotics surgery, there is no force sensing that controls this risk.

The valid range of tool orientations is limited by a cone with apex in the fulcrum point. The z-axis cone is defined by the normal vector to the surface in the fulcrum point. The base, perpendicular to the z-axis, is defined by one or more ellipses generating a convex surface. Following the parametric equations, the cone with heigh h oriented along the z-axis and base located at z=0 is defined as, (3.28).

$$\left.\begin{aligned} x &= \frac{h-u}{h} r cos\theta \\ y &= \frac{h-u}{h} r sin\theta \\ z &= u \end{aligned}\right\} \text{ where } u \in [0,k] \text{ and } \theta \in [0, 2\pi) \tag{3.28}$$

Fig3.15 illustrates a visual representation of the protection cone. The cone base is defined by two different ellipses, imposing tool orientation restrictions depending on the azimutal tool orientation, 3.29. The coefficients of both *cos* (0.8 and 0.6) varies the shape of the ellipsoid, restricting the maximum angle of the virtual cone.

$$\left.\begin{aligned} x &= 0.8cos\theta \\ y &= 1.0sin\theta \\ z &= u \end{aligned}\right\} \text{ for } \theta \in [3\pi/4, \pi/2) \text{ and } \frac{h-u}{h} = 1$$
$$\left.\begin{aligned} x &= 0.6cos\theta \\ y &= 1.0sin\theta \\ z &= u \end{aligned}\right\} \text{ for } \theta \in [\pi/2, 3\pi/4) \text{ and } \frac{h-u}{h} = 1 \tag{3.29}$$

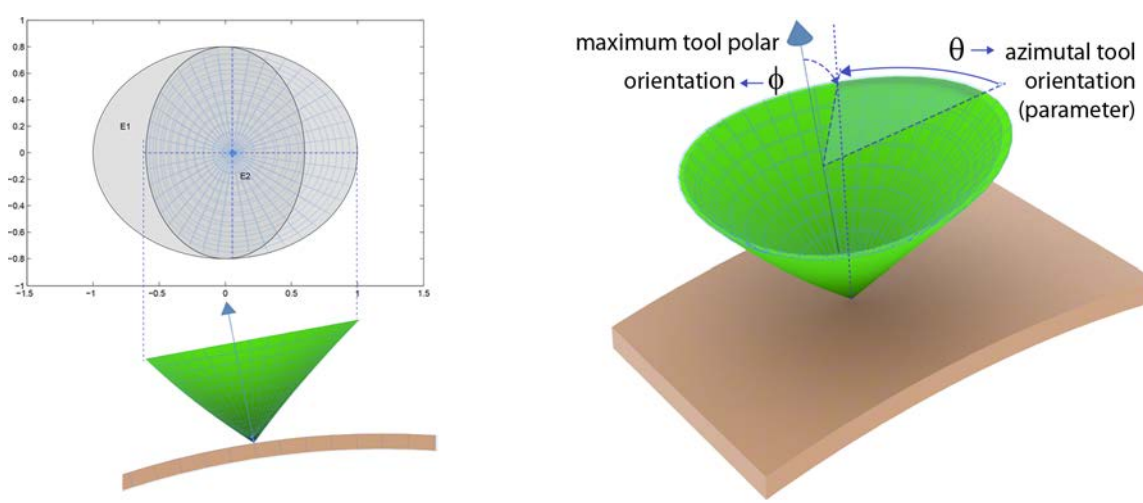


Figure 3.15: Visual representation of protection cone composed of two ellipsoids

To determine if the tool orientation is inside the protection cone, the azimuthal and polar tool orientations with respect the normal vector of the surface (θ, α) must be computed. The θ angle is used to obtain the maximum tool polar orientation with the parametric cone representation. Finally, the polar tool orientation is inside the protection cone when $\alpha \leq \phi$. Fig.3.16 illustrates two views of the tool orientation and

the maximum orientation cone. This index has been published as part of the work presented in [130].

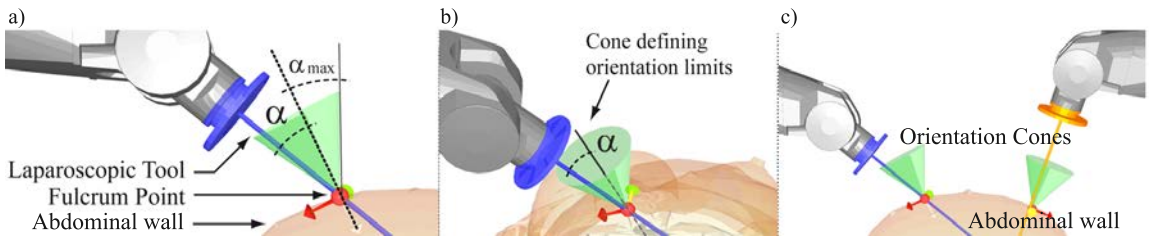


Figure 3.16: a) Lateral view of cone and the angle α formed by the tool and the cone. b) Generic view of the cone and the tool. c) Global view of two laparoscopic views and the associated protection cones

3.6.3 Cooperative manipulation: Force-Torque Index

Telemanipulation of long and heavy objects requires a control of the grasping point to minimize torques on the robots end effector. The grasping point must be close to the center of mass of the manipulated object.

The Force-Torque Index is designed to minimize the amount of forces and torques applied to the end-effector of the robots with respect to a pre-established maximum value. With this index the forces and torques can be minimized by selecting the optimal task point (e.g. grasping position in a telemanipulation task) and the time instant when a task transfer is required to not overcome the maximum allowed torque or force.

For a better understanding, let's use an example based on the co-manipulation of an object (a bar in this example). The robots must hold the bar while an external force is applied on it (e.g. a machining operation on the bar). MRCP is composed of two robots with a force/torque sensor and a grasper as end effector. The bar must always be held up by, at least, one robot. The force sensor of the robot that holds the bar obtains the force and torque (difference between the center of mass of the object without any external force and the displaced center of mass when a external

force is applied). The index detects when a task transfer must be executed to avoid overcoming maximum force and/or torque and the position and orientation of the grasping position is determined by the Force-Torque evaluation index as (3.30).

$$FTeval(\vec{X}) = min\left(\frac{F_{max} - F_{in}}{F_{max}}, \frac{\tau_{max} - \tau in}{\tau_{max}}\right), \vec{X} \in \mathbb{R}^6 \tag{3.30}$$

The Force-Torque index enables to find the optimal instant of time to perform a task transfer ($FTeval \rightarrow 0$) and select the optimal grasping position for the robot that will hold the bar, (3.31).

$$f(F_{in}, \tau_{in}) = \vec{X} \in \mathbb{R}^6 \mid FTeval(\vec{X}) \rightarrow 1 \tag{3.31}$$

Fig.3.17 shows several grasping solutions adopted by MRCP to counteract the excessive force and torque using the Force-Torque index. The task consists on holding a bar with constant density. The applied external force is consequence of drilling over the surface of the bar. The produced force, F, is illustrated as an arrow whereas the measured force and torques are represented as F_{in}, T_{in}.

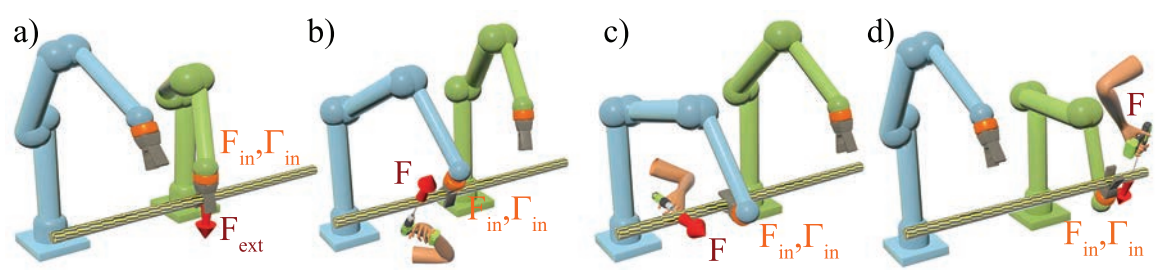


Figure 3.17: Examples of MRCP robots configurations to counteract an external force (An external human operator machining the manipulated object)

Chapter 4

MRCP Control Strategy

4.1 Introduction

This chapter analyses the control architecture and strategies of the Multi-Robot Cooperation Platform. MRCP proposes a new teleoperation paradigm based on a multi-robot cooperative platform with a task-oriented control architecture. Various control aspects are described for a better understanding of the teleoperated system: the change on the teleoperation paradigm, the control architecture and the operator's teleoperation interface.

The MRCP cooperative behavior is based on the complementarity of the slave robots to guarantee a successful teleoperation. MRCP is a teleoperated system with a centralized control system that controls several robots in a cooperative way. This cooperative behaviour is based on the generation of a control strategy with the premise that each robot complements the others. In other words, when a robot is not able to continue the execution of the on-going task it is substituted by another, guaranteeing the satisfactory execution of the task.

The behaviour of the cooperative robots is based on the game theory: all robots are competing to continue with the task execution. As players, they are mutually exclusive. The robot with the highest score is the best candidate (suitability evaluation). In order to obtain the best score in the next evaluation process, the robots that are not executing the task approach the operating point in order to facilitate a

task transfer, when necessary.

Automatic task transfers from one slave robot to another allow the operator to abstract from low level control aspects, understanding MRCP as a single robot with augmented capabilities. Since the human operator cannot easily detect in advance any of the possible causes that can prevent the operative robot from continuing with the teleoperated task execution, MRCP automatically decides, by periodically evaluating the slave robots, if a task transfer between robots is required and which of them is the most suitable to continue with the teleoperated task. From the operator's point of view, MRCP is transparent, allowing the operator to focus the attention on the task itself rather than on the slave robots. The positioning of the robots all along the task, the computation of the slave robots suitability, the selection of the most suitable robot and the task transfers operation are transparent to the operator.

MRCP proposes a change on the teleoperation paradigm, orienting teleoperation to the task. From the operator's point of view, MRCP allows the direct task execution. In standard teleoperation, the operator controls the robot or robots to execute a task on the remote environment. In MRCP, the operator uses the master device to execute the task and, it is the MRCP control who decides the robot actions to allow this task execution. For instance, in a telemanipulation task, the operator, using the MRCP approach, guides the object instead of guiding the robots.

From the architectural point of view, MRCP follows the standard teleoperation architecture; it is composed of a master and a slave station. The master console, a single device to interact with the operator, is used to define the task actions (e.g. remotely guide a manipulated object in telemanipulation). The slave part is composed of several robots that behave in a cooperative way. In robotics, cooperative behaviour is usually understood as the execution of a task by several robots at the same time to, for instance, manipulate a heavy, bulky or deformable object. In the MRCP context, the cooperative behaviour is based on executing the teleoperated task by a unique robot, one at a time and, when necessary, transfer the task execution to the most suitable candidate among the rest of slave robots, thus allowing the satisfactory task completion. Several causes can make a robot not suitable to continue with the

execution of a task (e.g. singularities, workspace limitation, collisions, ...). A review of the possible causes that prevent a robot to continue with the task execution were exposed in 3.6.3.

To achieve this new teleoperation paradigm, the robots selection and their actions should be transparent to the operator. The operator, then, can focus the efforts only on the task, improving, as demonstrated in the experiments carried out, its dexterity, execution time and economy of movement. To achieve transparency, a new control schema has been developed. Its architecture is based on a set of sequential steps including robot suitability evaluation, determination of the most suitable robot, computation of the optimal instant of time to execute a task transfer between robots and determination of the required robot actions to accomplish with the operator orders.

This chapter is organized as follows, first the MRCP teleoperation paradigm is described and referred to the teleoperation context, pointing at the major changes introduced with respect to standard teleoperation. Then, the control architecture and the control algorithm are reviewed. Several examples are presented and analysed at the end of the chapter to clarify and demonstrate the internal behaviour of MRCP.

4.2 MRCP in a teleoperation context

The MRCP proposed teleoperation paradigm is based on the use of multiple slave robots in an automatic cooperative manner. It introduces several changes with respect to standard teleoperation, the control architecture and the teleoperation mode. From the operator's point of view, the MRCP teleoperation paradigm represents a new way of executing a remote task, as well as interacting with the remote environment.

In teleoperation, intelligence does not impose operative barriers since there is a human behind, guiding the robot. However, the lack of propio and remote environment perception, as well as ergonomic limitations on the master device, might decrease the operators capabilities and/or dexterity. MRCP aims to compensate these limitations.

4.2.1 MRCP in Human Supervisory Control

In this section, the MRCP paradigm is classified using the Human Supervisory Control, HSC, criterion. First, the classic human supervisory control is presented.

HSC classifies the human role in teleoperation, defining several control models. The role that the human plays in HSC varies from manual control to fully automated. In manual control the operator executes a teleoperated task without any additional intervention from the control system. In fully automated mode, the operator is a passive observer of the remote process. In between, the supervisory control includes a variable contribution of humans and control systems. When humans contribution decreases, the internal control loop assumes progressively more control decisions.

For a better understanding, let's use a teleoperated mobile robot in a remote environment as an example of HSC. Different types of orders can be defined: drive the robot or define a target point to be reached autonomously. In the first control model, manual control, a human drives the robot and the internal control loop ensures the wheel rotation speed to allow drivability. A higher step of the supervised control is to indicate the desired robot movement direction, releasing the wheel direction to the control system. Increasing the automatic system contribution, the operator indicates a target point on a map and releases to the robot the decisions on the trajectory to be followed to reach the target. Finally, in fully automatic control, the robot is absolutely autonomous and decides its own path following certain pre-established criteria. For instance, in space robotics, where a robot should explore the surface of a planet. Variable communication delays and transition energy consumption leads to use semi autonomous systems. A teleinspection task can visualize the different operational modes. Let's suppose a bi-manual teleoperation system composed of two master devices and two slave robots. When using non supervisory control, the operator guides the robots without intervention of the teleoperation control system. A first level of control supervision can consist in introducing virtual aids to help the operator to obtain the desired views of the object. Increasing the contribution of the automated control, the system can provide the operator with a set of predefined object points of views that the operator can select. Once selected, the system moves

the object to the desired point of view. In a fully automated system, the operator is only a merely passive observant of the inspected object.

Following the work developed in [131], in Fig.4.1 different levels of human supervisory control are presented, to visualize the role of MRCP. The schemas have been modified to be adapted to a multi-robot teleoperation architecture. Dashed lines represent slight control loops whereas solid lines are strong control loops. In the figure, from left to right, the human role decreases. In Fig.4.1.a, the robots are directly guided by the operator, without any teleoperation control intervention. In Fig.4.1.b a minor control loop has been introduced to aid the operator or improve its skills (e.g. cancelling tremor, vary the M-S motion scaling, etc.). In Fig.4.1.c the teleoperation internal control loop releases the operator from the need of expliciting low level orders (e.g. in teleinspection tasks, decide one of the pre-programmed object views). In fully automated systems, the operator assumes a passive role observing the remote workspace, Fig.4.1.d. Finally, Fig.4.1.e shows the MRCP control schema. The MRCP module closes the robots control loops. The task oriented orders are generated by the operator. These orders are translated as robot actions in the control system. Robot positioning and task transfers between robots are computed and planned in the control layer. The operator generates high level orders oriented to the task (e.g. telemanipulated object movements) and the control assumes the low level orders. In [132] a task-oriented semi-autonomous unmanned ground vehicle, UGV, is presented. The robotised vehicle is equipped with different sensors to receive information about the workspace. The UGV executes a predefined task as autonomously as possible. The operator intercedes only when an unexpected situation occurs or when a high level decision is necessary.

4.2.2 Task-Oriented Teleoperation

From the control point of view, MRCP can be classified as a task-oriented teleoperation system. In the task oriented paradigm, the operator's actions are related to the task rather than to the agents (robotic arms in the teleoperation field). Following

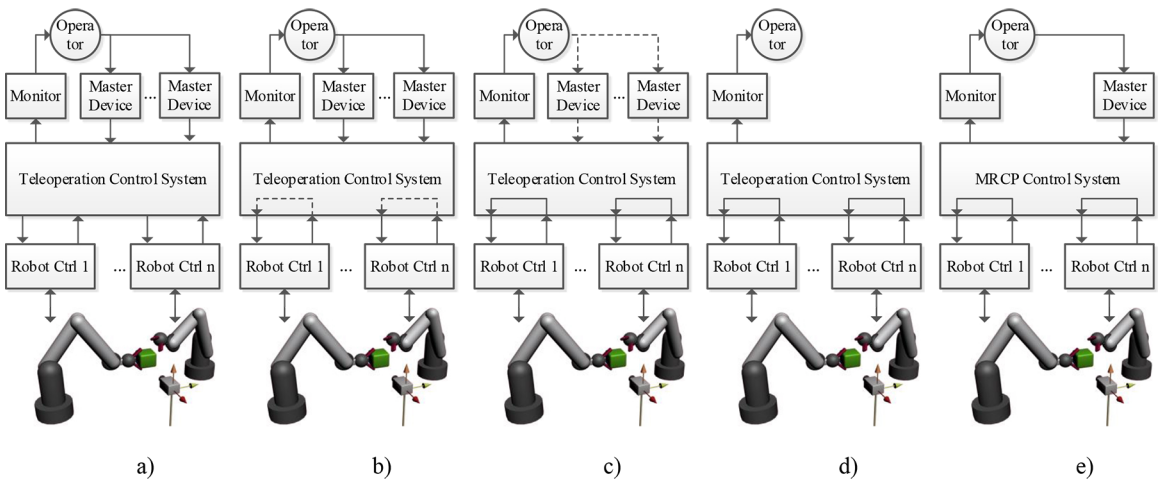


Figure 4.1: Human Supervisory Control schemas. From a) to d), the human control decreases. e) shows the MRCP control system. Dashed lines represent slight control loops whereas solid lines are strong control loops.

this control paradigm, the task is formulated in terms of its requirements (dexterity, payload, task trajectory, etc) rather than on the robot actions (trajectories, grasping, workspace, etc). MRCP provides a teleoperation interface following the task oriented paradigm: operator commands are interpreted as task orders instead of robot actions. E.g. a master movement is interpreted as a movement of the telemanipulated object instead of a robot arm movement.

The task oriented paradigm was introduced in teleoperation systems in works like [42], that propose a task oriented SOMR teleoperated system in which the operator does not drive the slave arms, but uses the master to generate specific task-oriented actions. Several examples are used to describe this approach: a co-manipulation of a heavy object by means of two robots, the coordination of two arms to screw an object, etc. The variables of these tasks are position and orientation, as well as the forces and torques to be applied. The control system then computes the coordinated actions of each robot. As in MRCP, in all tasks the master device is used to describe the manipulated object movements instead of those of the arms.

In standard teleoperation mode, the operator uses the master devices to control the slave robots actions in order to execute a task. In Task-oriented, the operator

executes the task itself: the master orders are directly mapped as task actions rather than robot actions. This higher level orders imply the automatic generation of robot actions, which require an additional layer in the control architecture.

In order to clarify the task-oriented teleoperation concept, an example can be used without loosing generality. Let's suppose a telemanipulation task in which the operator can freely move an object inside a workspace (e.g. a pipette inside an hazardous environment like a chemical laboratory), with a system composed of two master devices and two slave robots. In standard teleoperation, the operator guides the robots by means of the master devices to execute the telemanipulation. The operator is the responsible of the regrasping actions, as well as of robot collision avoidance. In the task-oriented MRCP approach, a single master device is used. The movements of the master device are mapped on the manipulated object and the robots must adapt their configuration to allow the desired operators task orders.

Now, looking at MRCP from the operator's point of view, and due to the automatic robot actions computation when teleoperating with MRCP, the operator feels as executing the task itself. Using this same example, the operator moves the telemanipulated object. To illustrate it, two schemas using two slave robots are described. Fig.4.2 illustrates a standard teleoperation schema, whereas Fig.4.3 illustrates the proposed MRCP approach. In standard teleoperation, the operator interacts with two master devices generating robot movement orders ($\vec{X}_{Op1}, \vec{X}_{Op2}$). The Teleoperation Control System generates the orders to the robot controllers ($\vec{X}_{R1}, \vec{X}_{R2}$) which are reflected as movements on the Tool Center Point, TCP, of the two slave robots ($\vec{X}'_{R1}, \vec{X}'_{R2}$).

In [133] a task oriented teleoperation system designed for micro and nano tasks is presented. Three simple tasks are defined: pick object, place object and move to. These tasks are automatically executed at the operator's indication. The operator assumes a supervisory role and generates high level orders (task-oriented commands). The task is executed once the target point is selected on the remote workspace view and the type of task is chosen under the operator's indication. In [134] authors propose a 3D user interface to command a teleoperation system using natural hand gesture

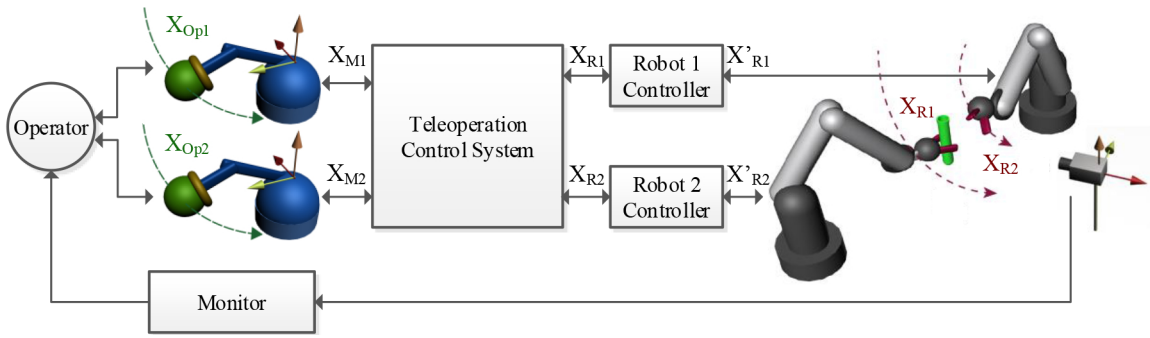


Figure 4.2: Schema of a standard bi-manual teleinspection system.

movements to ease the interaction with the remote environment.

In the proposed MRCP schema, the operator interacts with a single master device defining a movement ($\vec{X}_{Op}$) that is applied to the manipulated object ($\vec{X}_{Obj}$). The computed robot movements to achieve the desired objects' trajectory ($\vec{X}_{R1}, \vec{X}_{R2}$) and ($\vec{X}'_{R1}, \vec{X}'_{R2}$) are transparent to the operator. The visualized robots transparency on the figure reflects the idea of operator's abstraction with respect to the robots that make the task execution possible.

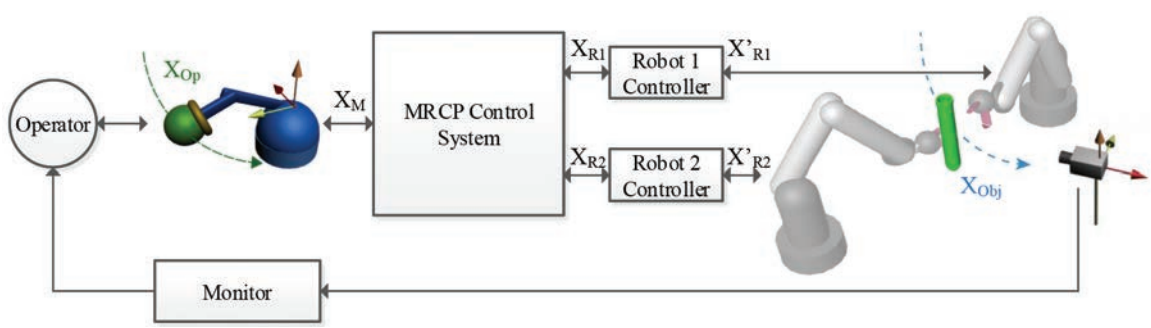


Figure 4.3: Schema of MRCP for a teleinspection tasks.

4.2.3 MRCP in Teleoperation Taxonomy

In this section, MRCP is classified following the teleoperation taxonomy based on the number of human operators and slave robots present in the system. This taxonomy, proposed in [135], classifies the systems in four groups, as reflected in Table 4.2.3. As later developed, MRCP can be classified inside this taxonomy into two different

groups depending on the point of view used: from the control point of view or from the operators' point of view.

Acronym	**Operators**	**Slave Robots**
SOSR	Single	Single
SOMR	Single	Multiple
MOSR	Multiple	Single
MOMR	Multiple	Multiple

Table 4.1: Teleoperation taxonomy classification based on number of Operators and Slaves involved in the teleoperation system

Single Operator, Single Robot

Single Operator, Single Robot teleoperation system configuration is composed of a unique master console, usually a single master device, and a single slave robot, 1M:1S. Fig.4.4 presents the block schema of SOSR in the context of RMIS.

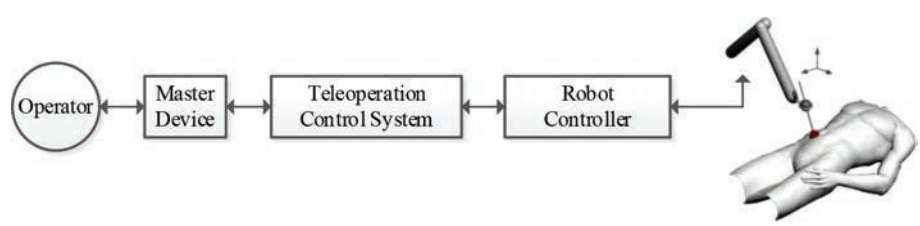


Figure 4.4: Single Operator Single Robot (SOSR) schema

The SOSR configuration is the simplest teleoperation architecture and is used in a wide range of teleoperation systems.

Single Operator, Multiple Robots

Single Operator, Multiple Robot, taxonomy is composed of a master console teleoperating two or more slave robots. The remote workspace is partially or totally shared

by the slave robots. The master console is equipped with one or more master devices to control the slave robots. The teleoperation system controls the matching between the master devices and the slave robots (which master device drives each slave robot). The number of master devices and slave robots depends on each system, nM:mS. When $n = m$, the M:S assignation is usually fixed: each master device controls always the same slave robot. On the contrary, when $n < m$ the M:S assignation can be done explicitly by the operator or internally by the teleoperation control system. The first solution is the most frequently used by SOMR systems. Fig.4.5 shows the block schema of SOMR in the context of RMIS.

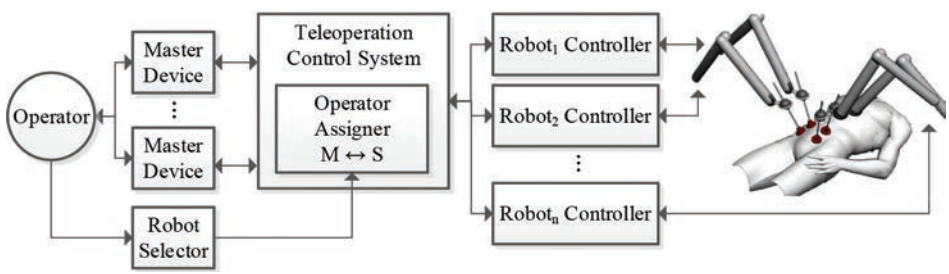


Figure 4.5: Single Operator Multiple Robot (SOMR) schema

Most of the Robotic Minimally Invasive Surgery, RMIS, systems are based on a SOMR schema. Systems like Zeus [136, 137], DaVinci [138], DLR MiroSurge [139] or Bitrack provide the surgeon with two master devices to control three or four slave robots. The operator explicitly selects the M:S mapping from the master console by means of declutching each master from the previous assigned slave, selecting the new slave and clutching the master again.

Multiple Operator, Single Robot

Multiple Operator Single Robot, MOSR system configuration is composed of a several master consoles teleoperating a single slave robot. One of the most common teleoperation problem is the communication time delay between master and slave. When the system is composed of several master devices, they can present different delays. The system requires from master synchronization to compensate the variable time delays

due to their potential different locations. Fig.4.6 shows the block schema of MOSR in the context of RMIS.

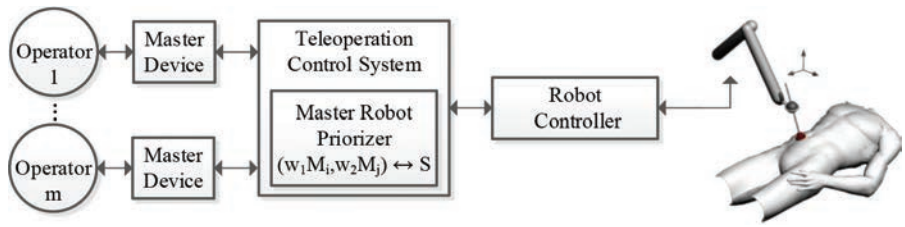


Figure 4.6: Multiple Operator Single Robot (MOSR) schema

Most MOSR systems are used as student-teacher scenarios, where different types and levels of haptic interaction between the masters is introduced. The interaction between the operators is the main research topic inside MOSR. In [140], a system based on a variation of the standard MOSR, and composed of a teacher and a student master consoles is proposed. The student executes a task on a simulated environment while the teacher supervises these actions. Four modes of interaction between teacher and student masters is established: independent, tele-mentoring (unilateral feedback from teacher to student), tele-evaluation (unilateral feedback from student to teacher) and bilateral tele-mentoring. [141] proposes a MOSR architecture in which the humans behind the master consoles have the same teleoperation skills and the study is focused on the interaction between them: visual or visual and haptic coupling.

Multiple Operator, Multiple Robot

Multiple Operator Multiple Robot, MOMR, system configuration is composed of several master consoles teleoperating a set of multiple slave robots at the same time. Examples of MOMR architecture can be found in the da Vinci Si Dual Console RMIS system with a $2M : 4S$ architecture. The multiple master consoles are used for both training and assisting in surgery (establishing the primary and auxiliary roles to each operator). In learning, the *see and repeat* learning paradigm is used to train inexperienced surgeons. During intervention, the roles of each operator are established.

The primary surgeon executes the surgery, while the auxiliary operator assists on secondary tasks (tissue grasping, organs holding...).

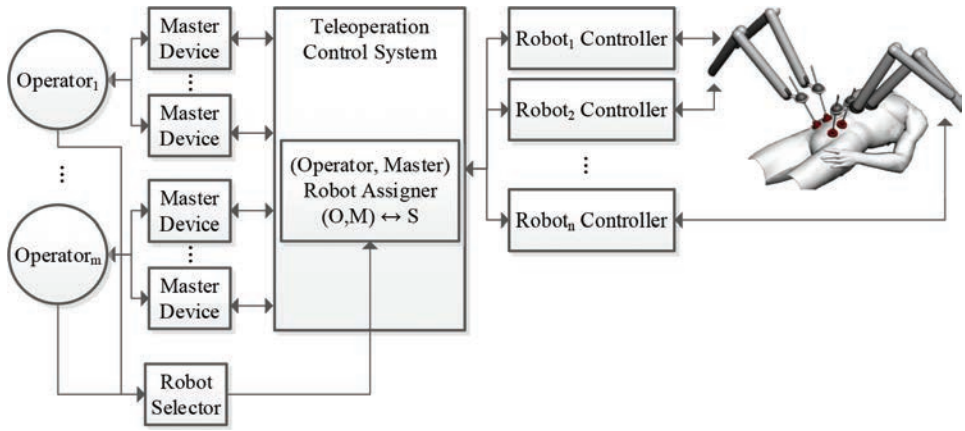


Figure 4.7: Multiple Operator Multiple Robot (MOMR) schema

MRCP Taxonomy

MRCP can be classified following the taxonomy criteria. MRCP introduces several modifications with respect to the standard teleoperation paradigms previously described. In the following, MRCP is classified from the control and the operators points of view. These two approximations classify MRCP in two different taxonomies: as SOMR from the control point of view and as SOSR from the operators point of view.

MRCP from the architecture point of view

MRCP is constituted by a single operator interacting with a single master device and a remote part with multiple slave robots. A centralized control system continuously obtains the inputs from the master devices, the robots and the workspace sensors. The periodic robot evaluation process generates the orders for the robot controllers. MRCP follows the mentioned teleoperation taxonomy: SOMR system. Fig.4.8 illustrates the block schema of MRCP from the architecture point of view. This schema is composed of a single operator managing a single master device in the local part. On the multi-

robot slave part of the system, several robots share the workspace and the task. The control system generates the required orders to enable the system to execute the teleoperated task in a cooperative manner.

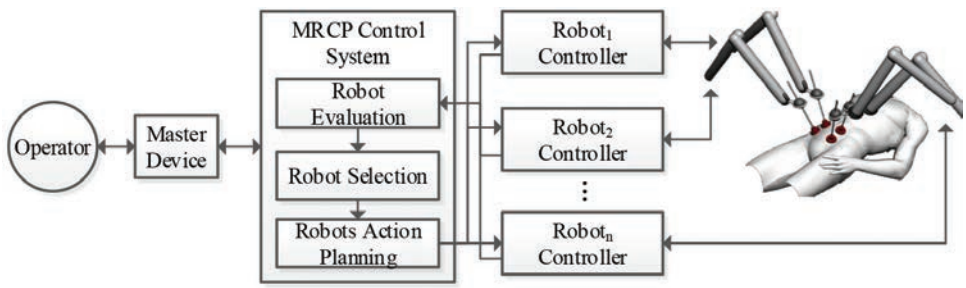


Figure 4.8: MRCP schema from the architecture point of view. Task-oriented telemanipulation.

From the architecture point of view, MRCP differs from classic SOMR systems in several aspects. In SOMR, the master console is composed of one or two master devices (single or bi-manual teleoperation: 1M:nS or 1M:nS + 1M:nS systems respectively), whereas in MRCP a single master device is used to teleoperate n slave robots. In SOMR the operator explicitly selects the matching between master device and slave robots while in MRCP the control system automatically decides which robot executes the task at every evaluation time to execute the operator commands. Consequently, the mapping between M:S is transparent to the operator.

MRCP from the operators point of view

MRCP has been designed to enable a change on the teleoperation paradigm from the operator's point of view. The operator teleoperates at a higher abstraction level than in standard teleoperation, executing the task itself instead of commanding the robots. The control layer that allows this teleoperation paradigm is transparent to the user. This transparency classifies MRCP closer to a SOSR instead of a SOMR when seen from the operator's point of view. The operator abstracts from which robot executes the task, obtaining the sensation of interacting with a single robot with augmented

capabilities.

Fig.4.9 shows the MRCP taxonomy, from the operator's point of view. The operator, interacts with a unique master device executing the task. In this example, guiding a needle inside the abdominal region by means of a multi-robot slave station. The MRCP control system manages the robots to allow the smooth execution of this guidance task, selecting the most adequate robot at every evaluation time, transferring the task (needle in the example) from one robot to another and positioning all robots to their best position. The automation of these tasks induces the operator to feel as if controlling a single robot, even more, abstracting from the robot and executing the task with the master device. In the presented example, the operator movements are directly mapped into the manipulated needle.

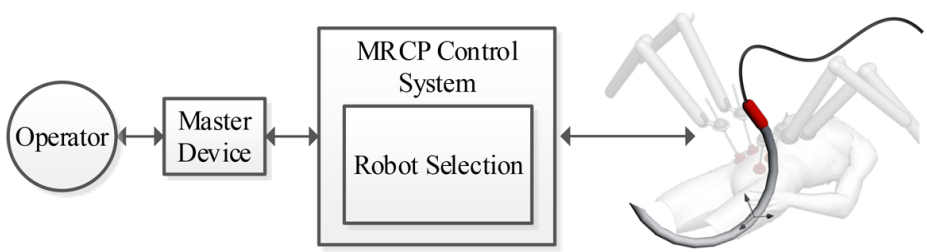


Figure 4.9: MRCP schema from the operators' point of view

In Fig.4.8 the workspace is composed of several robots and the task, represented by a needle, is pseudo-transparent. On the contrary, in Fig.4.9 the task (needle) plays a central role in the remote workspace, while the robots are represented pseudo-transparent. This representation illustrates the perception of the operator when teleoperating in standard mode (the robots play a central role) or when teleoperating with MRCP (the task is the goal of the operator).

4.2.4 MRCP Centralized Control Schema

One of the first aspects to establish when designing a teleoperation control system is its architecture. The classical control schemas in the literature are: centralized and distributed. In the centralized approach, all decisions and actions to be executed

are computed in a single unit: the central control unit, CCU. In the distributed control, the robots decide the further actions to be executed by themselves. Following the standard distributed control nomenclature, the slave robots act as autonomous agents. A typical example of application of this control approach are the formations of mobile robots. Several examples can be found in the literature, [142–144]. Between this two schemas, there is a wide range of hybrid control schemas that incrementally release part of the control computation and decisions from the CCU to the agents. In the context of sensors and actuators networks distributed on a workspace, [145] proposes an hybrid solution. The CCU generates control decisions based on global information obtained by the sensors. The introduction of control decision capabilities to the actuators (generating a distributed control), allows the actuators to execute local control orders.

Several aspects have been considered when deciding the control approach of MRCP. The most relevant are: the interaction between the different robots, the information required to determine the new actions to be executed and, finally, aspects related with communication: the amount of data (and the associated delays) to be transmitted between robots on the remote workspace and the control unit.

The considered optimal solution for the MRCP control is a strict centralized approach. Two aspects have determined the selected solution: first, the need of global information to determine the role of each robot and the consequent actions that every robot must execute and, second, the high interaction between the robots executing a cooperative teleoperation in a shared workspace. MRCP must decide which is the most suitable robot to continue with the task execution at every evaluation process by means of comparing the evolution of the suitability of each robot along time. This comparison requires a centralized computation: the necessary information to obtain the suitability of each robot does not only depend on the robot itself, but also on external sources like the workspace sensors and the configuration of the rest of robots. This information is transversal for all robots and, in a decentralized control schema, it must be distributed to all agents, increasing the amount of transferred data and, consequently, decreasing efficiency.

Fig.4.10 shows the inputs of the MRCP control: master device, robot controllers and workspace sensors.

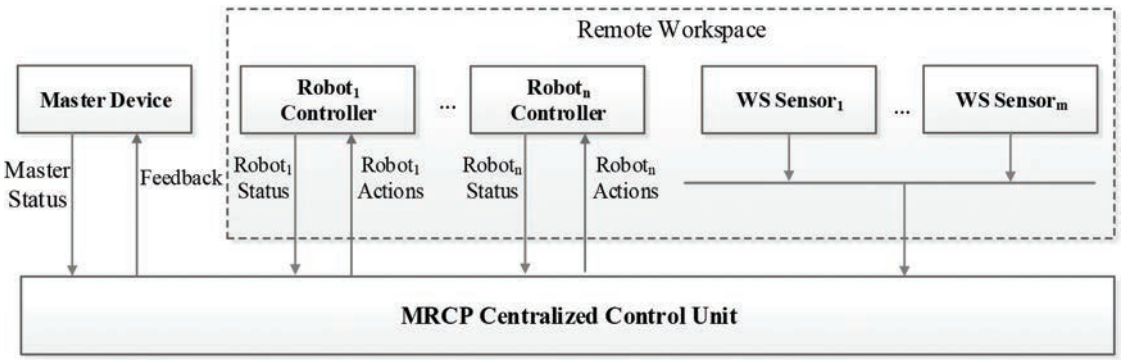


Figure 4.10: Information flow from the master, robot controllers and workspace sensors to the MRCP centralized control

4.3 MRCP Control Architecture

The control architecture must enable the successful execution of the MRCP teleoperation paradigm. To reach this objective, the control system evaluates the robots following the suitability criteria, estimates the most adequate instant of time to execute a task transfer and, finally, computes the set of robot actions to be executed until the next evaluation process.

The MRCP control system must operate in real time, with transparency and minimizing the number of task transfers. Concerning real time requirements, the presence of time delays decreases the performance of teleoperated systems, thus requiring higher operator efforts or predictive control systems that deal with delays. The MRCP control is optimized to execute a complete control cycle (robot evaluation, task transfer decision and robot actions) in less than 18ms, which corresponds to the robot controllers communication cycle in the current slave station setup. The internal control must be transparent to the operator; that is, the selection of the robot to execute the task or determining the instant when a task transfer is required, depends on internal

control aspects that do not involve the operator; the operator only generates task orders. Finally, it is necessary to minimize the negative effects of task transfers. A task transfer process represents the main perturbation to a smooth task execution, interrupting the regular teleoperation and releasing the control to the MRCP control unit. MRCP minimizes the number of task transfers using the *Need of Task Transfer* index, which activates a task transfer only when strictly required. A second aspect to be optimized is the time required to execute the transfer. MRCP updates at every evaluation step the position of all the robots to ease the transfer.

4.3.1 General description

MRCP is designed as a high level closed-loop system that, using the master commands as high level orders in the context of task-oriented teleoperation, generates the actions to be executed by the robot controllers. The robot controllers close the low level control loop, ensuring the correct execution of the MRCP output commands.

In Fig.4.11, MRCP is described by a control block diagram. In this schema, the MRCP control unit has three different inputs:

1. $\Delta\vec{X}'_{Task}$: Defines the next task status, in terms of task oriented paradigm (e.g: position of a manipulated object).

2. $\vec{X}_{R1}, .., \vec{X}_{Rn}$: Information about the robots provided by their controllers (robot position, end effector status, ...).

3. $\vec{X}_{Sensors}$: Information of the workspace provided by external sensors (e.g: obstacles position determined by a vision system)

The signals generated by the control unit close the control loop of the master device and the slave robots are:

1. Master console: Set of signals that provide the operator with augmented reality. Two main signals are generated: a) synthetic visual information that, jointly with the real visual information, is used by the visualization module to generate

the augmented workspace vision and b) haptic feedback. This haptic feedback can be of different nature and uses different sensitive human channels (audio feedback, force feedback, ...). Apart from these signals, the control unit has the capability of generating virtual fixtures to improve the operator's performance. This feature is not present in the schema.

2. Slave Robots: Set of signals for the robot controllers, $\Delta\vec{X}'_{R1}, .., \Delta\vec{X}'_{Rn}$. These orders are the result of computing the robot actions required to continue with the execution of the on-going task and the positions of the robots that are potential candidates to continue with its execution in a future robot evaluation process. The low level robot control loop is closed by the robot controllers.

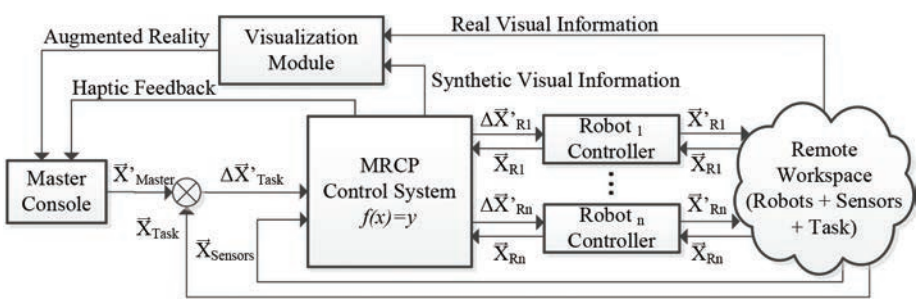


Figure 4.11: Block schema of MRCP seen as a plant control

For a formal description of MRCP, let's define MRCP as a set of n_r robots telemanipulating an object with n_q predefined grasping points. Then, MRCP can be described as a triplet MRCP($r = 1..n_r, q = 1..n_q, m = 1..n_m$) where r represents the set of slave robots, q the number of continuity points and m the evaluation indices. Before explaining the proposed MRCP operating methodology, some definitions and their acronyms are listed:

- **Robot Selected,** *RSel*: robot that, in the evaluation time, is executing the ongoing task.
- **Robots Candidates,** *RCan*: candidate robots to continue with the execution of the ongoing task.

- **Continuity Robot,** $RCnt$: best robot to continue with the ongoing task.
- **Current task point,** $CuTP$: current point of $RSel$.
- **Continuity task points,** $CnTP$: set of points where the $RCan$ are evaluated.
- **Task Points,** TP: Set of points that verify $TP = CuTP \cup CnTP$

Depending on the task, $CuTP$ and $CnTP$ represent the same workspace position, e.g. in probing or soldering tasks; or different, e.g. in manipulation tasks, in which the manipulated object can be grasped from different positions, Fig.4.12. The control algorithm is continuously evaluating the slave station as follows: first, compute the suitability of $RSel$ in $CuTP$ and all $RCan$ in every $CnTP$. Once $\{RCan, CuTP\}$ is obtained, the need of a task transfer is computed using the Need of Task Transfer index, NTT. Finally, the robot actions computation phase is executed.

Figure 4.12: Illustrative example of telemanipulation of a cube with 6 pre-defined grasping positions

4.3.2 MRCP Control Architecture Requisites

MRCP has been conceived as an open platform to enable teleoperated tasks of different nature. Consequently, the control algorithm must deal with different type and number of inputs and outputs (e.g. type of master devices, type and number of slave robots and workspace sensors). The open platform, as detailed subsequently, imposes a set of

requirements to be accomplished. The design of the MRCP control unit is organized in multiple layers, abstracting the control algorithm from the real set-up implementation issues.

The main requisites considered to design the control unit are:

1. Independence from different type of master devices.
2. Independence from the type of sensors that register the remote workspace status.
3. Be adaptable to deal with different slave configurations in terms of number and distribution of robots in remote workspace.
4. Be able to communicate with different robot controllers in terms of communication protocol, positioning commands, etc.
5. Be able to evaluate different robot kinematics (serial kinematic chains).
6. Computational efficiency to allow real time control as required in teleoperation, introducing a minimum time delay.
7. Introduce new evaluation indices without altering the control schema.

To deal with requisites 1 and 2, the MRCP control architecture includes a high level layer that acts as a communication link between the different devices (master devices, workspace sensors,...) and the control algorithm. This layer abstracts the control from the communication protocols of each master device and sensors. Thus enabling the use of the type of master device and sensors that best fits each task and workspace. In the presented experiments, two types of master devices have been used: a 6 DoF mouse and a Phantom Omni haptic device. The 6 DoF mouse, which generates incremental movements, results in an appropriate master device for tasks like teleinspection and telemanipulation with free movements and/or long trajectories that do not require high precision. On the contrary, the Phantom Omni, which allows haptic feedback, is used in telemanipulation tasks with reduced workspace and high dexterity requirements.

Requisites 3, 4 and 5, which refer to the slave station, reinforce the open platform paradigm: adapt the master and slave configuration to fit with the task requirements. The number, distribution in the workspace and the kinematic configuration (number of DoF, joint configuration and disposition) of the robots that compose the slave station is not fixed and is determined by each task requirements. This implies (requisites 3 and 4) that MRCP should be able to interact with different number of robots, communication protocols and robot movement methods (by position, velocity or acceleration). This requirement demands the inclusion of a high level layer to adapt MRCP to each robot communication protocol without altering the control algorithm. Requisite 5 refers to the need of evaluating different serial kinematic chains. The robot evaluation indices must be able to evaluate and compare serial kinematic chains with different configurations. This requirement is fulfilled using generic evaluation indices that can be adapted to each robot, and comparing the evaluation indices in the time space rather than in the evaluation space.

Following the open platform paradigm, the MRCP implementation has been designed as a multi-layered control. This approach, as requisites 1-5 demand, abstracts the control algorithm from the real set-up requirements. The external layers are used to translate and adapt the received information (that depends on the number and type of the used devices, sensors and robot controllers) to a standard data format required by the control algorithm. The same procedure is done to translate the generated control algorithm data (new robot actions and master feedback) to the devices and robots specific communication protocol. Fig.4.13 shows the implementation of the MRCP control kernel and the interaction between the different layers. The control algorithm, formed basically by the robot evaluation phase, followed by the robot selection, the task transfer decision process and finally the robot actions computation, interacts with the external layers without altering its composition. Three layers conform the external ring that act as drivers communicating the control algorithm with the environment (master console and workspace). In clock wise order first, the bidirectional Robots-Control communication layer, which sends the position of the robots to the Robot Evaluation control step and receives the new robot actions to be executed from

the Robot Actions Computation. Second, the Workspace Sensors-Control communication layer is an unidirectional communication layer that collects the different sensor data describing the workspace status. This information is send to the Robot Evaluation block. Finally, the bidirectional Master Console communication layer is used to receive the new teleoperation commands and generate, if required, haptic feedback. The feedback generation, as reflected in the schema, is not considered as part of the control algorithm itself: the feedback generation depends on both, the master device and the task. The control algorithm, as described later in detail, starts receiving the information from the three external layers (robot position, workspace status and new teleoperation commands from the master console), computes the new robot actions and the haptic feedback and sends the information to the corresponding layer.

