

# Toward Intuitive Teleoperation of Micro/Nano-Manipulators with Piezoelectric Stick-Slip Actuators

Manikantan Nambi, Aayush Damani, and Jake J. Abbott

**Abstract**—Commercial micro/nano-manipulators, which utilize piezoelectric stick-slip actuators to achieve high precision over a large workspace, are currently controlled by a human operator at the joint level, leading to unintuitive and time-consuming teleoperation. Prior work has considered the use of computer-vision-feedback to close a control loop for improved performance, but computer-vision-feedback is not a viable option for many end users. In this paper, we discuss how open-loop models of the micro/nano-manipulator can be used to achieve desired end-effector movements, and we explain the process of obtaining open-loop models. We propose a rate-control teleoperation method that utilizes the obtained model, and we experimentally quantify the effectiveness of the method using a common commercial manipulator (the Kleindiek MM3A).

## I. INTRODUCTION

With visualization from scanning and transmission electron microscopes (SEMs/TEMs), as well as light microscopes, researchers are able to conduct experiments and construct devices with a precision of only a few nanometers. State-of-the-art micro/nano-manipulators currently enable: manipulation and isolation of individual nanomaterials and nanostructures for imaging and analysis, 3D construction and characterization of MEMS/NEMS, failure analysis in semiconductor fabrication, and surgery on individual cells [1], [2], [3]. The development and use of commercial manipulators like the Kleindiek MM3A [4], the Zyvex Nanomanipulator [5], Imina Technologies miBot [6], and the Attocube Nanopositioners [7] has increased with the demand for precise standardized tools for micro/nano-manipulation.

Among the limited range of actuators available, piezoelectric stick-slip actuators have become the foundation of modern micro/nano-manipulation (Fig. 1). They have a very simple structure, high positional accuracy, unlimited movable distance, and have high stability and stiffness as they are supported by guiding surfaces [8]. These actuators consist of a piezoelectric element and a sliding mass that moves relative to the piezoelectric element. They have two modes of operation, namely the fine mode and the coarse mode. In the fine mode, used to achieve the highest resolution possible, a slowly varying voltage is applied to the piezoelectric element resulting in a stretch, and friction between the piezoelectric element and the sliding mass causes the mass to move continuously with the piezoelectric element. In the coarse mode, used to take relatively large discrete steps, the applied voltage is quickly reversed after the initial stretch, resulting

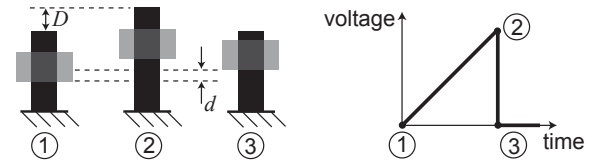


Fig. 1. Functional description of a piezoelectric stick-slip actuator. A saw-tooth voltage is applied to the piezoelectric element. As the voltage slowly increases from 1 to 2, the piezoelectric element stretches by a distance  $D$ , and due to stick-slip friction between the piezoelectric element and the sliding mass, the sliding mass also advances. When the voltage is quickly reduced from 2 to 3, the piezoelectric element quickly shrinks, but the inertia of the sliding mass prohibits it from moving backward as quickly, resulting in a net forward displacement of the sliding mass of  $d < D$ .

in a net displacement of the sliding mass relative to the piezoelectric element, as depicted in Fig. 1.

Rate control, which allows a user to command the end-effector of a manipulator at a desired speed and direction by coordinating the motion of the joints, has been known to the robotics community for decades, yet micro/nano-manipulators are operated in an inefficient open-loop manner using individual joint control (e.g., one knob per joint), ultimately due to the lack of sensor feedback at the joints. It is not always clear what combination of joint commands will lead to a desired end-effector movement from only a microscope image. In addition, manipulators are often mounted on moving stages and on different surfaces at varying angles. The user observing the end-effector of such a manipulator under a microscope has to perform the difficult task of mapping the image frame to his/her egocentric frame of reference. Use of multiple manipulators further complicates the situation. Thus, micro/nano-manipulation is currently unintuitive and time consuming.

Closed-loop control of micro/nano-manipulators using sensory data in real-time is challenging due to difficulty in getting real-time nanoscale visual and force feedback [3]. A number of different feedback control schemes such as voltage/frequency control [9], hybrid control [10], and sliding mode control [11] have been implemented for stick-slip actuators. Others have utilized vision feedback from SEM [12] and optical-microscope [13] images. Saeidpourazar and Jalili [14], [15] developed an adaptive controller to estimate the parameters of the manipulator on-line, and fused visual servoing and force feedback to enable closed-loop automatic control of the MM3A. Although using vision feedback has been shown to be successful for micro/nano-manipulation, it might not be feasible for many end users who work on shared SEMs, requiring them to book and pay for usage time. It can

The authors are with the Department of Mechanical Engineering, University of Utah, Salt Lake City, UT 84112, USA {m.nambi, aayush.damani, jake.abbott}@utah.edu

be difficult to implement a vision system on a shared SEM, owing to their high cost and sensitive nature.

