2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

# Control of a Teleoperated Nanomanipulator with Time Delay under Direct Vision Feedback

Oliver Tonet*, Martina Marinelli, Giuseppe Megali,
Arne Sieber, Pietro Valdastri, Arianna Menciassi, and Paolo Dario

*Abstract*— **Remote manipulation tasks in the small scale can often not be performed autonomously, due to the unstructured nature of the environments and the limited capabilities of sensor and localization technologies. For these tasks, teleoperated systems are used, in which the human operator is integral part of the control. In time-delayed teleoperation, the operator gradually adopts discrete control strategies, such as 'move-and-wait'. In this paper, we present and compare three different control strategies for driving a nanomanipulation system with direct vision feedback. Two strategies are based on a fixed step size to move the manipulator, while the third uses a variable step size. The strategies are compared on a 2-D fine positioning task. Experimental results are in agreement with Fitts' law and show that the third strategy, besides allowing movements of size ranging across several orders of magnitude, also allows to complete the fine positioning task in less time. The control strategies can be used in general to control vision-guided teleoperation systems affected by time delay.**

## I. INTRODUCTION

Remote manipulation systems are used to perform tasks at locations where direct human intervention is impossible or hazardous. These manipulation tasks, especially in the small scale, can often not be performed autonomously, due to the unstructured nature of the environments and tasks, the limited capabilities of remote sensing and localization technologies, and the dexterity required for performing some operations. For performing these tasks, teleoperated systems, in which the human operator is integral part of the control, are most often used. In a teleoperated system the human operator uses a master system to carry out a task at a remote location, by means of a slave robot. One of the important trends in robotics is the development of systems for handling and interaction at the small (micro and nano) scale [1], [2], [3], [4].

In many teleoperation and telerobotic systems there is an unavoidable delay in time imposed between the operator's actions and the corresponding feedback. In teleoperation in an unstructured environment, with no model of the interaction being carried out that would allow anticipation of the results of the operator's action, this delay is the time needed to sense the motion at the master unit, to send the command to the slave manipulator, to actually execute the motion and to send the achieved position back to the master. Delay is determined by the bandwidth and latency in the communication between master and slave, by the inertia and speed of the actuators, and sometimes also by the time required to measure the new position of the slave unit.

Most of the previous work in telerobotics has discussed control methods in the presence of substantial time delay, focusing on haptic perception of soft tissues in complex teleoperation systems with force sensors on the slave unit. Studies on the effects of transmission delay in teleoperated positioning tasks without haptics are more common in the field of user interface design [5], [6] and are especially relevant for space applications [7], [8].

It has been shown that, at lower time delays, most of the time is spent with the control of motion and hence the control appears to be fairly continuous. At higher values of delay, the operators change to a discrete mode of control, executing a movement and waiting to receive feedback from the machine about the results of the action before executing a new movement [9], [10]. This control mode has been called 'move-and-wait'.

We are developing a telemanipulation system based on a nanomanipulator, a three-axial force sensor [11], and a haptic interface [12]. We aim at obtaining a high positioning precision while maintaining the system sufficiently responsive for a smooth interaction with the operator. In this paper we assess the performance of several position control strategies by comparing the time required by operators to complete a repetitive 2-D precise positioning task.

A related method has been used by Dubey et al. [13] on a 1-D task, based on variable velocity mapping. However their system does not suffer from time delays and the position sensors in the slave system allow to maintain coupling of the positions of the master and slave units.

## II. SYSTEM DESCRIPTION

The teleoperation system, depicted in Fig. 1, is composed of a haptic interface (Phantom Premium 1.0, SensAble Technologies, Inc., MA, USA) as master unit and a 3 degrees of freedom (DoF) piezoelectric micromanipulator (MM3A Manipulator, Kleindiek GmbH, Germany) as slave unit.

The micromanipulator has two rotational and one linear DoF and allows stepper motor-like movements with a resolution in the sub-nanometer range; it is therefore also

* Corresponding author: oly@sssup.it

This work has been supported by the FIRB-2001 Project *ApprEndo* (no. RBNE013TYM) and by *EndoCAS*, the Center of Excellence for Computer-Assisted Surgery (COFINLAB-2001 no. CLAB01PALK), both funded by MIUR, the Italian Ministry of Education, University and Research, and by Fondazione Cassa di Risparmio di Pisa, in the framework of the *microSURF* project.

