

Toward safe and stable time-delayed mobile robot teleoperation through sampling-based path planning

Jorge Nieto^{†, ‡*}, Emanuel Slawiński[‡], Vicente Mut[‡] and Bernardo Wagner[†]

[†]Real-Time Systems Group, Leibniz Universität Hannover, D-30167, Germany

[‡]Instituto de Automática, Universidad Nacional de San Juan, J5400ARL, Argentina

(Received in Final Form: June 10, 2011; accepted June 9, 2011. First published online: July 11, 2011)

SUMMARY

This work proposes a teleoperation architecture for mobile robots in partially unknown environments under the presence of variable time delay. The system is provided with artificial intelligence represented by a probabilistic path planner that, in combination with a prediction module, assists the operator while guaranteeing a collision-free motion. For this purpose, a certain level of autonomy is given to the system. The structure was tested in indoor environments for different kinds of operators. A maximum time delay of 2 s was successfully coped with.

KEYWORDS: Mobile robots; Time-delayed teleoperation; Path planning; Supervisory control; Human operator.

1. Introduction

There is an increasing interest on assistance techniques for the human operator of a mobile robot teleoperation system in order to improve the quality of service in terms of stability and reliability as well as execution time and comfort for the operator. While the opposite branch to teleoperation, i.e., autonomy, aims at releasing the human from the workload of continuously acting onto the navigation system, it is not always possible to keep the human operator out of the control loop (control system). One essential reason for this is the fact that the human possesses decision-making, reasoning, and judgment skills that are beyond the limits of current computing systems. In a teleoperation system, the human operator can be regarded as belonging to a supervisory level; in such a system, the teleoperated robot must follow motion commands directly or indirectly sent by the operator. Since the human operator is one of the most relevant elements in a teleoperation system, special attention is paid to providing him/her with enhanced information in order to give him/her a sense of immersion (characterized by transparency), while releasing him/her from stress. It has been shown that cognitive stress, especially present in long-time teleoperation, can dramatically affect the performance of the human operator²⁷ and also the quality of teleoperation tasks. Additionally, the presence of time delay in the communication channel is a crucial factor that

brings along complications when performing teleoperation tasks. Examples of such complications are detriment of stability, transparency loss, and collision risk between robot and obstacles in the environment.

Various strategies used in teleoperation of manipulators,¹⁰ such as wave variables,^{3,18} teleprogramming,^{8,9} supervisory control,^{5,22} as well as predictive display,^{4,12} control based on transparency¹⁶ and remote impedance control,^{13,21} could be used in teleoperation of mobile robots. Nevertheless, few papers show a stability analysis and experiments with time delay, as in Elhadj *et al.*,⁶ where the transmission of commands and force feedback is discontinuous, while the bandwidth of the force perceived by the user is limited by the magnitude of the time delay, or Lee *et al.*,¹⁷ where only constant delay is considered, or Slawiński *et al.*,²⁴ where the delayed command generated by the human operator is compensated by using a model of the human reaction, or Slawiński and Mut,²⁶ where augmented reality is used. In the last two cases, it is assumed that the global goal is known *a priori*. The problems mentioned above motivate the design of new control schemes to improve the performance of teleoperation systems for mobile robots in the presence of time delay in order to increase its application in industry, service, office, and home. The authors point out that, as far as they know, the application of a “user-friendly” path-planning module to mobile robot teleoperation is a novel approach. Usually, path planning is used for autonomous systems and not simultaneously with a human operator as assistance module.

In the present work, special attention is paid to the safety of the teleoperated system in terms of collision with the obstacles in the environment while attaining the common objective of achieving a trade-off between perception and stability when considering an *a priori* unknown global goal (which is a typical situation in exploration and reconnaissance tasks). The main point in this paper is that in a time-delayed teleoperation system, the feedback information received by the operator at a given time instant is generally delayed due to the communication channel between the operator station and the teleoperated system. The motion of the robot is determined by some desired path dynamically defined by the operator (and therefore, *a priori* unknown), according to the feedback information received and therefore a product of the complex decision processes in the human

* Corresponding author. E-mail: nieto@rts.uni-hannover.de

brain. For this reason, some compensation of this information is required that helps ensuring the system's safety, especially concerning collision avoidance.

One interesting aspect of the present work is the premise that the teleoperation system should assist the human operation while giving him/her the feeling of having the total control of the system. This is mainly supported on the fact that the sensor information is up to date for the remote site whereas it becomes outdated (and probably contaminated with additional noise) when it arrives at the local site. That is why, the authors propose a solution to assist the operator by means of path planning in combination with prediction techniques. Since path planning is a task that requires preprocessing and computation time, the authors chose a strategy that allows to quickly find collision-free paths (at the expenses of losing optimality). Sampling-based probabilistic path-planning algorithms fulfill this requirement: they sample the configuration space and capture its connectivity instead of explicitly calculating it. Thus, one or more collision-free paths can be calculated in a fast way. Though probabilistic path planners are not complete, their strength lies on their speed even in complex high-dimensional spaces. Probabilistic Roadmaps (PRMs)¹¹ were developed to work without any structural information of the configuration space. They sample the configuration space at random and build a connectivity map from which the path with the shortest length is given as solution. On the other side, rapidly exploring random trees (RRTs)^{14,15} are tree-like structures capable to quickly expand over the free space, thus capturing its connectivity while finding a straightforward solution, which is advantageous when compared to PRMs. In Nieto *et al.*,¹⁹ the authors developed an enhanced RRT-planner that allows a certain degree of optimality by generating several candidate solutions paths at once and choosing the most suitable according to a given optimality criterion. Later, in Nieto *et al.*,²⁰ the authors presented a first version of a teleoperation architecture that incorporated path-planning techniques to assist the operator. On this occasion, such work is taken as start point. In the present paper, modifications to the expansion and to the path-selection processes are proposed and some of the parameters of the path planner are updated online (in the previous work, those parameters were constant). This paper also explores the operator side of the modified teleoperation architecture and includes more complex experiments to demonstrate the capabilities of the system, to analyze the behavior of the system when considering various operators, and to compare the system performance when using or not the *variable sampling of commands*.

The paper is organized as follows: Section 2 presents the statement of the control problem. In Section 3, the control scheme for delayed teleoperation of mobile robots is described in detail. Section 4 shows the experimental results. Finally, the conclusions of this paper are given in Section 5.

2. Problem Description

The system under analysis in this paper is given by a teleoperation system for mobile robots in indoor environments in the presence of time delay (nevertheless, the analysis can be extended to telemanipulation). The

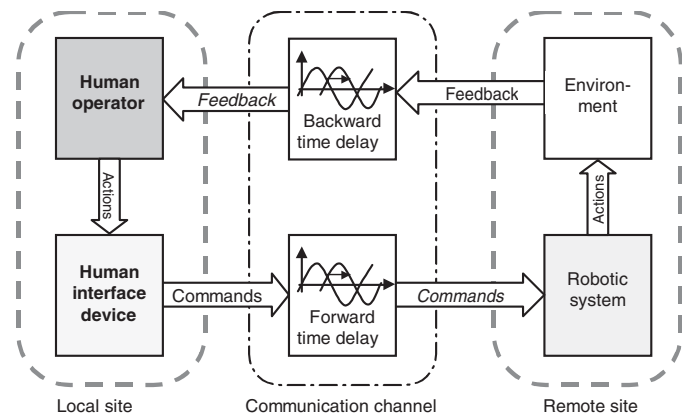


Fig. 1. General block diagram of a robot teleoperation system. The delayed information is represented in *italics*.

environment is considered to be partially unknown and fraught with obstacles. A reliable localization and orientation is assumed and the loss of information due to communication errors is not considered; these are not subject of study in the present paper.

