Virtual Robot: A new teleoperation paradigm for minimally invasive robotic surgery

Albert Hernansanz¹, Member IEEE, Josep Amat² and Alcia Casals¹ Senior Member IEEE

Abstract—This paper presents a novel teleoperation paradigm, the Virtual Robot (VR), focused on facilitating the surgeon tasks in minimally invasive robotic surgery. The VR has been conceived to increase the range of applicability of traditional master slave teleoperation architectures by means of an automatic cooperative behavior that assigns the execution of the ongoing task to the most suitable robot. From the user’s point of view, the VR internal operation must be automatic and transparent. A set of evaluation indexes have been developed to obtain the suitability of each robot as well as an algorithm to determine the optimal instant of time to execute a task transfer. Several experiments demonstrate the usefulness of the VR, as well as indicates the next steps of the research.

I. INTRODUCTION

This paper paper presents a novel proposal aiming to increase the applicability of current robotic systems in minimally invasive surgery (MIS). The proposed approach should facilitate surgeons in the execution of different surgical tasks by means of changing the paradigm of classical teleoperated systems. This approach is based on the cooperative use of the various robotic arms that current robotic minimally invasive surgery systems (RIMS) provide.

From the surgeons point of view, open surgery provides absolute 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 [1] a review of the haptic information in MIS is presented and in [2] a case study of haptic feedback in a concrete laparoscopic surgery is described. Surgeons have to perform an intervention using long tools with their movements restricted by the fulcrum point, which reduce their dexterity and Dos. Visual feedback is also limited by the endoscopic camera, providing partial 2D information. Two hands procedures, like suturing, are outstandingly affected by these restrictions. In RIMS, except for some research works [3]–[5], null tactile feedback is provided. The surgeon dexterity is also restricted by the tools (apart from the mentioned geometry, a reduced set of tools are available, affecting some surgical procedures), and the fulcrum point. More recent tools offer improved dexterity thanks to the higher number of DoFs with respect to those used in MIS. RIMS also offers some other performances as, tremor cancellation [6], motion scaling [7], [8] or trajectory guidance [9]. Visual feedback, although indirectly provided by a camera, is improved using stereoscopic vision, camera stabilization and motion tracking [10].

The Virtual Robot concept (VR), introduced in [11], has been conceived as a further step in dexterous telemanipulation. The goal is that in teleoperation systems provided with more than one robot, the surgeon pays attention more on the task itself, than in the specific actions and movements of each robot. Thus the use of VR becomes a feasible solution to be applied to the growing number of surgical multi-robot systems, [12], [13].

Apart from facilitating the surgeons task, the VR has been conceived with the aim of overcoming the limitations of robots in surgery in terms of kinematics or the own robot workspace limitations. These intrinsic causes prevent robots from completing the desired task, requiring their repositioning. They also introduce new disturbances to the surgeon. The limitations are also related to the surgical workspace. Examples of these extrinsic causes are collisions, occlusions, patient safety, etc. 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.

This paper is structured as follows: first, in Section II, the change of teleoperation paradigm is explained. Then, Section III describes the VR control architecture, as well as the different robots evaluation criteria and the robot selection algorithm. Once the VR is explained, in Section IV several operability tests are presented. Finally some conclusions and future work are described.

II. THE VIRTUAL ROBOT TELEOPERATION PARADIGM

Current RIMS rely on teleoperated systems with a master console consisting of a set of master devices that drive the slave robots; a vision system that provides the endoscopic camera view and some extra control elements like, for instance, pedals used as clutches to activate the slave movements, vary the camera magnification, motion scaling, etc. RIMS use classical teleoperation schemas, in which each master device movement is mapped on a slave robotic arm:
1M : 1S or 1M : 1S + 1M : 1S in bimanual teleoperated systems. Newer RIMS systems offer the surgeon the possibility of actively select which slave robot executes the movements of each master: 1M : nS or 1M : nS + 1M : nS. For instance, DaVinci robot by Intuitive is composed of two master devices and three robotic arms plus an additional one that holds the endoscopic camera. The surgeon can manually select on the master console which of the robots maps the movements of each master device. Fig. 1 illustrates the block diagram of a 2 : n teleoperated RIMS system.

Fig. 1. Block diagram of a 2 : n teleoperated RIMS system.

VR changes the previous schema, automatically selecting which of the robots executes the teleoperated task at each time. The master device is not used to drive the robot but to define the task actions. Fig. 2 illustrates the block diagram of a VR RIMS teleoperated system, where the VR has been embedded inside the general RIMS control unit.