O. Tonet, M. Marinelli, G. Megali, P. Valdastri, A. Menciassi and P. Dario are with CRIM Lab, Scuola Superiore Sant'Anna, Pisa, Italy. O. Tonet, G. Megali, and P. Dario are also with EndoCAS Center for Computer-Assisted Surgery, University of Pisa, Italy. A. Sieber is with ARC Seibersdorf Research GmbH Seibersdorf, Austria

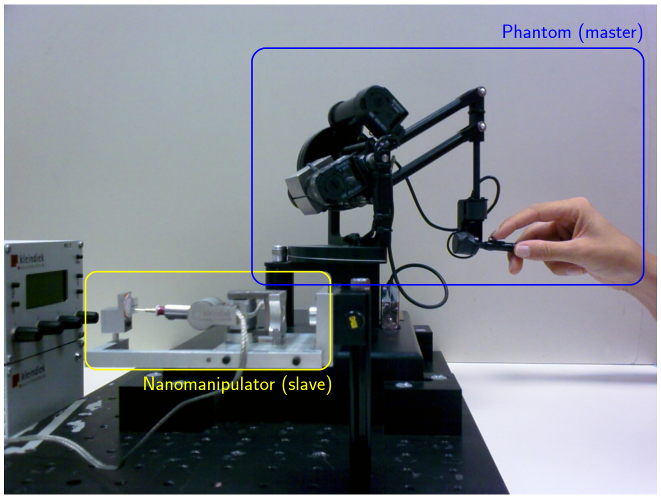

Fig. 1. The teleoperated system, highlighting the master and slave units.

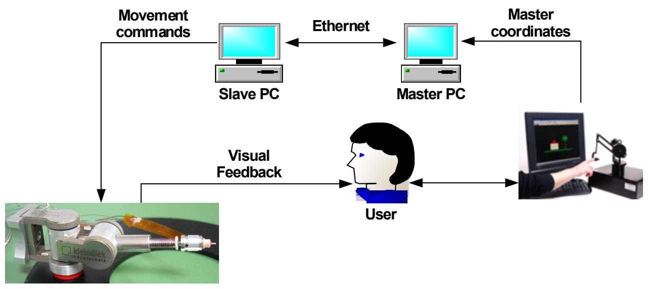

Fig. 2. The architecture of the teleoperated system.

called nanomanipulator. Other specifications include high robustness and stability, no drift, and zero backlash. The working range of the manipulator is 100 cm$^3$. The actuators, called Nanomotors® (Klocke Nanotechnik, Germany), can be driven in two modes: fine mode, for movements as small as 0.25 nm, and coarse mode, for bigger steps up to the mm range. By changing the *speed* parameter, the single step size can be varied by a factor of $2^{13}$, i.e. from 0.25 nm to about 2 $\mu$m for the linear DoF and from 5 nm to about 40 $\mu$m for the rotational DoF. By varying two additional parameters, *wave amplitude* and *wave frequency*, step sizes and movement speeds can be further varied in coarse mode.

The nanomanipulator can be driven by means of rotating knobs placed on a "joy-cube" interface, or by means of two joysticks on a joypad. In both cases the position control is incremental, as with joysticks, and not direct, as in master-slave systems. To achieve direct positioning of the manipulator end-effector, we integrated the Phantom haptic interface in the system and send movement commands to the serial port interface of the manipulator controller via the RS-232 protocol. The baud rate is fixed at 19200 bps.

The architecture of the teleoperation system is depicted in Fig. 2. The master and slave units are interfaced to two separate personal computers (PCs) with Microsoft Windows operating system: the master PC is connected to the Phantom by means of a PCI board and runs a C++ program compiled using the GHOST 2.1 haptics library (SensAble Technologies, Inc.). The slave PC is connected to the MM3A manipulator through the serial port and runs a C++ program with a graphic user interface written using the QT 3.2 GUI library (TrollTech ASA, Norway) for changing the parameters of the system and logging data for the experiments. Intercommunication between the PCs is performed by means of TCP/IP client/server modules based on MFC CSockets over Ethernet connection.