The goal of our research is to enable teleoperated rate control of micro/nano-manipulators without relying on any feedback from the vision system, other than the human user looking at the microscope image. Specifically, we would like to 1) empirically model micro/nano-manipulators so that reliable position estimates can be obtained in the absence of feedback, 2) develop methods to calibrate the micro/nano-manipulators *in situ* to reduce modeling errors that can be introduced due to changes in environmental conditions or applied loads, 3) develop control methods to move the end-effector using position estimates obtained from the aforementioned models, and 4) develop methods to mitigate drift due to joint-estimate errors. In this paper, we focus exclusively on the coarse mode of operation.

Accurate modeling of piezoelectric stick-slip actuators is not a trivial task as they are known to suffer from nonlinearities such as hysteresis, creep, and drift, which degrade their performance. These actuators are also susceptible to environmental changes. However, a number of studies on piezoelectric drives have shown that it might be possible to reliably model these actuators. Peng and Chen [16] developed a model to explain the effect of end-effector masses on their stick-slip actuator. Lockwood *et al.* [17] found that when gravitational force was acting parallel to the axis of their stick-slip actuator, the step size and corresponding displacement rate in the ‘downward’ direction was observed to be 14.7% greater than in the ‘upward’ direction. Inertial drives developed at ETH Zurich were found to have open-loop error of less than 1% [18]. In a study most closely related to ours, Tonet *et al.* [19] studied different strategies for time-delayed teleoperation of systems using an MM3A, under direct vision feedback. They used incremental position control for master-slave coupling; in their variable step amplitude strategy, which allowed for the greatest precision, the user sacrificed control of the manipulator once a command was sent to the manipulator, losing the ability to correct for error in position attained by the manipulator.

## II. THE IMPULSIVE MANIPULATOR JACOBIAN

Micro/nano-manipulators such as the MM3A are kinematically no different from traditional robotic manipulators; we can relate joint movement to end-effector movement through a configuration-dependent manipulator Jacobian  $J(\vec{q})$  as  $\dot{\vec{x}} = J(\vec{q})\dot{\vec{q}}$ , where  $\vec{q}$  is the vector of joint positions,  $\vec{x}$  is the position of the end-effector, and the “dot” indicates a time derivative. For a desired end-effector movement  $\dot{\vec{x}}_d$ , we can simply command the joints to move as  $\dot{\vec{q}} = J^{-1}(\vec{q})\dot{\vec{x}}_d$ , assuming the Jacobian is invertible. However, this method makes two critical assumptions that are potentially invalid in the case of micro/nano-manipulators. First, although the form of the Jacobian can be found analytically, its calculation is dependent on knowledge of the current configuration, and commercial micro/nano-manipulators are not equipped with sensor feedback of the joint positions. Second, the discrete stick-slip nature of the piezoelectric actuators, combined with

the lack of sensor feedback, makes directly controlling the velocity of each joint,  $\dot{\vec{q}}$ , challenging.

Dynamically, a serial-link micro/nano-manipulator such as the MM3A has the same governing Lagrangian dynamic equation as a traditional robotic manipulator:

$$M(\vec{q})\ddot{\vec{q}} + C(\vec{q}, \dot{\vec{q}})\dot{\vec{q}} + G(\vec{q}) - J^T(\vec{q})\vec{f} = \vec{\tau} \quad (1)$$

where  $M(\vec{q})$ ,  $C(\vec{q}, \dot{\vec{q}})$ , and  $G(\vec{q})$  are configuration-dependent inertia, Coriolis, and gravity matrices, respectively,  $\vec{f}$  is any load applied to the end-effector, and  $\vec{\tau}$  is the vector of joint torques/forces. This equation is useful to control traditional robotic manipulators, where we have control over joint torques. However, in the case of micro/nano-manipulators, we can only command discrete steps. Because (1) is still a valid dynamic equation, the configuration-dependent matrices  $M(\vec{q})$ ,  $C(\vec{q}, \dot{\vec{q}})$ , and  $G(\vec{q})$  affect the end-effector’s movement during a single discrete step, but how they affect the movement is currently unknown.

Experiments indicate that the average step size of the actuators is affected by inertial load ( $m$ ), static loads ( $g$ ), and environmental factors. We hypothesize that a function  $\gamma$  can be constructed for a given joint that maps the above factors to a joint step size  $\Delta q = \gamma(m, g, \vec{\alpha}, \delta)$ , where  $\vec{\alpha}$  is the set of joint-specific parameters to be determined through calibration. The values  $m$  and  $g$  can be taken from the appropriate entries from  $M$  and  $G$  in (1). The average step size can be a function of the number of steps commanded if they are commanded too quickly to be considered independent, so  $\gamma$  is a nonlinear function of  $\delta$  in general.

