Drift Compensation for Automatic Nanomanipulation with Scanning Probe Microscopes

Babak Mokaberi and Aristides A. G. Requicha, *Fellow, IEEE*

***Abstract*—Manipulation of nanoparticles with Atomic Force Microscopes (AFMs) has been under development for a decade, and is now well established as a technique for prototyping nanodevices and for other applications. Until now, the manipulation process for particles with sizes of a few nanometers has been very labor intensive. This severely limits the complexity of the structures that can be built. Particle sizes on the order of 10 nm are comparable to the spatial uncertainties associated with AFM operation, and a user in the loop has been needed to compensate for these uncertainties.**

This paper addresses thermal drift, which is the major cause of errors for AFMs operated in ambient conditions. It is shown that drift can be estimated efficiently by using Kalman filtering techniques. This approach has firm theoretical foundations and is validated by the experimental results presented in this paper. Manipulation of groups of 15-nm particles is demonstrated under program control, without human intervention, over a long period of time, in ambient air and at room temperature. Coupled with existing methods for high-level motion planning, the manipulation capabilities introduced here will permit assembling, from the bottom up, nanostructures much more complex than those being built today with AFMs.

***Note to Practitioners*—Nanomanipulation with Scanning Probe Microscopes (SPMs) has potential applications in nanodevice and system prototyping, or in small-batch production if multi-tip arrays are used instead of single tips. However, SPM nanomanipulation is still being used primarily in research labs. A major obstacle to its wider use is the labor and time involved in the process. These are due largely to spatial uncertainty in the position of the tip (which is analogous to a robot's end effector) relative to the sample being manipulated. Today, a skilled user is needed to determine where the tip is, and to correct manipulation errors due to inaccurate positional estimates. The major cause of this spatial uncertainty is thermal drift between the tip and the sample. At the time scales relevant to manipulation, the drift can reach values comparable to the size of the objects, especially if these are below ~ 10 nm. The techniques discussed in this paper compensate for the drift and enable automated manipulation, with associated savings in time and labor, and increased complexity of the resulting structures.**

***Index Terms*—Atomic Force Microscopes; Kalman filtering; nanomanipulation; nanoassembly; nanorobotics; Scanning Probe Microscopes; spatial uncertainty.**

I. Introduction

Atom manipulation with Scanning Probe Microscopes (SPMs) was first demonstrated in the early 1990s, and manipulation of particles with sizes on the order of a few nm to the tens of nm soon followed—see [1, 2] and references therein. Today, particles with diameters of ~ 10 nm are manipulated routinely with Atomic Force Microscopes (AFMs) at USC's Laboratory for Molecular Robotics (LMR) and elsewhere. (The AFM is a specific type of SPM that exploits interatomic forces between a sharp tip and a sample.) Nanomanipulation operations are used to prototype nanoscale devices, and to repair or modify nanostructures built by other means.

Manipulation of small nanoparticles (with diameters below ~ 30 nm, say) in ambient conditions, i.e., at room temperature, in air or in a liquid, and without stringent environmental controls, requires extensive user intervention to compensate for the many spatial uncertainties associated with AFMs and their piezoelectric drive mechanisms. Uncertainties are introduced by phenomena that range from non-linearities in the voltage-displacement curves that characterize the piezos, to creep, hysteresis, and thermal drift. The latter is the major cause of spatial uncertainty in our lab, and is due primarily to thermal expansion and contraction of the AFM components. For example, a one degree change in temperature will cause a 50 nm change in the length of a mechanical part that is 5 mm long (assuming a typical expansion coefficient of 10^{-5} /°C).

AFM vendor software usually compensates for piezo non-linearities. Hysteresis effects can be greatly reduced by scanning always in the same direction, and creep effects can be nearly eliminated by waiting a few minutes after each large motion of the scanner (although this is very inefficient). In addition, modern, top-of-the-line instruments are equipped with feedback loops that claim positioning errors below 1 nm in the x, y plane of the sample [3]. Feedback can compensate

Manuscript received October 31, 2004. This work was supported in part by the NSF under grants EIA-98-71775 and DMI-02-09678.

B. Mokaberi is with the Laboratory for Molecular Robotics, 941 Bloom Walk, University of Southern California, Los Angeles, CA 90089-0781 (e-mail: mokaberi@usc.edu).

A. A. G. Requicha is with the Laboratory for Molecular Robotics, 941 Bloom Walk, University of Southern California, Los Angeles, CA 90089-0781 (tel: 213.740.4502; fax: 213.740.7512; e-mail: requicha@usc.edu).

An earlier version of this paper was presented at the IEEE International Conference on Robotics and Automation, New Orleans, LA, April 25-30, 2004.

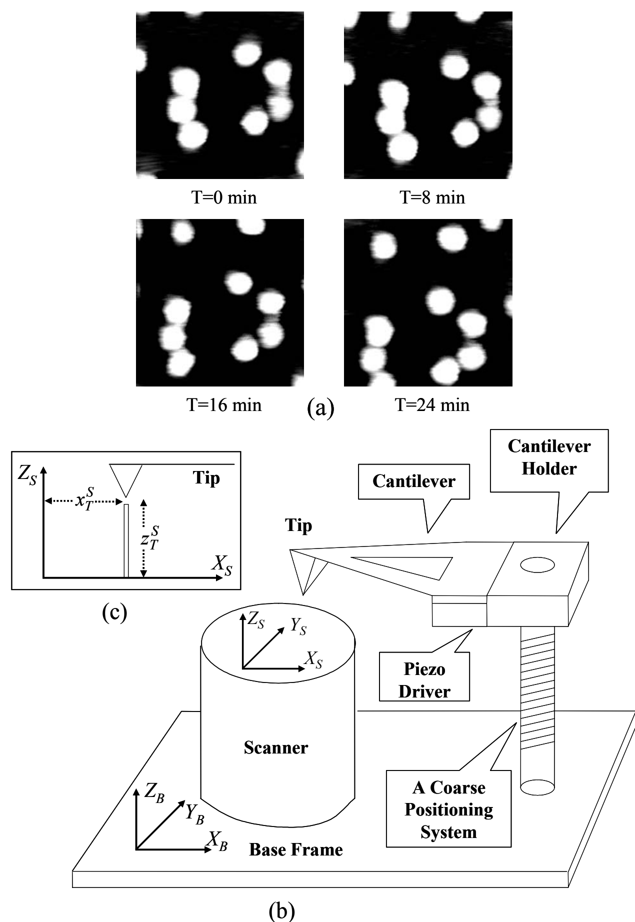


Fig. 1. (a) Four images from a 180 nm region of a sample, taken at 8 min intervals. The objects shown are Au nanoparticles with 15 nm diameters. (b) A schematic diagram of an AFM machine. (c) A spatial asperity below the tip.