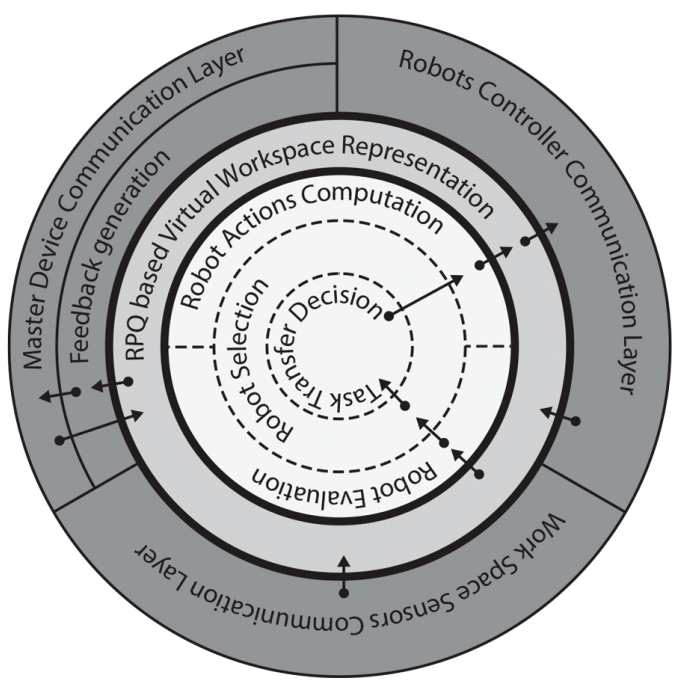


Figure 4.13: MRCP control layered schema

Requisite 6 responds to the demand of real time interaction between master and slave. In teleoperation, the time delay must be cancelled or, at least, minimized and then, compensated. In MRCP, the control algorithm should be as optimized as

possible to avoid delays in the control loop, while continuously evaluates the robots and decides their actions in real time. Therefore, the evaluation indices have been designed to be low computational cost.

Finally, requisite 7 reflects the fact that MRCP must be valid for a wide range of tasks. Some of them require specific evaluation indices (e.g. orientation of the surgical tool with respect to the fulcrum point in RMIS surgery). The modularity of the control algorithm eases the introduction of new indices without altering the control schema. To overcome this requirement, every evaluation index is independent from the rest and only depends on its own (shared or not) inputs. The evaluations of the indices are transferred to time space to be directly comparable in a common metrics.

4.3.3 Control modules and operational modes

MRCP is composed of a set of independent control modules, named Robot Suitability Evaluation, Need of Task Transfer decision and Robots Actions Planner. Robot Suitability Evaluation computes the adequacy of each robot to continue with the task. It is expressed in terms of the estimated useful time until the robot is not valid to continue with the task going on. This module has as inputs the operator commands and the slave station information (robots and obstacles positions). Need of Task Transfer determines if an immediate task transfer is required and generates a list with the robots ordered under the above suitability criteria. Finally, the Robots Actions Planner decides the actions to be executed by the robots until the next evaluation process. Fig.4.14 shows the interaction between control modules, including the master and slave stations among other auxiliary software modules (augmented reality and workspace simulator).

MRPC is designed as a sequential multi-step control with two operational modes defining the behaviour of the system: task-oriented teleoperation, TOp, and task transfer mode, TT. TOp enables the regular teleoperation control sequence including the master inputs. In TT mode, the control of the arms is completely released to the

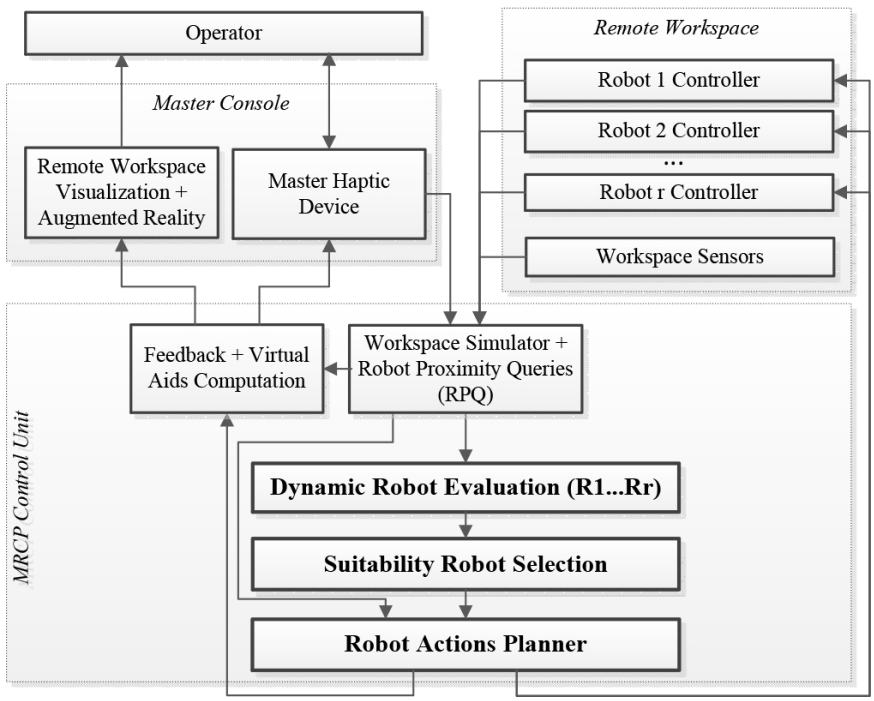


Figure 4.14: MRCP control architecture. The schema includes the master console and the remote workspace among other auxiliary modules.

control system that automatically executes a task transfer. Once the task transfer is completed, MRCP returns to *TOp* mode. Both modes generate different paths inside the control modules. Fig.4.15 shows the MRCP state transition graph. In *TOp* mode, the control is organized in three sequential steps according to the mentioned modules: Robot Suitability Evaluation, Need of Task Transfer determination and Robots Actions Computation. On the contrary, in *TT* mode, the control involves only the Robot Actions Computation module, which interacts with the slave system until the task transfer is finished. In Table 4.2, the states composing the MRCP control are reviewed, including the operational mode, the inputs, outputs and the conditions of a transition between modules.

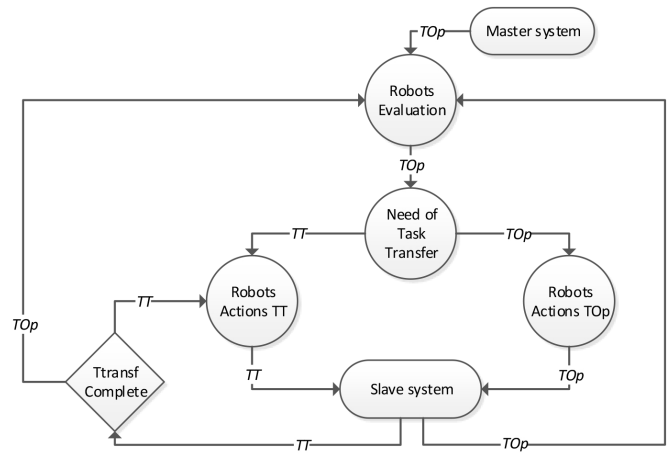


Figure 4.15: Graph of states of MRCP. *TOp* transitions in mode Teleoperation and *TT* transitions in mode Task Transfer

4.4 MRCP Control Algorithm

In this section the different control modules composing the MRCP algorithm: robot suitability evaluation, need of task transfer and robots actions computation are reviewed remarking the most important features of each one.

4.4.1 Robot Suitability Evaluation

This module, the first step of the MRCP control process, computes the estimated robots suitability to continue with the ongoing task. Robot Suitability uses the master device orders and the slave station information (robots positions and sensor lectures) as inputs. As output, this module provides the estimation of the remaining useful time until a robot is not suitable to continue with the ongoing task. Concerning the internal organization, this module can be divided in, first, the robots evaluation using different evaluation indices and, second, the estimation of the robots useful time, resulting from the evolution of the evaluation indices along time and the estimated time to complete a task transfer, if required.

Evaluation indices

The suitability evaluation indices have been defined considering the different causes that prevent a robot from continuing with a task. Three generic types of evaluation indices have been identified: Intrinsic, Extrinsic and Task dependant indices.

Intrinsic evaluators measure intrinsic robot parameters like kinematic limitations, singularities, joint limits, etc., that define its capability to execute the task. Examples of intrinsic evaluators are the Robot Joint Limits index that measures the proximity of each joint with respect to their limits, or Jacobian based indices like Directional Manipulability index which measures the manipulability (capability of a robot to generate a movement given a configuration) in a specific direction. Extrinsic evaluators measure the interaction between the robots and other objects in their workspace, detecting potential collisions. Finally, task dependant indices are specifically designed to achieve the task requirements. Examples of task dependant indices are imposed robot constraints, like kinematic or dynamic limitations (e.g. reduced workspace volumes, forbidden regions or maximum allowed tool velocities). In a previous work [130], some task oriented evaluators were introduced and used in the MRCP platform. One of them is the tool orientation index applied to RMIS (Robotic Minimally Invasive Surgery), designed to avoid damages on the patients abdominal wall.

To obtain the complete evaluation of the robots, $RSel$ is evaluated in its $CnTP$, whereas the rest of $RCan$ are evaluated in all $CnTP$. This process generates a list of $(n_r - 1) * (n_m) * (n_q) + (n_m)$ evaluation indices, defined as $I^r_{m|_q}(t)$ where, as defined previously, r is the robot, m the evaluation index and q the evaluated point. Fig.4.16 shows an illustrative example of an operation using a MRCP with $n_r = 2$ and $n_q = 3$. In Fig.4.16a MRCP is working in Teleoperation mode and $RSel = R1$, grasping the object in $CuTP$ whereas $R2$ is evaluated in $CnTP_1$ and in $CnTP_2$. In Fig.4.16b MRCP is executing a task transfer (TT mode). $R2$, now $R2 = RCnt$, will continue the task in the best evaluated $CnTP$. Finally, in Fig.4.16c MRCP is operating again in TOp mode with $RSel = R2$ and $R1$ is evaluated in $CnTP_2$ and $CnTP_3$. In what follows different internal control variables are described.

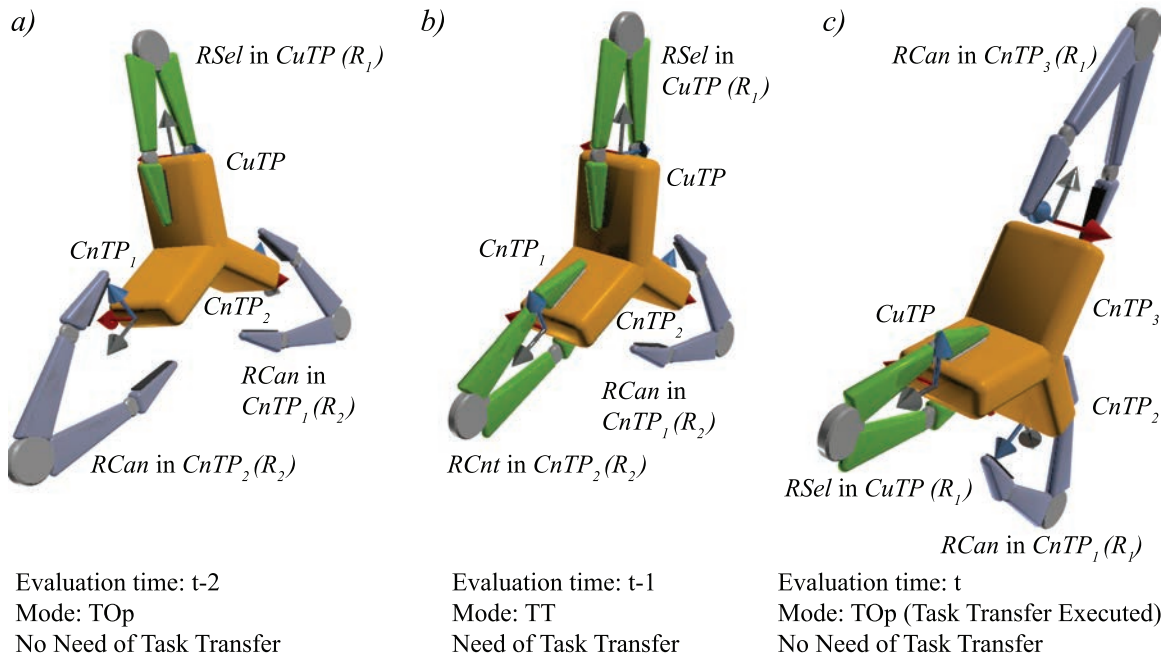


Figure 4.16: Illustrative sequence of MRCP teleoperation.

Useful time: tus

The variable number and different nature of the evaluation indices generate a non-compatible metrics that prevent their direct comparison. Even more, some indices are robot order and link length dependant, resulting valid only when the kinematics of the robots being compared are identical. The disparity of the metrics problem imposes the need of defining a common evaluation space to obtain the measure of the suitability of each robot using diverse indices. The time space is used to transform all indices in a common domain.

The useful time of each robot, $tus^r(t)$, is used to decide how appropriate a robot is to become the next $RCnt$. Two different factors define the $tus^r(t)$ of each robot: its suitability to continue with the ongoing task (evaluation indices transformed into time space, $ttransf^r_{m|_q^*}(t)$) and the easiness of the corresponding task transfer process.

Execution time: $texec^r_{m|_q}(t)$

To obtain $texec^r_{m|_q}(t)$, first, the polynomial that fits the evolution of each $I^r_{m|_q}(t)$ along the last evaluation times is obtained by means of the least squares curve fitting method technique. Second, the zeros of all evaluation indices, $I^r_{m|_q}(t) = 0 \mid$ are computed and, finally, $texec^r_{m|_q}(t)$ is fixed using the closest zero of all polynomials.

Let $P^r_{m|_q}(t)$ be the fitting polynomial of each $I^r_{m|_q}(t)$ that minimizes $\parallel P^r_{m|_q}(t) - I^r_{m|_q}(t, t-1, .., t-k) \parallel^2$ that is obtained using the last k obtained values. This polynomial determines the expected remaining time availability of robot r to continue with the teleoperation using q as TP under the criteria of the mth evaluation index, $texec^r_{m|_q}(t)$. The execution time is obtained selecting the closest zero of its associated polynomial, discarding these zeros previous to the current evaluation time, t, (4.1).

$$\begin{aligned} &texec^r_{m|_q}(t) = t_i \quad \text{where} \\ &t_i = min(t - t_i),\ \forall t_i \mid P^r_{i|_q}(t_i) = 0 \quad \text{and} \quad t - t_i \leq 0 \end{aligned} \tag{4.1}$$

If there is no polynomial zero that fulfils the required conditions, or the found t_i is a extremely long term zero, a maximum value of $texec^r_{m|_q}(t)$ is established. This maximum acts as a confidence time window and also is used to prevent numerical errors. The value of this window depends on the dynamics of the task: the maximum time decreases for fast dynamic movements.

Finally, to determine the estimated execution time of a robot, the most restrictive (shortest) $texec^r_{i|_q}(t)$ is selected, (4.2).

$$texec^r_q(t) = \min\left(texec^r_{1|_q}(t), texec^r_{2|_q}(t), ..., texec^r_{n_m|_q}(t)\right) \tag{4.2}$$

Task Transfer: $ttransf^r_q{}^*(t)$

Determines the estimated required time to execute a task transfer between $RSel$ and $RCan = r$ in $CuTP = q$. $ttransf^r_q{}^*(t)$ is the result of computing the required task transfer time including a safety margin, named Task Transfer Complexity, TTC^r_q.

TTC_q^r is measured in terms of the ratio between the mean of the required joint accelerations during the planned task transfer and the maximum achievable ones. During the trajectory $k = 1...n_k$ samples are observed to obtain the joint velocities. Both expressions, $ttransf_q^r\ ^*(t)$ and TTC_q^r are shown in (4.3). Numerically, $TTC_q^r \in [0, 1]$, being $TTC_q^r \rightarrow 1$ when the required accelerations approach to the robot maximum accelerations.

$$\begin{aligned} &ttransf_q^r\ ^*(t) = ttransf_q^r(t)(1 + TTC_q^r)\ , \quad \text{where} \\ &TTC_q^r = \frac{1}{n}\sum_{j=1}^{n}\left(\left(\sum_{k=1}^{K}\left(\mid \ddot{\theta}_{j,k} \mid / \mid \ddot{\theta}_{j,max}\right) \mid\right)/K\right) \end{aligned} \tag{4.3}$$

Once obtained $texec_q^r(t)$ and $ttransf_q^r\ ^*(t)$, the final evaluation of robots suitability is expressed in terms of useful time, $tus_q^r(t)$, which is the result of subtracting the necessary time to perform a task transfer to the remaining execution time, (4.4). The complete algorithm to determine all tus_q^r is shown in Algorithm 1.

$$tus_q^r(t) = texec_q^r(t) - ttransf_q^r\ ^*(t) \tag{4.4}$$

4.4.2 Need of Task Transfer

This second control module is dedicated to identify $RCan$ (best pair: robot, $CuTP$ to continue with the task) and, if required ($RSel \neq RCan$), determine the need of an immediate task transfer (NTT).

Selecting the continuity robot: $RCnt$

$RCnt$ represents the robot r and $CuTP_q$ with highest tus, including all $RCan_q^r\ \forall r = 1...n_r\ \forall q = 1...n_q$ and $RSel$ in $CuTP$, (4.5).

Algorithm 1 Continuity Robot Computation, RCnt

%Evaluate $RSel$ in $CnTP$

for $i = 1...n_m$ Evaluation Indices **do**

Compute $I^{RSel}_{i|_{CuTP}}$

$texec^{RSel}_{i|_{CuTP}} \leftarrow I^{RSel}_{i|_{CuTP}}$

end for

$texec^{RSel}_{CuTP} = min(texec^{RSel}_{i|_{CuTP}}), \forall i = 1...n_m$

$tus^{RSel}_{CuTP} = texec^{RSel}_{i|_{CuTP}}$ %(1)

%Evaluate every $RCan$

for $r = 1...n_r Robots, r \neq RSel$ **do**

%In every $CuTP$

for $q = 1...n_q, q \neq CuTP$ **do**

for $i = 1...n_m$ Evaluation Indices **do**

Compute $I^r_{i|_q}$

$texec^r_{i|_q} \leftarrow I^r_{i|_q}$

end for

$texec^r_q = min(texec^{RSel}_{i|_{CuTP}}), \forall i = 1...n_m$

Compute $ttransf^{r\ *}_q$

$tus^r_q = texec^r_q - ttransf^{r\ *}_q$

end for

end for

$List\_of\_RCan = sort\_max(tus^r_q), \forall r = 1...n_r$ and $q = 1...n_q$

(1):$ttransf^{RSel\ *}_{CuTP} = 0$

$$
\begin{aligned}
&RCnt = (r,q) \mid tus_q^r = max(tus_{q=q_j}^{r=r_i}, tus_{q=CuTP}^{r=RSel}) \\
&\text{where: } \; tus_{q_j}^{r_i} = max\left(tus_q^r\right) \\
&\forall r = 1...n_r, \; r \neq RSel, \text{ and } \forall q = 1...n_q, \; q \neq CuTP
\end{aligned}
\tag{4.5}
$$

When $RCnt \neq RSel$, the NTT index is computed to determine if a task transfer process must be initiated at the current evaluation time (determine the operative mode: TT or TOp).

Task Transfer Decision

Once $RCnt$ is identified and $RCnt \neq RSel$, the convenience of starting a task transfer process is evaluated. In order to minimize the number of task transfers, the beginning of the process is delayed as much as possible. Thus, even if $tus_q^{RCnt} > tus_{CuTP}^{RSel}$, an immediate task transfer process could not be required. The optimal instant to start is determined by the Need of Task Transfer, the index NTT, (4.6).

$$
NTT = 1 - \frac{tus_{CuTP}^{RSel} - ttransf^{RCnt\ *}}{tus_{CuTP}^{RSel}} \tag{4.6}
$$

NTT considers the relationship between the remaining useful time and the required time to perform a task transfer. $NTT \rightarrow 1$ indicates the need of a task transfer. Fig.4.17 shows an illustrative example of NTT computation process in a four robot MRCP system with two TP. The evaluation process gives the following results: $R1$ requires a low $ttrasnf^*$, but is not selected as the best candidate due to its small $texec$. $R2$, even having the highest $texec$, is not selected due to a high $ttrasnf^*$, that penalizes its tus. $R3$ is the selected candidate due to its highest tus. Once $RCnt$ is determined, the NTT index is computed. In this example, the optimal time instant has not yet been reached and, consequently, an immediate task transfer is not required. The estimated optimum instant to start a task transfer is also indicated in the example.

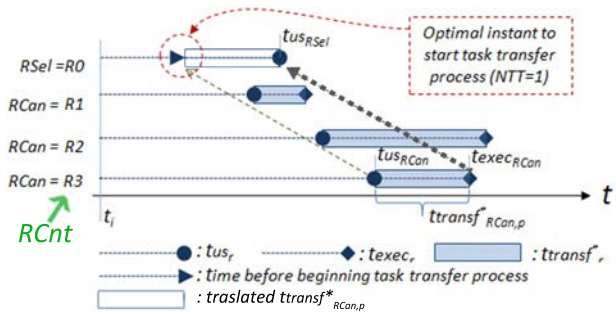


Figure 4.17: Example of determination of the optimal instant to begin a task transfer process ($NTT = 1$).

4.4.3 Robots Actions Computation

The action planner, the last control module, computes the robot actions orders to be executed until the next evaluation process. In TOp mode the action planner generates the robot actions to enable the task-oriented teleoperation and to ease future task transfers. In TT mode, it is the responsible of the task transfer, interacting with the slave station and neglecting the operator orders.

The action planner has been designed following several premises. First, MRCP is a teleoperated system thus, no path planning is possible. MRCP can be interpreted as a discrete system that changes its state at every evaluation time. The robot actions are computed to be accomplished during two consecutive evaluation times. New orders, adapted to the new system state, are sent to the robot controllers at every evaluation time. Second, MRCP generates a complex remote workspace with robots, among other obstacles, sharing their workspace. Every robot represents a dynamic obstacle for the rest of robots. Since this high complex workspace, with moving obstacles prevents the use of path planning, thus a reactive method is chosen instead of a predictive one.

Finally the action planner has to consider that each robot can execute a maximum amount of movement in a fixed period of time (maximum robots dynamics). As a discrete system, the robots are limited to generate a maximum of ΔR_{Max} displacement between two consecutive evaluation times.

To plan the movements, the actions corresponding to each robot are computed.

Each $RSel$ has an associated $CnTP$ as destination that corresponds to that one that maximizes tus_{CnTP}^{RSel}. The algorithm follows a hierarchical approach that determines the order in which the required robot actions are computed. The algorithm starts computing $RSel$ actions and then, the rest of $RCan$. The order of $RCan$ is determined by their associated tus: if $tus^r i > tus^r j$ the actions for $RCan_i$ will be computed before $RCan_j$.

Force guided trajectories computation

The action planner computes the robot actions using a reactive guidance force vector for each robot. Each vector is the resulting sum of attraction and repulsion force vectors. If the resulting force is a null vector (sum of attraction and repulsion forces is zero), a random force is applied. Attraction forces are generated by the goal position whereas repulsion forces are generated by the obstacles (including robots) that intercedes in the attraction vector trajectory.

The force vector based algorithm uses the convex hull representation of each robot link and obstacles, instead of realistic model to decrease the computation complexity, ensure smooth minimum distance vectors evolution and avoid local minima.

In what follows, the force vectors (attraction, repulsion and random) are introduced.

Attraction force: F_{Attr}

The attraction force, F_{Attr} is generated by the minimum distance vector with origin in the robots TCP, p_i, and destination in the pre-grasping position, g_i. One of the firsts formal approaches to an attraction force was developed in [146,147], defining an attractive force field that depends on the distance between the robot and the obstacle, (4.7), where $d = \mid p - g \mid$, p is the current robot pose, g the goal pose or attraction point and ξ an adjustable constant.

$$U_{Attr} = \frac{\xi d^2}{2} \quad \text{and} \quad \nabla U_{Atttr} = \xi(p - g) \tag{4.7}$$

Following this approach, the amount of force applied on the robot increases quadratically with distance. In MRCP, the attraction vector is developed under the premise of a greedy approach: reach the goal point as soon as possible. This principle breaks the relationship between distance and force. At every evaluation time, the robot must generate the maximum allowed movement, ΔR_{Max}, until the minimum distance $d_i = g_i - p_i$, is minor than this maximum movement, (4.8). The direction of the attraction force is determined by the versor of the distance vector, $\hat{d}_i$.

$$F_{Attr,i} = \begin{cases} \Delta R_{Max} \cdot \hat{d}_i & \text{when } |\vec{d_i}| \geq \Delta R_{Max} \\ \vec{d_i} & \text{when } |\vec{d_i}| < \Delta R_{Max} \end{cases} \tag{4.8}$$

Repulsion force: F_{Rep}

In order to avoid collisions between each robot and the obstacles (including the rest of the robots with better robot suitability evaluation), a repulsion force is included in the trajectory computation. Unlike F_{Attr}, which is unique, multiple repulsion force vectors, F_{Rep}, are generated: one from each obstacle. Every obstacle generates its own repulsion force and, following the hierarchical approach, the ith best evaluated robot has, at least, i repulsion vectors generated by R_{Sel} and the $i-1$ better evaluated robots.

Several proposals have been done to define repulsive force fields. Again, as one of the first references, in [146, 147], a repulsive field U_{Rep} that depends on the distance between the robot and the obstacle is defined, 4.9.

$$U_{Rep} = \begin{cases} \frac{1}{2}\eta \left(d^{-1} - d_0^{-1}\right) \frac{(q - q_0}{d^3} & \text{when } d \leq d_0 \\ 0 & \text{when } d > d_0 \end{cases} \tag{4.9}$$

Repulsion forces, like the above presented, suffer from local minima (e.g. objects with concave shapes) as well as mutual blocking (the contribution of attraction and repulsion forces generate a null force vector). One such solution is the use of potential fields with different behaviours or primitives: uniform, perpendicular, tangential, ...

to the obstacle surfaces, random, etc. An illustrative example of these primitives is shown in Fig.4.18. The combined use of these primitives and the attraction force field smoothly guides the robot to the goal overcoming the previous mentioned problems. Several examples of potential field combinations can be found in [148, 149].

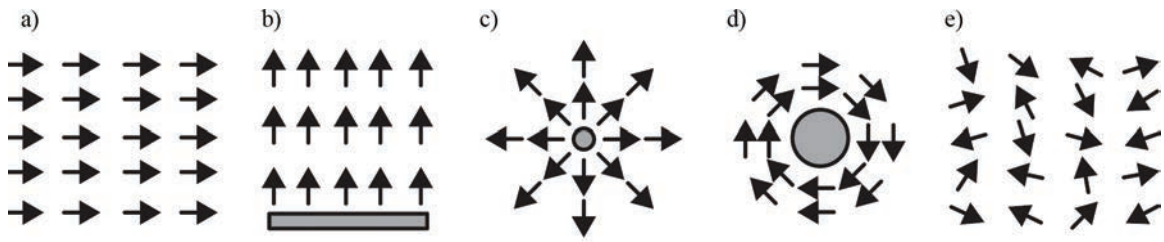


Figure 4.18: Examples of potential fields primitives: a) uniform, b) perpendicular, c) repulsion, d) tangential and e) random.

In MRCP, a modification of these primitives is used: the approach proposed is based on the tangential field but applying a variable direction field instead of generating a fixed one around the object. The repulsion force generated by each obstacle can be seen as a directional force field defined around its geometry. It is defined in terms of its magnitude, $h(\vec{d_i})$, and its direction, $-\Delta\hat{u}$, as shown in 4.10. The direction of F_{Rep} is determined by the evolution of the minimum distance vector over the obstacle surface, expressed as the versor $\Delta\hat{u} = (u_i - u_{i-1})/ \mid u_i - u_{i-1} \mid$.

$$F_{Rep,i} = \begin{cases} \Delta\hat{u} \cdot h(\vec{d_i}) & \text{when } |\vec{d_i}| > |\vec{d}_{i-1}| \\ 0 & \text{when } |\vec{d_i}| \geq |\vec{d}_{i-1}| \end{cases} \tag{4.10}$$

The repulsion force magnitude, $h(d_i)$ is defined as a function of the minimum distance vector, $\vec{d_i}$ between the robot and the obstacle. The function behaves as following: varies inversely proportional to the force to be applied with respect to the distance ($\mid \vec{d_i} \mid \rightarrow 0 \Rightarrow F_{Rep} \rightarrow \Delta R_{Max}$ and $\mid \vec{d_i} \mid \rightarrow \infty \Rightarrow F_{Rep} \rightarrow 0$). The parameter l modulates the response of the function.

$$h(d_i) = \frac{\Delta R_{Max}}{1 + \mid d'_i \mid^l} \quad \text{where} \quad l \geq 1 \quad \text{and} \quad d'_i = d_i + d_{offset} \tag{4.11}$$

In Fig.4.19 four parametrizations of $h(d_i)$ are shown. The parameter l generates different response function: when l increases, the slope increases. In the graphics, no offset is present ($d_{offset} = 0$). This parameter is used to determine the shape of the repulsion force around the obstacle. When the obstacle is isolated, low l values are recommended as they generate smooth force responses. When obstacles generate narrow passages, higher values of l generate corridors that the robot can cross to reach the final goal pose.

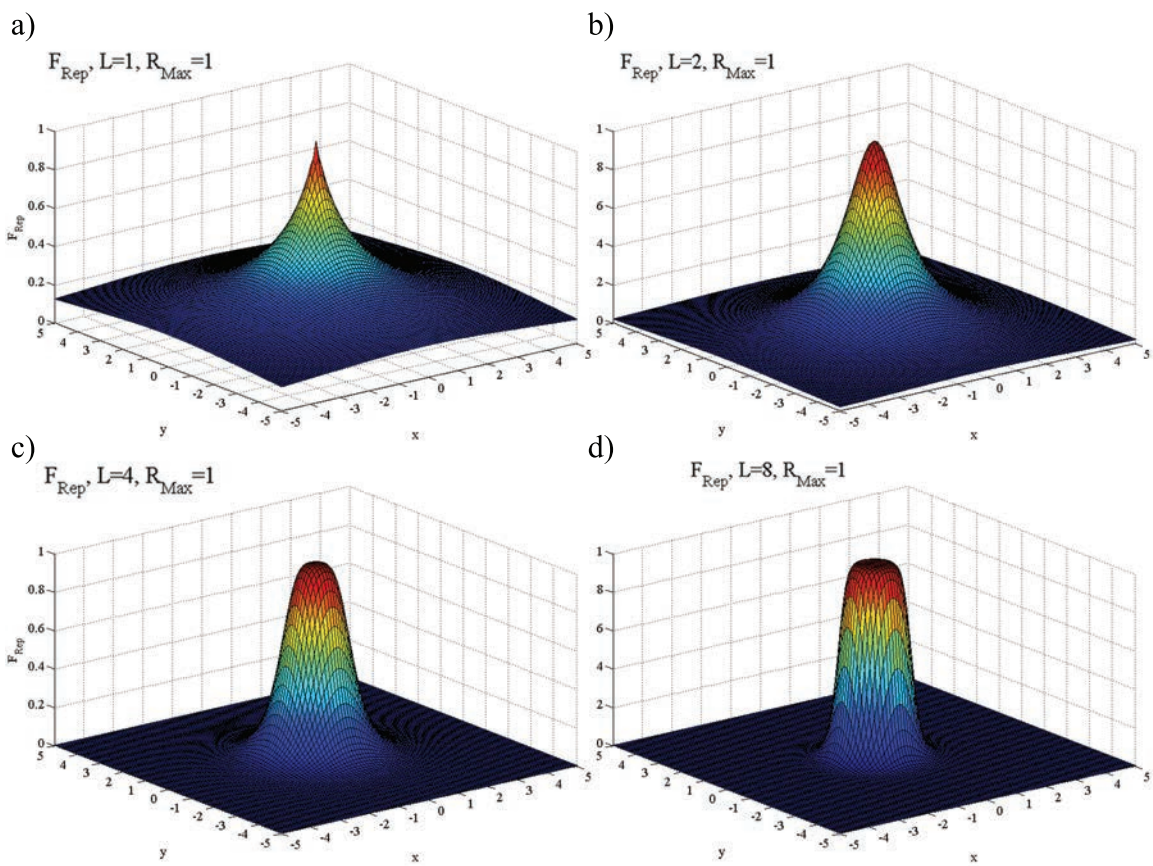


Figure 4.19: Examples of proposed potential field with different values for parameter l: a) $l = 1$, b) $l = 2$, c) $l = 4$ and d) $l = 8$.

Random Force: F_{Rand}

The combined use of attraction and repulsion forces may cause a null force vector $F_{Attr} = \Sigma F_{Rep}$, deadlocking the robot trajectory. To break this local minima, a random force field is usually injected in the final guidance force vector. Random forces, introduced in [150], are an efficient and low computational cost solution to break the forces equality. To avoid unexpected collisions due to the random force direction, the contribution of $\mid F_{Rand} \mid$ in the final guidance force must be residual. Recommended values of $w_{Rand} = [0, 0.2]$

Robots Guidance Resulting Force: F

Final guidance force is the result of the contribution of F_{Attr}, all the F_{Rep} and, if required, F_{Rand}. The resulting force must guide the robot to its pre-grasping pose while avoids collisions using the complex force field generated by the attraction and repulsion forces. Some authors introduce blending functions in complex vector fields composed by more than a single field (e.g. attraction and multiple repulsion fields) to determine the contribution of each individual field into the final one, smoothing the vector fields, [151, 152]. In MRCP, the blending function is generated by the l parameter of the F_{Rep} function, which modulates the behaviour of the repulsion forces. 4.12 shows the expression of the final guidance force for each robot.

$$\begin{aligned} &F = F_{Attr} + \Sigma F_{Rep} + w_{Rand} \cdot F_{Rand} \\ &\text{where:} \quad F \leq \Delta R_{Max} \quad \text{and} \quad 1 \geq w_{Rand} \geq 0 \end{aligned} \tag{4.12}$$

Fig.4.20 illustrates the F_Attr, F_{Rep} and final guidance force F computation. Fig.4.20.a shows the attraction force generated in the minimum distance vector, $\vec{d_i} = g_i - p_i$ that cannot exceed the maximum robot movement capacity, ΔR_{Max}. Fig.4.20.b shows the computation of the repulsion force with direction $\Delta\vec{u}$ and magnitude, $|\vec{F}_{Rep}|$, that depends on the minimum distance vector, $\vec{d_i}$. Finally, in Fig.4.20.c the final guidance force computation is shown.

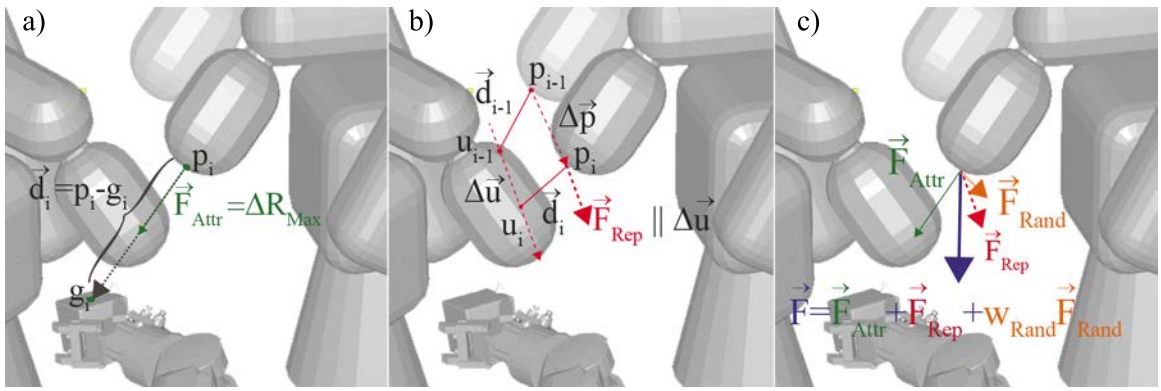


Figure 4.20: Example of guidance force computation: a) Attraction force, b) Repulsion force, c) Resulting guidance force.

The Algorithm 2 computes the required robot movements, $\Delta\vec{X}RSel_{t=i+1}$, for a MRCP with $r = 1...R$ robots and L obstacles in evaluation time $t = i$. It uses the obstacle position list $ObstacleList = \{\vec{X}_{Obstacle1}, .., \vec{X}_{ObstacleL}\}$ obtained from the workspace sensors and the previously computed priority robot list in which the hierarchical process is based $RPriority = \{RSel, RCan_1, .., RCan_{R-1}\}$ where $tus(RCan_i) \geq tus(RCan_j), \forall(i < j)$. RPQCollision and RPQMinDistance are functions of RPQ library in which the control algorithm is based (RPQ is used as development platform for MRCP control algorithm). Finally, ComputeForce is the function to compute the forces to be applied to the $RCan$, which has been described in the second part of this section.

4.4.4 MRCP Control Implementation

The MRCP control algorithm has been developed based on the Object Oriented paradigm. It has been programmed in C++, to generate a multiplatform code with high computation efficiency. MRCP uses the Robotic Proximity Queries package, RPQ, which was implemented on the initial phase of the MRCP development, [153, 154]. RPQ was initially developed to compute the minimum distance vectors and collision queries between serial kinematic chains sharing their workspace in real time. Three main classes compose the RPQ library: Object, Robot and Scenario. The

Algorithm 2 Robot Actions Computation

$\Delta\vec{X}RSel_{t=i+1} = \vec{X}RSel_{t=i+1} - \vec{X}RSel_{t=i}$

$QC = \text{RPQCollision}(\vec{X}RSel, ObstacleList)$

if $QC == TRUE$ **then**

exit("$RSelCollision$")

else

%Compute movement for every $RCan$

for $K = 1..R-1$ Robots **do**

$QD = \text{RPQMinDistance}(\vec{X}RSel,\ ObstacleList)$

$\vec{F}RCan_{k,t=i+1} = \text{ComputeForce}(\vec{X}RCan_k, QD)$

$\vec{X}RCan^k_{t=i+1} \leftarrow \vec{F}RCan^k_{t=i+1}$

$\Delta\vec{X}RCan^k_{t=i+1} = \vec{X}RCan^k_{t=i+1} - \vec{X}RCan^k_{t=i}$

$ObstacleList = \text{AddToList}(\vec{X}RCan^k_{t=i+1})$

end for

end if

library has been updated to include the MRCP requirements, modifying the original classes and adding some new classes: RobotExtended and MRCPControl.

Scenario is the workspace where the objects (robots and obstacles) cohabit, reflecting the real teleoperation remote workspace. Concerning its implementation, Scenario is a class that contains all the objects (Robots and obstacles), a global reference frame, and all the methods necessary to generate the proximity queries.

An Object is the minimum entity that exists in a Scenario. Objects can be simple or complex. A simple Object is represented by a geometrical model composed of a set of triangles referred to a frame tied to the Object and a transformation matrix to refer itself to the world reference frame. Simple objects are used to represent and follow the trajectory of the obstacles into the workspace. A complex Object is an Object composed of a set of geometrical models with joints between them. The complex objects are used to represent the robots.

A Robot is a complex Object where each of its links is represented by a simple Object. A Robot has a set of functions to describe the robots following the Denavit-Hartenberg notation, compute the direct and inverse kinematics considering the robots own restrictions (joint limitations, configurations, etc). Concerning implementation, the class Robot is derived from the class Object. Robot adds all the functions that are necessary to control a robot.

To achieve the MRCP requirements, two new classes have been added to the original RPQ library. The RobotExtended class, which includes the robot evaluation indices, the functions to transform the indices into time space and, finally, the estimated zero-cross function to estimate the instant of time when each index becomes zero. The MRCPControl class contains the main control algorithm, the input signals from the master devices and the robot actions computation, the functions to determine the most suitable robot to continue the teleoperated task execution and the evaluation of an immediate task transfer convenience are included in this class.

A robot controller emulator was developed to simulate the robots observing their real restrictions, safety parameters, etc. The real robot controllers have a high level security layer that restricts their dynamics and working limits. This layer has been

included in the emulator to ensure the fidelity between the emulated and real robots.

A high level communication layer between robots (Staübli RX60B with CS7 and CS8 controllers and ABB IRB140) and MRCP control has been programmed. This layer enables the communication with any controller isolating the MRCP from the implementation aspects. This high level communication layer includes all the required commands to perform robot actions (position and velocity movement commands, tool setting, robot status request, etc).

4.5 Experimental Results

This section presents several telemanipulation experiments to test and validate the internal operation of MRCP. These experiments require that the operator executes a task consisting on moving an object in the remote workspace. The experiments, more than requiring from the operator to follow a concrete trajectory or execute a fine telemanipulation, are designed to force the robots to fall in singular configurations so that a task transfer is required. Detailed information of each experimental set-up is provided in each experiment description. All the evaluation indices used to determine the robots suitability and the best instant of time to execute a task transfer are presented.

In all the experiments, the robots, other than $Rsel$, should move as close as possible to their pre-grasping positions, minimizing the task transfer time. $ttransf$ for each robot is determined by the necessary time to perform a straight movement between the current pose of the robot and the grasping pose, executing this path at half speed and assuming that neither obstacles are present nor singularities can be produced along the path.

Some assumptions are done in the experiments. First, concerning the manipulated object, the geometry is know and the set of grasping points are pre-defined. As all the theory and practical aspects concerning grasping are out of the scope of this work, it is assumed that these grasping points ensure stability to the manipulated object in static and dynamic mode. Second, in reference to the workspace, the geometry and

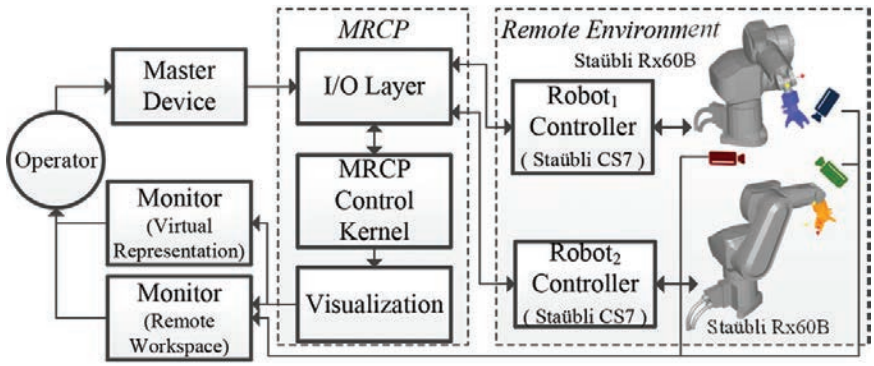


Figure 4.21: Block diagram of the experimental MRCP set-up.

the instantaneous position of the obstacles, other than the own robots, is known and updated in real time.

Concerning the MRCP control algorithm, it has been programmed using the oriented object paradigm and implemented in C++. For the experiments, the MRCP control algorithm is executed using a Windows XP Pro platform in an Intel Core2 Duo with 4GB RAM memory equipped with a NVidia GeForce 550Ti graphic card that provides a graphical representation of the MRCP and its workspace. The communication between MRCP and the robot controllers (real or simulated depending on the experiment) is determined by the robot controller specifications: 10Mb private Ethernet network, at a frequency of 556 Hz.

4.5.1 Simulated telemanipulation task

The MRCP set-up is composed of a 6 DoF master device (3DConnection Space Mouse) and two simulated 6 DoF Staübli Rx60B robots, both equipped with a gripper. The operator orders are directly mapped as incremental movements of the manipulated object (translation movements are mapped onto a fixed frame while rotations are mapped according to the objects reference frame). Fig. 4.21 shows a block diagram of the experimental set-up, which is valid for both the simulated and the real scenarios.

In this experiment MRCP executes several task transfers facing different critical situations. The operator manipulates an object with two predefined grasping posi-

tions, from an initial position to a final target point passing through a hole of a moving obstacle. Fig.4.22 shows a set of snapshots of the task indicating the different causes that force each task transfer. Initially, $R0 = RSel$ whereas $R1$ is on an initial random configuration, Fig.4.22a. and, for the complete task execution, three task transfers are required: in $t \approx 5.5s$ Fig.4.26.c, $t \approx 11.5s$ Fig.4.26.e and $t \approx 15.5s$) Fig.4.26.g.

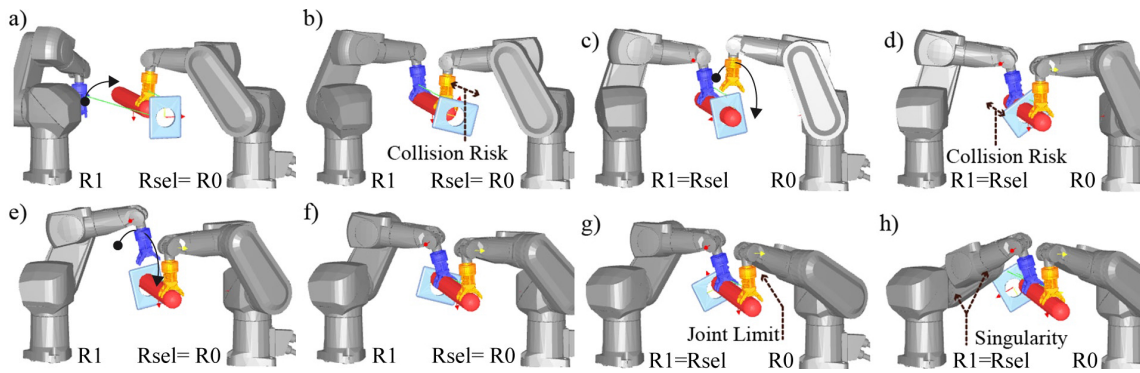


Figure 4.22: Sequence of snapshots of MRCP executing a manipulation task.

At the beginning of the task, high values are observed for $ttransf^{R1}$, Fig.4.26.b, and for NTT, Fig.4.26.c, that are consequence of the distance between $R1$ and its grasping position, $CnTP$. Before the first task transfer, Fig.4.22.a and .b, the CR index between each robot and the moving obstacle determine the two robots' suitability ($texec^{R0} = texec^{R0}_{CR}$ and $texec^{R1} = texec^{R1}_{CR}$). The NTT value reflects that, even if $texec^{R1} > texec^{R0}$, the task transfer can be delayed.

Between $t \approx 5.5s$ and $t \approx 11.5s$, Fig.4.22.c and .d, $RSel = R1$. $texec^{R1}$ is still determined by the CR index and $texec^{R0}$ by the unavailability of a free grasping position (occluded by the obstacle) until $t \approx 7.8s$, then by $texec^{R0}_{RJ}$, until $t \approx 8.1s$ and, finally by $texec^{R0}_{\psi}$, as shown in Fig.4.26.a. The obstacle avoidance trajectory increases $ttransf^{R0}$ and NTT ($t \approx 6s$ and $t \approx 8s$), Fig.4.26.b.

During the third part of the task, $t = 11.5s, t \approx 15.5s$, Fig.4.22.e and .f, $RSel = R0$ and $texec^{R0}$ is determined by $texe^{R0}c_{\psi}$ until $t \approx 13s$, and then by $texec^{R0}_{RJ}$. Finally, a joint limit in $\theta 5$ prevents $R0$ to continue with the task execution, Fig.4.23. $R1$ executes a collision avoidance trajectory. Notice the temporal unavailability of $R1$ ($texec^{R1} = 0$) due to an occlusion of $CnTP$ as well as the increasing values of $ttransf^{R1}$ and

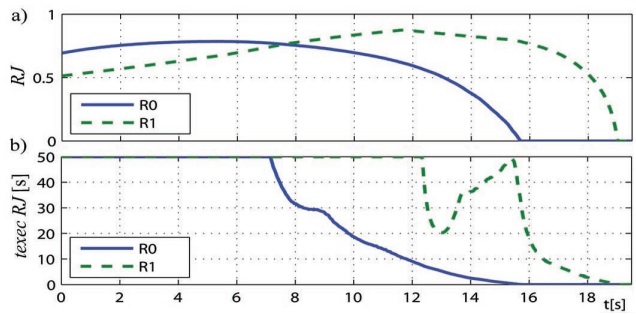


Figure 4.23: a) RJ_{R0} and RJ_{R1} indices. b) Estimated execution time for RJ_{R0} and RJ_{R1}.