In a general teleoperation system (Fig. 1), two main locations are considered: the control station, where the human operator is located, and the remote environment, in which the robot moves. The control station is usually considered as the local site whereas the site where the teleoperated system is located is called the remote site. At the local site, the human operator uses an interface device to send motion commands that the robot at the remote site must follow. At the same time, the operator receives feedback information from the teleoperated system by means of a visual interface; thus, the control loop is closed. A communication channel connects both sites making the bidirectional signal flow possible.

The term *remote* usually involves a large physical distance between both sites; this is, however, not always the case. In robot-aided surgery, for example, surgeon and robot can be in the same room. In the special case of mobile robots, a typical example is found on rescue operations with teleoperated bots, where the operator can be next to or far from the robot. In either case, whether separated miles away or just a few centimeters, there exists the possibility of time delay due to communication. Factors like distance, quality of the communication channel, and load on the data transport are responsible for this, even the human operator brings along inherent transport delays. In teleoperation, two main delays are considered: one implied in the data flow from the local to the remote site, known as *forward time delay*, and other in the opposite direction, called *backward time delay*. The operator interacts directly with a hardware that translates actions in electrical signals for their transmission to the remote site. In telemanipulation, joysticks are a suitable choice while in mobile robot teleoperation the use of steering wheel plus pedal system is more natural for the operator.

The information being fed back to the operator results from the continuous interaction between the robot and its environment, this information can be purely numerical, graphical, or sensorial, e.g., haptic feedback (aspects regarding the right presentation of this information to the

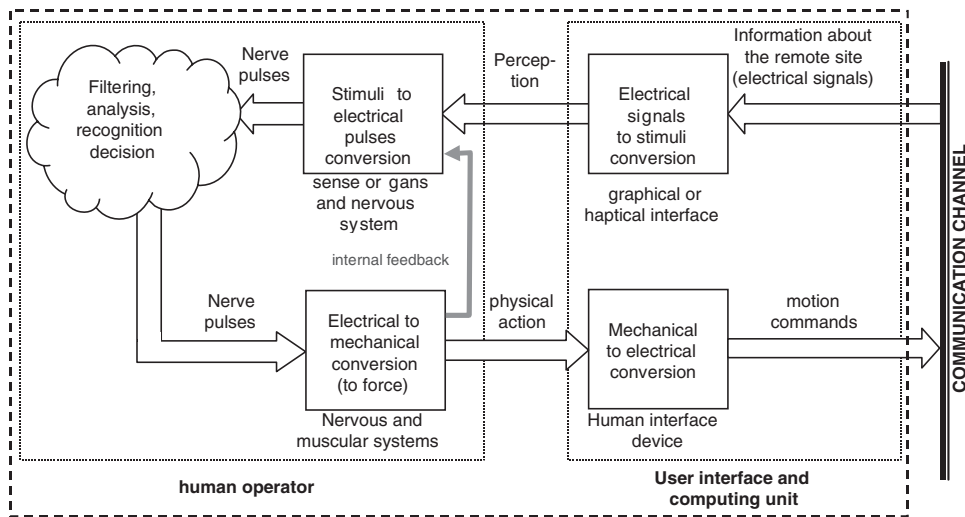


Fig. 2. Diagram of the local site from the point of view of signal conversion. The complex cognitive processes inside the human brain are regarded as a black box. The block-to-block delays were not explicitly included in the diagram.

operator are beyond the scope of this work). The main subject of this work is the study of a method for reducing the negative effects of time delay, such as inaccuracy and risk of collision (and potential damage to the equipment or failure in the execution of the task). The situation worsens when the time delay varies in time. In general terms, the objective of the control system is to cooperate with the human operator to keep the system stable by providing him/her with a good perception of the task while ensuring a collision-free motion.

2.1. Local site

A model for the human operator could have been used, but the cognitive processes are hard to model, hence the authors decided (just like in similar works) to consider him/her as a black box in which decisions are made based on external information. When teleoperating a robotic system, the human operator receives this information about the remote environment in the form of sensorial stimuli such as images, tactile feedback, and numeric displays. Inside the brain, several cognitive processes²⁷ occur based on this information, leading him/her to take decisions that are expressed physically as manipulation actions on the human interface device (HID). This in turn transforms physical actions into electrical signals that can be transmitted through the communication channel toward the remote site.

The remote system interacts with its environment generating new feedback information that is sent back to the operator, thus closing the loop. The authors regard the local site from the point of view of signal conversion (Fig. 2). First, information about the state of the remote system and its environment travels in the form of electrical signals through the communication channel (cable, internet, radio frequency, etc.) and reaches the local site. These signals are converted to human readable information and presented to the human operator in the form of pictures, displayed numbers, sounds, etc. by means of a graphical or haptic interface. Since this information stimulates the sense organs, it is referred to as sensory stimuli. On the other hand, the sense organs convert these sensory stimuli into electrochemical signals

that are transmitted via the nervous system to the brain, where cognitive processes like signal filtering, processing, recognition, and decision occur. Once a decision is taken, the human operator executes an action: commands are sent in the form of electrochemical pulses through the nerves toward the muscular system, where an electrical impulse is converted into a mechanical action. This occurs in form of a contact force applied to the HID, which will, in turn, convert the mechanical forces into electrical signals that will be sent back to the remote site as motion commands. Accordingly to the aforementioned representation, it is obvious that there are several internal communication channels (represented by connecting arrows between the blocks in Fig. 2). Although inherent transport delays exist therein, these are disregarded in the present work, which considers only the communication delay between local and remote site.

2.2. Remote site

The mobile robot model employed in the present work was presented in Aicardi *et al.*,² in which polar coordinates are used to describe the robot's pose. The robot with orientation θ is considered as located at a nonzero distance from a reference frame $\langle ref \rangle$. This frame is on its turn located at the desired robot position x_{ref} (regardless of its desired final orientation). There is also a mobile frame $\langle robot \rangle$ attached to the robot. The variables chosen to define the state of the robot are the distance error ρ and the angular error α (the desired final orientation is not considered). Both errors are measured between the frames $\langle ref \rangle$ and $\langle robot \rangle$ (Fig. 3). The system is then described by the kinematic equation system (1), which considers a fixed reference frame, for example, a desired position/orientation

$$\begin{cases} \dot{\rho} = -v \cos \alpha, \\ \dot{\alpha} = -\omega + v \frac{\sin \alpha}{\rho + \eta}, \end{cases} \quad (1)$$

where v and ω are the linear and angular velocity of the mobile robot, respectively. The constant η has been added to avoid division by zero and has a very small, positive value

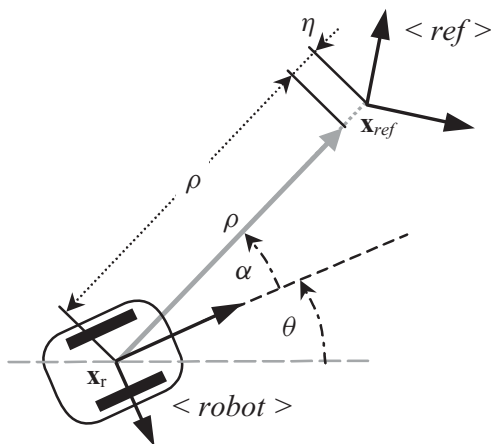


Fig. 3. State of a mobile robot described in polar coordinates with respect to the reference frame $\langle ref \rangle$.