for non-linearities, hysteresis and creep, but not tip drift—see Section II.B below. Drift tends to increase with time, which implies that complicated assemblies cannot be completed without frequent user interaction. Moreover, most of the AFMs in use today either have no x, y feedback, or their feedback loops have noise levels of several nm, which are too large for manipulation of particles with sizes ~ 10 nm. These machines are normally operated open loop for the small scan sizes (< 1 μm) used in the manipulation of small nanoparticles, to avoid introducing additional noise through the feedback circuitry. With or without x, y feedback, drift compensation remains a crucial issue for successful nanomanipulation.

Typical assemblies of small nanoparticles built by nanomanipulation today consist of some ten particles, and may take an experienced user a whole day to construct [13]. To move towards more complex assemblies requires that the manipulation process become more automated, and this in turn requires compensation of the spatial uncertainties associated with AFMs—especially drift, which is the most pernicious one, as we have argued above.

The remainder of this paper discusses the characteristics of drift, how to estimate it and compensate for it, our implementation of drift compensation, and experimental results.

II. Drift Characterization

A. The Problem

Successive AFM scans of a sample without changing any of the scanning parameters will appear as translated versions of the sample surface, as shown in Fig. 1.a. This is the physical manifestation of drift in the x, y plane of the sample. There is also drift in the z direction but it will be ignored in this paper, because it usually is reasonably well compensated by the z feedback system and has little impact on nanomanipulation. Note that, unlike atoms, particles of the sizes we are discussing in this paper do not diffuse over the surface at room temperature; they are fixed with respect to the surface. Many experimental observations indicate that the drift is essentially a translation (no rotation is involved) and the drift velocity is approximately constant over periods of several minutes, but changes on a longer time scale. A drift-compensated instrument would produce the same image in each of the panels of Fig. 1.a. The problem addressed in this paper is how to achieve drift compensation in such a way that not only images of the same region are constant in time, but other processes—in particular, programmed sequences of manipulation operations—also can be carried out as if drift did not exist.

Several authors have reported simple approaches to drift compensation [4, 5, 6, 7, 8]. These approaches assume that the drift velocity is constant and compute it by comparing successive images. The major drawback of these procedures is their failure to adapt when the drift velocity changes. They can be used for correcting images taken over a period of time of approximately constant drift, but cannot support a sequence of nanomanipulation operations, which requires real-time compensation over relatively long durations.

B. Spatial Analysis

Consider a typical AFM, schematically shown in Fig. 1.b. For concreteness we assume that the sample is placed on top of the scanner and that the unloaded (i.e., undeflected) cantilever and tip (or their average positions when the AFM is operated in dynamic mode) are fixed in space, except for drift. This is the most common AFM configuration; other configurations in which the sample is fixed and the cantilever moves can be analyzed by trivial changes to the arguments below. Consider also a sample which consists of a flat surface with a very sharp asperity on it (a spatial impulse), as shown in Fig. 1.c. For simplicity we ignore the y coordinate in the following discussion and the figure. In an ideal situation, in a contact mode scan the tip remains at a fixed height touching the flat surface until it encounters the asperity. Then the scanner moves very quickly downward (it contracts) for the tip to remain in contact with the feature, maintaining the same applied force. If we record the height of the tip z_T^S in a

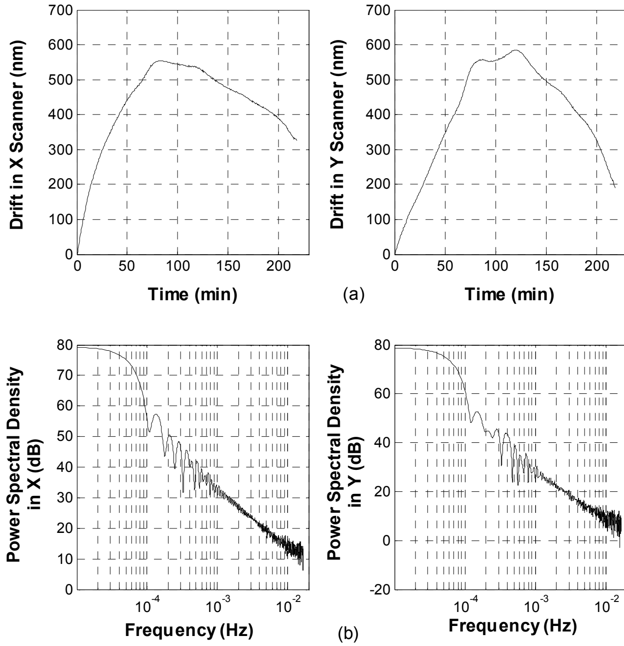


Fig. 2. (a) Measurement of drift in x and y directions over 215 minutes, with a sampling time of 30 seconds. (b) Corresponding power spectral density using the Welch method. Most of the signals' power is concentrated in the frequency band below 5×10^{-4} Hz.

coordinate system attached rigidly to the sample (or, equivalently, to the top of the scanner) as a function of the x_T^S coordinate of the tip in the same system, this gives us the true topography of the sample, which is the desired output. (A similar argument holds also for vibratory, or dynamic mode operation.)

