Accepted Manuscript

A multi-robot cooperation strategy for dexterous task oriented teleoperation

A. Hernansanz, A. Casals, J. Amat

PII: S0921-8890(14)00304-2

DOI: <http://dx.doi.org/10.1016/j.robot.2014.12.007>

Reference: ROBOT 2414

To appear in: *Robotics and Autonomous Systems*

Received date: 30 May 2014

Revised date: 7 December 2014

Accepted date: 12 December 2014

Please cite this article as: A. Hernansanz, A. Casals, J. Amat, A multi-robot cooperation strategy for dexterous task oriented teleoperation, *Robotics and Autonomous Systems* (2015), <http://dx.doi.org/10.1016/j.robot.2014.12.007>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

A Multi-Robot Cooperation Strategy for dexterous Task Oriented Teleoperation[☆]

A. Hernansanz^{a,b,*}, A. Casals^{a,b}, J. Amat^b

^a*Institute for Bioengineering of Catalonia. Baldori Reixac, 4-6, 08028 Barcelona, Spain*

^b*Universitat Politècnica de Catalunya. Barcelona Tech. Jordi Girona 1-3, 08034 Barcelona, Spain.*

Abstract

The use of multiple robots working cooperatively in a redundant way offers new possibilities in the execution of complex tasks in dynamic workspaces. The aim of this work 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: a Multi-Robot Cooperation Platform for Task-Oriented Teleoperation, MRCP. From the human operator commands, this robotic platform, the MRCP, dynamically selects the most suitable slave robot and manages, when necessary, a task transfer from one robot to another in order to achieve a smooth execution of teleoperated tasks.

The result of the proposed methodology is an improved teleoperated system in terms of reachable workspace (volume, maneuverability 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.

Keywords: Multi-Robot Cooperation, Task-Oriented Teleoperation, Single-Operator-Multiple-Robot

1. Introduction

The continuous evolution of telerobotics expands its range of applicability to areas as diverse as space, underwater, maintenance or surgery. Each of them presents specific requirements and workspace characteristics. For instance, long distance teleoperation, has to cope with communication delays [1]. Space robotics, suffers from the lack of gravity (apart from communication delays), whereas, the main challenges in surgical robotics are related to the increase of dexterity, accuracy and patient safety [2]. The increasing demands of these applications impose new usability requirements. This work aims at contributing to those teleoperated tasks that, due to their requirements, cannot be executed with a unique robot.

From the point of view of this research work, the causes that prevent a robot from executing a task can be classified into intrinsic, extrinsic and task dependent. The former refer to robot limitations: singularities, joint limits or the robot workspace itself. Those extrinsic are related to the task workspace: the presence of obstacles and other robots, forbidden regions, etc. The latter refer to specific task requirements, like the lack of degrees of freedom, DoF, with respect to those required for a task, occlusions in teleinspection, maximum tool orientation in minimally invasive surgery, etc.

In order to overcome these problems, a new platform composed of a set of robotic arms acting as a unique robot in a cooperative way is proposed: the Multi-Robot Cooperation Platform

for Task Oriented Teleoperation, MRCP. With the purpose of achieving a smooth and efficient teleoperation with the MRCP, the execution of the ongoing task is dynamically assigned to the most suitable robot. The suitability of each robot to continue with the task is continuously evaluated and task transfers from one robot to another are automatically executed when necessary. The rest of robots move to the best position to be able to continue with the task execution, minimizing task disturbances in future task transfers.

In teleoperation, the lack of direct perception produces a loss of proprioception and exteroception, thus decreasing operator's performance in aspects like accuracy, dexterity or completion time [3, 4, 5]. Additionally, in multiple slave systems, coordination and collision avoidance between robots is required. This might pose a problem in tasks that demand high operator's skills, such as a regrasping process, or a precise telemanipulation. The designed MRCP minimizes these limitations by proposing a change in the teleoperation paradigm: Following a classical teleoperation approach, the operator manipulates a master device to execute a task and its explicit orders are replicated by the slave robots. In the MRCP, the slave robots adapt, dynamically, their configuration to execute the teleoperated task from higher level orders. For a better understanding, let's use a telemanipulation task as example to illustrate the differences between standard and MRCP paradigms. In conventional teleoperation, the operator interacts with the master devices focusing on the explicit actions of each slave robot: arm positioning, collision avoidance, occlusions and object transfer from one robot to another. With the MRCP approach, the operator focuses on the task to be executed: object positioning. The master movements are mapped on the manipulated object, while the MRCP control strategy makes the robots act in a coordinated

[☆]This work has been developed under project: DPI2011-29660-C04, sub-projects 01 and 03, funded by the Spanish Research ministry, MINECO and Feder funds.

*Corresponding author

Email address: albert.hernansanz@upc.edu (A. Hernansanz)

minated way to execute the desired manipulation, avoiding collisions and occlusions and automatically transferring the object from one robot to another when necessary.

2. Related Work

In order to deal with complex robotic manipulation, different robot architectures can be considered; that is, redundant robots, cooperative robots, or specific robots. Redundancy represents a solution to the improvement of maneuverability and reachability. With redundancy, singularities [6, 7] or collisions [8, 9] can be avoided and some parameters, as joint torques [10], can be optimized all along the movement. In [11, 12] authors propose a control algorithm that automatically solves aspects like collision avoidance and robot limits in bilateral teleoperation using robot redundancy. However, depending on the task to be performed, redundant robots can become ineffective [13]. In the proposed approach, redundancy is based on the number of robots, thus obtaining not only higher accessibility, but also a larger working volume.

Multirobot cooperation approaches usually focus on manipulation tasks that a single robot cannot carry out, such as manipulation of large, heavy or even flexible objects [14]. Most of the research work developed in this field has been addressed to aspects related to the control of closed kinematic chains (chains composed of several robots holding an object at the same time): to optimize the grasping positions of the manipulated object and to the development of efficient path planning in shared workspaces [15, 16, 17].

Another field where multirobot systems have been recently applied is in bi-manual manipulation [18], in which more than one robot operate simultaneously to achieve a single goal. 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 several subtasks, which are assigned to the various robots involved and executed in parallel. Examples of bi-manual manipulation tasks can be found in [19], where a robot arm holds a glass while a second robot pours liquid from a bottle, or in [20], in which a motion planner for bi-manual complex insertion tasks is developed.

In the multi-robot path planning field, an automatic multi-robot path planner, presented in [21], computes the required motion to manipulate an object from an initial position 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 that work relies on the unavailability of path planning in teleoperation.

Cooperative robots are also applied to the field of 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 robot is individually controlled by an operator [22]. Instead, SOMR architectures

are composed of several robots controlled by a single operator. In [23], the control architecture of two slave robots, an arm mounted on a mobile robot, allows switching between different properties of each robot or between the robots themselves using a single master device. This SOMR proposal differs from the MRCP concept in two basic aspects: first, in the SOMR approach, each robot plays a different and specific role; whereas in the MRCP approach, all robots have complementary behaviors. A second difference refers to the knowledge about which robot is controlled at every moment: in SOMR, the user actively selects the robot to be controlled by the master device, whereas in MRCP the selection is made automatically and transparent to the user. 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 user's point of view.