(which can, in turn, cause a small precision loss in the position control).

Since only indoors are analyzed in this work, the environment's representation can be simplified, particularly in case two-dimensional sensors are applied. When using a roof-mounted camera, for example, the environment can be represented by the image plane; when using a laser scanner, it can be represented by a gridmap or a plane. The obstacles in the configuration space can be represented either by polygons, circles, or points (in the case of laser scanners). In this work, only one two-dimensional laser scanner is used as environment capture sensor, so the obstacles can be ideally represented by scan points or, when considering the inaccuracy due to the laser beam expansion with the distance, by circles, whose radii proportionally increase with their distance to the robot.

At the local site, the human operator (which is not modeled here) manipulates a *HID* to generate and issue motion commands in the form of linear and rotational velocity commands that are sent as set points to the remote site.

2.3. Communication channel

The signal flow occurs through the bidirectional communication channel. This can be represented by the sum $h(t)$ of a forward (from the local site to the remote site) time delay $h_1(t)$ and a backward (from the local site to the remote site) time delay $h_2(t)$ as in Eq. (2)

$$h(t) = h_1(t) + h_2(t). \quad (2)$$

In real applications, the communication delay can be varying in time. For this reason, a dependency with t is explicitly expressed here.

3. Proposed Control Scheme for Teleoperation of a Mobile Robot in the Presence of Time-Varying Delay

The authors assume an *a priori* unknown global goal. This is the case in many typical applications like exploration, for example, where the main objective is to perform reconnaissance of the terrain itself and not just to go from a given point A to a destination point B . In fact, inside the operator's head there

is a mental subgoal that mirrors his/her decision, which can change dynamically according to changes in the information about the environment. The idea is to estimate this decision by means of prediction techniques in order to reduce the effects of time delay. It is assumed that the human operator defines directly or indirectly the desired trajectory described by the mobile reference frame $\langle ref \rangle$. The objective of the present work is to design a stable control scheme that assists the human operator by calculating the control actions v and ω based on the commands issued by him so that the robot at the remote site can follow them safely despite the time delay. The operator's commands are modified by the system (if necessary) to guarantee a collision-free motion.

The proposed control scheme is implemented at the remote site, so no special modifications to the conventional local station are needed. At the remote site, the human decision is estimated by means of prediction techniques and represented as a position reference. The reference is sampled and held in order to keep it stable and then passed to a motion planner; this idea was inspired by the well-known technique *move-and-wait*.²³ The motion planner calculates a semioptimal collision-free path from the robot's current position to the desired one; it also guides the robot to the objective along the nodes of the chosen path by means of a stable position controller. The calculated path is semioptimal in the sense that instead of a single solution, a set of candidate solutions is calculated and, according to an optimality criterion, the best one is chosen. Since the motion planner has the direct control of the robot but the human operator gives the changing subgoal continuously, it can be stated that this is a case of partial autonomy. On the other hand, since the operator continuously controls the desired position and therefore the high-level guidance of the overall system, the system is still considered being teleoperated. The reason for using of a motion planner on the remote site of a delayed teleoperation system and not in the operator station is that the sensory information is local to the teleoperated system. Therefore, changes in the environment are detected first by the measuring system at the remote site and later by the local site (after a delay time imposed by the communication between both sites). On the other hand, human intelligence and decision-making skills are incorporated in the system and kept at a supervisory level with the objective of achieving a synergy between human and machine. The proposed teleoperation system is depicted in Fig. 4. The information associated to the obstacles detected by the system is represented by \mathbf{O} . Next, each subsystem of the proposed control scheme is described in detail.

3.1. Motion controller

In order to guarantee a stable behavior of the mobile robot during its motion, a control system designed on the basis of Lyapunov's stability theory is used. The motion controller was previously presented by the authors²⁵ and consists of a nonlinear position controller and an impedance controller. The impedance controller is based on the artificial generation of a fictitious force depending on the distance to the obstacles. The authors use this fictitious force to keep the robot away from the obstacles as a (redundant) collision-avoidance module in addition to the path planner. Figure 5 shows

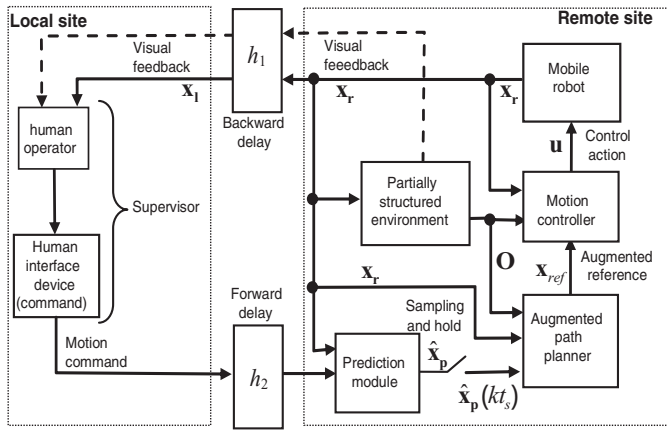


Fig. 4. Proposed control scheme for assisted mobile robot teleoperation in the presence of time delay.

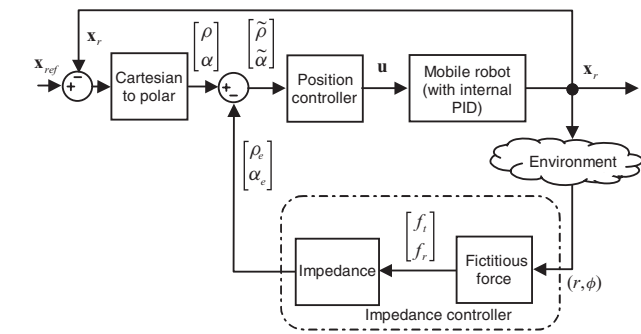


Fig. 5. Position control system used in the teleoperation system. A single obstacle was considered in the diagram.

the control system, which takes a position reference \mathbf{x}_{ref} to define the frame $\langle ref \rangle$ in Eq. (1); this requires a polar-to-Cartesian conversion. The relative positions and angles of the obstacles with respect to the robot are represented by (r, ϕ) .

3.1.1. Cartesian-to-polar conversion. According to the Eq. (1), the system is given in polar coordinates. Since the reference and position frames are defined in Cartesian coordinates, an appropriate coordinate transformation module is needed. The conversion is computed as follows:

$$\begin{aligned} \rho &= \sqrt{(\mathbf{x}_{ref1} - \mathbf{x}_{r1})^2 + (\mathbf{x}_{ref2} - \mathbf{x}_{r2})^2}, \\ \alpha &= \tan^{-1}((\mathbf{x}_{ref2} - \mathbf{x}_{r2}) / (\mathbf{x}_{ref1} - \mathbf{x}_{r1})) - \theta, \end{aligned} \quad (3)$$

where subscripts 1 and 2 are used to denote the components of \mathbf{x}_{ref} in Cartesian coordinates and θ was defined in Fig. 1 as the orientation of the robot. For the sake of simplicity, the desired final orientation angle is assumed zero and so the orientation of \mathbf{x}_{ref} will also be zero.