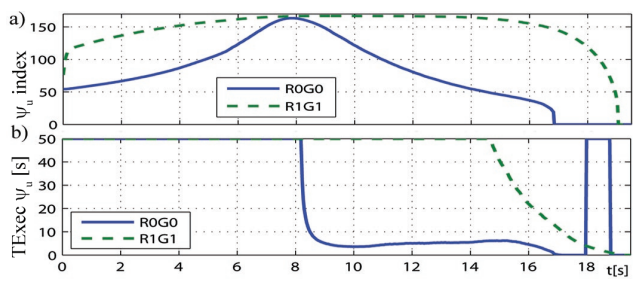


Figure 4.24: a) ψ_{R0} and ψ_{R1} indices. b) Estimated execution time for ψ_{R0} and ψ_{R1}.

NTT. Once $CnTP$ is reachable, $texec^{R1}$ is determined by $texec^{R1}_{RJ}$ and finally by $texec^{R1}_{\Theta}$.

From $t = 15.5s$ to $t \approx 20.5s$, $R0$ reaches a joint limit, Fig.4.22.g and $RSel = R1$, Fig.4.22.h. Finally, ψ_{R1} index indicates the vicinity of a singularity (alignment of two links), Fig.4.22.h, indicating the end of the $R1$ availability.

4.5.2 Real telemanipulation tasks

Next experiments consist of two telemanipulation tasks. An operator telemanipulates a cube with six different grasping positions (one at each face). These experiments differ from the previous one in the use of a real telemanipulation station. The set-up consists of the same 6 DoF master device and two real 6 DoF Staübli RX60B robots equipped with pneumatic grippers as remote slaves. The robots are fixed in their configuration to avoid abrupt movements which cannot be controlled by the MRCP

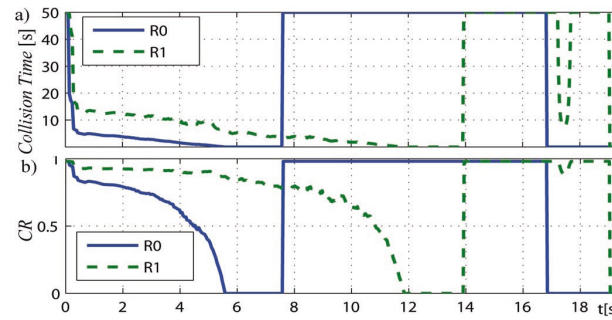


Figure 4.25: a) R0 and R1 estimated Collision Time. b) CR_{R0} and CR_{R1} indices, where $CR = 1$ indicates no Collision Risk and $CR = 0$ indicates no robot availability.

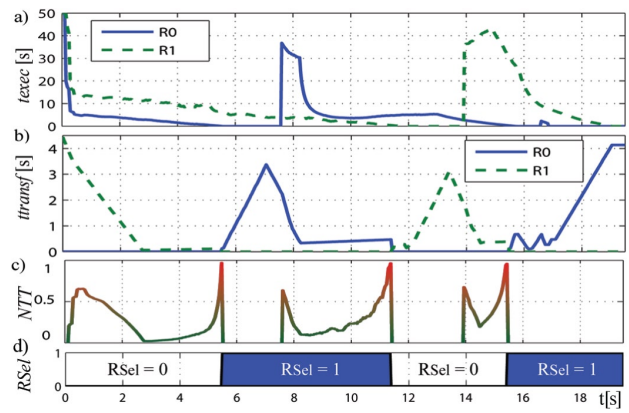


Figure 4.26: a) Execution time, $texec$ ($texec = 0$: robot is not suitable). b) Required Task Transfer, $ttransf$. c) NTT index evolution. d) $RSel$

robot actions planner. The set-up is completed with a camera providing the operator with information about the manipulated object inside the remote workspace. In the first experiment the operator rotates an object in front of the camera. In the second one, the operator translates the object in the XZ plane from an initial position to a goal position and, again, leave the object in its initial position.

Rotation telemanipulation task

The first telemanipulation task, Fig.4.27, consists in rotating the cube. Two situations must be pointed out: first, a task transfer and, second, a change on the best evaluated

grasping position (RCan changes its CnTP). Initially, Fig.4.27.a, $R1$ robot, right on the image sequence, holds the object, while $R0$, left on the image sequence, is in a pre-grasping position. $R1$ executes the task, Fig.4.27.b and .c, until it approaches a joint limit in $\theta 5$ ($texec^R 1_{RJ} \rightarrow 0$ and $NTT \rightarrow 1$). After the task transfer, Fig.4.27.d, $R0$ continues with the task execution while $R1$ moves towards its best evaluated $CnTP$, Fig.4.27.e and .f.

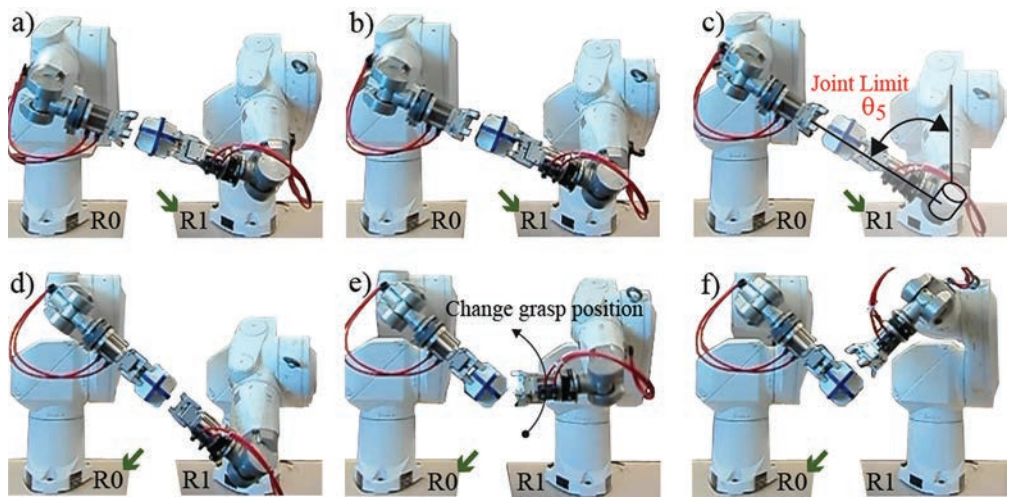


Figure 4.27: Snapshots sequence of the first real teleoperation experimental task.

Traslation telemanipulation task

In this second teleomanipulation task the operator moves the cube, which is laying on a table, along a vertical plane up to a determined height and then places it again on the table. The table represents an obstacle inside the workspace and determines the initial and final grasping position. Fig.4.28 shows a set of snapshots of the task execution.

Initially $R0 = RSel$ (left on the images), whereas the other robot, $R1$ (right), is on a random initial configuration, Fig.4.28.a. As soon as the task is started, $R1$ moves towards the best $CnTP$, as shown in Fig.4.28.b. A task transfer is needed when $R0$ is close to reaching a joint limit ($\theta 5$), Fig.4.28.c. Once the task transfer is

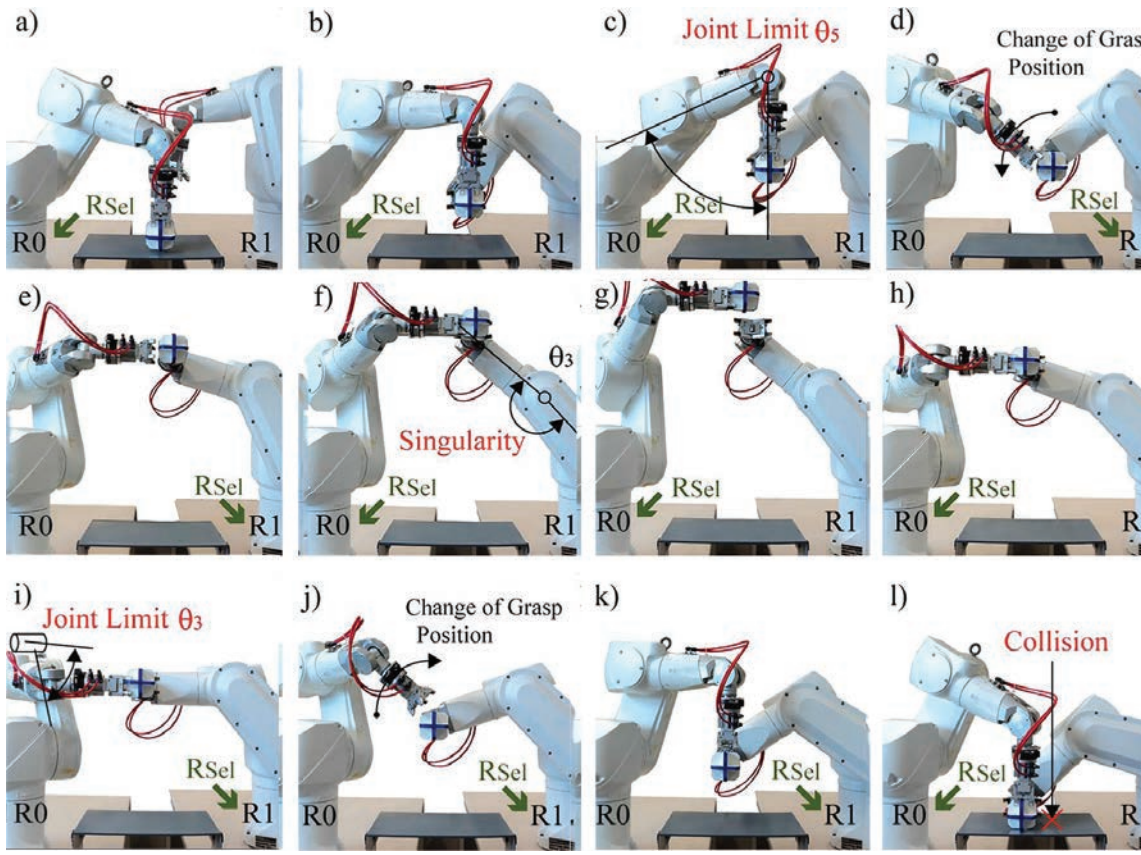


Figure 4.28: Snapshots sequence of the second real teleoperation task.

finished, $R1$ continues with the task, Fig.4.28.d and .e, while $R0$ changes its grasping position following the maximum *tus* criteria. In Fig.4.28.f, $R1$ is in the vicinity of a singularity produced by the alignment of two links ($L2$ and $L3$) and, consequently, $R0$ retakes the execution of the task. $R0$ continues with the task until it reaches again a joint limit, $\theta 3$, Fig.4.28.i. Finally, the presence of an obstacle in the direction of the predicted trajectory (the table), which is reflected in the CR index, modifies the $RSel$, Fig.4.28.j and .k. Finally, $R1$ stops the telemanipulation due to an imminent collision. $R0$, using the new grasping position (top of the cube) leaves the object on the table without colliding.

Table 4.2: Table with the states, transitions, inputs, outputs and operational modes of the MRCP control

State	Op Mode	Input	Transition when	Output
Robots Evaluation	TOp	Master Commands ($\vec{X}_{Mast}$), Slave Status output	Computed robots evaluation	Evaluation of robots in time space ($tus(R1), .. , tus(Rn)$)
NTT	TOp	Robots Eval Output	Need of Task Transfer determined (NTT)	Operation mode (TOp or TT) and NTT index
Robots Actions in TOp	TOp	New task status	New robot actions sent to controllers	New robot actions for teleoperation ($\vec{X}'_{R1}, .., \vec{X}'_{Rn}$)
Robots Actions in TT	TT	Slave system status	New robot actions sent to controllers	New robot actions for task transfer ($\vec{X}'_{R1}, .., \vec{X}'_{Rn}$)
Slave System	TOp, TT	Output of Robot Actions TOp or Robot Actions TT	Robot controllers generate response	Op Mode (TOp or TT), Robots and Sensors outputs

Chapter 5

Experimental results: MRCP in MIS

5.1 Introduction

This section presents several experiments to validate and demonstrate the capabilities of the proposed Multi-Robot Cooperation Platform, MRCP. The experiments deal with different teleoperation tasks that demand high dexterity and precision in the field of Robotic Minimally Invasive Surgery, RMIS. Several reasons have led to the selection of RMIS to demonstrate the contribution of MRCP in improving multi-arm teleoperation, which are: a) RMIS demands precise telemanipulation, b) RMIS presents complex and dynamic environments, and c) RMIS has proven its usefulness and benefits for the patient.

MRCP aims to increase the applicability of current robotic systems in RMIS. The proposed approach should facilitate surgeons the execution of different surgical tasks by means of changing the paradigm of classical teleoperated systems. The use of MRCP does not imply the suppression of standard bi-manual teleoperation; on the contrary, both can be complementary and be applied depending on the task or sub-task to be executed during the intervention. The surgeon can select the most adequate teleoperation mode. The set-up is based on the cooperative use of various robotic arms following the MRCP teleoperation paradigm and uses the current multi-robot architecture present in most RMIS systems (DaVinci, Zeus, MiroSurge, Bitrack,

etc.).

Open surgery provides absolute sensitive immersion: the surgeon interacts with organs and tissues obtaining direct tactile and visual feedback. Dexterity and tool movements are not restricted, allowing all the human degrees of freedom, DoF. MIS decreases the workspace perception in terms of tactile feedback, which is provided through the tool and altered by the tool flexion and the friction produced by the cannula. In [155] a review of the haptic information in MIS is presented, while [156] describes a case study of haptic feedback in a concrete laparoscopic surgery. Surgeons have to perform an intervention using long tools with their movements restricted by the fulcrum point, which reduce their dexterity and DoF. Visual feedback is normally limited by the endoscopic camera, providing partial 2D information. Two hands procedures, like suturing, are outstandingly affected by these restrictions. Except for some research works [157–159], null tactile feedback is provided. The surgeon dexterity is also restricted by the tools (a reduced set of tools are available) and the fulcrum point. Robotic tools offer improved dexterity thanks to the higher number of DoF with respect to those used in MIS. RMIS also offers some other performances as, tremor compensation [160], motion scaling [161, 162] or trajectory guidance [163]. Visual feedback, although indirectly provided by a camera, is improved using stereoscopic vision, camera stabilization and motion tracking [164, 165].

MRCP has been conceived with the aim of overcoming the limitations of robots in surgery in terms of kinematics or the robot workspace limits. These intrinsic causes prevent robots from completing the desired task, requiring their repositioning and thus, increasing the surgery time. The surgical workspace also imposes several restrictions: collisions, occlusions, etc. increasing surgery risks and decreasing patient safety. In MRCP, when one of these mentioned problems is detected and, acting in advance to avoid the need of stopping the task, a task transfer between robots is executed selecting dynamically the best robot to continue with the ongoing task.

Four experiments are presented in this section, all based on a multiple robot teleoperation set-up using different number of slave robots and various types of master devices (depending on the experiment requirements). The first experiment presents

the use of a task-oriented robots evaluation index to constraint the cannula movements in the entry points in RMIS. The second experiment, based on a simulated suture process, identifies the benefits of MRCP in front of classical teleoperation. A third experiment, an evolution of the previous one, studies the interaction forces and torques between tools, needle and deformable tissues. Finally, a suture experiment is presented, which shows the combined use of standard teleoperation, StdT, and task oriented teleoperation using the same teleoperation platform. The experiment shows the decomposition of a complex task into different sub-tasks and how the operator can switch among the different teleoperation modes, selecting the most adequate to accomplish all sub-task requirements.

5.2 Experiment 1. The use of virtual fixtures to constraint tool movements in RMIS

The first experiment evaluates the use of MRCP applied to a bi-manual task. Apart from the MRCP validation as a feasible RMIS solution, two more objectives are satisfactorily fulfilled: first, demonstrating the modularity of the control architecture, introducing a new task oriented evaluation index without altering the control schema, and second, testing virtual fixtures, in the form of force feedback in the MRCP paradigm. This work was presented in [130].

Task Oriented Evaluation Index: Tool Orientation Cone

The Tool Orientation Cone task dependant index measures the orientation of the surgical tool in the fulcrum point with respect to the abdominal wall. This index avoids excessive forces and torques produced by the troccar that cause patient injuries like haematoma. In manual surgery, the surgeon feels these forces and can prevent hurting the patient. Instead, in robotic surgery, there is no force sensing that allows the control of the applied efforts. The Tool Orientation Cone is designed to prevent this shortcoming.

The orientation angle is computed using the normal to the abdominal wall on the fulcrum point and the orientation of the tool. The value of the index, angular distance, is obtained using the quotient between the current tool orientation and the maximum permitted orientations, defined as a cone with its base represented as an ellipsoid. As in the RJ index, the angular distance is modulated using the same modulation function: five parameter probabilistic function. Fig.5.1 illustrates two views of the tool orientation and the maximum orientation cone.

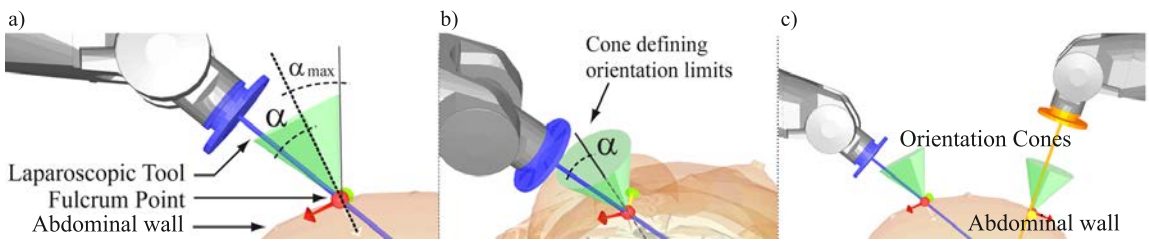


Figure 5.1: a) General description of the tool, the tool orientation, α, and maximum orientation cone; b) representation of the virtual cone and c) general view of two robots, the abdominal wall and the virtual protection cones

5.2.1 Experimental Set-Up

Two set-ups have been implemented for the experiment: The first one is composed of two simulated 6 DoF robots equipped with a laparoscopic tool as end effector, whereas the second one uses real Staübli Rx60B robots. The MRCP control runs under a Windows XP platform in an Intel Core2 Duo. The communication between MRCP and the robot controllers (real or simulated) is done using a private Ethernet network, at a frequency of 55.6 Hz (established by the robot controllers). The human operator uses a Phantom Omni haptic device, which provides 6 DoF movements and generates 3 DoF of force feedback, as master. The movement orders are directly mapped on the manipulated object: Translation onto a fixed frame and rotations according to the objects reference frame. Fig. 5.2 shows a snapshot of an operator guiding the used master device. This picture is part of a previous test in which, the operator manually has to follow a set of predefined trajectories with the master

device, allowing the study of the behavior of the evaluation indices and the trajectory predictor in a known set of trajectories.

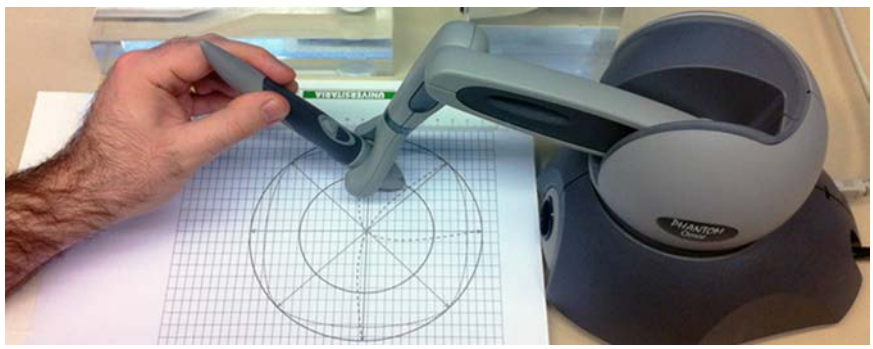


Figure 5.2: Force feedback haptic device used as master.

5.2.2 Experimental results

In this subsection the results obtained with MRCP using the virtual workspace (two simulated robots and a virtual abdomen), Fig.5.3.a are analysed. The operator telemanipulates an object inside the umbilical region. The manipulated object must be repetitively crossed under an elevated piece of the small intestine, which position is known and acts as an obstacle to be avoided, Fig.5.3.b. Fig. 5.3.c shows the virtual endoscopic view. The object has two predefined grasping positions, indicated in the graphic as $Gr0$ and $Gr1$ respectively.

For the experiment, four evaluation indices have been used: joint limits, anisotropic dexterity (both intrinsics), collision avoidance (extrinsic) and Tool Orientation Cone (task oriented). $ttransf$ has been determined as the necessary time to perform a straight movement from the current robot position to its $CnTP$, executing this path at half speed.

The following part describes the results obtained for an object manipulation following a trajectory that crosses the obstacle lower part (generating the need of a task transfer). Apart from this required task transfer, the Tool Orientation Cone also prevents the execution of excessive forces in the abdominal wall). As can be

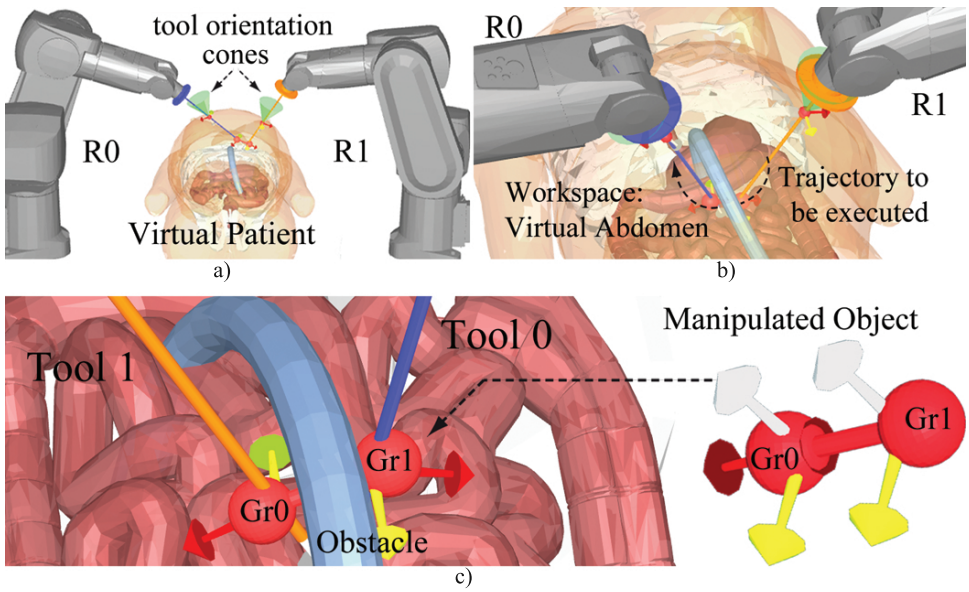


Figure 5.3: a) General view with the virtual cones and both robots. b) Manipulated obstacle with the two predefined grasping positions. c) Virtual endoscopic view of the workspace.

seen in Fig. 5.5.a, where R_Sel is shown, a task transfer is necessary during the task execution($t \simeq 9s$). In Fig.5.4.a, RJ shows the evolution of the joints during the task. The values of both indices show that no risk of reaching a limit is detected. In Fig.5.4.b Θ shows how $R0$ is moving towards a singularity (produced by the alignment of two links). The value of $R1$ indicates that this robot evolves from a poor dexterity region to a more dextereous one. Fig.5.4.c shows the cause of the task transfer: $R0$ reaches its maximum permitted orientation, becoming a non-valid candidate to continue with the task execution ($t \simeq 9s$). $R1$ is not a valid R_{Can} at the beginning, but its Tool Orientation index value increases during the task execution, becoming, first, R_{Can} and, finally, R_{Sel}.

Fig.5.5.a shows $ttransf$ of both robots. During the first part of the task ($Ro = R_{Sel}$), the $ttransf_{R1}$ low value indicates a non complex task transfer ($R1$ is close to its $CnTP$). From $t > 10s$ $ttransf_{R0}$ increases due to the impossibility of this robot to be close to its $CnTP$.

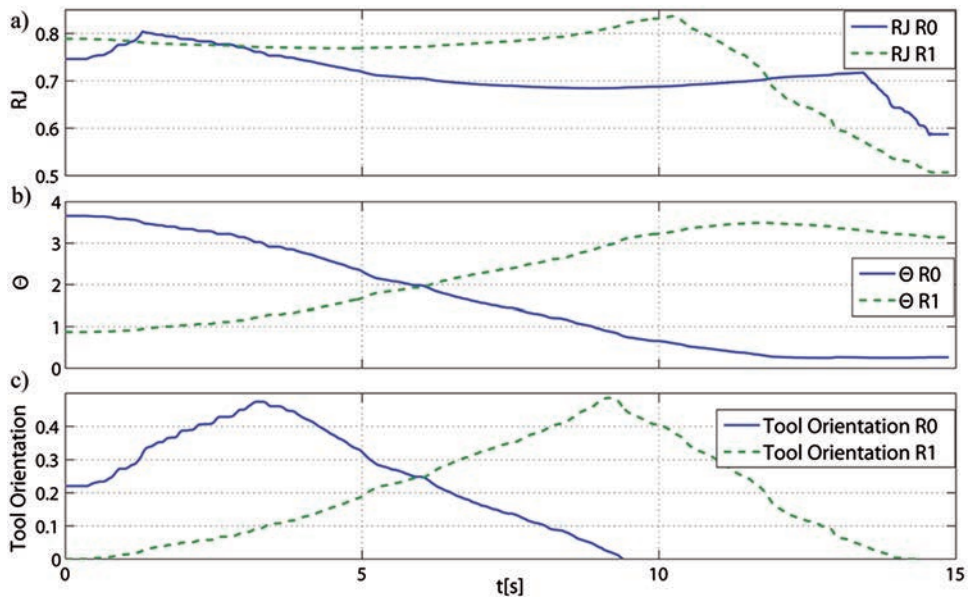


Figure 5.4: a) RJ, b) Θ and c) Tool Orientation indices evolution along task execution

Fig.5.5.b shows the NTT evolution. During the first $5s$, the need of task transfer increases with a constant slope, but its value indicates that a task transfer is still not required. Between $t \simeq 5s$ and $t \simeq 8s$, the fluctuations are produced by changes on the velocity of the operator's movements. Between $t \simeq 8s$ and $t \simeq 9s$ NTT increases its value until $NTT \rightarrow 1$, indicating the need of an immediate task transfer.

Once the experiment was evaluated using a virtual scenario, it was replied into a real remote workspace with two Staübli Rx60B robots. The obtained results were equivalent to those obtained in the simulated scenario. Fig. 5.6.a, shows the master console. That console consists of a Phantom Omni master device and a monitor with the endoscopic view, apart from other information (tool orientation, active robot, ...). Fig.5.6.b shows the slave robots and the workspace: two robots with the laparoscopic tools and a third robot holding the endoscopic camera.

This experiment is also used to apply Virtual Fixtures, VF, using MRCP. Two virtual rings are placed on the workspace. The operator must cross them while holding the telemanipulated object. The ring generates a repulsion force to avoid undesired contacts with the object and acts as a guidance function to correctly insert the object

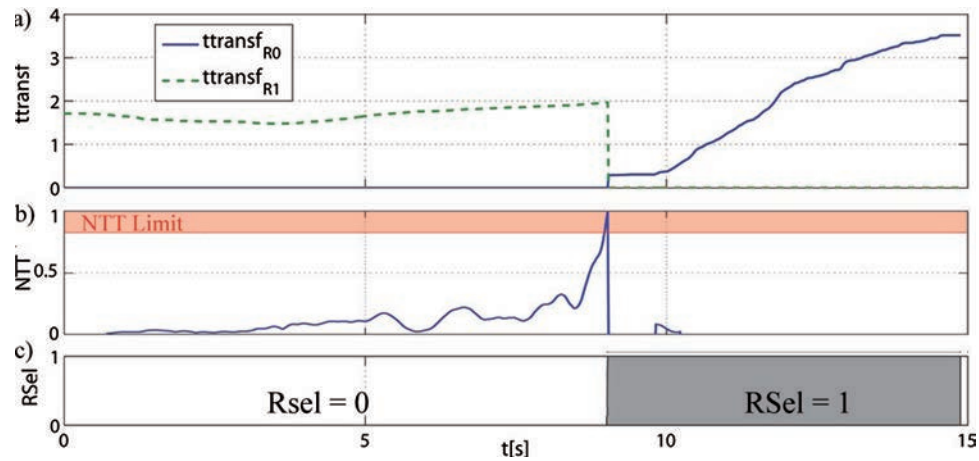
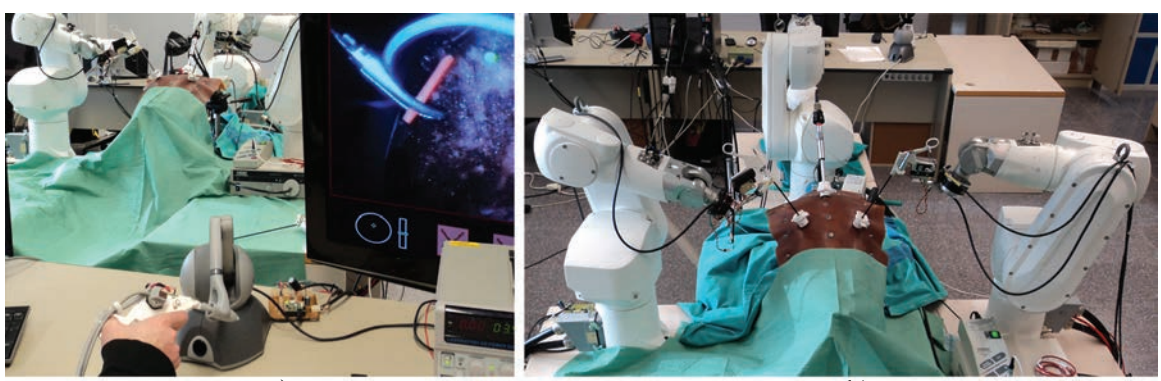


Figure 5.5: a) $ttransf*$, b) NTT and c) R_{Sel} Orientation indices evolution during task execution



a) b)

Figure 5.6: Set-up with two 6 DoF robots and the laparoscopic view.

into the ring. This experiment uses the simulated environment.

Fig.5.7.a shows a general view of the simulated workspace with the orientation cones and the new obstacle. Fig.5.7.b shows a detailed view of the workspace with the two rings obstacle and the required trajectory.

5.3 Experiment 2. MRCP performance analysis

The following experiment has been designed to evaluate the potential operator improvements using MRCP with respect to StdT in telemanipulation tasks. The simula-

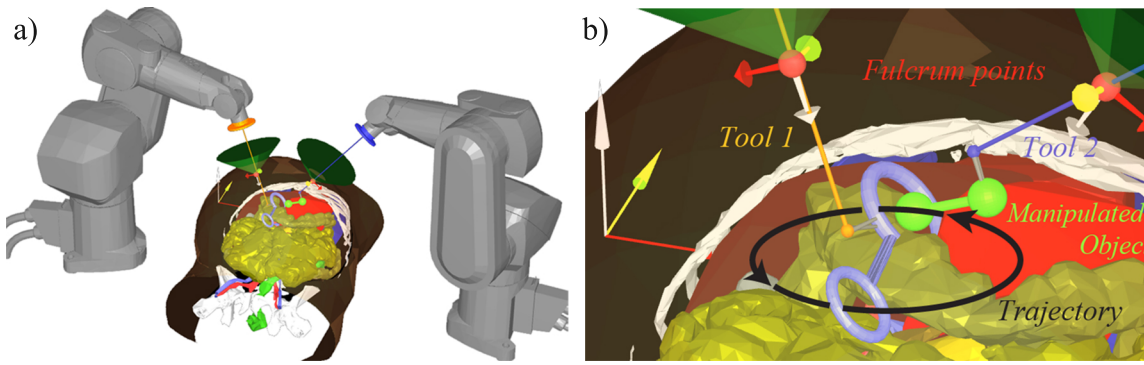


Figure 5.7: Set-up with two 6 DoF robots and the laparoscopic view.

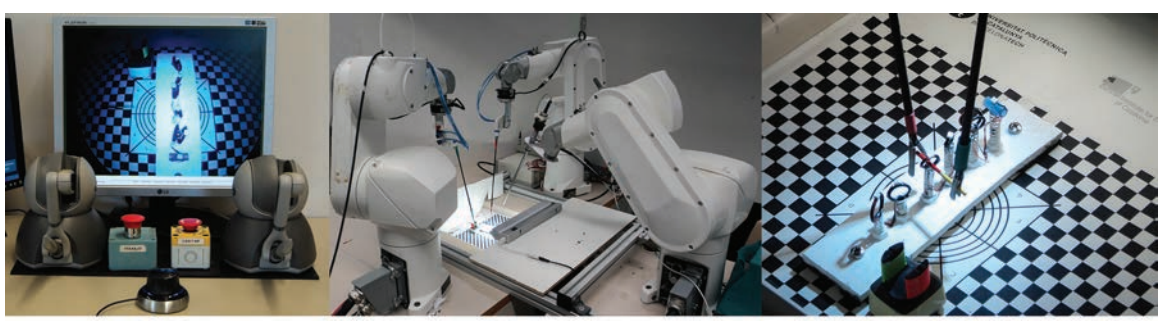
tion of a suture has been selected to assist the operator in the execution of a dexterous task. Three teleoperation modes are used: first, a bi-manual StdT (2M:2S), in which each master controls a slave robot. Second, MRCP in which the operator guides the object with a single master (1M:2S). Finally, and using the remaining free hand when operating in the MCPR mode, a second teleoperated system is included to control the remote workspace field of view, MRCP+Cam. A robot holding the laparoscopic camera is guided by a 6 DoF mouse as master device (1M:1S). All experiments results are compared in terms of execution time, ET, economy of movement (measured as the trajectory described by the telemanipulated object), EM, and quality (measured as the amount of target deformation produced by undesired collisions), Q.

Task

The task simulates a laparoscopic suture: the operator telemanipulates a needle by means of two laparoscopic tools which are set as robot end effectors. The operator must guide a needle through six deformable targets in a concrete order. The experiment demands high precision: correct position and orientation of the needle in front of the targets and accurate trajectory inside them to minimize undesired contacts. It also imposes a task transfer (needle transfer) at every suturing stitch.

Experimental Set-up

The set-up is composed of a master station with two Phantom Omni, a 6 DoF mouse and a screen showing the laparoscopic view, Fig.5.8.a. The operator screen does not offer any additional information. There is no information about the robot executing the task to preserve the transparency of MRCP. The slave station is composed of three 6 DoF robots (Staübli RX60B). Two of them are equipped with an Endo Grasp as tool and the third one holds a $0deg$ laparoscopic camera. (10mm 0deg Hopkins II Carl Zeiss), Fig.5.8.b. To monitor the position of the rings, an infrared led is attached to the top of each ring, and a camera with an infrared filter is used to track the movements. The rings are held up with a spring, which deforms when the needle contacts them. The led is used to control the quality of execution (measuring the trajectory of the led) and to compute the collision risk index in real time. Six targets, in the form of deformable rings (four single rings, one double ring and a small tube), all of different diameters, heights and orientations are placed in the workspace, as shown in Fig.5.8.c. The schematic representation of StdT and MRCP systems are shown in Fig.5.9 and Fig.5.10 respectively.



a) b) c)

Figure 5.8: a) Master console, b) Remote workspace with three robots, and c) the scenario with the rings and the needle(right).

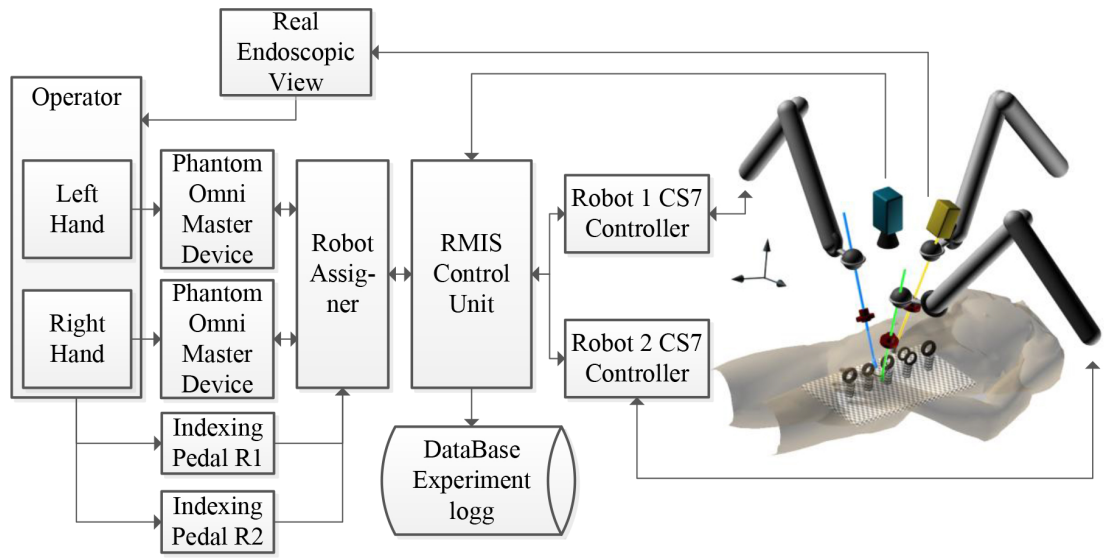


Figure 5.9: Block schema of StdT setup

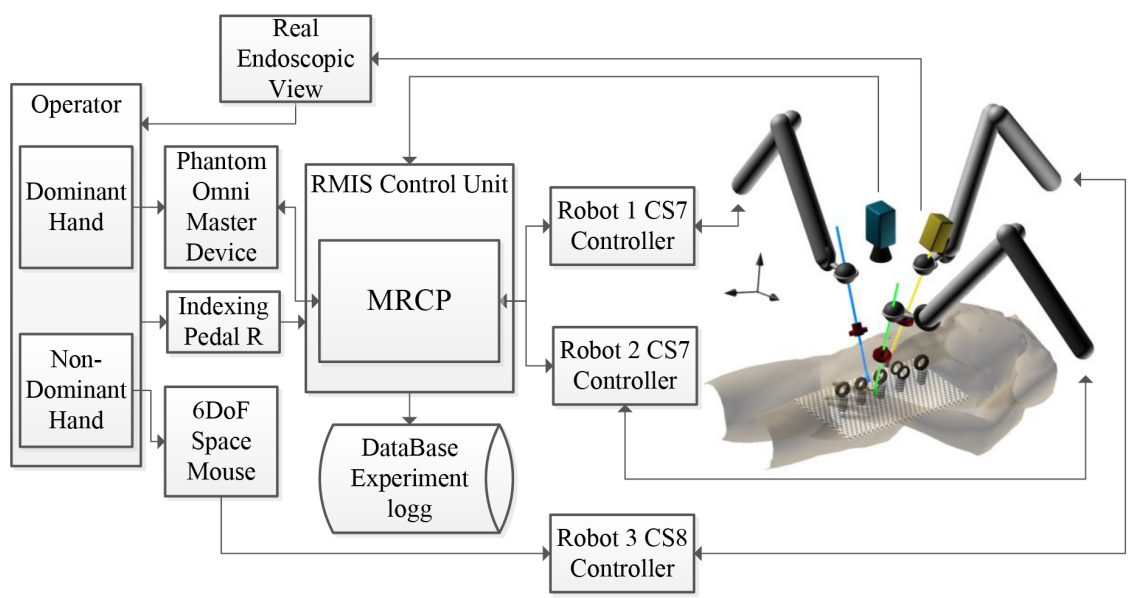


Figure 5.10: Block schema of MRCP setup.

Measures and Evaluation Methodology

The performance of MRCP is measured in terms of execution time, ET, economy of movement, EM, and quality of execution, Q. ET measures the trial completion time.

EM measures the needle's trajectory length. Q, determined from the trajectories described by the leds attached to the target points (leds pose vary due to collisions with the needle or with the laparoscopic tools), evaluates the execution quality. The target movements, caused by the interaction with the tools, emulate tissue deformation.

During the task execution all relevant data is stored in a file for its further analysis. The master, slave and targets poses (leds attached to them), the time stamp, the grasper status (open/closed), the RobSel (in MRCP and MRCP+Cam experiments) and the pose of the laparoscopic camera (in MRCP+Cam) are sampled and stored every 18ms (minimum communication cycle time imposed by the robots controllers).

The experiment studies both, the objective results obtained with the recorded data from all the trials and the subjective evaluation resulting from a questionnaire filled by the subjects at the end of the experiment. The objective study is based on a statistical analysis of the obtained ET, EM and Q results. The subjective evaluation consists in the comparison between StdT vs MRCP and MRCP vs MRCP+Cam in terms of generic usefulness and expected results (subjective expected performance in ET, EM and Q). The study also considers the comfortableness of the proposed task-oriented teleoperation and the achieved level of abstraction when using MRCP and MRCP+Cam (if the subject feels that is guiding the needle instead of the robots). The subjects had a fixed range of possible replies: Absolutely Not, Not, Just a Few, Yes, Absolutely Yes.

Experiment Sample and Experimental Methodology

The experiment sample is composed of 20 subjects (15 males and 5 females) between 20 and 57 years old (with a mean of 36). The participants had different teleoperation experience (from null to experienced subjects) providing a full spectrum to compare the performance of MRCP within the whole range of expertise. Vision and gender are other factors considered. The characterization of the experiment population is shown in Fig.5.11.

The experiment is divided into four parts: first the subjects received an introductory explanation of the system and the task to be executed. Second, they were invited

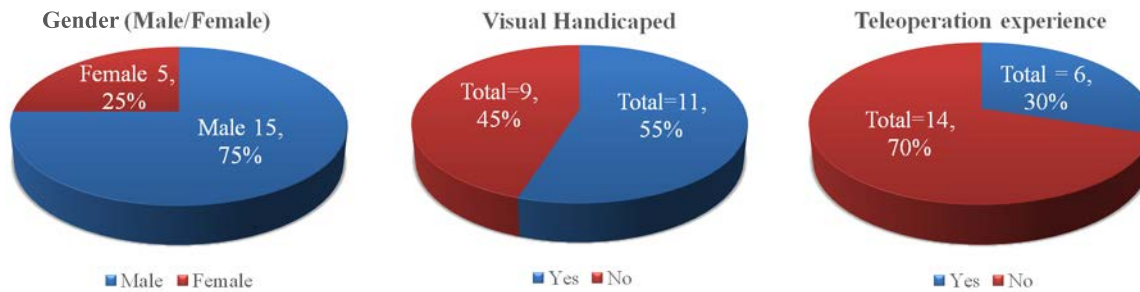


Figure 5.11: Characteristics of the experiment population: gender, visual handicapped and teleoperation experience..

to test all three teleoperation modes with the supervision of an expert. In this phase, all subjects executed a trial of each teleoperation mode, getting familiarized with the system and the task. After this initial part, the subjects had to execute six consecutive trials of each teleoperation mode (bi-manual StdT, MRCP and MRCP+Cam). The order of the modes was randomized to cancel undesired effects of the learning process and fatigue. Finally, they answered a set of questions evaluating their experience and subjective evaluation with each proposed teleoperation modality. In every trial, the initial conditions were identical: all three robots on an initial position, the master devices on their holders, the pedals not pressed and the needle on a fixed position. Setting the initial configuration lasts approximately one minute, time used by the subject to relax. Between each teleoperation mode, the subject had a pause of up to five minutes, if desired, to relax. In case the experiment controller appreciated any symptom of fatigue or lose of concentration, subjects were invited to use these relaxing periods. No noticeable stress or fatigue was detected. Even more, most of the subjects, during the experiment (normally more than one hour including initial explanations) demonstrated a competitive attitude and were interested about their results and learning curves. They also proposed test improvements or comments about the set-up or MRCP.

Experimental results

Statistical analyses rely on the improvements of the three measured aspects: ET, EM and Q . The analyses isolate and identify the effects of MRCP and MRCP+Cam with respect to StdT. The statistical analysis of the three evaluation indices shows a significant improvement between the proposed teleoperation methods (MRCP and MRCP+Cam) compared to StdT. There are no evidences of significant differences between MRCP and MRCP+Cam. The mean and standard deviation of all subjects in all three modes is shown in Table 5.1.

Table 5.1: Mean and standard deviation for all subjects in all trials. (▲: Best, ♦: Medium and ▼: Worst result.

		StdT	MRCP	MRCP+Cam
ET	Mean	142,51 ▼	101,13 ▲	105,29 ♦
	std	46,97 ▼	42,31 ♦	41,16 ▲
EM	Mean	2580,13 ▼	2195,05 ♦	2185,89 ▲
	std	505,05 ▼	373,37 ♦	308,56 ▲
Q	Mean	8477,62 ▼	2849,91 ▲	3060,19 ♦
	Std	4218,93 ▼	1463,68 ▲	1689,73 ♦

Reviewing the evaluation criteria between StdT and MRCP or MRCP+Cam, several conclusions can be extracted: ET benefits from the cancellation of failed task transfers and from the confidence on the system that allows the execution of faster actions (the subject moves a single object in the remote space in MRCP whereas in StdT the subject controls two different robots). EM shows shorter trajectories for the same reasons: MRCP avoids failed task transfers and erratic trajectories at the same time that minimizes robot movements. The amount of collisions has decreased (increasing Q) due to diverse factors: MRCP dynamically controls the position of the targets (using vision based techniques to detect them) eliminating the collisions of the robot that does not hold the needle, the task transfer are faster and safer, decreasing

the risk of collision and the tension of the needle due to misalignments between robot tool tips.

The results of MRCP and MRCP+Cam do not indicate a noticeable difference. The benefits of positioning the camera are compensated by: First, the subjects had to learn and manage a second teleoperation system with a different master device. The difficulty increases when a second master device should be managed simultaneously. Second, subjects should learn how to determine the best camera position at each step of the teleoperation (how to focus the entry point, how to obtain a general view when changing from one target to another, etc). The amount of proposed trials is clearly insufficient to reach a sufficient learning process of the combined MRCP+Cam system. Further trials will be proposed to the subjects to study the learning curves evolution.

More compact interquartile ranges are obtained in MRCP and MRCP+Cam in all three evaluation criteria (ET, EM and Q), as shown in Fig. 5.12, demonstrating that the proposed task-oriented teleoperation improvements are independent from the teleoperation experience (benefits are transversal for the whole population sample).

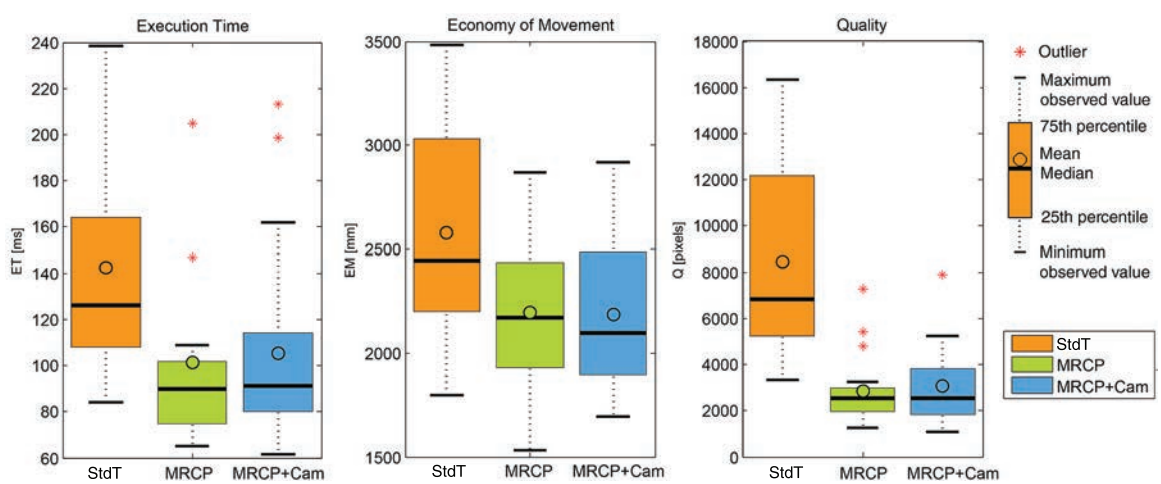


Figure 5.12: Mean, median, 25th and 75th quartiles, maximum, minimum and outliers observed in StdT, MRCP and MRCP+Cam.

The resulting learning curves, Fig. 5.13, show an improvement in all three teleoperation methods during the sequence of trials. The evolution of the subjects does not show a steeper learning curve in any of the tested teleoperation modes or in any

evaluation parameter. Extrapolating the learning curve suggests that no intersection between StdT and MRCP or MRCP+Cam occurs. The ET and EM learning curves need more trials to reach the asymptotic phase, whereas the Q learning curve stabilizes at the 5th trial. The execution of more trials could generate undesired fatigue due to the required time to finish the experiment (more than one hour per subject including all four phases of the experiment).