From the control point of view, MRCP can be classified as a task oriented teleoperation system. In this paradigm, the operator's actions are related to the task rather than on the agents (robotic arms in teleoperation field). Following 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 presents several similitudes with the task oriented paradigm: In MRCP the operator can focus its efforts only on the task. The control is oriented to guarantee a satisfactory task execution, minimizing the disturbances (in time or space). The task oriented paradigm was introduced in teleoperation systems in works like [24], proposing a task oriented SOMR teleoperated system in which the operator does not drives the slave arms but uses the master to generate specific task-oriented actions. Several examples are used to describe that approach: the 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 the 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 the arms movements.

3. MRCP in the teleoperation context

The teleoperation Multi-Robot Cooperative Platform, MRCP, was born with the aim of increasing the range of applicability of traditional master slave teleoperation architectures. The MRCP cooperative behavior is based on the complementarity of the slave robots that guarantee a successful teleoperation task execution. 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 robots, which of them is the most suitable to continue with the teleoperated task and when a task transfer between robots is needed. 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.

In order to clarify the MRCP concept, several criteria can be used. Then, MRCP is classified using the teleoperation taxonomy and the human supervisory control criteria, pointing out the main differences with standard definitions in the teleoperation area. Finally a high level overview of the MRCP control is presented.

3.1. MRCP in the context of teleoperation

MRCP can be classified following the teleoperation taxonomy proposed in [25]. This taxonomy, based on the number of operators and slave robots, classifies the teleoperation systems in four groups: Single-Operator Single-Robot, Single-Operator Multiple-Robot, Multiple-Operator Single-Robot and Multiple-Operator Multiple-Robot.

SOMR teleoperated systems rely on a master console composed of one or two master devices that drive several slave robots. Most Robotic Minimally Invasive Surgery systems, RMIS, use SOMR: two master devices controlling several slave robots (one slave at a time per master) $1M:nS+1M:nS$, $n \geq 2$. The operator manually selects which slave robot is controlled by each master. For instance, the DaVinci robot provides two master devices and four slave robots ($1M:4S+1M:4S$), and the surgeon manually selects the master-slave assignment. MRCP can be classified as a SOMR teleoperated system from the architectural point of view: a single operator interacting with a unique master device that implicitly controls several robots, $1M:nS$. The block schemas of standard SOMR and the MRCP architectures (both applied to RMIS) are shown in Fig 1.a and Fig 1.b respectively. Two differences can be pointed out when reviewing these schemas. First, in SOMR, the master-slave assignment module is independent from the robots control unit (the operator decides the master-slave assignment), while in MRCP, this assignment is part of the control itself. Second, in bimanual tasks using SOMR, the operator uses left and right hands, while in MRCP the task is executed by the dominant hand, improving the execution performance. The non-dominant hand can be used for other auxiliary tasks, such as controlling the laparoscopic camera in RMIS.

Following the teleoperation taxonomy, and changing the analysis from the architectural to the operator's point of view, MRCP can be classified as a SOSR: A single robot executing a task. The automatic control processes of robot selection and robot task transfer provides the operator with the perception of being working with a single robot. Increasing the abstraction level, with MRCP the operator executes the task instead of the robot actions, regardless which physical robot has been selected to execute each of them or if a task transfer between robots is required.

3.2. MRCP in Human Supervisory Control

The human role within a teleoperation system defines several control models. This role varies with the interaction with systems ranging from manual to fully automated control. In manual control the operator executes a teleoperated task

Figure 1: (Up) Standard SOMR ($1M:4S+1:4S$) and (Down) the MRCP proposed schema. Both systems are applied in the RMIS field (without losing generality).

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 the human and the control system. Following the work developed in [26], in Fig.2 different levels of human supervisory control are presented. From Fig.2.a to .d, the human role decreases. In Fig.2.a, the robots are directly guided by the operator, without any teleoperation control intervention. In Fig.2.b a minor control loop has been introduced to aid the operator or to improve its skills (e.g. cancelling tremor, vary the M-S motion scaling, etc.). In Fig.2.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.2.d. Finally, Fig.2.e shows the MRCP control schema. The MRCP module closes the robots control loops. The task oriented orders are generated by the operator and translated as robot actions in the control system. Robot positioning and task transfers between robots are computed and planned inside 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.

Figure 2: Human Supervisory Control schemas. From a) to d), the human control decreases. In e), the MRCP control system is shown. Dashed lines represent low level control loops whereas solid lines are strong control loops.

4. MRCP Control Review

The proposed MRCP can be classified, from the control point of view, as a Single-Operator Multi-Robot teleoperation system with the configuration of a single master device commanding several robots (1M:nS). It is designed to enable the task-oriented teleoperation: operator actions are translated as high level task orders. That is, in a bimanual telemanipulation system, following the standard telemanipulation approach, the movements of the master are directly mapped onto the slave robots (1M:1S + 1M:1S) whereas in MRCP the movements of the unique master are mapped as object movements.

The behaviour of the 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 are positioned close to the operating point in order to facilitate a potential task transfer.

4.1. Control design requirements

The MRCP control system must operate in real time, with transparency and minimizing the number of task transfers. Concerning real time, the presence of time delays decreases the performance of the 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. 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 are 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. The transfer process represents the main perturbation to a smooth task execution, cancelling 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 required. The 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.

4.2. 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.3, the MRCP is described by a control block diagram. In this schema, the MRCP control unit has three different inputs:

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

2. $X_{R1}, \dots, X_{Rn}$: Information about the robots provided by their controllers (robot position, end effector status, ...).
3. $X_{Sensors}$: Information about 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:

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, are 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 to generate virtual fixtures to improve the operator's performance. This feature is not present on the schema because the generation and the effects of the virtual fixtures are out of the scope of this paper.
2. Slave Robots: Set of signals for the robot controllers, $\Delta X'_{R1}, \dots, \Delta X'_{Rn}$. These orders are the result of computing the robot actions required to continue with the execution of the 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 3: Block schema of MRCP seen as a plant

For a formal approach to 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 method, 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 $RCan$ 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. 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.

4.3. Organization: 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 uses 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 shows the interaction between control modules, including the master and slave stations among other auxiliary software modules (augmented reality and workspace simulator).

Figure 4: MRCP control architecture. The schema includes the master console and the remote workspace among other auxiliary modules.

MRCP 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 control system that automatically executes the task transfer. Once the task transfer is completed, MRCP returns to TOp mode. Both status generate different paths inside the control modules. Fig.5 shows the state transition graph of MRCP. 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 Robots Actions Computation module, which interacts with the slave system until the task transfer is finished. In Table 4.3, the states composing the MRCP control are reviewed, including the operational mode, the inputs, outputs and the conditions of a transition between modules.

Figure 5: Graph of states of MRCP. TOp transitions in mode Teleoperation and TT transitions in mode Task Transfer

5. MRCP Control Algorithm

In this section the different control modules composing the MRCP algorithm are reviewed. The modules are analysed as high level control units, giving an overall view of the most important aspects of each one.

