Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Real-time State Estimation and Fault Detection for Controlling Atomic Force Microscope Based Nano Manipualtion [⋆]

Lianqing Liu [*,**,****] **Ning Xi** [**] **Yilun Luo** [**] **Jiangbo Zhang** [**]
Guangyong Li [***] **Yuechao Wang** [*]

[*] *Robotics Laboratory, Chinese Academy of Sciences, Shenyang, Liaoning Province, 110016 China*
[**] *Dept.of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, 48824 USA*
[***] *Dept.of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA, 15621 USA*
[****] *Graduate school of Chinese Academy of Sciences, Beijing 100001, China*

Abstract: The main problem of Atomic Force Microscope (AFM) based nanomanipulation is the lack of real-time visual feedback. Although the model based visual feedback can partly solve this problem, the incorrect display caused by the uncertainties in the nano-environment often leads to a failed nanomanipulation. In this paper, a general strategy with three-level structure is proposed to overcome this problem. With this three-level strategy, the incorrect display can be not only real-time detected, but also on-line corrected. The difficulty to implement this strategy is that there is no continuous way to describe and model the system since both discrete and continuous commands are involved. A Petri-net based method is proposed to organize this strategy such that task scheduling, which usually deals with discrete events, as well as task planning, which usually deals with continuous events can be treated in a unified framework. This Petri-net organized strategy provides general instructions to AFM based manipulation for displaying a visual feedback which can be as close as possible to the true environment. The experimental results presented in the paper demonstrate the advantage of the proposed strategy. It also shows the increased efficiency of the AFM based nanomanipulation.

1. INTRODUCTION

Atomic Force Microscope (AFM) (Binning (1986)) has been proved to be a powerful tool to characterize the roughness, grain size or features of surface in atomic scale. A typical AFM consists of an extremely sharp tip mounted or integrated on the end of a tiny cantilever which is driven by a mechanical scanner over the sample surface to be observed. The variation of the surface height varies the force acting on the tip and therefore varies the bending of the cantilever. This bending can be measured with a laser spot reflected from the top of the cantilever into an array of photodiodes. The surface information can be obtained by the electronic control device based on the measured bending signal.

In addition to characterize sample surfaces, the extremely sharp AFM tip can also be used as an end effector to modify the sample surface. Researchers use AFM to change the sample surface through nanolithography (Schaefer (1995), Wang (1995)) and nanomanipulation (Kim (1992), Luthi (1994), Junno (1995), Stark (1998)) taking advantage of its high resolution and high alignment accuracy. Since AFM collects image data by feeling rather than looking, the manipulation process can not be monitored in real-time due to the slow scanning speed. The operation result has to be verified by a new image scan after each step of manipulation, which often takes several minutes. Obviously, this scan-design-manipulation-scan cycle is very time-consuming and inefficient. To enhance the efficiency of AFM based manipulation, some virtual reality based methods have been developed. A 3-D virtual reality visual feedback is built from the static AFM image in Sitti (1998) and Guthold (2000). Although a virtual reality can display a static virtual environment and a dynamic tip position, it does not display the environment changes during manipulation. The operator is still 'blind' because the environment changes are not updated in real time. To solve this problem, an augmented reality system has been developed in Li (2003). With this method, a movie-like visual feedback is provided, the operator can manipulate nano-objects under the assistance of force and visual feedbacks. But this kind of real-time visual display is still not the real manipulation result, but the calculated real-time changes of the true environment from the objects' behavior models. Due to the complexities and uncertainties of the nano-world, such as surface tension, Van der Waals force, capillary force and so on, it is impossible to accurately describe the object's behavior with a mathmatical model. Thus there is mismatch between the model based visual display and the real environment. As a result, the false visual feedback leads to a failed manipulation. Although

[⋆] This work was supported in part by the National Science Foundation under grants CMMI-0115355, and IIS-0713346.

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.2546

the mismatch can be detected through force feedback technology (Li (2003), Zhang (2006)), a new AFM image scan is still needed to verify and correct the error. From above discussion we can see, there is no available method that can provide a real-time visual feedback which can matches the true environment very well during manipulation.