3.1.2. Nonlinear position controller. The position controlled is represented by Eq. (4). The inputs of the controller are the modified distance error $\tilde{\rho}$ and the angular error $\tilde{\alpha}$. These are calculated between the signals ρ, α and the output of the impedance controller ρ_e, α_e . The equations of the controller are following:

$$\mathbf{u} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} k_e \tilde{\rho} \cos \tilde{\alpha} \\ k_q \tilde{\alpha} + k_e \cos \tilde{\alpha} \sin \tilde{\alpha} \end{bmatrix}, \quad (4)$$

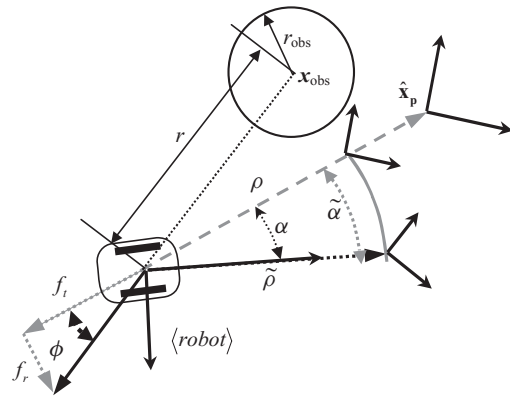


Fig. 6. Relationship between the repulsive force and the distance and angular errors.

where $k_q, k_e > 0$ are the controller parameters. The signals ρ and α are calculated from the augmented reference \mathbf{x}_{ref} and the position vector of the mobile robot $\mathbf{x}_r = [x_r \ y_r \ \theta_r]^T$. The controller establishes the velocity vector \mathbf{u} of the mobile robot making the motion of the robot exponentially stable to achieve a given position goal.²⁴

3.1.3. Impedance controller. As a redundant safety module, an impedance controller²⁸ based on fictitious force was used in the control system. While in teleoperation of robot manipulators, a force results from the physical contact between the robot and the obstacles, in the case of mobile robot teleoperation, a repulsive fictitious force is defined that depends on the distance between the robot and the obstacles. The magnitude of the repulsive fictitious force f generated by an obstacle can be defined as

$$f(t) = k_3 - k_4 r(t), \quad (5)$$

where k_3, k_4 are positive constants such that $k_3 - k_4 r_{max} = 0$ and $k_3 - k_4 r_{min} = 1$. The distance from the robot to the obstacle is denoted by r , which satisfies $r_{min} \leq r \leq r_{max}$, with r_{min} and r_{max} the minimum and maximum robot-obstacle distances, respectively. These define the area where the force acts. The magnitude of $f(t)$ is scaled to the interval $[0,1]$ so that its value can be amplified by the impedance controller. On the other hand, the angle of the fictitious force ϕ is defined as the orientation of the obstacle with respect to the mobile robot (Fig. 6).

The tangential fictitious force and the normal fictitious force are calculated as $f_t = f \cos \phi$ and $f_r = f \sin \phi$, respectively. When multiple obstacles are present, a total force is calculated by summing up all force vectors or using a Kalman filter.

The impedance model is defined as

$$[\rho_e \ \alpha_e] = \mathbf{Z} \mathbf{f}_e; \quad \mathbf{Z} = \begin{bmatrix} K_\rho & 0 \\ 0 & K_\alpha \end{bmatrix}, \quad (6)$$

where $\mathbf{f}_e = [f_t \ f_r]^T$ and $K_\rho, K_\alpha > 0$, which describe the elasticity parameters, f_t is the fictitious force on the robot motion direction, and f_r is the fictitious force on the normal direction to the robot motion.

When the mobile robot navigates interacting with the environment, the state is defined as

$$[\tilde{\rho} \quad \tilde{\alpha}]^T = [\rho \quad \alpha]^T - [\rho_e \quad \alpha_e]^T, \quad (7)$$

where with $\tilde{\rho} \geq 0$ and ρ, α are error signals with regard to a time-varying position reference, in this case $\langle xref \rangle$ (see Fig. 6).

3.2. Predictor

Since in a teleoperation system the motion commands from the human operator can be delayed by the communication channel, the reaction of the robot to these commands can also be delayed, which can affect the transparency of the system. Therefore, the authors proposed in a previous work²⁰ to use a predictor that estimates where the human wants the robot to move to h_2 seconds ahead. This implies that the prediction horizon is determined by the time delay. This estimation the authors called the *human decision* is denoted by $\hat{\mathbf{x}}_p = [\hat{x}_p \quad \hat{y}_p \quad \hat{\theta}_p]^T$. On that model, the estimation is based on the current robot's position and velocity, the delayed velocity command, and the forward time delay. The prediction horizon is limited by using a smooth function, like tanh, which is necessary for boundedness and stability of the system. The estimation is given in Cartesian coordinates so that it can be passed directly to the path planner as the current goal. The estimation of the orientation is compensated by using the difference between the current angular velocity of the mobile robot and the angular velocity that was received as feedback by the operator at the moment of sending the velocity command $\omega(t - h)$. The equation set describing the predictor is given by

$$\begin{aligned} \hat{\theta}_p(t) &= \theta_r(t) + G_\omega[\omega_h(t - h - h_0)]\Delta\omega, \\ \hat{x}_p(t) &= x_r(t) + G_v[v_h(t - h - h_0) \cos \hat{\theta}_p(t)], \\ \hat{y}_p(t) &= y_r(t) + G_v[v_h(t - h - h_0) \sin \hat{\theta}_p(t)], \end{aligned} \quad (8)$$

where $[v_h(t) \quad \omega_h(t)]^T$ represents the velocity command given by the human operator (with $v_h(t)$ linear velocity and $\omega_h(t)$ angular velocity) and the vector $\mathbf{x}_r(t) = [x_r(t) \quad y_r(t) \quad \theta_r(t)]^T$ represents the current position and orientation of the mobile robot in Cartesian coordinates. Furthermore, the bounded gain functions G_v and G_ω are given by $G_v = k_1 \tanh((h_2 + h_0)/k_1)$ and $G_\omega = k_2 \tanh((h_2 + h_0)/k_2)$, respectively, with $k_1, k_2 > 0$; they ensure the boundedness of $\hat{\mathbf{x}}_p$ notwithstanding the magnitude of the time delay. The factor $\Delta\omega = 1 - k_\omega |\omega(t) - \omega(t - h - h_0)|$ helps to compensate the prediction of the orientation, where $k_\omega > 0$ is chosen so that $\Delta\omega$ is confined to the interval $(0, 1]$. The compensation is necessary because of the mismatch between the information the operator uses to send a given motion command and the actual information generated at the remote site at that moment. The factor represents the difference between the current situation of the teleoperated robot regarding its angular velocity and the situation being perceived by the operator at the time instant where he/she issued a motion command. This difference is weighted so that it does not modify the sign of the velocity command but rather its magnitude, thus making this more appropriate to the real

situation at the remote site. The term $\Delta\omega$ confers more reliability to the prediction of the angular position θ by using ω_h , which was generated by the operator. If a strong change occurs in θ after $h_2 + h_0$ time units, the reliability of the prediction is reduced, since the information the operator used to generate the command has changed notably and therefore, it is better to keep the predicted value as constant. On the other side, if the change in ω can be neglected, the reliability of ω_h increases and so does the quality of the prediction.

The total current time delay h was defined in Eq. (2). In order to avoid a null prediction horizon due to a zero time delay, the term h_0 is added so that the prediction falls in a position different to that of the robot. Otherwise, $\langle rob \rangle$ and $\langle ref \rangle$ would be the same with zero delay and the robot would not move. If we assume that v_h and ω_h are bounded, then from Eq. (3), $\hat{\mathbf{x}}_p$ is bounded too. If $\hat{\mathbf{x}}_p$ were taken as the reference of the local controller, \dot{S}_ρ and \dot{S}_α would be calculated by projecting it onto the vector ρ and its perpendicular direction, respectively, to be used in Eq. (1). Instead, the prediction is fed to the path-planning module in order to guarantee collision-free motion.