Fig. 2. Block diagram of a 2 : n teleoperated RIMS system.

Reviewing the schemas shown in Fig. 1 and Fig. 2, two differences must be pointed out. First, in RIMS, the master-slave assignation is outside the control unit, while in VR the selection process is part of the control unit itself. Second, in the first schema, the user attention and action is divided between left and right hand, whereas in VR is divided in dominant and non-dominant hands. This means that with the second approach, surgeons can execute dexterous tasks with their dominant hand, independently of which slave robot arm executes it.

For a better understanding of VR implications, let’s suppose a suturing task, in which a needle has to be transferred from one robotic arm to another. Using the classical RIMS approach, the surgeon guides the tools and decides when a task transfer is necessary, re-grasping the needle from the exiting part, which requires high precision and results in a tedious and time consuming task. With the VR, the surgeon uses a master device to guide the needle: using a single master device, the movements of which are directly mapped on the needle. The control architecture automatically decides which robot grasps the needle (executes the task) and when the task transfer must be executed, automatically executing the re-grasping process.

III. VIRTUAL ROBOT

The Virtual Robot has been conceived to increase the range of applicability of traditional master slave teleoperation architectures by means of a cooperative behavior. Based on the automatic detection of the causes that prevent the execution of a teleoperated task as well as on the generation of the required transfers from one slave robot to another, the VR must guaranty the successful task completion.

From the users point of view, the VR internal operation must be automatic and transparent: The robot that carries out the task must not influence the user behavior or the task execution. An important aspect that must be taken into account is the minimization of perturbations on the teleoperated task in terms of time delays introduced by task transfers. To overcome this problem, the VR control schema must act in advance, preventing stopping the execution of the task, executing the required transfer before the robot falls into a critical situation and positioning the candidate robot to continue with the task execution in the vicinity of a task transfer region. Following this approach, the surgeon efforts can be focused on the task itself and not on the slave robots.

A. VR Control Architecture

The VR control architecture is responsible of determining which is the most suitable robot to continue with the task execution, deciding when a task transfer must be performed, as well as generating the robot actions to execute the task and facilitate the task transfer process.

The VR control is based on the Robotic Proximity Queries, RPQ, [14], [15], robotic simulation library. This library provides a platform to control several robots working into dynamic workspaces, including moving and deformable obstacles. Different robot evaluation indices can be computed with RPQ. The VR control also includes a set of configurable virtual fixtures for guidance and defining forbidden or restricted regions, as defined in [16]. It controls the velocity of the manipulated object as well, injecting artificial viscosity in the form of force feedback into the master device, varying the motion scale between the master and the manipulated object and generating audio and visual feedback.

The VR control architecture is organized in three sequential steps: a) Dynamic evaluation of robots suitability, which is the process of determining, in a predictive manner, the suitability of each robot to continue with the ongoing task. b) Based on each robots suitability, the next step determines the best robot to continue with the task execution and if an immediate task transfer between robots is required. c) Finally, the actions to be executed between the current and the next evaluation are planned and sent to the robot controllers. This is a hierarchical process: the robot that is currently executing a task has maximum priority, whereas the rest of robots must adapt their actions in order to not disturb the task and, if possible, be ready to facilitate, as much as possible, a future task transfer.
RPQ is able to control real set-ups by means of the communication layer that uses sockets to acquire data from the workspace and the robots and generate commands to the robots controllers. The VR requires information of the workspace status. For the purpose of this research work, the required information is: configuration of the slave robots, the abdominal organs in terms of their geometry, position and deformation and the new users orders (e.g. new desired pose of the manipulated needle). The block diagram of the VR control architecture is shown in Fig. 3.

Fig. 3. VR control diagram, including the user, master haptic device and the remote workspace with the robot controllers.

B. Robot Suitability Evaluation

Robot suitability evaluation provides a complete estimation of the adequacy of every robot composing the VR to continue with the execution of the ongoing teleoperated task.

The evaluation criteria have been defined considering: a) the different causes (intrinsic, extrinsic and task dependent) that can prevent the robots from continuing the execution of the ongoing task, b) the need of acting in advance to avoid stopping the task execution, c) the uncertainty produced by the lack of path planning in teleoperation and d) the computational cost required to achieve real time operation.

Due to the lack of path planning it is necessary to predict the future path so as to be able to foresee task evolution, and thus, decide future actions in advance. Prediction minimizes the potential disturbances that might affect the teleoperated task, e.g. object regrasping actions are minimized.

C. Trajectory estimation

The VR does not require a specific trajectory predictor and can be selected according to the application and its specifications. From the VR point of view, the prediction methodology must provide a set of predicted points expressed as a probabilistic distribution function, $\Omega(p_i, d_i)$.