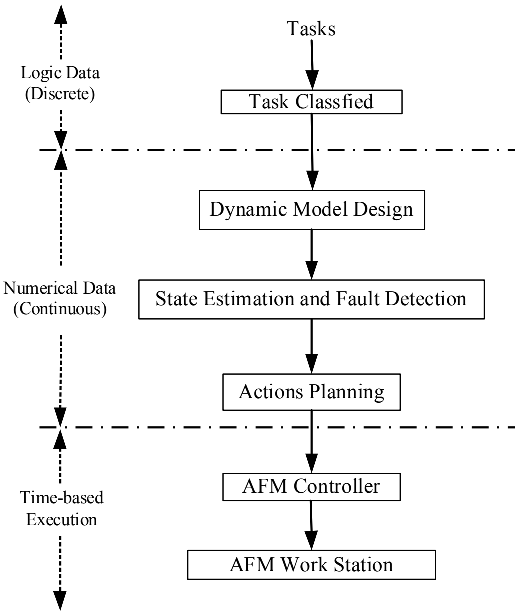


Fig. 1. A strategy with three-level structure for detection and real-time correction of faulty visual feedback in AFM based nanomanipulation.

A strategy based on real-time fault detection and correction provides a feasible way to overcome this problem. This strategy can be mainly divided into three levels as shown in Fig. 1. The first level is to classify the tasks based on the shape of the object. Some logical variables are generated in this level to represent different manipulation objects such as particle, rod, wire and so on. The second level is in charge of state estimation and fault detection. A kalman filter is developed in this level to provide a state estimation and judge whether there is a faulty display, based on which the desired AFM actions are planned. For example, if no fault is detected, the manipulation will be continued, else the manipulation is suspended and the tip's trajectory is designed to take an AFM scan and get the true manipulation result. The continuous commands are generated and sent to the low level controllers which drives the AFM tip. The final level is time-based execution, after receiving the motion commands, AFM controller drives the AFM tip following the planned trajectory. In this level, the action of manipulation or scanning is performed. Through these three levels, the faulty display can be detected and corrected on-line. One thing needs to be further addressed is the scan process. Since only part of the surface area is modified during manipulation, the whole image scan is not necessary, and local scan is enough to get the true manipulation result. Since local scan only needs take several lines scan, it can be finished in a very short time, and a movie-like visual feedback matching the true environment can be provided in real-time, which greatly facilitates the AFM based nanomanipulation. One major difficulty to design such a strategy is that there is no a unified model or framework which could describe both discrete and continuous events together. Here Petri net is used to organize this strategy. Through adopting Petri net, task classify, dynamic model design, state estimation, fault detection, and hardware execution can be handled in a unified framework. The experimental results presented in the paper demonstrate the advantage of the proposed strategy, which also shows the increased efficiency of the AFM based nanomanipulation.

2. METHOD FOR FAULT DETECTION AND CORRECTION

The functions of the three levels have to be well implemented to get an reliable visual feedback. How to classify manipulation tasks has been presented in Li (2005). The functions of the third level can be well performed by the commercial AFM system. So in this section, we mainly address the implementation of the second level, such as the dynamical model design, the development of Kalman filter and the trajectory planning of local scan.

2.1 System Dynamic Model

The object's dynamic model is a prerequisite for fault detection by using Kalman filtering techniques. To make our point clearer, a task of manipulating nano-particles will be used to explain this strategy. The strategy can be expanded to manipulate other types' objects by adopting corresponding dynamic models. To simply the analysis, the motion of the tip and objects is decomposed into horizontal motion and vertical motion along the fast scan direction and slow scan direction respectively. The dynamic model of the nano-particle in horizontal direction during manipulation can be expressed as:

$$\ddot{x} + k\dot{x} = \lambda F_x \tag{1}$$

Here x is the relative displacement to the start pushing point in horizontal direction, F_x is the corresponding force measured from the AFM tip, k is the damping coefficient of the dynamic system, λ is the proportional gain for adjusting the force. k and λ can be obtained through experiments. A similar equation can be written for the vertical direction. The state space representation of the nano-particle in horizontal direction is:

$$\begin{cases} \dot{x} &= v_x \\ \dot{v}_x &= \lambda F_x - k v_x \end{cases} \tag{2}$$

