ULTRAHIGH-SPEED LARGE-RANGE ATOMIC FORCE MICROSCOPE IMAGING: ADAPTIVE MULTILOOP MODE VIA FIELD PROGRAMMABLE GATE ARRAY

BY JIARONG CHEN

A thesis submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Qingze Zou

and approved by

_____

_____

_____

New Brunswick, New Jersey

May, 2018

ABSTRACT OF THE THESIS

Ultrahigh-speed Large-range Atomic Force Microscope Imaging: Adaptive Multiloop Mode via Field Programmable Gate Array

by Jiarong Chen

Thesis Director: Qingze Zou

This thesis presents the development of ultra-high-speed large-range dynamic mode imaging of atomic force microscope (AFM), through the extension of the adaptive multi-loop mode imaging technique to the field-programmable gate array (FPGA) signal acquisition and processing hardware platform. High speed imaging is needed in atomic force microscope imaging to observe constantly changing processes such as chemical reaction on the sample surface, microorganism activity or macromolecules reactions. However, conventional imaging modes are too slow to observe such processes. Contact mode is faster than tapping mode but can damage the sample while tapping mode has less distortion but much slower. Such a speed limiation can be largely alleviated by the adaptive multi-loop mode (AMLM) technique which utilizes an online iterative feedforward controller to overcome the time delay of the z-feedback loop in tracking the topography. The goal of this thesis is to integrate the AMLM technique with the FPGA platform to further image the imaging speed by an order of mangitude without loss of quality and imaging range. Challenges in FPGA programming must be overcome to account for both the AFM system dynamics characteristics and the FPGA hardware specifications. For instance, the (inverse) Fourier transform in the

feedforward controller is replaced with time domain signals along with introducing the tapping amplitude error into the feedforward control and the tapping mode deflection decoupling technique. Moreover, a modeling-free inversion-based iterative feedforward control (MIIFC) approach is implemented for the x-axis piezo actuator control to track the high frequency triangular wave needed for high speed imaging. In the experiment a calibration reference sample is scanned to validate the control scheme and present the imaging results by comparing with the much slower conventional tapping image. The experimental results show that by using the AMLM imaging technique on FPGA board, the imaging speed increases from 5 Hz to 100 Hz while maintaining good imaging quality. However, although the main features of the sample topography are captured, many details are loss due to the time delay in the control system and high frequency noise.

Acknowledgements

I would like to express my deep and sincere gratitude to my advisor, Dr. Qingze Zou. His novel ideas have leaded me all through this work. He also provided me strong support through constant encouragement and patient guidance. Moreover, his attitude toward research has had a remarkable impact on my entire career. Working with him has made this study enjoyable and meaningful.

I would also like to thank my committee members for their efforts and contributions to this work: Prof. Zoran Gajic and Prof. Robert Hayes.

At the same time I am deeply grateful to my lab colleagues. Without their assistance this work would not have been successful.

Dedication

For my grandma, mother, and my family

Table of Contents

List of Figures

Chapter 1

Introduction

Atomic Force Microscope (AFM) is a type of scanning probe microscope with high resolution on the order of fractions of a nanometer. The AFM has two major abilities: force measurement [1] and imaging. In force measurement, it can measure the intermolecular force between the sample and the tip of the probe as a function of their distance. The function can be further applied to measure the mechanical properties of the sample, such as Young's modulus of the sample. [2] In the imaging, by maintaining the distance between the probe and the sample, the height of the probe is recorded and used to form a three-dimensional shape (topography) of the sample surface with high resolution. Therefore, AFM is widely used for different range of disciplines of the natural science, such as solid-state physics [3], semiconductor science and technology [4],molecular biology, molecular engineering, polymer chemistry and physics [5], surface chemistry [6], cell biology [7], and medicine. In this thesis, the imaging ability of AFM is the main topic.

AFM has two main modes of operation: contact mode and tapping mode. In contact mode, the tip of the probe is in constantly contact with the sample. This constant contact enables the probe to track the topography very well even with large variations in heights. It also has faster imaging speed over tapping mode. However, it may damage the surface of some soft samples during scanning due to the larger force applied to the sample compared with tapping mode. To avoid such problems, tapping mode is widely used. In tapping mode,the tip oscillates at its resonance frequency and is in contact with the surface only at the lower part of its oscillation. [8] Therefore, it has superior image quality and subdued sample distortion when compared to contact mode imaging. However, in order to maintain the image quality, the imaging speed is rather

slow compared to contact mode [1] because with the increase of the imaging speed the interaction between the probe and the sample will loss quickly and/or annihilates the cantilever tapping vibration, especially when the scan size is rather large. This has become the major bottleneck of tapping mode imaging.

It is challenging to increase the imaging speed of tapping mode. The main objective is to maintain the tapping amplitude around the set point as closely as possible. [9] It can be achieved using amplitude feedback control when imaging at slow speed (usually slower than 5Hz). When imaging speed increases, this requirement becomes challenging due to the inevitable time delay introduced into the amplitude feedback loop. [9]The reponse speed of the amplitude feedback loop to the height variation is limited by such delay. Since the tapping amplitude is sensitive to the variation of the distance between the probe and sample and region of force-distance in tapping mode is highly nonlinear, [10] even small variation of the probe-sample distance caused by the time delay of feedback loop will lead to the loss of the interaction between the probe and sample and/or annihilation the cantilever tapping vibration when scanning fast in tapping mode. Thus, the imaging speed is mainly limited by the control scheme.

There are several research groups working on high speed AFM imaging recently, such as Hansma, Ando, and Miles. [11] All of them can achieve video rate AFM imaging. However, they come with drawbacks. All of their methods require a specifically modified AFM, leading to an extra cost for the hardwares. In addition, only a relatively small area can be imaged by Hansma or Ando's method. Hansma's group has developed a AFM scanner with intention of observing relatively wide areas; a scan range of 6-15 μm in x- and y-directions and 6 μm in z-direction. On the other hand, Ando's AFM has a limited scan range of 240 nm.

In order to achieve higher imaging speed in tapping mode while maintaining its advantage over contact mode, an imaging mode called the adaptive multi-loop mode (AMLM) is proposed in [12]. Although compared with the video rate AFM mentioned above, it is much slower, but still faster than traditional tapping mode scan. Meanwhile, it can be implemented on any commercial AFM without special modification and has a larger scan range of 30 μm. The experiments of such imaging mode were conducted on

a commercial AFM (Dimension Icon, Bruker AXS, Inc.) under the Matlab xPC-target environment. The imaging speed is improved to 25Hz. To push the imaging speed even faster, another commercial AFM (Dimension Fastscan, Bruker AXS, Inc.) with much wider bandwith and the USB-7856R FPGA(Field Programmable Gate Array) Board from National Instrument, Inc. is selected to implement the control algorithm of AMLM imaging because of its high speed of real-time operation which is essential for control.

In this thesis, the design of a real-time control system module which implements the adaptive multi-loop mode imaging on the chosen FPGA platform is reported. The thesis is organized as follows. In chapter 2, the control scheme of traditional tapping mode is reviewed and the adaptive multi-loop mode is introduced. In chapter 3, the selected hardware platform and the FPGA program on Labview, National Instrument, Inc, are eleborated. Some necessary modifications on the AMLM algorithm are also introduced. In chapter 4, the images using traditional tapping mode and the images using AMLM mode are compared and in chapter 5 the thesis is concluded.

