Proceedings of the ICAR 2009
14th International Conference on Advanced Robotics
Munich, Germany, June 22 - 26, 2009

Optimization criterion for safety task transfer in cooperative robotics

A. Hernansanz, J. Amat, and A. Casals.

Abstract— **This paper presents a strategy for a cooperative multirobot system, constituting a Virtual Robot. The Virtual Robot is composed of a set of robotic arms acting as only one, transferring the execution of a teleoperated task from one to another when necessary. To decide which of the robots is the most suitable to execute the task at every instant, a multiparametric decision function has been defined. This function is based on a set of intrinsic and extrinsic evaluation indexes of the robot. Since the internal operation of the Virtual Robot must be transparent to the user, a control architecture has been developed.**

I. Introduction

With the advances in robotics, robots are expected to operate in more and more complex environments solving new kind of applications. These increasing demands that require more dexterity and accessibility in larger and more complex working areas may imply that a single robot is not enough to perform the desired task.

Multirobot systems are usually designed to carry out manipulation tasks that a single robot cannot do. Typical examples of these tasks are the manipulation of large, heavy or even flexible objects [1]. In recent years, multirobot systems have also been applied in bimanual manipulation tasks, [2], [3]. The solution proposed in this paper, a Virtual Robot, is not focused on this kind of tasks, but to increase the accessibility, maneuverability or the size of the workspace of a single robot in teleoperated tasks.

To increase robot maneuverability and operating workspace, a solution is the use of redundant robots. Redundancy allows a robot to reach a pose in the workspace in multiple configurations, avoid singularities and collisions and execute trajectories minimizing some parameters, like joint torques. However, depending on the task to be performed, redundant robots could be ineffective.

An alternative solution consists in designing a Virtual Robot, constituted by a combination of a given number of robots acting as only one, transferring the execution of the desired task from one to another when necessary. One such Virtual Robot is expected to act as if there was only one physical robot, operating in such a way that its internal cooperative working mode is transparent to the user. To accomplish this goal, the Virtual Robot control architecture has to be carefully designed in order that an efficient strategy can decide which real robot is operative at every successive step during the execution of a task.

An interesting approach to increase the range of applicability of a single robot when manipulating an object was developed in [4]. In this work, the automatic generation of a motion plan in a cooperative robotic system was proposed. Once the initial and goal configurations of the task are determined, the effective motion plan of the robots involved in the execution of the trajectory is automatically planed. This motion plan includes the transfer of the manipulated object from one arm to another, thus, increasing the workspace and the applicability that a single robot can offer.

The main difference between the Virtual Robot proposed in this paper with respect to the previous approach, is the level of knowledge about the task to be performed. Since the proposed Virtual Robot is conceived for telemanipulation tasks, the robot trajectory has to be defined on-line, so no *a priori* motion planning is possible. To illustrate one such application, let's consider the need to remotely manipulate an object, for its inspection, in front of a camera within a complex and dynamic environment. The visualization of the object from all the desired points of view may require large accessibility and maneuvering. This task could not be accomplished with only one robot due to diverse reasons: restrictions of the own robot (workspace, singularities and joint limits), possible obstacles on the workspace, as well as occlusions of the object produced by the robotic arm, what might even require picking the object using different grasping positions. This regrasping process implies putting the object on a specific place in order to regrasp it when using a single robot, redundant or not. One such operation can significantly alter the desired trajectory, disturbance that can be avoided using a Virtual Robot. Fig. 1 shows a Virtual Robot composed of two robots executing one such remote manipulation. In Fig. 1a the robot that holds the object constitutes an occlusion within the camera field of view, whereas in Fig. 1b the occlusion is avoided.

Another illustrative example could consist in tasks like wrapping a large pipe with a tape. Fig. 2 shows the trajectory required in one such task, where the pipe

Albert Hernansanz is with the Research Group on Intelligent Robotics and Systems, Technical University of Catalonia, Barcelona, 08028 Spain e-mail: albert.hernansanz@upc.edu.