Here x, v_x are displacement and velocity of the particle in horizontal direction respectively. The corresponding discrete-time equations in horizontal and vertical direction with a sampling period T are:

$$\begin{cases} X_{k+1} = AX_k + BU_k \\ Z_k = CX_k \end{cases} \tag{3}$$

Where $X(k) = [x(k), y(k), \dot{x}_k, \dot{y}_k]^T$, $U_k = [0, 0, f_k^x, f_k^y]^T$, here f_k^x and f_k^y is the interaction force in horizontal and vertical direction respectively. Z_k is the states measurement from the visual display, the observation matrixes C is a 4×4 unit matrix, state transition matrix A and control input matrix B are given as:

$$A = \begin{bmatrix} 1 & 0 & \frac{1}{k}(1-e^{-kT}) & 0 \\ 0 & 1 & 0 & \frac{1}{k}(1-e^{-kT}) \\ 0 & 0 & e^{-kT} & 0 \\ 0 & 0 & 0 & e^{-kT} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & \frac{\lambda}{k}(T-\frac{1}{k}+\frac{1}{k}e^{-kT}) & 0 \\ 0 & 0 & 0 & \frac{\lambda}{k}(T-\frac{1}{k}+\frac{1}{k}e^{-kT}) \\ 0 & 0 & \frac{\lambda}{k}(1-e^{-kT}) & 0 \\ 0 & 0 & 0 & \frac{\lambda}{k}(1-e^{-kT})) \end{bmatrix}$$

2.2 Kalman Filter Based State Estimation and Fault Detection

With the object's dynamic model, it is now possible to test whether the visual display agrees with the real-time changes in the nano environment or not by using Kalman filter technology. An estimation $X_{k/k-1}$ and an associated covariance matrix $P_{k/k}$ are calculated in each step. The innovation e_k is calculated as:

$$e_k = Y_k - CX_{k/k-1} \tag{4}$$

Here Y_k is the state information from the visual display. In the ideal case, without error the innovation e_k from the Kalman filter should be Gaussian and white, its associated covariance $R_e = e_k e_k^T$ keeps a value near to zero. When the visual display does not agree with the estimation, the innovation e_k will not be Gaussian nor white, and its associated covariance R_e will have an abrupt increase. R_e is calculated as:

$$R_e = E[(Y_k - CX_{k/k-1})(Y_k - CX_{k/k-1})^T] \tag{5}$$

Substituting $Y_k = CX_k + V_k$ into (5) (here V_k is the measurement noise with the covariance R and independent of X_k). Equation (5) can be rewritten as:

$$R_e = P_{k/k} + R \tag{6}$$

R_e can be directly used to judge whether there is a inaccurate display or not by setting a threshold on it, but the performance of this measure is less of robustness and a false alarm signal is frequently triggered. One of the robust measures is Mahalanobis distance. The Mahalanobis distance of R_e is calculated as:

$$M_t = e^T R_e^{-1} e \tag{7}$$

A threshold on M_t is set to give an alarm signal when a fault happens. When the visual display loses match with the real environment, M_t will have an abrupt increase and overshoot the threshold. Then an alarm signal will be triggered to report the error to the system.

2.3 Trajectory Planning of Local Scan

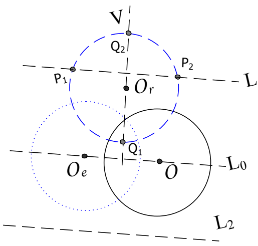


Fig. 2. Local scan pattern for getting the real position of nano-particles.

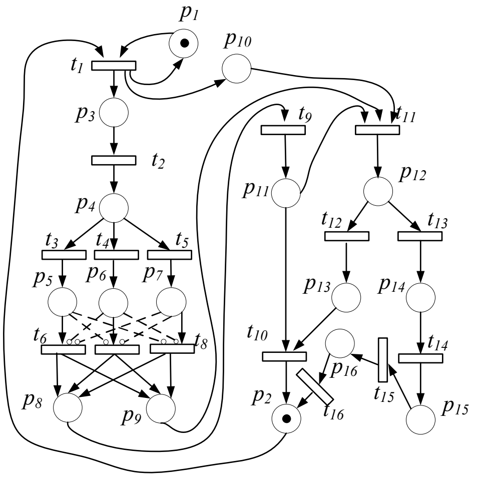


Fig. 3. Petri net model for the strategy of state estimation and fault detection.

