Correspondence _____

Failure Tolerant Teleoperation of a Kinematically Redundant Manipulator: An Experimental Study

M. Goel, A. A. Maciejewski, V. Balakrishnan, and R. W. Proctor

Abstract—Teleoperated robots in harsh environments have a significant likelihood of failures. It has been shown in previous work that a common type of failure such as that of a joint "locking up," when unidentified by the robot controller, can cause considerable performance degradation in the local behavior of the manipulator even for simple point-to-point motion tasks. The effects of a failure become more critical for a system with a human in the loop, where unpredictable behavior of the robotic arm can completely disorient the operator. In this experimental study involving teleoperation of a graphically simulated kinematically redundant manipulator, two control schemes, the pseudoinverse and a proposed failure-tolerant inverse, were randomly presented under both nonfailure and failure scenarios to a group of operators. Based on performance measures derived from the recorded trajectory data and operator ratings of task difficulty, it is seen that the failure-tolerant inverse kinematic control scheme improved the performance of the human/robot system.

Index Terms—Fault/failure tolerance, kinematically redundant, kinematics, locked joint failure, manipulators, redundant robots/manipulators, teleoperation.

I. INTRODUCTION

Operations in hostile and/or remote environments are often performed by robots under human control, as in teleoperation. In such applications, however, the harsh nature of the environment significantly increases the likelihood of robotic failures [1]. While there are several ways in which a robot may fail [2]–[6], one common failure mode is a "locked joint," where the affected joint's velocity is identically zero. While such failures can be appropriately handled if successfully identified [7]–[11], they can cause significant performance degradation if they remain unidentified [12], [13]. Such a scenario is critical in teleoperated systems because the operator may become disoriented by the erroneous motion of the arm prior to failure identification and activation of a recovery scheme [14], [15]. Even small time delays in teleoperated systems have been know to destabilize the human-robot control loop [16], [17]; with failures, the effects could range from a sluggish response of the arm to completely unpredictable motion, which is a much more serious problem. The post-failure performance must therefore be addressed in the overall system design.

While it is easy to envision a failure recovery strategy for identified failures, the situation is much more difficult when a failure is merely

detected, but not identified. In particular, consider a joint malfunction where a significant deviation is detected between the commanded joint velocities and those returned by the joint sensors. Identifying whether this malfunction is due to a joint locking up, or whether the sensor is (erroneously) returning a constant joint position, is not possible.[1] A simple response to such a situation is to assume implicitly that a joint has locked, and to adjust the control law accordingly so as not to command motion of the joint in question [18]. However, given that sensor failures are more likely than joint actuation failures [19], this may not be the best course of action, as the workspace is unnecessarily reduced (often quite severely [20]–[22]).

We therefore consider the scenario where the control law remains unmodified, and explore the consequences. For the case when the sensor has indeed failed, the ability of manipulators to converge to desired end-effector positions even with imperfect/approximated Jacobians/Jacobian-inverses has been addressed in [23], [24]. Therefore, we focus on the other case where a joint has actually locked, but the controller continues to command motion of that joint as though it were healthy. The study is restricted to purely kinematic effects, due to the fact that the dynamic effects of a failure are essentially transient in nature, and do not significantly affect the steady-state behavior being evaluated here [11], [25], [26].

Three different control schemes tolerant to unidentified failures in kinematically redundant manipulators were proposed in [27], [28]. The worst-case performance of these control strategies, designed with the explicit goal of improving post-failure performance, is seen to be substantially better than that of the pseudoinverse control scheme that was used as a baseline. While the proposed schemes yielded promising results for tasks performed under computer control, the question of how they actually perform with a human in the loop, as in teleoperation, is addressed in this experimental study. A brief mathematical framework for the study, specifics of the experiment, and the results are detailed in the following sections.

II. MATHEMATICAL FRAMEWORK

The position and/or orientation (henceforth referred to as "position") of the end effector of a manipulator can be expressed in terms of its joint variables by

$$\mathbf{x} = f(\mathbf{q}) \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^m$ is the position of the end effector, $\mathbf{q} \in \mathbb{R}^n$ is the vector of joint variables, and m and n the dimensions of the task space and joint space respectively. Manipulators that have more degrees of freedom (DOFs) than required for a task, i.e., $n > m$, are said to be redundant. The end-effector velocity is expressed in terms of the joint rates as

$$\dot{\mathbf{x}} = J\dot{\mathbf{q}} \tag{2}$$

where $J \in \mathbb{R}^{m \times n}$ is the manipulator Jacobian, $\dot{\mathbf{x}}$ is the end-effector velocity, and $\dot{\mathbf{q}}$ is the joint velocity.

If perfect servo control of the joints is assumed, then in a healthy manipulator the actual joint velocity $\dot{\mathbf{q}}_a$ equals the commanded velocity $\dot{\mathbf{q}}_c$. However, in the event of a locked-joint failure of the i-th joint,

[1]This assumes a typical commercial robot where duplicate sensing and/or analytical redundancy does not exist.
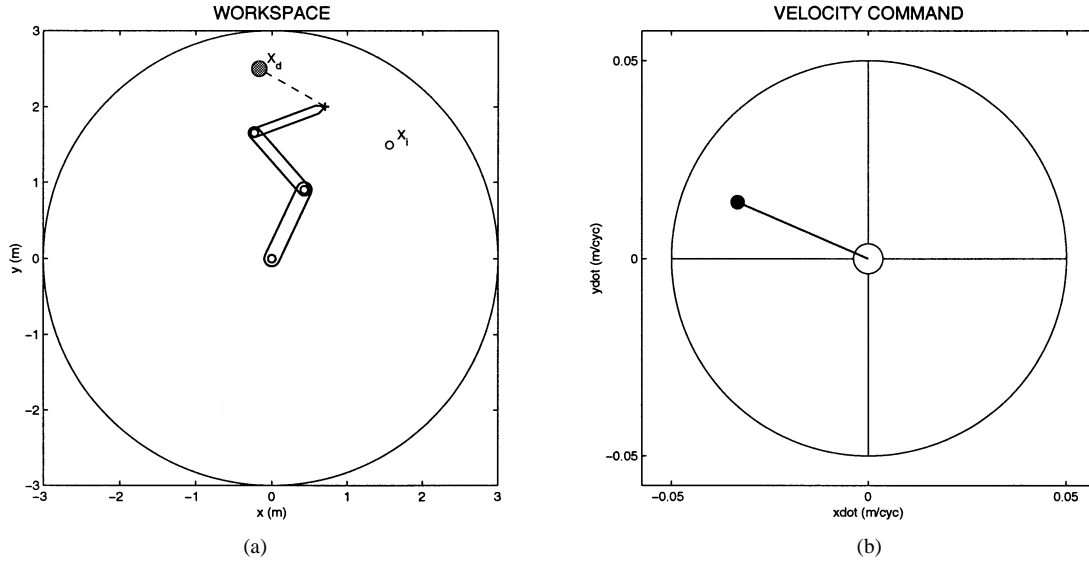
Fig. 1.   Graphical user interface presented to the human subject teleoperating the robot under different failure scenarios. (a) Graphical simulation of the robot and (b) the operator's joystick command.