Dr. Josep Amat is with the Institute of Robotics and Industrial Informatics (Technical University of Catalonia-CSIC) LLorens i Artigas 4-6 2nd Floor Technology Park of Barcelona 08028 Barcelona, Spain.

Dra. Alícia Casals is with the Institute for Bioengineering of Catalonia. Technical University of Catalonia, Barcelona, 08028 Spain e-mail: alicia.casals@upc.edu.

represents an insuperable obstacle for a unique robot, even if it is redundant.

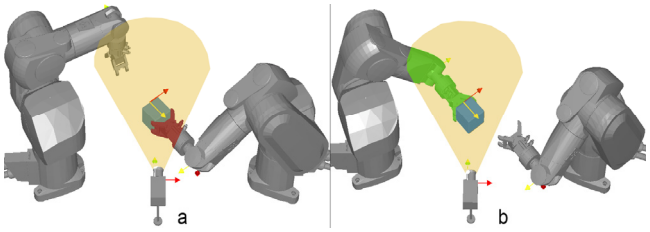

Fig. 1 Example of visual inspection of an object in front of a camera

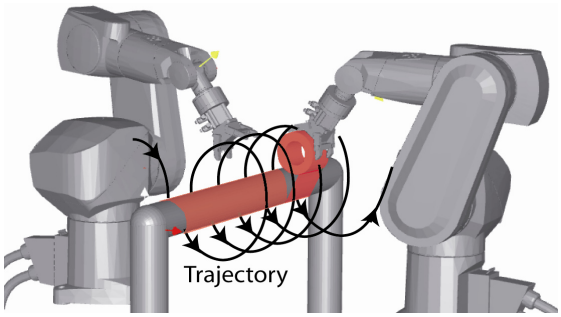

Fig. 2 Example of a trajectory around a pipe.

The paper is structured as follows: first, the concept and the control architecture of the Virtual Robot are described. In Section III the robot evaluation indexes are explained. The decision function which selects the best candidate to execute the trajectory at every moment is explained in Section IV. Experimental results are presented in Section V. The paper finishes with some conclusions.

At this stage of research, the actions related to task transfer, which mostly refer to grasping and regrasping the object, are not yet tackled, although the time required for this operation has been taken into account. It is also assumed that the geometrical model and the position of the obstacles in the workspace are known.

II. Virtual Robot

The Virtual Robot has been conceived with the aim of increasing the range of applicability of a single robot, especially in telemanipulation tasks. The cooperative behavior is based on transferring the execution of a manipulation task from one robot to another before the robot that is executing the task approaches a configuration that prevents its continuation. Each transfer has to be predicted well in advance, in order to avoid critical situations and to have enough time to react and execute the required task transfer, altering as less as possible the desired task trajectory. The internal operation of the Virtual Robot must be automatic and transparent to the user. Therefore, the user efforts can be focused on the task. Coming back to the example of a remote manipulation, the user should concentrate his efforts on the visual inspection of the object, instead of paying attention to other aspects, like which robot holds the object or if any occlusion or collision can occur. With this aim, a control architecture has been developed based on a robotic simulation library: Robotic Proximity Queries (RPQ), [5]. This library, a previous step of this research, provides a set of features which are useful for the Virtual Robot purposes:

1) A workspace simulator, which includes the different elements: robotic arms, objects and other obstacles.
2) A proximity query package that offers three different types of queries: collision detection, distance computation and tolerance verification.

The Virtual Robot control architecture is divided into three different phases: acquisition of the input data, selection of the robot that executes each task segment and finally, planning and execution control.

Three kinds of inputs are collected: user teleoperation orders (desired trajectory), position of the obstacles in the workspace and configuration of the robots.

In the second phase, the individual evaluation of the robots is done using intrinsic (joint limits and directional dexterity index) and extrinsic (collision risk) evaluators. A decision function uses these individual evaluators to select the most suitable robot to execute the desired trajectory.