In order to decide which type of trajectory predictor fits to the requirements of the type of trajectories, those presented in the experimental results section, different trajectories inside the virtual workspace have been executed and their dynamics analysed. Positions are predicted using a second order polynomial, which is enough to take velocity variations into account. The coefficients of this polynomial are calculated using the least squares fitting technique. To represent the orientation of each trajectory point, quaternions are used to obtain an efficient and concise representation of rotations and do not suffer from singularities. In order to predict orientations, the spherical linear interpolation technique is used.

D. Robot Evaluation indexes

To evaluate the causes that prevent a robot from continuing the execution of a task, several evaluation indexes have been defined: joint limit index, anisotropic dexterity index, collision risk and tool orientation with respect to the trocar. The first two indexes measure intrinsic robot characteristics, the third one is an extrinsic measure and, finally, the last one constitutes a task dependent criterion. The open control architecture allows the introduction of new evaluation index without altering the control schema. The definition of these indexes, briefly mentioned here, are deeper treated in [11].

1) Joint Limits index, $R_J$: The Joint Limits index, $R_J$, evaluates the risk of a robot to reach a joint limit (linear or angular) during a trajectory execution. The risk evaluation is based on the modulated minimum distance between the current joint position, $\hat{\theta}$, with respect to its limits $\text{dist}(\hat{\theta}) = \min(|\hat{\theta} - \hat{\theta}_{\text{Max}}|, |\hat{\theta} - \hat{\theta}_{\text{Min}}|)/(\hat{\theta}_{\text{Max}} - \hat{\theta}_{\text{Min}})/2$. The modulation parameter, $b \geq 1$, allows a realistic estimation of the joint limit pointing when joint distance starts to represent a significant risk. The $R_J$ evaluation function is shown in (1).

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

2) Anisotropic Dexterity index, $\Theta$: Dexterity is defined as the ability of a robot to perform a movement given a specific configuration. Several dexterity indices for robotic manipulators have been proposed, [17]–[19]. One of the most used dexterity index, proposed in [20], is the manipulability index, $w = \sqrt{\det(J^T J)}$, where $J$ is the Jacobian matrix of a robot, $J^T$ its transposed and $det$ the determinant. Robot 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. Based on $w$, the directional manipulability index, $\psi_d$, expresses the manipulability of a robot moving along a specific direction, (2), where $q$ is the robot configuration, $u_d$ is the desired direction and $\dot{x}$ the robot velocity vector.

$$\psi_d = \frac{||\dot{x}||}{||q||} = \frac{1}{\sqrt{\dot{u}_d(J^T J)u_d}} \ (2)$$

The Anisotropic Dexterity index measures the directional manipulability of a robot executing a predicted trajectory. To overcome with this, $\Theta$ computes the estimated $\psi_d$ of all the trajectory points. Let $\{p_i, \Omega(p_{i+1}, d_{i+1}), ..., \Omega(p_{i+k}, d_{i+k})\}$
be a predicted trajectory and \( \{w_i, w_{i+1}, ..., w_{i+k}\} \) their associated manipulability ellipsoids. \( \Theta \) is the result of computing all the \( \psi_r \), starting from the current robot pose moving towards the first predicted pose, \( u_i = \{p_i, \Omega(p_{i+1}, d_{i+1})\} \), then, supposing the robot in the first predicted pose and moving towards the second one. ... The evaluation process finishes with the last pair of predicted points, \( u_{i+k} = \{\Omega(p_{i+k-1}, d_{i+k-1}), \Omega(p_{i+k}, d_{i+k})\} \).

3) Collision Risk index, CR: The robots which the VR is composed of share part 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.

In [21], 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 impact time of the robots when executing a task, as well as on the time required for a task transfer.

The estimated collision time of each robot, \( timp_{i R} \), is determined obtaining the minimum distance vectors evolution from each robot link to all the obstacles in its workspace, treating the rest of robots as obstacles. The CR index 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).

\[
CR = \frac{timp_{i R}}{\max(timp_{i_1}, ..., timp_{i_R})}, \forall r = 1..R
\]  

4) Task dependant index: tool orientation: This task dependant index measures the orientation of the surgical tool with respect to the abdominal wall normal where the fulcrum is placed, controlling possible excessive forces and torques produced by the tool orientation that can cause patient damages like hematomas. 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 is obtained using the quotient between the current angle and the maximum permitted angle, defined as a cone with its base represented as an ellipsoid. As in the RJ evaluation index, the distance is modulated using the same modulation function. Fig.4 illustrates two views of the tool orientation and the maximum orientation cone.