the corresponding element of $\dot{\mathbf{q}}_a$ is identically zero. Then, the actual end-effector velocity is given by

$$\dot{\mathbf{x}}_a = {}^i J \dot{\mathbf{q}}_c \tag{3}$$

where ${}^i J$ is the post-failure Jacobian, given by

$$ {}^i J = [\, \mathbf{j}_1 \quad \cdots \quad \mathbf{j}_{i-1} \quad \mathbf{0} \quad \mathbf{j}_{i+1} \quad \cdots \quad \mathbf{j}_n \,]. \tag{4}$$

It is assumed that the joint position sensors are still operational.

A common method for generating $\dot{\mathbf{q}}$ is the inverse kinematic scheme

$$\dot{\mathbf{q}} = G \dot{\mathbf{x}} \tag{5}$$

where G is a generalized inverse of J satisfying the Penrose condition $JGJ = J$. A frequently encountered generalized inverse is the pseudoinverse J^+, which yields the least squares minimum norm solution. For full rank J, the pseudoinverse can be expressed as $J^+ = J^T (JJ^T)^{-1}$.

In this work a general class of tasks characterized by sequences of point-to-point moves is considered. The commanded end-effector velocity is simply straight line motion toward the desired task position $\mathbf{x}_d$

$$\dot{\mathbf{x}}_c = K_e (\mathbf{x}_d - \mathbf{x}_a) \tag{6}$$

where $\mathbf{x}_a$ is the actual position of the end effector and K_e is a constant position error gain that is adjusted when necessary to limit the commanded end-effector velocity to a maximum allowable value.

In the event of a locked-joint failure, the actual end-effector velocity in general will not be as commanded by (6). In particular, if joint i fails then the actual end-effector velocity is given by

$$\dot{\mathbf{x}}_a = ({}^i JG) K_e (\mathbf{x}_d - \mathbf{x}_a). \tag{7}$$

Thus, $\dot{\mathbf{x}}_a$ will equal $\dot{\mathbf{x}}_c$ only if the failed joint is not commanded to move so that ${}^i I \dot{\mathbf{q}}_c = \dot{\mathbf{q}}_c$.

The failure-tolerant inverse J^λ considered in this study minimizes the peak error in the end-effector velocity given by

$$\mathrm{PKE}(\dot{\mathbf{x}}_c) = \max_i \| \dot{\mathbf{x}}_c - ({}^i JJ^\lambda) \dot{\mathbf{x}}_c \| \tag{8}$$

where J^λ is a damped-least squares inverse, the damping factor of which is obtained by solving the following optimization problem that minimizes $\mathrm{PKE}(\dot{\mathbf{x}}_c)$

$$\begin{aligned} \text{Min.} \quad & \max_i \| \dot{\mathbf{x}}_c - ({}^i JJ^\lambda) \dot{\mathbf{x}}_c \| \quad i = 1, \ldots, n, \text{ over } \lambda \\ \text{S.t.} \quad & \| \dot{\mathbf{x}}_c - (JJ^\lambda) \dot{\mathbf{x}}_c \| < \epsilon_{\mathrm{pre}} \end{aligned} \tag{9}$$

where $J^\lambda = J^T (JJ^T + \lambda^2 I)^{-1}$ and ϵ_{pre} denotes the maximum allowable pre-failure error. An efficient method to compute the optimal damping-factor is presented in [28]. In this study, the maximum pre-failure allowable error for the damped least-squares inverse was set to be $\epsilon_{\mathrm{pre}} = 0.5$ for a unit-norm $\dot{\mathbf{x}}_c$. It was shown in [28] that for this choice of ϵ_{pre} the peak angular deviation (PAD) of the end-effector, defined as

$$\mathrm{PAD}(\dot{\mathbf{x}}_c) = \max_i \left| \arccos \left(\frac{\dot{\mathbf{x}}_c^T ({}^i JJ^\lambda) \dot{\mathbf{x}}_c}{\| \dot{\mathbf{x}}_c \| \| ({}^i JJ^\lambda) \dot{\mathbf{x}}_c \|} \right) \right| \tag{10}$$

where the range of $\arccos$ is $[-\pi \pi)$, is guaranteed to be less than $30°$. Preliminary trials indicated that this trade-off in the pre-failure performance does not significantly degrade an operator's performance with a healthy manipulator and so was chosen for this study.[2]

III. DESCRIPTION OF THE EXPERIMENT

A. Experimental Test Bed

The test bed used in this study was a computer graphic simulation of a 3-DOF planar manipulator with unit link lengths (1 m each)[3] that an operator controlled using a 2-D joystick. First order, i.e., velocity control, was employed to command motion of the end effector of the manipulator with the commanded end-effector velocity being proportional to the displacement of the joystick from its center position. The goal of the operator in each of the experimental tasks was to command the end effector from its initial position to a given target circle inside

[2]To ensure that the noticeable difference in the computation times of the pseudoinverse and the damped least-squares inverse did not affect the operators' strategy and performance, the computation of the pseudoinverse was intentionally slowed down to that of the damped least-squares inverse. However, the resulting computation time was sufficiently fast for real-time teleoperation.

[3]The manipulator and all other objects are displayed on the screen at 1/50th scale. All measurements presented henceforth must thus be appropriately scaled to obtain correct display size.

the workspace of the manipulator. The display presented to the operator consisted of two windows—one displaying the graphic simulation of the manipulator and the other depicting the joystick deflection, i.e., the commanded end-effector velocity. Fig. 1 shows a snapshot of the two parts of the display.