The master workspace is scaled down by a user-definable factor to the working range of the slave manipulator. To map the cartesian space of the manipulator workspace to its joint parameters, forward and inverse kinematics of the MM3A open kinematic chain have been computed. Since the nanomanipulator has no integrated position encoders, it is not possible to measure the position of the manipulator end-effector directly. Rather, coupling the slave position to the master position is an incremental process: first the displacement of the master, with respect to the previous movement, is measured; the displacement vector is downscaled to the slave workspace and the variation of the joint parameters of the slave is computed by means of the inverse kinematics; the movement commands that will achieve the desired variation of joint parameters are sent to the slave controller and the loop starts over. At the beginning, both the master and slave units are initialized at the zero position of their cartesian space. To determine the relation between movement commands and the corresponding variations of joint parameters, we have measured calibration curves. These curves are linear over a range of motion of several orders of magnitude. However, linearity is not perfect, which tends to introduce small errors in the coupling of the master and slave positions. These errors accumulate over time and introduce a position drift. Moreover, the Nanomotor® uses a stick-slip principle to behave like a stepper motor. Yet, the displacement step is not constant, but depends on external forces applied to the manipulator and environmental variables like temperature and humidity [14]. In our system, measurement of exact calibration curves is not needed, since we use direct vision as position feedback and these errors are compensated by the operator. Automatic position control would not be feasible.

Movements of the Nanomotor® are generated by a sequence of pulses. At a given *speed*, each *movement command* will send a series of those pulses to the actuators, e.g. the `coarse 60` movement command send 60 sequences. In the following, the number of sequences in a movement command, e.g. `60`, will be called *units of movement* and the resulting displacement will be called *step amplitude*. The total time spent $t$ will be given by the total movement time $t_m$, i.e. the total displacement $s$ divided by the movement speed $v$, plus the time required to receive the command from the serial port $t_c$. It is necessary to wait for a movement command to be completed before sending another command to the manipulator, otherwise the command will be ignored. Therefore the *minimum waiting time* $t_w$ between two commands is given by:

$$t_w \geq t = t_m + t_c = \frac{s}{v} + t_c \qquad (1)$$

## III. CONTROL STRATEGIES

Since both the serial port communication and the waiting times associated to the discrete nature of the manipulator displacements contribute to movement latency, we have developed several control strategies aiming at allowing users to complete the pointing task in the least time. Keeping the slave position coupled to the position of the master therefore requires some time, and the strategies differ in how the slave follows the movements of the master and on how the controller deals with movements of the master while the slave is still moving toward a previously defined position.

### A. Strategy A

In *Strategy A*, the step amplitude of the slave is constant during the whole task, i.e. the number of movement units of all commands is constant. To keep the coupling between the positions of the master (${}^m\mathbf{p}$) and the slave (${}^s\mathbf{p}$), the position of the master is measured and mapped to the slave reference frame; the number of steps required to perform the displacement from the *initial position* (${}^s\mathbf{p}_i$), i.e. the position reached in the previous movement, to the newly computed *final position* (${}^s\mathbf{p}_f$), along each DoF, is computed by means of inverse kinematics. A software loop then sends all the movement commands to the slave, with a waiting time $t_w$ after each command. The commands on the 3 DoF are alternated, proportionally to the remaining distance along each DoF, in order to keep the trajectory as linear as possible, resulting in a "staircase" motion pattern. At the end of the loop, the *effective final position* of the slave ${}^s\mathbf{p}_{f,\mathit{eff}}$ is computed using forward kinematics and mapped to the master reference frame: the master displacement for the next movement will be measured from this point (${}^m\mathbf{p}_{f,\mathit{eff}}$). This latter computation is necessary because the number of steps along each DoF is rounded to the nearest unit, thus the resolution of the slave movement is determined by the step amplitude: a smaller amplitude increases the resolution but requires more movement commands to achieve a given displacement, slowing down the system. This results in a speed/accuracy trade-off.

Three step amplitudes were chosen for strategy A, corresponding to 100, 50 and 25 units of movements of the manipulator (we called these *profiles* 1, 2, and 3, respectively). Table I shows the corresponding displacements along the three dofs of the manipulator (labeled A, B, and C). Also, the minimum and maximum time $t_w$ needed to perform a single movement is shown in the last column. The times have been measured experimentally. In implementing the strategy, we decided to wait the maximum time: as stated, the slave has no command queue and ignores commands received while it is still executing a movement.