5.1. Robot Suitability Evaluation

This module, the first step of the MRCP control process, computes the estimated robots suitability to continue with the ongoing task. The 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.

Table 1: Table with the states, transitions, inputs, outputs and operational modes of the MRCP control

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

5.1.1. Evaluation indices

The suitability evaluation indices have been defined considering the different causes that prevent the robot from continuing the execution. Three generic types of evaluation indices have been identified: Intrinsic, Extrinsic and Task dependant indices. The MRCP control algorithm is an open platform designed to be able to include all evaluation criteria required for a specific task.

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 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 [27], 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. An extended review of these indices can be found in Appendix A.

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 $I'_{m_l q}(t)$ where, as defined previously, r is the robot, m the evaluation index and q the evaluated point. Fig.6 shows an illustrative example of a telemanipulation using a MRCP with $n_r = 2$ and $n_q = 3$. In Fig.6a 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.6b MRCP is executing a task transfer (*TT* mode). R2, now R2 = *RCnt*, will continue the task in the best evaluated *CnTP*. Finally, in Fig.6c MRCP is operating again in *TOp* mode with *RSel* = R2 and R1 is evaluated in *CnTP*₂ and *CnTP*₃.

Figure 6: Illustrative sequence of MRCP teleoperation.

5.1.2. Common metric: 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, $ttransf_{m_l q}^r(t)$) and the easiness of the corresponding task transfer process.

Execution time, $texec_{m_l q}^r(t)$. To obtain $texec_{m_l q}^r(t)$, first, the polynomial that fits the evolution of each $I'_{m_l q}(t)$ along the last evaluation times is obtained by means of the least squares curve fitting method technique. Second, the zeros of all evaluation indices, $I'_{m_l q}(t) = 0$ are computed and, finally, $texec_{m_l q}^r(t)$ is fixed using the closest zero of all polynomials.

Let $P_{m_l q}^r(t)$ be the fitting polynomial of each $I'_{m_l q}(t)$ that mini-

mizes $\|P_{m_l}^r(t) - I_{m_l}^r(t, t-1, \dots, t-k)\|^2$ 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 m th evaluation index, $texec_{m_l}^r(t)$. The execution time is obtained selecting the closest zero of its associated polynomial, discarding these zeros previous to the current evaluation time, t , (1).

$$texec_{m_l}^r(t) = t_i \quad \text{where} \quad (1)$$

$$t_i = \min(t - t_i), \forall t_i | P_{i_l}^r(t_i) = 0 \quad \text{and} \quad t - t_i \leq 0$$

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 $texec_{m_l}^r(t)$ 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) $texec_{i_l}^r(t)$ is selected, (2).

$$texec_q^r(t) = \min \square texec_{1_l}^r(t), texec_{2_l}^r(t), \dots, texec_{n_{ml}}^r(t) \square \quad (2)$$

Task Transfer: $ttransf_q^r(t)$. Determines the estimated required time to execute a task transfer between $RSel$ and $RCan = r$ in $CuTP = q$. $ttransf_q^r(t)$ is the result of computing the required task transfer time including a safety margin, named Task Transfer Complexity, TTC_q^r . 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 (3). Numerically, $TTC_q^r \in [0, 1]$, being $TTC_q^r \rightarrow 1$ when the required accelerations approach to the robot maximum accelerations.

$$ttransf_q^r(t) = ttransf_q^r(t)(1 + TTC_q^r), \quad \text{where} \quad (3)$$

$$TTC_q^r = \frac{1}{n} \sum_{j=1}^n \frac{\|\ddot{\theta}_{j,k}\|}{\|\ddot{\theta}_{j,max}\|} \quad \square \quad K$$

Useful time, $tus_q^r(t)$. 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).

$$tus_q^r(t) = texec_q^r(t) - ttransf_q^r(t) \quad (4)$$

The complete algorithm to determine all tus_q^r is shown in Algorithm 1.

5.2. 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 ($RSel$, $RCan$), determine the need of an immediate task transfer (NTT).

Algorithm 1 Continuity Robot Computation, RCnt

```

%Evaluate  $RSel$  in  $CnTP$ 
for  $i = 1..n_m$  Evaluation Indices do
    Compute  $I_{i_{CuTP}}^{RSel}$ 
     $texec_{i_{CuTP}}^{RSel} \leftarrow I_{i_{CuTP}}^{RSel}$ 
end for
 $texec_{CuTP}^{RSel} = \min(texec_{i_{CuTP}}^{RSel}), \forall i = 1..n_m$ 
 $tus_{CuTP}^{RSel} = texec_{i_{CuTP}}^{RSel} \% (1)$ 
%Evaluate every  $RCan$ 
for  $r = 1..n_r$  Robots,  $r$ ,  $RSel$  do
    %In every  $CuTP$ 
    for  $q = 1..n_q, q$ ,  $CuTP$  do
        for  $i = 1..n_m$  Evaluation Indices do
            Compute  $I_{i_q}^r$ 
             $texec_{i_q}^r \leftarrow I_{i_q}^r$ 
        end for
         $texec_q^r = \min(texec_{i_{CuTP}}^{RSel}), \forall i = 1..n_m$ 
        Compute  $ttransf_q^r$ 
         $tus_q^r = texec_q^r - ttransf_q^r$ 
    end for
    end for
    List_of_RCan = sort_max( $tus_q^r$ ),  $\forall r = 1..n_r$  and  $q = 1..n_q$ 
    (1): $ttransf_{CuTP}^{RSel} = 0$ 

```

5.2.1. 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 $CuTP$, (5).

$$RCnt = (r, q) | tus_q^r = \max(tus_{q=q_i}^{r=r_i}, tus_{q=CuTP}^{r=RSel}) \quad (5)$$

where: $tus_{q_j}^{r_i} = \max \square tus_q^r \square$

$\forall r = 1..n_r, r$, $RSel$, and $\forall q = 1..n_q, q$, $CuTP$

When $RCnt$, $RSel$, 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).

5.2.2. Task Transfer Decision

Once $RCnt$ is identified and $RCnt$, $RSel$, 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 , (6).

$$NTT = 1 - \frac{tus_{CuTP}^{RSel} - ttransf^{RCnt}}{tus_{CuTP}^{RSel}} \quad (6)$$

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.7 shows an illustrative example of NTT computation process in a four robot MRCP system with two TP . The evaluation process gives

Figure 7: Example of determination of the optimal instant to begin a task transfer process ($NTT = 1$).

the following results: $R1$ requires a low $ttransf^*$, 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 $ttransf^*$, 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.

5.3. 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 the 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. The robot movements require to define the goal position of each robot to compute their trajectories. 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 of computing the required robot actions. 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 $RCan_j$.

5.3.1. 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 and, if required, a random force to cancel null forces (sum of attraction and repulsion forces is the null vector). 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 the 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. In Appendix 2 the reader can find more detailed information of these forces.