The manipulator display window [see Fig. 1(a)] consisted of a scaled wire-frame rendition of the 3-DOF planar manipulator, the boundary of the manipulator's workspace, and the initial $(\mathbf{x}_i)$ and target $(\mathbf{x}_d)$ end-effector positions. Since this study focuses on gross motion, the target end-effector position was represented as a circle of radius 0.1 m inside which the operator was required to position the end effector for successful task completion. The operators were instructed to command the end effector to move in a straight line toward the target at all times until the task was completed. In addition to the other items displayed in this window, a straight line [shown dashed in the Fig. 1(a)] connecting the current end-effector position, $\mathbf{x}_a$, to the desired end-effector position, $\mathbf{x}_d$, was also shown. The primary motivation for displaying this dashed line was to induce the operators to consistently employ the strategy of straight-line motion toward the target.

The velocity-command window [see Fig. 1(b)] displayed the displacement of the joystick which was scaled and used as the commanded velocity of the end effector in the task space. The maximum commandable end-effector velocity of 0.05 m/cyc was represented as a circle. The commanded-velocity vector was shown as a straight line with a circle at its head. To reduce the sensitivity of the joystick in centering (zeroing of the commanded velocity), a circular dead-zone of radius 0.005 m was defined about the center position of the velocity command.

The software was implemented in MATLAB on a Dell Pentium-Pro PC workstation running Windows NT. A standard 2-button, self-centering joystick (Logitech Wingman) was used as the input device. A task was started by pressing the index-finger button on the joystick and was terminated by bringing the end effector to rest in the target circle. The tasks were selected such that they could all be successfully completed, and so the operators were strongly encouraged to complete every task regardless of its difficulty. However, if the operator did find a task to be *extremely* difficult or disorienting, an option to abort the trial was provided.

B. Protocol

A group of twenty-four subjects was tested in this experiment. The group, comprised of faculty and students from the schools of science and engineering at Purdue University, included twenty-one males and three females, with ages ranging from 20–49. Since none of the subjects were assumed to have any prior experience with such tasks, they were all trained until they achieved a desired level of performance on point-to-point motion tasks using pseudoinverse control without failures. In each of these training tasks the initial and the final points were chosen to be spaced 2 m apart and were randomly selected from the entire workspace of the manipulator. A subject was considered to have been satisfactorily trained after the successful completion of five successive tasks, each within a predetermined time limit, and having successfully completed at least ten tasks. To determine the time limit for successful task completion, three operators (not of the group to be tested) were asked to repeatedly perform one such task twenty times. The minimum value of the task completion time over the twenty tasks was then determined for each of these three operators. The time limit for the training was set to 1.2 times the average minimum completion time (7.5 cycles).

The test tasks were presented to the operators immediately after they were trained and appeared essentially identical to the training tasks. Each subject was presented twenty-four sets of point-to-point motion

TABLE I
PERFORMANCE MEASURES

Measure	Description
M_1	*Task Completion Time (cycles)* (TCT_{cyc}): Computed in terms of the number of controller cycles rather than in elapsed time. Allows control schemes to be evaluated independent of the underlying computational cost.
M_2	*Task Completion Time (seconds)* (TCT_{secs}): A measure of the elapsed time.
M_3	*Total Path Length* (TPL): The total length of the end-effector trajectory.
M_4	*Peak Trajectory Deviation (Magnitude)* (PTD_{mag}): The maximum displacement of the end effector from the desired straight-line trajectory, measured over the entire task.
M_5	*Peak Trajectory Deviation (Angle)* (PTD_{ang}): The maximum angular deviation of the end effector from the desired straight-line trajectory, measured from the initial end-effector position, over the entire task.
M_6	*Peak End-Effector Velocity Error (Magnitude)* (PVE_{mag}): The peak magnitude of the difference in the commanded end-effector velocity $\dot{\mathbf{x}}_c$ and its actual value $\dot{\mathbf{x}}_a$, measured over the entire task.
M_7	*Mean End-Effector Velocity Error (Magnitude)* (MVE_{mag}): The mean magnitude of the difference in the commanded end-effector velocity $\dot{\mathbf{x}}_c$ and its actual value $\dot{\mathbf{x}}_a$, computed over the entire task.
M_8	*Peak End-Effector Velocity Error (Angle)* (PVE_{ang}): The peak angular deviation between the commanded end-effector velocity $\dot{\mathbf{x}}_c$ and its actual value $\dot{\mathbf{x}}_a$, measured over the entire task.
M_9	*Mean End-Effector Velocity Error (Angle)* (MVE_{ang}): The mean angular deviation between the commanded end-effector velocity $\dot{\mathbf{x}}_c$ and its actual value $\dot{\mathbf{x}}_a$, computed over the entire task.
M_{10}	*Number of Trajectory Corrections* (NTC): The total number of corrections made by the operator/computer to compensate for any increase in the position error of the end-effector.
M_{11}	*Excursion of Input Device/Master* (EIP): The total work done by an operator while operating the joystick to perform the task. It is expressed in terms of the total displacement of the device tip, over the task duration, normalized by the maximum possible offset of the tip.
M_{12}	*Subjective Operator Assessment* (SOA): The operator's rating of the the task as "easy", "moderate", "difficult", or "very difficult", assigned numeric values from 1–4 respectively.

tasks. Each of these sets consisted of four trials with identical initial configurations and final positions but four different scenarios:

1) healthy arm driven by pseudoinverse control;
2) healthy arm driven by the damped least-squares based control scheme;
3) failed arm driven by pseudoinverse control;
4) failed arm driven by the damped least-squares based control scheme.