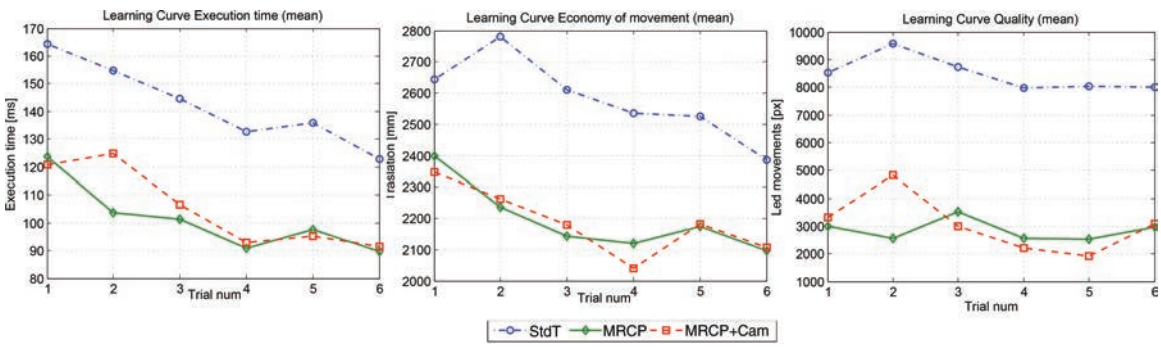


Figure 5.13: Learning curves of ET, EM and Q during the six trials for all three teleoperation modes.

In Table 5.2 several statistics are presented comparing the results of paired tests between the three teleoperation modes (StdT vs MRCP, StdT vs MRCP+Cam and MRCP vs MRCP+Cam). Statistical signification is validated by the paired sample Student's t-test applied to ET, EM and Q. There is a significant difference between StdT and MRCP or MRCP+Cam, but not between MRCP and MRCP+Cam. The inclusion of the possibility of moving the camera did not generate noticeable results. The second statistic, power test, ensures the validity of the test (the obtained distribution functions are different), using 20 subjects, within a probability of 0.9. The first column of Table 5.2 shows the minimum amount of subjects to get the probability of 0.9 and the second column the probability obtained with the 20 subjects. The test demonstrates that the difference between StdT and MRCP or MRCP+Cam is noticeable, generating two separate distributions. Again, this test does not allow to generate remarkable distinction between MRCP and MRCP+Cam. The last column shows the gains based on the mean results of each evaluation index for each teleoperation mode. Again, there is a significant improvement between StdT and MRCP or

MRCP+Cam teleoperation modes. The highest benefits are obtained in Q: gains >2. There is no significant gain between MRCP and MRCP+Cam, presenting MRCP a slight improvement with respect to MRCP+Cam in ET and Q.

Table 5.2: Paired t-tests and Power Test

	Type of test		Paired t-test		Power Test		Gains
	Test 0	Test 1	value	P	Min Sub (P=0,9)	P(N=20)	Test0/ Test1
ET	StdT	MRCP	6,69	2,14E-06	17	0,96	1,41
	Stdt	MRCP+Cam	6,43	3,60E-02	20	0,91	1,35
	MRCP	MRCP+Cam	-0.84	0,41	1040	0,07	0,96
EM	StdT	MRCP	5,76	1,48E-01	19	0,90	1,18
	StdT	MRCP+Cam	5,95	9,85E-02	18	0,91	1,18
	MRCP	MRCP+Cam	0,14	0,89	17388	0,05	1,00
Q	StdT	MRCP	7,49	4,39E-03	9	1,00	2,97
	StdT	MRCP+Cam	5,68	1,79E-01	9	1,00	2,77
	MRCP	MRCP+Cam	-49,00	0.62	584	0,09	0,93

The following analysis studies the obtained gains with the proposed task oriented paradigm (MRCP and MRCP+Cam) with respect to StdT mode. The analysis is divided into the three evaluation criteria ET, EM and Q. Concerning ET, only a single subject (6th subject) decreases the performance using MRCP with respect to StdT, and all subjects obtain better results with MRCP+Cam than with StdT. The ET obtained gains (StdT/MRCP) are the smallest of all three evaluation criteria. Fig.5.14 shows the results of ET using MRCP and MRCP+Cam. In conclusion, task oriented teleoperation clearly improves performance with respect to classical teleoperation, but, no benefits are proven when comparing MRCP and MRCP+Cam. The longest learning curve required for MRCP+Cam could influence in these results.

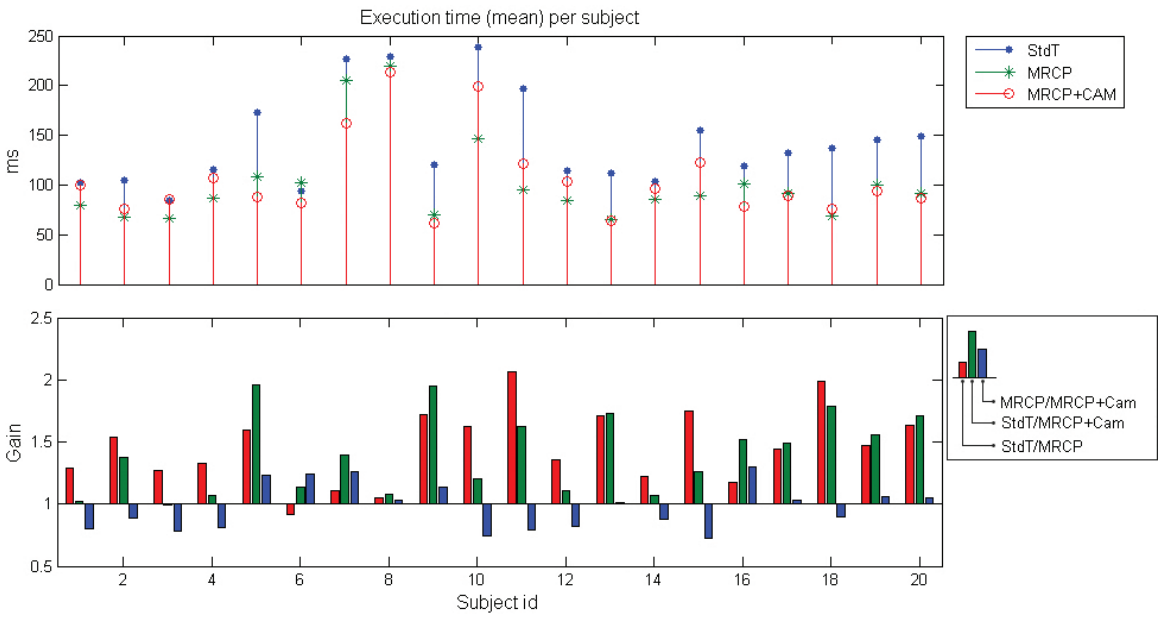


Figure 5.14: Mean and gains of Execution Time per subject.

The obtained gains of EM in MRCP and MRCP+Cam with respect to StdT are evident, as shown in Fig.5.15: all the subjects obtained significant gains. Again, there is not a common behaviour when comparing MRCP and MRCP+Cam. 50% of the subjects obtain better results using MRCP, 45% with MRCP+Cam and only one subject obtained the same results with both approaches. The differences between MRCP and MRCP+Cam are notably lower than the ones observed between StdT and MRCP or MRCP+Cam.

Finally, the quality of execution of all subjects is better with MRCP and MRCP+Cam than with StdT. Numerically, the obtained gains of Q are the highest of all three evaluation criteria. Quality criteria is the most important aspect when teleoperating in high precision tasks (e.g. surgery). These results verify the expected benefits of the task oriented paradigm against StdT. Again, there is not a common behaviour of the results between MRCP and MRCP+Cam: 55% of the subjects executed the task with better quality using MRCP+Cam in front of the 45% that performed better using MRCP. Subjects 11 and 14 presented noticeable worst results with MRCP+Cam. A deeper study will allow concluding whether they can be considered as outliers or aspecial group that require a different teleoperation solution. Fig.5.16 shows the quality

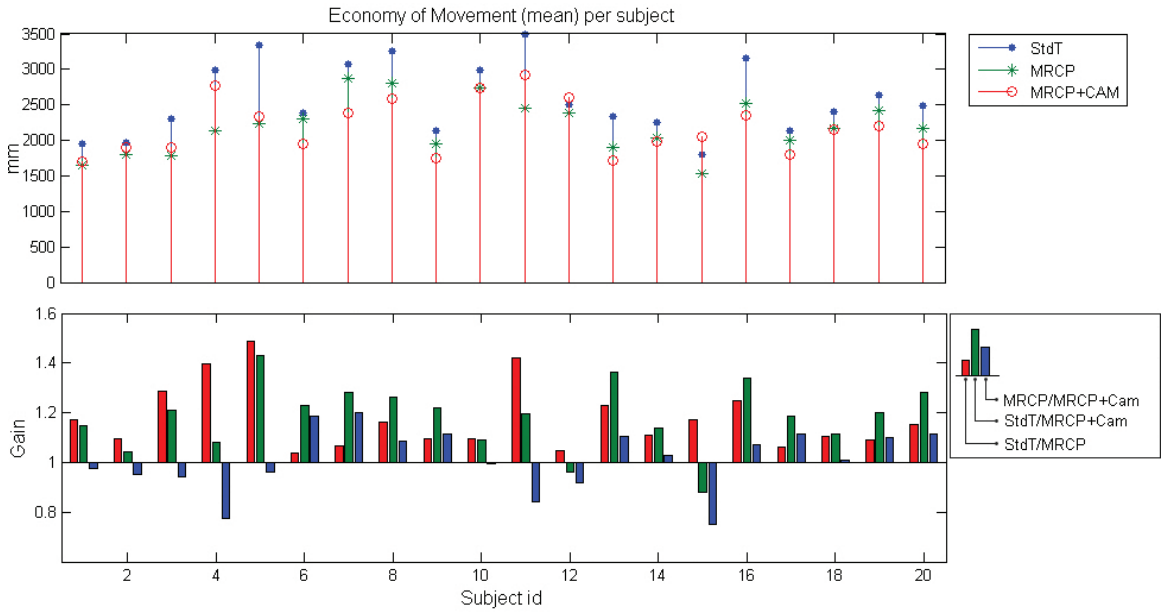


Figure 5.15: Mean and gains of Economy of Movement per subject.

means and gains obtained by all the subjects. Fig.5.17 classifies the gains into three groups: $> 1.05, [1.05, 0.95], < 0.95$ and shows their percentage. The range $[1.05, 0.95]$ can be considered as null gain: no numerical difference is observed.

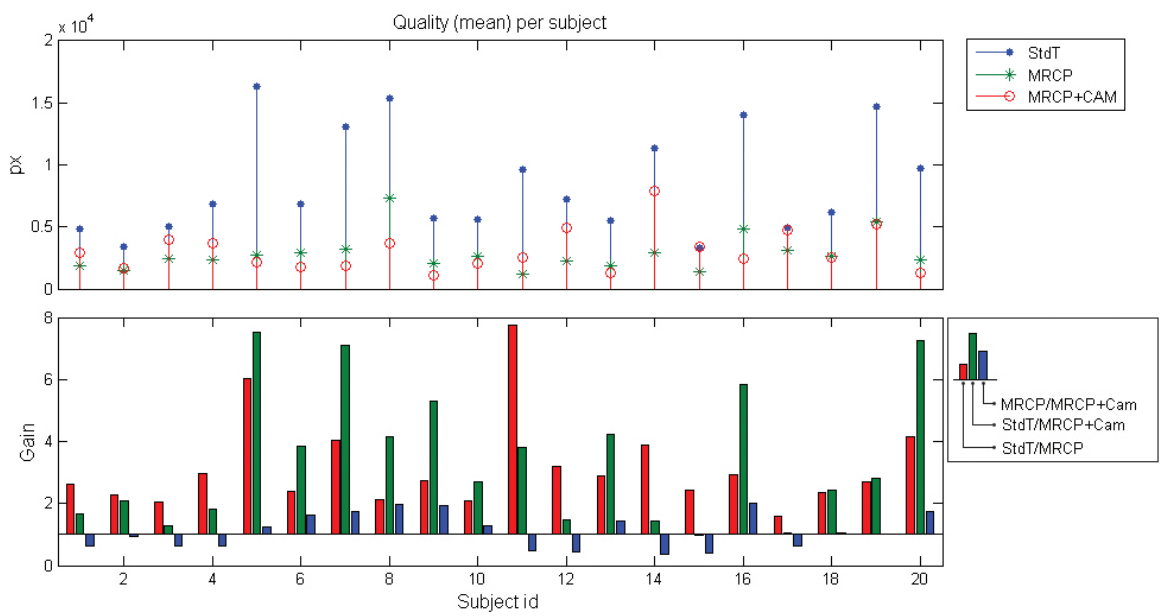


Figure 5.16: Mean and gains of Quality per subject.

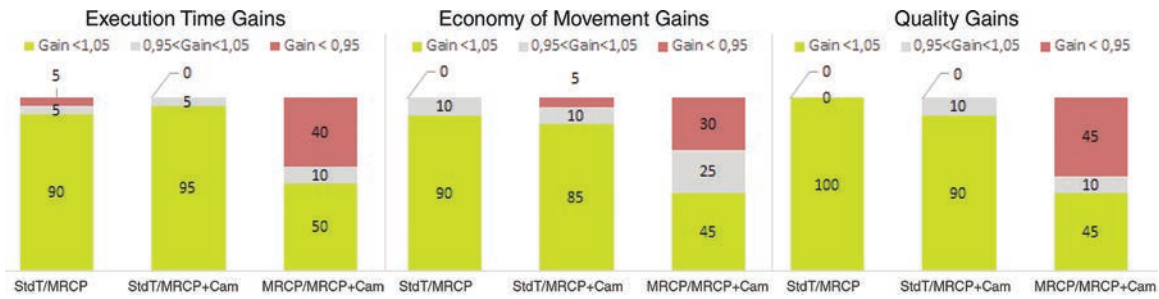


Figure 5.17: Percentage of the samples in each gain group, 1.05, $[1.05, 0.95]$, < 0.95.

In conclusion, it is well proven that task-oriented teleoperation has generated noticeable benefits with respect to StdT in the proposed task (dexterous telemanipulation task). The benefits of guiding the camera do not appear as a common characteristic for all subjects. The need of learning how to operate another master device (camera guidance) and to obtain the desired remote workspace visualization with the camera still needs more trials to determine the usefulness of this proposal.

Subjective Analysis

The subjective analysis compares the proposed methods MRCP vs StdT and MRCP+Cam vs MRCP in terms of perceived general improvement of the proposed task-oriented teleoperation (MRCP and MRCP+Cam) and the expected improvements obtained in ET, EM and Q. This analysis is also designed to measure the degree of comfortability of task-oriented teleoperation (MRCP and MRCP+Cam) and the degree of abstraction achieved (operator executes the task).

Fig.5.18 shows the result of the subjective analysis. Concerning MRCP vs StdT, the subjects evaluate MRCP as a useful teleoperation paradigm: answers Absolutely Yes or Yes sums the 80% in usefulness, 70% in ET, 75% in EM and 60% in Q. The unanimity in the degree of satisfaction is not present when evaluating MRCP+Cam vs MRCP. Answers Absolutely Yes or Yes represent the 70% in usefulness and 80% in Q, but just the 40% in ET and 45% in EM. These results are consistent with the expectations of MRCP+Cam: it helps to obtain the desired view to perform a higher quality task execution. On the contrary, camera positioning increases execution time,

especially in subjects with less teleoperation experience.

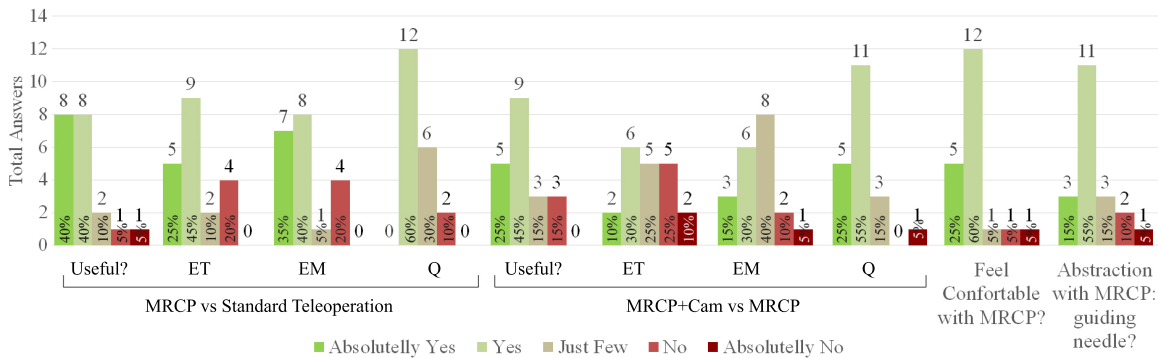


Figure 5.18: Results (absolute and percentage) of the subjective evaluation in the last phase of the experiment

Apart from the comparative analysis, the subjects had to evaluate the comfortability achieved with the task-oriented teleoperation paradigm and the level of abstraction. The acceptance of the system by the subjects was high (Absolutely Yes or Yes represent the 75% in comfortability and 70% in abstraction). These answers reinforced the idea behind MRCP: the task-oriented teleoperation. Table5.3 shows the statistics of the subjective analysis.

Table 5.3: Mean and Standard Deviation of the subjective analysis

	MRCP vs StdT				MRCP+Cam vs MRCP				Task oriented	
Question	Useful	ET	EM	Q	Useful	ET	EM	Q	Comfort.	Abstract.
mean:	4	3,75	3,9	3,5	3,8	3,05	3,4	3,95	3,9	3,7
std:	1,08	1,07	1,12	0,69	1,01	1,19	1,05	0,94	0,97	1,08

The analysis based on Spearman correlation between subjective and objective analysis using the five subjective evaluation levels do not provide a satisfactory result because the perception of gain from each subject is different.

Conclusions

The obtained results demonstrate the usefulness of the MRCP proposal. The results obtained with StdT vs MRCP and StdT vs MRCP+Cam evidence the performance improvements. The subjective analysis also shows that subjects appreciate the new teleoperation paradigm. On the contrary, the comparison between the obtained results of MRCP and MRCP+Cam did not present the expected results. The benefits of moving the camera as desired, even if proven in real MIS and RMIS, procedures, did not present beneficial evidences. Two reasons arises: on one hand, the need of longer learning process to control the task with the dominant hand and move the camera with the non dominant one. The need of focusing the attention on two separated processes decreases the global subjects performance. On the other hand, and also affecting the MRCP results, the insufficient grasping force presented by the laparoscopic tools used in the experiment, decreases the abstraction of executing the task. The subjects must reposition the needle after undesired needle-target contacts. Some task transfers failed due to incorrect needle grasping, decreasing the confidence of the subjects with the MRCP paradigm. When using the MRCP+Cam approach, the subjects did not feel safe if they did not have both tool tips in the camera field. Further experimentation should include new tools with better grasping capabilities and consider longer training processes.

The encouraging results open the possibility of executing new experiments including the use of Virtual Fixtures in task oriented teleoperation.

5.4 Experiment 3. Suture in a dynamic deformable surface

The experiment simulates a laparoscopic suture considering the natural deformation of the human organs and their movement (breathing or heart beating). The operator telemanipulates a needle by means of two laparoscopic tools which are set as robot end effectors. In the experiment the operator must guide a needle through four deformable

targets in a concrete order. These targets are placed onto a deformable surface that is continuously moving. The experiment has been designed to evaluate the potential operator improvements using MRCP with respect to a Standard Teleoperation, StdT, in precise telemanipulation of a needle in a dynamic and deformable remote workspace. Two teleoperation modes are used: bimanual StdT (2M:2S) and MRCP (1M:2S). Again, operators have the possibility to guide the camera with another master device.

Several differences can be pointed out with respect to the previous experiment: first, the workspace is dynamic (the targets are in continuous and periodic movement placed on a deformable tissue) and, second, the quality of execution is measured in terms of the interaction between the tools and the workspace: time in contact and forces and torques.

Experiment Set-up

The set-up is composed of a master station with two Phantom Omni, a 6 DoF mouse and a screen showing the laparoscopic view, Fig.5.19.a. The operator screen does not offer any additional information to preserve the transparency underlying in MRCP. The slave station is composed of three 6 DoF robots (Staübli RX60B). Two of them are equipped with an Endo Grasp as tool and the third one holds a laparoscopic camera. (10mm 0deg Hopkins II Carl Zeiss). A force sensor (Ati Gamma 6 DoF) measuring forces and torques is placed between the laparoscopic tool and the robot. The workspace, Fig.5.19.b, is composed of a deformable tissue with four rings on the external surface. The tissue deformation is produced by a servomotor attached to one of the lateral faces while the opposite one is fixed. With this disposition, the amount of deformation is variable for each target (the target closer to the motor executes the largest movement, while the one closer to the fixed face executes the shortest). Fig.5.19.c shows the tissue and targets. The schematic representation of bimanual StdT system is shown in Fig. 5.20 and the MRCP is shown in Fig.5.21.

Fig.5.22 illustrates the set-up for tissue deformation and the sequence of tissue deformation. The surface is fixed in one side and attached to a link with a servomotor, generating different amount of deformation in the targets (R1..4). The rings diameter

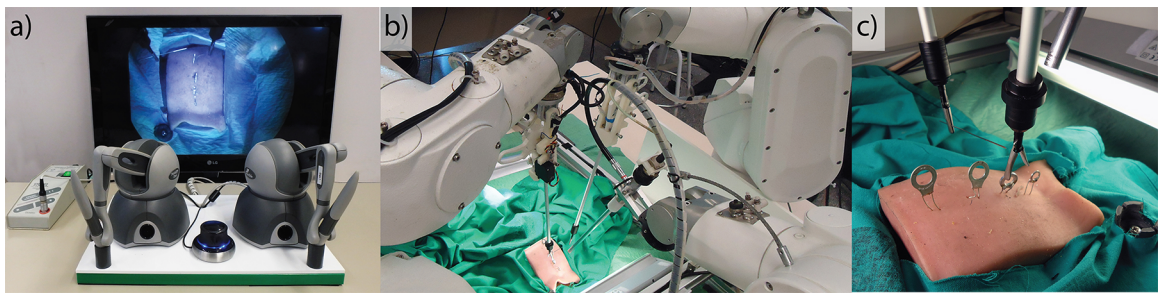


Figure 5.19: a) Master console, b) Remote workspace with three robots and c) the scenario with the targets and deformable surface.

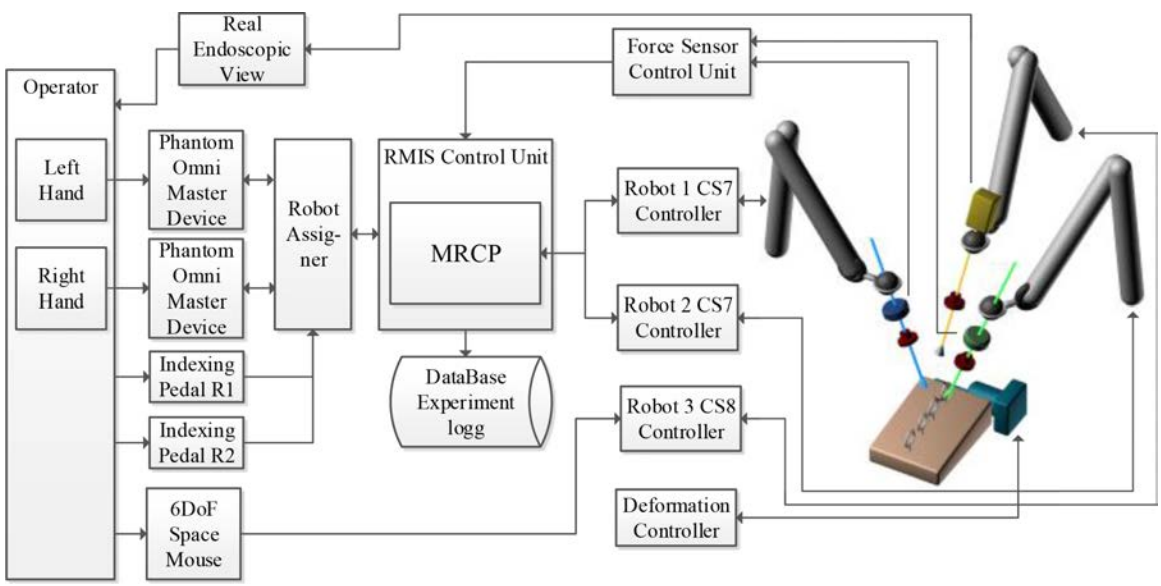


Figure 5.20: Schema of used task oriented teleoperation setup: StdT.

varies inversely proportional to the amount of movement. The sequence of servomotor movement ($0° \rightarrow 60° \rightarrow 30° \rightarrow 50° \rightarrow 0°$) simulates the natural tissue movement. There are four phases inside a complete cycle to avoid the operator to learn the movement.

Measures and Evaluation Methodology

The performance of MRCP is measured in terms of: ET, EM and Q. In this experiment, Q is measured in three different manners. First, the time in contact, which measures the accumulated time that the tools or the needle are in contact with the

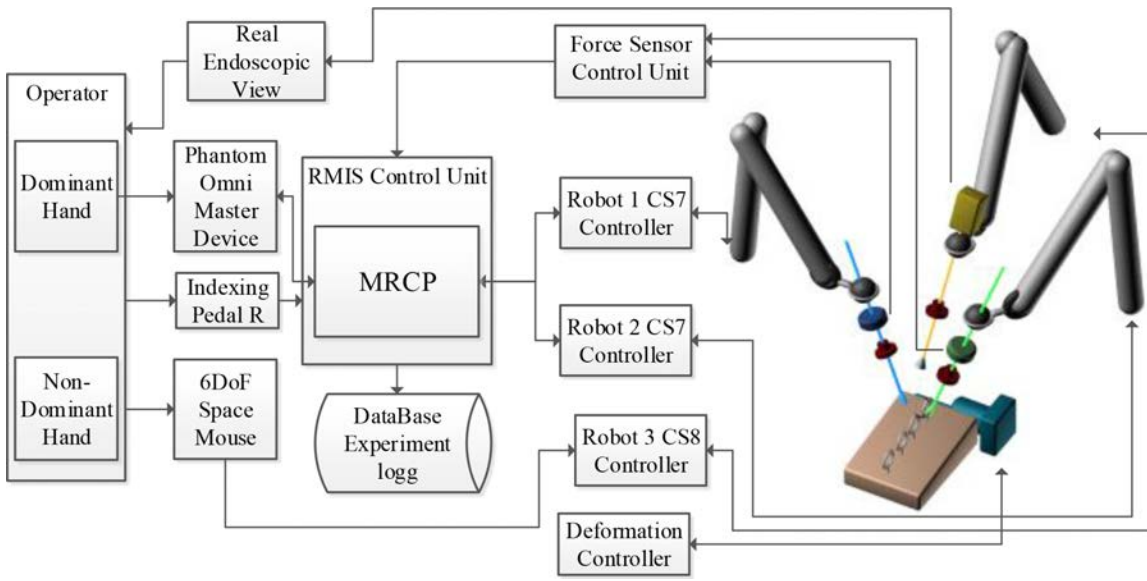


Figure 5.21: Schema of used task oriented teleoperation setup: MRCP + Cam.

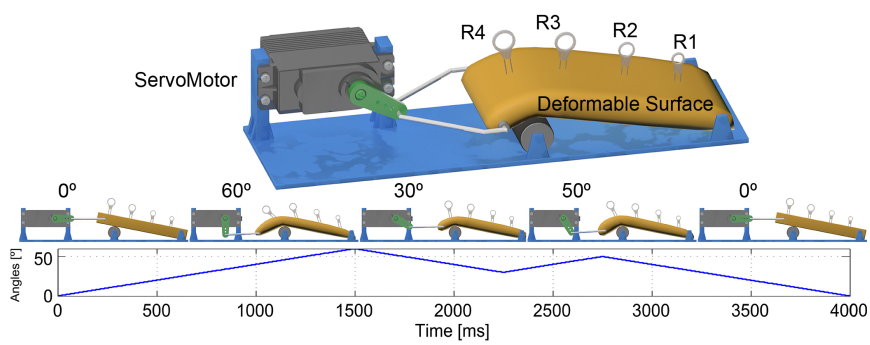


Figure 5.22: 3D model of the deformable surface.

targets. Second and third, the force and torque (accumulated) exerted by the tools into the workspace.

In addition, from the interaction between tools and workspace, the experiment studies the forces and torques produced by misalignments of the tools when transferring the needle. When a robot releases the needle, the forces and torques are transformed into a sudden movement that can damage the tissue. Fig.5.23.a shows an illustrative example of a misaligned needle transfer with the corresponding forces and torques, as well as the position of the needle with and without misalignment. Fig.5.23.b shows the needle transfer with MRCP, in which both tools are aligned and

no tensions are produced.

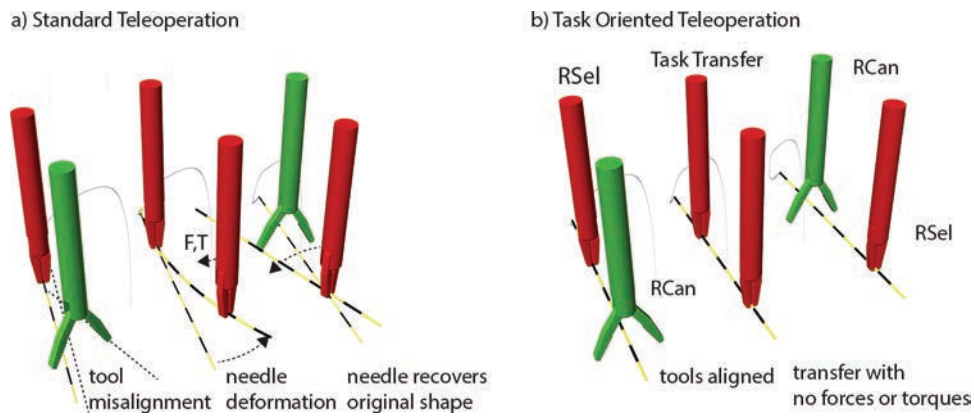


Figure 5.23: Example of needle deformation and the consequent forces and torques using StdT and MRCP teleoperation modes.

Every 18ms relevant data (master devices and slave robots pose, forces and torques of the two force sensors, position of the servomotor that deforms the tissue, etc) are sampled and stored for their further analysis.

The subjective evaluation is designed to determine the level of comfort offered by MRCP, if the subjects have the perception of improving (in ET, EM and quality) and the level of abstraction obtained with the task oriented paradigm.

Experiment Sample and Methodology

The experiment sample was composed of 19 subjects (17 males and 2 females) with ages between 22 and 41 years old (mean of 32.5). None of them presented any important visual handicap (9 need visual correction). Concerning the expertise, 11 subjects had previously used (at different levels) the same or a similar teleoperated system, while 8 of them were inexperienced. 14 subjects had a technical research background and 5 came from other professional fields. The diversity on professional field and teleoperation expertise enables an independent study from the experience or background of the subjects. Fig.5.24 shows the most relevant population data.

The experiment was divided into three parts: an introduction followed by the trials

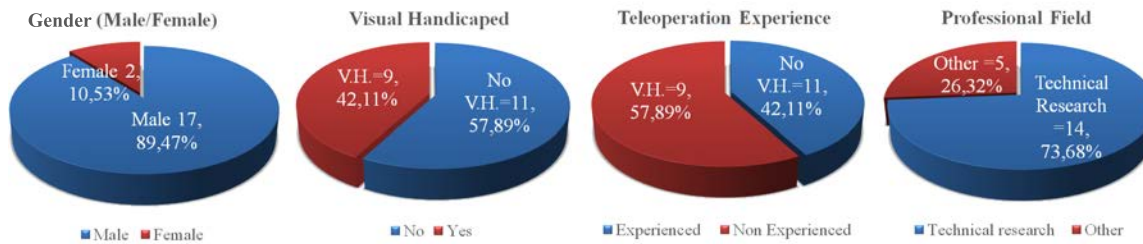


Figure 5.24: Characteristics of the experiment population: gender, visual handicap, teleoperation experience and professional field.

and finally the subjective evaluation. The experiment lasts between one hour and one hour and a half. During the whole experiment, the subjects could take a break for one or two minutes between each trial. None of the subjects verbalized fatigue, even more, some of them asked to continue executing trials to improve their performance.

In the first part, the introductory phase, they received information about the goals of the experiment, the set-up (master devices, slave robots, grasping tools, ...). They all could execute a non-recorded trial to understand the system and the task. After the initial phase, the subjects executed 6 consecutive trials of each teleoperation modality. The order of the teleoperation mode was randomized between subjects, cancelling learning influence or fatigue. At every trial, the initial conditions were the same: the robots on an initial configuration, one of them with the needle ready to be grasped and the tissue deformation system active. Finally, the subjects answered a questionnaire to study their opinions and subjective data and improvement suggestions.

Experimental results

This section reviews the obtained results, presents the statistical analysis of the recorded data from the experiment and the subjective analysis via a questionnaire.

The improvements of MRCP are analized with respect to bimanual StdT in terms of ET, EM and Q (patient safety: minimize contacts and interaction forces and torques). The statistical analysis of the recorded data points out a notable improvement in all evaluated indices: the mean of the observed results and the standard deviation are smaller when using MRCP. Mean, standard deviation for StdT and

MRCP and gains (StdT/MRCP) are shown in Table 5.4.

Table 5.4: Statistical analysis: Mean and standard deviation for all subjects in all trials

		StdT	MRCP	Gain
ET	Mean, Std	181,51 4879,74	113,20 703,89	1,60 6,93
EM	Mean, Std	2293,26 3,40E+05	1891,62 2,29E+05	1,21 1,48
Contact	Mean, Std	2012,61 4,48E+05	750,37 7,08E+04	2,68 6,32
Forces	Mean, Std	10740,27 1,45E+08	4314,74 1,12E+07	2,48 12,97
Torques	Mean, Std	1,19E+06 3,86E+11	6,12E+05 1,38E+11	1,95 2,80

Reviewing the statistical analysis of all evaluation criteria, all the statistics indicate improvements in the obtained means, medians and more compact percentiles. Table 5.4 shows means and standard deviations as well as the obtained gains in these statistics. Table 5.5 shows the results of the paired t-test and the double sided power test: the minimum number of subjects to obtain a P=0.9 and the probability obtained with the amount of subjects of the experiment (19 subjects). Finally, Fig.5.25 shows the obtained means, medians, 25th and 75th quartiles, maximum, minimum and outliers observed values for ET, EM, Contact, Forces and Torques.

Execution Time

Noticeable benefits are observed when using MRCP. A gain of 1.6 is obtained comparing the means of both approaches, as well as smaller variance and more compact quartiles. The power test demonstrates the validity of the difference with a probability of 0.98 (a P=0.9 is achieved with 11 subjects). The conclusion is a noticeable improvement in ET independently of the subject's expertise. MRCP accelerates the process offering automated task transfer and simplified teleoperation (operators only interact with a single device and do not have to control two robots). The number of required robot indexing has decreased: the operators have a free hand to be used to

Table 5.5: Paired t-tests and Power Test

	Paired t-test					Power Test	
	h	P	ci min	ci max	tstats	Min Subj (P=0.9)	P with 19 subjects
ET	1,00	0,00	43,05	93,59	5,68	11	0,98
EM	1,00	0,00	233,09	570,19	5,01	23	0,81
Contacts	1,00	0,00	949,95	1574,53	8,49	7	1
Forces	1,00	0,30	702,89	12148,17	2,36	25	0,59
Torques	1,00	0,00	2,49E+05	9,17E+05	3,66	13	0,97

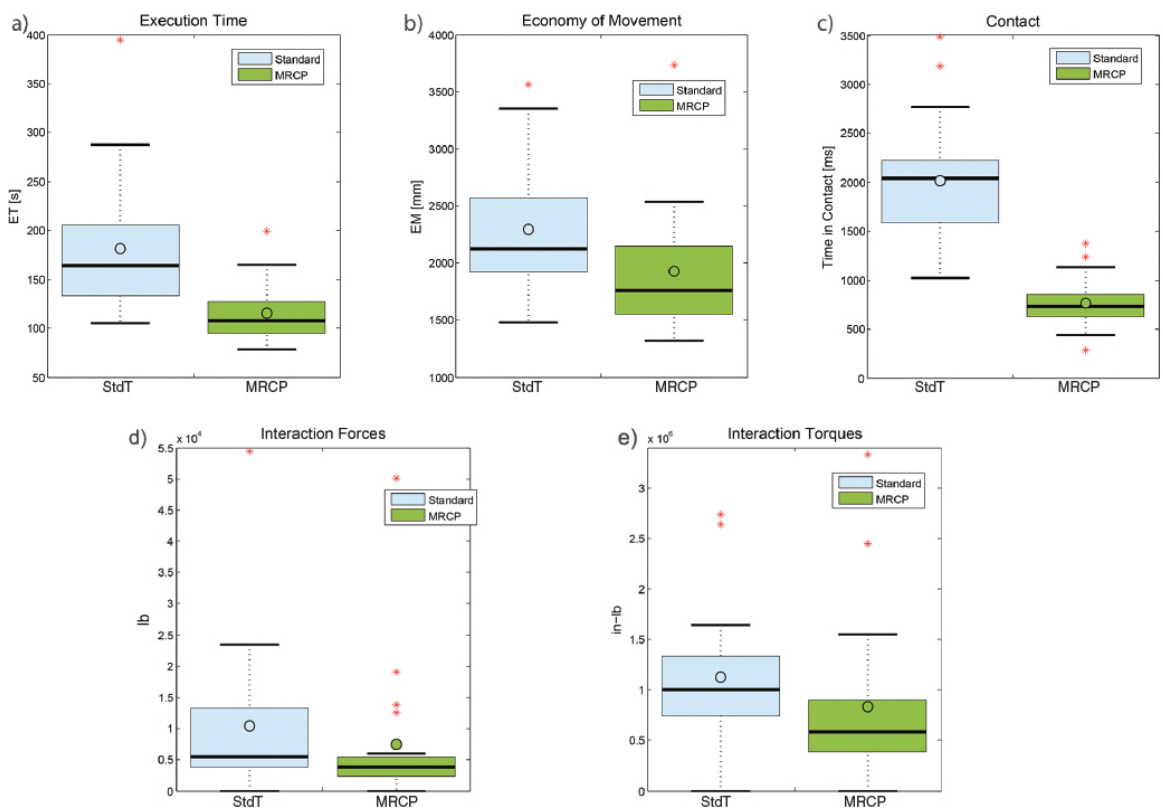


Figure 5.25: Statistics of evaluation criteria (∗ Outliers, ⊤ Maximum value, ⊓ 75%, ∘ Median,- Mean, ⊔ 25%, ⊥: Minimum value.

control the camera robot.

Economy of Movement

Statistical differences give an EM gain of 1.21. The validity of the hypothesis (there is a statistical significant benefit) is ensured with a P= 0.81 (4 more subjects are required to obtain a P=0.9). These results are influenced by an outlier in MRCP results. Mean and median are lower with MRCP, but the range of quartiles is similar in both experiments. Observing the tests execution, an improvement of the graspers is required to improve the needle subjection and, consequently, EM.

Contact

The contact index, which measures the time that the needle or the tools are in contact with the targets, measures the amount of time between the needle insertion and extraction (critical time). This term is important because it refers to one of the most delicate part of a suture (the needle and the tools are in contact with the tissues involving high risk of tearing tissues, affecting patient safety). A noticeable benefit has been obtained as the gain of the means indicates (Gain of 2.68). Paired t-test and power tests demonstrate the validity of the approach. The lose of depth perception is compensated with the automated teleoperation, generating faster and safer sutures.

Forces and Torques

Again, the obtained results indicate a noticeable improvement of the forces and torques applied over the tissue. In this case, the improvement means less forces and torques exerted by the tool, resulting in less tissue damage and increasing patient safety (higher teleoperation quality). Several causes generate the forces and torques: first, the tool misalignment during the needle transfer, second, undesired contacts between tools and workspace. Forces and torques means show improvements and high gains. Concerning forces, paired t-test and power test fail to ensure the null hypothesis. This is caused by the contribution of two important outliers in both approaches. The value of the

quartiles show the benefit but, the differential value indicates a similar dispersion of the obtained values. The following conclusion can be extracted: MRCP decreases the force exerted by each subject compared with the force exerted with StdT, but does not homogenize the results for all subjects. The relative force exerted by a subject in comparison with the rest of subjects is equivalent in both approaches. Concerning torques, the results show an important improvement of MRCP with respect to StdT: a noticeable mean of the gain and more compact quartiles and standard deviation. Some relevant outliers have been detected, increasing the obtained mean. The more compact quartiles with MRCP indicate the homogenization of all subjects, independently of their experience. The torques pass the t-test and the power tests, obtaining a P=0.97 with 19 subjects.

Learning Curves

Fig.5.26 shows the learning curves obtained during the 6 consecutive trials. There's an evident difference between both proposals. In the initial trials, the difference indicate that the benefits introduced by MRCP are independent from the expertise and are noticeable from the very first time that the system is used. Another aspect to be pointed out is that task-oriented teleoperation offers operators an intuitive interface between the master and the slaves, improving teleoperation results. During the execution of the trials, the subjects improve in all the measured aspects. Some more trials should be done to ensure the stabilization of the subject's results in StdT and determine the final improvement of MRCP. MRCP learning curves are close to the stabilization after the sixth trial. There must be pointed out the operators performance decrement during the third and fourth trial in StdT. The origin of this behaviour should indicate some relaxation or fatigue of the subjects during these trials.

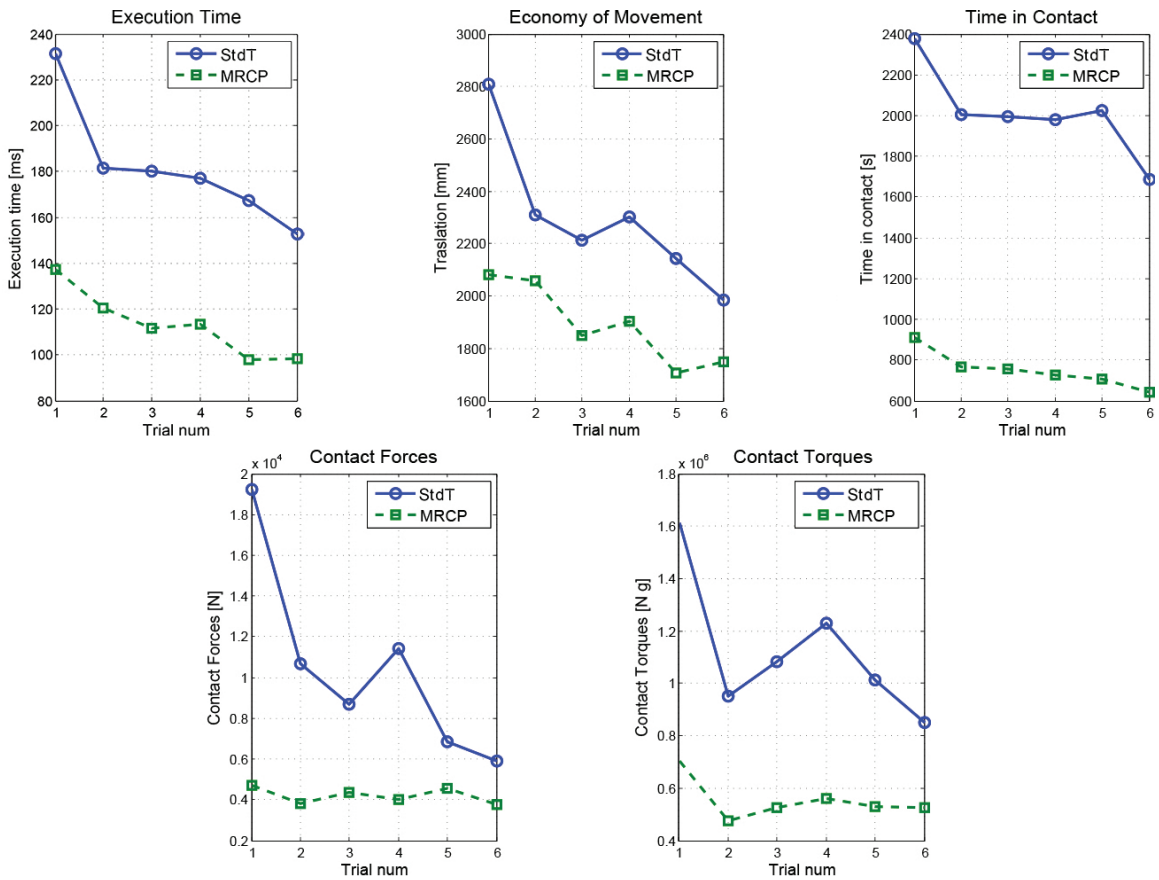


Figure 5.26: Learning curves for all evaluation indices along the six trials.

Robots Trajectories

Concerning robots trajectories, Fig.5.27 shows an illustrative example of the trajectory described by the robots (tool tip point) in StdT and MRCP teleoperation modes. Several conclusions can be extracted. First, the economy of movement is improved with MRCP, describing straightened trajectories. Second, in the task transfer regions, the difference of tool trajectories is evident: MRCP helps to ease the needle transference. Applying MRCP in RMIS, the patient safety increases: the needle describes shorter and straighten trajectories, specially in insertion and extraction phases. The risk of undesired tissue contacts decreases, reducing the number of failed needle grasping attempts. Forces and torques also decrease (clean insertions and extractions of the needle) respecting the integrity of the tissue (reduce the risk of tissue tearing).

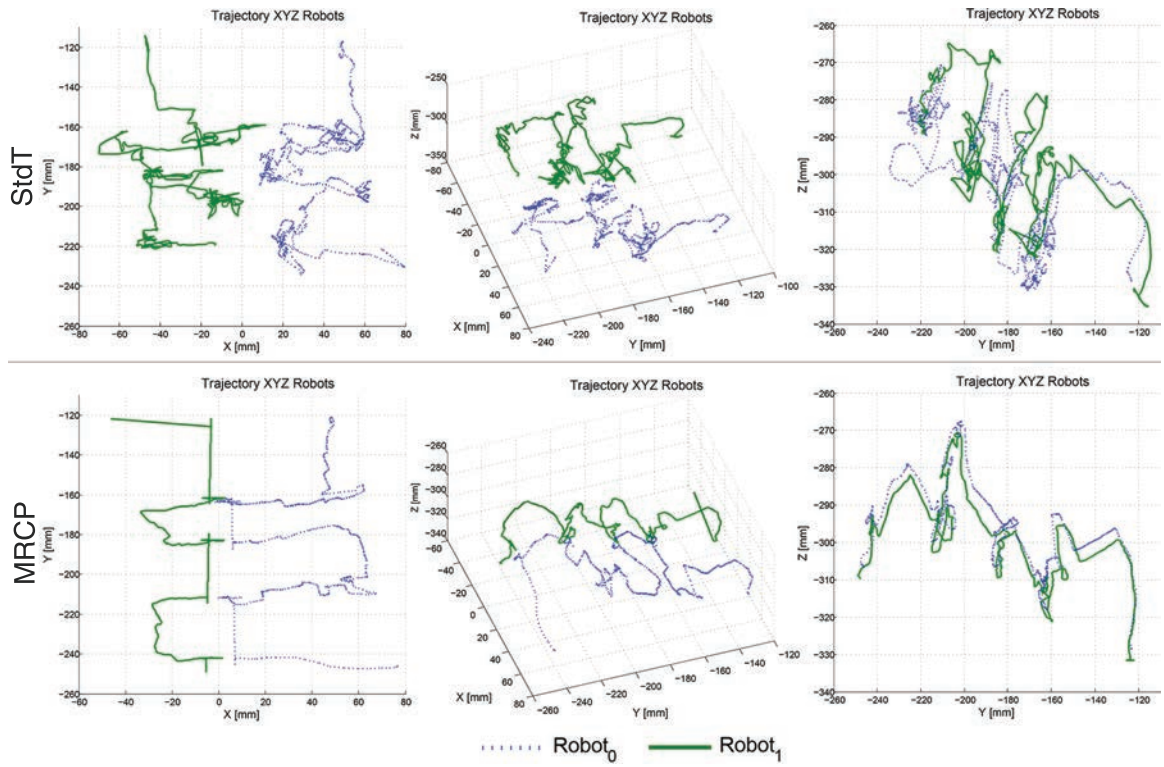


Figure 5.27: Schemas of task oriented teleoperation setup (MRCP + Cam).

Gains

To finish with the statistical analysis, the obtained gains (StdT vs MRCP) are reviewed.