![Fig. 4. Current angle, α, and maximum orientation cone](image)

5) Robot Selection: The robot selection process has been designed with the aim of achieving a smooth execution of a teleoperated task, minimizing task perturbations, either in the workspace (trajectory perturbations) or in the time space (time delays produced by task transfers), minimizing the number of task transfers and facilitating as much as possible the task transfer process. Based on the robot suitability evaluation indexes, the need of a transfer is obtained determining the most adequate instant of time in which it should take place. This process is applied continuously during the execution of a task. Previous to the explanation of the proposed method, some concepts are listed:

- Robot Selected, \( R_{Sel} \): Robot that, in the evaluation time, is executing the task.
- Robots Candidates, \( R_{Can} \): Candidate robots to continue with the execution of the ongoing task.
- Continuity Robot, \( R_{CuT} \): Best robot, among \( R_{Can} \) to continue with the task.
- Current task point, \( CuTP \): Current point of the task used by the \( R_{Sel} \).
- Continuity task point, \( CnTP \): Set of points along the task where \( R_{Can} \) are evaluated.

The control algorithm is continuously evaluating and deciding the adequacy of a task transfer. At every evaluation instant of time the VR selects the \( R_{CuT} \) by estimating the robots suitability in the current and the next predicted points: \( R_{Sel} \) in \( CuTP \) and the rest of robots, \( R_{Can} \) in all the \( CnTP \).

6) Selection of \( R_{CuT} \): The \( R_{Can} \) selection is based on the suitability to continue with the execution of the ongoing task and on the easiness of the corresponding task transfer.

Since the evaluation indices are of different nature, and thus, not directly comparable, a common measure must be used to determine each robots suitability. All suitability evaluation indexes are transformed into the time space. With this transformation, the robots are evaluated by means of the difference between their remaining execution time, \( texec_{r,p} \), and the required task transfer time, \( ttrans_{r,p} \). The execution and task transfer times are computed for each robot as follows: \( R_{Sel} \) at \( CuTP \), and every \( R_{Can} \) at all the \( CnTPs \). \( texec_{r,p} \) is the estimated minimal amount of time along in which a robot is available to execute the ongoing task (remaining time before one of the indexes reaches its zero value). \( ttrans_{r,p} \) is the estimated time required to execute a task transfer between the current \( R_{Sel} \) and the evaluated robot. As task transfer involves a high risk of introducing perturbations in tasks execution, a complexity dependant safety margin is added to the estimated task transfer time, \( TTC_{r,p} \). Transfer complexity is measured in terms of the relation between the required joint accelerations during a task transfer process with respect to the maximum achievable ones. In (4), the calculus of the index for an \( n \) joint robot executing a \( K \) sampled task transfer trajectory is shown.

\[
TTC_{r,p} = \frac{\sum_{j=1}^{n} \left( \sum_{k=1}^{K} (\tilde{\theta}_{j,k}/\tilde{\theta}_{j,max}) \right)/K}{n}
\]  

TTC index varies from 0 to 1, representing \( TTC = 1 \) the worst case: maximum joint accelerations are required.

Finally, the \( ttrans_{r,p}^* \) reflecting the task transfer complexity safety margin is obtained, (5).

\[
ttrans_{r,p}^* = ttrans_{r,p}(1 + TTC_{r,p})
\]
Once \( t_{exec,r,p} \) and \( t_{trans f^∗r,p} \) are obtained, the final evaluation of the robots suitability is done in terms of useful time, \( tus_{r,p} \), which is the result of subtracting the necessary time to perform a task transfer to the remaining execution time. \( RCnt \) is the robot that presents the maximum of all the obtained \( tus_{r,p} \).

\[
RCnt = \max (tus_{RSel}, tus_{RCan_1}, \ldots, tus_{RCan_k})
\]

where \( tus_{r,p} = t_{exec,r,p} - t_{trans f^∗r,p} \)

(6)

7) Task Transfer Decision: If \( RCnt \neq R_{Sel} \), the convenience of starting a task transfer must be taken. To minimize the number of task transfers, even if \( tus_{RCnt,p} > tus_{R_{Sel},p} \), an immediate transfer is not necessarily required. The optimal instant to start the process is determined by the need of task transfer, the index, \( NTT \), which is computed as, (7).

\[
NTT = 1 - \frac{tus_{RSel} - t_{trans f^∗RCan}}{tus_{R_{Sel}}}
\]

(7)

\( NTT \) considers the relationship between the remaining \( tus \) and the required time to perform a \( t_{trans f}^∗ \). The closer the index is to 1, the highest is the need to begin a task transfer.