Chapter 2

Introduction of Adaptive Multiloop Mode Imaging

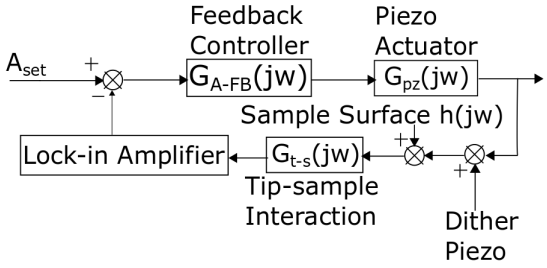

Figure 2.1: Schematic block diagram of z-feedback control in conventional TM imaging

2.1 Issues of Conventional Tapping Mode

Before introducing the Adaptive Multiloop Mode imaging, the drawback of the conventional tapping mode need addressed to understand why the AMLM imaging is developed. Tapping mode imaging is developed to overcome the disadvantage of contact mode imaging such as the image distortion [13,14] and the damage to the sample caused by probe sliding [1,10], particularly for soft sample. During tapping mode imaging the cantilever probe is excited by the dither piezo (see Fig 2.1) to vibrate near its resonance frequency and tap on the surface constantly. Then the tapping amplitude is measured by a lock-in amplifier and maintained around the set point A_{set} by a feedback control system using a piezoelectric actuator (called the z-piezo below). Under the condition that the sample topography is tracked closely, in other words, the tapping amplitude being close enough to the set point, the topography can be quantified as the z-piezo displacement.

However, in order to track topography closely, the speed of conventional tapping mode

is inherently hampered by its z-axis feedback control system. It will take several periods of tapping for the lock-in amplifier to measure the tapping amplitude. Therefore, this time delay is inevitable and leads to the difference between the measured amplitude and the instantaneous tapping amplitude $A_{set}(t)$. This difference would become larger when tapping amplitude varies rapidly with the sample topography variation. Although the time lag is relatively small and can be compensated for by the feedback control when imaging slowly, it becomes more evident and cause larger variation in the distance between the probe and sample as the imaging speed increases.

Nonetheless, because the probe-sample interaction force is highly nonlinear [9] regarding to the probe-sample distance, the cantilever tapping is sensitive to the probe-sample distance. Therefore, even a relatively small change in the probe-sample distance can cause the annihilation of the tapping and/or and a loss of probe-sample contact. A loss of contact tends to happen when the topography suddenly drops and the tapping approaches free vibration gradually due to the relatively long settling time of the control system. On the other hand, tapping can be annihilated completely around the regions where the topography rise fast, leading to the probe sliding on the surface. In conventional tapping mode, the feedback gain is limited by the time delay, making it challenging to avoid the issues mentioned above. Because a high feedback gain tends to cause overshoot [15] that results in the probe bouncing back and forth between the loss of contact and the annihilation of tapping. A feedback gain too small is incapable to account for the control as the scan speed increases. Therefore, the conventional tapping mode has a low speed limit.

2.2 Adaptive Multiloop Imaging

The adaptive multiloop mode imaging (AMLM imaging) [12] is proposed to address the above issues. As depicted in Fig 2.2, it introduced two parts on top of the z-feedback loop to control the z-axis motion: (i) a feedback control in the inner-outer loop structure to manage the tapping mode(TM) deflection, and (ii) an online iterative feedforward controller to overcome the time delay of the z-feedback loop in tracking the topography.

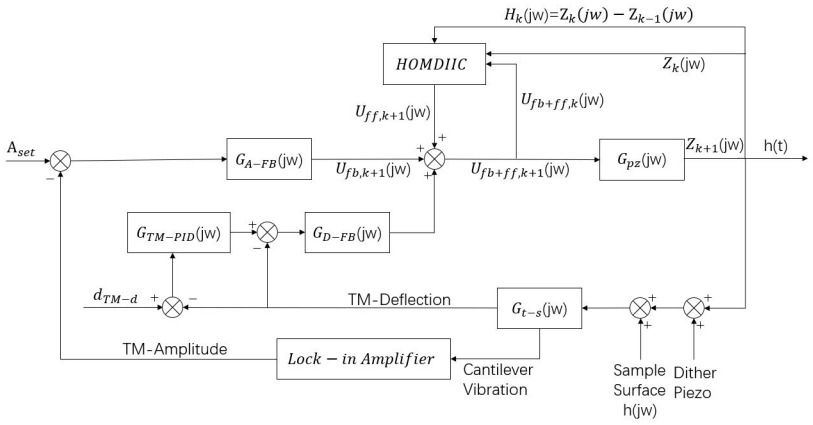

Figure 2.2: Schematic block diagram of AMLM imaging

The TM deflection inner-outer loop manage the average vertical position of the cantilever during tapping around the desired value to maintain stable tapping. Specifically, the outer loop updates the TM deflection set point $d_{TM\text{-}set}$ once a period of tapping while the inner loop tracks the regulated set point using a PID controller. The outer loop also employs a similar PID-style control as following:

$$d_{TM-set}(j+1) = k_I d_{TM-set}(j+1) + k_P e_{TM}(j) + k_D(d_{TM}(j) - d_{TM}(j-1)) \quad (2.1)$$

where

$$e_{TM}(j) = d_{TM-d} - d_{TM}(j) \quad (2.2)$$

and j=2,3,4,...,N-1,where N is the total number of sampling periods per image, and k_P,k_I and k_D are the proportional, integral, and derivative coefficients, respectively. The desired TM deflection $d_{TM\text{-}d}$ and tapping amplitude set point A_{set} are picked according to the ratio of the chosen tapping-amplitude set point to the free amplitude, A_{def}/A_{free}. To do so the $d_{TM\text{-}d}$ vs A_{def}/A_{free} relation is measured beforehand. After picking the A_{set} $d_{TM\text{-}d}$ is picked accordingly.

To enhance the tracking of the sample topography and boost the scan speed, the following high-order modeling-free difference-inversion-based iterative control (HOMDIIC) [16] algorithm is implemented online as a feedforward controller of the piezo actuator

integrated to the z-axis feedback loop,(see Fig.3)

$$U_{ff,0}(j\omega) = 0 \tag{2.3}$$

$$U_{ff,1}(j\omega) = \frac{U_{ff+fb,0}(j\omega)}{Z_0(j\omega)} H_{ffd,1}(j\omega) \tag{2.4}$$

$$U_{ff,k+1}(j\omega) = U_{ff,k} + \lambda \frac{U_{ff+fb,k}(j\omega) - U_{ff+fb,k-1}(j\omega)}{Z_k(j\omega) - Z_{k-1}(j\omega)} e_k(j\omega) \tag{2.5}$$

$$e_k(j\omega) = H_{ffd,k+1}(j\omega) - Z_k(j\omega) \tag{2.6}$$

where jω donates the Fourier transform of the correspoding signal. The $U_{ff+fb}(j\omega)$ donates the total input (feedback+feedforward) applied to the z-piezo.(i.e. $U_{ff+fb,k}(j\omega)$= $U_{ff,k}(j\omega)+U_{fb,k}(j\omega)$) The $Z_k(j\omega)$ and $H_{ffd,k}(j\omega)$ donate the z-piezo displacement measured and the desired trajectory the z-piezo actuator needs to track respectively on the kth line. It is worth mentioning that the ratio in the above control law is the inverse of the frequency response of the z-piezo actuator, and are updated iteratively every line. Therefore, λ is a constant chosen to ensure that the control law will converge. The feedforward input for the next scan line is computed during the sampling period between the last sampling point of the current line and the first sampling point of the next line using the HOMDIIC algorithm(Eq 2.3-2.6) .The computed feedfoward input is then applied one point at a (sampling) time during the next-line imaging. During the scan, the sample topography is quantitified by the displacement of z-piezo,i.e.$h_k(t) = z_k(t) - z_{k-1}(t)$ and the desired trajectory of the (k+1)th line $h_{k+1}(t)$ is approximated as that of the kth line, i.e. $h_{k+1}(t) \approx h_k(t)$, assuming that kth line and the (k+1)th line are so close that the small topography variation between them can be neglected.

Chapter 3

FPGA Implementation of Adaptive Multiloop Mode Imaging

The FPGA program is developed on the USB-7856R Multifunction Reconfigurable I/O Device (National Instrument, Inc.) with Kintex-7 160T FPGA core. Therefore, the FPGA program is implemented using the Labview 2016 (National Instrument, Inc.). Moreover, due to the inadequate functional block in Labview FPGA enviroment, some algorithms are developed on Matlab and the mat files are used to exchange data across this two platforms. The structure of the program is depicted as Fig 3.1. Each block in Fig 3.1 represents one or several functional block in Labview FPGA. In order to pursue a high sampling rate for high speed scan, some modifications(see Fig.4) are made to make the program more accurate and efficient.