Once it is detected that the visual display does not match the real-time changes in the nano environment, a local scan is needed to get the real manipulation result. To minimize the local scan time, the scan pattern should first scan the area with the highest probability that the object can be found. Since Kalman filter can give an optimal estimation of the position, the estimated position should have the highest probability that the object located. Thus this position should be scanned first. The scan pattern is designed as shown in Fig. 2. The solid circle O represents the particle position in the visual display interface. The dotted circle O_e represents the estimated position from the Kalman filter. In an ideal case, the particle's real position should be same with the estimated position from Kalman filter. Due to the environment noise or the dynamic modeling error, the real position of the particle may not be at point O_e. The scan pattern will first go along line L_0, which passes through the displayed center O and estimated center O_e. If the particle was not found, then the scanning line moves up and down along the direction perpendicular to Line L_0 alternatively by a distance of $3/2R$ until the particle is found. Once the particle is found, the scanning line forms two intersection points with the boundary of the particle, P_1 and P_2. Another line V, which goes through the midpoint between P_1 and P_2 and is perpendicular to the previous scanning line, is scanned to get the actual center O_r of the particle. The last scan line has two intersection points with the boundary of the particle, Q_1 and Q_2. The final actual center of the nano-particle O_r, is located at the midpoint between Q_1 and Q_2. The length of the scanning line L is determined by the maximum distance OO_e and the radius of the particle such that $L \geq OO_e + 2R$. The visual display is updated immediately after the actual position is obtained.

Table 1. The meaning of p_i and t_i

Place	Meaning	Trans	Meaning
p_1	Ready for manipulation	t_1	If object is manipulated
p_2	Real-time changed display	t_2	Length, width identify
p_3	Get object information	t_3	If $L/R > 15$
p_4	Length=L, Width=R	t_4	If 1.5< L/R <15
p_5	Wire	t_5	If $L/R < 1.5$
p_6	Rod	t_6	Wire model selected
p_7	Particle	t_7	Rod model selected
p_8	Model ready for visual display	t_8	Particle model selected
p_9	Model ready for Kalman filter	t_9	Model based calculation
p_{10}	Interaction force information	t_{10}	Update visual display
p_{11}	Behavior states in visual display	t_{11}	Residual calculation
p_{12}	Get the residual value	t_{12}	If Residual< δ
p_{13}	No fault in display	t_{13}	If Residual> δ
p_{14}	Alarm signal triggered	t_{14}	Ready for local scan
p_{15}	Ready for local scan	t_{15}	Local scan performed
p_{16}	True manipulation result	t_{16}	Update visual display