Attraction force, F_{Attr} . An attraction force that guides the robot towards its goal pose. $RSel$ is attracted to the new task status (e.g.: manipulated object position in a telemanipulation task) whereas every $RCan$ is attracted to its associated $CnTP$. The module of the attraction force in $RCan$ is not proportional to the distance between current robot pose and its assigned $RCan$. A greedy approach is applied, forcing the robots to reach their $CnTP$ in the shortest possible time.

Repulsion force, F_{Rep} . In order to avoid collisions between robots and obstacles, MRCP proposes the use of repulsion forces. Unlike F_{Attr} , which is unique, multiple repulsion force vectors, F_{Rep} , are generated for each robot: one from every obstacle. Following the hierarchical approach, a robot is only affected by repulsion forces generated by the robots with better evaluation (longest tus).

MRCP generates a directional force field around the obstacle convex hull generated by its geometry. This force field is a safety cover of the obstacle and guides the robot to the extreme of the object that facilitates reaching its goal. It also avoids the planner to fall into a local minima.

Random Force. 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 [28], 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 $|F_{Rand}|$ in the final guidance force must be residual. Previous works recommend a range of $w_{Rand} = [0, 0.2]$

Robots Guidance Resulting Force. The final guidance force is the result of the contribution of F_{Attr} , all the F_{Rep} and, if required, F_{Rand} . This force guides the robot to its safety $CnTP$

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 of several fields, e.g. multiple attraction and repulsion force fields) to determine the contribution of each individual field into the final force. Blending functions must generate a smooth vector field, [29, 30]. In MRCP, the blending function is controlled by the l parameter of F_{Rep} , which modules the behaviour of the repulsion forces, (7).

$$F = F_{Attr} + \sum F_{Rep} + w_{Rand} \cdot F_{Rand} \quad (7)$$

where: $F \leq \Delta R_{Max}$ and $0.2 \geq w_{Rand} \geq 0$

Fig.8 illustrates the F_{Attr} , F_{Rep} and final guidance force F computation. Fig.8.a shows the attraction force generated in the minimum distance vector, $\vec{d}_i = g_i - p_i$ ($|\vec{d}_i| \leq \Delta R_{Max}$). Fig.8.b shows the computation of the repulsion force with direction $\Delta \vec{u} = u_i - u_{i-1}$ where u_i and u_{i-1} are the surface points of the obstacle that generate minimum distance vectors and magnitude that depends on the minimum distance vector, $\vec{d}_i$. Finally, Fig.8.c shows the resulting guidance force.

Figure 8: Illustrative example of F_{Attr} , F_{Rep} and final F computation.

6. Experimental Results

This sections reviews four telemanipulation experiments in order to test and validate the proposed MRCP. The first experiment, using a virtual MRCP, presents its internal behaviour showing the evaluation indices and control signals. The second and third experiments use a real scenario with two robots telemanipulating an object. In these experiments the usefulness of MRCP is demonstrated through the execution of a task that could not be accomplished with a single robot. The last experiment, based on a RMIS scenario, shows the improvements of MRCP in precise telemanipulation in terms of execution time, quality of execution and economy of movement with respect to a standard bi-manual teleoperation.

In all these experiments, the robots, other than R_{sel} , should move as close as possible to their pre-grasping positions, minimizing the task transfer time. t_{transf} 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.

The MRCP has been developed following the oriented object paradigm and implemented in C++. The kernel of the MRCP control uses the previously mentioned RPQ library. For the

Figure 9: Block diagram of the experimental MRCP set-up.

experiments, the MRCP control algorithm is executed using a Windows XP Pro platform in an Intel Core2 Duo with 3GB RAM memory equipped with a NVidia GeForce GTX550Ti graphic card that provides a graphical representation of the MRCP and its workspace. The communication between the MRCP and the robot controllers (real or simulated) is determined by the robot controller specifications: 10Mb private Ethernet network, at a frequency of 556 Hz. A complete control cycle (evaluate robots, decide the working mode and compute robot actions) must be executed in less than 18ms (robot communication period) to obtain a smooth teleoperation without time delays.

6.1. Simulated telemanipulation task

The MRCP set-up is composed of a 6DoF master device (3DConnection Space Mouse) and two simulated 6 DoF Stäubli Rx60B robots, both equipped with a gripper. 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. 9 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 positions, from an initial position to a final target point passing through a hole of a moving obstacle. Fig.10 shows a set of snapshots of the task indicating the different causes that force each task transfer. Initially, $R0 = R_{Sel}$ whereas $R1$ is on an initial random configuration, Fig.10a. and, for the complete task execution, three task transfers are required (in $t \approx 5.5s$, $t \approx 11.5s$ and $t \approx 15.5s$), Fig.14.d.

At the beginning of the task, high values are observed for $t_{transf_{R1}}$, Fig.14.b, and for NTT , Fig.14.c, that are consequence of the distance between $R1$ and its grasping position, $CnTP$. Before the first task transfer, Fig.10.a and .b, the CR index between each robot and the moving obstacle determine the two robots' suitability ($texec_{R0} = texec_{CR,R0}$ and $texec_{R1} = texec_{CR,R1}$). 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.10.c and .d, $R_{Sel} = R1$. $texec_{R1}$ 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 $texec_{RJ,R0}$, until $t \approx 8.1s$

Figure 10: Sequence of snapshots of the MRCP executing a manipulation task.

Figure 11: a) R/R_0 and R/R_1 indices. b) Estimated execution time for R/R_0 and R/R_1 .

Figure 12: a) ψ_{R0} and ψ_{R1} indices. b) Estimated execution time for ψ_{R0} and ψ_{R1} .

and, finally by $t_{exec\psi_{R0}}$, as shown in Fig.14.a. The obstacle avoidance trajectory increases $t_{transf_{R0}}$ and NTT ($t \approx 6s$ and $t \approx 8s$), Fig.14.b.

During the third part of the task, $t = 11.5s, t \approx 15.5s$, Fig.10.e and .f, $RSel = R0$ and $t_{exec_{R0}}$ is determined by $t_{exec\psi_{R0}}$ until $t \approx 13s$, and then by $t_{exec_{R/R0}}$. Finally, a joint limit in Θ_5 prevents $R0$ to continue with the task execution, Fig.11. $R1$ executes a collision avoidance trajectory. Notice the temporal unavailability of $R1$ ($t_{exec_{R1}} = 0$) due to an occlusion of $CnTP$ as well as the increasing values of $t_{transf_{R1}}$ and NTT . Once $CnTP$ is reachable, $t_{exec_{R1}}$ is determined by $t_{exec_{R/R1}}$ and finally by $t_{exec_{\Theta_{R1}}}$.

Finally, from $t = 15.5s$ to $t \approx 20.5s$, $R0$ reaches a joint limit, Fig.10.g and $RSel = R1$, Fig.10.h. Finally, ψ_{R1} index indicates the vicinity of a singularity (alignment of two links), Fig.10.h, indicating the end of the $R1$ availability.

Figure 13: 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 14: a) Execution time, t_{exec} ($t_{exec} = 0$: robot is not suitable). b) Required Task Transfer, t_{transf} . c) NTT index evolution. d) $RSel$

6.2. Telemanipulation tasks

The second experiment consists 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 two real 6DoF Stäubli RX60B robots equipped with pneumatic grippers as slave station.