### B. Strategy B

Control strategy B differs from strategy A in that it continuously monitors the position of the master interface and correspondingly updates the final point of the slave movement to match the current position of the master.

TABLE I
PROFILES USED IN CONTROL STRATEGIES A AND B.

| Profile | Command | Step (rad) | Step (mm) | $t_w$ (ms) |
|---|---|---|---|---|
| Profile 1 | coarse a +100 | 0.014 | 0.7 | 68–73 |
| | coarse a -100 | 0.014 | 0.7 | |
| | coarse b +100 | 0.012 | 0.4 | |
| | coarse b -100 | 0.0104 | 0.4 | |
| | coarse c +100 | | 0.9 | |
| | coarse c -100 | | 0.9 | |
| Profile 2 | coarse a +50 | 0.007 | 0.35 | 48–53 |
| | coarse a -50 | 0.007 | 0.35 | |
| | coarse b +50 | 0.006 | 0.2 | |
| | coarse b -50 | 0.005 | 0.2 | |
| | coarse c +50 | | 0.4 | |
| | coarse c -50 | | 0.4 | |
| Profile 3 | coarse a +25 | 0.0035 | 0.17 | 37–42 |
| | coarse a -25 | 0.0035 | 0.17 | |
| | coarse b +25 | 0.0026 | 0.1 | |
| | coarse b -25 | 0.0028 | 0.1 | |
| | coarse c +25 | | 0.2 | |
| | coarse c -25 | | 0.2 | |

More precisely, the position of the master ${}^m\mathbf{p}$ is measured after each movement command sent to the slave inside the movement loop. If the displacement of the master, with respect to the final position ${}^m\mathbf{p}_f$ of the ongoing loop of movement commands, exceeds a minimum threshold the loop is stopped and the position ${}^s\mathbf{p}_{\mathit{eff}}$ effectively reached by the slave is computed. Then, the new final point of the slave ${}^s\mathbf{p}'_f$ is computed, which corresponds to the current master position, and the movement sequences that will drive the slave from the current position ${}^s\mathbf{p}_{\mathit{eff}}$ to the new final position ${}^s\mathbf{p}'_f$ is computed. Finally, a new loop sends all required movement commands to the slave, monitoring the master position after each command to check for over-threshold displacements, which will cause the loop to be broken again. The movement threshold in the master reference frame is equal to the displacement corresponding to the step amplitude of the slave, i.e. its movement resolution.

Strategy B has the same three profiles as strategy A, as shown in Tab. I.

### C. Strategy C

Strategy C differs from A and B in that the amplitude of the movement step is not fixed, but it is computed from the displacement of the master with respect to the position reached at the previous movement. To reduce waiting times and to move as fast as possible to the final position (${}^s\mathbf{p}_f$), only one command per DoF is issued to the manipulator. To compute the the number of *units of movements* of the one movement command, we have used the previously mentioned calibration curve that relates *step amplitude* and number of *units of movements*. To compute the calibration curve, we have measured the displacement of the manipulator corresponding to various movement commands. The data are distributed linearly, as shown in Fig. 3. Linear regression of

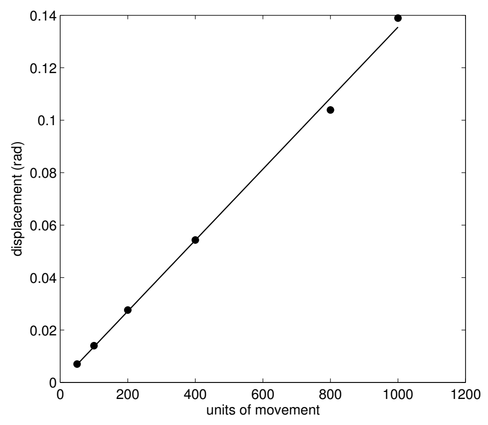

Fig. 3. Relation between step amplitude and number of units of movement.