To minimize any carry-over effects within a set, these four conditions were presented in a random order. To minimize carry-over and learning effects over the different tasks, the order in which the twenty-four test tasks were presented to the twenty-four operators was determined using a balanced Latin square [29]. This ordering ensured that over the twenty-four subjects each test task appeared once in each position, and moreover, that each test task was preceded and followed by every other test task equally often. At the end of each trial the operators were required to rate the task as "easy," "moderate," "difficult," or "very-difficult," and were instructed to base the rating on their level of comfort in controlling the manipulator.
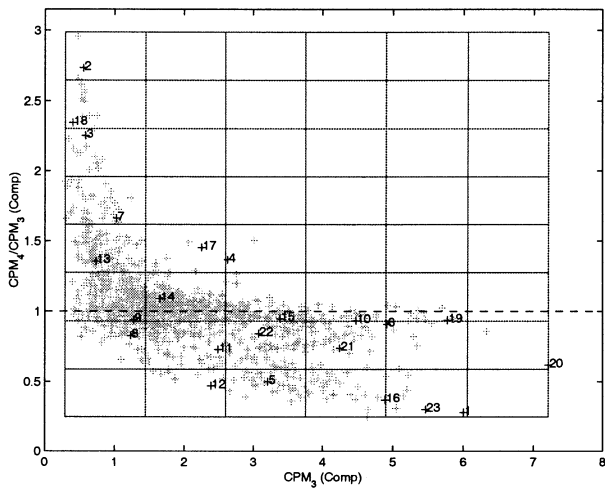
Fig. 2. Twelve-hundred randomly generated tasks plotted with gray plus signs as functions of task "difficulty," CPM_3, and relative performance, CPM_4/CPM_3, where the subscript 3 denotes the pseudoinverse case with failure, and 4 denotes the damped least-squares inverse case with failure. The values corresponding to the test tasks are indicated with black plus signs and labeled with the test task index numbers. Since the manipulator under computer-control was unable to complete test task 24 for scenario 3, CPM_3 for test task 24 is undefined and so is not shown.

C. Measures of Performance

The twelve measures of performance used in this study are summarized in Table I. While some of these measures have been incorporated from previous work on performance evaluation of teleoperated systems [15], [30]–[33], additional measures focusing specifically on failure related performance issues are also proposed. Based on these twelve measures, a cumulative performance measure (CPM) is defined as

$$CPM = \sum_{i=1}^{12} \frac{M_i}{\max(M_i)} \qquad (11)$$

where all measures, normalized by their respective maximum values over all the different cases (tasks/control schemes) being compared, are combined with equal weighting.

D. Choice of Test Tasks

The test tasks were selected from a set of 1200 tasks where the initial configurations were randomly generated and the final task positions were randomly selected from those that were 2 m from the initial position and were reachable with the failure considered. The 1200 tasks consisted of three sets of four-hundred tasks, corresponding to failures of each of the three joints. To select a set of test tasks encompassing a wide spectrum of post-failure behaviors, the following procedure was adopted. Each of the 1200 tasks was performed under computer control and the measures M_1–M_{11} and the CPM were computed for each of the four scenarios listed in Section III-B. The CPM of scenario 3, denoted by CPM_3, is a measure of the post-failure performance degradation of the system when operated under pseudoinverse control and the CPM of scenario 4, denoted by CPM_4, is the corresponding measure of performance for the proposed failure-tolerant inverse. The selection of tasks over a range of values of CPM_3 yields test tasks of varying degrees of "difficulty," while that based on the ratio CPM_4/CPM_3 yields test tasks with varying relative performance of the two control strategies. Using these two quantities, tasks exhibiting a range of post-failure characteristics were selected. Each of the 1200 tasks is plotted as a function of CPM_3 and CPM_4/CPM_3, in Fig. 2. A CPM_4/CPM_3 value of one represents a break-even condition where the proposed inverse
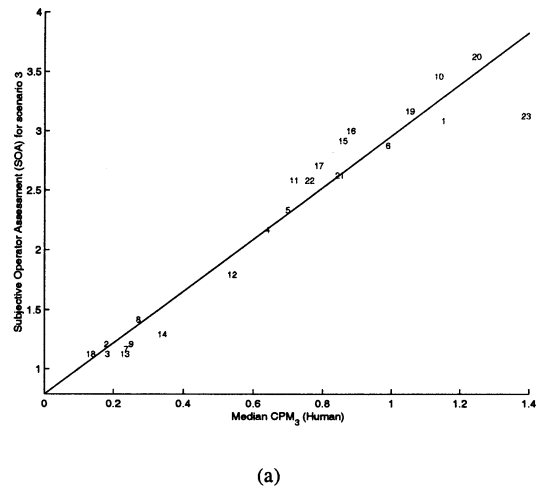
(a)
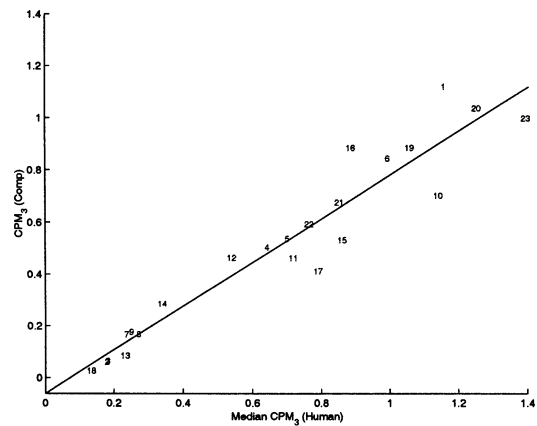
(b)

Fig. 3. (a) Twenty-four test tasks plotted as a function of the "actual" task difficulty as measured by the median value of CPM_3 for the operators, and the operators' "perceived" difficulty as measured by the median SOA for scenario 3. (b) The test tasks plotted as a function of the "actual" task difficulty as measured by the median value of CPM_3 for the operators, and the "anticipated" task difficulty as measured by CPM_3 for the computer.
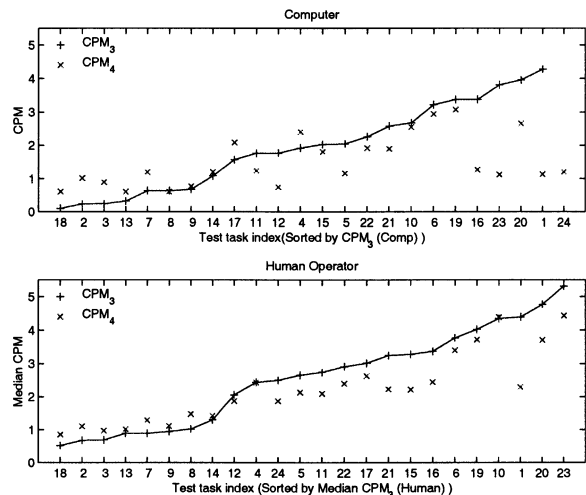

