

Improving the Precision of the Hard X-Ray Nanoprobe

By Qingyi Wang

Dartmouth College, Hanover, NH

Paper submitted in partial fulfillment of the requirement for

Lee Teng Internship

Mentor: Dr. Curt Preissner

Advanced Photon Source, Argonne National Laboratory

August 13th, 2010

Contents

Abstract	1
1. Introduction	2
2. Measurements and Methods	4
2.1 Noise Measurement.....	4
2.2 Data Reconstruction.....	4
2.3 Open-Loop Frequency Response Measurement.....	7
3. Results and Analysis	9
3.1 LDDM Serial Port Rough Displacement Measurement	9
3.2 FPGA Reconstructed Data.....	9
3.3 Frequency Response Function.....	10
4. Conclusion	12
5. Acknowledgement.....	13
6. References	14
7. Graphs.....	15
8. Tables.....	24

Abstract

Nano-scientific research demands devices that work on nanometer scale resolution and are insensitive to environment vibrations. The hard X-ray nanoprobe was designed by engineers at APS (Advanced Photon Source) to accommodate optics with resolution limit of 10nm. It needs mechanical repeatability of better than 5nm. However, the nanoprobe has vibrations on the scale of 30nm due to ambient mechanical noises. This can limit the speed and accuracy of devices it stages. We want to design a closed-loop control system that can correct the motion of the nanoprobe. As the first step to build the control system, signals from the laser encoder displacement measurement system need to be reconstructed to provide accurate displacement measurement of the nanoprobe. During the Lee Teng internship, we worked on the hard X-ray nanoprobe prototype. We wrote a LabView FPGA module to reconstruct accurate displacement measurement of the nanoprobe. It has been shown that high frequency reconstructed data compare well to low-frequency measurement of the nanoprobe displacement. Since the data reconstruction module is currently working at a frequency slower the desired rate of 12.5 kHz, more studies need to be done to increase its frequency. Frequency response analysis was also performed to develop a model for the control system. The data reconstruction module and frequency response function results can be used towards further development of a closed-loop control system that can effectively correct motion of the hard X-ray nanoprobe.

Research Category: Nanopositioning

School Author Attends: Dartmouth College/Colby College

DOE National Laboratory Attended: Argonne National Laboratory

Mentor's Name: Curt Preissner

E-mail Address: preissne@aps.anl.gov

Presenter's Name: Qingyi Wang

Mailing Address: Hinman 4356

City/State/ZIP: Hanover, NH 03755

Phone: (207) 347-9638

E-mail Address: qwang@dartmouth.edu

1. Introduction

The focus of this project is to work on developing an improved closed-loop control system for the hard X-ray nanoprobe prototype instrument at the APS. A hard X-ray nanoprobe prototype is shown in Fig. 1a. Nano-scientific research demands devices that can work at nanometer level. However, because nanopositioning devices are extremely sensitive to changes in operating conditions such as ambient temperature changes and mechanical vibrations, their performance can limit the accuracy and speed of nano-investigation devices, such as X-ray scanning probe microscopy as described in reference (1) and reference (2). For this reason, it is important to have nanopositioning devices that are insensitive to the operation conditions and can give repeatable and stable measurements.

The hard X-ray nanoprobe is currently operating with 30nm resolution. It is designed to accommodate X-ray optics with resolution limit of 10nm. Therefore it requires staging of the X-ray optics with a mechanical repeatability of better than 5nm. In a typical lab environment, we measured the vibrations of the hard X-ray nanoprobe on the scale of 30nm. In order to achieve robustness to operation conditions, the nanoprobe needs a control system that corrects mechanical vibrations.

The overall goal of our work is to develop a controller similar to the two-degree-of-freedom (2DOF) control system described by Lee and Salapaka (3). A 2DOF control system uses both feed-forward and feedback signals to correct scanning performances. To exploit the feedback signal, we first need to construct a module to decode the incorporated laser encoder system that measures accurate nanoprobe displacement. Once we reconstruct accurate displacement information, we can correct the nanoprobe motion. In order to adapt the method of Lee and Salapaka, we also need to measure the system dynamics and fit a transfer function model to measured responses.