In the third phase, the actions to be performed by each robot are planned and executed. The inactive robots must adapt their configuration in a way that they do not disturb the active robot trajectory, but at the same time they should be ready to facilitate a possible future task transfer, which is one of the most problematic issues of the Virtual Robot. Fig. 3 shows the block diagram of the Virtual Robot control architecture.

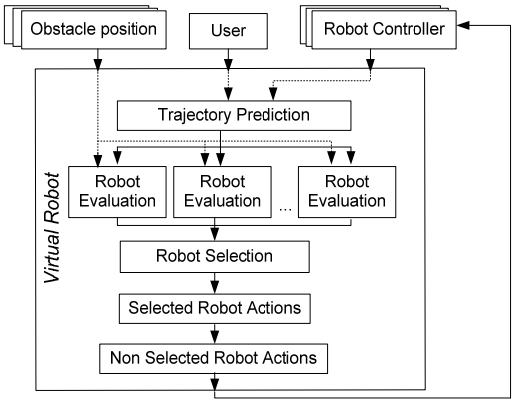

Fig. 3 Diagram of the control architecture of the Virtual Robot

III. Robots Suitability Evaluation

The robots suitability evaluation has to provide a complete estimation of the adequacy of each robot in the execution of the desired trajectory. In order to be able to react on-time, as required in teleoperation, a trajectory predictor is necessary. Based on this prediction, three different evaluation criteria have been considered taking into account the different causes that can prevent a robot from continuing the execution of a trajectory: joint limits, dexterity, and collision risk.

A. Trajectory Prediction

The need of a trajectory predictor arises from the need to act in a predictive manner in teleoperation tasks, as previously explained. The Virtual Robot does not require a specific trajectory predictor and, depending on the application and its specifications, different predictor methods can be applied.

Each point of the predicted trajectory, p_i, is considered as a probability distribution function in the workspace. Its estimated probability distribution is based on a probabilistic distribution function, $\Omega(p_i,d_i)$, with mean $p_i=\{p_{i,1},...,p_{i,m}\}$ and standard deviation d_i, where $d_i=\{d_{i,1},...,d_{i,m}\}$, being m the workspace dimension.

In order to decide which type of trajectory predictor accomplishes with the requirements, different manipulation tasks were executed and recorded in order to analyze the dynamics of teleoperated tasks. Finally, a polynomial trajectory predictor was selected due to its performance, robustness and its low computational costs. Positions and orientations are predicted separately.

Positions prediction is done using a second order polynomial, in order to take velocity and acceleration variations into account. The coefficients of the polynomial are calculated using the least squares fitting technique. The sampling rate is constant whereas the number of sample points is variable and depends on the acceleration observed on the last sampled points. The use of a variable number of points allows modifying the inertia of the prediction polynomial: the more points are used to fit the polynomial, the more inertia it has. Following this principle, the number of points is inversely proportional to the mean value of the acceleration of the last part of the executed trajectory.

To represent the orientations of each trajectory point, quaternions are used. Quaternions provide an efficient and concise representation for rotations and do not suffer from singularities like Euler angles or axis angles do, [6]. In order to predict orientation, the spherical linear interpolation technique (SLERP) is used. In this case, a first order polynomial has been selected because only low variations on the angular velocity are expected. For large angular accelerations, higher order polynomials can be used, as in [7], where Lagrangian cumulative polynomial trajectories are applied.

B. Joint limits

This first evaluation criterion measures the distance between the joints positions and their respective limits to avoid reaching them. Using the minimum distance between a joint position, θ, and its limits, $(\theta_{min}, \theta_{max})$, a continuous and smooth modulation function has been used. The modulated function J_{eval} is computed from the normalized minimum distance function, $dist(\theta)$, as follows, (1).

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

This modulation function penalizes a joint evaluation index only when it is near to its limits. Fig. 4 shows the $dist(\theta)$ and J_{eval} functions and the effect of modifying the parameter b, $(b\geq 1)$, for a rotational joint.