However, in a real AFM, when the tip contacts the top of a feature such as the spatial impulse of Fig. 1.c, what we measure is the voltage V_z applied to the vertical piezo motor versus the voltage V_x applied to the horizontal motor. If we ignore (or compensate for) hysteresis and creep, and assume that the AFM is properly calibrated and compensates for non-linearities in the voltage-displacement curve, the voltages can be converted into piezo extensions E_z and E_x. But the x position of the scanner (or sample) with respect to the base of the instrument does not equal the piezo extension because there is drift between the sample and the base. Rather, the scanner position is $x_S^B = E_x + dx_S^B$, where dx_S^B is the scanner drift. (We assume the drift is a translation, based on experimental observations, as noted earlier.) In other words, even with no applied voltages, the scanner is moving (drifting) with respect to the base. In addition, the tip is itself drifting with respect to the base. Thus the position of the tip with respect to the base is $x_T^B = dx_T^B$. The position of the tip with respect to the sample, which is the desired topography signal, is the difference between the position of the tip and the position of the sample, both measured with respect to the base:

$$x_T^S = x_T^B - x_S^B = (dx_T^B - dx_S^B) - E_x = dx - E_x \qquad (1)$$

The combined effect of these two drifts, dx , is what we call simply AFM drift. To image the impulse of Fig. 1.c the tip must be on top of the feature, and therefore the tip position with respect to the sample must be constant. Since the drift varies with time, it follows from (1) that the piezo extension and corresponding applied voltage vary with time and so do the images. To stabilize the image and compensate for the drift, it suffices to change the origin of the x axis by dx . This can be done by changing the *offset* values associated with a scan, which is precisely what our system does, as we will show later. It is also clear from (1) that the drift between two instants of time t_1 and t_2 can be measured by subtracting the corresponding piezo extensions for an impulse feature. These can be read directly from the images of the feature taken at times t_1 and t_2. (In practice, the situation is more complicated because the features imaged are not pure impulses, as we will see below.)

Feedback in the horizontal directions x, y is normally used in AFMs to ensure that the scanner is in the correct position with respect to the base. Therefore, x, y feedback can compensate (within the noise level constraints of the system) for non-linearities and scanner-base drift. However, it cannot compensate for tip-base drift.

C. *Statistical Properties*

The behavior of the drift depends on such factors as temperature, humidity, the construction of the instrument, and thermal expansion coefficients. In our lab, drift velocities tend to vary from 0.01 to 0.1 nm/s. Therefore, for 256x256 pixel images taken at a 1 Hz rate (these are typical values) the drift between two successive images can be as much as 25.6 nm, which is larger than the diameter of the particles we normally manipulate.

Fig 2.a shows a time series of drift displacement values in the x and y directions, which was measured by comparing images of the same feature taken at sampling times 30 seconds apart. These measurements were performed after an initial period of two to three hours of AFM operation, to let the instrument stabilize; immediately after the AFM is turned on, the drift is considerably larger. The corresponding velocities inferred from the figures are approximately constant for several minutes, and then change in a seemingly random manner. The power spectra of the drift time series are shown in Fig 2.b and exhibit most of the signal power in the frequency band below 5×10^{-4} Hz, which corresponds to a time constant of about 30 minutes. The slow-varying character of the drift compared to the typical time required for a manipulation operation, which is at most a few seconds, makes it possible to estimate drift while performing a series of manipulations.

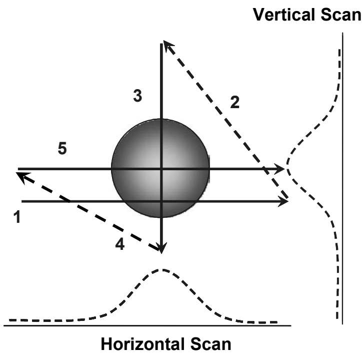


Fig. 3. Successive single line scans for finding the center of particle.

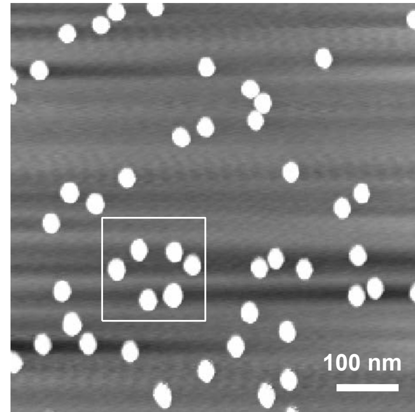


Fig. 4. The optimal tracking window. Horizontal stripes in the image are due to line-by-line flattening.

III. Estimation and Compensation of Drift

A. Drift Measurement

The basic method for the measurement of drift involves tracking a number of fixed features in the sample. We employ two methods depending on the sample type.

The first method is applicable only when the objects to be manipulated have a known and simple shape. This is usually the case in our experiments, in which we manipulate spherical particles. For these particles, drift can be measured by tracking the center of a particle.

We search for the center of a spherical particle as follows. First we look for the highest point of a single line scan in the x direction (see Figure 3). Then we scan along a single line in the y direction and passing through the previously-found high point. We find the highest point of this y-scan, pass an x-line through it and find a new maximum, continuing the process until we reach a desired accuracy. (This process fails if the first line scan misses the particle altogether.)

The second method uses a general approach, applicable to objects of arbitrary shapes, and involves correlating images, by using the three following steps.

1) Selection of tracking window: We typically use a 64x64 window, which can be scanned in a few seconds and normally contains enough features for successful tracking. The area is selected by maximizing an *interest* operator, defined in terms of the following characteristics [9, 14].

- Distinctness of features from immediate neighbors.
- Global uniqueness of the features (in the whole image).
- Invariance of features under expected distortions.

In order to evaluate the optimality of the selected window in terms of these criteria, we first construct the matrix

$$N = \sum_{i=1}^{m} \begin{bmatrix} f_{x_i}^2 & f_{x_i} f_{y_i} \\ f_{y_i} f_{x_i} & f_{y_i}^2 \end{bmatrix} \tag{2}$$

where f_{x_i} and f_{y_i} are the gradients of the image in the x and y directions, and the sum is over all the pixels in the window. Assuming a constant noise level in the image, the optimum window is selected in regions in which the eigenvalues of N are approximately equal to each other. This ensures that the selected window does not contain a straight edge or a strongly oriented texture. We also require that the confidence ellipse (defined by *trace* N^{-1}) be smaller than those obtained from neighboring windows. This ensures maximum local separability or distinctness of points in the selected window (for more details see [14]). Figure 4 shows a standard scan of nanoparticles and the optimum selected tracking window.

2) Coarse computation of translation: The normalized cross-correlation between two images is a good measure of their similarity. It is defined mathematically as:

$$c(dx,dy) = \frac{\sum I_1(x+dx, y+dy) \cdot I_2(x,y)}{\sqrt{\sum I_1^2(x,y)} \cdot \sqrt{\sum I_2^2(x,y)}} \tag{3}$$

where I_1 and I_2 are the two images, assumed to have zero mean, and the summations are over the discretized values of x and y. If I_2 is a perfect translation of I_1 by (a,b), then the cross-correlation exhibits a peak at $(dx,dy)=(a,b)$, and the peak value is 1. In this case, the rest of the points in $c(dx,dy)$ take values between 0 and 1, depending on how well the two images match at each (dx, dy) translation. In general the match is not perfect, and we use the dx, dy values that correspond to the maximum value of the correlation function as the measured translation, and the peak value as an indication of how well the two images match. The cross-correlation is computed efficiently in the frequency domain, by using the Fast Fourier Transform (FFT).