3.3. Enhanced path planning

In the proposed teleoperation system, the motion of the robot is guided by an estimation of the operator's desired position. Because of the presence of time delay, the quality of this estimation cannot be guaranteed. In fact, it degrades with the increasing of the time delay's instant value. Moreover, since the commands from the operator are based on outdated (delayed) information about the remote site, the desired position (or its estimation) can fall in the obstacle space of the environment. For these reasons, instead of passing the desired position as reference to the motion controller, a reference is generated based on a path calculated by a path planner. The planner permits a collision-free motion of the robot from its actual position to the desired one, since every node in the resulting paths belongs to collision-free. On the other side, the operator acts as a supervisor by indirectly setting waypoints on the motion space while the system actively supports him/her in the execution of the teleoperation task.

The chosen sampling-based path planner is an enhanced RRT.¹⁹ Because of its properties, an RRT is adequate for the purposes of this work. At the first run, the algorithm generates a solution set (the *solution fan*) that goes from the start position to a goal position. If this goal position is not reachable, the paths end as close as possible to it. Hereupon, the path that is most correlated to a straight path between start and goal is chosen, which results in less curvy paths. In the subsequent replanning, a correlation to the previously chosen path is also taken into account, which helps the paths to maintain a certain direction and to avoid oscillations. For this purpose, the two correlation values are weighted and added together in order to obtain a total correlation value. Again, the path with highest correlation is chosen. The chosen path is then output for its further processing in a motion controller. In the present work, the authors propose the inclusion of some improvements to the previous algorithm. First, the inclusion of a waypoint cache,⁷ that is, some random points from the previously selected path are used in the generation of the next

one, and second, the online adjustment of the weights that determine the total correlation. Next, the process is described including formal notation.

3.3.1. Initialization. The current robot position \mathbf{x}_r is set as the start node and the human decision $\hat{\mathbf{x}}_p$ as the goal. The RRT is initialized with the start node as root. The waypoint cache is initialized as the empty set.

3.3.2. Generation of the human proposal. The authors refer to the straight one-segment path defined by the edge $\langle \mathbf{x}_r, \hat{\mathbf{x}}_p \rangle$ as the *human proposal*. According to the nature of $\hat{\mathbf{x}}_p$, the human proposal is not necessarily collision-free; it is only used as a reference the path-selection process to achieve solution paths with predominant straight segments.

3.3.3. Generation of the solution set. The RRT-planner generates the solution fan, by iterating N_k times. The *solution fan*, denoted by Π_k , consists of N_k collision-free paths $\pi_{i,k}$, each of which represented by a list of $M_{i,k}$ two-dimensional points (in this case, the orientation is disregarded and the robot considered as a circle). As additional feature in this work, after the path is selected, some points are selected at random and stored in a waypoint cache. In the subsequent replanning, these points are used in the expansion process. Usually, the algorithm chooses randomly between the goal and a random position of the motion space a target node toward the tree is expanded. When using the waypoint cache, one point of the cache (also selected at random) is considered in the selection of the target node. To each target selection criterion, a probability is assigned so that $p_{goal} + p_{rand} + p_{wayp} = 1.0$.

The idea behind the waypoint cache is that the next generated paths be attracted toward previously found paths. It can be interpreted as a kind of memory that stores a trace of successful solutions.

3.3.4. Selection of the best matching path. The criterion for the selection of one path among the N_k solutions is the combined value c_T of two independently calculated correlations. The chosen path π_k^* is the one with the highest c_T . Both correlations are calculated as the absolute value of the weighted Pearson coefficient.¹ The weights are chosen so that more relevance is given to the nodes nearest to the start node. Let $c_{H,i,k}$ and $c_{L,i,k}$ be the correlation with the human proposal and the correlation with the last chosen path, respectively, at the instant k . Then, the combination is defined as

$$c_T = K_{H,k}c_{H,i,k} + (1 - K_{H,k})c_{L,i,k}, \tag{9}$$

where $K_{H,k} \in R \cap [0, 1]$ is a variable gain that gives more or less significance to the human proposal. To simplify the notation, the index k will be dropped out. If a small value is assigned to K_H , the subsequent paths will keep one direction, which will result in a motion with such biasing that the operator will barely be able to make the robot turn. On the other side, a value near one can lead to oscillations; thus, K_H establishes a tradeoff between transparency and stability.

In the present work, the parameter K_H is adjusted online to improve the performance of the system. More precisely, the value will be set proportionally to the total time delay h_k

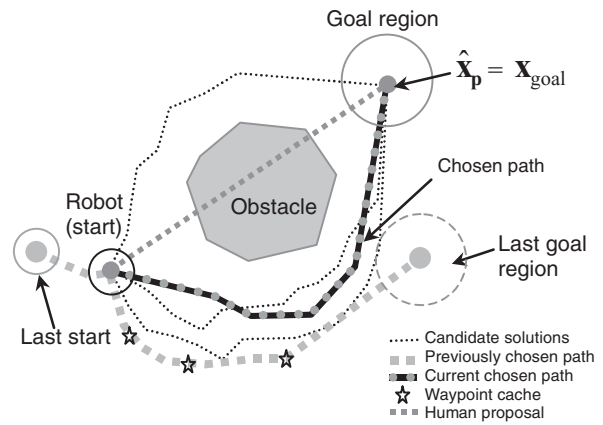


Fig. 7. Example solution set generated by the extended RRT.

at instant k ($h_k \in R \cap [0, h_{max}]$), as

$$K_{H,k} = mh_k + K_{Hmin}, \tag{10}$$

where $m = (K_{Hmax} + K_{Hmin}) / (h_{2max} + h_{2min})$ with K_{Hmax} , K_{Hmin} the desired values for K_H corresponding to the expected maximum (h_{2max}) and minimum (h_{2min}) values of h .

The solution fan and the chosen path for a typical complete run of the planner at some instant k are shown in Fig. 7. The human proposal and the last chosen path are also included for the sake of illustration.

3.3.5. Update of the waypoint cache. After the best path is calculated, some of its nodes are chosen at random and stored in the waypoint cache. The number of nodes can also be randomly chosen based on the length of the chosen path.

3.3.6. Augmented reference. The chosen path at the current instant k has M_k^* collision-free nodes starting at the mobile robot position and ending as close as possible to the goal region centered in $\hat{\mathbf{x}}_p$. In the time slots between successive executions of the planner, a motion controller steers the robot along the chosen path. For this, each node $\mathbf{x}_{i,k}^{\pi^*}$ is passed successively to the stable controller, which is capable of driving the mobile robot to reach a vicinity of the current node rapidly. Then, the current node (goal) is changed by the next node of the chosen path. The components of $\mathbf{x}_{i,k}^{\pi^*}$ define the position of the reference frame $\langle ref \rangle$. In Fig. 4, \mathbf{x}_{ref} denotes the two-dimensional vector formed by the coordinates of $\langle ref \rangle$, where for simplicity the index k is dropped. This vector is called the *augmented reference*.

3.3.7. Replanning. The planner is not necessarily called every time instant k . Instead, a replanning is performed in one of the following cases:

- (a) After updating sensor information from the environment the current path enters in collision with at least an obstacle.
- (b) The goal coordinates change considerably with respect to the instant when the path was selected.

Those nodes already traveled are disregarded in the updating of the waypoint cache.