During the Lee Teng internship, my work focused on developing and debugging the displacement reconstruction code. With results from the frequency response analysis, we studied the dynamic of the Physic Instrument (PI) piezo control system. It can be implemented into the closed-loop control system in future studies. The instrument we worked on was built as a prototype for the hard X-ray nanoprobe. To simplify the task, initial measurements and data reconstruction were conducted only on the y-axis of the nanoprobe motion. Once operation for the y-axis has been verified, the same algorithm will be applied to the x-axis. A high-resolution laser-based encoder system measured the position of the motion axis and sent out signals with information of the nanoprobe displacement. Using the laser encoder system signals, a program written for the LabView FPGA module reconstructed the position measurement and returned the actual displacement of the nanoprobe. Frequency response analysis was then performed on signals sent out from the PI piezo sensor/servo control modules and the reconstructed actual displacement of the nanoprobe to model the system dynamics.

2. Measurements and Methods

2.1 Noise Measurement

The Optodyne Inc. LDDM (laser Doppler displacement measurement) incorporated in the nanoprobe has two types of output, “slow” RS-232 data and “fast” encoded analog and digital signals. The RS-232 data can be used to measure the nanoprobe displacement up to a frequency of about 10Hz. The structure of the nanoprobe is shown as in Fig. 1b and Fig. 1c. To quantify motion of the nanoprobe due to vibrations in the environment, we measured displacement of the nanoprobe with the low resolution RS-232 serial port data. Using a LabView Visual Instrument (VI) from previous research conducted on the hard X-ray nanoprobe, we recorded displacement of the y-axis when the control system was not active.

2.2 Data Reconstruction

The first step in controlling the nanoprobe system is accurate measurement and reconstruction of the system displacement. The LDDM laser encoder system measures nanoprobe displacement with high resolution of 2nm. It consists of a rigid reference frame, a set of LDDMs, and laser optics (4). It provides three signals, the up count digital signal, the down count digital signal, and a position analog signal. An example of laser encoder system signals is shown as in Fig. 2. When the actual displacement of the nanoprobe increases, the analog signal moves between 1.7V and 4.2V. As soon as the analog signal reaches its maximum, 4.2V, up count becomes true, and the analog signal goes back to 1.7V. When the actual displacement decreases, the analog signal moves between 0.6V and 3.1V. Down count becomes true when analog signal reaches its minimum 0.6V, and the analog signal goes back to 3.1V immediately. The voltage difference 2.5V corresponds to one wavelength divided by eight, which is $632.8\text{nm}/8=79.1\text{nm}$ for our LDDM. This is because the LDDM laser beam is reflected between fixed prism base and moving prism target for eight times (4).

Thus the actual measurement is scaled off by a factor of eight and eight times more accurate. This discontinuous set of signals needs to be interpreted to reconstruct the actual position.

We used a National Instrument CompactRio FPGA (Fig. 1d) to reconstruct the actual displacement of the nanoprobe. CompactRio input/output hardware modules are connected to the CompactRio chassis with a reconfigurable FPGA core. The FPGA can be programmed to read or write signals from the input/output modules. It can conduct local signal processing and pass information from one module to another. Because the chassis has a 40 MHz onboard clock, operations on each module can be synchronized with 25ns resolution. The FPGA is also connected to the embedded real-time processor via a high-speed PCI bus. The real-time controller can retrieve data from the FPGA board and perform control, waveform analysis, data logging, and Ethernet/serial port communication. National Instruments hardware and software can work on real-time at a maximum frequency of 1MHz, which is slower than the FPGA. Calculations that have greater priority should be performed on the FPGA level.

The data reconstruction was conducted on LabView FPGA level, and results were displayed and logged on the host computer at 80 us intervals (12.5 kHz). A block diagram of the Data Reconstruction FPGA module is shown as in Fig. 3a. We used a NI9024 real-time controller, a NI9118 reconfigurable chassis embedded with a Virtex-5 LX110 FPGA, an NI9401 CompactRio digital input module, an NI9239 CompactRio analog input module, and an NI9263 CompactRio analog output module. A VI named Position Count calculated the position base using the digital inputs. The logic of the Position Count module is shown as in Fig. 3b. The Position Count VI outputted position count that provided a base for the analog signal. When summed up with the analog signal, it generated a value as if the analog signal was never interrupted at its maximum or minimum levels. In Data Reconstruction VI, calculation was conducted with the position count and the analog input to give the actual displacement of the nanoprobe. Actual position data were then

sent from the FPGA level to the real-time level, displayed on a waveform chart, and logged into a file on PC.