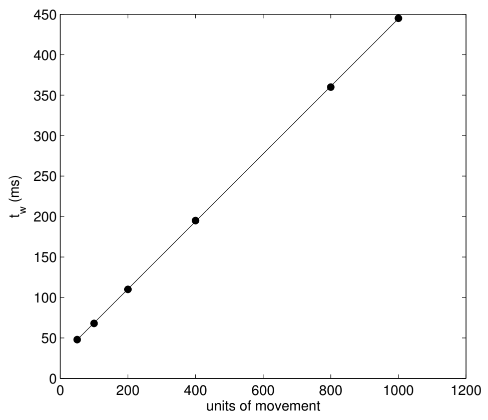

Fig. 4. Relation between waiting time and number of units of movement.

the data yields the equation:

$$d = m_d \cdot n + c_d \ ,$$

where $d$ is the displacement (in rad) resulting from a movement command of $n$ units of movement, and $m_d = 0.00013$ and $c_d \cong 0$ are the regression parameters. This equation is used to compute of the movement command to send to the manipulator to obtain the desired displacement.

Moreover, we have measured the time $t_w$ needed by the manipulator to carry out movements of various step amplitudes. Figure 4 plots $t_w$ against the number of units of movement. $t_w$ is a measure of the *minimum waiting time* defined in (1). The linear regression of the data yields:

$$t_w = m_t \cdot n + c_t \ ,$$

where $t_w$ is in ms, $n$ is the number of units of movement, and $m_t = 0.417$ and $c_t = 26.79$ are the regression parameters. We have added a fixed waiting time of 5 ms to the interpolated data to be sure that the movement has stopped before sending a new command, otherwise it might not be executed.

Therefore, in strategy C, we compute both the number of units of movement $n$ and the waiting time $t_w$ corresponding to the desired displacement $d$. We then send the movement command to the manipulator and wait for a time $t_w$ before measuring the master position again.

## IV. EXPERIMENTAL SETUP

To compare the strategies, 4 right-handed users (3 male, 1 female, age 24–33) performed a precise positioning task with the teleoperation system. The task, as common in literature of human computer interfaces, is a 2-D pointing task derived from the ISO9241-9 Standard [15]. A 2-D, rather than a 3-D, task was chosen because micromanipulation is more often performed under monoscopic vision. A transparent square with a black cross was placed on the manipulator end-effector. The users had to touch all circular targets placed at the corners of a regular hexagon, starting from the center of the hexagon and drawing a star-like pattern (see Fig. 5). Since no position encoders are integrated in the manipulator, the correct reaching of the targets was assessed visually. There were a total of 6 different tasks, with outer diameter (i.e. target distance $A$) ranging from 5 to 10 mm. Each task was repeated 5 times, for a total number of 120 tasks.

The 2-D pointing task (a type of *Fitts' task*) is commonly used in the literature of human computer interfaces to assess the throughput of interfaces according to Fitts' law [16], [17]. Fitts' tasks were first developed as a method of quantifying human performance in rapid-aimed movements [16], but have since been used also in the area of robotics, for measuring performance of autonomous and manual teleoperators [18], [13], [19]. Fitts' law states that the movement time ($t_m$) to tap between two targets of width $W$ and separated by a distance $A$ is given by

$$t_m = a + b \cdot ID$$

where $a$ and $b$ are task- and interface-dependend constants and $ID$ is the *Index of Difficulty*, which, in the formulation of Fitts' law according to Theorem 17 [20], is given by

$$ID = \log_2(A/W + 1)$$

It can be seen that the difficulty of the task is given by the ration $A/W$, rather than by $A$ itself. By magnifying both $A$ and $W$, we have reproduced tasks, to be performed under direct vision guidance, with the same $ID$s typically associated to micromanipulation tasks performed under a microscope.In our tasks, the $ID$ ranged from 2.12 to 2.94.

Fitts' law has been shown to be applicable in aimed movements tasks with transmission delay [9], [6], and this is the context in which we are using it. Discussing the many mathematical formulations of Fitts' law is out of scope of this paper; an in-depth analysis can be found in [21].

## V. RESULTS

Figure 6 shows the average time $MT$ spent for completing the 2-D task, averaged over the 4 users and the 5 repetitions of the same task by the users. For each strategy and profile, the movement times scale linearly with the ID, as can be shown from the linear regressions plotted in Fig. 6, which is consistent with Fitts' law.