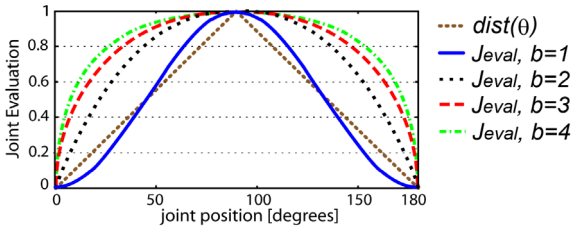

Fig. 4 Modulation of the distance function ($dist(\theta)$) using different values of the parameter b.

More efficient than obtaining the J_{eval} index for the current pose of the robot is extending this evaluation to the predicted points. The joint configurations of the predicted points are also expressed as a probability distribution: $\Omega'(\theta_i,d'_i)$. Each θ_i is obtained applying the inverse kinematics to the predicted points. The standard deviations, d'_i, are not calculated using the same procedure due to its high computational cost. To obtain them, the mean of the observed errors of the previous predictions is used. The valid joint ranges are modified reflecting the uncertainness of the predicted points: $\theta'_{max,i}=\theta'_{max}- d'_i$ and $\theta'_{min,i}= \theta'_{min}- d'_i$.

Once the joint limits evaluation of the current joint configuration ($t=0$) and the next k predicted points ($t=1..k$) are obtained, the weighted average, $\overline{J}_{eval}$, of all J_{eval} is calculated, (2)

$$\overline{J}_{eval} = \sum_{t=0}^{k} w_t J_{eval,t} \Big/ \sum_{t=0}^{k} w_t \ ,\ w_t = 1/t+1 \qquad (2)$$

Finally, the combined evaluation of all the joints, RJ, of each robot is computed from the product of all $\overline{J}_{eval}$. Fig. 5 shows the $\overline{J}_{eval}$ and the RJ of the robot used in the experimental part in the execution of a polynomial trajectory. The joint limits, in radians, are shown in the following table.

[rad]	Joint1	Joint2	Joint3	Joint4	Joint5	Joint6
Limit 1	-2.8	-2.2	-0.8	-3.1	-1.9	-3.1
Limit 2	2.8	2.2	3.9	3.1	2.1	3.1

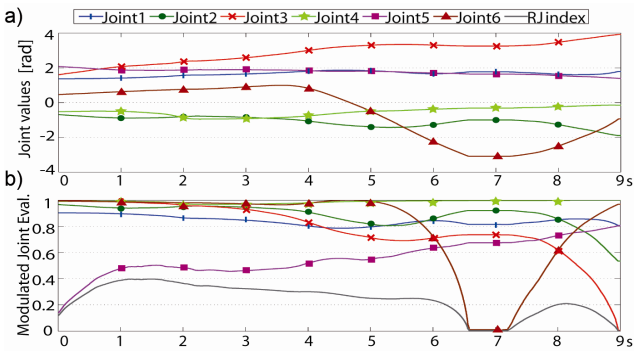

Fig. 5 a): Joint positions of a 6 rotational dof robot following a polynomial trajectory. b): The J_{eval} of each joint and RJ index.

C. *Anisotropic Dexterity Index*

Dexterity is defined as the ability of a robot to perform a movement given a concrete configuration. A wide number of dexterity indices for robotic manipulators have been proposed in the literature, [8], [9]. One of the most used dexterity measures is the manipulability index, proposed in [10]. This index is defined as the square root of the determinant of the product between the Jacobian matrix of a robot, J, and its transposed, J', $w = \sqrt{\det(JJ')}$.

This measure indicates the ability of a robot to move along certain directions. When the robot reaches a singularity, the Jacobian loses rank, and then the manipulability index becomes zero.

Robot manipulability can be expressed as an ellipsoid in the manipulator work space. The volume of the ellipsoid, which is proportional to w, denotes the ability of a manipulator to perform a movement. To compute the manipulability ellipsoid, the singular values of the Jacobian matrix are used, (3).

$$w = \sqrt{\lambda_1 \lambda_2 ... \lambda_m} = \sigma_1 \sigma_2 ... \sigma_m \qquad (3)$$

Where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_m$ are the eigenvalues of J and $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_m$ are the singular values obtained using the singular value decomposition technique: $J = U\Sigma V'$.