3. PETRI NET MODEL

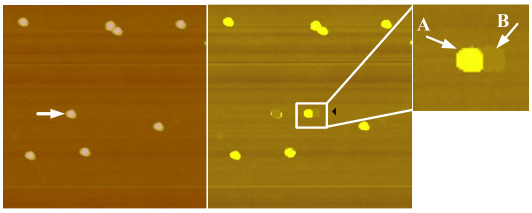
From above discussions we can see this strategy should perform the following actions during manipulation, such as task classifying, state estimation, fault detection, trajectory planning, hardware real-time execution and so on. Obviously, both discrete and continuously events are involved in this strategy. There is no a continuous method that can analyze and describe this hybrid system. As shown Fig. 3, a Petri net is used here to organize this strategy in an unified frame. This graph consists of two kinds of nodes: places and transitions, where arcs are either from a place to a transition or vice versa. Places represent conditions and system state, such as being "Ready for manipulation" or "Model Ready for visual display". Transitions represent events and actions, such as "Model based calculation" or "Update visual display". Tokens, which are represented by dots in places, correspond to signals owing in the system and their collection reflects the state of the system. All the arc weights in this petri net are 1's. The meaning of the places p_i and the transactions t_i (i=1,2,...16) is explained in Table 1. The initial state of the petri net has two tokens in p_1 and p_2 respectively, which means the system is ready for manipulation. The token will be fired under the transition rule of the petri net. After manipulation starts, t_1 is first fired when an object is manipulated by the AFM tip. The topography information of the object is acquired in p_3. At the same time, the real-time interaction force is obtained and recorded in p_{10}. Through firing t_2, the length 'L' and width 'R' is calculated and reserved at p_4. Based on the ratio between 'L' and 'R', the current manipulation task is classified into one of the three manipulations: wire, rod and particle (Li (2005)). To get the unique model of the object through Petri net, a new kind of arc called *inhibitor arc* is introduced to Fig. 3 (Peterson (1981)). An inhibitor arc connects a place to a transition and is represented by a dashed line terminating with a small circle instead of an arrowhead at the transition, like the arc from p_6 to t_6 in Fig. 3. The inhibitor arc disables the transition when the input place has a token and enables the transition when the input place has no token. No tokens are moved through an inhibitor arc when the transition fires. By this way, the transitions of t_6, t_7, t_8 each time only one of them can be fired, so the corresponding model for current manipulation task is unique and determined. Then the tokens are moved to p_8 and p_9. After t_9 is fired, the object's behavior states in visual display is calculated and reserved in p_{11}. Each input place of transaction t_{11} has a token now, which satisfied the firing qualification of t_{11} and the residual between the estimated states by the Kalman filter and states in the visual display is calculated. After the token is moved to place p_{12}, t_{12} is fired if the residual value is less than the threshold δ, otherwise t_{13} is fired to generate an alarm signal to trigger the local scan. Before local scan is performed, the token is removed first from p_{14} and added to p_{15} to switch the AFM work mode from manipulation mode to image mode. Then local scan is performed, and the true manipulation result is obtained. Finally the visual display is corrected by the true manipulation result. By the tokens' flowing in the Petri net, the request for time sequence and synchronization of the strategy are satisfied. The various manipulation task can be handled in a unified framework, which makes the strategy more flexible to the general problems. The Petri net also gives facility to add new function to the strategy for more complex application in future.

4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The experiments were performed in ambient condition. The experimental system mainly consists of a NanoScope IV Atomic Force Microscope (Vecco Inc., Santa Barbara, CA) and some peripheral devices including an optical microscope, a haptic device (SensAble Technologies Inc., Woburn, MA), two Multifunction Data Acquisition (DAQ) cards NI PCI-6036E and NI PCI-6733 (National Instruments), and three computers.

4.1 Implementation of Kalman Filter Based Fault Detection

As shown in Fig. 4a, one latex particle with a diameter of 350nm was pushed along the arrow direction. The displacement as a function of time is shown in Fig. 5a. The solid curve represents the displacement in the visual display, the dashed curve is the estimation displacement from the Kalman filter. The interaction force during manipulation is shown in Fig. 5b. Before $t = 4.9s$, the estimation displacement from the Kalman filter agreed with the displacement in the visual display. The Mahalanobis distance kept a near zero value. Due to the uncertainties and complexities of nano-environment, at $t = 4.9s$, the tip slipped aside the round particle. As shown in Fig. 5b, the interaction force



(a) (b)

Fig. 4. Manipulating latex particles (diameter 350nm) on polycarbonate surface with operation range of $11.5\mu m$. (a) Pushing a particle along the arrow direction; (b) Manipulation result in the visual display. The right picture is the zoom in image of the rectangle area in b. Area A represents the true position which is online obtained by local scan. The area B represents the faulty display position.