The major change is that the Fast Fourier Transfrom(FFT) block consumes many resources and hampers the sampling rate. Furthermore it introduces a initial latency the length of signal. In this case it means the FFT block will return its output of the very first scan line after two scan line finishes. The inverse FFT will also introduce the latency. This latency is unacceptable as the algorithm is supposed to overcome the time delay. Thus, the inverse response of the z-piezo actuator is measured offline. Thanks to the high quality of the actuator, the frequency response can be approximated by a constant gain and the FFT/IFFT operations are removed from the program. The outer loop of TM deflection is also removed to simplify the program.

Note that the feedforward controller starts working on the second scan line (see (2.3)) and the desired trajectory is approximated by the first line, which requires accurate tracking on the first line using feedback loop only. As explained above feedback loop is not capable for high speed imaging due to the time delay. Hence, the scan rate is set

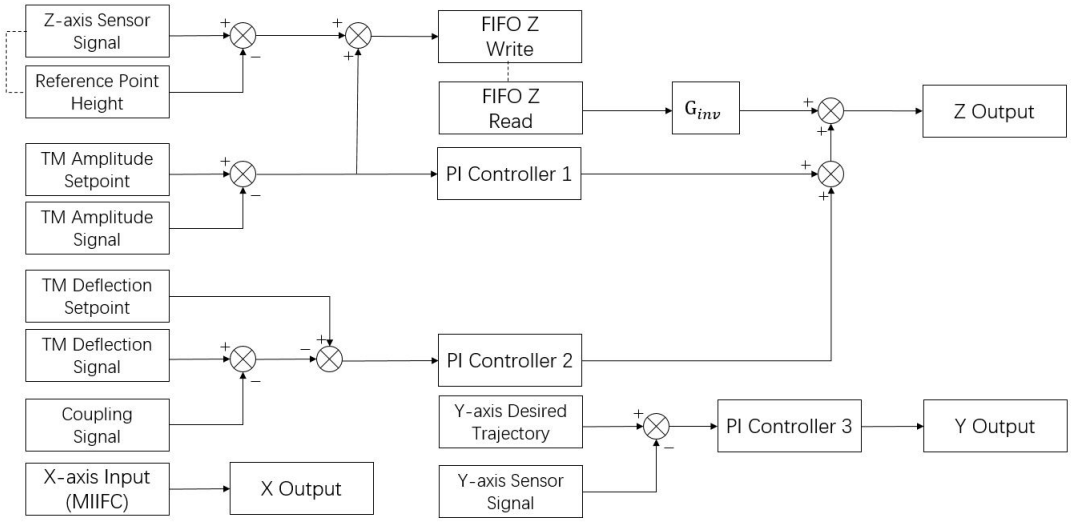

Figure 3.1: Schematic block diagram of AMLM imaging on FPGA

at slow speed for the first five scan lines and increases to high speed afterwards. The sampling rate is adjusted accordingly online.

As depicted in Fig 3.1, the program contains three main parts:(i) X and Y axis control, (ii)feedback controls to regulate the tapping mode amplitude and deflection and (iii) an online feedforward controller to overcome the time delay of the z-feedback loop.

3.1 X and Y Axis Control Scheme

The probe goes back and forth on x-axis to scan the surface so a high frequency triangular wave needs tracking by the piezoelectric actuator on x-axis(called the x-piezo below). Due to the relatively narrow bandwith of the x-piezo, the traditional PID closed-loop control is inadequate for high speed imaging. In the experiment such closed loop control with PID is unstable and causes oscillations with high feedback gain in X

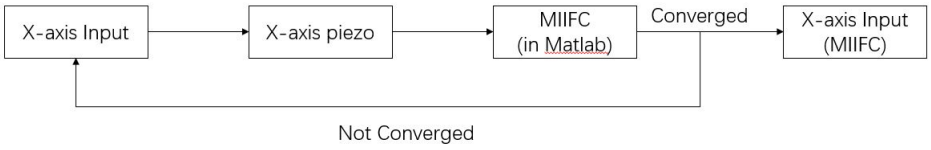

Figure 3.2: Schematic block diagram of MIIFC Algorithm

direction. In contrast, small feedback gain cannot track the triangular wave accurately. Therefore, a modeling-free inversion-based iterative feedforward control (MIIFC) [17] approach is implemented for the x-piezo control. The MIIFC algorithm is given as:

$$U_0(j\omega) = \alpha X_d(j\omega) \tag{3.1}$$

$$U_k(j\omega) = \frac{U_{k-1}(j\omega)}{X_{k-1}(j\omega)} X_d(j\omega) \tag{3.2}$$

where $\alpha \neq 0$ is a prechosen constant and k donates the number of the iterations. U($j\omega$), X($j\omega$), $X_d(j\omega)$ donate the Fourier transform of the input, output and desired output respestively. As The MIIFC algorithm involves FFT and inverse FFT too, it is too complicated to implement on FPGA board online without sacrificing sampling rate and thus, a standalone FPGA program is implemented to apply the input signal and retrieve the output signal for the MIIFC algorithm and the algorithm is implemented on Matlab offline(see Fig 3.2) . This process is conducted while tapping in the air and the final converged input is then stored and applied to the AFM for imaging. By utilizing MIIFC the tracking error is below 2%.

For Y-axis, the probe just goes along a straight line at a constant speed. The movement is much slower that PID control is sufficient. So the desired Y-axis trajectory is stored and applied to the PI controller during the scan.

3.2 Z-piezo Feedback Control

For the z-piezo, two PI feedback control are applied to regulate tapping amplitude and tapping deflection respestively. The set point is set at 30% of A_{free}. The proportional gain and integral gain for the amplitude feedback are tuned so that the settling time of step response is optimized with the overshoot maintaining under 5%.

Regarding to the deflection loop, the TM deflection signal retrieved from the AFM is coupling with the X and Y axis movements during scanning. In order to retrieved the decoupled TM deflection, the TM deflection signal is retrieved and stored while scanning in the air without touching the surface of sample beforehand. During the actual scan this coupling signal TM deflection signal is substracted from the raw deflection signal

to decouple the signal from X and Y-axis movements.

After the probe goes down and touches the surface, the TM deflection will have an offset. Then this offset is added to the pre-stored deflection signal so that during the scan the offset is removed. Therefore, the set point of the PI feedback of deflection is set at zero. The gains are tuned so that it will not conflict with the amplitude feedback loop.

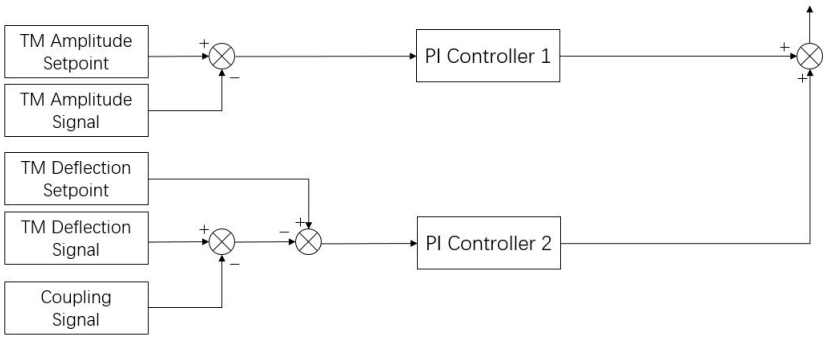

Figure 3.3: Schematic block diagram of Z-piezo Feedback Control

3.3 Z-piezo Feedforward Control

The essence of feedforward control is to predict the sample topography of next scan line and overcome the time delay. The desired trajectory of the $(k+1)$th line $\mathrm{h_{k+1}(t)}$ is approximated as that of the kth line. Note that the time delay causes spikes in the cantilever reponse to the tapping amplitude after the probe already passes those locations. [18, 19] To overcome such issues, the error of the tapping amplitude is also introuduced into the feedforward input to help the z-piezo to drive the cantilever to respond in advance to the topography variation, thereby reducing the tapping amplitude spikes. Therefore, the feedforward input is given as

$$U_{ff,0}(t) = 0 \tag{3.3}$$

$$U_{ff,k+1}(t) = G_{inv}(h_k(t) + \alpha e_k(t)) \tag{3.4}$$

$$e_k(t) = A(t) - A_{set} \tag{3.5}$$

where k=0,1,2,...,N-1 and N is the number of scan lines. G_{inv} is the inverse of the approximated z-piezo gain and α is the correction factor hand tuned based on the estimation of the sample surface height. $h_k(t)$ is the z-piezo displacement of the kth line. In the FPGA program(see Fig.), in order to avoid noise being fed into the feedforward channel and then amplified, the signal from z-sensor which collect the height information of the probe and the amplitude error signal are passed through two low-pass filter with different cut-off frequency seperatively. In order to get the displacement $h_k(t)$, the height of the first sample point, named as the reference point, is stored and substracted from the z-sensor signal. Then they are added together and stored into a FIFO (First In First Out, the implementation of queue in Labview FPGA). Then it is multiplied by G_{inv} and applied to the next scan line and so on and so forth. To compensate the phase lag introduced by the low pass filters, the feedforward input is released earlier (several dozens sample points before the next scan line) to align with the topography of the next line.