Before we decoded the signal, we had to address a limitation of the NI9401 digital input module. According to the NI9401 manual, the minimum digital signal pulse width that can be detected by the NI9401 module is 100ns. However the pulse width of the laser encoder system digital output is 64nm. We generated 1Hz digital signals of 10ns, 60ns, 100ns, and 200ns pulse width to NI9401 module's sensitivity. As shown in Fig. 4, at 60ns pulse width, signals were barely visible to the device. Only when the pulse width was greater than 100nm, the NI9401 module recognized signals clearly and stably. To make up count and down count digital signals visible to the NI9401 module, we used two external trigger function generators. When triggered by laser encoder system digital signals, external trigger function generators would output signals with 350nm pulse width to the NI9401 module. The process is shown as in Fig. 5.

Loops inside FPGA VIs tend to iterate at the fastest rate possible. Nevertheless, depending on the amount of calculation and data acquisition conducted inside each loop, it might take a long time for the loop to execute once. The digital signals going into the NI9401 module had 350ns pulse width, which were eight clock ticks. To assure that digital signals were recognizable, the loop in which digital signals were read should be iterating at a frequency greater than $1/350\text{ns}=2.8\text{ MHz}$. We used several benchmark VIs to test the execution rate of each loop. Only when digital input nodes were put in independent loops did the FPGA board have enough time to read them correctly. The structure of benchmark tests is in the format shown in Fig. 6. We placed the interested data acquisition or calculation loop inside the middle frame of the LabView flat sequence structure, set number of iterations on the numeric control, and ran the VI. The number returned is the time taken to run the loop for the set number of iterations. The results of the benchmark tests are as shown in Table 1.

It is also important to decide the rate at which the analog signal would be collected. We expect important motions of the nanoprobe stages to be 5 kHz. To meet the Nyquist sampling theorem, we need to acquire our data at a minimum of 10 kHz. The NI9239 analog input module we used has a built-in anti-aliasing filter. The effect of the anti-aliasing filter reduces the available bandwidth, the so-called alias-free bandwidth. As specified on the NI9239 manual, its alias-free bandwidth is $0.453 \cdot f_s$ (sampling frequency). Thus if we want an alias-free bandwidth of at least 5 kHz, we need a sampling rate of $5\text{kHz}/0.453 \cdot f = 10.9645\text{ kHz}$. The closest sample rate we can set for NI9239 is 12.5 kHz. We also set the rate at which Calculate Position VI updates to 12.5 kHz.

2.3 Open-Loop Frequency Response Measurement

In order to design a controller, the behavior of the system needs to be measured. The behavior can be quantified with the frequency response function (FRF). Our FRF is a measure of our system output for a unit input across the 5 kHz frequency range that we are interested in. To measure the FRF, the system was driven in such a way that all significant dynamics were excited within the 5 kHz frequency range with a noise signal. The control voltage and the actual displacement of the nanoprobe were then measured simultaneously, and the FRF was calculated in MatLab.

PI piezo actuators with strain-gauge sensor/servo control modules are used to drive the high-resolution and medium-resolution weak link stages of the nanoprobe (4). Most of the current high precision positioning devices utilize piezoelectric materials for actuation. Our system consists of two E501.00 chassis, two E503.00 piezo amplifiers, two E509.x3 sensor/servo modules, and six p-814.1B piezo actuators. Two actuators control the x-axis, while the other four control the y-axis. The sensor/servo control modules send out 0 to +10V signals to control the piezo expansion.

On the y-axis the motion ratio is 4:15, and the output sensitivity is:

$$\frac{1V}{1} \times \frac{100V}{10V} \times \frac{15 \times 10^{-6} m}{100V} \times \frac{4 \times 10^{-6} m}{15 \times 10^{-6} m} = 0.4 \times 10^{-6} m/V$$

In terms of displacement this is 2.5mV/nm.

A random waveform was generated on LabView FPGA level and sent to the PI through the analog output. The analog voltage then drove the sensor/servo control modules and controlled the y-axis piezo actuators. At the same time, the Data Reconstruction FPGA module reconstructed the y-axis actual displacement. Block diagram of the system identification is shown in Fig. 7.