3) Fine computation of translation: The cross-correlation computation is done with pixel accuracy, which may not be sufficient for successful nanomanipulation. Sub-pixel accuracies can be obtained by the following procedure [9]. First the cross-correlation method is used to find a coarse value for the translation, (dx_0, dy_0), say. Then $I_2(x, y)$ is expressed as a first-order expansion of I_1:

$$I_2(x,y) = I_1(x+dx_0, y+dy_0) + \frac{\partial I_1}{\partial x} \cdot \Delta dx + \frac{\partial I_1}{\partial y} \cdot \Delta dy + n \tag{4}$$

where n is a noise term that includes higher-order effects, and Δdx and Δdy are subpixel translations. Next, a best estimate for the deltas is computed by least square estimation over the entire picture. Finally, the refined estimate for the translation is given by $dx = dx_0 + \Delta dx$ and $dy = dy_0 + \Delta dy$. In our experience, by using this method accuracies in the order of $1/10^{th}$ of a pixel dimension can be easily achieved (for example, if we scan a 1 μm area in 256×256 pixel size, the drift can be measured with an error less than 0.4 nm).

Observe that both techniques for measuring drift require a coarse estimate of the drift value. Without an approximate position for the spherical particle whose position we want to measure, the search procedure may fail, or produce grossly wrong values, because the first single-line scan may well miss the particle altogether, or, even worse, hit a different particle. The correlation-based technique also may fail if the images in the selected windows are too different. This may produce low correlation values or even spurious peaks that do not correspond to the translation we want to find.

The particle-center search procedure is very fast. Travelling at ~ 1 μm/s, which is a relatively slow speed, good estimates require less than a second for particles ~ 10 nm. The correlation method takes longer, on the order of 10 s for a 64x64 pixel window, but is more generally applicable.

B. Dynamic Model of Drift

We need a dynamical model of the drift to be able to estimate and predict it by using Kalman filtering techniques. We assume decoupled dynamics for the drift in x and y directions. Although in reality (especially in tube shape scanners) there is some correlation between the two directions, our assumption greatly simplifies the compensation system design and implementation and produces satisfactory results, as we will show later in section IV.E.

The drift behavior is very similar to that of a maneuvering target, in which the velocity is approximately constant for a relatively large amount of time, and then changes randomly. A suitable model for such targets was introduced by Singer for radar tracking of manned air vehicles [12]. Singer's model uses an acceleration that is correlated in time. Intuitively, this implies that if a target is accelerating at a time t, it is likely to be still accelerating at a time $t+\tau$, for sufficiently small τ. We model the drift acceleration $a(t)$ by a first-order Markov process governed by the first order differential equation

$$\dot{a}(t) = -\alpha\, a(t) + w(t) \tag{5}$$

with a corresponding exponential auto-correlation

$$R(\tau) = E\left[a(t)a(t+\tau)\right] = \sigma_m^2 e^{-\alpha|\tau|} \quad \alpha > 0\,. \tag{6}$$

Here σ_m^2 and $1/\alpha$ are the variance and time constant of acceleration, respectively, and $w(t)$ is white noise with a variance $2\alpha\sigma_m^2$.

Thus the state space formulation for the drift in the x direction is

$$\begin{aligned} \dot{x}(t) &= v(t) \\ \dot{v}(t) &= a(t) \\ \dot{a}(t) &= -\alpha\, a(t) + w(t) \end{aligned} \tag{7}$$

where $x(t)$ and $v(t)$ are the drift displacement and velocity, respectively. (Similar equations apply to the y direction in this uncoupled system.)

The corresponding discrete-time equations for a sampling period T are

$$\mathbf{x}(k+1) = F\mathbf{x}(k) + \mathbf{u}(k) \tag{8}$$

where $\mathbf{x}(k) = [x(k)\, v(k)\, a(k)]^T$, $\mathbf{u}(k)$ is a 3×1 process noise, and the transition matrix F given by

$$F = \begin{bmatrix} 1 & T & (\alpha T - 1 + e^{-\alpha T})/\alpha^2 \\ 0 & 1 & (1 - e^{-\alpha T})/\alpha \\ 0 & 0 & e^{-\alpha T} \end{bmatrix}. \tag{9}$$

In this model, the dynamics of drift can be expressed in terms of three parameters: the variance, or magnitude, of drift acceleration, the time constant, and the sampling interval.

C. Kalman Filter Estimation of Drift

The state of the drift defined in (8) can be estimated recursively, by Kalman filtering. (We consider only the x direction, because the system is decoupled; a similar treatment applies to the y direction.) Given the current state estimate $\hat{\mathbf{x}}(k-1\,|\,k-1)$ and the state error covariance $P(k-1\,|\,k-1)$, the filter predicts the state and covariance at the next time step by the standard equations

$$\hat{\mathbf{x}}(k\,|\,k-1) = F\,\hat{\mathbf{x}}(k-1\,|\,k-1) \tag{10}$$

$$P(k\,|\,k-1) = F\,P(k-1\,|\,k-1)\,F^T + Q(k-1) \tag{11}$$

Here Q is the covariance of the process noise, which can be computed in terms of T, α and σ_m^2—see [11, 12] for the actual expressions.

The measurement is modeled as a linear combination of drift state corrupted by uncorrelated noise, i.e.

$$z(k) = H\,\mathbf{x}(k) + \nu(k) \tag{12}$$

in which H = [1 0 0] and $\nu(k)$ is a sequence of white noise, independent of the process noise $\mathbf{u}(k)$, and with covariance $R(k)=R_0\,\delta(k)$.