At the beginning of the imaging, the scheme described above is applied to scan the first line repetitively until the difference of the z-piezo displacement between two consecutive iterations is close to noise level. The feedforward input for upcoming scan lines only need to update once without scanning repetitively because the convergence of the feedforward input is faster than the topography variation between scan lines.

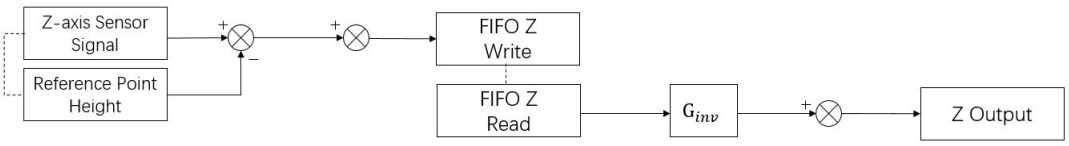

Figure 3.4: Schematic block diagram of Z-piezo Feedforward Control

3.4 Host VI Configuation

VI, as in Virtual Instrument, is the program format used by Labview. In a Labview FPGA project the host VI runs on the PC to control the FPGA and exchange data

between them. And there is the client VI which contains the FPGA program and runs on FPGA board.

In the host VI, the client VI that will be executed is designated first. Then the static program parameters are transferred into the FPGA. After that the X and Y-axis trajectories(see chapter 3.1) are transferred and stored into the memory blocks on the FPGA via DMA(Direct Memory Access) FIFO to avoid possible data loss caused by the desynchronization between FPGA and PC. An interruption signal will be sent by PC after all data required is transferred to tell the FPGA program VI to execute. During the imaging process, the z-piezo signal and the tapping deflection signal are transferred back to PC via two DMA FIFOs and stored into mat files to import to Matlab for topography analysis.

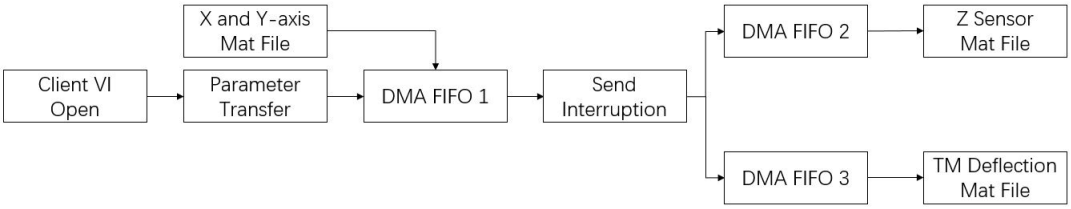

Figure 3.5: Schematic block diagram of Host VI

Chapter 4

Experimental Implementation and Discussion

The calibration reference sample provided by Bruker AXS Inc. was imaged at 20 μm range to demonstrate the technique on FPGA, by comparing to TM imaging at a much lower speed as reference. The experiments were conducted on a commercial AFM(Dimension Fastscan, Bruker AXS,Inc.) on which all the drive of the piezo actuators of X,Y and Z-axis and all of the sensor signals including the TM amplitude and deflection signals can be accessed directly. All of the signals were acquired by the DAQ(Data Acquisition) built in the FPGA board(USB-7856R) directly.

Throughout the imaging experiments, the AMLM imaging was implemented to image the calibration sample at a scan rate of 100 Hz, and then compared to the results obtained using TM imaging at 5 Hz. All algorithm, such as the feedforward channel and the feedback channels can be turned on and off to demonstrate their effect respectively.

4.1 X and Y-axis Trajectory

After the MIIFC algorithm converges the X-axis actuator output and the desired ouput is shown in upper part of fig 4.1 and the error is shown in the lower part. As mentioned in Chapter 3, the imaging speed was set at 5 Hz so that the topography tracking for the first scan line is reliable with feedback control only as the initial input of the z-piezo feedforward controller.

As shown in the fig 4.1, the tracking is very satisfying for the slower portion as the error is within the range of ± 0.02 while the imaging speed up later and the error goes larger relatively, but is still within the range of ± 0.05. Overall, the 2-norm error is 1.37%. The PI control result is not shown as it caused oscillations and would damage the AFM

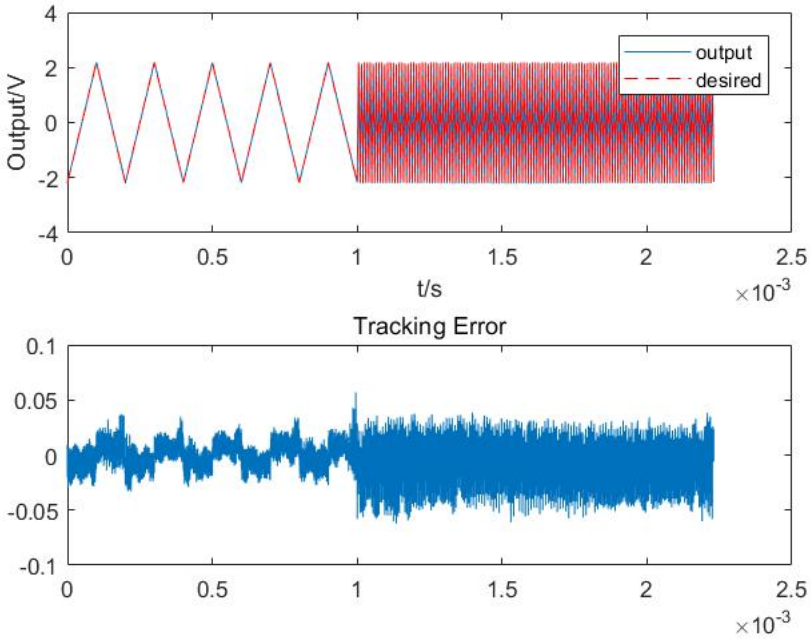

Figure 4.1: X-axis output VS desired output

under such high frequency. The zone-in figure of the X-axis output is shown in Fig 4.2. For comparision, the slower portion(5 Hz) is on the left and the faster portion(100 Hz)

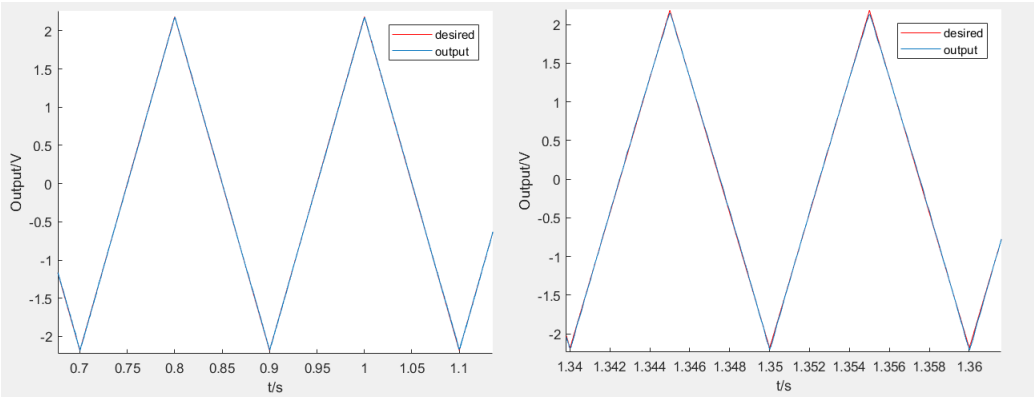

Figure 4.2: X-axis output VS desired output(zone-in)

is on the right of Fig 4.2. The major difference is the peak of triangular wave. As the scan speed increases the MIIFC algorithm cannot compensate for the smaller gain of the X-axis actuator as good as when the imaging speed is slow.