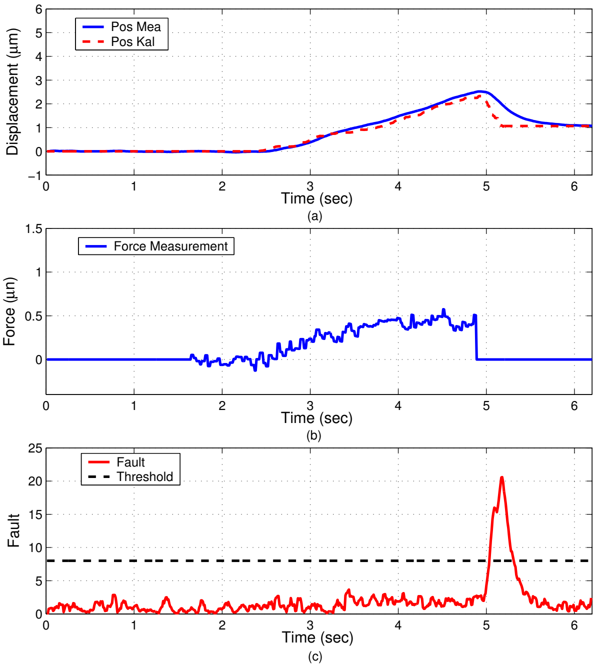


Fig. 5. Kalman filter based error detection. (a) Position displayed in the visual feedback and position estimated from Kalman filter; (b) The real-time interaction force; (c) The Mahalanobis distance.

suddenly dropped to zero. But in the visual display interface, the particle still kept moving with the AFM tip, the residual between the estimated behavior and the behavior in the visual display would have a significantly increase. Then as shown in Fig. 5c, the residual's Mahalanobis distance increased rapidly and exceeded the threshold. A inaccurate display was detected and an alarm signal was triggered.

4.2 Implementation of Local Scan Based Fault Correction

After the alarm signal was triggered, local scan was performed to on-line get the particle's real position. The scanning pattern for relocating nano-particle is shown in Fig. 6a. The topography information of the scanning lines is shown in Fig. 6b. From the topography information, the actual position of the nano-particle can be identified as described in Section 2.3. After the actual position is obtained, the visual display is updated with the real manipulation result. Fig. 4b shows the position error between the visual display and the real environment. The solid area A shows the real particle position from the local scan. The indent area B represents the particle location in the visual display. The position difference between A and B shows that the visual display loses match with the true environment, it also verifies that the position error can be detected by the Kalman filter efficiently and correctly.

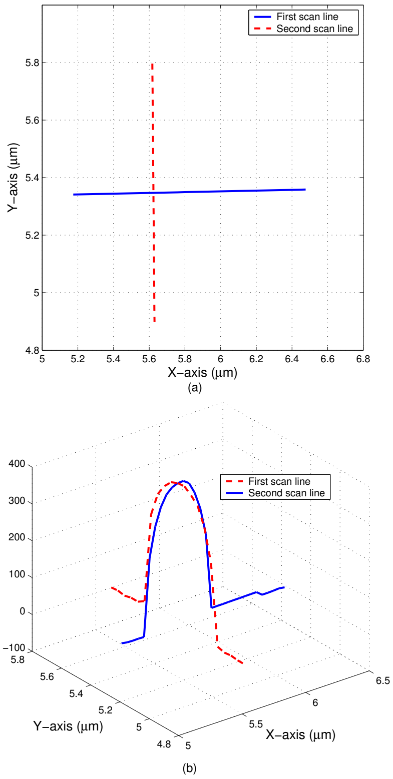


Fig. 6. Scan pattern and the scan result. a) The local scan pattern for nano-particle; b) The topography information along the scanning lines.

With the state estimation and fault detection, the reliability of AFM based nanomanipulation is greatly enhanced, the inaccurate display not only can be real-time detected, but also can be on-line corrected. Continuous manipulation can be done without any new AFM image scan in between. Fig. 7 shows the experimental result to form a triangle pattern by manipulating the nano-particles. Fig. 7a shows some latex particle deposited on polycarbonate surface. Fig. 7b shows the image of the real-time display interface. Fig. 7c shows the real manipulation results obtained from a new AFM image scan after finishing the manipulation task. Since the faulty display was online detected and corrected during manipulation, the final manipulation result in visual display interface perfectly matches the real AFM image. The whole manipulation process were finished continuously, which proved the significantly increased efficiency of AFM based nanoassembly.