Concerning ET, all subjects obtained a positive gain (less time with MRCP than with StdT). Two causes can justify this result: first, a reduction on the required time to execute a task transfer (tool positioning) and higher confidence on the system, executing straight and self confident movements.

Economy of movement is another benefit of using MRCP. Only three subjects (5.7%) did longer trajectories with MRCP than with StdT. The automatic needle transfer reduces the trials to obtain a satisfactory tools position to execute the transfer, reducing the trajectories. Tools positioning is critical when the targets are over a moving surface, obliging the operator to control and continuously reposition two tools at the same time according to the target trajectory. With MRCP, the accommodation

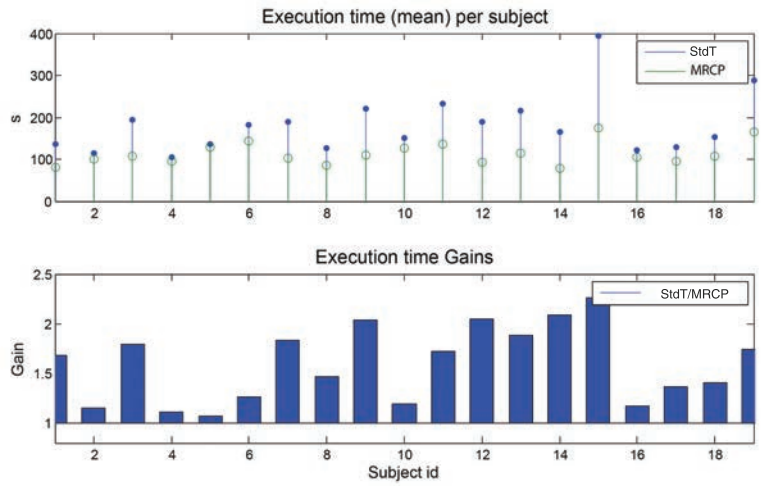


Figure 5.28: Obtained gains for ET and for every subject in StdT and MRCP.

of the robots to the desired needle position automates this process and releases to the internal control these positioning efforts.

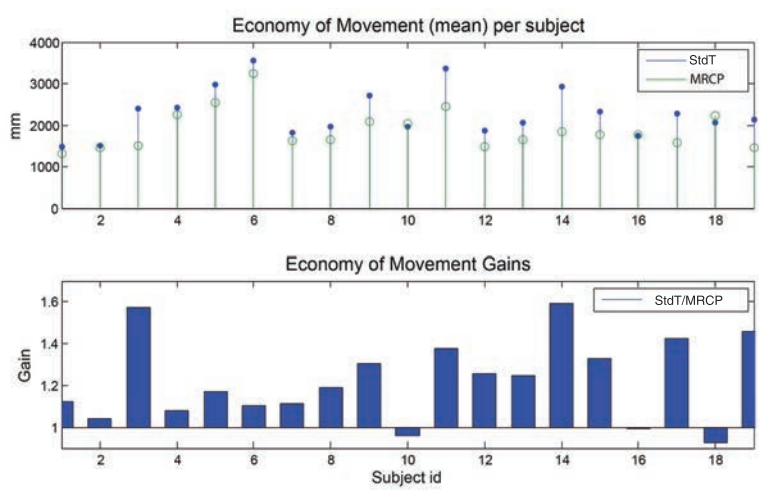


Figure 5.29: Obtained gains for EM and for every subject in StdT and MRCP.

Studying the EM, only three subjects (5.7%) did longer trajectories with MRCP than with StdT. The automatic needle transfer reduces the trials to obtain a satisfactory tools position to execute the transfer, reducing the trajectories. The tools positioning is critical when the targets are over a moving surface, obliging the operator to control and continuously reposition two tools at the same time according to

the target trajectory. With MRCP, the accommodation of the robots to the desired needle position is automatic.

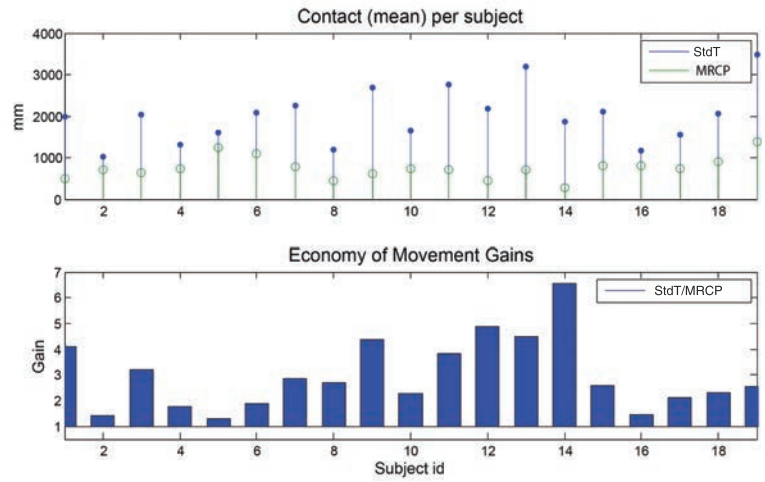


Figure 5.30: Obtained gains for Contacts and for every subject in StdT and MRCP.

The time in contact registered during the trials decreased with MRCP in all subjects. This shortens the critical insertion-extraction operations. The automatic task transfer and the task-oriented teleoperation contributed to this improvement.

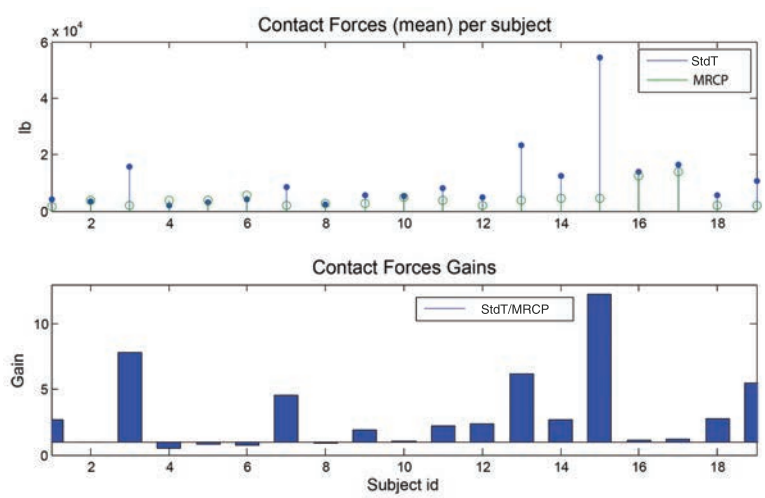


Figure 5.31: Obtained gains for Forces for every subject in StdT and MRCP.

The interaction between tools and tissue is one of the aspects to be observed in terms of patient safety. In RMIS there is no haptic feedback, consequently, tissue

damages produced by excessive pressure are an important aspect on patient safety. 14 out of 19 (73.7%) subjects improved (decreased) the amount of force exerted during the teleoperation with MRCP and 16 out of 19 (84%) improved the amount of exerted torque.

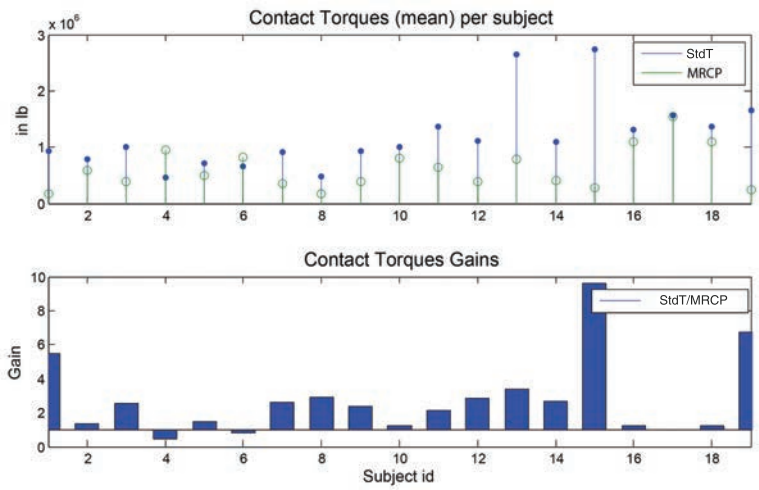


Figure 5.32: Obtained gains for Torques for every subject in StdT and MRCP

5.4.1 Subjective analysis

The subjective analysis studies the proposed task oriented teleoperation from the operators' point of view. The questionnaire is composed of seven questions related to the generic usefulness, the expected improvements (in terms of ET, EM, amount of collisions and tissue damages) of MRCP with respect to StdT and, finally about the comfortability and abstraction achieved with MRCP. The questionnaire is done at the end of all the trials. The possible answers are defined by Five point Likert scale: Absolutely Agree, Agree, Undecided, Neutral, Disagree and Absolutely Disagree.

MRCP usefulness

This question measures the general impression of the subjects with respect to MRCP. A 94'74% of the subjects had a positive opinion (57'89% Absolutelly Yes and 36,84%

Yes). Only one subject did not find MRCP useful. The subjects evaluated as positive several MRCP features: the use of a single master device, the automatic needle transfer and the possibility to position the camera with the remaining free hand simultaneously with the needle guidance.

Evaluation criteria

Four questions related again with EM, ET and Q (in terms of amount of contacts and tissue damage). 84'47% of the subjects agreed that they improve ET and 78'95% replied positively about the reduction of EM. Again, a 78'95% opined that MRCP helps to improve the quality of execution in terms of quantity of collisions and a 52.63% opined that they had produced less tissue damage.

Task Oriented teleoperation

The last two questions are related with the interaction between the operator and MRCP. The first studies the comfortability felt by the operators with MRCP. 94.7% of the subjects obtained a successfully interaction (felt comfortable). The second question analyses the abstraction obtained with MRCP. 73'68% (26.32% Absolutelly Yes and 47'43% Yes) of the subjects achieved a high level of abstraction during the teleoperation. This means that the subjects guide the needle and the robots become invisible. In other words, they could focus their efforts in the needle trajectory rather than to the robots.

5.4.2 Conclusions

Several aspects influenced in the positive results. The moving surface forces the subject to dynamically adapt the tool/needle position with respect to the targets. In StdT, the operator must position both tools according to moving targets whereas with MRCP the operator adapts the needle position with a single master device. This, jointly with the automatic task transfer, reduces the amount of forces and torques sensed on the tools, increasing Q (patient safety for the experiment).

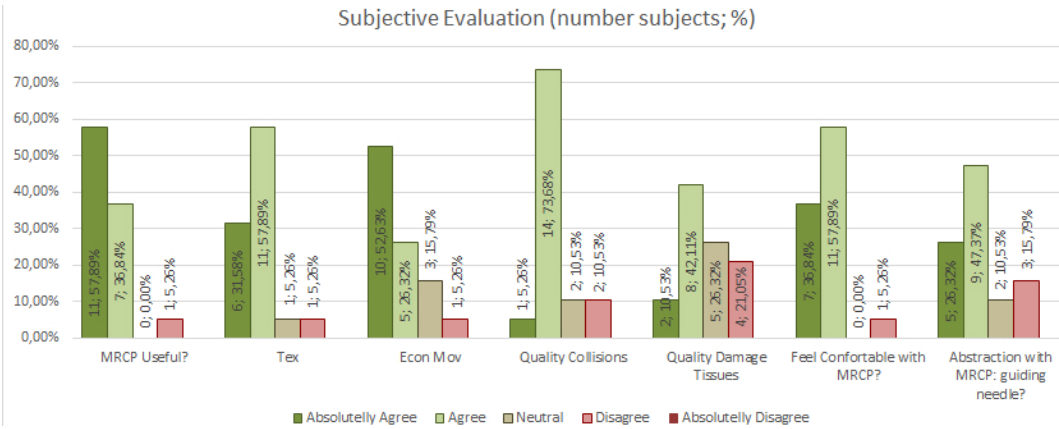


Figure 5.33: Schemas of task oriented teleoperation setup (MRCP+Cam).

The combined study of the objective and subjective analysis leads to several conclusions. First, the coincidence between the objective results and the perception of the subjects. The success of the proposed task oriented teleoperation requires a numerical improvement in all evaluation aspects, as well as a positive subjective analysis from the operators. The comparative analysis is consequent and robust enough to conclude that the proposed teleoperation paradigm is a feasible teleoperation modality than can be combined with StdT to improve dexterous teleoperation.

5.5 Experiment 4. Combined use of teleoperation modes

The previous experiments have been addressed to analyse the performance and differences between StdT and MRCP teleoperation paradigms. This last part of the experimental phase proposes the combined use of both paradigms to teleoperate complex tasks.

The approach is based on the division of complex tasks into a set of simpler sub-tasks which are executed in a certain sequence. Each sub-task presents the requirements and specifications that can be better executed by a certain teleoperation

paradigm. The control of the teleoperation set-up is based on a graph of states that, depending on the selected teleoperation paradigm or mode, assigns the M:S pairing, as shown in Fig.5.34. In bimanual StdT, each master controls a slave whereas in MRCP (task oriented paradigm) a master device controls the task and, optionally, the second master device can control an auxiliary robot (e.g. the laparoscopic camera).

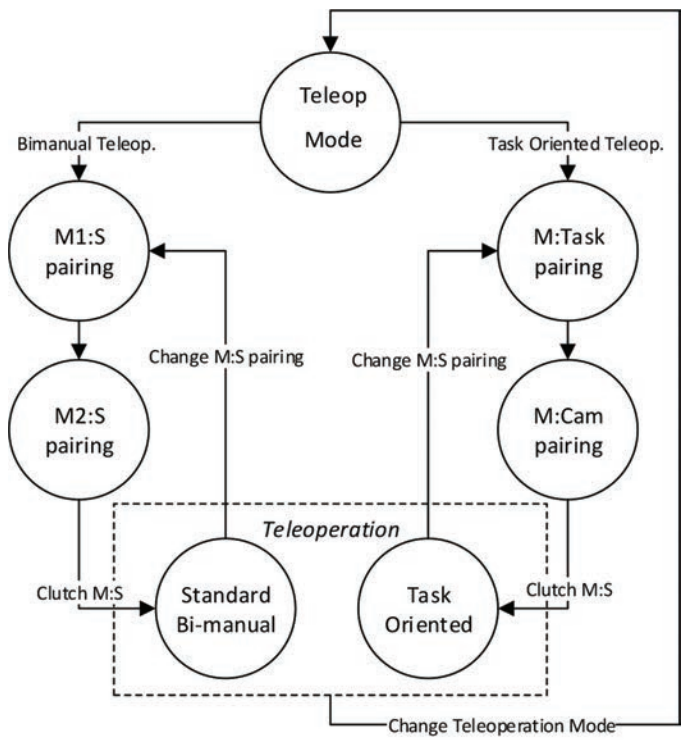


Figure 5.34: Graph of states of the combined teleoperation modes.

A continuous suture task in the field of RMIS is used as example of complex teleoperated task, [166]. This surgical procedure can be splitted into three main subtasks: pre-suture, stitching and knot tying. Each subtask can also be divided into several atomic subtasks and the optimal teleoperation mode can be determined. Following the suture example, Fig.5.35 shows a two level of subtasks division and the teleoperation paradigm that best fits with of them. Pre-Suture and Knot tying require two laparoscopic tools acting separately, consequently StdT is the appropriate teleoperation paradigm to be used. The stitching process is based on the guidance

of the needle, transferring it from one tool to another when the tissue blocks the laparoscopic tool that holds the needle.

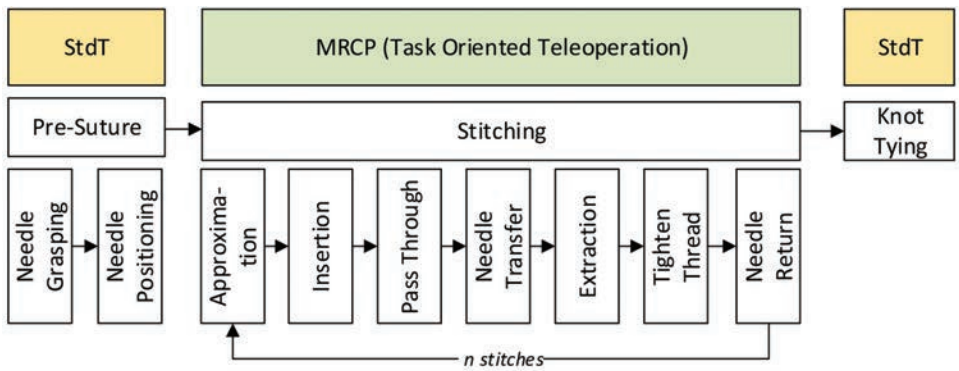


Figure 5.35: Schemas of task oriented teleoperation setup (MRCP+Cam).

To validate the combined approach, the first experiment consists in simulating the stitching sequence in which the needle must be always held by, at least, one robot. The subtask is divided into seven atomic subtasks for each stitch:

- Approximation: Needle approximation to the initial point of the suture.
- Insertion: Needle insertion into the tissue.
- Pass Though: Once the needle is inserted, it must pass through the tissue to be sutured.
- Needle Transfer: The grasping tool is blocked by the tissue requiring a task transfer to continue passing through and extract the needle. The robot that will continue with the execution grasps the needle from the other side of the tissue to be able to continue with the trajectory.
- Extraction: The needle must be completely extracted from the tissue on the opposite side of the needle insertion.
- Tighten Thread: The thread is tightened to joint both parts of the tissue.

- Needle return: The needle is returned to the starting side of the cut and transferred between robots to start the next stitch.

Fig.5.36 shows the seven atomic subtasks of the stitching task jointly with a schematic representation of the stitching process (single step).

Fig.5.37 shows the complete process of suturing, starting from the Pre-Suture subtask (Fig.5.37.a) and Fig.5.37.b)), a sample of stitching (from Fig.5.37.c) to Fig.5.37.i)) and, finally, after three stitches, the initial phase of knot tying (Fig.5.37.j)).

Firsts trials of the experiment have demonstrated the viability of the combined approach, enabling the complete execution of complex tasks. This approach eases the process of continuous stitching while confers the operator with absolute teleoperation control.

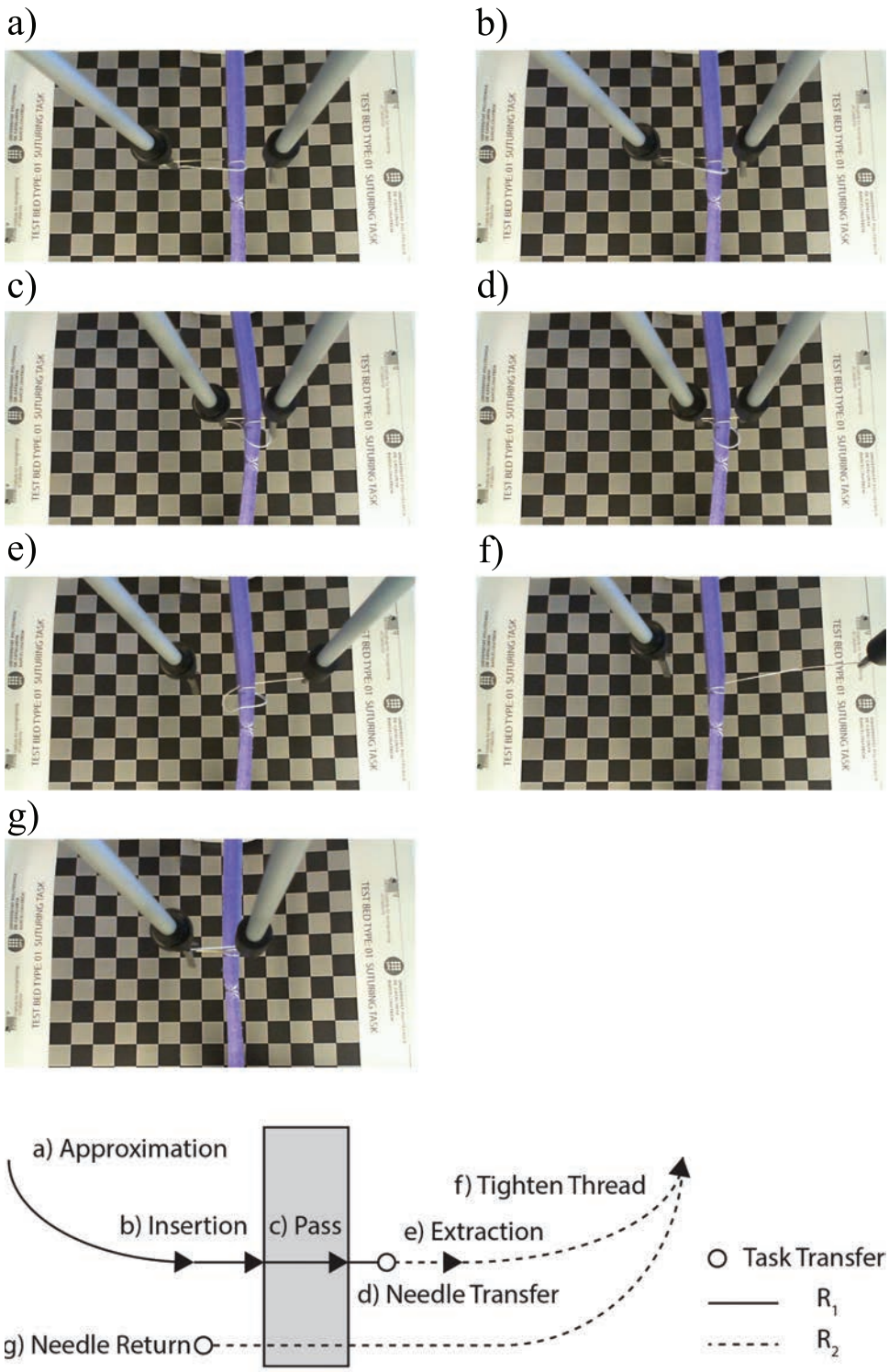


Figure 5.36: Stitching process with its 7 atomic subtasks using MRCP.

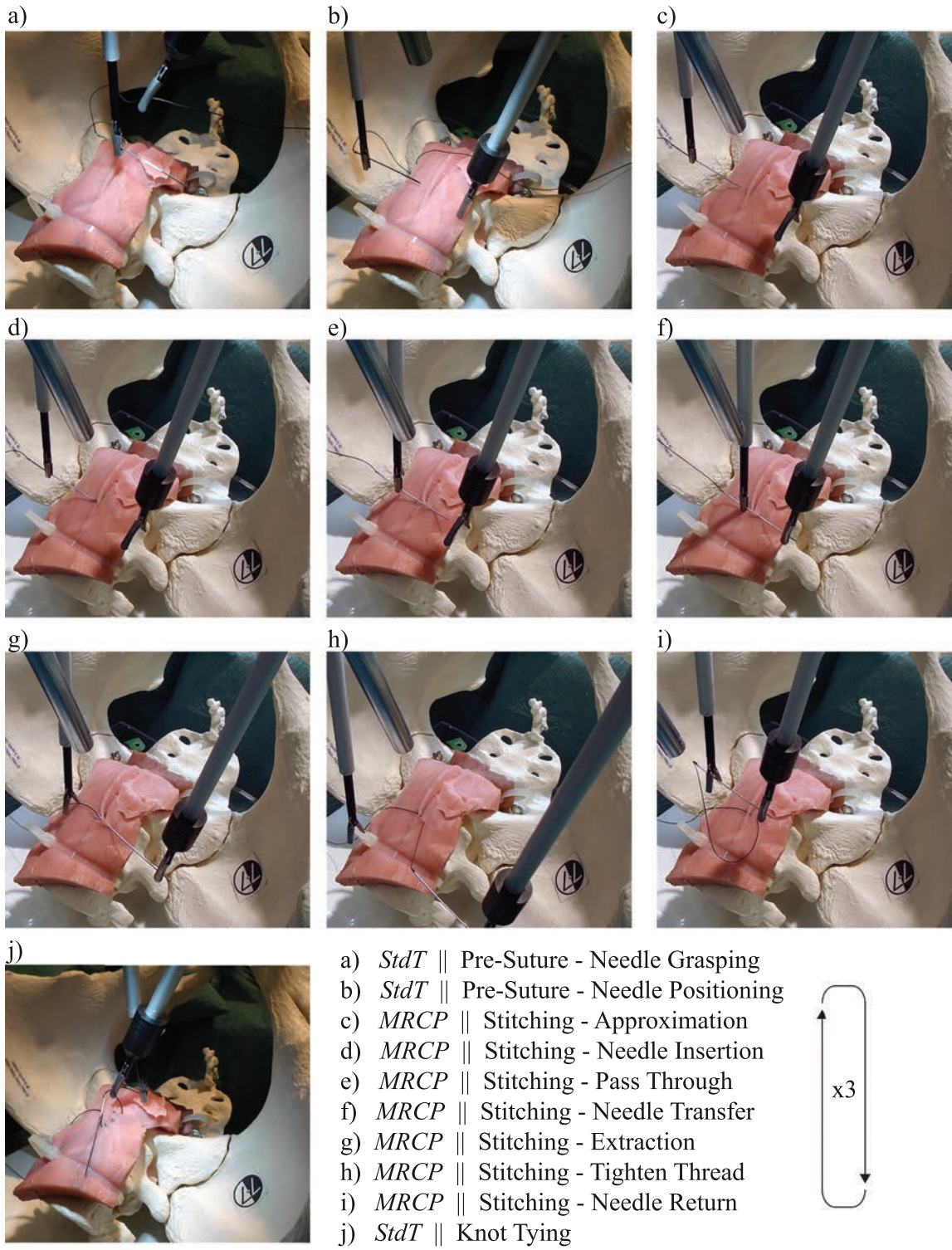


Figure 5.37: Complete process of a suture with three stitches using the combined approximation: StdT and MRCP

Chapter 6

Virtual Fixtures in MRCP

6.1 Introduction

This chapter studies the effects of different Virtual Fixtures, VF, in precise and dexterous teleoperation and introduces the task-oriented VF. Complementarity, this approach has also enabled the evaluation of VF as a means to improve the learning of basic skills in laparoscopic surgery. First, the chapter reviews the VF concepts and, second, introduces the task-oriented VF, an extension of VF that follows the same task-oriented principles used by MRCP. Then a case study of different VF acting in a trajectory guidance task in a virtual abdomen is presented and the results analysed. Finally, the possibility of developing task specific assistance VF, based on the combined use of several VF, is introduced.

VF are used to improve certain aspects of the operator's skills (e.g. accuracy, completion time, etc.) or to protect certain regions of the workspace (forbidden regions, tool guidance, etc.).

MRCP opens the possibility of generating task-oriented VF. In standard teleoperation, VF are applied to the master devices reflecting a certain behaviour or action on the slave robot. For instance, in a telemanipulation task, the effect of VF could be the generation of an attraction force that guide the robots to a collision free region. MRCP would apply forces to guide the task itself, reinforcing the idea of task oriented teleoperation. Following the same telemanipulation example, forces guiding

the manipulated object to a collision free region.

In order to evaluate the integration of VF in MRCP, five different VF have been defined: visual guidance, audio guidance, motion scaling, magnification and force feedback. These VF involve different sensory channels to provide information to the operator (a surgeon): visual, auditive and force feedback. These VF have been integrated into a virtual surgical simulator to define an experimental setup based on a trajectory following task inside a dynamic abdominal region.

A study was done to a total amount of 46 subjects both, surgeons and residents, who underwent a training session based on the proposed setup. Their performance when executing the designed experiment was recorded and used to identify the effect of virtual fixtures on the task execution and on the learning curve in terms of accuracy and completion time. Experimental results proved VF to be effective in improving the learning and surgeon performance, affecting differently accuracy and completion time. This suggests the possibility to tailor virtual fixtures on the specific task requirements.

This research work was developed under the Patient Safety in Robotics Surgery, SAFROS, European Project in ALTAIR laboratory in the Computer Science department of the University of Verona. This project is a Seventh Framework Programme research project (FP7-ICT-2009.5.2). Several hospitals collaborated in the experimental phase, offering their facilities and personnel (surgeons and residents): Ospedale Policlinico G.B. Rossi (Verona, Italy), Istituto Scientifico Universitario San Raffaele (Milano, Italy), Ospedale Policlinico Universitario di Padova, Consorci Sanitari de l'Anoia - Hospital d'Igualada (Igualada, Catalonia-Spain), Hospital Universitari Arnau de Vilanova (Lleida, Catalonia-Spain), Clínica de Ponent (Lleida, Catalonia-Spain), Hospital de la Santa Creu i Sant Pau (Barcelona, Catalonia-Spain), Hospital Universitari Vall d'Hebron (Barcelona, Catalonia-Spain), Corporació Sanitària Parc Taulí (Sabadell, Catalonia-Spain).

6.2 Virtual Fixtures

Virtual Fixtures can be defined as artificial software-generated signals and functions that, using diverse sensitive channels, are addressed to the operator. The additional information is oriented to correct or modify the operators actions during teleoperation in order to improve their skills, [167].

There are multiple VF classification criteria. One classification divides VF into assisted guidance and forbidden region. Guidance VF are designed to help the operator to guide the robot along a preferred path. For instance, a force attraction, in the form of a mass-spring-damper between the robot's Tool Center Point, TCP, and the path to be followed, minimizes undesired trajectory deviations. While, forbidden regions VF keep the robot out of certain regions of the workspace. For instance, a repulsion force like a virtual wall, can be generated when the robot TCP reaches the limit of a forbidden region.

Another possible classification of VF is based on the actuation over the transfer function between master and slave, M-S. Visual or auditive fixtures do not modify the M-S transfer function, while motion scaling and force feedback do. Motion scaling acts over the motion transfer function, varying, as desired, the scale between the movements executed by the operator on the master device and the trajectory executed by the slave robot. A force feedback VF injects additional forces on the master device in order to modify the operators behaviour.

Dextereous telemanipulation has a special interest in this research work. One of the most widely applied VF in surgical training (in precise telemanipulation) is motion scaling, MS. The benefits of MS in robotic surgery is evaluated in [168], where different fixed scaling factors (2:1, 4:1 and 6:1) are applied to a telesurgery system. The improvement in operators performance is measured in terms of errors and completion time. The usefulness of virtual fixtures in assisting a human operator is widely accepted; for instance, DaVinci surgical system integrates MS and allows the surgeon to set the scaling factor. In [169], the operators' dexterity obtained with different MS when manipulating a surgical telemanipulator is compared. The telemanipulator is a DaVinci surgical robot that provides three different fixed scaling motions (2:1,

3:1 and 5:1). For the experimental phase, a 3:1 scaling factor is used, in addition to hand tremor filtering (6 Hz). In this work the useful range of motion scaling is fixed between 1:1 and 10:1, but no explanation or citations are provided.

[170] presents a study of the MS and tremor reduction benefits using the Zeus Surgical System. Subjects touch six different targets with an endoscopic tool with and without robotic assistance. When this aid is enabled, three different levels of motion scaling are used (1:1, 2.5:1 and 7:1) in addition to tremor filtering. Authors state that MS greatly improves accuracy whereas tremor filtering has a limited effect.

[171] proposes a pick-and-place task in a micro-metric workspace using three different modes: unassisted, hand held (with compliant robot) and autonomous. During the experiments, fixed motion scaling is combined with a magnified vision on a Steady Hand robot [172] and a LARS robot [173]. Setup accuracy and reliability are compared. The hand held results more intuitive for operators and help to avoid collisions with the anatomical structures, relaxing the operator during the intervention.

[174] studies the integration of motion scaling and magnification. The paper states that MS reduces the errors when high magnification is used but, on the other hand, this increases the task completion time. Thus the authors suggest the need of determining a compromise between motion scaling and magnification to optimize time and accuracy. Similarly, the work described in [175] deforms the robot workspace to provide higher resolution on a predefined region of interest: the scaling factor is a function of the distance between the TCP and the target point. The authors also propose a vector-based approach, in which the scaling factor depends on the direction of motion. In this work, no numerical results are provided.

Transparency, which guarantees that the dynamics of the environment is displayed to the operator with no distortion, represents a big issue in the introduction of assistive technologies in teleoperation scenarios, as they can lead to distorted perception of the environment [176]. In particular, the introduction of force feedback in real surgical interventions may unstabilize the robot when the force feedback control laws are not tuned correctly. The injection of forces in the feedback channel may confuse the surgeon and make him/her unable to distinguish between an over-imposed force and

the force generated by the actual contact. Thus, during the development of assistive technologies attention must be paid to not bias operators perception and to design a training curriculum that avoids that the trainee learns to rely on information that will not be available during a real intervention.

6.3 Task-oriented Virtual Fixtures

This section develops the proposed task-oriented VF approach, once proved the effectiveness of VF in teleoperation.

In the field of VF, two types of approaches can be distinguished: those that focus on developing a generic set of VF and those designed for a concrete task. Following the generic VF approach, in [177], a set of task primitives are defined: stay on a point, maintain a direction, move along a line, etc. These constrains are based on an optimization function to define a robot concrete behaviour. A similar approach is proposed in [163, 178]. In these works, authors define a set of VF primitives based on point attraction and reactive VF. These VF are based on a set of three-dimensional geometric primitives: cone, circle, cylinder and cube, that are applied to generate preferred movement directions to a concrete point or to act as single or two-sided virtual walls. The primitives are applied to a cutting path over the surface of a deformable object in a virtual environment. VF have also been proposed for concrete purposes. These VF are designed for a particular task and system. In [179], a forbidden region applied to orthopaedic surgery is developed. More concretely, this work proposes a task oriented VF designed to avoid undesired femur and tibia cuts in prosthetic knee surgery. In [180], an adaptive VF based on a subtask decomposition is proposed. In each subtask (probabilistically estimated), the VF that most fits to the subtasks requirements is applied. Following this approach, the resulting system is more flexible and can be dynamically adapted to unpredicted situations; for instance, a non predicted obstacle inside a trajectory. Following the subtask division of a complex task, in [181] a stitching task by means of a circular needle in RMIS is addressed. The suture process is divided into five subtasks: select, align, bite, loop and knot. Each

step has a different composition of guidance and forbidden region VF.

The application of VF in multi-robot systems is not widely used yet, and there are still a set of open questions to be solved. One of these open fields is the cognitive load: how does the operator receive and process several sensitive inputs from different sources. For instance, a VF applied to each robot arm. The application of different VF or an incorrect modulation of these VF can create confusion and fatigue to the operator, and consistency is lost. For instance, applying a motion scaling, function of the distance to a target. If the robots have different distances with respect to their targets, different motion scaling factors are applied. A second solution could pass through the application of the same scaling factor to both robots following a unified criteria (e.g. the highest distance of both robots with respect to their targets determines the amount of motion scaling factor). Unfortunately, this solution presents the same drawbacks.

The described drawbacks are common in other fixtures like automatic visual positioning. When teleoperating with two arms, the system or the operator have to decide which of the tools defines the camera positioning and magnification. Some works have been developed to address the use of VF in multiple robot systems. In [182], a framework to constrain the robot movements by means of VF in a multi-robot teleoperation system is defined and an example of a surgical task (knot placement) is used to test and validate the proposal. The preferred motion direction is used to guide the operator.

6.3.1 MRCP + Task-Oriented VF

Following the task-oriented paradigm inherent in MRCP, a new research field arises: the generation of task-oriented VF integrated in the MRCP platform. This task-oriented VF approach differs from standard VF in different aspects: In standard teleoperation, VF are generated and applied to each master device as consequence of the interaction between the corresponding slave robot and the workspace. Instead, in MRCP, a VF is generated by the interaction between the task and the workspace.

Let's illustrate the differences of both approaches with an example of VF. The task consists in the telemanipulation of an object following a preferred path. Deviations from that path generate a force attraction to the mentioned path. In standard teleoperation, the set-up is composed of two master devices with haptic feedback that guide two robotic arms. When the TCP of each arm deviates from its predefined path, an attraction force appears on the corresponding master device, 6.1, where $\vec{F}_{M1}$ and $\vec{F}_{M2}$ are the force feedback of the master devices, $\vec{X}_{R1}$ and $\vec{X}_{R2}$ the robot's TCP pose, $\Delta\vec{X}_{M1}$ and $\Delta\vec{X}_{M2}$ the incremental master movements and, $\vec{X}_{Optimal1}$ and $\vec{X}_{Optimal2}$ the optimal or preferred paths.

$$\left.\begin{aligned}\vec{F}_{M1} &= f(\vec{X}_{R1} + \Delta\vec{X}_{M1}, \vec{X}_{Optimal1}) \\ \vec{F}_{M2} &= f(\vec{X}_{R2} + \Delta\vec{X}_{M2}, \vec{X}_{Optimal2})\end{aligned}\right\} \tag{6.1}$$

With MRCP, the system is formed by the same two robots and a single master device that guides the manipulated object. The VF are generated by the deviation between the current manipulated object position and the preferred path, 6.2, where $\vec{X}_{OptimalTask}$ is the preferred task path.

$$\vec{F}_{M} = f(\vec{X}_{Task} + \Delta\vec{X}_{M}, \vec{X}_{OptimalTask}) \tag{6.2}$$

This approach reinforces the sensory immersion to the proposed MRCP, generating the operator's perception of being executing the task with the master device instead of telemanipulating two slave robots.

6.4 Case study: trajectory guidance in MIS

The following experiment, trajectory guidance in MIS, is focused on a double objective. To test and validate the integration of VF in MRCP, and to analyse the effects of the different proposed VF in the learning curves of residents and surgeons in RMIS.

The task oriented VF applied to MRCP is studied by means of a test comparing

the results obtained with different individual VF. The results are analysed in terms of completion time and execution accuracy. Its statistical signification as well as the subjective evaluation are analysed. In future works, this study will be extended to the combined use of different VF acting cooperatively.

6.4.1 Experiment: VA + VF + MRCP

A virtual surgical environment, the Virtual Abdomen, VA, has been used to evaluate the effects of VF in robotic surgery. The advantages of a virtual environment over a real one are manifold: it allows the full control of each trial parameters, enables the generation of multiple scenarios, ensures repeatability and provides a portable and low cost setup.

Virtual environment

The chosen virtual environment is based on the work developed in the A Laboratory for Teleoperation and Autonomous Intelligent Robots, ALTAIR, of the Computer Science department of Verona's University in the frame of the Patient Safety in Robotics Surgery, SAFROS. This virtual abdomen simulator is described in [183, 184]. The simulator provides a reconstruction of abdominal anatomy in which the operator can perform surgical tasks such as probing, grabbing, clamping or cutting. The organs can be generated from real patient data, reproducing the specific patient information, including volumetric and tissue information. The physics of organs and tissues are modelled with mass spring models [185] to increase the realism of simulation. These models generate a realistic scenario and also allow haptic rendering that is provided to the operator through a Sensable PHANToM Omni. Fig.6.1 shows a general view of the abdominal simulator.

The integration of any VF in this virtual environment requires sharing some information between the physics engine and the VF computation. The current implementation downloads from the GPU memory the required data to compute each VF. The limited amount of data needed to fully specify VF (coordinates of one point and

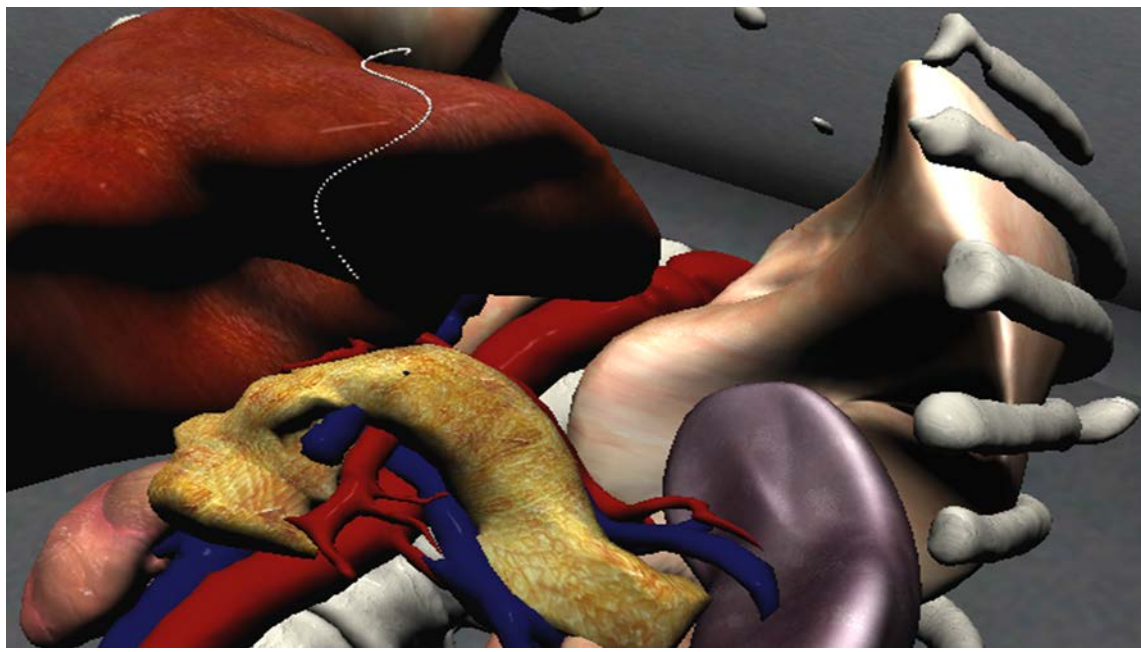


Figure 6.1: General view of the abdominal region simulated by the Virtual Abdomen. A predefined path to be followed by a laparoscopic tool is shown.

its normal in the case of point targeting and coordinates of few points in the case of trajectory following) and the asynchronous download function introduces a limited delay (1ms) in the updating of VF, what allows generating a smooth interaction with the operator. Fig.6.2 presents the block schema of the VA, working with MRCP and VF in the proposed experimental setup.

This integration allows the development of the experiments required to analyse the effects of each VF individually. The evaluation of any fixture requires the assessment of subject's achieved proficiency without VF assistance. Thus, each subject base line performance is identified before the application of any VF with various initial trials without VF assistance. This allows to compare the effects of different VF and the cognitive load associated to each of them. To isolate the learning effect of using VF in the assessment of operator's performance, the subject learning curve is estimated through the comparison of only the VF-free trials that are executed during the experiment.

Fig.6.3 shows a sequence of images of an operator executing a trajectory following

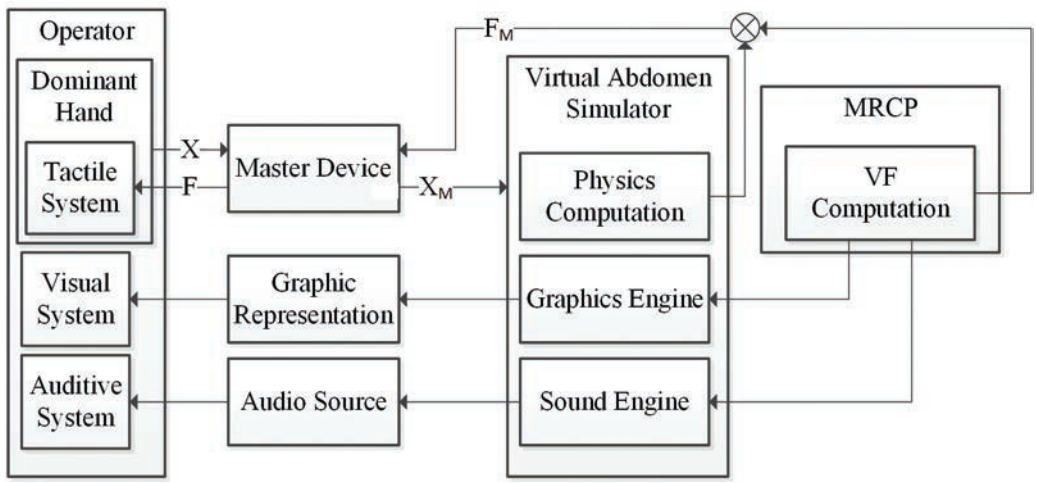


Figure 6.2: Block schema of the VA working with MRCP and VF in the experimental setup.

task. As the images reflect, the laparoscopic tool does not follow the laparoscopic constrained movement simulating the MRCP behaviour: multiple robots allow higher reachability.

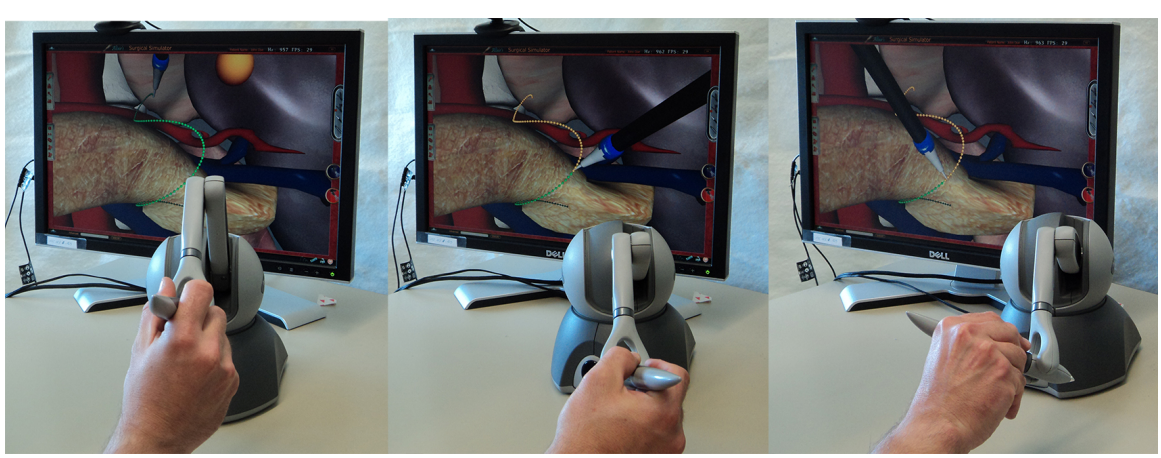


Figure 6.3: Sequence of images of an operator executing a trajectory following task. A Phantom Omni is used as master device

6.4.2 Proposed Virtual Fixtures

For the purposes of this research work, five individual fixtures have been identified and parametrized: Visual Guidance, Audio Guidance, Motion Scaling, Magnification and Force Feedback. Visual Guidance provides the operator with a visual aid (a guidance vector) to follow a preferred path. Audio Guidance points the deviation of the tool with a sound signal. Motion Scaling varies the relation between master commands and slave movements, reducing motion when the tool deviates from the path. Magnification varies the workspace view according to the task requirements. Finally Force Feedback generates a haptic feedback that helps the operator to not deviate from the path.

Several criteria drove the definition of the used VF:

- Task oriented: each VF behavior should be adaptable to the task requirements for which it is designed.
- Reduced set of VF: A small set of VF ensures better use and shorter learning process.
- Different sensory modalities: VF should correctly exploit different sensory modalities to efficiently convey information to the operator.
- Master/slave interaction: VF should allow to change the mapping between master and slave.

To meet the first two design criteria, a modulation function that controls the behaviour of the VF has been introduced. This function should satisfy some properties:

- Complete control of its behavior by means of a compact set of parameters.
- Ease of parameter understanding and tuning.
- Derivability in the entire modulation domain.
- Applicable to all VF: the use of the same modulation function eases surgeon interaction.

The function selected for the modulation of all the proposed fixtures is the five-parameter logistic function, f5PL, 6.3, derived from the basic sigmoid function.

$$y = d + \frac{a - d}{(1 + (x/c)^b)^g} \tag{6.3}$$

where a and d are the values of the function at $x = 0$ and $x = \infty$ respectively, b controls the slope of the curve, c is the mid-range concentration and g is the asymmetry factor. Further information on the f5PL can be found in [186–188].

Fig.6.4 shows an example of the modulation function applied to motion scaling in a point targeting task. Three regions can be identified in the function domain. First, in the vicinity of a target (POI), $d_{norm} \to 0$, the modulation remains constant and $y \simeq a$: high motion scaling to improve operator's accuracy. This provides the operator with a stable region, ensuring dexterity and transparency. Second, when $d_{norm} \to 1$ (long distance between tool and target), the function tends to $y \simeq b$. In this region, the MS factor remains constant. The scaling factor is low to enable fast tool approximation to the target. Between these two extremes, the function provides a smooth transition region. When the tool approximates to the target, the scaling factor increases.

The modulation function enables the adjustment of the VF to the task. For instance, let's assume the use of motion scaling for trajectory following and point targeting tasks. In trajectory following, motion scaling prevents undesired deviations from the planned path. Scaling factor is small (motion is amplified) when the TCP is close to the trajectory and increases (leading to reduced motion) when the TCP deviates from its path. This behaviour reduces the tool movements when the error becomes too big. In point targeting, instead, scaling factor is high (motion is reduced) when the TCP is close to the POI, thus increasing the operator's ability to accurately reach the point.