The control voltage and reconstructed actual displacement of the y-axis were recorded. We used MatLab TFESTIMATE function to perform frequency response analysis on the data.

3. Results and Analysis

Before we could implement the data reconstruction algorithm into our control system, we need to verify that it correctly measures the displacement. Comparison between the LDDM slow measurement and the reconstructed data was made. Also we need to measure the frequency response function of the system to quantify the control system.

3.1 LDDM Serial Port Rough Displacement Measurement

LDDM serial port displacement measurements were logged when the instrument was not active. Data are shown as in Fig 8. Fig 8a is measurement of 30-second intervals over a two-hour period. Since the nanoprobe is extremely sensitive, daily temperature change and noise in the environment induces vibrations of the instrument. Fig 8b is measurement of 1-minute intervals over four days. The displacement peaked every night at the same time. This was probably due to the noise made by air conditioning system or the water treatment system in the APS or due to the room temperature increase during the afternoon.

3.2 FPGA Reconstructed Data

FPGA Data Reconstruction output was compared with LDDM serial port slow measurement. As shown in Fig. 9a, the two lines have the same trend. The red reconstructed data line has greater amplitude and peaks more often. This is because the reconstructed data were measured at a much faster rate (80us, 12.5 kHz) than the LDDM serial port data collection (100ms, 10 Hz). Thus the reconstructed data shows more detailed high frequency readings that could not be detected by the LDDM serial port slow measurement.

Measurements were also performed when the nanoprobe was driven by a 1V peak to peak sine wave from the PI piezo controller. A 1V peak-to-peak sine wave corresponds to 400nm sine wave movement of the nanoprobe. On Fig. 9b, the reconstructed data show a 3200nm peak-to-peak sine wave movement. Our scaling was off by a factor of eight. So the reconstructed data was a 400nm sine wave. Calculate Actual Position VI was later changed to correct the scaling. The reconstructed data correspond well with control voltage. However there are still glitches on the reconstructed waveform. The ones on the upward slopes are pointing down while the ones on the downward slopes are pointing up. This is probably because on the FPGA level the Position Count loop is updating one cycle behind the Calculate Actual position loop, and position count is not updating fast enough to catch up with the change in analog signal.

3.3 Frequency Response Function

The transfer function estimate of our piezo control system is shown in Fig. 10. The x-axis of the graph is normalized. The frequency range of our data is from 0 to 12.5 kHz. And we are only interested in FRF of our output across the 5 kHz range. That would be the frequency response on the left of the black line.

We expected one or a few peaks in the 5 kHz range. However the transfer function does not show any peaks except when the frequency is very close to normalized zero. After looking at the control voltage and reconstructed displacement data, we realized that the data was updating at a rate much slower than the 12.5 kHz we set. We conducted a benchmark test on the speed of the analog input to test its sample rate. The analog input sampled input at a rate of about 10 ms per iteration (100 Hz). Even if we set the Calculate Actual Position loop rate to be 12.5 kHz, the actual speed at which the data was updated was no faster than 100Hz. Benchmark test results are shown in Table 2. The peak that is very close to normalized zero corresponds to the very slow rate at which Calculate Position Actual VI updated. We also tried changing the loop cycle from 80us (12.5

kHz) to 10ms (100Hz). Changing the loop cycle did not change the resolution of our reconstructed data. This verifies our speculation that the Calculate Actual Position VI is running at a much slower rate than we expected.

4. Conclusion

During the Lee Teng internship, we developed a module that works on National Instrument CompatRio FPGA to reconstruct the hard X-ray nanoprobe displacement information. The module utilizes LDDM laser encoder system digital and analog signals to calculate correctly the actual position of the nanoprobe.

However, after frequency response analysis, we realized that the calculation was not performed at the frequency we expected. In order to achieve the 12.5 kHz data reconstruction rate we aimed for, we still need to address the hardware limitations and come up with a more efficient way to make calculations on the FPGA level.

5. Acknowledgement

I would like to thank the United States Department of Energy – Science Division and Argonne National Laboratory for giving me the opportunity to participate in the Lee Teng Internship in summer of 2010.

I am very grateful to my mentor, Dr. Curt Preissner, for providing guidance and giving me the opportunity to work on this project.

The work described in this paper was performed at Advanced Photon Source, located in Argonne National Laboratory, a national scientific user facility sponsored by US Department of Energy.