Note that the error shown in Fig 4.1 and Fig 4.3 increases rather significantly when the imaging speed changes. As discussed in Chapter 3.1 the MIIFC algorithm relys on

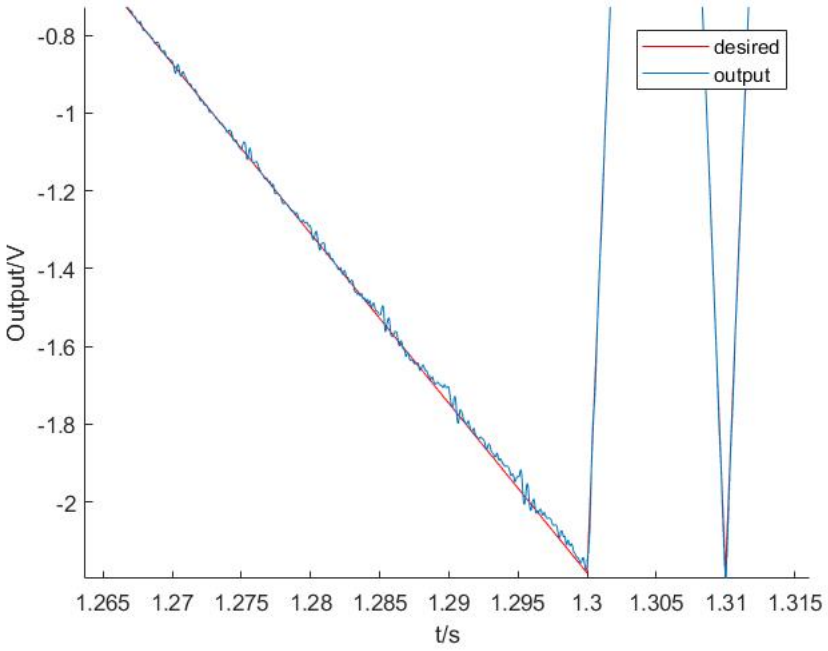

Figure 4.3: X-axis output VS desired output(Speed Shift)

(inverse) Fourier transform. In order to get the discrete Fourier transform, the sample frequency of the signal should be fixed. However, the sample frequency was changed along with the imaging speed in the experiments. To tackle this problem the actual output of the first five scan lines was expanded using linear interpolation to match the sampling frequency before doing the Fourier transform and the according input computed by MIIFC algorithm was also downsample in the same way to get the actual input signal. This method works well for the slower and faster portion individually but will increase the error when speed shifts.

As for Y-axis actuator, it used a simple PI feedback control. Although the optimal proportional and intergral gain should be different under different sampling rate, utilizing two sets of gain would cause transition response when sampling rate (scan rate) shifts. Therefore, the gains were tuned under fast sampling rate to ensure the stability of high speed imaging. As it turned out, it can control the slow imaging, too. The overall 2-norm error is 1.03%. Note that the probe scan the first line repetitively until the HOMDIIC algorithm converged, i.e. the difference of the z-piezo displacement between two consecutive iterations is close to noise level.

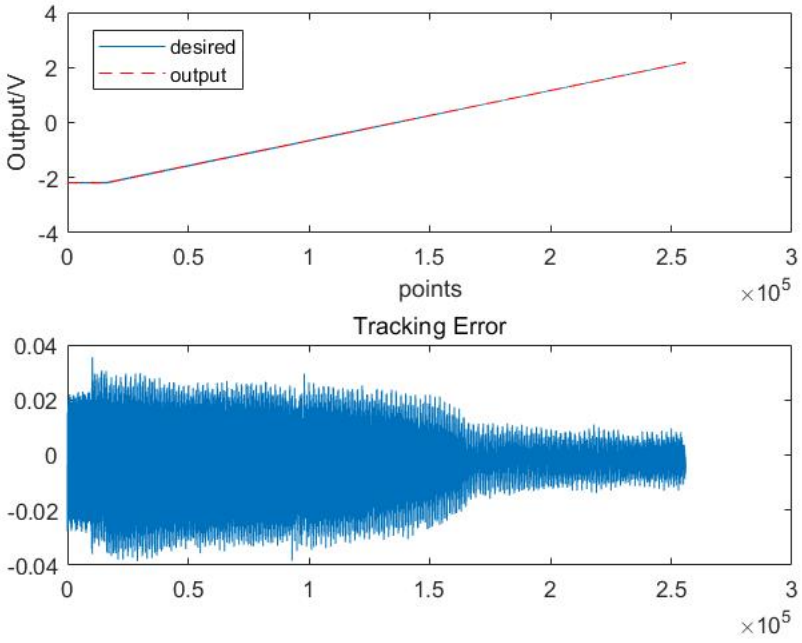

Figure 4.4: Y-axis output VS desired output

4.2 Imaging Quality Comparision

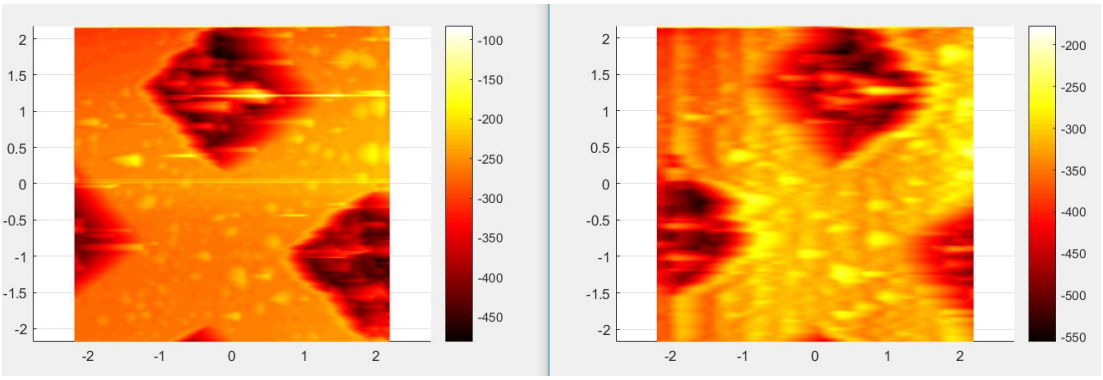

Figure 4.5: Slow TM (5 Hz) VS Fast TM (100 Hz)

First of all, the images using conventional tapping mode to scan are shown in Fig 4.4 to demonstrate the influence of the time delay discussed above. In the experiment the X-axis for slow TM imaging is controlled using PI feedback while the high speed TM imaging used the MIIFC open-loop control scheme. Although the imaging started at the same spot, there was a offset that was not compensated well by MIIFC and thus the probe went right a bit in the high speed image. From the slow (5 Hz) tapping

mode image several bright lines (which means it is relatively higher) can be seen. They are the tiny raised dots (most probably dust on the sample surface) that were not well tracked due to the time delay. The z-piezo detected the high portion but did not go down afterwards and left a bright line instead of a dot. After the imaging speed increases this effect is much more significant. Severe distortions on X-axis are shown over whole image, especially on the edge of the pitches and the dots on the surface.