The next part of this chapter describes these VF applied to a trajectory following task. The trajectory is described by a set of points (POIs) forming a 3D path with different curvatures and distances to the abdominal organs.

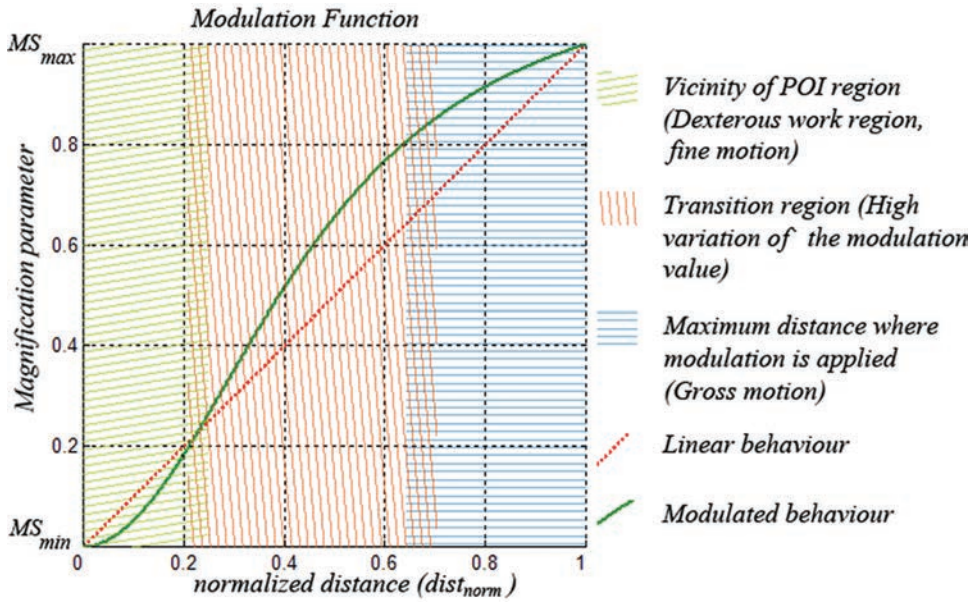


Figure 6.4: f5PL modulation function applied to MS for point targeting task

Visual Guidance, VG

Visual Guidance provides the surgeon with a guidance tool that acts through the visual sensory channel. VG consists in a graphical representation of the minimum distance vector between the TCP and the goal point, which provides surgeons with information about the proximity to delicate regions and of possible collisions while approaching the POI. The distance vector is represented with a color that can be easily distinguished from the rest of the workspace. Fig.6.5 shows the VG used in a trajectory following task. The VF Computation module receives the Physics computation output and

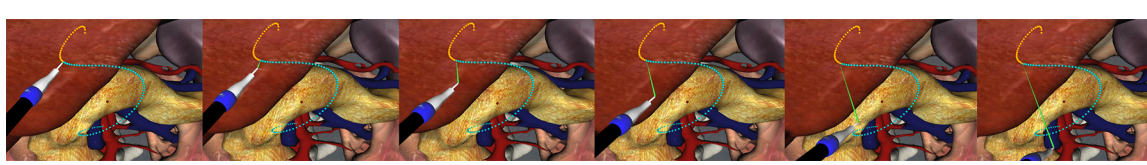


Figure 6.5: Sequence of different tool positions and corresponding minimum distance vectors generated by the VG

generates the guidance vector, which is injected in the Graphic Engine. Fig.6.6 shows the modified block schema.

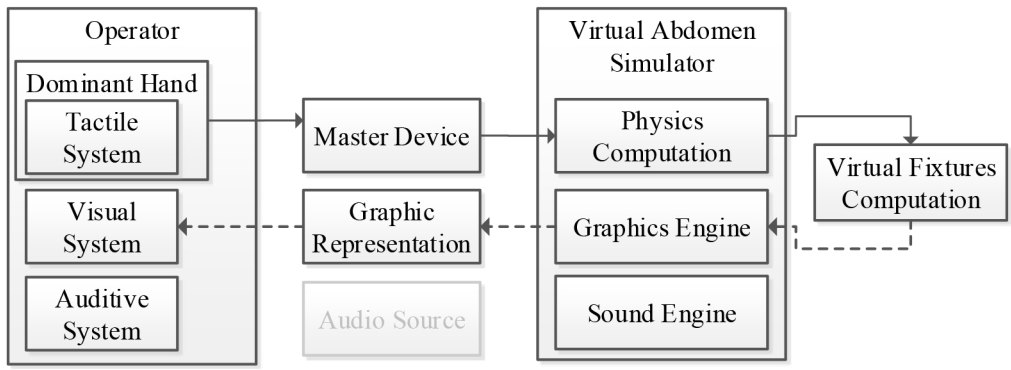


Figure 6.6: Block diagram of the required control modules for Visual Guidance. Segmented lines show the variations with respect to the original block schema, without VF.

Audio Guidance, AG

Audio Guidance follows the same principle as VG but, using the auditive domain: i.e. errors are signalled to the operator through sounds that may help to correct the motor behavior, helping the surgeon to keep the TCP close to the trajectory. Changes in the properties of the sound signal are controlled by the modulated distance between the tool and the goal point: when the error increases, the elapsed time between two consecutive sound signals decreases and the sample frequency increases, generating a higher frequency sound. In presence of significant errors, the sound changes are easily perceived and provide a guidance effect. If the deviation reaches a critical distance, the sound becomes annoying (high pitch with short intervals between two consecutive sound signals). Audio Guidance introduces a new module into the experiment. The

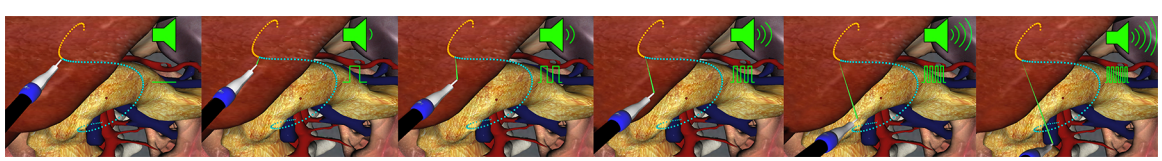


Figure 6.7: Sequence of snapshots of the tool with Audio Guidance VF applied in trajectory following. The volume and pitch of the sound are illustrated on the images.

sound engine, inside the Virtual Abdomen Simulator, generates the required sound

signals that the audio source reproduces, Fig.6.8

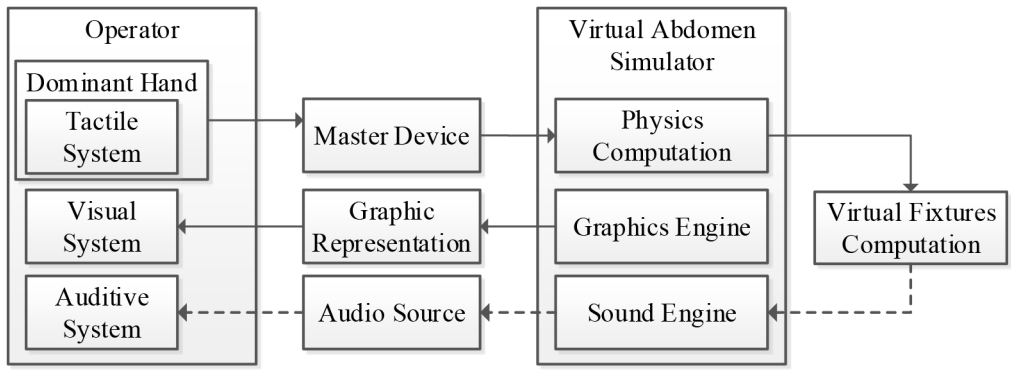


Figure 6.8: Block diagram of the experiment including Audio Guidance. Segmented lines shows the variations with respect to the original block schema without VF.

Motion Scaling, MS

Motion Scaling is designed to increase the accuracy of the surgeon by modifying the scaling factor between the master and the slave. Let X_M and X_S be the incremental movements of the master, M, and slave, S, actuators. The relationship between them, $MS = X_M/X_S$, can be modified in order to increase the surgeons skills. In bilateral teleoperated systems, the motion of the slave depends on the master device input, i.e. $X_S = f(X_M)$. Usually, f includes a fixed scale parameter, whereas MS provides a variable vector-based scale parameter that is function of the error between the TCP and the POI. The MS factor is computed and applied independently to each TCP component. Fig.6.9 illustrates an example of vector-based motion scaling for a trajectory following task. The MS factor applied to TCP1 and TCP2 depends on the distance between each TCP component (TCP_x, TCP_y) and the POI. For $TCP1$, where $d_x \ll d_y$, the MS factor in x is greater than in y, $MS_x \gg MS_y$, whereas in $TCP2$, the MS factor per component is inverse, $MS_x \ll MS_y$. The f5PL function for each component is also shown inside the graphic. The MS function strongly depends on the considered task. Fig.6.10 shows a set of snapshots of the application of MS

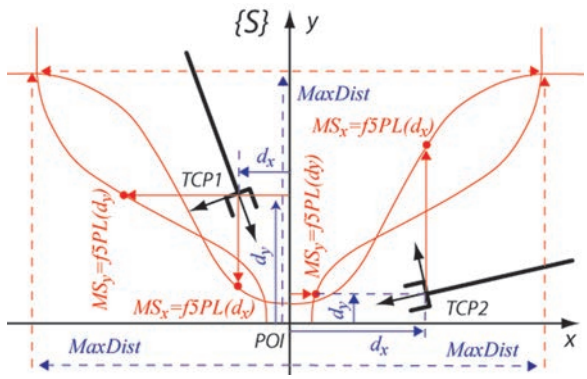


Figure 6.9: Vector based MS in a 2d workspace

parametrized to follow a trajectory. The cone represents the position of the tool as if no MS was applied.

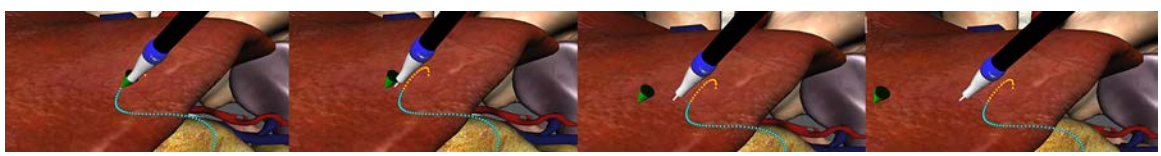


Figure 6.10: Sequence of snapshots of the tool with the MS applied in trajectory following. The cone represents the tool position without MS.

Motion Scaling modifies the master device output signal to vary the amount of displacement. To generate MS, it requires information from the master device, and the physics computation modules become those of Fig.6.11.

Magnification, Mag

Magnification provides the surgeon with an automated magnification and positioning of the endoscopic camera. This approach is based on the division of a surgical workspace into different spatial regions: tool navigation inside the abdomen in relatively free spaces, POI approaching and dexterous work. When the tool is in the navigation area, a wide view is provided that shows the position of organs and tools. In the approaching phase, the amount of magnification rapidly varies, allowing an accurate reaching of the POI while keeping the vision of surrounding organs for contact

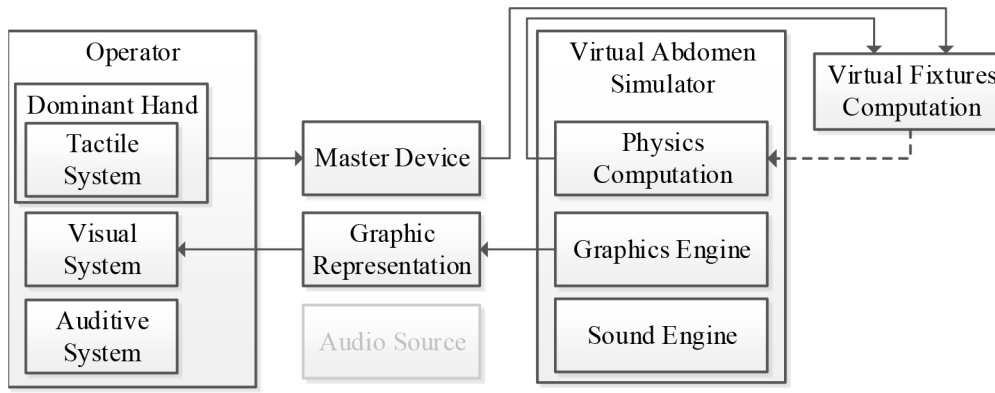


Figure 6.11: Block diagram of the experiment including Motion Scaling. The segmented lines show the variations with respect to the original block schema without VF.

avoidance. Finally, in the dexterous work region, a highly magnified view is provided. This ensures that when the TCP is away from the target point the operator has a wide view of the environment, whereas when the TCP is close to the target point a magnified view allows the operator to perceive fine details of the area of interest. During navigation and dexterous work, minimum Mag variations are introduced to not disturb the surgeon. Fig.6.12 presents some screenshots of Mag: the amount of magnification decreases when the tool deviates from the path.

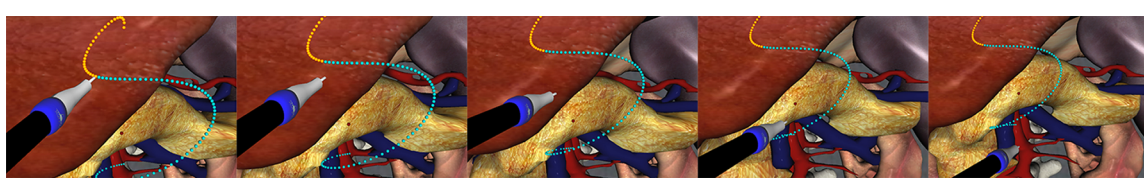


Figure 6.12: Sequence of snapshots of the tool with the Mag applied to the operator's view.

Magnification uses the relative position of the TCP with respect to the trajectory to determine the workspace visualization (camera position and magnification). Fig.6.13 presents the modifications of the original schema.

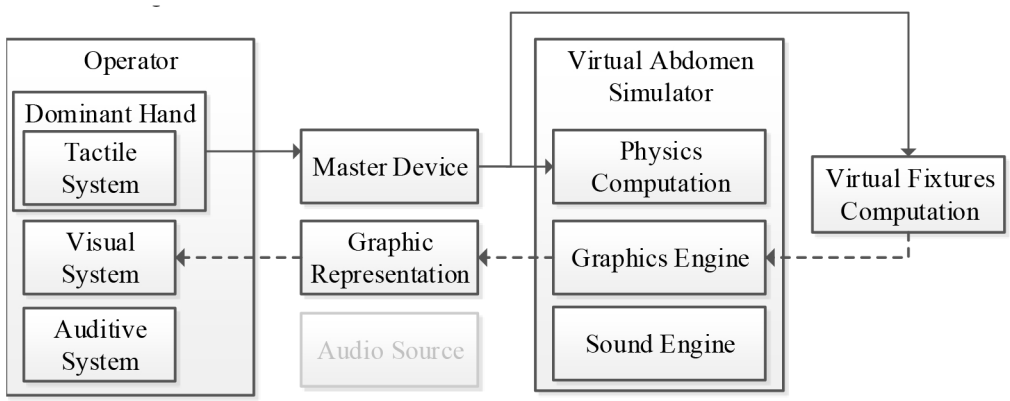


Figure 6.13: Block diagram of the experiment including Magnification module. Segmented lines show the variations with respect to the original block schema without VF.

Force Feedback, FF

Force Feedback guides the tool towards the POI with an attraction force, providing the operator with a smooth tool guidance. This attraction force is mapped on the tool center point, TCP. The direction of the applied force vector corresponds to the minimum distance vector from the TCP to the POI. The behavior of the attraction force follows the spring-damper model 6.4, where d_{norm} is the normalized distance for each axis.

$$F = -Kd_{norm} - C\partial d_{norm}/\partial t. \tag{6.4}$$

The elastic and damping coefficients (K and C respectively) are the result of modulating two pre-defined coefficients, k and c, using d_{norm} as input parameter, $K = kf5PL(d_{norm})$ and $C = cf5PL(d_{norm})$. This modulation provides a non linearly-varying force that is negligible when the error is small and smoothly reaches its maximum when the error increases.

Force Feedback uses physics computation to generate the haptic response of the master device. The output signal of the virtual fixture module is used, again, in the physics to obtain organs deformation and to generate the force feedback.

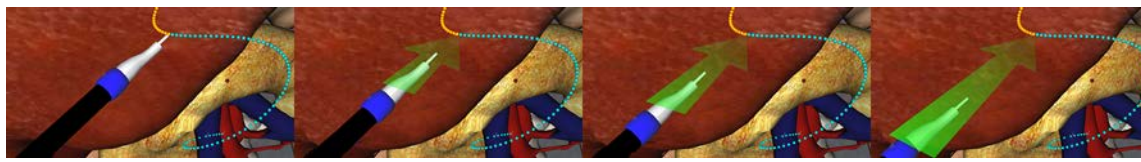


Figure 6.14: Sequence of snapshots of the tool with the attraction FF. The amount and direction of the force is represented as an arrow in the images.

6.4.3 Design of the experiment

In this section, the experiment and the experimental procedure, the validity methodology and the evaluation process are described. To test and validate the VA+VF+MRCP approach, a trajectory following task inside the abdominal region has been designed and the different VF described above applied. The operator must guide the tool tip along a predefined path.

Task: Trajectory following

The trajectory following task requires from the operator point of view, a good spatial interpretation when following a 3 DoF path in the vicinity of deformable organs, trajectory occlusions and path curvatures. These organs are in movement due to breathing and heart beating. The vicinity of the organs demands high dexterity from the operator to avoid undesired organs collisions. In order to cancel the learning effect, six different orientations, randomly sorted, of the same trajectory are proposed. The use of the same trajectory guaranties maintaining the same difficulty degree in each trial. Fig.6.15 shows the six orientations. A Phantom Omni is used as master device to manipulate the simulated surgical tool.

Experimental procedure

The experimental procedure has been validated and the results analysed by two psychologists (involved in SAFROS project) specialized on human cognitive load and analysis of different haptic feedback applied to robotics. The evaluation of any fixture requires the assessment of subject's proficiency without VF assistance. Thus, the first

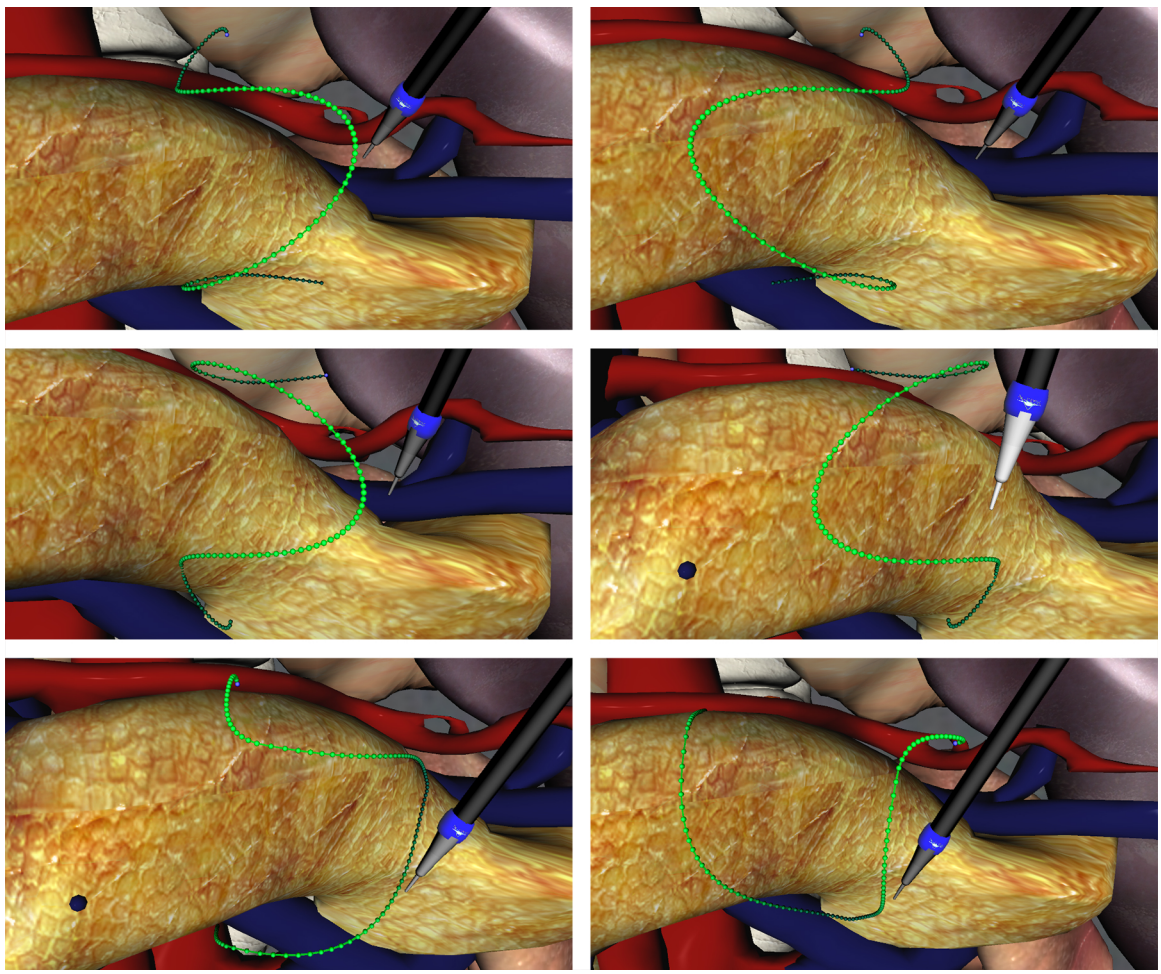


Figure 6.15: Snapshot of the six different trajectory orientations inside the virtual abdomen.

part of the experiment identifies the subject base line performance with 3 VF-free trials. The second part proposes 5 blocks composed, each, by 6 trials, all of them assisted by the same VF, followed by one VF free trial. During the five blocks, scheduled to cover all the considered VF, the subject faces 6 different trajectories obtained by mirroring and rotating a single original path. This ensures that their difficulty is the same in terms of curvature and length and reduces any learning effect. The six trajectories are presented in randomized order to avoid any facilitatory effect due to specific orientation sequence. On the contrary, all the VF-free trials are carried out on the same trajectory to exclude trajectory specific bias.

Before the experiment, subjects are provided with a short introduction to the experiment goals, the virtual simulator capabilities and the VF. After the introduction, a questionnaire collects subject personal and experience data, then the trials are proposed to the subject. Finally, a second questionnaire gets the subjective evaluation for each VF and possible suggestions.

Statistical analysis isolate and verify the effects of each virtual fixture on the learning process and identifies the presence of significant differences between them. The learning curve is estimated by comparing only the set of VF-free trials, Repeated Measures Analysis of Variance (RM-ANOVA) and Tukey's Honestly Significant Difference (HSD) post-hoc test of the aggregated data.

Basis of the experiment in learning process

As introduced above, the experiment was designed with two main goals: a) prove the use of several VF using different sensory channels and cognitive load and, b) test the effect of these VF in the learning curve of residents and surgeons in RMIS.

Surgeons training is mainly based on a Halstedian apprenticeship model [189] whereby residents learn by directly watching a more experienced surgeon during the intervention, slowly increasing their hands-on experience with a variable degree of autonomy over time. There are considerable ethical, economic and legal problems related to this procedure. These problems have led to the development of alternative tools and modalities for the improvement of laparoscopic and robotic surgical skills, whose goal is to ensure surgeons proficiency before they start operating on real patients.

Some basic abilities are prerequisites for the correct and safe execution of any surgical procedure, and, in particular in robotic surgery. They, in fact, allow the subject to cope with the perceptual abnormalities that are specific of robotic surgery, mainly the lack of haptic feedback and the dissociation between operator and robot movements. Surgeon's proficiency can be improved through the training of visuo-spatial and perceptual-motor abilities in presence of an indirect mapping between operator hands and robot end effector and in absence of force feedback. The proper

training of visuo-spatial skills allows the subject to compensate the lack of haptic information. The development of these abilities is fundamental for the completion of a surgical procedure and positively affects the outcome of an intervention, avoiding the occurrence of unexpected events caused by the wrong execution of correct procedures.

Virtual simulators allow surgeons to acquire skills required by minimally invasive surgery in a safer, less stressing and cheaper way. Real time data recording constitutes a relevant advantage provided by virtual simulators and makes their application in training extremely valuable and effective. In addition, the integration of assistive technologies in simulation may increase the efficacy of training. Current surgical training procedures, in fact, rely on the repetition of a task to increase their skills. Operator performance is evaluated after the completion of a task through the analysis of psychomotor data or, alternatively, through the subjective evaluation performed by an external observer. The introduction of assistant support, on the contrary, allows the surveillance of the trainee during the execution of tasks providing prompt correction of errors, with relevant and positive effects on the learning process [190].

The goal of this experiment is the development of virtual simulators that assist the operator (surgeon or resident) in shortening the learning curve in robotic surgery training. To this extent VF are integrated into an existing surgical simulator. This approach follows the Skill-Rule-Knowledge (SRK) taxonomy formulated in [191]. It is expected that, the analysis of the effects of virtual fixtures may lead to the definition of more complex assistive schemas and also to their integration into real scenarios.

Experiment sample

The experiment sample is composed of 46 subjects. 31 of them are surgeons and 15 residents with different background. Fig.6.16 shows the composition of the sample by specialization (Fig.6.16.a), experience in laparoscopy (Fig.6.16.b) and experience with surgical simulators (Fig.6.16.c) or in robotic surgery (Fig.6.16.d).

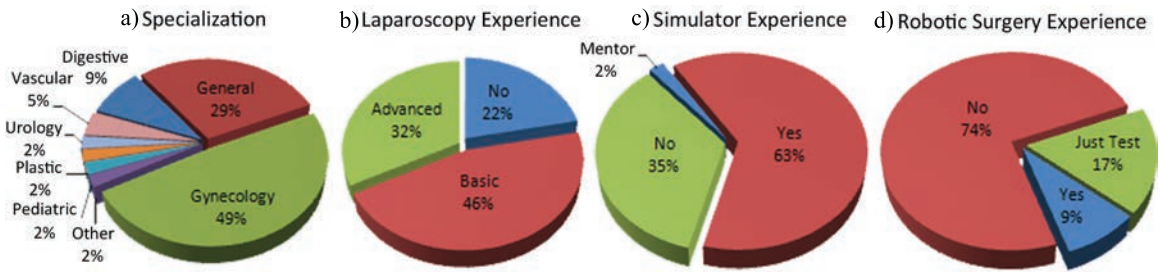


Figure 6.16: Statistics of experiment population

Experiment data recording

At each temporal step k the system logs the position of the TCP $\mathbf{x}(k)$ and the coordinates of the goal point $\mathbf{g}(k)$ on the trajectory. In addition, when the operator passes through a trajectory point, p, the system stores the elapsed time since the beginning of the trial t_p. Since each trial has a different completion time and thus a different amount of sampled data, values logged during each trial have been aggregated to ensure the comparability of the results. For the i-th trajectory point of coordinates $\mathbf{p}_i$, the cumulative error e_i and the latency l_i are defined, 6.5 as well as the total distance covered (measured with the Manhattan distance) and the time spent to move from one point to the next.

$$\begin{aligned} e_i &= \sum_{j|\mathbf{p}_i=\mathbf{g}(j)} \|\mathbf{x}(j) - \mathbf{p}_i\|_1 \\ l_i &= t_i - t_{1-i} \end{aligned} \tag{6.5}$$

6.4.4 Experimental Results

The review of the experimental results is divided between the statistical analysis of the recorded trials data and the subjective evaluation of the participants. Both analysis are presented in the next sections.

Objective results evaluation

The experiment includes two statistical factors: virtual fixtures (6 levels, one for each VF plus the VF free scenario) and the trial number (8 levels: one for each VF free trial in the sequence). The analysis of the obtained subjects performance trend during the whole experiment by the VF free trials and among each VF trials provides different results for the error measure and for the latency measure. Fig.6.17 shows the mean values of the cumulative error and Fig.6.18, the latency.

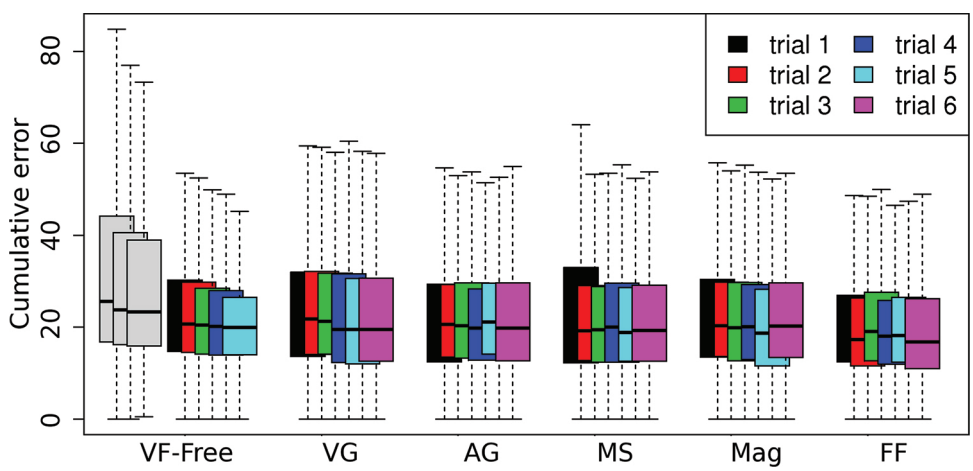


Figure 6.17: Mean value and interquartile ranges (lower bound, first and third quartile, upper bound) for the cumulative error along trials

The trend of the cumulative error for the VF free trials shows a considerable gap between the third and the fourth repetition. The gap between trials when VF are used decreases significantly. The step in the cumulative error trend appears in correspondence with the first block of VF assisted trials after the three initial VF free repetitions. These results lead to the conclusion that the introduction of any virtual fixture strongly affects the operator performance and that this effects continues, even when the support of the VF is disabled. This analysis does not allow to identify the most effective VF, since their random presentation order reduces the effect of VF. The trend of latency plots does not allow to assume the presence of any positive effect of VF on the completion time. Latency together with the considerations provided about

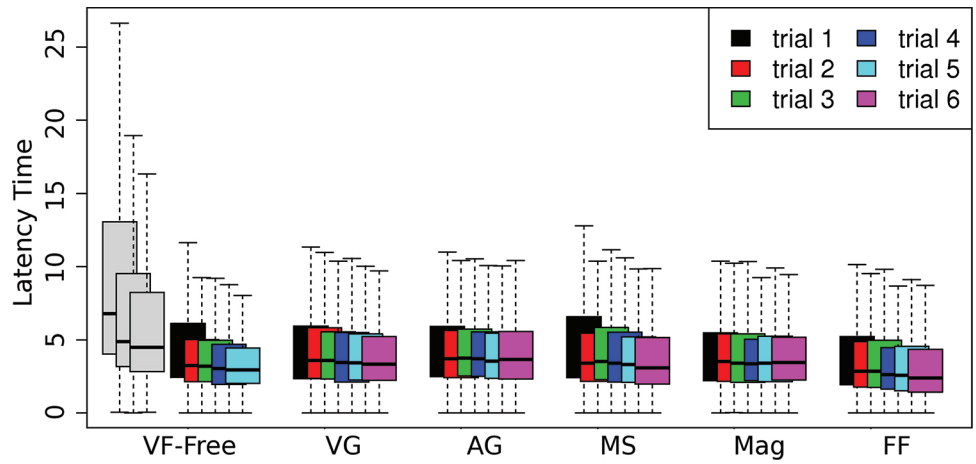


Figure 6.18: Mean value and interquartile ranges (lower bound, first and third quartile, upper bound) for the latency along trials

the cumulative error, suggests that VF are effective in increasing subject's accuracy, but they do not increase the speed of motion.

RM-ANOVA applied to data collected during VF assisted trials shows the statistical significance of the VF with a value of $F_{(5,98131)} = 35.030$, $p < 0.001$ for the cumulative error analysis and a value of $F_{(5,98131)} = 44.099$, $p < 0.001$ for the latency analysis. The effect of the trial factor is also significant for cumulative error $F_{(7,98131)} = 22.871$, $p < 0.001$ and latency $F_{(7,98131)} = 87.344$, $p < 0.001$. This proves that different VF provide different effects on operator performance. Tukey HSD test shows that FF is the most effective VF in reducing cumulative error, but its effect is not significantly different from AG and MS. The same test on latency values shows that completion time is improved by Mag and that its effect is not significantly different from that of AG.

Operators subjective evaluation

At the end of the experiment, the subjects where asked to answer a set of questions to obtain the subjective evaluation of the different VF. Three questions where formulated with the aim of evaluating which were the best and worst VF, as well as

a numerical evaluation, from 0 to 10, of each VF. The order of the best evaluated VF was the following: 36%Mag, 24%FF, 19%AG, 13%VG and 8%MS. Concerning the less useful VF, the results where: 39%MS, 35%VG, 14%AG 10%Mag and finally 2%FF. The means of the numerical evaluation of each VF where: 7.08FF, 6.85Mag, 6.38AG, 5.63VG and 4.74MS. Fig.6.19 presents the graphical representation of these evaluations.

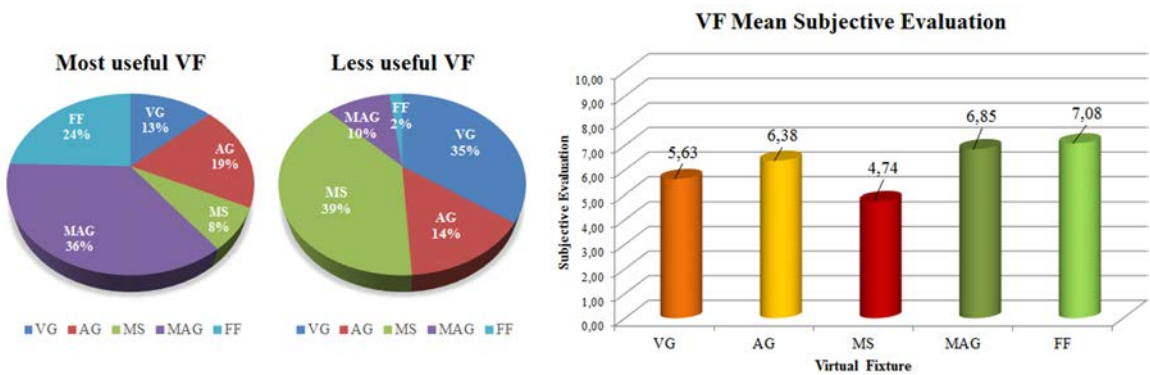


Figure 6.19: Subjective evaluation of the VF used during the experiments. From left to right: percentage of the best and the worst evaluated VF. Finally, the mean of the numerical evaluation of each VF.

The results point that FF and Mag are the preferred VF from the surgeons subjective evaluation, being FF the best evaluated. Haptic feedback increases the operator's remote workspace immersion and, consequently, improves teleoperation performance. FF as guidance VF has demonstrated its effectiveness is several studies. It also provides a friendly user interface where the operator feels comfortable and safer when executing a task. The amount of Force applied to the master device is soft enough to allow the operator execute a free navigation and does not produce fatigue if the operator decides to freely move around the workspace. Higher Forces improve accuracy, but limits the operators movement. Reviewing the results and the subjective analysis, the use of FF seems to need a short adaptation period, property that cannot be associated to all VF. Concerning Mag, surgeons are used to manual camera guidance to obtain the desired point of view and magnification. The automatic camera guidance, Mag,

is a tool that surgeons are used to. Mag imitates the real behaviour during an intervention while cancelling the camera assistant tremors, fatigue and misunderstandings between surgeon and camera assistant. This previous knowledge of this VF makes it more friendly and better evaluated. Mag must provide the possibility of manual guidance in a real intervention when it is desired to make this VF more useful and flexible. AG also obtained a positive evaluation as a guidance VF. Surgeons are used to work with different signal sounds during surgeries (e.g. energy tool for coagulation, vital signs, ...), resulting an already known communication tool. Sound signals, when noticeable deviations occur, were too annoying for some subjects. Other subjects disappointed with the remaining sound signal (low frequency and long playing interval) when the tool was close to the path. These subjects suggest to cancel any sound signal when the tool is on the correct path.

MS and VG obtained the worst subjective evaluation results. MS was introduced with the aim of generating a free corridor in the path vicinity with no tool movement restriction. Acting in this manner, the effects of MS are not appreciated by the operator. Reviewing their answers, most of them gave a low evaluation not because of its usefulness, but for not being able to appreciate its effects. The use of a visual cone showing the tool as if no MS were applied could increase the perception of VF acting on the task execution. VG provides the subject with a virtual guide to retake the correct path execution. Most of the subjects mentioned that the depth sensation of the simulation and the minimum distance vector were insufficient, generating confusion more than helping. Another important conclusion obtained with the subjective evaluation is the disparity of criteria to select the best VF and its parametrization. This reinforces the idea of generating a set of VF available to the operator and easily configurable to fit with the task requirements which are the operators preferences.

6.5 Task Specific Assistance

During the development of the previously presented set of VF, the possibility of generating more complex VF arose. Through cooperation with surgeons and a review

of related work, the need of designing adaptive and task oriented VF was identified. The proposed solution is based on the construction of VF schemas, VFS, which are the result of the combination of different VF applied at the same time assisting for a specific teleoperated task.

The design and evaluation of different VF requires a strict definition of all the basic VF that will compose the future VF schemas. The set of VF can be defined as an algebraic structure, where VF can be combined through a set of operators. This strict theoretical model is under development and will help in future demonstration of schemas properties.

To effectively address the specificity of different surgical procedures, two VFS have been developed taking into account the peculiarities of two representative surgical tasks: trajectory following and point targeting.

The introduction of complex VFS, implies considerable advantages from the psychophysical point of view. Carefully designed VFS may convey high amount of data without affecting the cognitive load of the operator [192]. This is the case, for example, of camera magnification associated with motion scaling, as introduced in [174, 193].

Once introduced the task specific assistance and the proposed VFS, two tasks and their associated VFS are presented. The first task is point targeting (a tool should reach a certain point over an organ surface with high accuracy) and the second is trajectory following (a tool should follow a predefined path inside the abdominal wall without excessive deviations).

6.5.1 Point targeting

Point targeting is a frequent action in laparoscopic surgery: it consists in reaching a point inside the abdomen coping with the constraints introduced by the laparoscopic tool and by the surrounding tissues and anatomical structures.

Several surgical procedures can be classified as point targeting tasks: from the basic cutting action, obtained by moving an energized tool to the point in which the incision has to be performed and by activating it; to the tissue dissection, in

which the grasped point plays and important role for the correct execution of the task. To ensure patient safety, the execution of these actions should satisfy some primary requirements: high accuracy, high repeatability and safe motions. Following these requirements a preliminary VFS was designed. Fig.6.20 illustrates the schematic representation. The proposed schema applies, first, a variable Motion Scaling ratio:

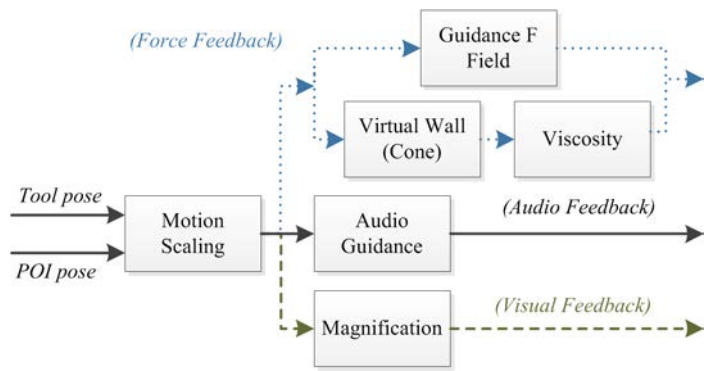


Figure 6.20: Point targeting VFS block diagram

higher values of scaling are used when the tool is closer to the point of Interest, POI, increasing the accuracy and repeatability of surgeon's movements. The VFS then follows three parallel tracks. This division is based on the sensory modality exploited by each proposed VF: haptic force feedbak, audio and visual feedback.

Force feedback, oriented to increase patient's safety, depends on the tool position inside the workspace, which is divided into two regions by a virtual cone aligned with the normal to the POI. Inside the conic region, a force guides the tool towards the POI while variable viscosity, that increases with POI proximity, damps excessive velocities. This prevents unwanted contacts and decreases dangerous side effects of sudden tool movements. If the tool is outside the cone, the force guides the operator toward the closest point on the cone surface. Audio feedback generates a signal that varies in frequency and sampling period as a function of the POI distance. Magnification module feeds the visual sensory modality: it controls the position and magnification of the endoscopic camera in accordance with the motion scaling factor. When the tool is far from the POI, a global view of the abdominal region is provided, the surgeon thus,

has a better understanding of the tools position with respect to the patients anatomy. When the tool approaches the POI, magnification smoothly increases, providing a better view of the POI and, consequently, increasing surgeon's accuracy. The camera always supplies the view of both the POI and the tool TCP.

Fig.6.21 shows an instance of the point targeting assistance. The POI is the blue point on the liver surface. Fig.6.21.a depicts task execution without any fixture, whereas Fig.6.21.b shows some of the VF involved in the schema: the blue arrow represents the attraction force field, whereas the green cone represents the virtual wall. Applied magnification can be observed by comparing the two snapshots.

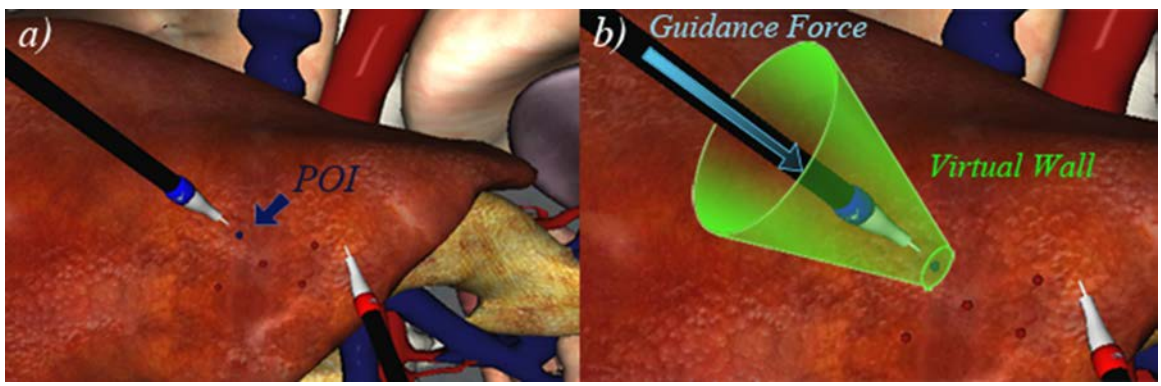


Figure 6.21: Application of the point targeting VFS.

6.5.2 Trajectory following

The second analysed surgical task is trajectory following that requires the surgeon to move from one point to another following a pre-planned path. The sensitive nature of some abdominal organs and tissues, along with the constraints imposed by the fulcrum point, limits robotic tools freedom and forces the tool motion on specific safe paths.

These robotic surgery restrictions lead to the identification of the principal requirements of a trajectory following task: safe trajectories and fine psycho-motor skills. VFS may help in teaching both of them: by helping the operator identifying sensitive tissues and by developing their visual-motor coordination.

Fig.6.22 depicts the VFS developed for the trajectory following schema. As can be noticed, it differs from the point targeting schema in the type of VF applied and in their behavior. The VFS flow starts with a distance dependent motion scaling that decreases the motion of the surgical tool as it moves away from the planned trajectory, preventing excessive tool deviations, but allowing free movements when following a path. The schema then separates into three parallel tracks. The tool is

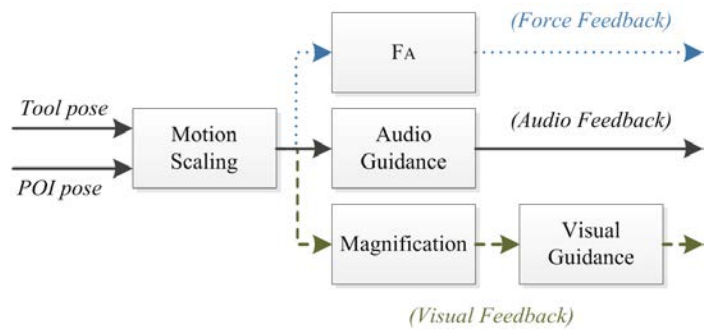


Figure 6.22: Trajectory following VFS block diagram

guided toward the trajectory by an attraction force, FA, that is computed as a non-linear function of the tool deviation, ensuring transparency when the tool is close to the desired trajectory. Audio feedback works as in the point targeting VF schema, guiding the surgeon to the planned trajectory. Visual feedback helps with a twofold action: it increases camera magnification when the tool is close to the trajectory and it also adapts its position focusing the view on the ensuing trajectory segment, thus easing task execution and increasing accuracy. In addition this VF over imposes the minimum distance vector between the tool and the trajectory point to guide the surgeon toward the trajectory, minimizing undesired deviations in terms of distance and recovery time. Fig.6.23 shows an example of this VFS. Workspace and trajectory are shown in Fig.6.23.a. Fig.6.23.b shows some of the applied VF. F_A represents the attraction force towards the trajectory. The small cone represents the position of the tool as if no motion scaling were applied (motion reduction is the difference between the tool and the cone). The minimum distance vector between the current trajectory point and the tool tip is also shown.

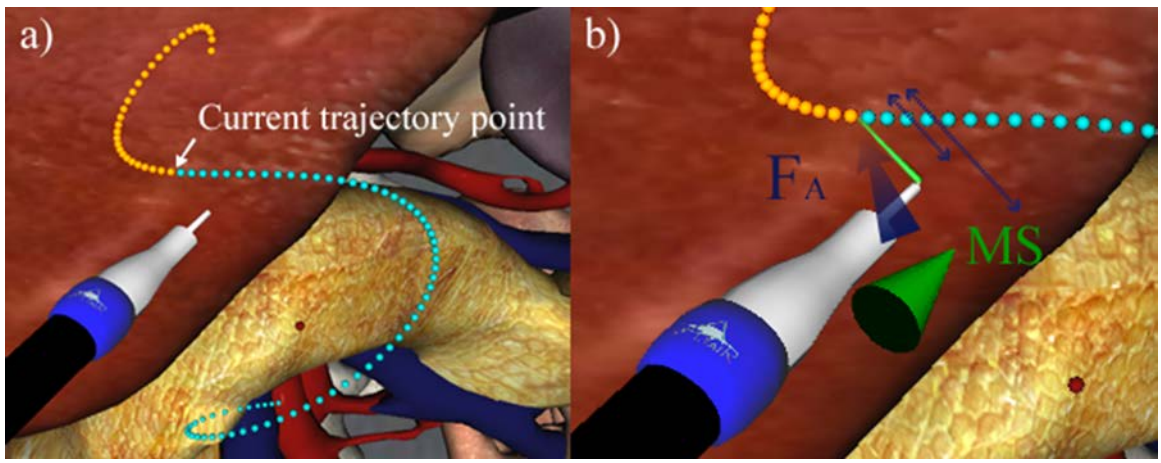


Figure 6.23: Trajectory following VFS applied in the VA simulator

The use of different sensory channels within a single scheme provides benefits to the operators' cognitive load, but it also introduces some methodological problems. The evaluation of the performance of an operator while he/she is supported by several VF doesn't allow, either to clearly distinguish the contribution of each VF, or to identify the effect of their interaction. The superimposition of VF effects due to their integration in a complex schema, in fact, may not obey the linear superimposition principle.

To clearly identify the benefits and drawbacks of each single assistance tool, all the VF have been individually defined and integrated into a surgical simulator. This provides an experimental set-up that offers the operator all the VF individually and that produces useful data for the evaluation of each VF benefits. The knowledge of characteristics and effects of a single VF will help in the adaptation of current VFS and in the development of more effective schemas.

6.5.3 Conclusions of VFS from the experimental results

The use of several VF generating VFS could disturb the analysis of the influence of each VF. The used experimental procedure was based on a three step experimental phase. The first one consists in evaluating individually each VF, as well as asking the subjects, once finished the experiment and trained on the use of each VF, to

propose different VFS. The second one was the development of different VFS and their implementation on the VA simulator. And, finally, the third part, still not developed, consists in the test and analysis of the use of these VFS. The data collected in the first phase will be used to refine and test the proposed VFS and the design of new VFS.