The first telemanipulation task, Fig. 15, 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 (R_{can} changes its $CnTP$). Initially, Fig.15.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.15.b and .c, until it approaches a joint limit in Θ_5 ($t_{exec_{R1,R1}} \rightarrow 0$ and $NTT \rightarrow 1$). After the task transfer, Fig.15.d, $R0$ continues with the task execution while $R1$ moves towards its best evaluated $CnTP$, Fig.15.e and .f.

In the second telemanipulation task the user 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 positions. A set of snapshots of the task execution is shown in Fig.16.

Initially $R0 = RSel$ (left on the images), whereas the other robot, $R1$ (right), is on a random initial configuration, Fig.16.a. As soon as the task is started, $R1$ moves towards the best $CnTP$, as shown in Fig.16.b. A task transfer is needed when $R0$ gets close to reach a joint limit (Θ_5), Fig.16.c. Once the task transfer is finished, $R1$ continues with the task, Fig.16.d and .e, while $R0$ changes its grasping position following the maximum tus criteria. In Fig.16.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, Θ_3 , Fig.16.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.16.j and .k. Finally, $R1$ stops due to an imminent collision. $R0$, using the new grasping position (top of the cube) leaves the object on the table without colliding.

6.3. Performance analysis: Standard teleoperation vs MRCP in precise teleoperation task

While the previous experiments are focused in the study of internal behaviour of MRCP, this last experiment is oriented to evaluate MRCP in terms of operator performance in dexterous teleoperation. A precise telemanipulation task, in the RMIS field (suturing simulation), has been used to force the operator to execute a high dexterous task. Three teleoperation modes are used: a standard bi-manual teleoperation (2M:2S), where each master controls a slave robot and a MRCP (1M:2S), where the operator guides the object with a single master (MRCP decides which robot executes the task at every evaluation time). Finally, and using the remaining free hand when using MRCP, a second teleoperation system is included to control the remote workspace view. A robot holding the laparoscopic camera guided by a 6DoF mouse as master device (1M:1S). The results of all experiments 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 due to undesired collisions).

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.

Set-up. The set-up is composed of a master station with two Phantom Omni, a 6DoF mouse and a screen showing the laparoscopic view, Fig.17.a. The operator screen do not offer any additional information. There is no information about the robot executing the task to preserve the transparency underlying in MRCP. The slave station is composed of three 6DoF robots (Stäubli 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.17.b. To control 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 control their positions. The rings are hold up with a spring, which is deformed when the rings and the needle collide. With respect to the material, it offers high grip in order to difficult the task when the needle contacts it. The leds position is used to control the quality of execution (measure 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 on the workspace, as shown in Fig.17.c. The schematic representation of both systems is shown in Fig.18.

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 trajectory of the needle. Q, which is defined as the trajectories described by the leds attached to the target points, measures the trajectory of the targets when their pose vary due to collisions with the needle or with the laparoscopic tools. The target movements emulate tissue deformation.

During the task execution all relevant data are stored on a file for its further analysis. The master, slave and targets (leds attached to them) pose, the time stamp, the grasper status (open/closed), the RobSel (in MRCP and MRCP+Cam modes) and the pose of the laparoscopic camera (in MRCP+Cam) is 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 the last part 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 standard teleoperation vs MRCP and MRCP vs MRCP+Cam in terms of generic usefulness and expected results (subjective performance in ET, EM and Q). It 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 felt that was 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

Figure 15: Snapshots sequence of the first real teleoperation experimental task.

Figure 16: Snapshots sequence of the second real teleoperation task.

Figure 17: Master console (left), Remote workspace with three robots (center) and the scenario with the rings and the needle (right).

Figure 18: Schemas of standard teleoperation (left) and MRCP (right) setups.

participants had different teleoperation experience (from null to experienced users) to obtain a full-scale spectrum to compare the performance of MRCP in the whole range of expertise.

The experiment is divided in four parts: first the subjects received an introductory explanation of the system and the task to be executed. Second, the subjects were invited to test all three teleoperation modes with the supervision of an expert. All

subjects execute 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 (standard bi-manual teleoperation, MRCP and MRCP+Cam). The order of the modes was randomized to cancel undesired effects of learning process and fatigue. Finally, the subjects reply a set of questions evaluating their experi-

ence and subjective evaluation with each proposed teleoperation modality. At every trial, the initial conditions are identical: all three robots on an initial position, the master devices on their holders and the pedals not pressed. Finally, the needle is on a holder with the robot that initiates the task over it. Setting the initial configuration lasts approximately one minute, that is 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, 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 the MRCP.

6.3.1. 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 standard teleoperation. The statistical analysis of the collected data shows a significant improvement in all three evaluation indices between the proposed teleoperation methods (MRCP and MRCP+Cam) compared to standard teleoperation. 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 2.

Table 2: Mean and standard deviation for all users in all trials. (N: Best, □: Medium and H worst result).

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

Reviewing the evaluation criteria between NoMRCP 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 (subject moves a single object in the remote space in MRCP whereas in NoMRCP the subject controls two different objects). 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 factor) due to diverse factors: MRCP dynamically controls the position of the targets (using the attached IR-leds) eliminating the collisions of the robot that do 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 between them in any of the evaluation criteria. The benefits of positioning the camera are cancelled by different aspects: First, the subjects had to learn and manage a second teleoperation system with a different master device. The difficulty increases when both teleoperated systems 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.

More compact interquartile ranges are obtained in MRCP and MRCP+Cam in all three evaluation criteria (ET, EM and Q), as shown in Fig. 19, demonstrating that the proposed task-oriented teleoperation improvements are independent from the teleoperation experience (benefits are transversal for the whole population sample).

Learning curves, Fig. 20, show an improvement in all three teleoperation methods during the sequence of trials. The evolution of the subjects do not show a higher learning curve in any teleoperation mode or in any evaluation parameter. Extrapolating the learning curve suggests that no intersection between NoMRCP and MRCP or MRCP+Cam will occur. ET and EM learning curves need more trials to reach the asymptotic phase. Q learning curve stabilizes at the 5th trial. Executing 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).

In Table 3 several statistics are presented comparing the results of paired tests between the three teleoperation modes (Standard vs MRCP, Standard vs MRCP+Cam and MRCP vs MRCP+Cam). The statistical signification is validated by the paired sample Student's t-test applied to ET, EM and Q. There is a significant difference between standard and MRCP or MRCP+Cam, but not between MRCP and MRCP+Cam. The inclusion of the possibility to move the camera has not generated 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 the 0.9. The first column 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 standard and MRCP or MRCP+Cam is noticeable, generating two separate distributions. Again, this test do 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 of each teleoperation mode. Again, there is a significant improvement between standard and MRCP or MRCP+Cam teleoperation modes. The highest benefits are obtained in quality, all over a gain of 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. Fig. 21 shows the results of each subject (mean of all trials) in ET, EM and Q in all three teleoperation modes (upper graphics) and the obtained gains (lower graphics).