The drift displacement is measured by the techniques discussed earlier, in section III.A. We use the pre-measurement estimate of the drift to change the origin of the coordinates when we position the first scan line (for the sphere center search) or the window used for the cross-correlation computation. This change of origin ensures that we don't miss the particle in the first line scan, or that the windows used in the correlation method are not too different. Thus the measured value is given by

$$z(k) = z(k-1) + dz + \delta \tag{13}$$

where δ is the distance between the origins of coordinates used to make the two measurements, and is given by

$$\delta = H[\hat{\mathbf{x}}(k\,|\,k-1) - \hat{\mathbf{x}}(k-1\,|\,k-2)] \tag{14}$$

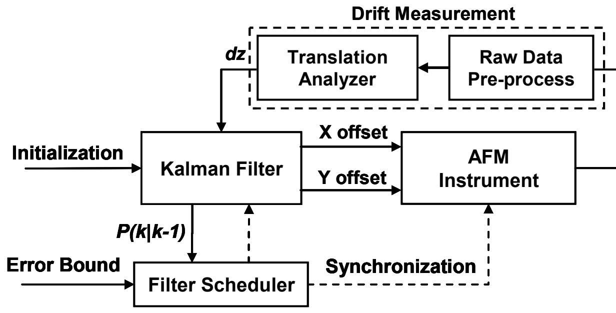


Fig. 5. Block diagram of the AFM and the drift compensator.

and dz is the amount of measured drift as explained in III.A. The Kalman filter computes the Kalman gain and uses the measurement $z(k)$ to update the state estimate $\hat{\mathbf{x}}(k \mid k)$ and the error covariance $P(k \mid k)$ by the standard formulas [11].

IV. Implementation and Results

A. System Architecture

The overall system architecture is shown in Fig. 5. The filter maintains current estimates of drift displacement and error covariance. The drift displacement is written onto the offset registers of the controller. This is equivalent to a change of origin. For example, if the x offset is 20 nm and a tip motion to x = 10 nm is requested, the controller applies to the x piezo the voltage required to move the tip by 30 nm. If the drift estimate was perfect, this procedure would ensure that successive requests for scans from x = 0 to x = 500 nm, say, would produce always the same image. The filter also maintains the last acquired image or particle location (depending on which drift measurement technique is used), together with the corresponding offsets and sampling times.

The image or scan line data that result from a measurement job are first pre-processed. The raw data from the tracking window are flattened line by line to remove the existing slope. Then a threshold is applied to the image for discarding the artifacts due to the image flattening. Finally the whole data from the window are offset to a zero mean. In the case of tracking a particle using its center, each single line scan is smoothed with a fourth order IIR filter.

After the raw data are processed they are passed to the translation analyzer, which compares them with the previous image or particle position and computes the displacement, taking into consideration the changes of origin associated with the offsets. The Kalman filter uses the measured drift and the current estimates to update the state and the covariance.

The scheduling of filter operation is performed according to the flowchart of Figure 6. Requests for drift measurements are made from the compensator to the AFM (see bottom of flow chart), and the AFM acknowledges the request. This indicates that a time slot has been assigned to the filter to perform any operation necessary for the drift measurement, i.e., either an imaging scan or a series of line scans. Observe that drift measurement requires tip motion, and therefore has to be

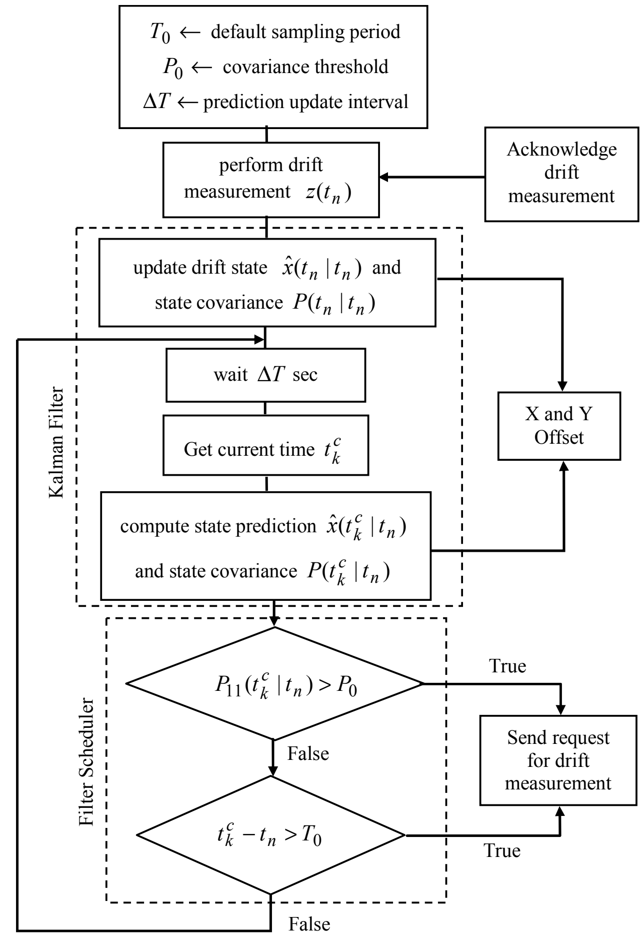


Fig. 6. Synchronization of the drift compensator with the AFM.

scheduled by the AFM controller after the end of any executing process that also moves the tip. The Kalman filter starts the drift measurement procedure once it receives the acknowledgement from the AFM. The current estimates of drift are updated by using the measurement and provided as new x and y offsets to the AFM. From this point until the next measurement, the filter uses the prediction equations (10) and (11) to compute the most up-to-date estimates of x and y offsets at times t_k^c, a given sampling interval apart. Typically, this *prediction update* sampling period is on the order of a few seconds and much smaller than the default *measurement* sampling interval T_0, which can go from 30 sec to several minutes. During the inter-measurement period, the error covariance is also monitored (as a measure of uncertainty in the estimates) and once it exceeds a user-specified threshold a request for a new measurement is sent to the AFM machine. Otherwise, the measurements are scheduled at the periodic default measurement times, T_0 apart.