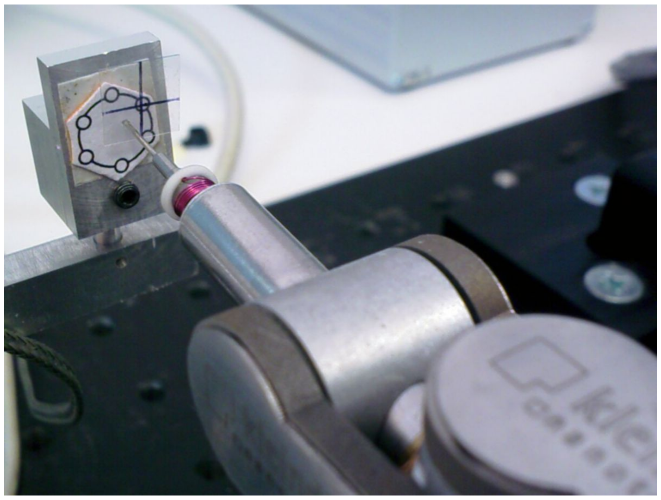

Fig. 5. A close-up view, showing the precision pointing task.

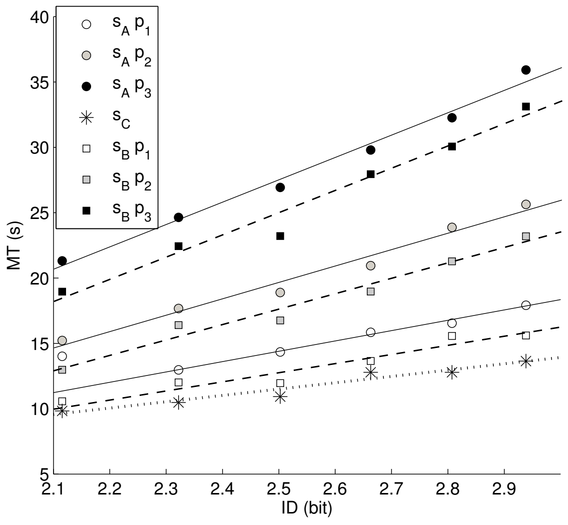

Fig. 6. Averaged movement times and linear regressions. Strategy C ($s_C$) clearly allows to complete the task in the shortest time. The first point (ID=2.12) of $s_A\ p_1$ has been rejected as an outlier for the fit (see text).

It can be seen that $s_B$ allows faster completion of the task than $s_A$, which is not surprising since $s_B$ allows the user to correct the movements of the slave while it is still moving, while $s_A$ forces the user to a sheer 'move-and-wait' strategy. What is more interesting is that the regression lines of $s_A$ and $s_B$ are parallel at all three movement profiles. This means that $s_A$ introduces a constant delay over $s_B$, which is independent from the movement amplitude. One reason could be that there is a constant number of "fine adjustment" movements that are done by the operator in proximity of the targets, and not while approaching the targets and that $s_B$ allows the user to correct for errors during these adjustment movements, thereby saving a constant time.

Also, the figure clearly shows that $s_C$ outperforms both $s_A$ and $s_B$, allowing to complete the task in the shortest time at all movement profiles. An interesting point is the outlier at ID=2.12 of the series $s_A\ p_1$: in this setup, the ID is low because the distance $A$ between targets is only about 3 times the target size $W$. Moreover, in $p_1$ the step amplitude has the same order of magnitude of the required precision: about 0.7 mm for DoF A, see Tab. I. With this setup, the users had trouble in making fine movements between the targets and much of the time was spent correcting overshoot movements, resulting in a longer overall time needed to complete the exercise. Therefore, using $p_1$ for this target results in high relative speed, since only a few movement commands are required to cross the whole distance between targets, but in low accuracy. In this task, $s_A\ p_1$ offers only about the same speed/accuracy trade-off as $s_A\ p_2$, despite the latter has about half the movement speed of the former, due to the smaller step amplitude. On the other hand, in $s_B\ p_1$ the performance is consistent across all IDs, due to the fact that overshoot movements can be corrected by interrupting the loop of movement commands.