For very small steps, the Jacobian can be approximated as  $\Delta\vec{x} = J(\vec{q})\Delta\vec{q}$ , where small joint steps  $\Delta\vec{q}$  lead to small end-effector steps  $\Delta\vec{x}$ . Using an open-loop model  $\gamma$  for each joint, we can relate the joint step size of the manipulator to the joint step commands as

$$\Delta\vec{q} = \Gamma(M(\vec{q}), G(\vec{q}), A, \vec{\delta})\vec{\delta} \quad (2)$$

where the matrix  $\Gamma$  is diagonal with the  $i^{th}$  element being the function  $\gamma$  described above for the  $i^{th}$  joint, and the vector  $\vec{\delta}$  contains the integer number of steps commanded to each joint, with sign indicating direction. The relationship (2) assumes that the joint step commands are given sequentially. Finally, an *impulsive manipulator Jacobian*  $J_\delta$  is formed as

$$\Delta\vec{x} = J(\vec{q})\Gamma(M(\vec{q}), G(\vec{q}), A, \vec{\delta})\vec{\delta} = J_\delta\vec{\delta} \quad (3)$$

The result is an open-loop model relating impulsive joint step commands to movements of the micro/nano-manipulator in both joint space and Cartesian space.

## III. RATE-CONTROL TELEOPERATION

When operating under a microscope, only a magnified view of the end-effector is visible to the user. We would like the user to be able to command a desired velocity to the end-effector (both direction, and magnitude within bounds) using only what can be observed in the image. We would like the actual velocity to be as close to the intended velocity as possible, but hypothesize that humans will be able to account for small errors with limited cognitive load [20]. We would

**Algorithm 1** Proxy-based rate control without memory

---

```

1: read  $\dot{\vec{x}}_d$ 
2:  $\vec{p} = \vec{x} + \dot{\vec{x}}_d \Delta t$ 
3:  $\Delta \vec{x} = \vec{p} - \vec{x}$ 
4:  $\Gamma = \text{compute}\Gamma(M(\vec{q}), G(\vec{q}), A, \vec{\delta})$ 
5:  $J = \text{compute}J(\vec{q})$ 
6:  $J_\delta = J\Gamma$ 
7:  $\vec{\delta}_{real} = J_\delta^{-1} \Delta \vec{x}$ 
8:  $\vec{\delta} = \text{round}(\vec{\delta}_{real})$ 
9:  $\Delta \vec{q} = \Gamma \vec{\delta}$ 
10:  $\vec{q}_f = \vec{q} + \Delta \vec{q}$ 
11:  $\vec{x}_f = \text{forwardKinematics}(\vec{q}_f)$ 
12:  $\vec{x} = \vec{x}_f$ 
13:  $\vec{q} = \vec{q}_f$ 
14: return  $\vec{\delta}$ 

```

---

like the discrete-step nature of the controller to be transparent to the user. To realize this goal, we propose a rate-control method that handles the step commands for the user, based on commanded end-effector velocity.

Algorithm 1 shows the basic steps for the proposed method. The algorithm is called in a continuous loop by the software controlling the manipulator, which we assume is running with a constant time step of  $\Delta t$  seconds. The algorithm takes the desired input velocity ( $\dot{\vec{x}}_d$ ) commanded by the user (e.g., from an input device such as a joystick) and returns the number of steps ( $\vec{\delta}$ ) to be taken at each joint during the current cycle. Starting from current estimates of end-effector position  $\vec{x}$  and joint-configuration  $\vec{q}$ , a proxy point ( $\vec{p}$ ) is projected away by a distance  $\dot{\vec{x}}_d \Delta t$ , giving the desired end-effector location. The number of steps that the manipulator should take to reach the proxy is then calculated as  $\vec{\delta}_{real}$  by inverting the impulsive manipulator Jacobian. However, the result is a vector of real numbers that must be converted to an integer number of steps to be commanded as  $\vec{\delta}$ ; we simply round to the nearest integer. This rounding step implies that we will never perfectly attain the desired proxy position. The estimated final joint configuration ( $\vec{q}_f$ ) is then calculated based on the steps actually commanded, and the final end-effector position ( $\vec{x}_f$ ) is estimated using the manipulator's kinematic model. Finally, the current estimates of end-effector position and joint configuration are updated for the next cycle. The algorithm sequence as given assumes that  $\Gamma$  is not a function of  $\vec{\delta}$ ; if it is, then lines 4–8 must be called iteratively from an initial guess of  $\vec{\delta}$  until it converges.

We have also explored an alternative to the basic algorithm, which we will call Algorithm 2. It is identical to Algorithm 1, except that Line 2 is replaced by

$$\vec{p} = \vec{p} + \dot{\vec{x}}_d \Delta t \quad (4)$$

Algorithm 2 maintains memory of the proxy position from the last cycle, which can be beneficial considering that the previous desired proxy position was not perfectly attained.