The principal axes of the manipulability ellipsoid are $\sigma_i \cdot u_i$, with $i=1..m$ and u_i the column vectors of U.

Given a robot configuration, q, and a desired movement direction in the workspace, expressed as a unitary vector u_d, a directional manipulability measure, ψ_u, is defined, (4).

$$\psi_u = \|\dot{x}\| / \|\dot{q}\| = 1 / \sqrt{u'(JJ')u} \qquad (4)$$

Geometrically, ψ_u, is the distance from the center of the ellipsoid to the point where the line with direction u intersects the ellipsoid.

The dexterity indexes explained above evaluate how dexterous a robot is to move along any (3) or a concrete direction (4), given a robot configuration. These measures can also be used to evaluate the dexterity of a robot during the execution of a task. Fig. 6 shows the evolution of the manipulability ellipsoid for the x,y,z components during the execution of a linear trajectory. The closer the robot is to a singularity, the smaller becomes the ellipsoid.

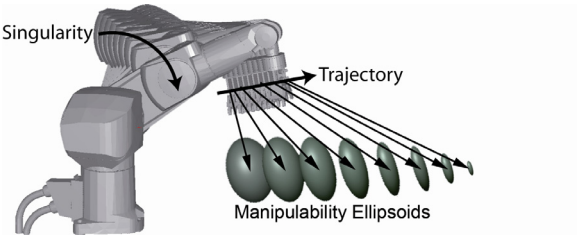

Fig. 6 Robot executing a linear trajectory and the evolution of the manipulability ellipsoid.

As in the joint limits index, the measure of directional manipulability is extended to evaluate the predicted trajectory. This implies the need of dealing with the uncertainness of the predicted points. Let's assume that the current position of the robot is p_i and the next predicted point is represented by $\Omega(p_{i+1}, d_{i+1})$ then, the modified manipulability index, $\phi_{i,i+1}$, represents all the directional manipulability indexes, ψ_u, with origin in p_i and intersection with the probability distribution of p_{i+1}. Each ψ_u is weighted using its probability of occurrence, $P(u)$, (5).

$$\phi_{i,i+1} = \int \psi_u P(u)\, du \qquad (5)$$

The physical analogy of the $\phi_{i,i+1}$ index is the mass generated by all the ψ_u inside the manipulability ellipsoid of p_i. The density of this mass is determined by the probability of occurrence of ψ_u, $P(u)$. This probability of occurrence can be interpreted as the line segment inside the distribution probability of p_{i+1}. Fig. 7 illustrates the geometrical meaning of the calculus of the index $\phi_{i,i+1}$ in a 3D space.

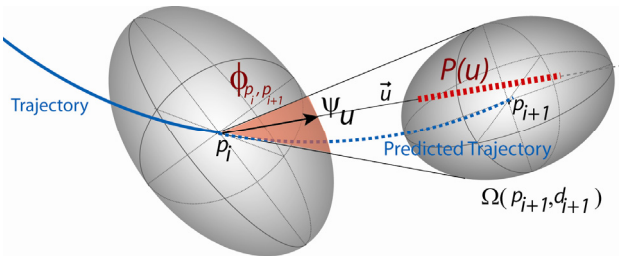

Fig. 7 A directional manipulability index ψ_u and its probability of success, $P(u)$ in the probability distribution, $\Omega(p_{i+1},d_{i+1})$. The $\phi_{p_i,p_{i+1}}$ from p_i to p_{i+1} is also shown.