Figure 19: Mean, median, 25th and 75th quartiles, maximum, minimum and outliers observed values for standard, MRCP and MRCP+Cam.

Figure 20: Learning curves of ET, EM and Q during the six trials for all three teleoperation modes.

Figure 21: Mean of results obtained per subject for ET, EM and Q (top) and obtained gains (bottom).

Fig.22 shows the classification of the gains in three groups: > 1.05 , $[1.05, 0.95]$, < 0.95 and the percentage of the subjects inside each group.

6.3.2. Subjective Analysis

The subjective analysis compares the proposed methods MRCP vs NoMRCP and MRCP+Cam vs MRCP in terms of perceived general improvement of the proposed task-oriented teleoperationan (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 the task-oriented teleoperation (MRCP and MRCP+Cam) and the degree of achieved abstraction (operator executes the task).

In Fig. 23 the result of the subjective analysis is shown. Concerning MRCP vs NoMRCP, 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

Table 3: Add caption

Type of test			Paired t-test		Power Test		Gains
	Test id 0	Test id 1	value	P	Min Subjects (P=0,9)	P with N=20 subjects	Test0 —— Test1
ET	NoMRCP	MRCP	6,69	2,14E-06	17	0,96	1,41
	NoMRCP	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	NoMRCP	MRCP	5,76	1,48E-01	19	0,90	1,18
	NoMRCP	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	NoMRCP	MRCP	7,49	4,39E-03	9	1,00	2,97
	NoMRCP	MRCP+Cam	5,68	1,79E-01	9	1,00	2,77
	MRCP	MRCP+Cam	-49,00	0.62	584	0,09	0,93

Figure 22: Percentage of the samples in each gain group, 1,05, [1,05, 0,95], < 0,95.

just the 40% in ET and 45% in EM. These results are consistent with the expectations of MRCP+Cam: helps to obtain the desired view to perform a higher quality task execution. On the contrary, the camera positioning increases the execution time, especially in the subjects with less teleoperation experience that had to learn the use of a new master device system and control an additional teleoperation system.

At the end of the experiment a subjective evaluation was done by the subjects filling a questionnaire. They had to evaluate the comfortableness with the proposed task-oriented teleoperation and the achieved level of abstraction (guide the needle instead of the robots). 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 (task-oriented teleoperation). Table4 shows the statistics of the subjective analysis.

Tests 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. A simplified analysis dividing the levels between three (gain, null gain, loose off performance) provides better results.

7. Discussion and Conclusions

In this paper, a novel open platform for task oriented teleoperation has been presented, the Multi-Robot Cooperative Platform. MRCP is designed to enable a teleoperation focused on the task itself rather than on the slave robots. The system, in spite of its current limitations (e.g. pre-defined grasping points, non deformable objects and obstacles, etc) is generic enough to be considered as an alternative or complementary to standard teleoperation. MRCP is based on a multi-layered control that continuously evaluates the robots suitability, selects the most convenient one to continue with the teleoperation task, decides when a task transfer is required and, finally, computes the robot actions for all robots.

From the presented evaluation indices, the availability of each robot to continue with the task is estimated. The number and nature of these indices is diverse and covers all the causes that can prevent the robots to continue with the teleoperated task. A time space transformation allows the direct indices comparison. The control is an open platform designed to allow the introduction of new indices.

The first three experiments are addressed to illustrate the internal operation of MRCP. The cooperative behaviour of MRCP enables a continuous and smooth teleoperation. Several critical situations (e.g. joint limit, singularities, collisions, etc) force task transfers that had been solved in advance by the MRCP

Figure 23: Results (absolute and percentage) of the subjective evaluation of the subjects at the last phase of the experiment

Table 4: Mean and Standard Deviation of the subjective analysis

Teleop type	MRCP vs Standard				MRCP+Cam vs MRCP				Task oriented with MRCP	
Question	Useful	ET	EM	Q	Useful	ET	EM	Q	Feel Comfortable	Abstraction
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

control. The number of task transfers has been minimized using the NTT index.

The last experiment focusses on the study of the operators performance using the proposed task oriented paradigm in a precise telemanipulation task. Several benefits have arisen with this experiment. First, the reduction of execution time and the improvement on the movement economy due to the automatic task transfer. Second, an improvement of the execution quality and a reduction on the execution time using the MRCP teleoperation paradigm. Finally, the use of the remaining free hand to, for instance, dynamically guide the laparoscopic camera, provides the desired view at every instant, thus improving the execution quality if the operator is trained enough. In these robotic systems with more slave arms than master devices, the operator must select which slave will be teleoperated with each master device, remaining the rest of the slaves inactive. The selection of the robot imposes a loose of time that is cancelled with the MRCP. The subjective analysis indicates a general acceptance of the proposed teleoperation method and can be used as a guide to improve several aspects of the MRCP.

Appendix A. Evaluation Indices

Appendix A.1. Joint Limits index, RJ

The Joint Limits index, RJ , evaluates the risk of the robot joints of reaching their limits. The distance to these limits is modulated by means of a smooth and continuous function which warns about the risk only when θ_i is close to one of its limits. The RJ is computed as (A.1), where $\text{dist}(\theta)$ is the minimum distance between each joint position and its limits.

$$J_{eval} = (1 - \cos(\text{dist}(\theta)^{1/b}\pi))/2, \quad b \geq 1 \quad (\text{A.1})$$

Figure A.24: Modulation of the Joint Limit index in a rotational joint. Range of the joint: $[\theta_{min}, \theta_{max}]$, and $b \in [1, 6]$

The b modulation parameter introduces the approximation velocity into the risk evaluation. Numerically, b varies inversely proportional to the ratio between the current and the maximum joint velocities (A.2). The recommended b values vary between $b = 1$ and $b = 6$ for high and low velocities respectively. Values lower than 1 degenerate the desired function behaviour. Values higher than 6 generate a central plateau which limits are too close to the joint limits.

$$b_i = \frac{(b_{max} - b_{min})\dot{\theta}_i}{\dot{\theta}_{max,i}} + b_{min}, \quad \forall i = 1..n \text{ joints} \quad (\text{A.2})$$

Fig.A.24 shows the functions $\text{dist}(\theta)$ and J_{eval} with different values of the modulation parameter b , ($b \geq 1$), for a rotational joint with limits $[\theta_{min}, \theta_{max}]$.

The complete robot joint evaluation, RJ index, is determined by the minimum of all the J_{eval} , (A.3).

$$RJ = \min(J_{eval,i}) \quad \forall i = 1..n \text{ joints} \quad (\text{A.3})$$

Fig.A.25 illustrates an example of the RJ index evolution of two robots during the execution of a telemanipulation task

Figure A.25: RJ index for two robots executing a teleoperated task. The first graphic shows the evolution of RJ_0 and RJ_1 . The two bottom graphics show the J_{eval} of each joint for both robots.

consisting in rotating an object. The object has two predefined grasping positions and the robots have 6 joints.