B. Hardware and Software

The system is implemented on an AutoProbe CP-R AFM (Park Scientific Instruments, now Veeco Instruments). The drift compensation software runs on top of our own Probe Control Software (PCS) for nanomanipulation, which in turn

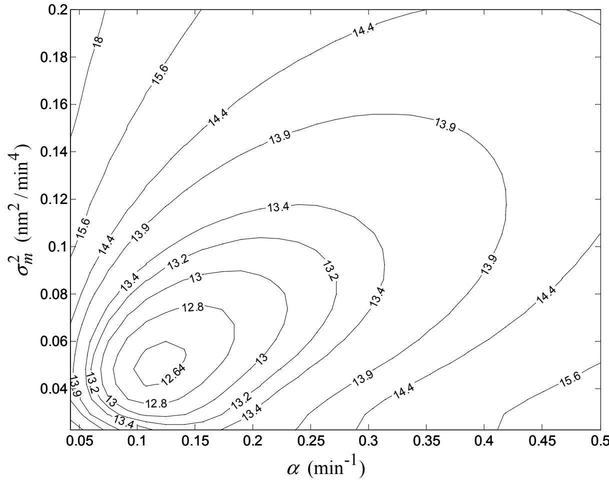


Fig. 7. A contour plot of isolines of the cost function (16) versus σ_m^2 and α. The minimum of J_N occurs at σ_m^2=0.048 nm^2/min^4 and α=0.11 min^{-1}.

is implemented through the vendor supplied API (Application Programming Interface). (A version of PCS is now commercially available.) The API maintains its own job queue, with no preemptive scheduling. Before any job that involves tip motion is executed, we check for updated offset values, to ensure that we compensate for drift in all imaging and manipulation operations.

C. Selection of Drift Model Parameters

For a fixed-order Kalman filter to be optimal (in the sense of having minimum error covariance) the filter parameters have to be selected as close as possible to the real signal's parameters. These parameters are measurement noise covariance R_0, acceleration time constant 1/α and acceleration noise variance σ_m^2. Any mismatch between the filter parameters and the real parameters results in suboptimality of the Kalman filter gain [10].

The estimation of measurement noise covariance R_0 is based on the autocorrelation of the innovation sequence [15]. The innovation sequence is defined as

$$\varepsilon(k) = z(k) - H\hat{x}(k \mid k-1)\,. \tag{15}$$

The innovation process for an optimal Kalman filter is white Gaussian noise. Using the algorithm of Mehra [15, 16], the autocorrelation of the innovation process is used to estimate R_0 once the filter reaches a steady-state condition. This can be checked by looking at the state covariance, and in our case typically is achieved in 5-10 min. By using this method, the value of R_0 is estimated to be about 1.2 nm^2 for the set of data in Fig. 2.a.

The estimation of σ_m^2 and α is more complicated. Since the number of samples in drift measurements is small (the typical sampling time is in the range of 30-120 seconds before the filter reaches the steady state condition, and thereafter may be higher) we have run a simulation for different values of σ_m^2 and α using the measurement data in Fig 2.a The criterion for choosing these two parameters is the minimization of the

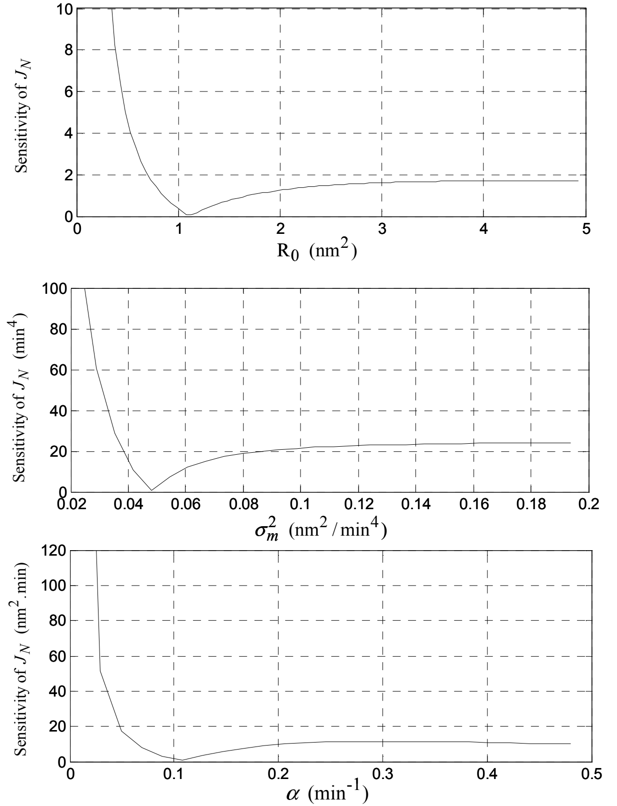


Fig. 8. The sensitivity of J_N to parameters R_0, σ_m^2 and α. In each curve the other two parameters have been kept constant at their nominal values of R_0=1.2 nm^2, α=0.11 min^{-1} and σ_m^2=0.048 nm^2/min^4.

mean square of the prediction errors, i.e.,

$$J_N(\alpha, \sigma_m^2, R_0) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon^2(k)\,. \tag{16}$$

The procedure for the selection of σ_m^2 and α is as follows.

1) Take a series of drift measurements with some small sampling time (30 sec in our case).

2) Divide these samples into a set of measurement-prediction intervals in such a way that each measurement interval is followed by a prediction interval (in our case we used 5 min measurement and 10 min prediction). Run a filter simulation in which we use measurements 30 sec apart for 5 min, do not measure for 10 min (use prediction only), then measure again for 5 min at 30 sec intervals, and so on. Do this for various values of the parameters,

3) With the computed R_0 from the Mehra's method, search for the minimum value of (16). Figure 7 shows the contour plot of isolines of J_N and its minimum at σ_m^2=0.048 nm^2/min^4 and α=0.11 min^{-1}.

D. Sensitivity of Parameters

The sensitivity of the Kalman filter with respect to a parameter θ is defined as

$$S_\theta = \left|\frac{\partial J_N(\theta)}{\partial \theta}\right| \tag{17}$$