The calculus of $\phi_{i,i+1}$ represents a hard computational problem. To deal with real time requirements, the probability distribution is sampled with N samples, (6).

$$\overline{\phi}_{i,i+1} = \sum_{k=1}^{N} \left(\phi_{(i,i+1),k} P(p_k) \right) \Big/ \left(N \sum_{k=1}^{N} P(p_k) \right) \qquad (6)$$

The probability distributions are uniformly sampled, in order to ensure the best space coverage. In [11] a fast and accurate method for uniform sampling of quaternions is proposed. The number of samples is selected according to the real time requirements. The inexistence of a singularity within the probability distribution function can be assured by demanding a minimum value for the minimum singular value, σ_{min}, of each sample. The minimum singular value is a good indicator of how far a robot is from a singularity. The closer is the singularity, the smaller is σ_{min}.

Up to now, p_i has been considered as a single point. When considering p_i also as a probability distribution function, what means that p_i is a predicted point, it is necessary to find the mean of all the directional manipulability indexes $\psi_{i,ui}$ within its distribution $\Omega(p_i, d_i)$, (7).

$$\Phi_{i,i+1} = \int_{\Omega} \phi_{i,i+1} P(p_i) d\Omega \quad (7)$$

Where $p_i \in \Omega$, is the origin point and Ω is the probability distribution around the predicted point.

Again, the probability distribution associated to the origin point, p_i, is sampled using the same method as the one used for the destination point, p_{i+1}, (8).

$$\overline{\Phi}_{i,i+1} = \sum_{k=1}^{M} \left(\Phi_{(i,i+1),k} P(p_k) \right) \Big/ \left(N \sum_{k=1}^{N} P(p_k) \right) \quad (8)$$

Finally, the weighted product of all the $\overline{\Phi}$ from the actual robot pose, p_i, to the last predicted pose, p_{i+k}, is done in order to obtain the predictive dexterity evaluation of each robot. The weights associated to each $\overline{\Phi}_{i,i+1}$ are proportional to the deviation of their probability distribution function.

D. Collision Risk

An effective collision avoidance system has to detect and react in an anticipative manner. Classical approaches use distance as a unique parameter to determine the risk of impact between two objects. In [12], a new approach defines the degree of collision danger based on the relative velocity between two objects and their maximum acceleration, which determines their reaction capability. In this Virtual Robot approach, the collision risk is used not only to prevent collisions, but also to determine the suitability of each robot to perform the task.

The estimated collision time of each robot, $t'_{imp,Ri}$, is determined obtaining the minimum distance vectors from each robot link to all the obstacles. These distances are obtained both in the current configuration of the robot as if it was executing the task, d_{act}, as well as in the configuration of the robot in the next predicted point of the task, d_{next}. Using these distances, the approximation velocities for every link are obtained, v_{appro}. Then, all the estimated impact times can be calculated as follows: $t'_{imp,Ri,Lj} = dist_{act} / v_{appro}$. Finally, the impact time is fixed using the minimum of all the $t'_{imp,Ri,Lj}$.

Before the evaluation of the collision risk of the robots is done, the availability of the robots to perform the task transfer is determined, (9). A robot is able to execute the task only if the necessary time to perform the task transfer, $t_{transf,Ri}$, is less than the minimum expected collision time of robot executing the task, $t'_{imp,RSel}$. At this stage of research, the actions related to task transfer are not automatically planned, so the necessary time is manually determined.

$$Av_{R_i} = 1 - \left(t_{transf,R_i} / t'_{imp,R_{Sel}} \right) \quad (9)$$

Those robots verifying that $Av_{Ri} < 0$ are discarded as possible candidates to execute the task.

Once the available robots are obtained, the collision risk is determined by the quotient between the estimated impact time of each robot and the maximum estimated impact time of all the robots, (10).

$$CR_i = t'_{imp,R_i} \Big/ \max(t'_{imp}) \quad (10)$$

E. Robot Evaluation Index

Finally, the different evaluation indexes: Joint Limits: *RJ*, Anisotropic Dexterity: $\overline{\Phi}$ and Collision Risk: *CR*, are used to obtain a complete evaluation of the expected robot performance. All the indexes are normalized between 0 and 1. The proposed robot evaluation index, *Reval*, is the result of the product of all the individual evaluation indexes, (11). This function ensures that if one of the individual indexes is zero, the corresponding robot will be discarded.