Some conclusions about VFS can be extracted from the data collected at the end of the experiment. The first one is the amount of VFS: 16 different VFS were proposed in a total of 42 proposals. This reinforces the idea of generating a set of individual VF easily configurable by the operator. The selection of each VFS depends on the task itself and on the operator's preferences. With the subtask division, a complex task (e.g. a complete surgery) can be divided into several subtasks. Each task may demand a different VFS depending on the concrete requirements.

Fig.6.24 presents all the VFS proposed by the subjects. Mag is the most selected VF by the surgeons when proposing VFS, followed by AG. One of the reasons of selecting Mag and AG is because surgeons are used to work with them in real live. FF is the third most selected VF. From the subjects opinion, haptic feedback always received positive comments and seems to be an attractive aid. VG and MS are significantly less selected as a part of VFS. The selection of components for a VFS is congruent with the evaluation of the individual VF described in Sec.6.4.4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	Total times selected in VFS	% times selected in VFS
VG		•			•	•			•				•		•							•					•				•			•	•	•						•	13	30,95
AG	•	•			•	•			•	•		•	•	•		•	•	•	•	•		•		•				•	•	•	•		•	•	•				•	•	•		26	61,90
MS	•		•								•				•							•	•									•						•	•		•		10	23,81
MAG			•	•		•	•	•		•	•	•		•		•	•			•	•	•	•	•	•	•		•	•	•	•	•	•			•	•	•				•	30	71,43
FF		•	•	•	•		•	•										•	•	•	•	•	•	•	•	•	•	•			•	•					•		•	•			23	54,76

Figure 6.24: Table with the VFS proposed by the surgeons during the experiment

Concerning the suggested VFS, two schemas, AG+Mag and Mag+FF, are the most proposed ones. AG+Mag was proposed 8 times and Mag+FF 7 times. The first VFS includes the two tools that surgeons are used to: variable endoscopic view (in position and magnification) and sound signals transferring information. The second one includes the magnification, but changing the auditive channel by haptic feedback.

The amount of force suggesting or limiting the movements of the master device was also a discussion topic during the final interviews with the surgeons participating in the experiments. VG+AG is another schema preferred by the surgeons. The combined use of both mitigates the drawbacks of each one: occlusions in VG and the non directional aid provided by AG.

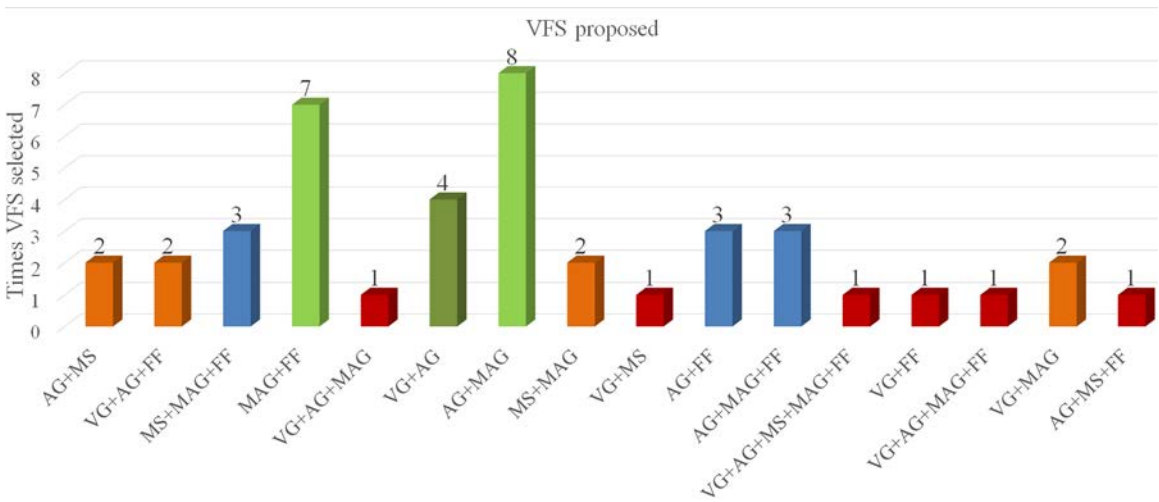


Figure 6.25: Graphic with the repetitions of VFS proposed by the subjects

Chapter 7

Conclusions and Future Developments

7.1 Conclusions and Contributions

The use of multiple robots working cooperatively in a redundant way has proved to offer new possibilities in the execution of complex or dextereous tasks in dynamic workspaces. The aim of MRCP is to increase the range of applicability of teleoperated systems by means of the automatic cooperation of multiple slave robots which, controlled by a human operator, act as if they were a unique robot. The result of the proposed methodology is an improved teleoperation architecture in terms of reachable workspace (volume, manoeuvrability and accessibility) and dexterity, thus widening its range of applicability. This approach allows human operators to focus their attention on the ongoing task more than on the teleoperated robots, as demonstrated in various experiments.

The first step of the research was oriented to develop a general purpose robotic platform that allows the control, simulation, test and evaluation of robotic set-ups. Jointly with this process, the need of a proximity queries library, specialized in robotic environment arose. Therefore, the Robotic Proximity Queries package, RPQ, was developed which resulted to be a useful tool and solution. The RPQ simulation platform was published in [153, 154]. RPQ has been used to develop different testbeds for many different research projects in the laboratory, like [194–196], in which

the research was oriented to improve the 3D perception and hand-eye coordination of surgeons in RMIS. RPQ was used to control the test-bed (a 6 DoF mouse controlling a robot with the RMIS constrained movements) and to record and verify the obtained results.

RPQ has enabled the implementation of MRCP and the development of different set-ups with several robots and master devices. MRCP has been developed as a high level layer of RPQ, integrating evaluation indices, control algorithms (robots evaluation, robot selection and action planner algorithms) and the definition of the different experimental robotic set-ups to test MRCP. The real-time and modularity requirements have been satisfactorily achieved with MRCP. However, current implementation of MRCP can be improved to obtain more efficient computation results. A first option is to include newer proximity queries libraries, like *Chai3d* [1] that integrates haptic simulation, deformable models and rigid body dynamics, operating in real-time. New robotic platforms have been developed, like *ROS* [2]. Several simulators have been developed using ROS like STDR or Gazebo. A second option is to upgrade the current RPQ version, using more efficient algorithms, using graphic process units, GPUs, and parallelizing the computation of robot evaluation indices and proximity queries. However, RPQ has demonstrated its usefulness in several projects and has an efficient minimum distance computation engine improved with the added robotic layer. RPQ computation performance can be improved with parallel programming.

After this initial phase, the research focusses on the development of the MRCP concept. First, establishing the robots suitability evaluation criteria and the methodology to define a common metrics to compare them, as the evaluation criteria belong to different evaluation spaces. And second, developing the control architecture and algorithms that enable the automatic internal control of MRCP. The resulting MRCP was conceived as an open platform that enables its use in many teleoperation set-ups, allowing the introduction of new evaluation indices and controlling all type and number of slave robots without altering the control architecture. The task-oriented control

[1] Chai3D is an open source haptic framework. www.chai3d.org

[2] Robot Operation System, ROS, is a flexible framework for writing robot software. http://www.ros.org/

paradigm was introduced as teleoperation interface, enabling the successful execution of tasks. Different experiments demonstrated the internal behaviour of MRCP. The results of these experiments were published in [130, 197].

The computation of the robot evaluation indices presented two main challenges. First, identifying the potential factors that can prevent a robot to continue with the execution of a teleoperated task. Second, defining an evaluation methodology independent of the robots kinematics and compatible with all the evaluation indices codomains. This methodology has satisfied the evaluation requirements. The evaluation indices were adapted to deal with the uncertainness associated with trajectory prediction but, it was not possible to find closed form solutions for those evaluation indices that depend on the Jacobian matrix like, for instance, the Anisotropic Dexterity Index. A discrete solution was presented for these cases. An interesting research challenge to solve is the closed form solution for them.

Various experiments were developed to test the internal control of MRCP and validate its performance. The obtained results indicated a noticeable improvement of operators performance in all the measured parameters: execution time, economy of movement and different quality measures.

These experiments demonstrated that task-oriented teleoperation eased the execution of tasks decreasing execution time and generating shorter and direct trajectories. Task transfer (needle transfer in the experiments) is a challenging process that requires dexterous and precise movements of the teleoperated robots, and 3D spatial perception (even more when using a 2D camera to visualize the remote workspace). Using classical bimanual teleoperation, a considerable number of subjects had problems in the execution of a correct needle transfer and required several trials up to its correct execution. In consequence, requiring long execution time and providing poor economy of movement and quality. With automatic task transfers, teleoperation becomes smoother and safer, with shorter execution times, shorter trajectories and higher quality. However, the proposed combined teleoperation system MRCP+Cam, 5.3, did not present the expected results compared with MRCP with fixed camera position: the subjects did not improve their results. A feasible explanation is that

the subjects had to learn how to manage with two different master devices and use them simultaneously, understand how the camera positioning can help to improve teleoperation and how to obtain the best workspace views during task execution.

From these experiments, it is possible to asseverate that MRCP improves operators performance in multirobot teleoperation systems. In terms of RMIS, patient safety increases with interventions with less pain and shorter times. The results of these experiments were published in [198] and [199][3]. The subjective evaluation presented a high level of coincidence with the numerical results. Most of the subjects felt comfortable with MRCP and the underlying task-oriented teleoperation paradigm.

The task-oriented teleoperation concept proposed in MRCP has been extended to Virtual Fixtures, VF. Task-oriented VF provide the operator with a set of virtual aids in accordance to the MRCP task oriented paradigm. Task oriented VF are the result of the combined use of different single VF acting together. This extension of the MRCP concept to VF was motivated by the need of providing new assistance to surgical training. For this reason, a trajectory following task using a virtual surgical simulator was implemented. It enabled the validation of these task-oriented VF. Two main conclusions were extracted from these experiments: first, VF demonstrated their usefulness, improving teleoperation in several parameters. Second, VF are a valid tool for training, shortening the learning curves, [200, 201].

7.2 Future Research

The research developed during this thesis has opened several new fields and has left some topics for further improvements. First, MRCP control algorithms can be computationally improved, enabling the introduction of more robots and evaluation indices taking into account the real-time computation restriction. Second, the development of an improved MRCP surgical test bed to enable the execution of complete surgical complex task. Third, the development of task-oriented complex Virtual Fixtures schemas that help the operator in the execution of teleoperation. Finally, in order to

[3]In review process for ICRA 2016 conference

test and validate the improvements, a new benchmark for MRCP will be established: a suture of an uterine myoma.

7.2.1 Improvement of MRCP control algorithms

Analysing the MRCP control from the performance point of view, its main weakness is the requirement of real time computation, which limits its scalability. Two bottlenecks were detected in the control process. First, each new robot added in the slave station increases the computational load during robot evaluation and robots actions computation processes. Second, every evaluation index increases the required time for a complete evaluation process, decreasing the maximum evaluation frequency. Although the current MRCP algorithms work in real-time for configurations of up to four 6 DoF robots and four evaluation indices, immediate future work will be focused on improving the MRCP computational performance to enable the inclusion of more robots and evaluation indices. The improvement will be based on parallelizing the control algorithms using Graphical Processing Units, GPUs, and programming with platforms like CUDA [4].

Some initial work of parallelization of the robot evaluation process is in development. A first sequential algorithm computes, for each robot, all the evaluation indices, transforms the evaluation results into time space, selects the TExec of each robot and, finally, computes the ordered list of robots suitability $(RSel, RCan_1, ..)$. Its computational time complexity is expressed as 7.1.

$$(m * n)(t_i + t_{tr}) + m * t_{TExec} * t_{ROrd} \tag{7.1}$$

where each t corresponds to a computation time: t_i the evaluation index, t_{tr} space transformation of every evaluation index, t_{TExec} the required time to determine the $TExec$ of every robot and t_{ROrd} is the required time to determine the order of each robot in the priority list.

[4]Compute Unified Device Architecure, CUDA, is a Compute Unified Device Architecture,[1] is a parallel computing platform and application programming interface (API) model. https://developer.nvidia.com/cuda-zone

The new parallel algorithm exploits the independence of the evaluation indices: Every evaluation index is independent from the rest of indices and robots and, consequently, their computation can be parallelized. This process requires $n * m$ computation units. Fig.7.1 shows a block schema of the parallelization process and its associated complexity is expressed as 7.2.

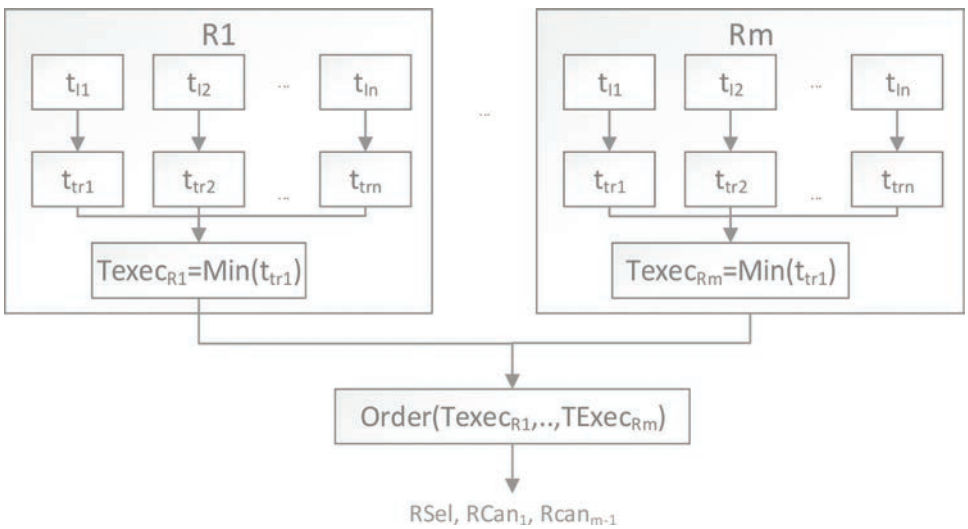


Figure 7.1: Parallelization of MRCP robot evaluation process

$$t'_i + t'_{tr} + m * t_{TExec} + t_{ROrd} \tag{7.2}$$

where t'_i and t'_{tr} include the required time to upload the data to the computation unit, process it and return the result.

Next step to improve the control algorithm performance is the parallelization of the computation of the minimum distance required to obtain the CR index and to compute the new robot movements. To simplify the computation complexity, let's assume that all robots have the same number of links, n, the number of robots is m and t_d the mean of required time to solve a minimum distance computation. Then, the complexity associated with this process is 7.3.

$$C_m^2 * n^n * t_d = \frac{m!}{2!(m-2)!} * n^n * t_d \tag{7.3}$$

The minimum distance queries between each pair of links is independent from the rest, enabling parallel computation. The number of minimum distance queries for the set-up used to evaluate MRCP (three robots of 6 DoF each) is $C_3^2(7^7) = 2470629$ queries. Incorporating a new 6 DoF robot, the number of queries increases to $C_4^2(7^7) = 4941258$. Current medium-level graphic cards have between 512 and 2048 GPUs, which can drastically reduce the required computation time, 7.4, where k is the number of GPUs .

$$C_m^2 * n^n * t_d/k + C_m^2 * n^n * (t_d \mod k) \tag{7.4}$$

7.2.2 Development of an improved MRCP surgical test bed

The experiments pointed out the need of improving the teleoperation set-up with better laparoscopic tools in terms of grasping force, types of tools and number of DoF. Several set-up improvements have been planned for the immediate future development.

An experimental test bed has been implemented using three 6 DoF robots equipped with conventional laparoscopic tools (including two endo-graspers and a laparoscopic camera). This test bed has been useful to develop and test the proposed MRCP. The current set-up presents some limitations that prevent the execution of more complex surgical tasks. The used laparoscopic tools have less DoF's and grasping forces than those required for a correct RMIS task execution and a deformable shank that disturbs the TCP positioning. To improve the teleoperation set-up, new robotised surgical tools must replace the current ones, which have only one DoF. The tool that is currently in development is a roll-pitch-roll mechanism plus a griper. There is an open research field in the laboratory developing a new tool that fits with the requirements. Fig.7.2 shows the kinematics schema and a picture of the end effector of the tool. This new configuration will require to introduce redundant robot control algorithms to manage these redundant DoF of each robot.

Additionally, third 6 DoF robot will be included in the MRCP set-up to improve the performance of MRCP in terms of workspace and manoeuvrability. The resulting

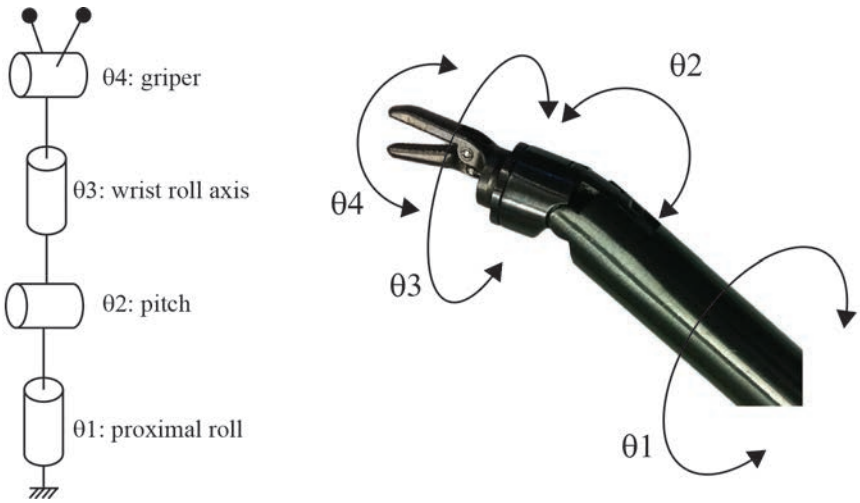


Figure 7.2: Robotised roll-pitch-roll laparoscopic tools

test platform will be composed of three robots for the MRCP plus an extra robot controlling the laparoscopic camera. This set-up configuration opens the possibility of creating a more flexible and powerful development platform.

To test the resulting new experimental set-up, a benchmark task will be defined: a complete RMIS suture of a uterine myoma.

7.2.3 Experimental study of complex task-oriented VF schemas in MRCP

Multiple experiments in the literature have demonstrated the benefits of Virtual Fixtures in teleoperation. In this thesis the VF concept has been extended to task-oriented VF schemas. Using a virtual abdominal simulator several single VF were presented and their performance analysed from several experiments. Jointly with these experiments, the subjects suggested some VF schemas to be implemented and tested on both, the Virtual Abdomen simulator and on MRCP. The first step of this new research will be the definition of VF as an algebraic group. This process will enable the strict mathematical definition of VF, as well as the definition of a framework for their further design and analysis. The current MRCP set-up is equipped with the

required tools to develop VF: each robot has a force sensor to measure the interaction forces and torques of the tools and the workspace among other workspace sensors, the master devices provide haptic feedback and RPQ can generate visual augmented reality and auditive signals.

7.2.4 New benchmark: RMIS suture of an uterine myoma with MRCP

MRCP was conceived with the aim of providing a complementary tool for those tasks that require an extended teleoperated robot, by means of automatic robot cooperation and a task-oriented teleoperation control and interface. A first experiment of complete stitching demonstrated the usefulness of this approach, enabling the operator to switch between both teleoperation paradigms: standard bi-manual teleoperation and task-oriented MRCP. The next step is to improve the integration of both schemas, automating, if convenient, the selection between operative modes. This work will be based on the division of complex tasks in a set of sub-tasks with different teleoperation configurations (standard vs task-oriented). A step further is the automatic integration of different VF that can improve the teleoperation results. A benchmark will be fixed: a complete laparoscopic suture of an uterine myoma.

7.3 SurgiTrainer: A surgical training spin-off

The experimental phase of the thesis has required the direct contact with a high number of surgeons. In general, the medical community is open to express their technological needs. Resulting from the knowledge and experience acquired during this interaction different proposals arose. The need of establishing a standardized formation and accreditation program for minimally invasive surgery techniques has resulted in a spin-off company from Hospital Sant Pau, the Institute for Bioengineering of Catalonia and the Technical University of Catalonia. The spin-off, named SurgiTrainer, is developing different surgical simulators based on the combined use of a physical

model and the required technology to record, analyse and interact with the users. Two patents have been presented as result of the research. The European Society for Gynaecology Endoscopic Surgery, ESGE, is a key partner of the company and both are developing a training and accreditation program. The simulators integrate the courses, methodologies and practices suggested by ESGE.

Appendix A

Robotic Proximity Query Package: RPQ

A.1 Introduction

In robotics, one of the most important problems to be solved is collision avoidance between a robot and its environment (other moving robots or obstacles). The control of a robot must perceive the potential collisions and react before they occur. Path planning techniques, [202], demand from high computation capacity and efficient proximity and collision computation engine. These computational requirements increase when there is no path planning and the collisions must be computed in real time; for instance, in human-robot interaction or teleoperation. The use of Virtual Fixtures, VF, requires an intensive use of minimum distances, collisions and depth penetration computation between geometrical models. This computation allows determining the behaviour of the masters haptic feedback generation. Given these premises and, following the requirements of the robotic laboratory where this work has been developed (specialized in human-robot interaction and teleoperation), the need of a proximity queries package arises. A new library, the Robotic Proximity Queries package, RPQ, has been developed. This library, initially specialized only as a proximity queries computation, has evolved to a robotic simulation framework, where dynamic robotic scenarios can be created, including robotic arms and obstacles. The RPQ library has been presented in [153, 154] and as part of the set-up for research development

in [194, 195]. Fig.A.1 illustrates the application of RPQ to different fields of robotic surgery.

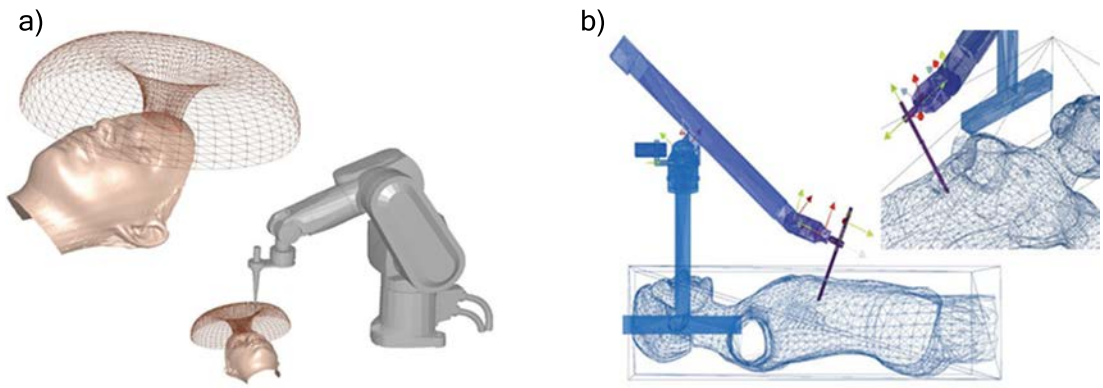


Figure A.1: RPQ applied in several surgery fields: a) Virtual fixture in a cutting bone surgical application, b) Study of a new robot for laparoscopic procedures

A.2 RPQ library

The goal of the Robotic Proximity Queries (RPQ) library is to offer an easy, modular and fast proximity query package oriented to robotics. In the initial phase of RPQ development, two options were observed: program a new collision library or specialize and optimize an existing one. After a wide review of the existing open source collision packages, presented at the end of the chapter, A.2.2, the use of an existing library result the most adequate solution. The Proximity Queries Package, PQP, was selected from a wide set of general purpose proximity query packages. The criteria used to choose PQP as the best candidate for the development of RPQ were:

1. Types of proximity queries available.

2. High performance on proximity queries.

3. Ability to use geometrical models based on triangulated meshes of points.

4. Lack off restrictions on possible geometrical models.

5. Open source library.

6. Easy API.

The PQP library has been developed by UNC Research Group on Modelling, Physically-Based Simulation and Applications and offers three different kind of queries:

- Collision detection: detecting whether two models overlap, and optionally, give the complete list of overlapping triangle pairs.

- Distance computation: computing the minimum distance between a pair of models.

- Tolerance verification: determining whether two models are closer or farther than a given tolerance distance.

RPQ Class description

RPQ has been implemented in C++ language and its graphical interface has been developed using OpenGL. The RPQ library can be easily integrated into any software application. The library interface allows non expert programmers to use it in an easy manner. The graphical interface is a separate module, allowing the programmer to decide whether using it or not. Fig. A.2 shows the integration of the library and its graphical interface into a generic application.

Scenario

Scenario is the workspace where the objects cohabit. Concerning its implementation, Scenario is a class that contains all the objects (Robots and generic objects), a global reference frame, and all the methods necessary to generate the proximity query.

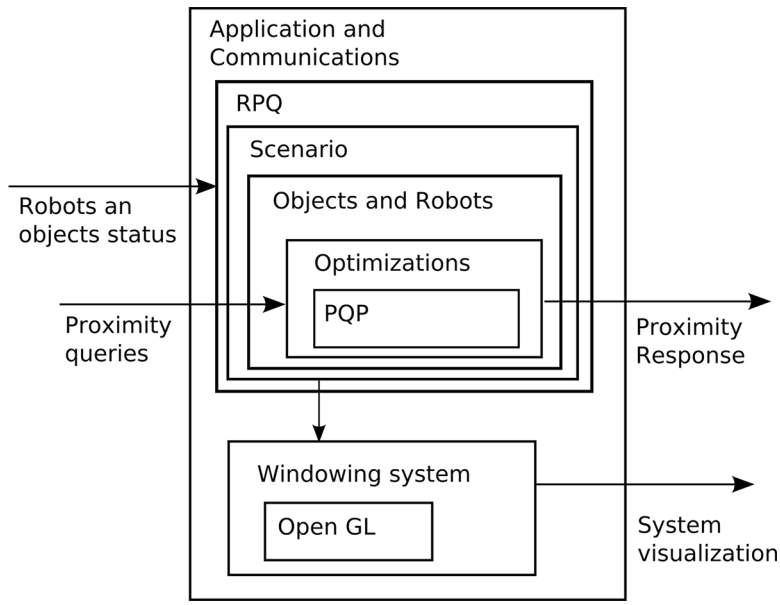


Figure A.2: Schema of integration of RPQ into a generic application.

Object

An Object is the minimum entity that exists in a Scenario. There are two types of Objects: simple and complex. A simple Object is represented by a geometrical model composed of a set of triangles referred to a frame tied to the Object. The Object has also a transformation matrix to refer itself to the world reference frame. A complex Object is an Object composed of a set of geometrical models with joints (rotational or prismatic) between them. Thus, a complex Object is an open kinematic chain composed of sub objects. The transformation matrix M_i refers subobject_i to subobject_{i-1}. The transformation matrix M_0 refers the object base (subobject_0) to the world. The object stores its own geometrical model. Concerning its implementation, an Object is a class containing its geometrical model, the transformation matrix and a set of methods to position and to orient itself in space. This class also contains methods to calculate the different detail representations of its geometrical model.

Robot

A Robot is a particularization of a complex Object where each of its links is represented by a simple Object. A Robot has a set of functions to make a complex Object as similar as possible to a real robot. For instance, the spatial relationship between links is described using the Denavit-Hartenberg notation. Direct and inverse kinematics can be calculated considering the robots own restrictions (joint limitations, configurations, etc). Concerning implementation, the class Robot is derived from the class Object. Robot adds all the functions that are necessary to control a robot. For instance joint positioning of a robot (direct kinematics), position and orientation of its tool center point (inverse kinematics), change of the robot configuration, joints overshoot ... These added functions with respect to an Object are very helpful when a new robot is created or used in robotic applications like simulators, path planners, etc.

A.2.1 Improvements: Robotics environment specialization

The set of optimizations introduced by RPQ to the generic proximity queries engine in which is based, PQP, are based on the specificities of the robotic field where it is applied. PQP is a generic package that does not use the knowledge of the object's kinematics. In contrast, RPQ is oriented to robotics, and the knowledge of robot's kinematics is the base of the optimizations. RPQ is designed to answer proximity queries between two robots or between a robot and any kind of rigid object.

RPQ introduces a high level layer that optimizes the proximity queries to be solved by the computation engine. Three optimizations are introduced to resolve and improve the proximity queries, decreasing the use of the force brute computation. These optimization strategies are designed to minimize the amount of proximity queries solved by the PQP computation engine; and are based on generating an intelligent order of these queries. First, multi-resolution objects representation generates simplified geometric models to decrease the amount of triangles involved in the queries. Second, the order of queries is based on the probability of success. Finally, Collision

Matrix acts as a filter to discard generating queries which result can be discarded a priori.

A test bed has been developed to evaluate the performance of the proposed optimizations. It consists in a couple of virtual robotic arms (simulation of Staübli RX60B) that are placed one in front of the other in a empty scenario. The distance between the robots varies from 400mm to 800mm and with an increment of 100mm at each experiment. The geometrical models used for the test are from high resolution (composed of 23012 triangles each). A total of 9216 different joint positions are used to generate robot configurations to be tested (if there's a collision or not). This test allows the study of the dependency on the performance of the proposed improvements in terms of the probability of collision. Fig.A.3 shows the percentage of collisions and not collisions in each robot distance trial.

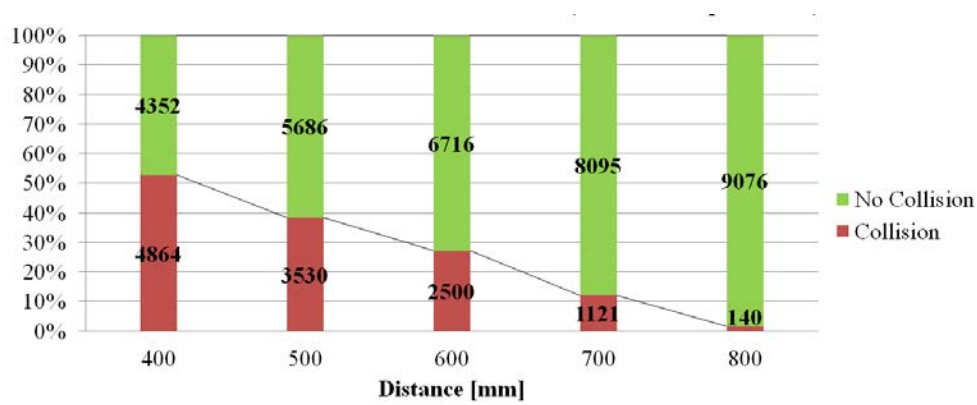


Figure A.3: Percentage of collisions and not collisions for different distances of the robots.

Table A.1 contains the statistics obtained in the experiments concerning all the queries and the amount of collisions and not collisions depending on the distances between the robots.

Table A.1: Statistics of the test

Distance	Total Queries	Collisions		Not Collisions	
		Total	%	Total	%
400mm	9216	4864	52,78	4352	47,22
500mm	9216	3530	38,30	5686	61,70
600mm	9216	2500	27,13	6716	72,87
700mm	9216	1121	12,16	8095	87,84
800mm	9216	140	1,52	9076	98,48

Multi-resolution representation

Designed to discard collisions, the Multi-resolution representation exploits the relationship between the complexity of the geometrical models and the query computational cost (the lower the number of triangles of the geometric model is, the faster the collision queries are executed). Objects (robot links and other objects inside the workspace) can be represented in several different resolution levels. The basis of this optimization is the use of the simplest representation models (minimum number of triangles) to discard rigid objects collisions. Robots are represented in several resolution levels (usually three or four) and the rest of the simple objects (single rigid body with no mobile joints) by two or three levels. The typical four resolution levels used in the robots are:

- L1: the highest resolution level, which is the geometrical model itself.
- L2: the convex hull of each link.
- L3: the oriented bounding box (OBB) of each sub object in which a complex object is divided in.
- L4: the lowest resolution level is the bounding box of the whole complex object. This last level is only applied to complex objects and is computed every time

one of the joints changes its position.

The convex hull of each link, L2, has been used in several applications (e.g. to determine the Collision Time and to compute the repulsion vectors for robot actions computation in MRCP). The convex hull of each link offers a good ratio between the resolution of the representation and the amount of triangles forming the model. Fig. A.4 illustrates the simulated robot in three resolution levels: the geometrical model of each link (L3), the OBB of each link (L2) and the whole bounding box(L1).

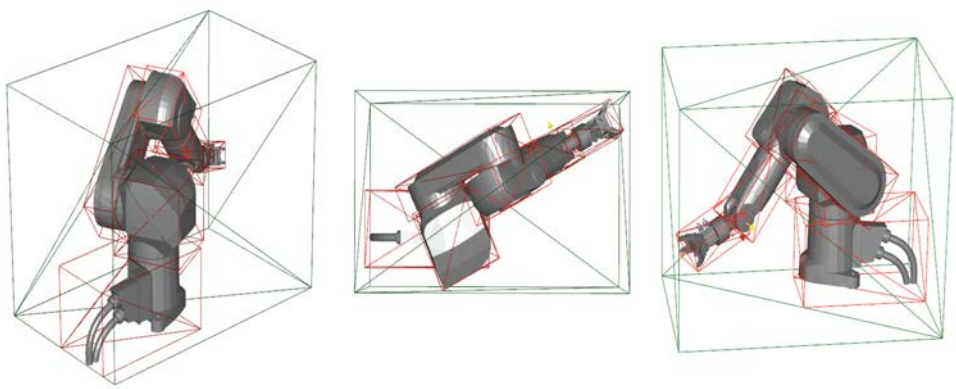
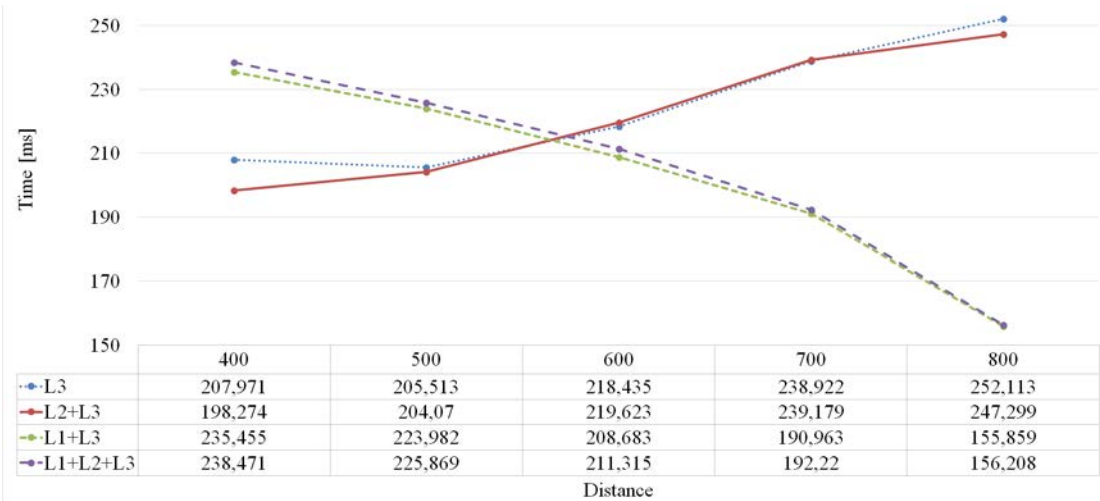


Figure A.4: Robot with three resolution level representation: L1,L3 and L4

The multi-resolution optimization improves the proximities queries in two different situations. First, in applications where no high precision is required; for instance, when the precision of the OBB or the convex hull of each link is enough to determine if a collision is present or not. The second situation occurs when the different resolution levels are used in a complementary and sequential manner: when a collision query is performed, a low to high resolution level list of collision queries is generated. Starting with the lowest resolution level, queries are generated until any collision can be completely discarded. For instance, if a possible collision between two 6 DoF robots is studied, the first query is done between the bounding boxes of each robot. If the collision can not be discarded, then the bounding box of each link is used. If at this level collisions still can not be discarded, the geometrical models of each link are checked. The obtained results of using a combined multi-resolution can be observed in Fig.A.5.



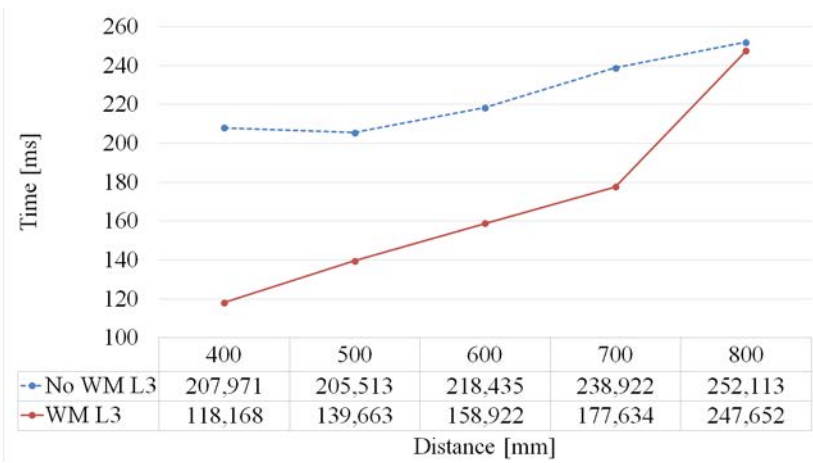
	400	500	600	700	800
L3	207,971	205,513	218,435	238,922	252,113
L2+L3	198,274	204,07	219,623	239,179	247,299
L1+L3	235,455	223,982	208,683	190,963	155,859
L1+L2+L3	238,471	225,869	211,315	192,22	156,208

Figure A.5: Computational time required to solve all the collision queries with different resolution levels.

Probabilistic collision queries order: the Weight Matrix

This optimization is designed to detect whether a collision occurs or not, but not on the number of them. The Probabilistic Collision Queries order assumes that the kinematics and the morphology of the robots are well known. Given these assumptions, the objective of the optimization is to find, as soon as possible, whether there is collision or not, minimizing the number of collision queries. The knowledge of the kinematics and the morphology of the robots give the possibility of assigning a collision probability to each link of the robot with respect to the rest of the obstacles present in the shared workspace. During execution time, these probabilities are automatically updated depending on the result of the collision queries: Probability increases in case of detecting a collision and decreases otherwise. Therefore, a weight matrix C is generated combining the probability of collision between each pair of objects in the workspace. Each component $c_{ij} \in C$ verifies $c_{ij} = P_i + P_j$ where P_i and P_j are the assigned probability of collision of Object_i and Object_j respectively. These weights determine the order of the collision queries, that is if $c_{ij} > c_{kt}$ the collision query between Object_i and Object_j is generated before Object_k and Object_t.

A simple way to set up the initial values of the collision probability to the links of a robot is to assign higher probability to those links that are farther in the kinematic chain, with respect to the base of the robot. Fig.A.6 shows the required time to solve the collisions with and without using the Weight Matrix.



	400	500	600	700	800
No WM L3	207,971	205,513	218,435	238,922	252,113
WM L3	118,168	139,663	158,922	177,634	247,652

Figure A.6: Computational time required to solve all the collision queries using the Weight Matrix.

Collision Matrix

The last optimization is based on minimizing the required collision queries by means of a matrix, the Collision Matrix, that reflects the possibility of a collision between two objects (robot links or simple objects). The basis of the optimization is the reduction of computational time to obtain a value on a matrix with respect to executing a collision query. The Collision Matrix is a binary matrix that indicates the possibility of collision between two objects. If the collision matrix points that a collision between two objects is impossible, its correspondent collision query is not performed. Of course, a matrix query is much less expensive than a collision query in computational terms. This optimization improves the performance of the system when a high number of collision queries are discarded by the Collision Matrix. This optimization improves the performance of the system when a high number of collision queries are discarded

by the Collision Matrix. Computationally, this condition can be expressed as (A.1), where n: Total number of queries, m: Queries resolved with the Collision Matrix, k: Queries resolved with the Query Collision, QC: Average time to solve a Query Collision and QM: Time to solve a query with the Collision Matrix.

$$n \cdot QC > m \cdot QM + k \cdot (QC + QM) \tag{A.1}$$

$$\text{with} \qquad n = m + k \tag{A.2}$$

The performance of the Collision Matrix has been studied using the same test designed for the rest of the optimization methods, but the distance of the robots has increased until 1200mm. Fig.A.7 shows the obtained results with Collision Matrix. The farther the robots are, the lower is the number of links that can collide, and therefore, the higher is the number of queries that are solved with the Collision Matrix.

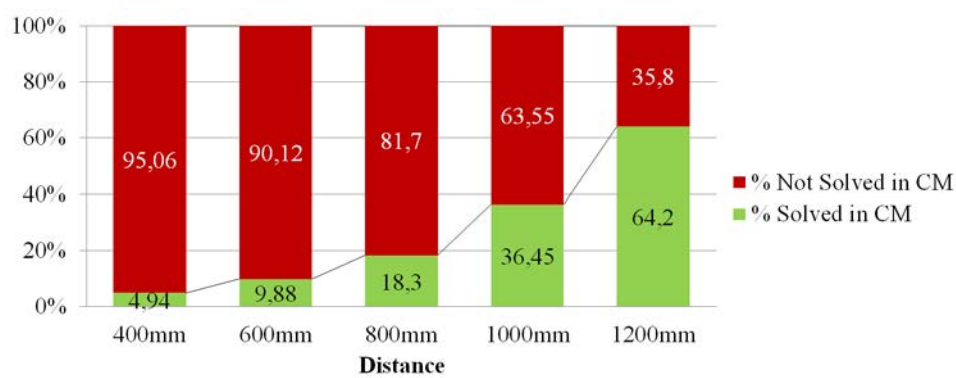


Figure A.7: Percentage of queries solved in Collision Matrix depending on the robot distance.

Proposed algorithm combining all optimizations

The reviewed optimizations improve the original PQP library performance in a stand alone manner. However, the combined use of all of them improves even more the global

performance of the application. Fig. A.8 shows the complete algorithm applying all three optimizations applied to a collision query between two robots. First, a query collision between the whole bounding box of both robots is performed. If at this level the collision cannot be solved then it is necessary to study collisions among the whole set of links of both robots. The order in which these queries must be performed is given by the Weight Matrix. The query finishes as soon as a collision appears between a pair of links either in the second or third level, or when all pairs have not reported any collision. For each pair of links, the second and third representation levels are studied consecutively, so if a collision is detected in the second level, the third level has to be studied as well.

A.2.2 Collision libraries

In what follows,, a review of the studied collision libraries during the design phase of RPQ library is done. This review is based on the GAMMA study, presented in their website, [203], where a review of the most relevant features and performance of all available collision packages is done. The reader familiar with collision libraries will miss some new packages appeared after the development of RPQ (during years 2004-2005). The studied packages (DEEP, SWIFT, PIVOT, H-COLLIDE, I-COLLIDE, V-COLLIDE, RAPID, PQP and IMMPACT) are reviewed and their most relevant aspects pointed.

DEEP

DEEP, Dual-space Expansion for Estimating Penetration Depth, is a library designed to determine the penetration depth of solid bodies. DEEP is based on an incremental algorithm that estimates the penetration depth between convex polytopes along with the associated penetration direction. Penetration depth is defined as the minimum translation distance to make the interiors of two polytopes disjoint. DEEP finds a locally optimal solution by walking on the surface of the Minkowski sums. Furthermore, DEEP is designed to fully utilize the frame-to-frame motion coherence in the

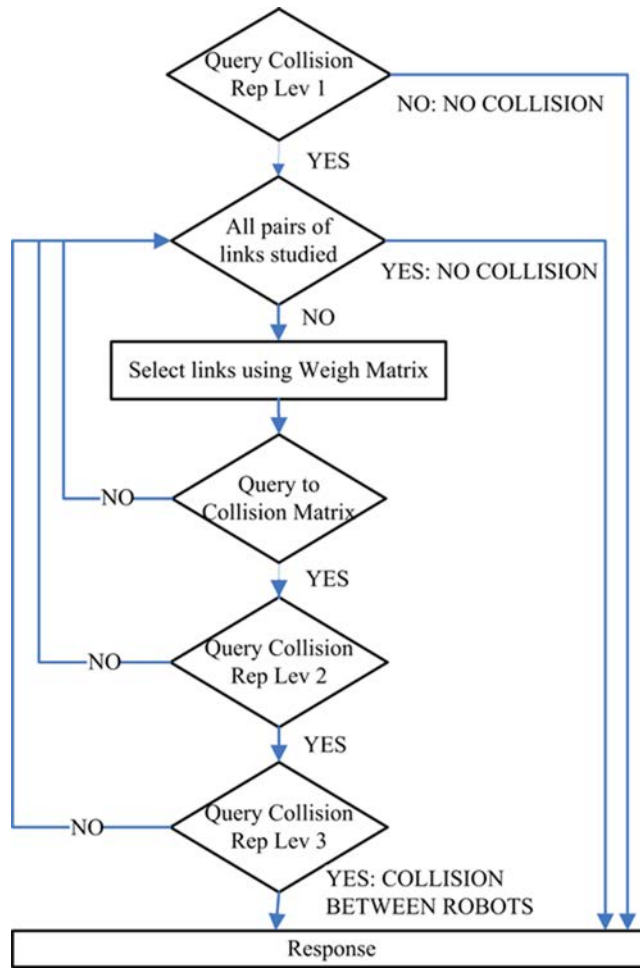


Figure A.8: Algorithm for collision detection between two robots using all three optimizations.

environment. In [204, 205] further information about DEEP can be found.

SWIFT, SWIFT++

SWIFT and SWIFT++, [206], provide proximity queries such as intersection detection, exact and approximate distance computation, and contact determination of three-dimensional objects undergoing rigid motion. The allowed objects are convex polyhedra or composite objects constructed from convex pieces. SWIFT uses a sweep

and prune test to remove uninteresting pairs from further consideration and then uses an improved Lin-Canny closest features algorithm to find the answer for the "close" pairs of objects. The updated version, SWIFT++, provides proximity queries such as intersection detection, tolerance verification, exact and approximate distance computation, and contact determination of general three-dimensional polyhedral objects undergoing rigid motion. SWIFT++ operates by first computing a surface decomposition of the input models. The pieces are then grouped hierarchically using convex hulls. A pair of bounding volume hierarchies (BVHs) are tested using an improved Lin-Canny closest feature tracking algorithm. SWIFT++ uses the SWIFT core for the overlap test between convex pieces in the BVHs.

PIVOT

PIVOT, Proximity Information from Voronoi Techniques, is a 2D proximity engine. The engine is based on hardware accelerated multi-pass rendering techniques and distance computation to perform a variety of proximity queries between objects. The supported queries include detecting collisions, computing intersections, separation distances, penetration depth, and contact points with normals. PIVOT uses an hybrid geometry- and image-based approach that balances CPU and graphics subsystems that allows the use to customize settings for their particular application. PIVOT handles 2D simple closed polygons. The polygons can be non-convex however, saving the application the problem of decomposing objects into triangles. PIVOT is specially designed for these applications that require detailed information about the proximity even when objects are penetrating, such as penalty-based simulators. Extended information about 2D and 3D proximity algorithms can be found in [207, 208].

H-COLLIDE

H-Collide, [209–211], is a framework for fast and accurate collision detection for haptic interaction. It consists of a set of algorithms and a system specialized for computing contacts between the probe of the force-feedback device and objects in the virtual

environment. To reach the performance requirements for haptic interaction, an approach that specializes many earlier algorithms for this application are used. The framework uses:

- Spatial Decomposition: It decomposes the workspace into uniform grids or cells, implemented as a hash table to efficiently deal with large storage requirements. At runtime, the algorithm can quickly find the cell containing the path swept out by the probe.

- Bounding Volume Hierarchy based on OBBTrees: An OBBTree is a bounding volume hieachy and each node of the hierarchy corresponds to a tight-fitting oriented bounding box (OBB). For each cell consisting of a subset of polygons of the virtual model, we pre-compute an OBBTree. At run-time, most of the computation time is spent in finding collisions between an OBBTree and the path swept out by the tip of the probe between two successive time steps. To optimize this query, we have developed a very fast specialized overlap test between a line segment and an OBB, that takes as few as 6 operations and only 36 arithmetic operations in the worst case, not including the cost of transformation.

- Frame-to-Frame Coherence: Typically, there is little movement in the probe position between successive steps. The algorithm utilizes this coherence by caching the contact information from the previous step to perform incremental computations.