The right image in Fig 4.5 is acquired using only TM amplitude feedback and z-

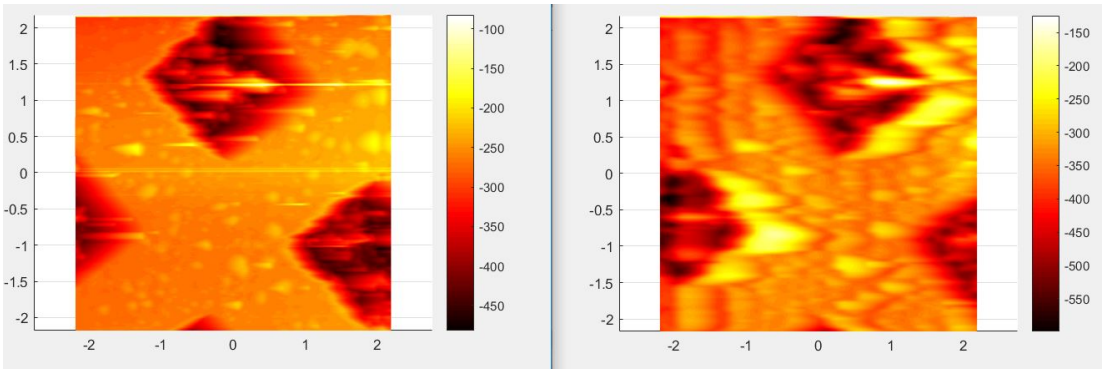

Figure 4.6: Slow TM (5 Hz) VS Feedforward w/o Error Term (100 Hz)

piezo feedforward without introducing the amplitude error term.(i.e. α=0 in (3.4)) As the Fourier transform in HOMDIIC algorithm is replaced with time domain signal to increase imaging speed, the proposed approximation is not accurate and cannot help reducing the time delay. On the contrary, the time delay effect gets even worse on the edge portion of the pitches.

Fig 4.6 shows the necessarity of introducing the error term to helpthe z-piezo to drive

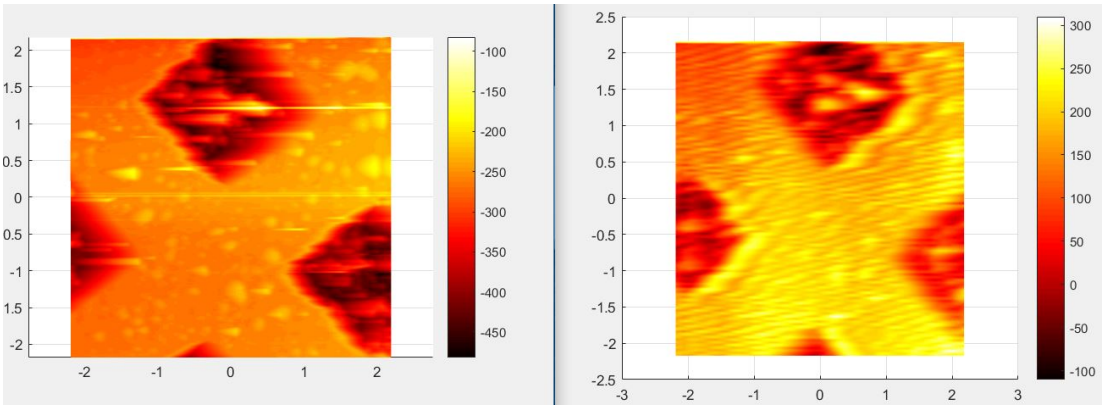

Figure 4.7: Slow TM (5 Hz) VS Feedforward with Error Term (100 Hz)

the cantilever to respond in advance. The edges of pithces become sharper than the those in the conventional TM image indicating that the time delay is reduced. However, large number of small oscillations come along with it as the amplitude error also introduces much noise into the feedforward channel. Although the error signal goes pass the low pass filter, the cut-off frequency is difficult to tune because the noise and the topography information overlap each other in frequency domain. If the cut-off frequency is too low, much topography information would also lost and thus, hamper the effect of amplitude error feedforward. As depicted in Fig 4.6, only a small portion of the dots can still be seen and many details (relatively smaller dots) are lost due to the low pass filter while some other dots are covered by the oscillations caused by noises.

Finally the TM deflection feedback loop is introduced upon the algorithm discussed

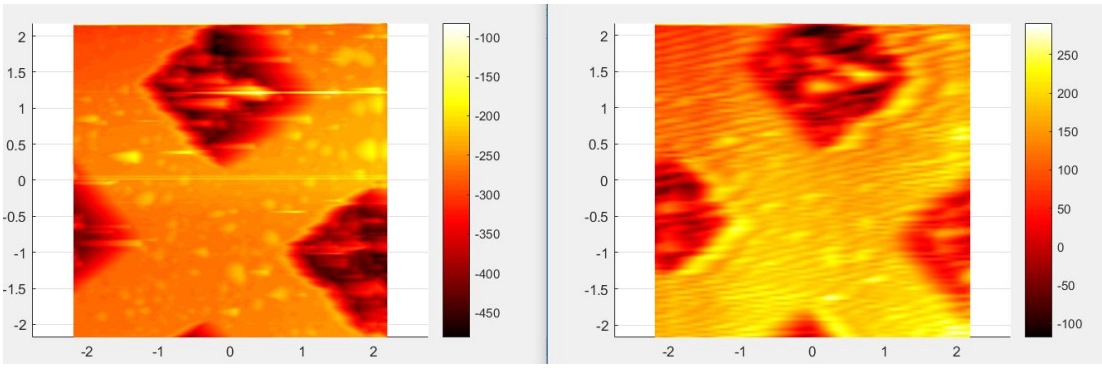

Figure 4.8: Slow TM (5 Hz) VS Modified AMLM (100 Hz)

above to form the modified AMLM imaging. While its effect is not as obvious as feedforward, it help reduce the oscillations introduced by the feedforward amplitude error term and further reduce the time delay in the conventional tapping mode. Moreover, the tracking for the small topography variations is more accurate as more details shown in Fig 4.7 compared with Fig 4.6.

4.3 Advantage and Drawback of FPGA

The FPGA control system is very fast with high sampling rate and it is easy to program in Labview without any hand-written HDL(Hardware Discription Language) code. The FPGA board is very small comparing to a computer which means it is highly portable

and the program can be modified to suit the need of the project. However, it also reveals its drawback in the project.

In the experiments, resolution (i.e. the number of points on each scan line) is crucial for the image quality. The PI controller implemented on FPGA is sensitive to the distance between each scan point due to the precision loss between fix-point operations and floating-point operations. If the scan points are too scarce the PI controller would act slowly and induce more delay.

Secondly, FPGA has its own hardware limitation. Due to the cost some hardware is very limited on the FPGA board, such as the block memory and DMA FIFO. The block memory is so small that it limits the number of scan lines because the X-axis trajectory MIIFC input and the deflection coupling signal need to store on board. The AMLM implementation on FPGA has only 120 scan lines while the conventional TM implementation has 256 scan lines. There are only 3 DMA FIFO on the USB-7856R FPGA board, which means only 3 signal can be transferred between FPGA and PC at the same time. And the DMA FIFOs are unidirectional. Direction of data transfer is designated before compilation of the program.

Last but not least, the software limitation. Labview is easier to program than HDL. But it only provides certain function blocks. Although all the function blocks are useful, they are still limited. Labview also provides self-program functional block but it is difficult to combine HDL and Labview together without inducing more bugs in the program.

Chapter 5

Conclusion

In this thesis dissertation, the development of adaptive multiloop high speed imaging on Labview FPGA board for Atomic Force Microscope is presented. AFM is widely used for different range of disciplines of the natural science, such as solid-state physics, semiconductor science and technology, molecular engineering, polymer chemistry and physics, surface chemistry, molecular biology, cell biology, and medicine. High speed imaging not only means efficiency but also enables observations for microorganism or macromolecules without disturbing their reactions. However, both of the conventional mode of operation, contact mode and tapping mode, are not satisfying for high speed imaging. Contact mode has faster imaging speed over tapping mode and enables the probe to track the topography very well even with large variations in heights. Nonetheless, the probe is constantly contacted on the sample. Hence, the imaging speed is limited by the interaction force between the probe and sample. Moreover, it may damage the surface of some soft samples during scanning due to the larger force applied to the sample compared with tapping mode. Contrarily, tapping mode has superior image quality and subdued sample distortion when compared to contact mode imaging. However, in order to maintain the image quality, the imaging speed is rather slow compared to contact mode because with the increase of the imaging speed the interaction between the probe and the sample will loss quickly and/or annihilates the cantilever tapping vibration, especially when the scan size is rather large.