Fig. 2 shows the simulation results for a 2D task in a horizontal plane for both algorithms applied to the MM3A.

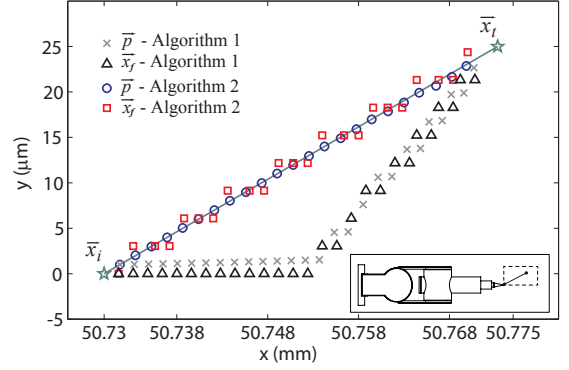


Fig. 2. Simulation results for Algorithms 1 and 2 implemented on an MM3A, for a 2D task in a horizontal plane. At each instant, the user commands the end-effector to move from its current location toward  $\vec{x}_t$  at 0.1 mm/s. A line joining  $\vec{x}_i$  and  $\vec{x}_t$  is shown for reference. The step-size model used has  $\gamma_1 = 60 \mu\text{rad}$  and  $\gamma_3 = 0.8 \mu\text{m}$ , where  $\gamma_1$  and  $\gamma_3$  are diagonal elements of  $\Gamma$ . Inset shows the top view of the MM3A with its workspace in the horizontal plane used in simulation.

A 2D task in the horizontal plane can be performed by actuating joints 1 and 3, and fixing joint 2 at  $q_2 = -\pi/2$  (i.e., outstretched as shown in Fig. 3); the kinematics of the resulting 2-DOF manipulator are trivial to compute, and are omitted here. For this simulation, the initial position  $\vec{x}_i$  is computed for the joint configuration  $\vec{q} = [0, -\pi/2, 0]^T$ . The desired target position  $\vec{x}_t$  to be reached by the user is set at a distance of  $50 \mu\text{m}$  from the initial position at an angle of  $30^\circ$  from the x-axis. In our simulation, we assume an ideal human operator that is trying to reach  $\vec{x}_t$  at a rate of 0.1 mm/s, and that always points the desired velocity of the end-effector perfectly from the current end-effector location toward  $\vec{x}_t$ . The simulation is stopped when the error between  $\vec{x}$  and  $\vec{x}_t$  is less than  $5 \mu\text{m}$ . We are assuming the model of the micro/nano-manipulator to be perfect in this simulation, but in reality the actual position observed under the microscope will be different from that estimated; this issue will be discussed subsequently.

For some low velocities, Algorithm 1 shows a drift as can be seen in Fig. 2. Because the algorithm updates the proxy from the end-effector's current position, at low velocities it decides that taking a step along the x-axis and no step along the y-axis is the best solution for many cycles. This is due to the rounding of  $\vec{\delta}_{real}$  to  $\vec{\delta}$ , and is a function of the end-effector step  $\Delta \vec{x}$  caused by each joint. As the direction of commanded velocity is updated by the ideal operator, at some point the error along the y-axis will be high enough for the algorithm to command movement in that direction. In Algorithm 2, because the proxy is updated from its previous position, it acts as memory of the previous error between commanded and actual velocity and is able to compensate for this error. Algorithm 2 appears to have better performance than Algorithm 1, but this is due to the target position being stationary. If we assume that the period  $\Delta t$  is small compared to human bandwidth, then Algorithm 2 is likely to give desirable performance even if the desired target  $\vec{x}_t$  is changing. A high value of  $\Delta t$  would cause

larger deviations of the end-effector from the desired path and could result in unstable behavior as the operator tries to overcorrect for the deviation. If the commanded velocity is below a certain threshold, then Algorithm 1 will result in no movement of the end-effector with no error accumulation, but using Algorithm 2 will cause the proxy to keep moving until sufficient distance has been generated such that the end-effector can take a step toward the proxy; whether or not this property is desirable requires further research.

#### IV. EXPERIMENTAL METHODS

##### A. Apparatus

1) *MM3A Manipulation System*: The Kleindiek MM3A manipulation system consists of the MM3A micromanipulator and the NanoControl (NC) unit (Fig. 3). The MM3A manipulator has a RRP configuration with two rotary (R) and one prismatic (P) joint. The MM3A has two modes of operation, the fine mode and the coarse mode, which enable it to achieve high resolution of up to 0.25 nm for the P joint and  $10^{-7}$  rad for the R joints in the fine mode, and high speeds of up to 10 mm/s in the coarse mode.