I-COLLIDE

I-COLLIDE, [212] works with convex polyhedra models. It exploits the special characteristics of convex polytopes to fast determine contact status. It also exploits temporal coherence, so that collision query times are extremely fast when the models are moved only a relatively small amount between frames. I-COLLIDE employs a similar "Nbody" processing algorithm as does V-COLLIDE. I-COLLIDE maintains the placements of all the models, and updates the potential contact pair list as the models placements' are modified. So, objects may be added and deleted from the managed

set where the client wishes. One of I-COLLIDE's great strengths is that it returns the distance between interesting pairs of objects (you get to decide beforehand what pairs are considered interesting). The SWIFT package provides the same functionality as I-COLLIDE and more. It is also faster and much more robust and should be used instead.

V-COLLIDE

V-COLLIDE is an "Nbody" processor built on top of the RAPID system. Once a client program tells V-COLLIDE where all the models are in world space, V-COLLIDE performs a fast sweep-and-pune operation to decide which pairs of models are potentially in contact, and then for each potential contact pair it uses RAPID to determine true contact status. V-COLLIDE remembers where all the models are, and some or all of the model placements by telling V-COLLIDE their new placements can be updated. V-COLLIDE also works with polygon soups, and reports only contact status (but not distance). In addition to updating the models' positions, the client program can also add or delete models from the collection being managed by the V-COLLIDE collision detection engine. Finally, V-COLLIDE also supports multiple independent collision detection engines.

RAPID

RAPID, [213] is the smallest and easiest to use package of all analysed ones. It woks with "polygon soups", which are just polygonal models which do not require any particular topological structure, such as forming a mesh or even a closed object. RAPID will accept a cloud of disconnected triangles as a model. RAPID does require that the models be composed of triangles (as opposed to quadrilaterals, for instance). Given two models and their placement within a world coordinate system, RAPID returns a list of the triangle contact pairs - where each contact pair is a triangle taken from each model. If the list it returns is an empty list, then the models do not touch. To process a pair of models, the client program must explicitly call a

collision procedure, passing those two models and their placements. Hence, RAPID does not perform "Nbody" processing - which is the determination of which pairs of a collection of models are in contact. RAPID only processes a specific pair of models upon an explicit command from the client program. RAPID is designed for small or moderate number of complex polygonal models on which the client programs are willing to make pair-processing queries explicitly.

PQP

PQP, [213,214] offers the same RAPID features (collision detection) but also provides support for distance computation and tolerance verification queries. Its API is similar to that of RAPID. It is applicable to general polygonal models and needs no topological information. Given two models, PQP supports a number of different queries. It makes use of swept sphere volumes as the choice of BV for distance queries. Furthermore, it allows the client program the flexibility of using more than one bounding volume for a given query. More details are available.

IMMPACT

IMMPACT, [215] is designed as an approach for interactive collision detection and proximity computations on massive models composed of millions of geometric primitives. It addresses issues related to interactive data access and processing in a large geometric database, which may not fit into the main memory of typical desktop workstations or computers. The IMMPACT algorithm uses overlap graphs for localizing the "regions of interest" within a massive model, thereby reducing runtime memory requirements. The overlap graph is computed off-line, pre-processed using graph partitioning algorithms, and modified on the fly as needed. At run time, it traverses localized sub-graphs to check the corresponding geometry for proximity and pre-fetch geometry and auxiliary data structure. To perform interactive proximity queries, it uses bounding-volume hierarchies and take advantage of spatial and temporal coherence. IMMPACT uses as benchmark the interaction with a CAD model of a power

plant consisting of over 15 million triangles. IMMPACT is able to perform a number of proximity queries in real-time on such a model.

Bibliography

[1] D. Lee and M. Spong, "Bilateral teleoperation of multiple cooperative robots over delayed communication networks: Theory," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, April 2005, pp. 360–365.

[2] D. Lee, O. Martinez-Palafox, and M. Spong, "Bilateral teleoperation of multiple cooperative robots over delayed communication networks: Application," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, April 2005, pp. 366–371.

[3] R. Mohajerpoor, I. Sharifi, H. Talebi, and S. Rezaei, "Adaptive bilateral teleoperation of an unknown object handled by multiple robots under unknown communication delay," in *Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on*, July 2013, pp. 1158–1163.

[4] N. D. Do, Y. Yamashina, and T. Namerikawa, "Multiple Cooperative Bilateral Teleoperation with Time-Varying Delay," *SICE Journal of Control, Measurement, and System Integration*, vol. 4, pp. 89–96, 2011.

[5] H. Durrant-Whyte, N. Roy, and P. Abbeel, "Assisted Teleoperation Strategies for Aggressively Controlling a Robot Arm with 2D Input," in *Robotics:Science and Systems VII*, 1st ed. MIT Press, 2012, pp. 354–361. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6301030

[6] M.-D. Hua, T. Hamel, H. Rifai, P. Morin, and C. Samson, "Automatic Collision Avoidance for Teleoperated Underactuated Aerial Vehicles using Telemetric Measurements," INRIA," Rapport de recherche, 2010. [Online]. Available: http://hal.inria.fr/inria-00545662

[7] S. Salmanipour and S. Sirouspour, "Teleoperation of a mobile robot with model-predictive obstacle avoidance control," in *Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE*, Nov 2013, pp. 4270–4275.

[8] J. Y. C. Chen, E. C. Haas, and M. J. Barnes, "Human Performance Issues and User Interface Design for Teleoperated Robots," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1231–1245, Nov. 2007.

[9] C. E. Lathan and M. Tracey, "The Effects of Operator Spatial Perception and Sensory Feedback on Human-Robot Teleoperation Performance," *Presence: Teleoperators and Virtual Environments*, vol. 11, no. 4, pp. 368–377, Aug. 2002. [Online]. Available: http://dx.doi.org/10.1162/105474602760204282

[10] D. Hainsworth, "Teleoperation user interfaces for mining robotics," *Autonomous Robots*, vol. 11, no. 1, pp. 19–28, 2001. [Online]. Available: http://dx.doi.org/10.1023/A%3A1011299910904

[11] I. B. K. Manuaba, K. Taylor, and E. Widzyk-capehart, "Building Effective Teleoperation Interfaces Applications by Utilising Mixed Reality for Mining," in *22nd World Mining Congress & Expo 11 - 16 September 2011, Instambul Turkey*, 2011. [Online]. Available: http://wmc-expo2011.com/index.php

[12] T. Fong and C. Thorpe, "Vehicle teleoperation interfaces," *Auton. Robots*, vol. 11, no. 1, pp. 9–18, July 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1011295826834

[13] C. Hu, M. Meng, P. Liu, and X. Wang, "Visual gesture recognition for human-machine interface of robot teleoperation," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2, Oct 2003, pp. 1560–1565 vol.2.

[14] C. J. Bell, P. Shenoy, R. Chalodhorn, and R. P. N. Rao, "Control of a humanoid robot by a noninvasive braincomputer interface in humans," *Journal of Neural Engineering*, vol. 5, no. 2, p. 214, 2008. [Online]. Available: http://stacks.iop.org/1741-2552/5/i=2/a=012

[15] Y. Chae, J. Jeong, and S. Jo, "Noninvasive brain-computer interface-based control of humanoid navigation," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, Sept 2011, pp. 685–691.

[16] J. Abbott, G. Hager, and A. Okamura, "Steady-hand teleoperation with virtual fixtures," in *Robot and Human Interactive Communication, 2003. Proceedings. ROMAN 2003. The 12th IEEE International Workshop on*, Oct 2003, pp. 145–151.

[17] J. Abbott, P. Marayong, and A. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," in *Robotics Research*, ser. Springer Tracts in

Advanced Robotics, S. Thrun, R. Brooks, and H. Durrant-Whyte, Eds. Springer Berlin Heidelberg, 2007, vol. 28, pp. 49–64. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-48113-3_5

[18] R. Goertz and R. Thompson, "Electronically controlled manipulator. Nucleonics," *National Geographic*, no. 41, p. 163, 1954.

[19] E. S. Conkur and R. Buckingham, "Clarifying the definition of redundancy as used in robotics," *Robotica*, vol. 15, pp. 583–586, 9 1997. [Online]. Available: http://journals.cambridge.org/article_S0263574797000672

[20] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 13, no. 3, pp. 398–410, 1997.

[21] M. Shimizu, H. Kakuya, W. K. Yoon, K. Kitagaki, and K. Kosuge, "Analytical inverse kinematic computation for 7-dof redundant manipulators with joint limits and its application to redundancy resolution," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1131–1142, 2008.

[22] H. P. Xie, R. V. Patel, S. Kalaycioglu, and H. Asmer, "Real-time collision avoidance for a redundant manipulator in an unstructured environment," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 3, 1998, pp. 1925–1930 vol.3.

[23] A. D. Luca and L. Ferrajoli, "Exploiting robot redundancy in collision detection and reaction," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 3299–3305.

[24] H. Choset, M. Zenati, T. Ota, A. Degani, D. Schwartzman, B. Zubiate, and C. Wright, "Enabling medical robotics for the next generation of minimally invasive procedures: Minimally invasive cardiac surgery with single port access," in *Surgical Robotics*, J. Rosen, B. Hannaford, and R. M. Satava, Eds. Springer US, 2011, pp. 257–270. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-1126-1_12

[25] P. Neuzil, S. Cerny, S. Kralovec, O. Svanidze, J. Bohuslavek, P. Plasil, P. Jehlicka, F. Holy, J. Petru, R. Kuenzler, and L. Sediva, "Single-site access robot-assisted epicardial mapping with a snake robot: preparation and first clinical experience," *Journal of Robotic Surgery*, vol. 7, no. 2, pp. 103–111, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11701-012-0343-6

[26] M. Cianchetti, T. Ranzani, G. Gerboni, I. De Falco, C. Laschi, and A. Menciassi, "Stiff-flop surgical manipulator: Mechanical design and experimental characterization of the single module," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 3576–3581.

[27] D. Bruno, S. Calinon, and D. G. Caldwell, "Null space redundancy learning for a flexible surgical robot," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2014.

[28] R. Costa-Castello and L. Basanez, "Understanding workspace structure of multi-robot systems," in *Emerging Technologies and Factory Automation, 1999. Proceedings. ETFA '99. 1999 7th IEEE International Conference on*, vol. 1, 1999, pp. 129–135 vol.1.

[29] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulationA survey," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1340–1353, Oct. 2012. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S092188901200108X

[30] F. Caccavale and M. Uchiyama, "Cooperative manipulators," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, 2008, pp. 701–718. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30301-5_30

[31] M. Saha and P. Isto, "Motion planning for robotic manipulation of deformable linear objects," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 2478–2484.

[32] J. Krüger, G. Schreck, and D. Surdilovic, "Dual arm robot for flexible and cooperative assembly," *CIRP Annals - Manufacturing Technology*, vol. 60, no. 1, pp. 5–8, Jan. 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0007850611000187

[33] S. M. LaValle, *Planning Algorithms*. New York, NY, USA: Cambridge University Press, 2006.

[34] J. Yakey, S. LaValle, and E. Kavraki, "Randomized path planning for linkages with closed kinematic chains," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 6, pp. 951–958, Dec 2001.

[35] Y. Koga and J. C. Latombe, "On multi-arm manipulation planning," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, 1994, pp. 945–952 vol.2.

[36] F. Zacharias, D. Leidner, F. Schmidt, C. Borst, and G. Hirzinger, "Exploiting structure in two-armed manipulation tasks for humanoid robots," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 5446–5452.

[37] A. Edsinger and K. C. C., "Two arms are better than one: A behavior based control system for assistive bimanual manipulation," in *International Conference on Advanced Robotics, ICAR'07*, 2007.

[38] C. Edsinger, Aaron and Kemp, "Two Arms Are Better Than One: A Behavior Based Control System for Assistive Bimanual Manipulation," in *Recent Progress in Robotics: Viable Robotic Service to Human*, 2008, pp. 345–355.

[39] M. J. Hwang, D. Y. Lee, and S. Y. Chung, "Motion planning of bimanual robot for assembly," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, 2007, pp. 240–245.

[40] N. Y. Chong, T. Kotoku, K. Ohba, K. Komoriya, N. Matsuhira, and K. Tanie, "Remote coordinated controls in multiple telerobot cooperation," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 4, 2000, pp. 3138–3143.

[41] *Teleoperation of multi-robot and multi-property systems*, 2008.

[42] T. Fukuda and K. Kosuge, "Advanced telerobotic systems single-master multi-slave manipulator system and cellular robotic system," in *Teleoperation: Numerical Simulation and Experimental Validation*, ser. Eurocourses, M. Bequet, Ed. Springer Netherlands, 1992, vol. 4, pp. 195–208. [Online]. Available: http://dx.doi.org/10.1007/978-94-011-2648-9_12

[43] A. Codourey, M. Rodriguez, and I. Pappas, "A task-oriented teleoperation system for assembly in the microworld," in *Advanced Robotics, 1997. ICAR '97. Proceedings., 8th International Conference on*, Jul 1997, pp. 235–240.

[44] L. Sentis and O. Khatib, "Task-oriented control of humanoid robots through prioritization," *IEEE International Conference on Humanoid Robots*, 2004.

[45] ——, "Control of free-floating humanoid robots through task prioritization," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, April 2005, pp. 1718–1723.

[46] M. Nikolic, B. Borovac, M. Rakovic, and S. Savic, "A further generalization of task-oriented control through tasks priorization," *International Journal of Humanoid Robotics*, vol. 10, no. 03, p. 1350012, 2013. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0219843613500126

[47] Y. Kamiya, T. Asahi, S. Ando, and O. Khatib, "Task oriented control with constraints for industrial robot," in *Robotics (ISR), 2013 44th International Symposium on*, Oct 2013, pp. 1–6.

[48] J. Neumann and O. Morgenstern, *Theory of games and economic behavior*, 3rd ed. Princeton, NJ: Princeton Univ. Press, 1953. [Online]. Available: http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+021850372&sourceid=fbw_bibsonomy

[49] J. Liu, L. Sun, T. Chen, X. Huang, and C. Zhao, "Competitive Multi-robot Teleoperation," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 75–80.

[50] X. Huang, J. Liu, L. Sun, T. Chen, and W. Sun, "Tele-Lightsaber - A Kind of Competitive Teleoperation¡sup¿*¡/sup¿," *2006 6th World Congress on Intelligent Control and Automation*, pp. 9237–9241, 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1713788

[51] X. Huang, J. Liu, L. Sun, W. Sun, and T. Chen, "A Criterion for Evaluating Competitive Teleoperation Systemn," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, pp. 5214–5219.

[52] C. Zhao, J. Liu, Y. Li, T. Chen, and L. Sun, "Virtual Repulsive Force in Competitive Multi-Robot Teleoperation," in *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on*, vol. 2, 2005, pp. 15–20.

[53] T. L. Huntsberger, "Hybrid teleoperated/cooperative multirobot system for complex retrieval operations," in *Sensor Fusion and Distributed Robotic Agents*, P. S. Schenker and G. T. McKee, Eds., vol. 2905. SPIE, October 30, 1996 1996, pp. 11–17.

[54] J. J. Abbott, P. Marayong, and A. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," *Robotics Research*, pp. 49–64, 2007. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-48113-3_5

[55] D. Aarno, S. Ekvall, and D. Kragic, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 1139–1144.

[56] G. T. McKee and P. S. Schenker, "Human-robot cooperation for automated viewing during teleoperation," in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 1, 1995, pp. 124–129 vol.1.

[57] T. M. Sobh and D. Y. Toundykov, "Optimizing the tasks at hand [robotic manipulators]," *Robotics & Automation Magazine, IEEE*, vol. 11, no. 2, pp. 78–85, 2004.

[58] P. I. Corke, *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011.

[59] K. Ben-Gharbia, A. Maciejewski, and R. Roberts, "Examples of spatial positioning redundant robotic manipulators that are optimally fault tolerant," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, oct. 2011, pp. 1526 –1531.

[60] K. Miller, "Optimal design and modeling of spatial parallel manipulators," *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 127–140, February 01 2004.

[61] B.-H. Jun, P.-M. Lee, and J. Lee, "Manipulability analysis of underwater robotic arms on rov and application to task-oriented joint configuration," in *OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04*, vol. 3, nov. 2004, pp. 1548 –1553 Vol.3.

[62] M. Ishitsuka and K. Ishii, "Development of an underwater manipulator mounted for an auv," in *OCEANS, 2005. Proceedings of MTS/IEEE*, sept. 2005, pp. 1811 –1816 Vol. 2.

[63] K. Kawamura, H. Seno, Y. Kobayashi, and M. G. Fujie, "Pilot study on effectiveness of simulation for surgical robot design using manipulability," in *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE*, 2011, pp. 4538–4541.

[64] R. Konietschke, T. Ortmaier, H. Weiss, R. Engelke, and G. Hirzinger, "Optimal design of a medical robot for minimally invasive surgery," in *Jahrestagung der Deutschen Gesellschaft fr Computer- und Roboterassistierte Chirurgie (CURAC)*, 2003.

[65] L. Guilamo, J. Kuffner, K. Nishiwaki, and S. Kagami, "Manipulability optimization for trajectory generation," in *Proc.of the IEEE Int.Conf.on Robotics and Automation, ICRA*, 2006, pp. 2017–2022.

[66] H. Okubo, T. Matsudo, and N. Katsuda, "Reduction of control torque for space manipulators using dynamic manipulability ellipsoid," in *The 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2003.

[67] K. Ikeda, M. Minami, and Y. Mae, "Real-time trajectory tracking/quasi-optimal obstacle avoidance control of redundant manipulators," in *SICE-ICASE, 2006. International Joint Conference*, 2006, pp. 2828–2833.

[68] V. F. Munoz, I. Garcia-Morales, C. P. del Pulgar, J. M. Gomez-DeGabriel, J. Fernandez-Lozano, A. Garcia-Cerezo, C. Vara-Thorbeck, and R. Toscano, "Control movement scheme based on manipulability concept for a surgical robotic assistant," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 245–250.

[69] S. Lee and J. M. Lee, "Reconfiguration of a redundant manipulator with task-oriented manipulability," in *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, 1991, pp. 1239–1244 vol.2.

[70] L. Zlajpah and B. Nemec, "Kinematic control algorithms for on-line obstacle avoidance for redundant manipulators," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 2, 2002, pp. 1898–1903.

[71] H. Tanaka, M. Minami, and Y. Mae, "Trajectory tracking of redundant manipulators based on avoidance manipulability shape index," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, aug. 2005, pp. 4083 – 4088.

[72] M. Minami and M. Takahara, "Avoidance manipulability for redundant manipulators," in *Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings. 2003 IEEE/ASME International Conference on*, vol. 1, july 2003, pp. 314 – 319 vol.1.

[73] B. Tondu, "A theorem on the manipulability of redundant serial kinematic chains." *Engineering Letters*, vol. 15, no. 2, pp. 362–369, 2007.

[74] I. Gravagne and I. Walker, "Manipulability, force, and compliance analysis for planar continuum manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 3, pp. 263 –273, jun 2002.

[75] M. Ueberle, N. Mock, and M. Buss, "Towards a hyper-redundant haptic display," in *In Proceedings of the International Workshop on High-Fidelity Telepresence and Teleaction, jointly with the IEEE Conference on Humanoid Robots (HUMANOIDS2003*, 2003.

[76] H. Date, Y. Hoshi, and M. Sampei, "Locomotion control of a snake-like robot based on dynamic manipulability," in *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 3, 2000, pp. 2236 –2241 vol.3.

[77] B. Tondu, "A zonotope-based approach for manipulability study of redundant robot limbs," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, 2011, pp. 261–268.

[78] J. Lee, "A study on the manipulability measures for robot manipulators," in *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, vol. 3, 1997, pp. 1458–1465.

[79] P. Chiacchio, S. Chiaverini, and B. Siciliano, "Dexterous reconfiguration of a two-arm robot system," in *Control 1991. Control '91., International Conference on*, mar 1991, pp. 347 –351 vol.1.

[80] A. Bicchi and D. Prattichizzo, "Manipulability of cooperating robots with unactuated joints and closed-chain mechanisms," *Robotics and Automation, IEEE Transactions on*, vol. 16, no. 4, pp. 336 –345, aug 2000.

[81] J. Lee and H. Shim, "On the dynamic manipulability of cooperating multiple arm robot systems," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 2, sept.-2 oct. 2004, pp. 2087 – 2092 vol.2.

[82] S. Lee, "Dual redundant arm configuration optimization with task-oriented dual arm manipulability," *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 1, pp. 78–97, 1989.

[83] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Global task space manipulability ellipsoids for multiple-arm systems," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 5, pp. 678–685, 1991.

[84] P. Dauchez and M. Uchiyama, "Kinematic formulation for two force-controlled cooperating robots," in *Proc. 3rd Int. conf. Advanced Robotics*, 1987, pp. 457–467.

[85] T. Padir, "Kinematic redundancy resolution for two cooperating underwater vehicles with on-board manipulators," *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, 2005.

[86] T. Padir and J. Nolff, "Manipulability and maneuverability ellipsoids for two cooperating underwater vehicles with on-board manipulators," *2007 IEEE International Conference on Systems, Man and Cybernetics*, 2007.

[87] K. Abdel-Malek and W. Yu, "Placement of robot manipulators to maximize dexterity," *The International Journal of Robotics Research*, vol. 19, no. 1, pp. 6–15, 2004.

[88] T. Asokan, G. Seet, M. Lau, and E. Low, "Optimum positioning of an underwater intervention robot to maximise workspace manipulability," *Mechatronics*, vol. 15, no. 6, pp. 747–766, 7 2005.

[89] Z. Li, D. Glozman, D. Milutinovic, and J. Rosen, "Maximizing dexterous workspace and optimal port placement of a multi-arm surgical robot," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 3394–3399.

[90] S. Selha, P. Dupont, R. D. Howe, and D. F. Torchiana, "Dexterity optimization by port placement in robot-assisted minimally invasive surgery," in *Proc. SPIE*, vol. 4570, 2002, pp. 97–104. [Online]. Available: http://dx.doi.org/10.1117/12.454734

[91] R. Konietschke, H. Weiss, T. Ortmaier, and G. Hirzinger, "A preoperative planning procedure for robotically assisted minimally invasive interventions," in *CURAC 2004 3. Jahrestagung der Deutschen Gesellschaft fur Computer- und Roboterassistierte Chirurgie*, 2004, lIDO-Berichtsjahr=2004,. [Online]. Available: http://elib.dlr.de/12111/

[92] T. Yoshikawa, "Dynamic manipulability of robot manipulators," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, 1985, pp. 1033–1038.

[93] P. Chiaacchio and M. Concilio, "The dynamic manipulability ellipsoid for redundant manipulators," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 1, 1998, pp. 95–100 vol.1.

[94] M. T. Rosenstein and R. A. Grupen, "Velocity-dependent dynamic manipulability," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 3, 2002, pp. 2424–2429.

[95] R. Kurazume and T. Hasegawa, "A new index of serial-link manipulator performance combining dynamic manipulability and manipulating force ellipsoids," *Robotics, IEEE Transactions on*, vol. 22, no. 5, pp. 1022–1028, 2006.

[96] D. J. Cox, "Cooperative Manipulation Testbed Development Kinematics," in *2004 Florida Conference of the Recent Advances in Robotics, Proc. on*, 2004, pp. 1–12.

[97] A. Elkady, M. Mohammed, and T. Sobh, "A new algorithm for measuring and optimizing the manipulability index," *Journal of Intelligent and Robotic Systems*, vol. 59, no. 1, pp. 75–86, 2010.

[98] D. Pai and M. Leu, "Generic singularities of robot manipulators," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, May 1989, pp. 738–744 vol.2.

[99] P. S. Donelan, C. Science, and N. Zealand, "Singularity-theoretic methods in robot kinematics," *Robotica*, vol. 25, no. 06, pp. 641–659, 2007.

[100] M. J. D. Hayes, M. L. Husty, and P. J. Zsombor-Murray, "Singular configurations of wrist-partitioned 6r serial robots: A geometric perspective for users," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 26, no. 1, pp. 41–55, 2003.

[101] L. Djimon and C. Akonde, "Singularity Condition of Wrist-Partitioned 6-R Serial Manipulator based on Grassmann-Cayley Algebra," *IJRET: International Journal of Research in Engineering and Technology*, vol. 2, no. 12, pp. 700–705, 2013.

[102] Y. T., "Manipulability of robotic mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[103] A. A.Maciejewski and C. A.Klein, "The singular value decomposition: Computation and applications to robotics," *The International Journal of Robotics Research*, vol. 8, no. 6, pp. 63–79, 1989.

[104] W. Press, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007. [Online]. Available: http://books.google.es/books?id=1aAOdzK3FegC

[105] K. Jin-Oh and K. Khosla, "Dexterity measures for design and control of manipulators," in *IEEE/RSJ Int.Workshop on Intelligent Robots and Systems IROS*, Nov. 3-5 1991, pp. 758–763.

[106] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann, "Manipulability analysis," *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, no. 3, pp. 568–573, Nov. 2012.

[107] P. Hoppenot and E. Colle, "Directional Manipulability for Motion coordination of an Assistive Mobile Arm," in *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Angers, 2007, pp. 1–8.

[108] H. Chen, S.-I. Lee, J.-H. Do, and J.-M. Lee, "Directional Manipulability to Improve Performance Index of Dual Arm Manipulator for Object Grasping," in *Intelligent Autonomous Systems 12 SE - 56*, ser. Advances in Intelligent Systems and Computing, S. Lee, H. Cho, K.-J. Yoon, and J. Lee, Eds.

Springer Berlin Heidelberg, 2013, vol. 193, pp. 595–602. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33926-4_56

[109] C. A.Klein and B. E.Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *Int.J.Rob.Res.*, vol. 6, no. 2, pp. 72–83, 1987.

[110] C. Gosselin and J. Angeles, "A global performance index for the kinematic optimization of robotic manipulators," *Journal of Mechanical Design*, vol. 113, no. 3, pp. 220–226, September 1991 1991.

[111] L. Stocco, S. E. Salcudean, and F. Sassani, "Fast constrained global minimax optimization of robot parameters," *Robotica*, vol. 16, no. 6, pp. 595–605, nov 1998.

[112] L. J. Stocco, S. E. Salcudean, and F. Sassani, "Optimal kinematic design of a haptic pen," *Mechatronics, IEEE/ASME Transactions on*, vol. 6, no. 3, pp. 210–220, 2001.

[113] F. Gao, X. Liu, and W. A. Gruver, "Performance evaluation of two-degree-of-freedom planar parallel robots," *Mechanism and Machine Theory*, vol. 33, no. 6, pp. 661–668, 8/1 1998.

[114] X.-J. Liu, J. Wang, and H.-J. Zheng, "Optimum design of the 5r symmetrical parallel manipulator with a surrounded and good-condition workspace," *Robotics and Autonomous Systems*, vol. 54, no. 3, pp. 221–233, 3/31 2006.

[115] B.-H. Jun, P.-M. Lee, and S. Kim, "Manipulability analysis of underwater robotic arms on rov and application to task-oriented joint configuration," *Journal of Mechanical Science and Technology*, vol. 22, no. 5, pp. 887–894, 2008. [Online]. Available: http://dx.doi.org/10.1007/s12206-008-0201-7

[116] Q. Zhang and A. Chen, *Kinematics and Task-Oriented Manipulability Measure for the Bionic Cricket Hopping Robot*, ser. Intelligent Robotics and Applications. Springer Berlin Heidelberg, 2008, vol. 5314, pp. 342–351. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-88513-9_37

[117] S. Lee and J. M. Lee, "Task-oriented dual-arm manipulability and its application to configuration optimization," in *Decision and Control, 1988., Proceedings of the 27th IEEE Conference on*, 1988, pp. 2253–2260 vol.3.

[118] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Global task space manipulability ellipsoids for multiple-arm systems," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 5, pp. 678–685, 1991.

[119] A. Bicchi and D. Prattichizzo, "Manipulability of cooperating robots with unactuated joints and closed-chain mechanisms," *Robotics and Automation, IEEE Transactions on*, vol. 16, no. 4, pp. 336–345, Aug 2000.

[120] F. Park and J. Kim, "Kinematic manipulability of closed chains," in *Recent Advances in Robot Kinematics*, J. Lenari and V. Parenti-Castelli, Eds. Springer Netherlands, 1996, pp. 99–108. [Online]. Available: http://dx.doi.org/10.1007/978-94-009-1718-7_10

[121] F. C. Park and J. W. Kim, "Manipulability and singularity analysis of multiple robot systems: a geometric approach," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2, 1998, pp. 1032–1037 vol.2.

[122] ——, "Singularity analysis of closed kinematic chains," *Journal of Mechanical Design*, vol. 121, no. 1, pp. 32–38, Mar. 1999. [Online]. Available: http://dx.doi.org/10.1115/1.2829426

[123] S. Lee, "Dual redundant arm configuration optimization with task-oriented dual arm manipulability," *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 1, pp. 78–97, 1989.

[124] P. Lancaster and K. Salkauskas, *Curve and surface fitting :an introduction.* London etc.: Academic Press, 1986, peter Lancaster and Kestutis Salkauskas; :il., gr f.;24 cm; Bibliografia. ndex.

[125] G. J. Hahn and W. Q. Meeker, *Statistical intervals :a guide for practitioners.* New York etc.: Wiley, 1992, gerald J. Hahn, William Q. Meeker; :il.;24 cm; "A Wiley-Interscience publication."; Bibliografia p. 368-381. ndexs; Wiley series in probability and mathematical statistics. Applied probability and statistics.

[126] C. Zhao, M. Farooq, and M. Bayoumi, "Kinematic motion constraints in the robot's joint space," in *Industrial Electronics, Control, and Instrumentation, 1996., Proceedings of the 1996 IEEE IECON 22nd International Conference on*, vol. 2, Aug 1996, pp. 1246–1251 vol.2.

[127] M. Stilman, "Task constrained motion planning in robot joint space," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, Oct 2007, pp. 3074–3081.

[128] F. Flacco, A. De Luca, and O. Khatib, "Motion control of redundant robots under joint constraints: Saturation in the null space," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 285–292.

[129] E. Freund and J. Rossman, "The basic ideas of a proven dynamic collision avoidance approach for multi-robot manipulator systems," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2, 2003, pp. 1173–1177.

[130] A. Hernansanz, A. Casals, and J. Amat, "Virtual robot: A new teleoperation paradigm for minimally invasive robotic surgery," in *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob 2012*, 2012.

[131] T. B. Sheridan, *Telerobotics, automation, and human supervisory control*. Cambridge, MA, USA: MIT Press, 1992.

[132] P. Appelqvist, J. Knuuttila, and J. Ahtiainen, "Development of an unmanned ground vehicle for task-oriented operation - considerations on teleoperation and delay," in *Advanced intelligent mechatronics, 2007 IEEE/ASME international conference on*, Sept 2007, pp. 1–6.

[133] A. Codourey, M. Rodriguez, and I. Pappas, "A task-oriented teleoperation system for assembly in the microworld," in *Advanced Robotics, 1997. ICAR '97. Proceedings., 8th International Conference on*, 1997, pp. 235–240.

[134] H. Kim, J.-S. Kim, K. Ryu, S. Cheon, Y. Oh, and J.-H. Park, "Task-oriented teleoperation through natural 3d user interaction," in *Ubiquitous Robots and Ambient Intelligence (URAI), 2014 11th International Conference on*, Nov 2014, pp. 335–338.

[135] N. Y. Chong, T. Kotoku, K. Ohba, K. Komoriya, N. Matsuhira, and K. Tanie, "Remote coordinated controls in multiple telerobot cooperation," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 4, 2000, pp. 3138–3143 vol.4.

[136] M. Ghodoussi, S. E. Butner, and Y. Wang, "Robotic surgery - the transatlantic case," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 2, 2002, pp. 1882–1888 vol.2.

[137] H. Reichenspurner, R. J. Damiano, M. Mack, D. H. Boehm, H. Gulbins, C. Detter, B. Meiser, R. Ellgass, and B. Reichart, "Use of the voice-controlled and computer-assisted surgical system zeus for endoscopic coronary artery bypass grafting," *The Journal of thoracic and cardiovascular surgery*, vol. 118, no. 1, pp. 11–16, 7 1999.

[138] U. Hagn, R. Konietschke, A. Tobergte, M. Nickl, S. Jrg, B. Kbler, G. Passig, M. Gröger, F. Fröhlich, U. Seibold, L. Le-Tien, A. Albu-Schffer, A. Nothhelfer,

F. Hacker, M. Grebenstein, and G. Hirzinger, "Dlr mirosurge: a versatile system for research in endoscopic telesurgery," pp. 183–193, 2010.

[139] G. S. Guthart and J. K. S. Jr., "The intuitivetm telesurgery system: overview and application," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 1, 2000, pp. 618–621 vol.1.

[140] B. Chebbi, D. Lazaroff, F. Bogsany, P. X. Liu, L. Niy, and M. Rossi, "Design and implementation of a collaborative virtual haptic surgical training system," in *Mechatronics and Automation, 2005 IEEE International Conference*, vol. 1, 2005, pp. 315–320 Vol. 1.

[141] D. Feth, B. Tran, R. Groten, A. Peer, and M. Buss, "Shared-control paradigms in multi-operator-single-robot teleoperation," in *Human Centered Robot Systems*, ser. Cognitive Systems Monographs, H. Ritter, G. Sagerer, R. Dillmann, and M. Buss, Eds. Springer Berlin Heidelberg, 2009, vol. 6, pp. 53–62. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-10403-9_6

[142] T. D. Ngo, "Quantitative analysis of distributed control paradigms for robot swarms," in *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, 2010, pp. 116–122.

[143] H. Wei, D. Li, J. Tan, and T. Wang, "The distributed control and experiments of directional self-assembly for modular swarm robots," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 4169–4174.

[144] P. Fraisse, D. Andreu, R. Zapata, J. P. Richard, and T. Divoux, "Remote decentralized control strategy for cooperative mobile robots," in *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, vol. 2, 2004, pp. 1011–1016 Vol. 2.

[145] X. Cao, J. Chen, Y. Xiao, and Y. Sun, "Building-environment control with wireless sensor and actuator networks: Centralized versus distributed," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 11, pp. 3596–3605, 2010.

[146] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, 1985, pp. 500–505.

[147] ——, "Real-time obstacle avoidance for manipulators and mobile robots," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, Mar 1985, pp. 500–505.

[148] R. R. Murphy, *Potential Fields Methodologies*, 1st ed., ser. Introduction to AI Robotics. Cambridge, MA, USA: MIT Press, 2000, pp. 105–153.

[149] R. C. Arkin, *An Behavior-based Robotics*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[150] T. Balch, "Avoiding the past: a simple but effective strategy for reactive navigation," in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, 1993, pp. 678–685 vol.1.

[151] T. J. Liddy, T.-F. Lu, P. Lozo, and D. J. Harvey, "Obstacle avoidance using complex vector fields," in *Proceedings of the 2008 Australasian Conference on Robotics and Automation*, 2008, pp. 1–7.

[152] S. C. Costa, "Behaviour blending for multiple robot coordinated navigation through virtual potential field," 2013.

[153] A. Hernansanz, X. Giralt, A. Rodriguez, and J. Amat, "Rpq: Robotic proximity queries - development and applications," in *International Conference on Informatics in Control, Automation and Robotics, ICINCO*. INSTICC Press, 2007, pp. 59–66.

[154] X. Giralt, A. Hernansanz, A. Rodriguez, and J. Amat, *Robotic Proximity Queries Library for Online Motion Planning Applications*, ser. New Developments in Robotics Automation and Control. InTech, 2008, vol. 1, pp. 12–7.

[155] V. Westebring, R. H. M. Goossens, J. J. Jakimowicz, and J. Dankelman, "Haptics in minimally invasive surgery: a review," *Minim Invasive Ther Allied Technol*, vol. 17, no. 1, pp. 3–16, 01/01; 2012/01 2008.

[156] C. R. Wagner, N. Stylopoulos, P. G. Jackson, and R. D. Howe, "The benefit of force feedback in surgery: Examination of blunt dissection," *Presence: Teleoper. Virtual Environ.*, vol. 16, no. 3, pp. 252–262, June 2007.

[157] S. Schostek, C. Ho, D. Kalanovic, and M. O. Schurr, "Artificial tactile sensing in minimally invasive surgery - a new technical approach," *Minim Invasive Ther Allied Technol*, vol. 15, no. 5, pp. 296–304, 01/01; 2012/01 2006.

[158] T. Hu, A. E. Castellanos, and J. P. Desai, *Real-Time Haptic Feedback in Laparoscopic Tools for Use in Gastro-Intestinal Surgery*, ser. Lecture Notes in Computer Science. Springer Berlin, 2002, pp. 66–74.

[159] T. Ortmaier, B. Deml, B. Kbler, G. Passig, D. Reintsema, and U. Seibold, "Robot assisted force feedback surgery," pp. 361–379, 2007.

[160] N. Riviere, S. Rader, and V. Thakor, "Adaptive cancelling of physiological tremor for improved precision in microsurgery," *Biomedical Engineering, IEEE Transactions on*, vol. 45, no. 7, pp. 839–846, 1998.

[161] R. V. Clayman, "Surgical robotics: Impact of motion scaling on task performance," *The Journal of Urology*, vol. 174, no. 3, pp. 953–953, 9 2005.

[162] S. Salcudean, S. Ku, and G. Bell, "Performance measurement in scaled teleoperation for microsurgery," pp. 789–798, 1997.

[163] R. Prada and S. Payandeh, "On study of design and implementation of virtual fixtures," *Virtual Reality*, vol. 13, no. 2, pp. 117–129, 2009.

[164] T. Ortmaier, M. Groger, D. H. Boehm, V. Falk, and G. Hirzinger, "Motion estimation in beating heart surgery," *Biomedical Engineering, IEEE Transactions on*, vol. 52, no. 10, pp. 1729–1740, 2005.

[165] S. Yuen, N. Vasilyev, P. del Nido, and R. Howe, "Robotic tissue tracking for beating heart mitral valve surgery," *Medical image analysis*, 07/06 2010.

[166] C. Koh, *Laparoscopic Suturing in the Vertical Zone*. Endo-Press, 2005. [Online]. Available: https://books.google.es/books?id=dZ6NAAAACAAJ

[167] S. A. Bowyer, B. L. Davies, and F. y Baena, "Active Constraints/Virtual Fixtures: A Survey," *Robotics, IEEE Transactions on*, vol. 30, no. 1, pp. 138–157, Feb. 2014.

[168] S. E. Salcudean, S. Ku, and G. Bell, "Performance measurement in scaled teleoperation for microsurgery," in *Proceedings of the First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine and Medial Robotics and Computer-Assisted Surgery*. London, UK: Springer-Verlag, 1997, pp. 789–798.

[169] V. Falk, J. McLoughlin, G. Guthart, J. K. Salisbury, T. Walther, J. Gummert, and F. W. Mohr, "Dexterity enhancement in endoscopic surgery by a computer-controlled mechanical wrist," *Minimally Invasive Therapy & Allied Technologies*, vol. 8, no. 4, pp. 235–242, 1999.

[170] R. V. Clayman, "Surgical robotics: impact of motion scaling on task performance," *The Journal of Urology*, vol. 174, no. 3, p. 953, 2005.

[171] M. Li, M. Ishii, and R. H. Taylor, "Spatial Motion Constraints Using Virtual Fixtures Generated by Anatomy," *Robotics, IEEE Transactions on*, vol. 23, no. 1, pp. 4–19, 2007.

[172] R. Kumar, P. Jensen, and R. Taylor, "Experiments with a steady hand robot in constrained compliantmotion and path following," in *8th IEEE International Workshop on Robot and Human Interaction, 1999. RO-MAN'99*, 1999, pp. 92–97.

[173] R. H. Taylor, J. Funda, B. Eldridge, S. Gomory, K. Gruben, D. LaRose, M. Talamini, L. Kavoussi, and J. Anderson, "A telerobotic assistant for laparoscopic surgery," *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, no. 3, pp. 279–288, 1995.

[174] R. Cassilly, M. D. Diodato, M. Bottros, and R. J. Damiano, "Optimizing motion scaling and magnification in robotic surgery," *Surgery*, vol. 136, no. 2, pp. 291 – 294, 2004.

[175] L. Munoz, A. Casals, J. Amat, M. Puig-Vidal, and J. Samitier, "Improved afm scanning methodology with adaptation to the target shape," in *International Conference on Robotics and Automation*, Apr. 2005, pp. 1529 – 1534.

[176] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 624–637, Oct. 1993. [Online]. Available: http://dx.doi.org/10.1109/70.258054

[177] M. Li, A. Kapoor, and R. Taylor, "Telerobotic control by virtual fixtures for surgical applications," in *Advances in Telerobotics*, ser. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2007, vol. 31, pp. 381–401. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-71364-7_23

[178] R. Prada and S. Payandeh, "A study on design and analysis of virtual fixtures for cutting in training environments," in *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, 2005, pp. 375–380.

[179] B. L. Davies, S. J. Harris, W. J. Lin, R. D. Hibberd, R. Middleton, and J. C. Cobb, "Active compliance in robotic surgerythe use of force control as a dynamic constraint," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 1997.

[180] D. Aarno, S. Ekvall, and D. Kragic, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 897–903.

[181] A. Kapoor, M. Li, and R. H. Taylor, "Spatial motion constraints for robot assisted suturing using virtual fixtures," *Medical image computing and computer-assisted intervention : MICCAI05 International Conference on Medical Image*

Computing and Computer-Assisted Intervention, vol. 8, no. Pt 2, pp. 89–96, 2005.

[182] T. Xia, A. Kapoor, P. Kazanzides, and R. Taylor, "A constrained optimization approach to virtual fixtures for multi-robot collaborative teleoperation," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 639–644.

[183] D. Zerbato, D. Baschirotto, D. Baschirotto, D. Botturi, and P. Fiorini, "Gpu-based physical cut in interactive haptic simulations," *International Journal of Computer Assisted Radiology and Surgery*, vol. 6, pp. 265–272, 2011.

[184] ——, "Gpu-based physical cut in interactive haptic simulations," *Int. J. Computer Assisted Radiology and Surgery*, vol. 6, no. 2, pp. 265–272, 2011.

[185] S. Gibson, C. Fyock, E. Grimson, T. Kanade, R. Kikinis, H. Lauer, N. McKenzie, A. Mor, S. Nakajima, H. Ohkami, R. Osborne, J. Samosky, and A. Sawada, "Volumetric object modeling for surgical simulation," *Medical Image Analysis*, vol. 2, no. 2, pp. 121–132, 1998.

[186] J. J. Z. Liao and R. Liu, "Re-parameterization of five-parameter logistic function," *Journal of Chemometrics*, vol. 23, no. 5, pp. 248–253, 2009.

[187] P. G. Gottschalk and J. R. Dunn, "The five-parameter logistic: A characterization and comparison with the four-parameter logistic," *Analytical Biochemistry*, vol. 343, no. 1, pp. 54–65, 8/1 2005.

[188] ——, "Fitting brendan's five-parameter logistic curve."

[189] W. Halsted, "The training of the surgeon," *Bullettin of the Johns Hopkins Hospital*, 1904.

[190] C. R. Wagner, R. D. Howe, and N. Stylopoulos, "The role of force feedback in surgery: Analysis of blunt dissection," in *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, ser. HAPTICS '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 73–. [Online]. Available: http://dl.acm.org/citation.cfm?id=795682.797542

[191] J. Rasmussen, *Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models.* Piscataway, NJ, USA: IEEE Press, 1987, pp. 291–300.

[192] C. Wickens, *Processing resources and attention.* Bristol: Taler & Francis, Ltd., 1991, p. 334.

[193] R. Kumar, T. M. Goradia, A. C. Barnes, P. Jensen, L. L. Whitcomb, D. Stoianovici, L. M. Auer, and R. H. Taylor, "Performance of robotic augmentation in microsurgery-scale motions," in *Proceedings of the Second International Conference on Medical Image Computing and Computer-Assisted Intervention*, ser. MICCAI '99. London, UK: Springer-Verlag, 1999, pp. 1108–1115.

[194] A. Navarro, E. Villarraga, X. Giralt, A. Hernansanz, and J. Aranda, *Enhancing Perception in Minimally Invasive Robotic Surgery through Self-Calibration of Surgical Instruments.* IEEE, 2007.

[195] A. Navarro, A. Hernansanz, E. Villarraga, X. Giralt, and J. Aranda, *Automatic positioning of surgical instruments in minimally invasive robotic surgery through vision-based motion analysis.* IEEE, Oct 2007.

[196] A. Navarro, A. Hernansanz, J. Aranda, and A. Casals, "An approach to perception enhancement in robotized surgery using computer vision," in *Robot Vision*, La, Ed. Ales Ude, 2010, ch. 30, pp. 597–613.

[197] A. Hernansanz, J. Amat, and A. Casals, "Optimization criterion for safety task transfer in cooperative robotics," in *2009 International Conference on Advanced Robotics.* Munich: IEEE, 2009, pp. 1–6.

[198] A. Hernansanz, A. Casals, and J. Amat, "A multi-robot cooperation strategy for dexterous task oriented teleoperation," *Robotics and Autonomous Systems*, Feb. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889014003042

[199] ——, "Multi-robot cooperative platform: Performance evaluation in assisted teleoperated suturing," 2016.

[200] A. Hernansanz, D. Zerbato, L. Gasperotti, M. Scandola, P. Fiorini, and A. Casals, "Assessment of virtual fixtures for the development of basic skills in robotic surgery," in *16th Annual Conference of the International Society for Computer Aided Surgery, CARS 2012*, vol. Pisa, 2012.

[201] ——, "Improving the development of surgical skills with virtual fixtures in simulation," in *Information Processing in Computer-Assisted Interventions*, ser. Lecture Notes in Computer Science, P. Abolmaesumi, L. Joskowicz, N. Navab, and P. Jannin, Eds. Springer Berlin Heidelberg, 2012, vol. 7330, pp. 157–166. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30618-1_16

[202] J.-C. Latombe, *Robot Motion Planning.* Norwell, MA, USA: Kluwer Academic Publishers, 1991.

[203] G. research group. Collision detection and proximity query packages. [Online]. Available: http://gamma.cs.unc.edu/research/collision/packages.html

[204] Y. J. Kim, M. C. Lin, and D. Manocha, "Deep: dual-space expansion for estimating penetration depth between convex polytopes," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 1, 2002, pp. 921–926.

[205] ——, "Incremental penetration depth estimation between convex polytopes using dual-space expansion," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 10, no. 2, pp. 152–163, 2004.

[206] S. A. Ehmann and M. C. Lin, "Accurate and fast proximity queries between polyhedra using convex surface decomposition," in *IN COMPUTER GRAPHICS FORUM*, 2001, pp. 500–510, publisher:.

[207] K. E. Hoff, III, A. Zaferakis, M. Lin, and D. Manocha, "Fast and simple 2d geometric proximity queries using graphics hardware," 2001.

[208] ——, "Fast 3D Geometric Proximity Queries between Rigid and Deformable Models Using Graphics Hardware Acceleration Technical Report TR02-004," 2002.

[209] A. Gregory, M. C. Lin, S. Gottschalk, and R. Taylor, "A framework for fast and accurate collision detection for haptic interaction," in *Virtual Reality, 1999. Proceedings., IEEE*, 1999, pp. 38–45.

[210] ——, "Fast and accurate collision detection for haptic interaction using a three degree-of-freedom force-feedback device," *Comput.Geom.Theory Appl.*, vol. 15, no. 1-3, pp. 69–89, feb 2000. [Online]. Available: http://dx.doi.org/10.1016/S0925-7721(99)00041-3

[211] M. C. Lin, A. Gregory, S. Ehmann, S. Gottschalk, and R. Taylor, "Contact determination for real-time haptic interaction in 3d modeling, editing and painting," in *in 3D Modeling, Editing and Painting. Proc. 1999 Workshop for PhanTom User Group. 7*, 1999.

[212] J. D. Cohen, M. C. Lin, D. Manocha, and M. Ponamgi, "I-COLLIDE: An interactive and exact collision detection system for large-scale environments," in *In Proc. of ACM Interactive 3D Graphics Conference*, 1995, pp. 189–196.

[213] S. Gottschalk, M. C. Lin, and D. Manocha, "Obbtree: a hierarchical structure for rapid interference detection," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH

'96. New York, NY, USA: ACM, 1996, pp. 171–180. [Online]. Available: http://doi.acm.org/10.1145/237170.237244

[214] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, "Fast proximity queries with swept sphere volumes," Department of Computer Science, UNC Chapel Hill, Tech. Rep., 1999.

[215] A. Wilson, E. Larsen, D. Manocha, and M. C. Lin, "Immpact: Partitioning and handling massive models for interactive collision detection," in *Eurographics*. Blackwell Publishers, 1999, pp. 319–329.