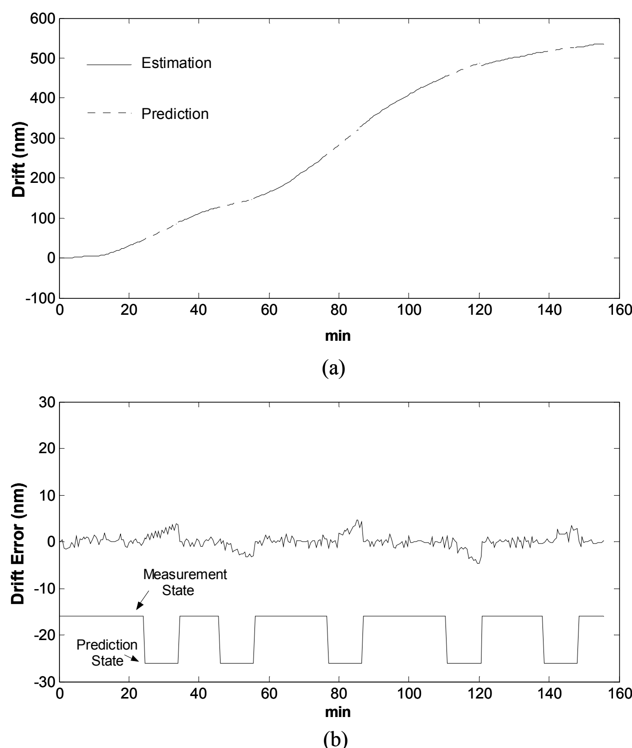


Fig. 9. (a) Drift measurements in the x direction with 30 sec sampling period. (b) Drift error in the x direction, defined as the difference between estimated and measured values. The measurements are used to estimate the state of drift during the time intervals labelled “Measurement State” at the bottom of the figure, whereas during the “Prediction State” intervals the estimates are predictions only.

where θ can be any of the parameters R_0, α or σ_m^2 while the other two parameters are kept constant at their nominal values which correspond to the minimum of the cost function (see section IV.C). Theoretically, the sensitivity vanishes at the extremum values of the cost function (16).

By studying the sensitivity curves in Fig. 8, it is clear that choosing the parameters below their nominal values has a more negative impact on the performance than selecting higher values. However, it should be noted that higher parameter values imply a larger prediction error (16).

E. *Experimental Results*

The following results were obtained on the AutoProbe AFM with a sharp tip in dynamic mode, imaging and manipulating gold nanoparticles with nominal diameters of 15 nm, deposited on a mica surface, covered with poly-L-lysine, in air, at room temperature and humidity. Filter parameters were selected at the nominal values computed as explained above.

In order to assess the tracking capability of the filter we conducted two experiments. In the first experiment we measured the drift with the method explained in section III.A at time intervals 30 sec apart. At specific time instants several min apart, we switched the filter from a measurement state (in which the measurements are used to update the drift estimate) to a prediction state (in which the measurements are ignored and the state is updated by prediction only), and vice-versa. Figure 9.a depicts the measured values of drift in the x direction. Note that over a period of 160 min the total drift was over 500 nm or some 40 times the diameter of the particles being manipulated. The variance of the residuals, defined as the difference between the estimates and the measurements, during the measurement periods is about 1.14 nm^2. The drift error (residual) is plotted in Fig. 9.b, which shows a growing error once the filter is switched to the prediction state. This deviation depends on the amount of measurement time before the prediction and on the drift behavior. However, the prediction error is not growing faster than 0.5 nm/min while the speed of the drift in the same experiment can be as high as 6 nm/min. This implies an order of magnitude improvement in drift speed in a pure prediction mode.

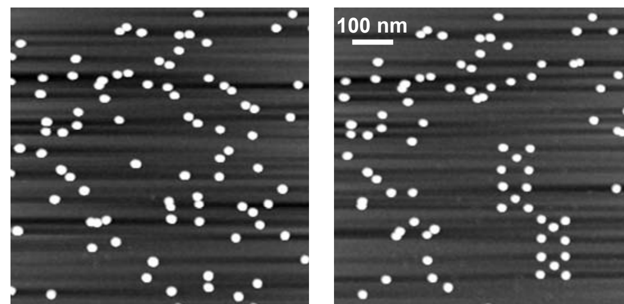


Fig. 10. Manipulation of 15 nm Au particles using the drift compensator and an automatic manipulation procedure. The drift measurements are only carried out before the manipulation of each set of five nanoparticles. The sampling time for the measurements is 30 sec and the manipulation of each particle takes about 40 sec, which includes waiting time to combat creep, plus time for the particle-center search. The left panel is the initial, random configuration, and the right panel is the final structure, proposed in [17] for a QCA inverter.

In the second experiment, we used the drift compensator for the automatic nanomanipulation of particles. Manipulation needs accurate estimates for the position of particles, especially when their diameters are below ~ 15 nm. The procedure for integrating the drift compensator and the automatic manipulator is the following. We first scan and choose an appropriate area in the sample. The desired particles and their destinations are marked in the acquired sample area. The drift compensator is started with a default sampling time (30 sec) and it continues running until it reaches a stable condition (typically after 5-10 minutes). Then the AFM starts the manipulation process by using the prediction states provided by the Kalman filter. The manipulation process is terminated once the uncertainty in the drift estimate exceeds a certain limit, and new measurements are performed. Manipulation continues after the measurements, and so on.

Figure 10 shows the automatic manipulation of particles using the drift compensator. In this experiment the drift measurements are only carried out before the manipulation of each set of five nanoparticles. Before each manipulation the tip waits about 30 seconds (to combat the effects of creep) at the approximate location of a particle computed using the drift value predicted by the filter. Then it searches for the center of

the particle (using the center-search procedure discussed in Section III.A and Figure 3) in an area comparable to the size of the particle. After that, the particle is manipulated. Because large-distance manipulation motions typically are less accurate than short ones, we ran the whole process of manipulation twice, the first for a gross displacement of particles and the second for accurately placing the particles at their target positions. The final result is the structure proposed in [17] for a Quantum-dot Cellular Automaton (QCA) inverter. Similar experiments without drift compensation failed: particles were missed or moved to the wrong locations.

The computational time for the filter operation in these experiments is associated primarily with scanning of the tracking window (about 8 sec for a 64×64 pixel window) and processing of the resulting image (less than 2 sec). The compensator is capable of predicting the drift state with an error less than 0.5 nm/min for at least 10 minutes. Therefore, drift compensation can be run as a background process while the AFM executes other tasks such as the manipulation of nanoobjects.

V. Conclusion

Drift is a major cause of spatial uncertainty in AFMs. It causes distortion in AFM images, and it has even more deleterious effects on nanomanipulation, where it is often responsible for outright failure of the desired operations. Drift compensation in today's instruments is done primarily through user interaction.

This paper presents a Kalman filtering approach to drift estimation. The filter updates the origin of the AFM coordinates, scheduling measurements when the covariance of the error exceeds a given threshold. AFM tip motions are always executed in the updated coordinate system and become largely immune to drift.