One thing Fig. 6 cannot show is that, independently from the time required to complete the tasks, $s_C$ truly gives the operator a feeling of smoother teleoperation, which can be demonstrated by graphically comparing the trajectories described by the manipulator during the Fitts' task, as shown in Fig. 7. The top row and bottom left plots show the trajectories of the slave system with $s_B$ at the three profiles: notice that when the step amplitude decreases (from $p_1$ to $p_3$), the trajectory shape gets more precise, but at a cost of lower speed, as previously shown. The bottom right plot shows the slave trajectory with $s_C$: notice that the distance between the targets is covered in few large movements, while the number of movement commands gets higher and the points are more dense in the region of the targets (the corners of the star). This is even more evident if we plot the distribution of the step amplitudes of the movement commands for $s_C$. Figure 8 plots this distribution for different Fitts' tasks with target distances $A$ ranging from 5 to 10 mm. The step amplitudes $a_s$ have been grouped in 4 bins: $a_s < 25$, $25 \leq a_s < 50$, $50 \leq a_s < 100$, and $a_s \geq 100$, to match profiles 1, 2 and 3. Figure 8 shows a global trend of preferring large $a_s$ (to approach targets) and small $a_s$ (to touch targets), over medium-sized $a_s$. Moreover, more small steps $0 < a_s < 25$ are performed when the distance between the targets $A$ is small (e.g. $A = 5$ mm) than when $A = 10$ mm. The opposite is true for large steps with $a_s \geq 100$.

The reader may comment that the trajectories in Fig. 7 do not even look close to the ideal trajectory of the Fitts' task. It has to be kept in mind, though, that those are not the *actual* trajectories of the manipulator, but rather the concatenation of displacements, computed by the system, to make the slave manipulator follow the master interface position. Since there are no position encoders on the manipulator, we cannot keep track of the real positions.

## VI. CONCLUSIONS

In this paper we have presented and compared three different control strategies for driving a nanomanipulation system with direct vision feedback. One of the proposed

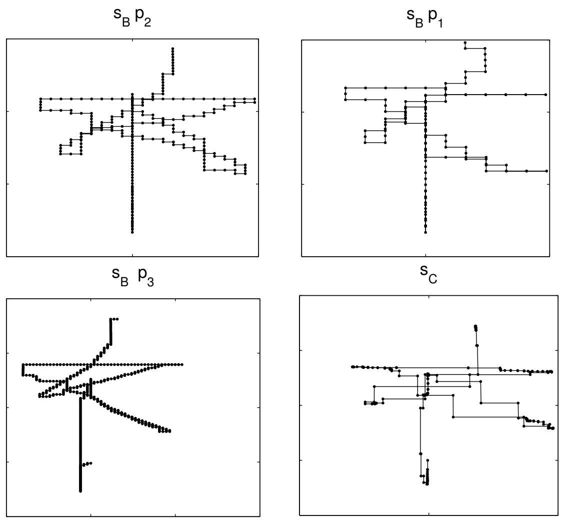

Fig. 7. Sample trajectories of the manipulator in the positioning task, using strategy B (top row and bottom left) and C (bottom right).

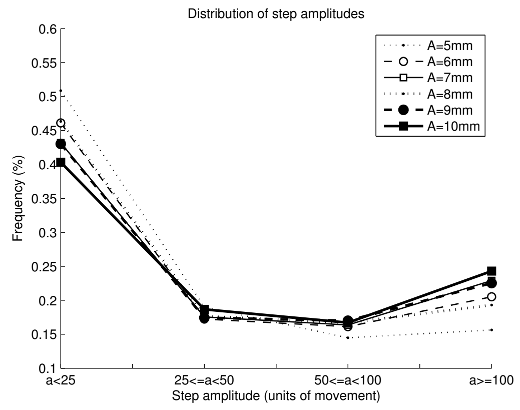

Fig. 8. The distribution of step amplitudes of the movement commands for $s_C$ in the Fitts' task, at different target distances $A$.

movement strategies, namely strategy C ($s_C$), achieves a satisfactoryspeed/accuracy trade-off for aimed precision movements. $s_C$ can be used for teleoperation tasks with precision movements over distances that range across several orders of magnitude: in fact, with $s_C$ the precision of movements is limited only by the precision of the operator. Moreover $s_C$ allows to overcome the latency of the system and, despite the intrinsic delays due to actuation and communication, the system is responsive and reaches a good level of interactivity. Due to the aforementioned characteristics, $s_C$ can be used in general to control vision-guided teleoperation systems affected by time delay.

The motion characteristics of the manipulator scale linearly with distance. Next, we will apply $s_C$ for movements performed under microscope guidance in the $\mu$m and sub-$\mu$m range, for a more complete characterization of the MM3A manipulator. We will verify Fitt's law across orders of magnitude and in particular compare the *fine* and *coarse* driving modes of the MM3A in a common movement range.