Fig. 4. Performance of the computer and that of the median operator, plotted for scenarios 3 and 4, for the 24 test tasks. For each mode, i.e., computer/human controlled, the test tasks are sorted based on the corresponding task difficulty as measured by CPM_3.
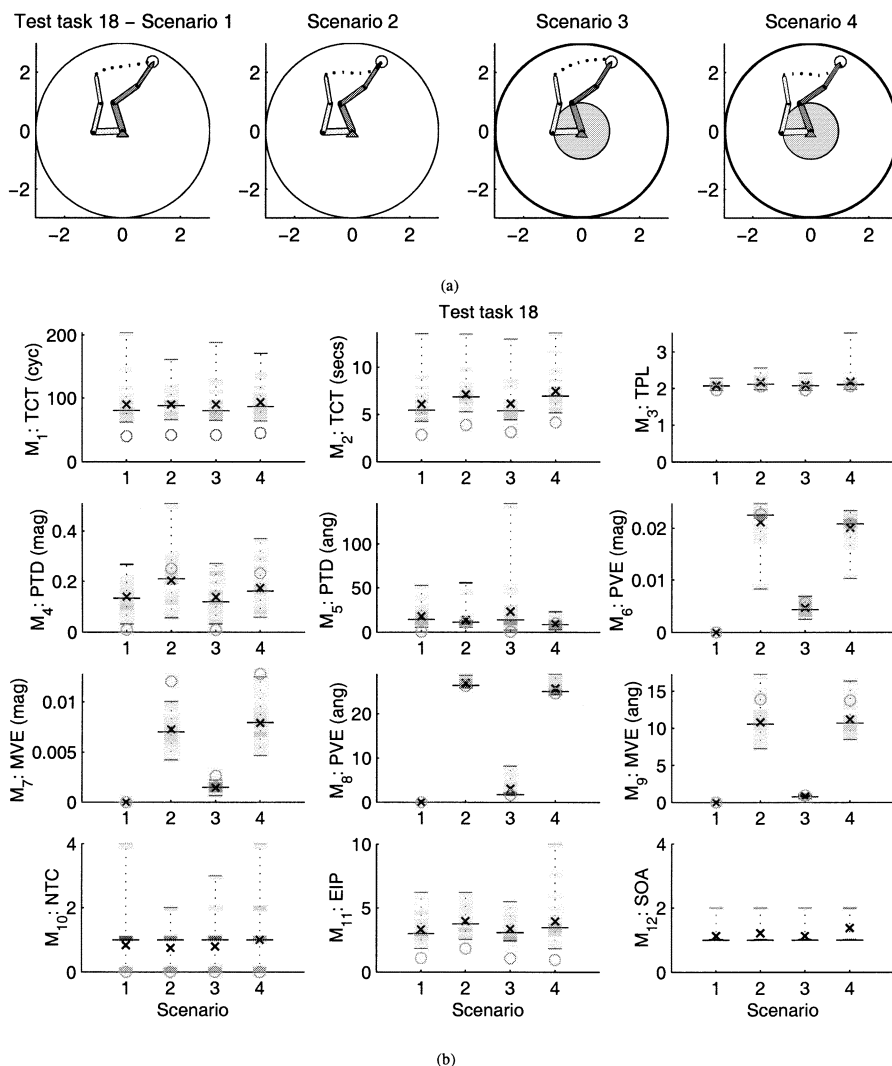
Fig. 5. (a) Initial and final configurations of the manipulator and the end-effector trajectory followed by the median operator for test task 18. (b) Distribution of operator data for each of the twelve measure and the four scenarios. The median values of the operator data are shown using horizontal lines, the mean values using crosses, and the frequency distribution of the data using gray shading. The performance of the computer is plotted for reference with circles.

matches the performance of the pseudoinverse after the failure. When the value is less than one, the proposed inverse out-performs the pseudoinverse and vice-versa. As seen in Fig. 2, the entire range of data on this plot was divided into forty-eight equal-area bins formed by a 6×8 linearly-spaced grid. Twenty-two test tasks (labeled 1–22), one from each of the twenty-two nonempty bins, were selected such that the entire set of test tasks covered the range of values for each of the eleven individual performance measures M_1–M_{11}. This ensured both a wide range of CPM values and a roughly uniform distribution of test tasks over the eleven measures. While twenty-two test tasks were selected using the procedure defined above, tasks 23 and 24 were added as special cases. Task 23 was seen to be particularly disorienting in preliminary teleoperation experiments, while task 24 is an example where for scenario 3, under computer control, the manipulator has a tendency to get drawn into a "stationary" configuration and thus does not converge to the desired end-effector position [13].

IV. RESULTS AND ANALYSIS

All operators were able to complete each of the twenty-four test tasks in each of the four scenarios. As an example, three of the test tasks are shown in Figs. 5–7, where for each of the four scenarios the task-space

trajectories followed by the manipulator under the command of an operator representative of the median are shown in part (a). In each of these figures the manipulator is shown in light gray in its initial configuration and in dark gray in the final configuration. The trajectory is shown with a dotted line and the desired task position is shown with a circle. The portion of the manipulator's workspace lost due to the failure is highlighted in these figures in gray. However, this information was not presented to the operators. The performance of all the operators for the three tasks is summarized in part (b) of Figs. 5–7. Each of these figures shows bar-graphs of all four scenarios for the corresponding task for each of the twelve measures defined in Table I. The bar shown as a dotted line denotes the range of the data for the entire group of operators, with the upper and lower limits of the data marked by short, horizontal lines. The average value of the data set is indicated with a cross, and the median with a horizontal line. The frequency distribution of the data for the twenty-four operators is indicated along the dotted bar in shades of gray, with white indicating zero operators, and black denoting twenty-four operators. Also shown in these bar charts is the performance of the system when operated under computer control. The value of each measure for the computer is indicated with a circle.