The NC unit is used to command control signals to the MM3A. It has a four knobs, three of which are used to control the individual joints on the MM3A in both the fine and the coarse mode. The number of steps to be taken by a joint for a turn of the knob can be set in the NC unit. Commands can also be sent to the NC unit via a serial port as a string of the form “Mode  $\delta$  Joint”, where mode is either ‘coarse’ or ‘fine’, “ $\delta$ ” is an integer value which specifies the number of steps to be taken in each command with sign indicating direction, and “Joint” is either “A”, “B”, or “C” corresponding to joints 1, 2, and 3, respectively. Only one joint can be controlled by the NC unit per command. Hence, three commands have to be sent sequentially to control the three joints of the MM3A.

2) *Vision System*: A Basler A601fc firewire camera with a VZM 450i zoom lens is used to obtain position information for experiments herein (Fig. 3). The VZM 450i has a primary magnification of  $0.7\times - 4.5\times$ , with a respective field view of 12.0 mm – 1.9 mm. The Basler A601fc has a resolution of  $659\times 491$  pixels with a maximum frame rate of 60 fps. The vision system has a resolution of  $10\ \mu\text{m}$  at a magnification of  $1\times$  for the lens. An LED backlight was used as the primary light source for the vision system.

3) *Software*: A custom GUI program developed in C++ using the Qt 4.6 ui framework and OpenCV vision library was used to collect calibration data and implement teleoperation. The program communicates with the NC unit using serial communication. Once a command is sent to the NC unit, the software waits for feedback from the NC unit, which indicates the execution of a command. Audio inputs are taken from a microphone and the Fast Fourier Transform (FFT) of the audio signal is computed for the calibration method described below. Counters are used to keep track of the number of commands ( $n$ ) commanded to each joint, and the total number of steps along a joint is given by  $\sum_{i=1}^n |\delta_i|$ .

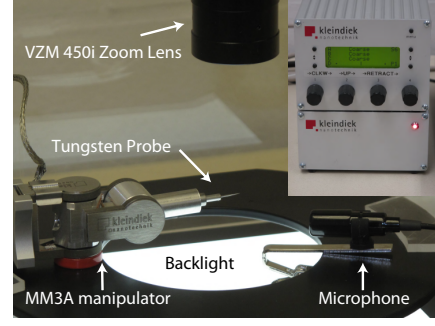


Fig. 3. Experimental setup. A tungsten probe with a tip diameter of  $1\ \mu\text{m}$  is attached as an end-effector. The NanoControl unit is shown in the inset.

OpenCV library is used to capture frames from the camera at a rate of 60 fps, and a blob-detection algorithm is run to track the tip of the tungsten probe attached to the manipulator in real time. Commands for a single joint are given to the NC unit at a rate of 100 Hz. The program was run on a Dell Optiplex computer (2 Ghz Dual Core, 2 GB ram) running the Ubuntu 10.04 operating system.

##### B. Calibration

To use the impulsive manipulator Jacobian, we need accurate knowledge of  $\Gamma$ , which relates joint step commands  $\vec{\delta}$  to the joint step size  $\Delta\vec{q}$ . A deterministic  $\Gamma$  would result in the best performance for the algorithms in Section III, but due to uncertainty in the step size and lack of sensor feedback in the MM3A, we calculate the average step size as:

$$\bar{\gamma}_i = \frac{R_i}{\delta_{t,i}} \quad (5)$$

where  $\bar{\gamma}_i$  is the average step size for a joint  $i$ ,  $R_i$  is the total range of motion for joint  $i$  ( $4\pi/3$  rad for joints 1 and 2, and 12 mm for joint 3), and  $\delta_{t,i}$  is the total number of steps required to travel across  $R_i$ :  $\delta_{t,i} = |\delta|n$ , where  $n$  is the number of commands given to a joint at  $\delta$  steps/command.

The displacement of the end-effector for a single step along a joint is so small that it is not visible to the naked eye. Hence, it is not possible to visually detect that a joint has reached the limit of its joint space. However, there is a perceptible change in sound made by the manipulator when a joint of the MM3A hits a mechanical stop as it reaches the limit of its travel, and the software detects this intensity rise in the FFT of the audio signal (captured by the microphone) at a frequency of 2.7 kHz. This property was used to develop an audio limit switch that enables us to accurately detect the end of travel for a joint. The change in intensity of the FFT has to be tuned for each joint.