$$Reval_i = RJ_i \cdot \overline{\Phi}_i \cdot CR_i \quad (11)$$

IV. Robot Selection

Once the individual evaluation of each robot is obtained, a decision function, R_{sel}, must select the best candidate to execute the manipulation task. This selection, is not done based uniquely on the current evaluation of each robot, but also on the evolution of the last evaluated points, (12). These two factors are weighted respectively with the weights *A* and *B*, that verify $(A+B=1,\ A \geq 0,\ B \geq 0)$, and depend on the dynamics of the trajectory. If high accelerations are expected, *A* decreases and *B* increases.

$$Q_i = k_i (A \cdot Reval_i + B \cdot (dReval_i / dt)) \ , \forall i = 1..r \quad (12)$$

When two or more robots have a similar *Q* in consecutive evaluations, a continuous task transfer might happen. This situation can occur, for instance, when the Virtual Robot trajectory passes near the boundary of the workspace of two robots. To minimize as much as possible the number of task transfers, a hysteresis parameter, $k \geq 1$, has been added to the *Q* index of the robot that is executing the trajectory. For the remaining robots, the value of the hysteresis parameter is equal to one. The selected robot, R_{sel}, is then determined by the maximum value of all the Q_i indexes, (13).

$$R_{sel} = \max(Q_i) \ , \forall i = 1..r \quad (13)$$

V. Experimental Results

The test bed for experimentation of the Virtual Robot consists of two 6 dof robotic arms (Staubli RX60) equipped with pneumatic grippers and controlled from a PC. The user executes the desired trajectory using a 6 dof mouse. To determine the correct robot selection frequency, the joint values and the trajectory described by the robot end-effector during the execution of several manipulation tasks have been recorded. The spectral analysis of the recorded data has determined that the highest frequency to be taken into account is 1'3Hz. Following the Nyquist-Shannon sampling criterion, 5 samples are taken every 1'3Hz, so the frequency

required to select a robot is 6.5Hz. In order to synchronize the robot selection with the robot controllers, the sample time is finally fixed at 144ms (6.9Hz), conditioned by the robot communications period (18ms).

The manipulated object is a cube with a grasp point defined in each face. While the evaluation of the robot that is executing the trajectory is done based on the current grasping point, those of the other robots are evaluated considering all the free grasping positions.

Fig. 8, shows three shots of the sequence of movements corresponding to the experimental test of visual inspection of a cube in front of a camera. Initially, Fig. 8a, R1 holds the object while R2 is approaching the best evaluated grasp point. In Fig. 8b, the task is being transferred before R1 produces an occlusion. Finally, in Fig. 8c, R1 continues the execution of the task while R2 prepares itself to reach the best suitable configuration to retake the task.

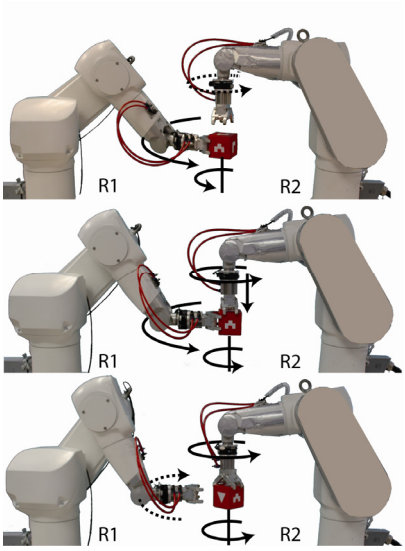

Fig. 8. Visual inspection of a cube in front of a camera.

A second experiment consists on moving an object following a long trajectory. Fig. 9 shows a simulated sequence of the trajectory that requires a transfer due to the existence of a singularity in R1.