As a first step in the analysis of the data, the relationship between the "anticipated" difficulty of a test task as represented by CPM_3 for the
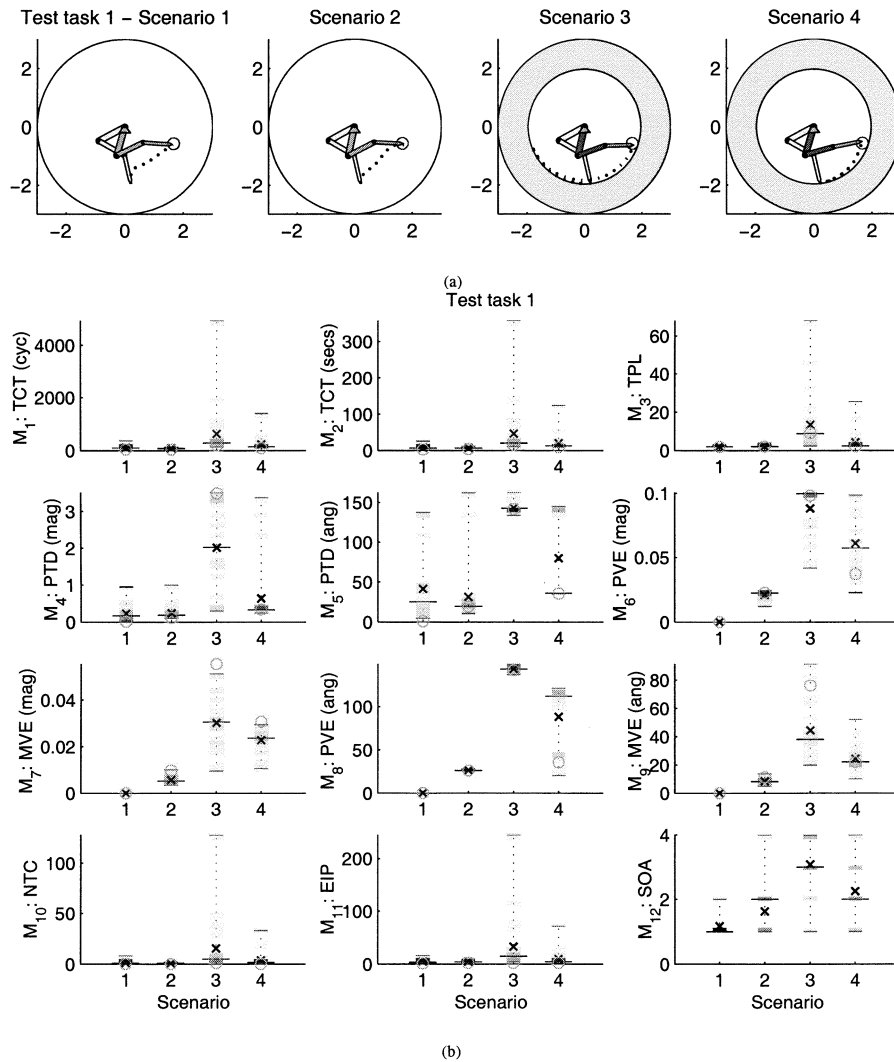
Fig. 6. (a) Initial and final configurations of the manipulator and the end-effector trajectory followed by the median operator for test task 1. Note that for the initial part of the task in scenario 3 the end effector actually moves away from the desired position. (b) Distribution of operator data for each of the twelve measure and the four scenarios. The median values of the operator data are shown using horizontal lines, the mean values using crosses, and the frequency distribution of the data using gray shading. The performance of the computer is plotted for reference with circles.

computer, its "actual" difficulty as measured by the median CPM_3 for the operators, and its "perceived" difficulty as measured by the median SOA, was studied. The linear relationship between the actual difficulty of the tasks and the corresponding perceived difficulty (correlation coefficient = 0.969) as seen in Fig. 3(a) indicates that CPM is a good measure of task difficulty. The linear relationship between the actual and the perceived task difficulty (correlation coefficient = 0.956) shown in Fig. 3(b) implies that the test task selection process indeed helped identify tasks with desired levels of difficulty.

Shown in Fig. 4 are CPM_3 and CPM_4, plotted independently as functions of the tasks (sorted by CPM_3), for both the computer controlled and operator-controlled modes. This figure indicates that as the difficulty of the test task increases, the performance of the proposed failure-tolerant scheme, relative to that of the pseudoinverse, improves. As expected, the failure-tolerant inverse seems to outperform the pseudoinverse for difficult tasks. In addition, this figure indicates that the performance degradation for the proposed scheme in "easy" tasks is small relative to the gain observed for "difficult" tasks.

To validate our hypothesis that the post-failure performance of the proposed inverse is significantly better than that of the pseudoinverse, a within subjects, 2×3 analysis of variance was performed, with the

inverse type being one of the factors, and task difficulty (easy, medium, difficult) being the other. The post-failure pseudoinverse performance was compared with that of the proposed inverse across the 24 test tasks, grouped into three sets (of eight tasks each) based on the level of difficulty—easy, medium, and difficult, as defined by CPM_3. The main effect of the control scheme was significant, $F(1, 23) = 105.08$, $p < 0.001$, with the overall mean for the pseudoinverse (2.661) being higher than that for the proposed inverse (2.313). So, the proposed inverse resulted in better performance overall. The main effect of difficulty was also significant, $F(2, 46) = 1,093$, $p < 0.001$. The overall mean increased from easy (1.096) to medium (2.586) to difficult (3.778). This just confirmed that the difficulty distinction was valid, that is, performance decreased as difficulty increased. Finally, the interaction between the control scheme and difficulty was significant as well, $F(2, 46) = 101.53$, $p < 0.001$. This interaction shows that the relation between the two algorithms differed as a function of difficulty. The performance of the two control schemes was then compared using T-tests. While for the easy tasks performance was traded off (difference of 0.276) with the proposed inverse, $t(23) = -10.77$, $p < 0.001$, for the medium tasks performance was significantly better (difference of 0.504) with the proposed inverse
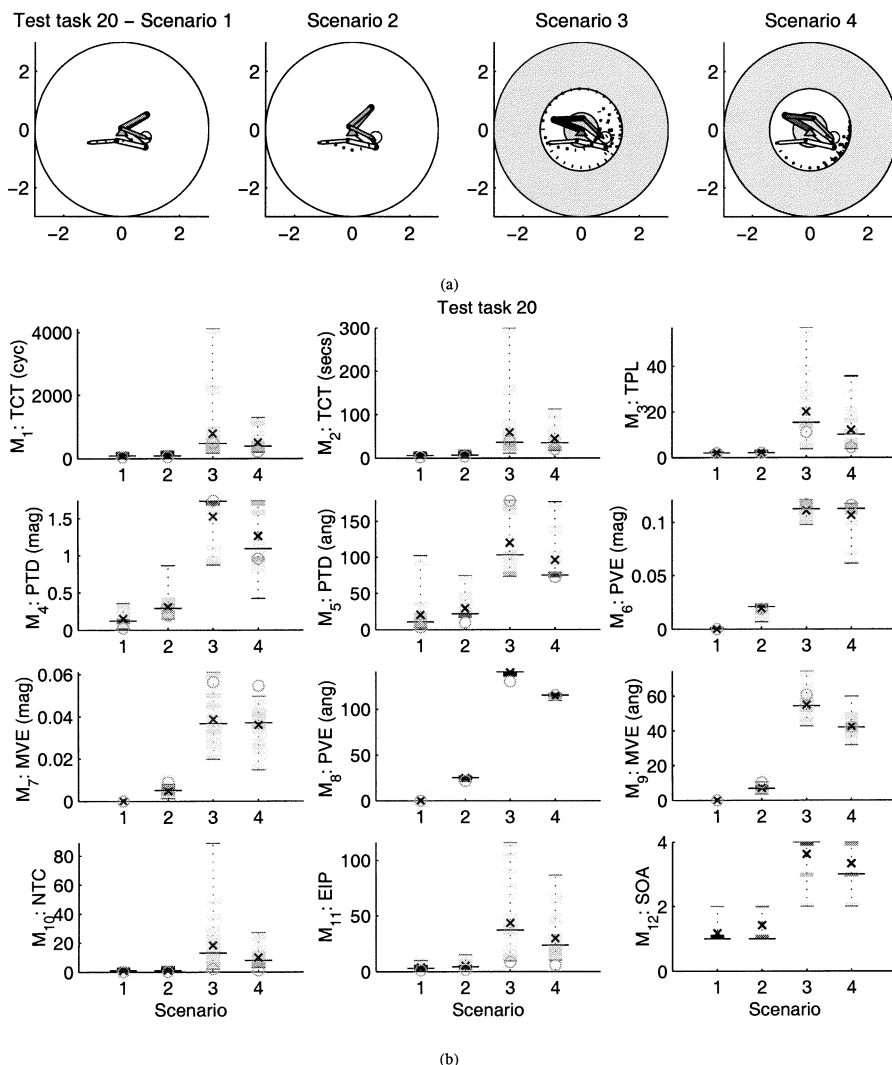
Fig. 7. (a) Initial and final configurations of the manipulator and the end-effector trajectory followed by the median operator for test task 20. Note the large deviation from the desired path in scenario 3. (b) Distribution of operator data for each of the twelve measure and the four scenarios. The median values of the operator data are shown using horizontal lines, the mean values using crosses, and the frequency distribution of the data using gray shading. The performance of the computer is plotted for reference with circles.