Appendix A.2. Directional manipulability index, ψ_u

Manipulability, defined as the ability of a robot to perform a movement given a specific configuration [31], has been used extensively as evaluation index, [32, 33, 34]. The manipulability, w , is based on the Jacobian matrix of the manipulator, $w = \sqrt{\det(J(q)J'(q))} = \sigma_1\sigma_2\ldots\sigma_m$, where $q = q_1, \dots, q_n$ is the robot joint configuration, $J(q)$ is the Jacobian matrix in q , σ_i its i th singular value and $m \geq n$ the task space dimension.

Manipulability can be expressed as an ellipsoid in the robot workspace. The ellipsoid volume, which is proportional to w , denotes the ability of a manipulator to perform a movement.

The directional manipulability index, ψ_u expresses the manipulability of a robot moving along a specific direction, (A.4), where q is the robot configuration, u is the desired direction and $\dot{x}$ the robot velocity vector.

$$\psi_u = \frac{\|\dot{x}\|}{\|\dot{q}\|} = \frac{1}{\sqrt{u'(JJ')u}} \quad (\text{A.4})$$

Geometrically, ψ_u represents the distance from the center of the manipulability ellipsoid to the point where u intersects with the ellipsoid surface. In MRCP, the u_i directional vector at $t = i$ is determined by extrapolating the value of the second order regression polynomial obtained from the last observed robot positions.

Fig. A.26 shows a set of snapshots of a 6 DoF robot following a vertical trajectory, $p_i = p_{i-1} + \Delta_z$, 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.A.26.c and $t \approx 2.5s$, Fig.A.26.d) are produced by the alignment of joints 3 and 5, whereas the last one ($t \approx 3.6s$, Fig.A.26.g) is produced by a boundary singularity. In the same figure, three different ψ_u are shown; each one corresponds to three different directions: $u_i = p_{i-1} + \Delta_x$, $u_i = p_{i-1} + \Delta_y$ and $u_i = p_{i-1} + \Delta_z$. The last graphic shows the robot joints positions along the trajectory. Observing the evolution of θ_3 and θ_5 the

Figure A.26: Top: robot executing a trajectory and the associated manipulability ellipsoids. Middle: ψ_u in three different trajectory directions. Bottom: Evolution of the joint positions.

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.

Appendix A.3. Collision Risk index, CR

The robots that compose the MRCP share their workspace with each other and with other moving obstacles, generating a dynamic and complex workspace that requires a strict collision detection and an efficient obstacle avoidance control strategy.

Classical collision avoidance approaches use distance as a unique parameter to determine the risk of impact between two objects. To obtain a more realistic collision risk evaluation, the relative direction of the movement between objects and their estimated time to impact must also be studied. In [35], the collision risk is computed using the relative velocity between two objects and their maximum achievable acceleration, which determines their reaction capability to avoid a collision. Based on the same principle, here, the Collision Risk index, CR , uses the estimated time to impact, $timpr_i$, as well as the time required for a task transfer, t_{transf} .

$timpr_i$ is determined by obtaining the minimum distance vectors from each robot link to all the obstacles in the workspace (treating the rest of robots as obstacles). Using the distances evolution, the estimated times to impact of each robot are obtained. In order to smooth the evolution of minimum distance vectors, as well as to reduce the computation complexity, each robot link is represented using its convex hull. Fig.A.27 illustrates the minimum computed distance between three robots co-operating in a teleoperation application in front of an external obstacle (a camera for the object visual inspection).

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, (A.5).

Figure A.27: Visualization of the minimum distance computed between three robots and an external obstacle.

$$CR = \frac{timpr'_r}{\max(timpr'_1, \dots, timpr'_R)}, \forall r = 1..R \quad (A.5)$$

Appendix B. Guidance Forces

This appendix extends the guidance forces introduced in Robots Actions Computation of Section 5.

Appendix B.0.1. 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 a safety position close to CnTP, g_i . One of the firsts formal approaches to an attraction force was developed in [36, 37], defining an attractive force field that depends on the distance between robots and obstacles; where the amount of force increases quadratically with distance. MRCP follows a greedy approach: reach the goal position 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 lower than this maximum movement, (B.1). The direction of the attraction force is determined by the versor of the distance vector, $\hat{d}_i$.

$$F_{Attr,i} = \begin{cases} \Delta R_{Max} \cdot \hat{d}_i & \text{when } |\vec{d}_i| \geq \Delta R_{Max} \\ \vec{d}_i & \text{when } |\vec{d}_i| < \Delta R_{Max} \end{cases} \quad (B.1)$$

Appendix B.0.2. Repulsion force, F_{Rep}

Several proposals have been done defining repulsive force fields. Again, as one of the first formal proposals, in [36, 37], a distance dependent repulsive field U_{Rep} is defined, (B.2).

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

Repulsion forces, like those presented above, 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). A possible solution is the use of potential fields based on primitives applied to the obstacle surfaces (e.g. uniform, perpendicular, tangential or random force

fields). The combined use of these primitives and the attraction force field smoothly guides the robot towards the goal, overcoming the previous mentioned problems. Several examples of the combined use of potential fields can be found in [38, 39]. MRCP proposes the use of a directional force field, F_{Rep} , around the object geometry. F_{Rep} can be defined in terms of its magnitude, $h(\vec{d}_i)$, and its direction, $-\Delta \hat{u}$, (B.3). The direction of F_{Rep} is determined by the evolution of the minimum distance vector over the obstacle surface, expressed as the versor $\Delta \hat{u} = (u_i - u_{i-1}) / |u_i - u_{i-1}|$.

$$F_{Rep,i} = \begin{cases} \Delta \hat{u} \cdot h(\vec{d}_i) & \text{when } |\vec{d}_i| > |\vec{d}_{i-1}| \\ 0 & \text{when } |\vec{d}_i| \geq |\vec{d}_{i-1}| \end{cases} \quad (B.3)$$

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. It varies inversely proportional to the distance: $|\vec{d}_i| \rightarrow 0 \Rightarrow F_{Rep} \rightarrow \Delta R_{Max}$ and, on the contrary, $|\vec{d}_i| \rightarrow \infty \Rightarrow F_{Rep} \rightarrow 0$.

$$h(d_i) = \frac{\Delta R_{Max}}{1 + |d'_i|^l} \quad \text{where } l \geq 1 \quad \text{and} \quad d'_i = d_i + d_{offset} \quad (B.4)$$

The parameter l modules the response of the function, determining the shape of the repulsion force around the obstacle: increasing l , the force slope increases. When the obstacle is isolated, low l values are recommended, generating a smoother force response. When the position of obstacles produce narrow passages, higher values of l generate safe corridors. Fig.B.28 shows $h(d_i)$ with different values of l .

Figure B.28: Examples of proposed potential field with different values of the parameter l : a) $l = 1$, b) $l = 2$, c) $l = 4$ and d) $l = 8$. No offset is present, $d_{offset} = 0$.

References

- [1] P. Arcara, C. Melchiorri, Control schemes for teleoperation with time delay: A comparative study, *Robotics and Autonomous Systems* 38 (2002) 49 – 64.