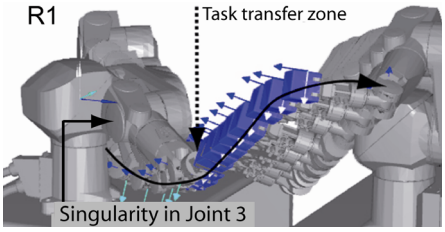

Fig. 9 Task transfer (R1 to R2) in order to avoid a singularity in R1.

VI. CONCLUSION

In this paper, a strategy for multirobot cooperation in teleoperation tasks has been presented. This strategy constitutes the base of a Virtual Robot. This Virtual Robot, composed of a set of robots acting as only one, increases the accessibility, maneuverability or workspace achievable with a single robot.

In order to evaluate the suitability of the different robots to execute a manipulation task, three different and complementary criteria have been selected, generating three different evaluation indexes: Joint Limits: *RJ*, Anisotropic Dexterity: $\overline{\Phi}$ and Collision Risk: *CR*. These indexes evaluate the robots not only based on their current configurations, but also on the next predicted ones. The combined use of these indexes provides a complete evaluation of the suitability of each robot.

Given the individual evaluation of the robots, a decision function has been implemented to decide which robot is the most adequate to continue with the execution of a task. This function also minimizes the number of task transfers applying a hysteresis parameter. In the future, the automatic tuning of this parameter will be tackled.

To illustrate the usefulness of the Virtual Robot, two examples have been presented. In these examples the Virtual Robot executes manipulation tasks that a single robot is not able to perform.

Further research should be focused on the strategy to automatically decide the configuration of the robots that do not execute the task. This automation will minimize the effects of a task transfer along the user desired trajectory.

REFERENCES

[1] M. Saha and P. Isto, "Motion Planning for robotic manipulation of deformable linear objects," in Proc of the IEEE Int. Conf on Robotics and Automation, 2006, pp. 2478-2484.

[2] A. Edsinger and C. Kemp, "Two Arms Better than One: A Behavior Based Control System for Assistive Bimanual Manipulation," in Proc of Int. Conf. on Advanced Robotics, 2007

[3] Myun Joong Hwang, Doo Yong Lee, and Seong Youb Chung, "Motion Planning of bimanual robot for assembly," in IEEE Int. Conf. on Systems, Man and Cybernetics, 2007, pp. 240-245.

[4] Y. Koga and J. C. Latombe, "On multi-arm manipulation planning," in IEEE Proc. of the Int. Conf. on Robotics and Automation, 1994, pp. 945-952.

[5] A. Hernansanz, X. Giralt, A. Rodriguez and J. Amat, "RPQ: Robotic Proximity Queries - Development and Applications", in Int. Conf. on Informatics in Control, Automation and Robotics, 2007.

[6] J. Funda and R.P. Paul, "A Comparison of Transforms and Quaternions in Robotics," in Proc. of the IEEE Int. Conf. on Robotics and Automation, 1988.

[7] Yim-Pan Chui and Pheng-Ann Heng, "Attitude dead reckoning in a collaborative virtual environment using cumulative polynomial extrapolation of quaternions" in Concurrency and Computation: Practice and Experience, vol 16, issue 15, Ed John Wiley & Sons, 2004, pp. 1575-1599.

[8] C. A. Klein and B. E. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulator," in Int. Journal of Robotic Research, vol 6, issue 2, 1987, pp. 72-83.

[9] L. Zlajpah, "Dexterity Measures for Optimal Path Control of Redundant Manipulators," 1996.

[10] T. Yoshikawa, "Manipulability of Robotic Mechanisms," in Int. Journal of Robotic Research, vol 4, issue 2, 1985, pp. 3-9.

[11] James J. Kuffner, "Effective Sampling and Distance Metrics for 3D Rigid Body Path Planning" in Proc. of the IEEE Int. Conf. on Robotics and Automation, vol. 4, 2004, pp. 3993-3998.

[12] E. Freund, J. Rossmann, "The basic Ideas of a proven Dynamic Avoidance Approach for Multi-Robot Manipulator Systems," in Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, vol. 2, 2003, pp. 1173-1177.