3.4. Sampling of the human decision

The main problem with time delay due to communication is the accumulation of commands. In the presence of time delay, as long as the operator does not receive the sensory feedback from the remote site he/she keeps acting on the HID until a reaction of the robot can be perceived. By that time, the motion commands have been accumulated due to the forward delay so that when these reach the remote site, the robot performs an abrupt motion but, because of the backward delay, this behavior cannot be noticed in due time so that the operator cannot react to correct this but instead keeps teleoperating. By the time the operator discovers the wrong behavior it is too late. The problem can go on in a vicious circle and so degrading the systems stability and risking the systems safety. To cope with this problem, Sheridan proposed the well-known strategy *move-and-wait*. When using that strategy, the operator performs a short-duration motion, then he/she stops sending commands and waits until reaction from the system is noticed to send the next command. The disadvantage of this strategy is its negative influence on the quality of the teleoperation because the operator has either to drive the robot slowly or to send repeatedly stop commands. The proposed scheme samples and holds the estimated human decision with a variable sampling time that is set proportional to the instant value of the total time delay. The sampling is defined by $\hat{\mathbf{x}}_p(k t_s) = \hat{\mathbf{x}}_p \zeta(t - k t_s)$ with $k = 0, 1, 2, 3, \dots$, ζ representing a Dyrac delta-function. The proposal establishes $t_s := k_s h + h_0$, where $k_s > 0$ and $h_0 > 0$ are set empirically.

This sampling and hold emulates a move-and-wait strategy but performs internally the work so that the operator does not have to explicitly send additional stop commands, which, in turn, guarantees a fluent motion of the mobile robot.

Remark: Let us assume that the human operator behaves like a bounded-input bounded-output system. On the other hand, the closed loop including the robot and motion controller is exponentially stable. Considering both features, the prediction module also has a bounded output. In addition, the augmented path planner generates a collision-free path pushing reference points to the stable motion controller in a discrete way as the robot achieves a vicinity of the current reference point of the path. Therefore, the proposed teleoperation system has input–output stability. This analysis does not assure some convergence-type of the system to the user’s intentions (the goal is unknown *a priori*) neither some performance level. Therefore, an experimental analysis will be used to test the proposed control scheme.

4. Experiments and Results

In order to validate the theoretical performance of the proposed system, some experiments were carried out in an indoor environment. The authors used a Pioneer 3DX mobile robot (www.activmedia.com) with a mounted SICK LMS200 laser scanner. The scanner has a visibility of 180° and a distance range of 80 m. The operator used a Logitech Speed Racer steering wheel with gas pedal to issue the motion commands. The maximal linear velocity was set at 0.4 m/s and the maximal angular velocity was set at 0.35

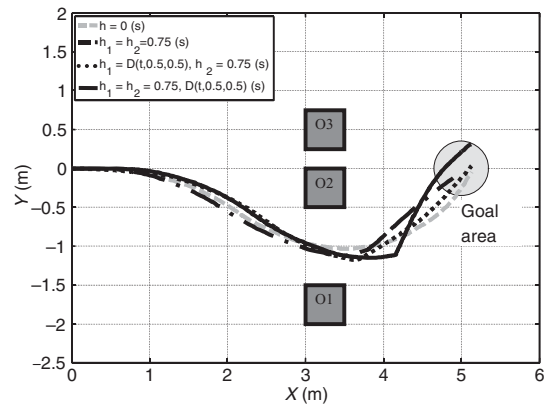


Fig. 8. Paths followed by the mobile robot during experiment one: Assisted teleoperation under the presence of different time delays.

rad/s. The software ran under windows XP on a Pentium III @ 1.3GHz. The robot and the operator were connected by an intranet network using the IP/UDP protocol with a baud rate of 8 Kbytes/s computed from the transmitted data quantity and sampling time. We assume that the communication channel is capable of handling such information flow. The physical distance between operator and robot varied between 17 and 25 m approximately during the experiments. Visual information from the laser scanner is sent back to the operator by means of a graphical user interface, which builds on-line a two-dimensional map of the remote environment. During the experiments, the time delay added by the Intranet is very small, so artificial delay by means of FIFO buffers was added. The internal proportional–integral–derivative (PID) velocity controller on board of the mobile robot was running in a faster loop than the general control loop, which had a sampling time of 0.1 s. The parameters used for the controllers were $k_3 = 1.5N$, $k_4 = 0.75N/m$, $k_e = 0.6s^{-1}$, $k_q = 0.75rad/s$, $k_\alpha = 1.5m/N$, and $k_\omega = 0.2rad/N$. The parameters for the predictor were set as $k_1 = k_2 = 1.8s$ and $k_\omega = 1.3s/rad$. All position coordinates are given in meters.

An auxiliary triangle function was defined to facilitate the notation for the time-delay signals used. From now on, the function $D(\cdot)$, given by Eq. (11) will describe a triangle function in terms of its amplitude A and its slope m (rising equals falling). The function is given by

$$D(t, A, m) = f_{\text{triangle}}(t, A, A/m), \tag{11}$$

where the periodical signal f_{triangle} is defined as

$$f_{\text{triangle}}(t, A_{tri}, T_{tri}) = \begin{cases} t & \text{if } t \in [nT, nT + T/2) \\ A_{tri} - t & \text{if } t \in (nT + T/2, (n+1)T) \end{cases}, \tag{12}$$

$$\forall n \in N.$$

In the first experiment, the objective was to show the influence of the time delay on the performance of the proposed system. The human operator had to remotely drive the robot from the start position at (0,0) to the circular goal region centered at (5.0, 0.0) with radius 0.3 m. Three squared obstacles were placed in the environment as shown in Fig. 8 so that the operator had to perform two main turns: one turn to the right

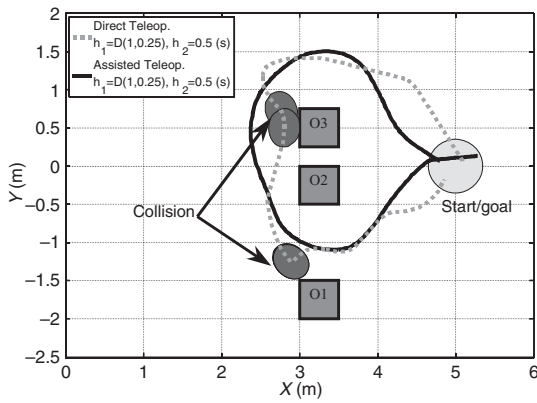


Fig. 9. Trajectories followed by the robot during experiment two. While operating directly (without assistance) the robot collided at three positions (marked with ovals) whereas when using assisted teleoperation no collisions occurred.

to enter the corridor region between obstacles O1 and O2 and a second turn to the left in order to reach the goal region. The experiment was repeated four times for different time delays, using the proposed control scheme in all cases. The maximum total delay was 2 s and the waveforms for the time delay were chosen so that no abrupt changes in the signals were present.

In a second experiment using the same setup as in the first one, the operator had to drive the robot in a closed loop starting and ending at (5,0) and passing between obstacles O1 and O2. In the first part of this experiment, the system was teleoperated directly and in the second part, the proposed assistance system was used. In both cases, a constant forward delay of 0.5 s and a variable backward delay of $D(1,0.25)$ s were used. Figure 9 shows the trajectories followed by the robot in both cases.

As it can be seen in the figure, by using the assisted scheme, the task could be successfully executed with no collisions. In the case of direct teleoperation, the robot collided approximately at positions (2.8, -1.25), (2.7, 0.5), and (2.65, 0.75).