Sequential manipulation of nanoparticles over a relatively long period of time without user intervention is demonstrated experimentally. By using drift compensation it will be possible, for the first time, to perform manipulation tasks that last several hours without a user in the loop, and to construct by AFM manipulation nanostructures much more complex and useful than those which have been built until now.

Future work in this area includes more accurate drift measurement methods (e.g., by removing the effect of tip convolution from the images), and compensation for other sources of spatial uncertainty such as creep and hysteresis. Coupling the results of this research with high-level planning algorithms such as those in [18] will lead to the fully automatic construction of complex nanoassemblies.

References

[1] A. A. G. Requicha, "Nanorobots, NEMS and Nanoassembly", *Proc. IEEE*, Vol. 91, No. 11, pp. 1922-1933, November 2003.

[2] A. A. G. Requicha, "Nanorobotics", in S. Nof, Ed., *Handbook of Industrial Robotics*. New York, NY: Wiley, 2nd. ed., pp. 199-210, 1999.

[3] http://www.asylumresearch.com/.

[4] V. Y. Yurov and A. N. Klimov, "Scanning tunneling microscope calibration and reconstruction of real image: Drift and slope elimination", *Rev. Sci. Inst*., Vol. 65, No. 5, pp. 1551-1557, May 1994.

[5] R. Staub, D. Alliata and C. Nicolini, "Drift elimination in the calibration of scanning probe microscopes", *Rev. Sci. Inst,*. Vol. 66, No. 3, pp. 2513-2516, March 1995.

[6] J. T. Woodward and D. K. Schwartz, "Removing drift from scanning probe microscope images of periodic samples", *J. Vac. Sci. Technol. B*, Vol. 16, No. 1, pp. 51-53, January/February 1998.

[7] S. H. Huerth and H. D. Hallen, "Quantitative method of image analysis when drift is present in a scanning probe microscope", *J. Vac. Sci Technol. B*, Vol. 21, No. 2, pp. 714-718, March 2003.

[8] K. J. Ito, Y. Uehara, S. Ushioda and K. Ito, "Servomechanism for locking scanning tunneling microscope tip over surface nanostructures", *Rev. of Sci. Inst*., Vol. 71, No. 2, pp. 420-423, February 2000.

[9] R. Haralick and L. Shapiro, *Computer and Robot Vision*. Reading, MA: Addison-Wesley Inc., Vol. 2, 1993.

[10] A. Gelb, *Applied Optimal Estimation.* MIT Press, 2001.

[11] Y. Bar-Shalom and Rong Li X., *Estimation with Application to Tracking and Navigation*. New York, NY: John Wiley & Sons Inc., 2001.

[12] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets", *IEEE Trans. on Aerospace and Electronic Systems*, Vol. 6, No. 4, pp. 473-483, July 1970.

[13] A. A. G. Requicha, C. Baur, A. Bugacov, B. C. Gazen, B. Koel, A. Madhukar, T. R. Ramachandran, R. Resch and P. Will, "Nanorobotic assembly of two-dimensional structures", *Proc. IEEE Int'l Conf. on Robotics & Automation*, Leuven, Belgium, pp. 3368-3374, May 16-21, 1998.

[14] W. Förstner, E. Gülch, "A fast operator for detection and precise locations of distinct points, corners, and centers of circular features", *Proc. of the Intercommission Conference on Fast Processing of Photogrammetric Data*, 1987, pp. 281–305.

[15] R. K. Mehra, "On the identification of variances and adaptive Kalman filtering", *IEEE Trans. on Automatic Control,* Vol. 15, No. 2, April 1970, pp. 175-184.

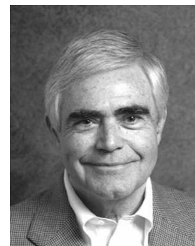
[16] R. K. Mehra, "On-Line identification of linear dynamic systems with applications to Kalman filtering", *IEEE Trans. on Automatic Control,* Vol. 16, No. 1, February 1971, pp. 12-21.

[17] C. S. Lent *et al.,* "Quantum cellular automata", *Nanotechnology*, Vol. 4, Jan. 1993, pp. 49-57.

[18] J. H. Makaliwe and A. A. G. Requicha, "Automatic planning of nanoparticle assembly tasks", *Proc. IEEE Int'l Symp. on Assembly & Task Planning* (*ISATP '01*), Fukuoka, Japan, pp. 288-293, May 28-30, 2001.

Babak Mokaberi received the B.S. degree from Isfahan University of Technology, Isfahan, Iran in 1995 and M.S. degree from Sharif University of Technology, Tehran, Iran in 1998 in Electrical Engineering. In 2001 he joined the University of Southern California, Los Angeles where he is currently a Ph.D. candidate, in Electrical Engineering.

His research interests include dynamical system control of atomic force microscopes for manipulation of nanometer-scale objects. He is currently working in the Laboratory for Molecular Robotics, University of Southern California.



Aristides A. G. Requicha (Fellow, IEEE) was born in Monte Estoril, Portugal, in 1939. He received the Engenheiro Electrotécnico degree from the Instituto Superior Técnico, Lisbon, Portugal, in 1962, and the Ph.D. in electrical engineering from the University of Rochester, Rochester, NY in 1970. He was a college and high school Valedictorian.

He is currently the Gordon Marshall Professor of Computer Science and Electrical Engineering at the University of Southern California, where he also directs the Laboratory for Molecular Robotics. He has authored some 150 scientific papers, and has served in numerous conference program committees and journal editorial boards. His past research focused on geometric modeling of 3–D solid objects and spatial reasoning for intelligent engineering systems.

Currently he is working on robotic manipulation of nanometer-scale objects using scanning probe microscopes; nanorobot components and nanorobotic system integration; fabrication of nanostructures by robotic self-assembly; sensor/actuator networks; and applications in NEMS (nanoelectromechanical systems) and nanobiotechnology. The long-term goals are to build, program, and deploy nanorobots and networks of nanoscale sensors/actuators for applications to the environment and health care.

Dr. Requicha is currently a member of the IEEE Nanotechnology Council AdCom, representing the Robotics and Automation Society (RAS), and he also co-chairs the Micro and Nanorobotics Technical Committee of the RAS. He is also a member of the AAAS, ACM, AVS and SME. Personal web page: http://www-lmr.usc.edu/~requicha.