than with the pseudoinverse, $t(23) = 7.83$, $p < 0.001$, and the same pattern held for the difficult tasks (difference of 0.815), $t(23) = 11.67$, $p < 0.001$. These results show that the post-failure performance is better with the proposed inverse than with the pseudoinverse, with this advantage being, in general, restricted to the medium and high difficulty conditions.

Upon comparing the results for the human operators with the computer-controlled system, it is seen that human operators suffer less in terms of the performance when using the proposed scheme in easy tasks, while gaining slightly less than the computer in difficult tasks. As will be seen with a specific example shortly, easy tasks make the control schemes and the failure/nonfailure conditions indistinguishable to an operator, resulting in less relative performance degradation. For difficult tasks, operators often compromise the straight-line motion strategy with the hope of improving their performance. This compromise results in the smaller relative performance gain for the operators as compared to the computer.

For the purpose of illustration, the operator performance for test tasks 18, 1, and 20 are analyzed in greater detail. In test task 18 the pseudoinverse performs better than the failure-tolerant inverse, whereas in test tasks 1 and 20 the failure-tolerant inverse out-performs the pseu-

doinverse. From the trajectory plots of the median operator for test task 18 shown in Fig. 5(a) and the corresponding performance measures shown in Fig. 5(b), it is seen that the failure-tolerant scheme is only marginally inferior to the pseudoinverse in performance. In fact, the performance measures are so small that the operators are unable to distinguish between the control schemes or even realize if a failure has occurred, as seen from the operators' subjective rating (SOA). The trajectory plots of the median operator for test task 1 shown in Fig. 6(a) and the corresponding performance measures of both the median operator and the computer shown in Fig. 6(b) indicate that, while only marginally degrading the performance of the healthy manipulator, the failure-tolerant scheme significantly improves each of the measures when a failure occurs. From Fig. 6(a) it is seen that for the initial part of the task the end effector actually moves away from the desired position. Such behavior can be very disorienting for a human operator. Task 20 demonstrates behavior similar to that of Task 1 (see Fig. 7). For this task as well, the manipulator exhibits very disorienting behavior in scenario 3, while being considerably better behaved in scenario 4. The importance of stimulus-response compatibility is well studied in human factors literature [34]. In terms of the stimulus-response compatibility issue, two aspects of the change in control operation resulting

from a failure contribute to the operators' disorientation, as in each of the scenarios above. One is that at several starting configurations the initial movement of the manipulator is in a direction opposite to that of the control movement. The other is the unpredictable and continuously changing stimulus-response relationship as the manipulator moves.

While the performance of the operators for the other test tasks were similar to those of either task 18, 1, or 20, test task 24 presented a special example for which the manipulator, under computer control, converges to a position other than that desired when using the pseudoinverse in the failure scenario. Interestingly, by changing the control strategy to one other than straight-line motion toward the target, the operators were able to resist the manipulator's tendency to get drawn into the stationary configuration, and were thus able to complete the task when using pseudoinverse control. The performance of the failure-tolerant scheme was better than that of the pseudoinverse in failure scenarios for this task as well.

V. CONCLUSIONS

This experimental study demonstrates that the post-failure performance of a teleoperated system can be significantly improved, even when failures are not identified, by using an appropriate failure-tolerant control strategy. With enhanced local behavior after a failure, as with the failure-tolerant scheme considered here, operators find the manipulator to be more maneuverable and predictable in behavior as compared to the behavior observed with pseudoinverse control. The extent to which the post-failure performance of a manipulator can be improved depends upon the extent to which one is willing to compromise the pre-failure performance.

REFERENCES

[1] R. Colbaugh and M. Jamshidi, "Robot manipulator control for hazardous waste-handling applications," *J. Robot. Syst.*, vol. 9, no. 2, pp. 215–250, 1992.

[2] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "Robotic fault detection and fault tolerance: A survey," *Reliab. Eng. Syst. Safety*, vol. 46, pp. 139–158, 1994.

[3] *Unimate Puma Mark II Robot 500 Series Equipment Manual*, Unimation Inc.,, Danbury, CT, 1985.

[4] K. N. Groom, A. A. Maciejewski, and V. Balakrishnan, "Real-time failure tolerant control of kinematically redundant manipulators," *IEEE Trans. Robot. Automat.*, vol. 15, pp. 1109–1116, Dec. 1999.