To achieve higher imaging speed, the adaptive multiloop imaging mode is proposed and implemented on FPGA board. The AMLM control scheme introduces a feedback control to manage the tapping mode(TM) deflection, and an online iterative feedforward controller on top of the conventional tapping amplitude feedback control to overcome

the time delay of the z-feedback loop in tracking the topography. In order to implement it on the FPGA more efficiently and therefore, maintain the high sampling rate, the (inverse) Fourier transform in the feedforward controller is replaced with time domain signals along with introducing the tapping amplitude error into the feedforward control. The imaging speed is increased to 100 Hz, which is 20 times of the conventional tapping imaging, while maintaining a satisfying image quality compared with the slow tapping image. The experimental results presented show the effect of each channel separately. In the future, there is much work to be done. The image quality is still far from slow tapping image that it can only maintain the significant feature in the sample topography and many details are lost in the imaging process. Moreover, the AMLM imaging mode on FPGA is not very consistent that the image quality can be worse on other samples. In order to improve the image quality the low pass filter for the feedforward controller can be replaced with a Kalman filter according to the probe-sample interaction to predicted the next-line topography more accurately.

Appendix A

Matlab Code

```
%File Name: x_y_control.m
%Function: Generate X and Y desired Trajectory
scan_rate=1; %scan rate hz
x_scan_size=20; % x scan size um
x_y_ratio=1; %aspect ratio
line_n=round(120/x_y_ratio); %scan line number
x_y_sen=4.58; %sensitivity of x_y sensor: 0.156 um/v

y_scan_size=x_scan_size/x_y_ratio;
T=line_n/scan_rate;
repeat_line=8;%first line repeat 5 times
Sample_fre=2000; %sampling frequency
% T=5;
t=0:1/Sample_fre:T-1/Sample_fre;
t1=0:1/Sample_fre:(T+T*repeat_line/line_n)-1/Sample_fre;
y_ramp=y_scan_size./x_y_sen/line_n;
% Z_control=1+0.5*sawtooth(scan_rate*2*pi*t1,0.5);
X_control=x_scan_size./2/x_y_sen*sawtooth(scan_rate*2*pi
*t,0.5); %t1
Y_control=linspace(0,y_scan_size./x_y_sen,T*Sample_fre);
Y_control=Y_control-mean(Y_control);
```

```
L=length(t);

x1=linspace(0,0,0.5*Sample_fre);
x2=linspace(0,X_control(1),0.5*Sample_fre);
x3=linspace(X_control(1),X_control(1),0.5*Sample_fre);
x4=linspace(X_control(L),X_control(L),0.5*Sample_fre);
x5=linspace(X_control(L),0,0.5*Sample_fre);

y1=linspace(0,0,0.5*Sample_fre);
y2=linspace(0,Y_control(1),0.5*Sample_fre);
y3=linspace(Y_control(1),Y_control(1),Sample_fre*(0.5+T*
repeat_line/line_n));
y4=linspace(Y_control(L),Y_control(L),0.5*Sample_fre);
y5=linspace(Y_control(L),0,0.5*Sample_fre);

X_control=[x1 x2 x3 X_control x4 x5 x1]';

Y_control=[y1 y2 y3 Y_control y4 y5 y1]';

%File Name:Main_x_miic.m
%Function:Setting parameters for MIIFC
%clc;format short ;close all; %clear all;
first_input_scale= 1; % Scaling factor for first input data
from desired trajectory
trj_type=2  ;       % Desired trajectory type 1 for triangle,
 2 for random
sim_on=0    %1 for simulation 0 for experimental measurement
num_iter=15  ;            % Number of Iteration
a=2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Hz_desired=100;
Hz_rate=Hz_desired;
Actual_smpl_rate=200;%kHz
Amp=2;
cut_off_frq=Hz_desired*5;
%eff_frq_rto=(cut_off_frq*1)/Actual_smpl_rate/1000*2;
eff_frq_rto=0.1;

axis='X';
tt1=linspace(0,13000,13000);
tt2=linspace(0,13000,13000*20);
temp=interp1(tt1,X_control(1:13000),tt2);
desired=[temp';X_control(13001:262000)];
dt=1/Actual_smpl_rate/1000;
iii=0;

%File Name:Input_Compute_MIIC.m
%Function:MIIFC computation

function input=Input_Compute_MIIC(y_desired,input,output,
periods,eff_frq_rto,dt,Actual_smpl_rate)

% figure(1);plot([input])
% figure(8);plot([y_desired,output])
%figure(9);plot(output);
y_desired_new=y_desired;
len_input=length(input);
```