IV. EXPERIMENTAL RESULTS

In order to test and validate the proposed VR, as well as detect its drawbacks, a set of experiments have been carried out. Two set-ups have been implemented: The first one is composed of two simulated 6DoF robots equipped with a laparoscopic tool as end effector, whereas the second one uses real Staübi Rx60B robots.

A. Experimental Set-Up

The VR control runs under a Windows XP platform in an Intel Core2 Duo. The communication between the VR and the robot controllers (real or simulated) is done using a private Ethernet network, at a frequency of 55.6 Hz (establish by the robot controllers). The control strategy has been developed following the oriented object paradigm and implemented in C++ and the OpenGL graphic library. The human operator uses a Phantom Omni haptic device, which provides 6DoF movements and generates 3DoF of force feedback, as master. The movement orders are directly mapped on the manipulated obstacle: Translation onto a fixed frame and rotations according to the objects reference frame.

In Fig. 5 a snapshot of the experimental task developed in the following subsection is shown. The experiment, consisting on the telemanipulation of an object inside the abdominal region negotiating an obstacle, is executed by the set-up with real robots (left part of the image). The user obtains the endoscopic view, apart from other useful information, on the monitor present at the right of the image.

B. Experimental results

In this subsection an experiment consisting on a VR with two simulated robots in a virtual abdomen, Fig.6 (Top,Left) is analysed. The user telemanipulates an object inside the umbilical region, guiding the object to cross under an elevated piece of the small intestine, which position is known acts as an obstacle to be avoided, as shown in Fig.6 (Top,Right). The virtual endoscopic view is shown in Fig. 6 (bottom).

For these experiments, \( t_{trans f} \) is determined by the necessary time to perform a straight movement from the current robot pose to its \( CnTP \), executing this path at half speed.

As can be seen in Fig. 8.a, where \( R_{Sel} \) is shown, a task transfer is necessary during the task execution (\( t \approx 9s \)). In Fig.7.a \( RJ \) shows the evolution of the joints during the task. The values of both indexes show that no risk of reaching a limit is detected. In Fig.7.b \( \Theta \) shows how \( R0 \) is moving towards a singularity (produced by the alignment of two links). On the contrary, the value of \( R1 \) indicates that this robot evolves from a poor dexterity region to a better one. In Fig.7.c the cause of the task transfer is shown: \( R0 \) reaches its maximum permitted orientation, becoming a non-valid candidate to continue with the task execution (\( t \approx 9s \)). \( R1 \) 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.8.a shows \( t_{trans f} \) of both robots. During the first part of the task (\( Ro = R_{Sel} \)), the \( t_{trans f_{R1}} \) low value indicates a non complex task transfer (\( R1 \) is close to its \( CnTP \)). From
Fig. 7. a) RJ, b) Θ and c) Tool Orientation indexes evolution during the task execution

t > 10s \ t_{\text{transf}} \ f_R^0 \text{ increases due to the impossibility of this robot to be close to its } CnTP.

The \ NTT \ evolution is shown in Fig.8.b. 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 \approx 5s \text{ and } t \approx 8s, \text{ fluctuations are produced by changes on the user movements velocity. Between } t \approx 8s \text{ and } t \approx 9s \text{ NTT increases its value until } NTT \rightarrow 1, \text{ indicating the need of an immediate task transfer.}

Fig. 8. a) \ t_{\text{transf}}^*, b) NTT \text{ and } c) RSc, \text{ Orientation indexes evolution during the task execution}

V. CONCLUSIONS

A new paradigm for teleoperation focused on surgical tasks using multiple robots on RIMS systems has been presented. Since RIMS usually operates with multiple robots, the VR does not represent a costly solution. Different evaluation indices have been presented that allow determining the suitability of each robot to continue with the execution of a going-on task. The open control architecture of the VR allows the introduction of new evaluation criteria depending on the task requirements. The analysis of a task has been exposed in order to clarify the concepts presented in the paper and demonstrates its usefulness. An interesting aspect of this proposal is that the use of the VR paradigm does not discard the use of traditional bimanual teleoperation. The surgeon can switch between both methods depending on the task to be performed.

VI. FUTURE WORK

Immediate work will focus on test the VR in different surgical tasks performed by experienced surgeons, introduce a third arm into the real set-up and the use of virtual fixtures as a complementary assistance to improve the surgeon dexterity. Mid-term work will be focused on organs registration to be able to define more accurate VF protections.

REFERENCES