## References

[1] T. Sato, J. Ichikawa, M. Mitsuishi, and Y. Hatamura, "A new micro-teleoperation system employing a hand-held force-feedback pencil," in *Proc. IEEE International Conference on Robotics and Automation – ICRA*, 1994, pp. 1728–1733.

[2] M. Sitti and H. Hashimoto, "Tele-nanorobotics using atomic force microscope," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems – IROS*, vol. 3, 1998, pp. 1739–1746.

[3] D.-H. Kim, K. Kim, K.-Y. Kim, and S.-M. Cha, "Dexterous teleoperation for micro parts handling based on haptic/visual interface," in *Proc. of International Symposium on Micromechatronics and Human Science – MHS*, 2001, pp. 211–217.

[4] A. Pillarisetti, W. Anjum, J. Desai, G. Friedman, and A. Brooks, "Force feedback interface for cell injection," in *Proc. Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems – WHC*, 2005, pp. 391–400.

[5] W. M. Smith and K. F. Bowden, "The effects of delayed and displayed visual feedback on motor control," *Journal of Motor Behavior*, vol. 12, pp. 92–101, 1980.

[6] E. R. Hoffmann, "Fitts' law with transmission delay," *Ergonomics*, vol. 35, pp. 37–48, 1992.

[7] T. Sheridan, "Space teleoperation through time delay: review and prognosis," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 592–606, Oct. 1993.

[8] T. Imaida, Y. Yokokohji, T. Doi, M. Oda, and T. Yoshikawa, "Ground-space bilateral teleoperation of ets-vii robot arm by direct bilateral coupling under 7-s time delay condition," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 499–511, June 2004.

[9] W. R. Ferrell, "Remote manipulation with transmission delay," *IEEE Transactions on Human Factors in Electronics*, vol. 6, pp. 24–32, 1965.

[10] J. W. Hill, "Comparison of seven performance measures in a time-delayed manipulation task," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, pp. 286–295, 1976.

[11] L. Beccai, S. Roccella, A. Arena, F. Valvo, P. Valdastri, A. Menciassi, M. C. Carrozza, and P. Dario, "Design and fabrication of a hybrid silicon three-axial force sensor for biomechanical applications," *Sensors & Actuators: A. Physical*, vol. 120, no. 2, pp. 370–382, 2005.

[12] A. Sieber, P. Valdastri, K. Houston, C. Eder, O. Tonet, A. Menciassi, and P. Dario, "Triaxial force sensing mems device for tactile haptic force feedback applications in the nanoscale," in *in Proc. of 20th Eurosensors*, Gothenburg, Sweden, September 17-20 2006.

[13] R. Dubey, S. Everett, N. Pernalete, and K. Manocha, "Teleoperation assistance through variable velocity mapping," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 761–766, Oct. 2001.

[14] Q. Zhou, C. del Corral, P. Esteban, A. Aurelian, and H. Koivo, "Environmental influences on microassembly," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems – IROS*, vol. 2, 2002, pp. 1760–1765.

[15] ISO 9241-9:2000(E), "Ergonomic requirements for office work with vdts - part 9 - requirements for non-keyboard input devices," International Standards Organization, Tech. Rep., 2000.

[16] P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," *Journal of Experimental Psychology*, vol. 47, pp. 381–391, 1954.

[17] I. S. MacKenzie, T. Kauppinen, and M. Silfverberg, "Accuracy measures for evaluating computer pointing devices," in *Proc. CHI 2001 Conference on Human Factors in Computing Systems*, 2001, pp. 9–16.

[18] D. Cannon, "Experiments with a target-threshold control theory model for deriving fitts' law parameters for human-machine systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 8, pp. 1089–1098, Aug. 1994.

[19] I. Emeagwali, P. Marayong, J. Abbott, and A. Okamura, "Performance analysis of steady-hand teleoperation versus cooperative manipulation," in *Proc. International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems – HAPTICS*, 2004, pp. 316–322.

[20] I. S. MacKenzie, "A note on the information-theoretic basis for Fitts' law," *Journal of Motor Behavior*, vol. 21, pp. 323–330, 1989.

[21] R. Plamondon and A. M. Alimi, "Speed/accuracy trade-offs in target-directed movements," *Behav Brain Sci*, vol. 20, no. 2, pp. 279–303; discussion 303–349, Jun 1997.