The values of  $\bar{\gamma}_i$  are measured before the start of an experiment to reduce the effect of environmental changes, as there is a significant change in step size from day to day. As the experiments in this paper are performed in a horizontal plane using joints 1 and 3, only  $\bar{\gamma}_1$  and  $\bar{\gamma}_3$  are measured. Joints 1 and 3 are moved across  $R_1$  and  $R_3$  by commanding a single step ( $\delta = 1$ ) in each command, and  $\delta_{t,1}$  and  $\delta_{t,3}$



are obtained.  $\Gamma$  is constructed by substituting the diagonal elements  $\Gamma_1$  and  $\Gamma_3$  with  $\bar{\gamma}_1$  and  $\bar{\gamma}_3$ . Two different step size models  $\bar{\gamma}_1$  and  $\bar{\gamma}_1(q_3)$  were evaluated. For the constant value of  $\bar{\gamma}_1$ ,  $q_3$  is kept at zero.  $\bar{\gamma}_1(q_3)$  takes into account the increase in inertial load on joint 1 due to the position of joint 3. To determine the effect of increase in inertial load, the number of steps required by joint 1 to travel across  $R_1$  with  $q_3 = 0$  mm and  $q_3 = 12$  mm are determined as  $\delta_{t,1,0}$  and  $\delta_{t,1,12}$  respectively. The step size function  $\bar{\gamma}_1(q_3)$ , which gives a configuration dependent step size for joint 1, is then calculated using linear interpolation as:

$$\bar{\gamma}_1(q_3) = \frac{R_1}{\delta_{t,1,0} + (\delta_{t,1,12} - \delta_{t,1,0})\left(\frac{q_3}{R_3}\right)} \quad (6)$$

Before starting an experiment, each joint is driven along the full range of its motion three times to reduce any warming effect that might be present in the actuators.

### C. Experiments

Experiments were designed to test the effectiveness of the proposed method with the MM3A. For this purpose, a 2D task was performed in the horizontal plane using joints 1 and 3, with the simulated ideal human operator driving the manipulator from the initial position ( $\vec{x}_i$ ) to the desired target position ( $\vec{x}_t$ ) (Fig. 4). At the start of a trial, the initial position ( $\vec{x}_i$ ) is computed for the joint configuration  $\vec{q} = [0, -\pi/2, 0]^T$ . In the initial position, a step along joint 3 leads to tip movement along the x-axis and a step along joint 1 leads to tip movement along the y-axis. The desired target position ( $\vec{x}_t$ ) is set at a distance of 4 mm from  $\vec{x}_i$ , at different angles of  $\theta = 0^\circ, 30^\circ, 60^\circ$ , and  $90^\circ$  from the x-axis. The ideal human operator is given the position of the tip (i.e., the end-effector) from the vision system, so that it can always point the desired input velocity ( $\vec{x}_d$ ) towards  $\vec{x}_t$ . The number of steps to be taken by the manipulator  $\vec{\delta}$  in each cycle is then calculated using Algorithm 2, and a set of two commands corresponding to joints 1 and 3 are sequentially sent to the NC unit with a period of 0.01 s between commands. As the ideal human operator is continuously correcting  $\vec{x}_d$  to point towards  $\vec{x}_t$ , the tip will always reach  $\vec{x}_t$ . However, the estimated position  $\vec{x}_f$  will be different than  $\vec{x}_t$  because of modeling errors. Trials are conducted with the ideal human operator commanding two different velocities of  $|\vec{x}_d| = 0.05$  mm/s and 0.5 mm/s. At the end of each trial, the position of the tip is manually reset to  $\vec{x}_i$ . Five trials are conducted for each  $|\vec{x}_d|$  and  $\theta$  value. Due to limitations in the speed of serial communication and the time taken for executing a coarse step by the manipulator, a maximum of 24 coarse steps can be commanded to the manipulator along a joint when using a rate of 100 Hz. Hence, the maximum magnitude of velocity that can be achieved by the manipulator at any instant, at any point in the 2D workspace under consideration, is limited by the maximum value of  $24\bar{\gamma}_{min}$ , where  $\bar{\gamma}_{min}$  is the average step size of the joint with smaller end-effector movement for a single step. The experiments were carried out for the two step size models  $\bar{\gamma}_1$  and  $\bar{\gamma}_1(q_3)$ .

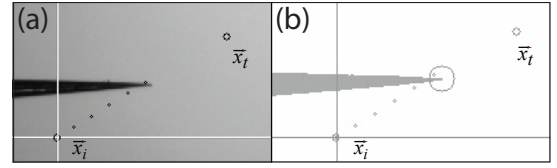


Fig. 4. Experimental results, corresponding to the simulation of Fig. 2. (a) Screenshot of the camera-image display on the GUI with the end-effector moving at  $\vec{x}_d = 0.5$  mm/s. (b) Screenshot of the display after applying blob detection algorithm. The medium circles indicate the initial ( $\vec{x}_i$ ) and desired positions ( $\vec{x}_t$ ) at the start of a trial. The large circle indicates the probe tip as seen by the vision system and the small circles show the history of estimated positions during the trial.

## V. RESULTS AND DISCUSSION

As a metric to quantify the error in our step-size models, we use the relative error in the estimated position defined as:

$$E_r = \frac{|\vec{x}_f - \vec{x}_t|}{|\vec{x}_t - \vec{x}_i|} \quad (7)$$

where  $\vec{x}_f$  is the estimated position of the end-effector and  $\vec{x}_t$  is the target position (and actual position, within the convergence criterion) reached by the end-effector in a trial.  $E_r$  gives the relative error between the actual and estimated position of the end-effector with respect to the total displacement.