5. CONCLUSION

Due to the uncertainties and complexities of nano-environment, the model based visual feedback can not describe the object's behavior accurately, which will lead

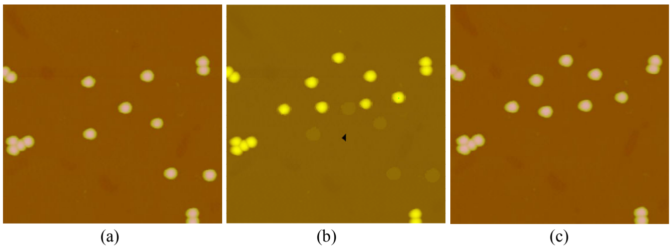


Fig. 7. Manipulating latex particles (diameter 350nm) on polycarbonate surface with operation range of $10\mu m$. (a) Image of latex particles on polycarbonate surface before manipulation (b) the real-time display on visual feedback interface after the final operation. (c) A new AFM scan image after manipulation

to a faulty display and a failed manipulation. In this paper, a strategy is proposed to overcome this problem by fault detection and correction. Since both discrete and continuous commands is involved during implementation, Petri net is adopted to model and organize this strategy. The experimental results shows that under the assistance of the proposed method, the inaccurate display can not only be real-time detected, but also be on-line corrected without interrupting manipulation. Hence, the efficiency and effectiveness of AFM based nanomanipulation has been significantly improved. This strategy also provides general instructions to develop new methods and new technologies to facilitate AFM based nanomanipulation in the future.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Chanmin Su of Veeco Instrument Inc. for his technical advice and help during the process of this research.

REFERENCES

G. Binning, C. F. Quate, and C. Gerber. Atomic force microscope. *Physical Review Letters*, 56:930-933, 1986.

D. M. Schaefer, R. Reifenberger, A. Patil, and R. P. Andres. Fabrication of two-dimensional arrays of nanometer-size clusters with the atomic force microscope. *Applied Physics Letters*, 66:1012-1014, 1995.

D. Wang, L. Tsau, et al. Nanofabrication of thin chromium film deposited on Si(100) surfaces by tip induced anodization in atomic force microscopy. *Applied Physics Letters*, 67:1295-1297, 1995.

Y. Kim and C. M. Lieber. Machining oxide thin films with an atomic force microscope: pattern and objective formation on the nanometer scale. *Science*, 257:375-377, 1992.

R. Luthi, E. Meyer, H. Haefke, L. Howald, W. Gutmannsbauer, and H.-J. Guntherodt. Sled-type motion on the nanometer scale: Determination of dissipation and cohesive energies of c60. *Science,* 266:1979-1981, 1994.

T. Junno, K. Deppert, L. Montelius and L. Samuelson. Controlled manipulation of nanoparticles with an atomic force microscope. *Applied Physics Letters*, 66:3627-3629, 1995

R. W. Stark, S. Thalhammer, J.Wienberg, and W. M. Heckl, The AFM as a tool for chromosomal dissection-the influence of physical parameter. *Applied Physics A*, 66:579-584, 1998.

M. Sitti and H. Hashimoto. Tele-nanorobotics using atomic force microscope. *In Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 1739-1746, 1998.

M. Guthold, M. R. Falvo, W. G. Matthews, S. washburn S. Paulson, and D. A. Erie. Controlled manipulation of molecular samples with the naomanipulator. *IEEE/ASME Transactions on Mechatronics*, 5:189-198, June 2000.

G. Li, N. Xi, M. Yu, and W. Fung. Development of augmented reality system for AFM-based nanomanipulation. *IEEE/ASME Transactions on Mechatronics*, 9:358-365, 2004.

J. Zhang, N. Xi, G. Li, H. Y. Chan, and U. C. Wejinya, Adaptable End Effector for Atomic Force Microscopy Based Nanomanipulation. *IEEE Transactions on Nanotechnology,* 5:628-642, 2006.

G. Li, N. Xi, H. Chen, C. Pomeroy, and M. Prokos, “Videolized” atomic force microscopy for interactive nanomanipulation and nanoassembly. *IEEE Transactions on Nanotechnology*, 4:605-615, 2005.

P. M. Frank, Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy : A survey and some new results. *Automatica* 26:459-474, 1990.

J. L. Peterson, Petri Net Theory and the Modeling of Systems. Englewood Cliffss, NJ: Pretice-Hall, Inc., 1981.