- [2] G. H. Ballantyne, The pitfalls of laparoscopic surgery: challenges for robotics and telerobotic surgery, *Surg Laparosc Endosc Percutan Tech* 12 (2002) 1–5.
- [3] J. Y. C. Chen, E. C. Haas, K. Pillalamarri, C. N. Jacobson, Human-Robot Interface : Issues in Operator Performance , Interface Design , and Technologies, *Engineering* 3834 (2006) ARL Technical Report ARL–TR–3834.
- [4] J. Y. C. Chen, E. C. Haas, M. J. Barnes, Human Performance Issues and User Interface Design for Teleoperated Robots, 2007. doi:10.1109/TSMCC.2007.905819.
- [5] C. E. Lathan, M. Tracey, The Effects of Operator Spatial Perception and Sensory Feedback on Human-Robot Teleoperation Performance, 2002. doi:10.1162/105474602760204282.
- [6] S. Chiaverini, Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators, *Robotics and Automation, IEEE Transactions on* 13 (1997) 398–410.
- [7] M. Shimizu, H. Kakuya, W. K. Yoon, K. Kitagaki, K. Kosuge, Analytical inverse kinematic computation for 7-dof redundant manipulators with joint limits and its application to redundancy resolution, *Robotics, IEEE Transactions on* 24 (2008) 1131–1142.
- [8] H. P. Xie, R. V. Patel, S. Kalaycioglu, 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*, volume 3, 1998, pp. 1925–1930 vol.3.
- [9] A. D. Luca, 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.
- [10] C. M. Rwan, Redundancy resolution by minimization of joint disturbance torque for independent joint controlled manipulators, in: *Advanced Intelligent Mechatronics, 1999. Proceedings. 1999 IEEE/ASME International Conference on*, 1999, pp. 392–397.
- [11] Y. and Cheung, J. H. Chung, Semi-Autonomous Control of A Multi-Agent Robotic System for Multi-Target Operations, *International Journal of Robotics & Automation* 2 (2011) 107–127.
- [12] Y.-C. Liu, N. Chopra, Control of semi-autonomous teleoperation system with time delays, *Automatica* 49 (2013) 1553–1565.
- [13] A. Hernansanz, J. Amat, A. Casals, Optimization criterion for safety task transfer in cooperative robotics, in: *Advanced Robotics, 2009. ICAR 2009. International Conference on*, 2009, pp. 1–6. ID: 1.
- [14] M. Saha, 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.
- [15] M. Gharbi, J. Cortes, T. Simeon, A sampling-based path planner for dual-arm manipulation, in: *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, 2008, pp. 383–388.
- [16] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, D. Kragic, Dual arm manipulation A survey, *Robotics and Autonomous Systems* 60 (2012) 1340–1353.
- [17] J. Krüger, G. Schreck, D. Surdilovic, Dual arm robot for flexible and co-operative assembly, *CIRP Annals - Manufacturing Technology* 60 (2011) 5–8.
- [18] F. Zacharias, D. Leidner, F. Schmidt, C. Borst, 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.
- [19] 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.
- [20] M. J. Hwang, D. Y. Lee, 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.
- [21] Y. Koga, 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.
- [22] N. Y. Chong, T. Kotoku, K. Ohba, K. Komoriya, N. Matsuhira, K. Tanie, Remote coordinated controls in multiple telerobot cooperation, in: *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 4, 2000, pp. 3138–3143.
- [23] I. Farkhatdinov, J.-H. Ryu, Teleoperation of multi-robot and multi-property systems, in: *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on*, 2008, pp. 1453–1458.
- [24] T. Fukuda, K. Kosuge, Advanced telerobotic systems single-master multi-slave manipulator system and cellular robotic system, *Teleoperation: Numerical Simulation and Experimental Validation* 4 (1992) 195–208.
- [25] N. Y. Chong, T. Kotoku, K. Ohba, K. Komoriya, N. Matsuhira, K. Tanie, Remote coordinated controls in multiple telerobot cooperation, in: *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 4, 2000, pp. 3138–3143 vol.4.
- [26] T. B. Sheridan, *Telerobotics, automation, and human supervisory control*, MIT Press, Cambridge, MA, USA, 1992.
- [27] A. Hernansanz, J. Amat, A. Casals, Virtual robot: A new teleoperation paradigm for minimally invasive robotic surgery, in: *Biomedical Robotics and Biomechanics (BioRob), 2012 4th IEEE RAS EMBS International Conference on*, 2012, pp. 749–754. doi:10.1109/BioRob.2012.6290803.
- [28] 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.
- [29] S. C. Costa, Behaviour blending for multiple robot coordinated, Ph.D. thesis, Universidad Complutense de Madrid, 2013.
- [30] M. A. P. Castañeda, J. Savage, A. Hernández, Local Autonomous Robot Navigation using Potential Fields, in: X.-J. Jing (Ed.), *Motion Planning*, 2008, pp. 1–23.
- [31] Y. T., Manipulability of robotic mechanisms, *The International Journal of Robotics Research* 4 (1985) 3–9.
- [32] C. A. Klein, B. E. Blaho, Dexterity measures for the design and control of kinematically redundant manipulators, *Int.J.Rob.Res.* 6 (1987) 72–83.
- [33] K. Jin-Oh, K. Khosla, Dexterity measures for design and control of manipulators, in: *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems IROS, 1991*, pp. 758–763.
- [34] L. Zlajpah, Dexterity measures for optimal path control of redundant manipulators, ? (1996).
- [35] E. Freund, 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*, volume 2, 2003, pp. 1173–1177.
- [36] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, in: *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, 1985, pp. 500–505.
- [37] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *Autonomous Robot Vehicles* (1990) 396–404.
- [38] R. R. Murphy, *Potential Fields Methodologies, Introduction to AI Robotics*, 1st ed., MIT Press, Cambridge, MA, USA, 2000, pp. 105–153.
- [39] R. C. Arkin, *An Behavior-based Robotics*, 1st ed., MIT Press, Cambridge, MA, USA, 1998.

Received the B.S. degree and PhD degree from the UPC. From 1991 professor at UPC. She is currently leading the research group on Robotics of IBEC, and is member of the research group GRINS: Intelligent Robotics and Systems at UPC. The research is oriented to improve human robot interaction through multimodal perception, focused mainly in the area of medical robotics. In this field she is working both in rehabilitation, assistance and surgical applications.

Received the B.S. degree in Computer Engineering from the Universitat Autònoma de Barcelona (UAB) in 2002. From the 2000 has been working in several research projects in the Automatic Control Department robotics laboratory at the Universitat Politècnica de Catalunya. BarcelonaTech (UPC). He is currently finishing the PhD and working at the Institute for Bioengineering of Catalonia (IBEC).

Received his B.S. degree in Electrical Engineering in 1967 from the School of Engineering of Barcelona, and obtained his PhD in 1977 from the UPC.

He is Professor Emeritus at UPC. His research work initially was in the field of Digital Signal Processing. Since 1980 his work is oriented to the Computer Vision field for Robotics applications, especially in Medical and Surgical Robotics. He leads the research in this field in the Center of research of Biomedical Engineering of UPC.

*Photo of A Casals

*Photo of A Hernansanz

*Photo of J Amat