%%

```
% 1. Duplicate input, output and desired trajectory
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
input_1=input(2:len_input);
output_1=output(2:len_input);
y_desired_1=y_desired_new(2:len_input);

for ii=2:periods
input=[input;input_1];
output=[output;output_1];
y_desired_new=[y_desired_new;y_desired_1];
end
if periods==1
input=input_1;
output=output_1;
y_desired_new=y_desired_1;
end

ave_input=mean(input);
ave_output=mean(output);
ave_y_desired_new=mean(y_desired_new);

input=input-ave_input;
output=output-ave_output;
y_desired_new=y_desired_new-ave_y_desired_new;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2. Discrete Fast Fourier Transform
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
U = fft(input);
Y=fft(output);
Y_d = fft(y_desired_new);
```

```
len_f=floor(length(U)/2)+1;% Number of Frequcncy component
of input, output, desired tracking output after FFT
transform

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3. Find new input
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
len_H=floor(len_f*eff_frq_rto);% Size of data to be used
in inversion. Specific value (eff_frq_rto < 1) can be
adjusted by user.

% Finding proper even or odd number for len_H depending
on input data size
if length(U)/2==floor(length(U)/2)
if len_H/2==floor(len_H/2)
len_H=len_H-1;
%           disp('Case 1')
else
%           disp('Case 2')
end
else
if len_H/2==floor(len_H/2)
%           disp('Case 3')
else
%           disp('Case 4')
len_H=len_H-1;
end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Core input computation
% U_k+1(jw)=U_k(jw) / Y_k(jw) * Y_d(jw)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sm_out_idx=find(abs(Y(1:len_f))<10^-8);
G_pre=Y(1:len_f)./U(1:len_f);
%
%       G_pre_mag=ones(39,1)*abs(G_pre(41));
%       G_pre(2:40)=G_pre_mag.*cos(angle(G_pre(2:40)))+
G_pre_mag.*sin(angle(G_pre(2:40)))*i;
lnth=len_input;
fD=1/dt*(0:floor(lnth/2))/lnth;

new_u_half_pre=1./G_pre.*Y_d(1:len_f);% First Half size
data from inversion
new_u_half_pre(sm_out_idx)=zeros(size(sm_out_idx));
 % Ignore small output frequency component

new_u_half_pre(len_H+1:len_f,1)=zeros(len_f-len_H,1);
% First Half size data from inversion

x=2:10;
y=abs(G_pre(2:10));

new_u_half_pre(1)=new_u_half_pre(1)-length(Y)*
((ave_y_desired_new-ave_output)/abs(G_pre(2)))+length(Y)
*ave_input;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 4. Input Generation for one period
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
if length(U)/2==floor(length(U)/2)
disp('Even')
new_u_half_post=conj(flipud(new_u_half_pre));% Second
half size data
new_u=[new_u_half_pre ;new_u_half_post(2:len_f-1)];
% Combining First and second half size data
new_input=real(ifft(new_u));
 % Generation new multiple periods of input in real value


else
%      disp('Odd')
new_u_half_post=conj(flipud(new_u_half_pre));
% Second half size data
new_u=[new_u_half_pre ;new_u_half_post(1:len_f-1)];
% Combining First and second half size data
new_input=real(ifft(new_u));
% Generation new multiple periods of input in real value
end
% figure(100);plot(abs(new_u))
% new_u(1)
if periods==1
input=[new_input(length(new_input));new_input];
else
input=new_input((len_input-1)*floor(periods/2):
(len_input-1)*(floor(periods/2)+1));
 % Picking one representing input period
end
figure(1);plot((0:length(input)-1)/
Actual_smpl_rate/1000,input);title('Input signal')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File Name: result_compare.m
% Author: Kim, Kyongsoo (kyongsoo@iastate.edu)
% Date:       May 13,2007
% Inversion-based iterative control sub-file
%   "Error Norm computation"
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Norm_inf,Norm_2_per,Norm_inf_per]=result_X_Y
(output,y_desired1,Actual_smpl_rate)
%l=length(output);
output=output(60001:509000-3000);
y_desired1=y_desired1(60001:509000-3000);

Diff=(output-y_desired1);% Error data
l=length(output);
figure(8);
subplot(211);plot((0:l-1)/Actual_smpl_rate/1000,output,
(0:l-1)/Actual_smpl_rate/1000,y_desired1,'r--');
%legend('output','desired');
subplot(212),plot((0:l-1)/Actual_smpl_rate/1000,Diff);
title('Tracking Error');
Norm_2=norm(Diff,2)/sqrt(length(Diff))*100;
%/norm(y_desired,2)*100;  % 2 Norm of error in percentage
Norm_inf=norm(Diff,inf)*100;
%/(max(y_desired)-min(y_desired))*100; % inf Norm of error
in percentage
Norm_2_per=norm(Diff,2)/norm(y_desired1,2)*100;
%/norm(y_desired,2)*100;  % 2 Norm of error in percentage
Norm_inf_per=norm(Diff,inf)/norm(y_desired1,inf)*100;
```

```
%/(max(y_desired)-min(y_desired))*100;
% inf Norm of error in percentage


%File Name: plot.m
%Function: plot the sample topography
f_sample=200000; %% change
scan_rate=100;   %% change
probe_stiff=100;  %% change
points_n=f_sample/(2*scan_rate);
line_n=120;  %% change
%z_sensor=z_sensor_raw;
%deflection=deflection_raw;
hz=-z_sensor*461;%z_ffd_input   fast image topograph
l=length(x_sensor1);
f_sample1=19000; %% change
f_sample2=3000; %% change
x_sensor=x_sensor1(f_sample1+1:l-f_sample2);
y_sensor=y_sensor1(f_sample1+1:l-f_sample2);
hz=hz(f_sample1+1:l-f_sample2);
x=x_sensor(points_n*1+1:points_n*1+points_n);
y=y_sensor(points_n*1+1:points_n*1+points_n);
h1z=hz(1:points_n);

for i=1:line_n-1
x=[x;x_sensor(2*points_n*i+points_n+1:2*points_n*i+2*points_n)];
y=[y;y_sensor(2*points_n*i+points_n+1:2*points_n*i+2*points_n)];
% change coef before x_sensor and y_sensor
x_coe=0;%-8;
y_coe=0;%-100;
```

```
h1z=[h1z;hz(2*points_n*i+points_n+1:2*points_n*i+2*points_n)+
x_coe*x_sensor(2*points_n*i+points_n+1:2*points_n*i+2*points_n)
+y_coe*y_sensor(2*points_n*i+points_n+1:2*points_n*i+2*points_n)];
end
xi=min(x):(max(x)-min(x))/1000:max(x)-(max(x)-min(x))/(2*points_n);
yi=min(y):(max(y)-min(y))/1000:max(y)-(max(y)-min(y))/(2*points_n);
[xi,yi] = meshgrid(xi,yi);
Hz = griddata(x,y,h1z,-xi,-yi);
figure(1); % fast image topograph
mesh(xi,yi,Hz)
```

References

[1] H.-J. Butt, B. Cappella, and M. Kappl, "Force measurements with the atomic force microscope: Technique, interpretation and applications," *Surface Science Reports*, vol. 59, no. 1-6, pp. 1–152, 2005.

[2] C. Reynaud, C. F.Sommer, N. Bounia, and T. M. Duc, "Quantitative determination of young's modulus on a biphase polymer system using atomic force microscopy," *Surface and Interface Analysis*, vol. 30, no. 1, pp. 185–189, 2000.

[3] G. katsaros, T. Stergiopoulos, I. Arabatzis, K. Papadokostaki, and P. Falaras, "A solvent-free composite polymer/inorganic oxide electrolyte for high efficiency solid-state dye-sensitized solar cells," *Journal of Photochemistry and Photobiology A:Chemistry*, vol. 149, no. 1-3, pp. 191–198, 2002.

[4] S. C. Minne, H. T. Soh, P. Flueckiger, and C. F. Quate, "Fabrication of 0.1 μm metal oxide semiconductor fieldeffect transistors with the atomic force microscope," *Applied Physics Letters*, vol. 66, no. 6, pp. 703–705, 1995.

[5] T. Hugel and M. Seitz, "The study of molecular interactions by afm force spectroscopy," *Macromolecular Rapid Communications*, vol. 22, no. 13, pp. 989–1016, 2001.

[6] G. A. Somorjai and Y. Li, *Introduction to Surface Chemistry and Catalysis*. Wiley, 2010, vol. 1.

[7] J. H. Hoh and P. K. Hansma, "Atomic force microscopy for high resolution imaging in cell biology," *Trends in Cell Biology*, vol. 2, no. 7, pp. 208–213, 1992.

[8] N. Gadegaard, "Atomic force microscopy in biology: technology and techniques," *Biotechnic & Histochemistry*, vol. 81, no. 2-3, pp. 87–97, 2006.

[9] T. Sulchcek, G. G. Yaralioglu, and C. F. Quate, "Characterization and optimization of scan speed for tapping-mode atomic force microscopy," *Review of Scientific Instruments*, vol. 73, no. 8, p. 2928, 2002.

[10] C. Su, L. Huang, and K. Kjoller, "Direct measurement of tapping force with a cantilever deflection force sensor," *Ultramicroscopy*, vol. 100, no. 3-4, pp. 233–239, 2004.

[11] T. Ando, T. Uchihashi, N. Kodera, D. Yamamoto, A. Miyagi, M. Taniguchi, and H. Yamashita, "High-speed afm and nano-visualization of biomolecular processes," *Pflügers Archiv- European Journal of Physiology*, vol. 456, no. 1, pp. 221–225, 2008.

[12] J. Ren, Q. Zou, B. Li, and Z. Lin, "High-speed atomic force microscope imaging: Adaptive multiloop mode," *Physical Review E*, vol. 90, p. 012405, 2014.

[13] A. G. Khurshudov, K. Kato, and H. Koide, "Wear of the afm diamond tip sliding against silicon," *Wear*, vol. 203-204, no. Supplement C, pp. 22 – 27, 1997.

[14] G. Bar, Y. Thomann, R. Brandsch, H.-J.Cantow, and M.-H.Whangbo, "Factors affecting the height and phase images in tapping mode atomic force microscopy. study of phase-separated polymer blends of poly(ethene-co-styrene) and poly(2,6-dimethyl-1,4-phenylene oxide)," *Langmuir*, vol. 13, no. 14, pp. 3807–3812, 1997.

[15] J. M. Sloss, I. S. Sadek, J. C. B. Jr., and S. Adali, "Stabilization of structurally damped systems by time-delayed feedback control," *Dynamics and Stability of Systems*, vol. 7, no. 3, pp. 173–188, 1992.

[16] Z. Wang and Q. Zou, "Iterative-control-based high-speed direct mask fabrication via ultrasonic-vibration-assisted mechanical plowing," in *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2013, p. V003T37A003.

[17] K. S. Kim and Q. Zou, "A modeling-free inversion-based iterative feedforward control for precision output tracking of linear time-invariant systems," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, pp. 1767–1777, 2013.

[18] Q. Zou, K. K. Leang, E. Sadoun, M. J. Reed, and S. Devasia, "Control issues in high-speed afm for biological applications: Collagen imaging example," *Asian Journal of Control*, vol. 6, no. 2, pp. 164–178, 2004.

[19] Y. Wu and Q. Zou, "An iterative-based feedforward-feedback control approach to high-speed atomic force microscope imaging," *Journal of Dynamic Systems, Measurement, and Control*, vol. 131, no. 6, p. 061105, 2009.