Fig. 5 shows experimental results for  $E_r$  plotted against  $\theta$ , for different values of  $|\vec{x}_d|$ , for step size models  $\bar{\gamma}_1$  and  $\bar{\gamma}_1(q_3)$ .  $E_r$  seems to increase with an increase in  $|\vec{x}_d|$ . A higher value for  $E_r$  at a velocity of 0.5 mm/s can be explained by the fact that the models  $\bar{\gamma}_1$  and  $\bar{\gamma}_1(q_3)$  were measured at  $\delta = 1$ . At  $|\vec{x}_d| = 0.05$  mm/s,  $\delta$  commanded to each of the joints is at most 1, while at  $|\vec{x}_d| = 0.5$  mm/s value of  $\delta$  commanded to each joint is much higher than 1. As  $\delta$  increases there is significant change in the step size and hence, higher velocities lead to a higher value for  $E_r$ .

$\bar{\gamma}_1(q_3)$  seems to be a better predictor of step size than  $\bar{\gamma}_1$  at  $|\vec{x}_d| = 0.05$  mm/s. For stick-slip actuators, we would expect that an increase in inertial load would increase the efficiency of the stick-slip phase and result in an increase in step size. We observe this behavior for joint 1 at  $\delta = 1$  and  $\delta = 2$ , but at values of  $\delta > 2$ , this effect is reversed and inertial load seems to reduce the step size. This results in degraded performance for model  $\bar{\gamma}_1(q_3)$  at high velocities.

$E_r$  has a maximum value of around 20% for velocities considered in this paper, which corresponds to a maximum error of about 20% in the joint estimates. Unless we have a perfect model of the manipulator, the error in joint estimates will accumulate over time resulting in an unintuitive response from the system for a given user input. From the experiments conducted herein, we know that even with an error of 20% in the joint estimates the ideal human operator is able to position the end-effector within 1 pixel of the desired target position in a stable manner. It was observed that even if the joint estimates were always set to the initial joint configuration (i.e., the joint estimates were not updated), the ideal human operator was able to position the end-effector accurately with no visible drift in the path taken

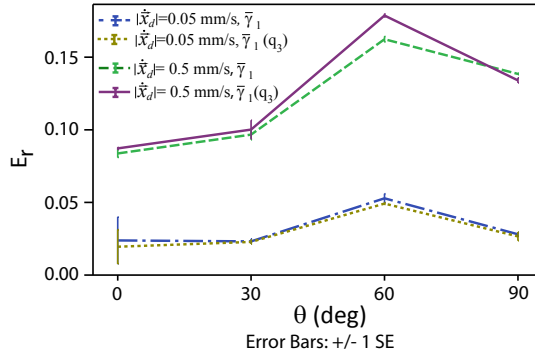


Fig. 5. Experimental results for  $E_T$ . The step size models used were  $\bar{\gamma}_1 = 68 \mu\text{rad}$ ,  $\bar{\gamma}_1(q_3) = (14688 - 171q_3)^{-1} \mu\text{rad}$ , and  $\bar{\gamma}_3 = 0.77 \mu\text{m}$ .

by the end-effector as compared to the path taken when the joint estimates were being updated. When the joint estimates are not being updated, there would be no drift in joint estimates, but an error in the estimated Jacobian (maximum of 8% for the workspace used) due to the error in the joint estimates would result in an error in the velocity actually achieved by the manipulator. The fact that the ideal human operator is able to position the end-effector accurately means that this error in velocity is relatively low, and we believe that a human operator should also be able to compensate for small errors in velocity. Thus, in small workspaces typically encountered in micro/nano-manipulation, drift in joint estimates can be avoided by setting the joint estimates to a fixed known configuration. In the future, we would like to develop drift-mitigation techniques that would allow the human operator to use the manipulator for long periods of time without the need for recalibration.

The ideal human operator has infinite visual resolution and no processing time delay. However, this is not true of a real human operator. Thus, stability of our method has to be tested with human subject tests.

From step-size measurements recorded under different loading conditions and orientations for the joints, we know that there is a significant effect of inertial loads, static loads, and  $\delta$  on the step size. The step size for joint 3 was found to vary by more than 50% when gravity was acting along the axis of the joint. However, the trends in step size are highly repeatable, which tells us that a model of the manipulator joints of the form of (2) can be developed. Additionally, the current calibration routine for joints 1 and 3 are run at  $\delta = 1$ , which takes about 25 minutes to complete. Using a model that takes into account the effect of  $\delta$  would allow us to perform the calibration routine in a shorter period of time by using a higher value for  $\delta$  during calibration. We would like to develop efficient calibration techniques that would take advantage of the observed trends in step sizes and enable us to obtain a complete open-loop model of the manipulator with a minimum number of measurements.