[5] J. D. English and A. A. Maciejewski, "Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 566–575, Aug. 1998.

[6] G. Liu, "Control of robot manipulators with consideration of actuator performance degradation and failures," in 2001 Int. Conf. Robot. Automat., Seoul, Korea, May 21–26, 2001.

[7] E. C. Wu, J. C. Hwang, and J. T. Chladek, "Fault-tolerant joint development for the space shuttle remote manipulator system: Analysis and experiment," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 675–684, Oct. 1993.

[8] C. L. Lewis and A. A. Maciejewski, "Dexterity optimization of kinematically redundant manipulators in the presence of failures," *Comput. Electr. Eng.*, vol. 20, no. 3, pp. 273–288, May 1994.

[9] C. J. J. Paredis, W. K. F. Au, and P. K. Khosla, "Kinematic design of fault tolerant manipulators," *Comput. Elect. Eng.*, vol. 20, no. 3, pp. 211–220, May 1994.

[10] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "A dynamic fault tolerance framework for remote robots," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 477–490, Aug. 1995.

[11] Y. Ting, S. Tosunoglu, and D. Tesar, "A control structure for fault-tolerant operation of robotic manipulators," in *1993 Int. Conf. Robot. Automat.*, Atlanta, GA, May 2–6, 1993, pp. 684–690.

[12] M. Goel, A. A. Maciejewski, and V. Balakrishnan, "An analysis of the post-fault behavior of robotic manipulators," in *1997 Int. Conf. Robot. Automat*, Albuquerque, NM, Apr. 20–25, 1997, pp. 2583–2588.

[13] ——, "Undetected locked-joint failures in kinematically redundant manipulators: A workspace analysis," in *Proc. Int. Conf. Intell. Robot. Syst.*, Victoria, BC, Canada, Oct. 12–16, 1998, pp. 317–322.

[14] C. C. Hood, J. V. Draper, and S. J. Handel, "The impact of manipulator failures on remote task performance," in *Trans. Ameri. Nuclear Soc. Summer Meeting*, Nashville, TN, 1990, pp. 424–425.

[15] J. V. Draper, S. J. Handel, and C. C. Hood, "The impact of partial joint failure on teleoperation task performance," in *Robot. Remote Syst.: Proc. Fourth American Nuclear Society Topical Meeting Robot. Remote Syst.*, M. Jamshidi and P. J. Eicker, Eds., Albuquerque, NM, Feb. 24–28, 1991, pp. 433–439.

[16] W. R. Ferrell, "Remote manipulation with transmission delay," *IEEE Trans. Hum. Factors Electron.*, pp. 24–32, Sept. 1965.

[17] T. B. Sheridan, "Space teleoperation through time delay: Review and prognosis," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 592–605, Oct. 1993.

[18] S. Kimura, M. Takahashi, T. Okuyama, S. Tsuchiya, and Y. Suzuki, "A fault-tolerant control algorithm having a decentralized autonomous architecture for space hyper-redundant manipulators," *IEEE Trans. Syst., Man, Cybern. A*, vol. 28, pp. 521–527, July 1998.

[19] B. M. Harpel, J. B. Dugan, I. D. Walker, and J. R. Cavallaro, "Analysis of robots for hazardous environments," in *Proc. Ann. Reliab. Main. Symp.*, Philadelphia, PA, Jan. 1997, pp. 111–116.

[20] A. K. Pradeep, P. J. Yoder, R. Mukundan, and R. J. Schilling, "Crippled motion in robots," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 24, pp. 2–13, Jan. 1988.

[21] C. L. Lewis and A. A. Maciejewski, "Fault tolerant operation of kinematically redundant manipulators for locked joint failures," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 622–629, Aug. 1997.

[22] C. J. J. Paredis and P. K. Khosla, "Mapping tasks into fault tolerant manipulators," in *1994 Int. Conf. Robot. Automat.*, San Diego, CA, May 8–13, 1994, pp. 696–703.

[23] F. Miyazaki and Y. Masutani, "Robustness of sensory feedback control based on imperfect Jacobian," in *Robot. Res.: 5th Int. Symp.*, H. Miura and S. Arimoto, Eds., 1989, pp. 201–208.

[24] M. Shoham and Y. Koren, "Motion control algorithms for sensor-equipped robots," *ASME J. Dyn. Syst., Measure. Contr.*, vol. 109, pp. 335–344, Dec. 1987.

[25] Y. Ting, S. Tosunoglu, and B. Fernández, "Control algorithms for fault-tolerant robots," in *Int. Conf. Robot. Automat.*, San Diego, CA, May 8–13, 1994, pp. 910–915.

[26] E. Wu, J. Hwang, and J. Chladek, "A failure tolerant joint design for the space shuttle remote manipulator system: Analysis and experiment," in *Proc. 1st IEEE Conf. Control Applications*, Dayton, OH, Sept. 13–16, 1992, pp. 330–335.

[27] M. Goel, A. A. Maciejewski, and V. Balakrishnan, "The design of control strategies tolerant to undetected failures in kinematically redundant manipulators," presented at the Int. Conf. Robot. Automat., Detroit, MI, May 10–15, 1999.

[28] M. Goel, "Tolerating Undetected Failures in Robotic Manipulators," Ph.D. dissertation, Purdue University, West Lafayette, IN, 1998.

[29] B. H. Kantowitz and H. L. R. Roediger, III, *Experimental Psychology: Understanding Psychological Research*. St.Paul, MN: West, 1984.

[30] V. Lumelsky, "On human performance in telerobotics," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, pp. 971–982, Sept./Oct. 1991.

[31] G. F. McLean, B. Prescott, and R. Podhorodeski, "Teleoperated system performance evaluation," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 796–804, May 1994.

[32] G. Magenes, J. L. Vercher, and G. M. Gauthier, "Hand movement strategies in telecontrolled motion along 2-D trajectories," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 242–257, Mar./Apr. 1992.

[33] D.-Y. Hwang and B. Hannaford, "Teleoperation performance with a kinematically redundant slave robot," *Int. J. Robot. Res.*, vol. 17, no. 6, pp. 579–597, June 1998.

[34] R. W. Proctor and T. G. Reeve, Eds., *Stimulus-Response Compatibility: An Integrated Perspective*. Amsterdam, The Netherlands: North-Holland, 1990.