In a more complex experiment, a maze-like environment was used. Figure 10 shows the two-dimensional map built online and the trajectories followed by the robot teleoperated by a user. The forward communication delay was 0.4 s and the backward delay was 0.3 s, both constant. In the first phase of this experiment, the robot was teleoperated directly, while in the second phase, the operator used the proposed assisted teleoperation system. It can be observed that in the first phase of the experiment a poor performance was reached; moreover, the robot collided with the walls approximately at positions (1.0, 1.5) and (3, 1.5), marked with ovals. At the latter position, the operator even had to drive backward to correct the trajectory. With the assisted teleoperation, no collisions happened and a smoother continuous motion was achieved.

Figure 11 shows the velocity profiles for the teleoperated robot. It can be observed that approximately at the time instants 65 s and 80 s, the operator using direct teleoperation was forced to drive backward after collisions with the walls of the environment. Furthermore, the velocity profiles

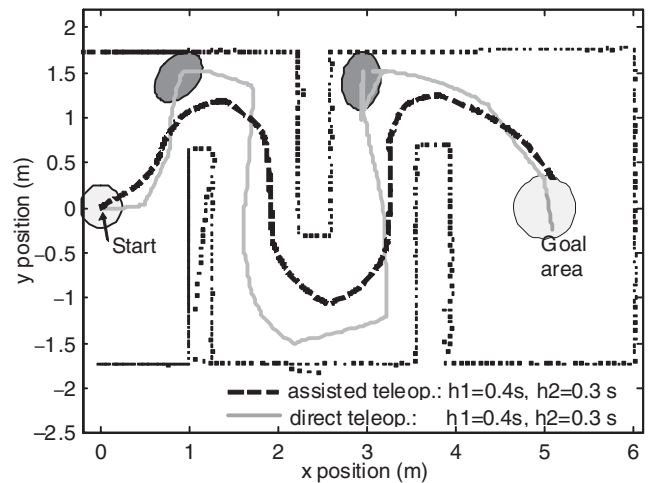


Fig. 10. Trajectories followed by the robot in a maze-like environment (experiment 3), which demanded a high dexterity from the operator (collision poses marked with ovals).

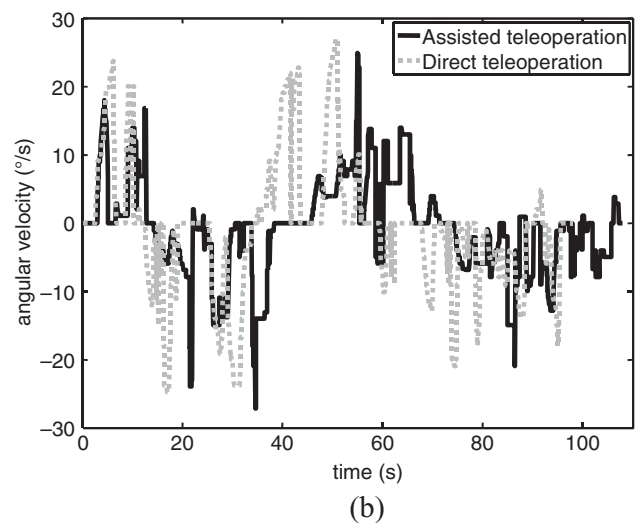
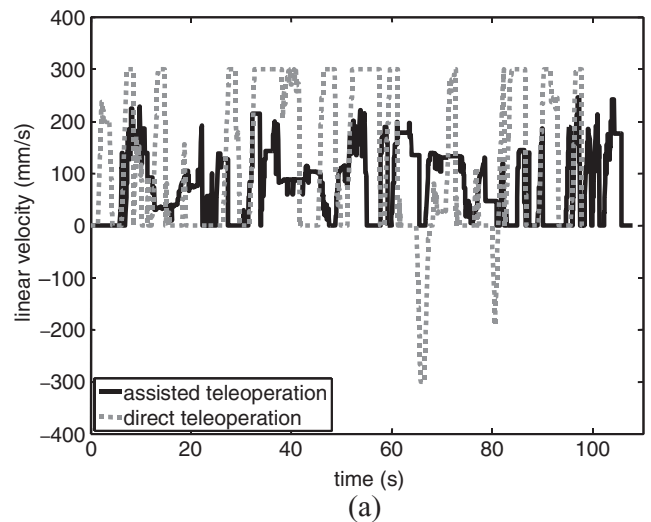


Fig. 11. Velocity profiles for the third experiment: (a) linear velocity and (b) angular velocity.

Table I. Performing indexes for the teleoperation system tested for 10 operators during the four experiments. All times are given in seconds.

Time delay	Assisted modus?	Execution time (mean)	Execution time (σ)	Relative correlation (mean)	Relative correlation (σ)	Collisions
$h_1 = 0.0, h_2 = 0.0$	No	38.86	12.84	1.00	0.00	Yes
$h_1 = 0.5, h_2 = 0.5$	No	63.43	26.34	0.91	0.08	Yes
$h_1 = 0.0, h_2 = 0.0$	Yes	45.57	19.36	0.75	0.08	No
$h_1 = 0.5, h_2 = 0.5$	Yes	58.14	22.06	0.90	0.14	No

show how the assisted teleoperation adapted the robot's velocity to better dominate the turns. In general, the assisted teleoperation required less angular velocity because the system was aware of the obstacles' position. This also remarks the importance of using path planning to assist the operator.

Next, a series of teleoperation experiments was performed with 10 operators ranging between 25 and 45 years, who were trained to operate the system. The training consisted in first asking the operators to teleoperate a simulation platform (ARIA's MobileSim) within a free environment using direct teleoperation without time delay. After that they practiced by executing the experiment's main task of bringing the simulated robot from point (0,0) to the circular objective region at (5,0) with a radius of 0.35 m and passing between two obstacles. The task was repeated 10 times for each one of the operators. The experiments consisted in repeating the main task used for training using and not using the proposed assisted teleoperation system but driving a real Pioneer 3DX mobile robot. Each case was tested with and without time delay, yielding four experiments. Table I shows the mean values and the standard deviations of the execution times required to complete the task. From the information on the table, it can be observed how the proposed assistance system helps improving the execution time of the teleoperation task in the presence of time delay. The table also shows the mean values and standard deviations of the "relative correlations," where the term "relative correlation" refers in this work to the Pearson¹ correlation between the curve described by the robot when directly teleoperated without time delay and that obtained during the remaining three teleoperation experiments. The table also reveals that in the presence of time delay, compared to direct teleoperation by using the proposed scheme, the system will show a more similar behavior to that achieved under favorable conditions.

It is important remark that when using direct teleoperation, sometimes collisions occurred; even when no time delay was present (this was due to the lack of dexterity or errors by the operators). On the other hand, using the proposed assisted teleoperation, a collision-free motion was reached by all operators.