## VI. CONCLUSIONS

In this paper, we have presented a rate-control teleoperation method for control of manipulators using stick-slip actuators. The formulation of an impulsive manipulator

Jacobian was explained, which enables us to use open-loop models of the manipulator to solve for the input number of steps required by the manipulator for a desired end-effector movement. Experimental results quantifying the effectiveness of the proposed methods were presented. We found that effective teleoperation is possible despite inaccurate joint measurements, and we discussed ways to minimize errors.

## ACKNOWLEDGEMENT

This research was born of discussions with Dr. Lixin Dong at Michigan State University.

## REFERENCES

- [1] J. J. Abbott, Z. Nagy, F. Beyeler, and B. J. Nelson, "Robotics in the small, Part I: Microbotics," *IEEE Robotics and Automation Mag.*, vol. 14, no. 2, pp. 92–103, 2007.
- [2] L. Dong and B. J. Nelson, "Robotics in the small, Part II: Nanorobotics," *IEEE Robotics and Automation Mag.*, vol. 14, no. 3, pp. 111–121, 2007.
- [3] S. Fatikow and V. Eichhorn, "Nanohandling automation: trends and current developments," *Proc. Institution of Mechanical Engineers, Part C: J. Mechanical Engineering Science*, vol. 222, no. 7, pp. 1353–1369, 2008.
- [4] "Kleindiek Nanotechnik MM3A," <http://www.nanotechnik.com/>.
- [5] "Zyvex Nanomanipulator," <http://www.zyvex.com/>.
- [6] "Imina Technologies miBot," <http://www.imina.ch/>.
- [7] "Attocube Nanopositioners," <http://www.attocube.com/>.
- [8] T. Higuchi and Y. Yamagata, "Micro robot arm utilizing rapid deformations of piezoelectric elements," *Advanced Robotics*, vol. 6, no. 3, pp. 353–360, 1992.
- [9] M. Rakotondrabe, Y. Haddab, and P. Lutz, "Voltage/Frequency Proportional Control of Stick-Slip Micropositioning Systems," *IEEE Trans. on Control Systems Technology*, vol. 16, no. 6, pp. 1316–1322, Nov. 2008.
- [10] B. Sedghi, B. Srinivasan, and R. Longchamp, "Control of hybrid systems via dehybridization," in *Proc. American Control Conference*, 2002, pp. 692–697.
- [11] H. C. Liaw, B. Shirinzadeh, and J. Smith, "Enhanced sliding mode motion tracking control of piezoelectric actuators," *Sensors and Actuators A: Physical*, vol. 138, no. 1, pp. 194–202, 2007.
- [12] B. E. Kratochvil, L. Dong, and B. J. Nelson, "Real time rigid-body visual tracking in a scanning electron microscope," *Int. J. Robotics Research*, vol. 28, no. 4, pp. 488–511, 2009.
- [13] H. Hötzenndorfer, I. Giouroudi, S. Bou, and M. Ferros, "Evaluation of different control algorithms for a micromanipulation system," in *Int. Conf. Engineering and Mathematics*, 2006.
- [14] R. Saeidpourazar and N. Jalili, "Nano-robotic manipulation using a RRP nanomanipulator: Part A- mathematical modeling and development of a robust adaptive driving mechanism," *Applied Mathematics and Computation*, vol. 206, pp. 618–627, 2008.
- [15] —, "Nano-robotic manipulation using a RRP nanomanipulator: Part B- Robust control of manipulator's tip using fused visual servoing and force sensor feedbacks," *Applied Mathematics and Computation*, vol. 206, pp. 628–642, 2008.
- [16] J. Y. Peng and X. B. Chen, "Modeling of piezoelectric-driven stick-slip actuators," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 2, pp. 394–399, 2011.
- [17] A. J. Lockwood, J. J. Wang, R. Gay, and B. J. Inkson, "Characterising performance of TEM compatible nanomanipulation slip-stick inertial sliders against gravity," *Journal of Physics:Conference Series*, vol. 126, no. 1, 2008.
- [18] W. Zesch, R. Buchi, A. Codourey, and R. Siegwart, "Inertial drives for micro- and nanorobots: Two novel mechanisms," in *SPIE Micro-robotics and Micromechanical Systems*, 1995, pp. 80–88.
- [19] O. Tonet, M. Marinelli, G. Megali, A. Sieber, P. Valdastrì, A. Menicci, and P. Dario, "Control of a teleoperated nanomanipulator with time delay under direct vision feedback," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2007, pp. 3514–3519.
- [20] C. Ware and R. Arsenault, "Frames of reference in virtual object rotation," in *Symp. Applied Perception in Graphics and Visualization*, 2004, pp. 135–141.