Finally, to evaluate the effects of the emulated *move-and-wait*, an additional experiment was performed. By using the same setup as in experiments 1 and 2, the operator had to drive the robot from position (0,0) to the goal region centered at (5,0) in the presence of a varying forward time delay of $D(t, 0.5, 0.5)$ s and constant backward time delay of 0.5 s. The experiment was divided into three tasks, one using direct teleoperation, one using assisted

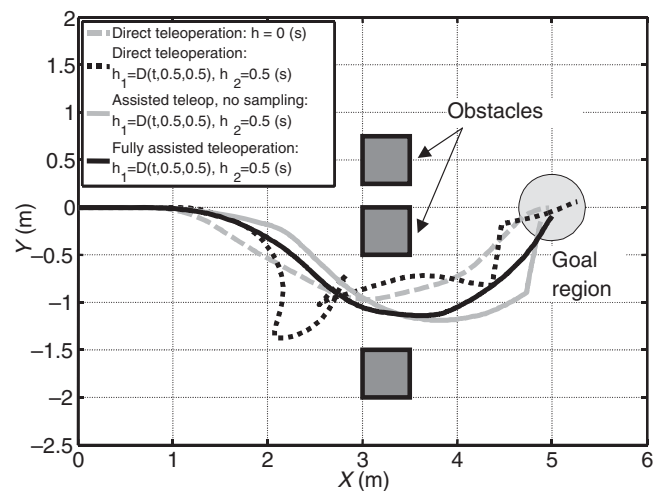


Fig. 12. Trajectories of the robot during experiment 4. The only use of the assistance function (prediction/planning with no command sampling) already produces a positive effect in the security of the teleoperated system. The emulated "move-and-wait" helps improving the performance of the system, which is reflected in a smoother curve.

teleoperation, and the last one with assisted teleoperation but deactivating the command-sampling module. One additional task was performed with direct teleoperation and no time delay for reference. The resulting paths can be seen in Fig. 12, where the negative effects of the time delay, present in the direct teleoperation, are drastically reduced by the inclusion of the assistance module. When using the complete assisted teleoperation, the trajectory becomes smoother. In addition, the fully assisted teleoperation avoids a possible accumulation of undesired commands. For example, Fig. 12 shows that the robot with fully assisted teleoperation turns right and later to left earlier than it with assisted teleoperation since the accumulation of commands in straight line is reduced.

5. Conclusions

In this paper, an enhanced teleoperation system for time-delayed mobile robots is presented. Two enhancements were incorporated in the path-planning module: the use of a waypoint cache and the online tuning of the correlation weights. The system is applied to teleoperation of a mobile robot with variable time delay considering the goal unknown and the environment partially structured. The strategy uses a path planner, a predictor with variable sampling time and a motion controller, all of them situated on the remote site

in order to help the user driving a mobile robot in the presence of time delay. Experimental results showed the good performance of the system with different time delays. They also illustrate the benefits of assisting the operator by the use of path planning and prediction techniques. The experimental results have shown a stable behavior of the mobile robot for different time delays when the control scheme is used. In this work, a successful integration of the human and a control system was achieved. As a future work, the system could be tested in a telemanipulation system. Another possible enhancement could be its adaptation to outdoor systems and integration with three-dimensional environment acquisition sensors.

Acknowledgments

This work was partially financed by the German Academic Exchange Service DAAD and the National Council for Scientific Research CONICET (Argentina).

References

1. J. Aldrich, "Correlations genuine and spurious in pearson and yule," *Stat. Sci.* **10**(4), 364–376 (1995).
2. M. Aicardi, G. Casalino, A. Bicchi and A. Balestrino, "Closed loop steering of unicycle-like vehicles via Lyapunov techniques," *IEEE Robot. Autom. Mag.* **2**, 27–35 (1995).
3. R. J. Anderson and M. Spong, "Bilateral control of teleoperators with time delay," *IEEE Trans. Autom. Control* **34**(5), 494–501 (1989).
4. A. K. Bejczy, W. S. Kim and S. C. Venema, "The Phantom Robot: Predictive Displays for Teleoperation with Time Delay," *Proceedings of the IEEE International Conference on Robotics and Automation*, Cincinnati, OH, USA (1990) pp. 546–551.
5. K. Brady and T. J. Tarn, "Internet-Based Teleoperation," *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea (2000) pp. 644–649.
6. I. Elhadj, Ni. Xi, W. K. Fung, Y. H. Liu, Y. Hasegawa and T. Fukuda, "Supermedia-enhanced internet-based telerobotics," *Proc. IEEE.* **91**(3), 396–421 (2003).
7. D. Ferguson, N. Kalra and A. Stentz, "Replanning with RRTs," *Proceedings of the IEEE International Conference on Robotics and Automation*, Orlando, FL, USA (2006) pp. 1243–1248.
8. J. Funda and R. P. Paul, "Teleprogramming: Toward delay-invariant remote manipulation," *Presence: Teleop. Virt. Environ.* **1**(1), 29–44 (1992).
9. M. Hernando and E. Gamba, "A Robot Teleprogramming Architecture," *Proceedings of the International Conference on Advanced Intelligent Mechatronics*, Kobe, Japan (2003) pp. 1113–1118.
10. P. F. Hokayem and M. W. Spong, "Bilateral tele-operation: An historical survey," *Autom.* **42**(12), 2035–2057 (2006).
11. L. E. Kavraki, P. Svetska, J. C. Latombe and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.* **12**(4), 566–580 (1996).
12. J. Kikuchi, K. Takeo and K. Kosuge, "Teleoperation System via Computer Network for Dynamic Environment," *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium (1998) pp. 3534–3539.
13. W. Kim, B. Hannaford and A. Bejczy, "Force reflection and shared compliant control in operating telemanipulators with time delay," *IEEE Trans. Robot. Autom.* **8**(2), 176–185 (1992).
14. S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," Technical Report No., 98–11. Computer Science Department, Iowa State University (1998).
15. S. M. LaValle, *Planning Algorithms* (Cambridge University Press, Cambridge, MA, USA, 2006).
16. D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Trans. Robot. Autom.* **9**(5), 624–637 (1993).
17. D. J. Lee, O. Martinez-Palafox and M. W. Spong, "Bilateral Teleoperation of a Wheeled Mobile Robot over Delayed Communication Networks," *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, FL, USA (2006) pp. 3298–3303.
18. G. Niemeyer and J. J. E. Slotine, "Stable adaptive teleoperation," *IEEE J. Ocean. Eng.* **16**(1), 152–162 (1991).
19. J. Nieto, E. Slawiński, V. Mut and B. Wagner, "Online Path Planning Based on Rapidly-Exploring Random Trees," *Proceedings of IEEE, International Conference on Industrial Technology*, Valparaiso, Chile (2010) pp. 1431–1436.
20. J. Nieto, E. Slawiński, V. Mut and B. Wagner, "Mobile Robot Teleoperation Augmented with Prediction and Path-Planning," *Proceedings of the International Symposium on Analysis, Design and Evaluation of Human-Machine Systems (IFAC-HMS '10)*, Valenciennes, France (Aug. 31–Sep. 3, 2010).
21. J. H. Park and H. C. Cho, "Sliding-Mode Control of Bilateral Teleoperation Systems with Force-Reflection on the Internet," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Takamatsu, Japan (2000) pp. 1187–1192.
22. T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control* (The MIT Press, Cambridge, MA, USA, 1992).
23. T. B. Sheridan, "Teleoperation, telerobotics and telepresence: A progress report," *Control Eng. Pract.* **3**(2), 205–214 (1995).
24. E. Slawiński, V. Mut and J. Postigo, "Teleoperation of mobile robots with time-varying delay," *IEEE Trans. Robot.* **23**(5), 1071–1082 (2007).
25. E. Slawiński and V. Mut, "Control using prediction for teleoperation of mobile robots," *Proceedings of the IEEE International Conference on Mechatronics and Automation*, Harbin, China (2007) pp.1172–1178.
26. E. Slawiński and V. Mut, "Control scheme including prediction and augmented reality for teleoperation of mobile robots," *Robotica* **28**(1), 11–22 (2010).
27. M. Staal, "Stress, Cognition, and Human Performance: A Literature Review and Conceptual Framework," NASA TM 2004-212824 [Online] 2004, http://human-factors.arc.nasa.gov/flightcognition/Publications/IH_054_Staal.pdf. (accessed Jul 8, 2010).
28. R. Carelli, H. Secchi and V. Mut, "Algorithms for stable control of mobile robots with obstacle avoidance," *Latin Am. Appl. Res.* **29**(2–3), 191–196 (1999).