6. References

1. *High bandwidth nano-positioner: A robust control approach*. **S. Salapaka, A. Sebastian, J. P. Cleveland, M. V. Salapaka**. 2002, Review of Scientific Instrument, p. 10.
2. *Focus on X-ray Diffraction*. **Chapman, Henry N.** 2008, Science, p. 2.
3. *Two Degree of Freedom Control for Nanopositioning Systems: Fundamental Limitations, Control Design, and Related Trade-offs*. **Salapaka, Chibum Lee and Srinivasa M.** St. Louis : Authorized licensed use limited to Argonne National Laboratory. 2009 American Control Conference. p. 6.
4. *Optomechanical Design of a Hard X-ray Nanoprobe Instrument with Nanometer-Scale Active Vibration Control*. **D. Shu, J. Maser, M. Holt, R. Winarski, C. Preissner, A. Smolyanitskiy, B. Lai, S. Vogt, and G. B. Stephenson**. s.l. : American Institute of Physics, 2007. Synchrotron Radiation Instrumentation: Ninth International Conference.
5. **Moheimani, S. O. Reza**. *Invited Review Article: Accurate and fast nanopositioning with piezoelectric tube scanners: Emerging trends and future challenges*. New South Wales : Review of Science Instruments, 2008.

7. Graphs



Figure 1a. Nanoprobe Prototype (left) and the piezo controllers (right).

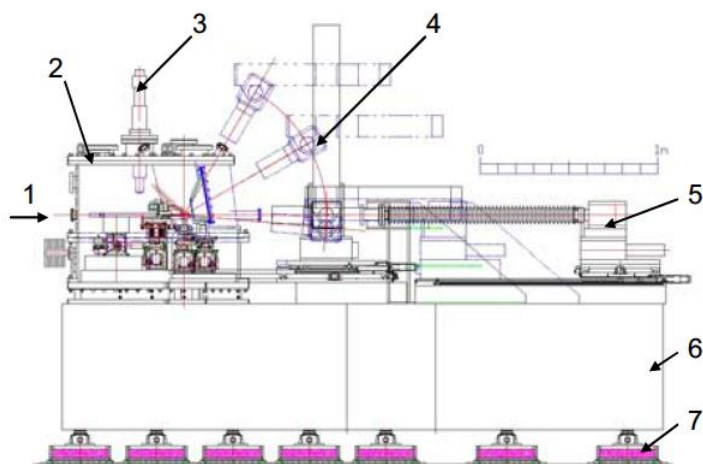


Figure 1b. Side view (4) of the hard X-ray nanoprobe instrument. [1] Incident beam; [2] instrument chamber; [3] Optical microscope; [4] Diffraction detector; [5] Transmission imaging detector; [6] Granite base; [7] Isolator.

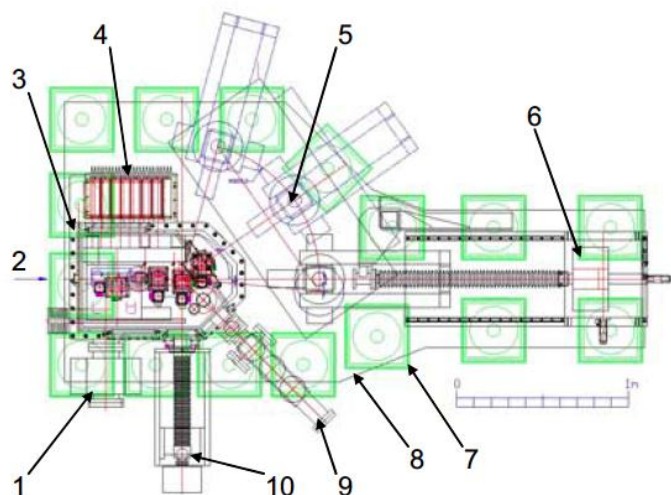


Figure 1c. Top view (4) of the hard x-ray nanoprobe instrument. [1] Ion pump; [2] Incident beam; [3] Instrument Chamber; [4] Laser head for LDDM; [5] Diffraction detector; [6] Transmission imaging detector; [7] Isolators below base; [8] Granite base; [9] Airlock for specimen exchange; [10] Fluorescence detector.



Figure 1d. LabView FPGA chassis embedded with real-time controller and digital/analog input/output modules.

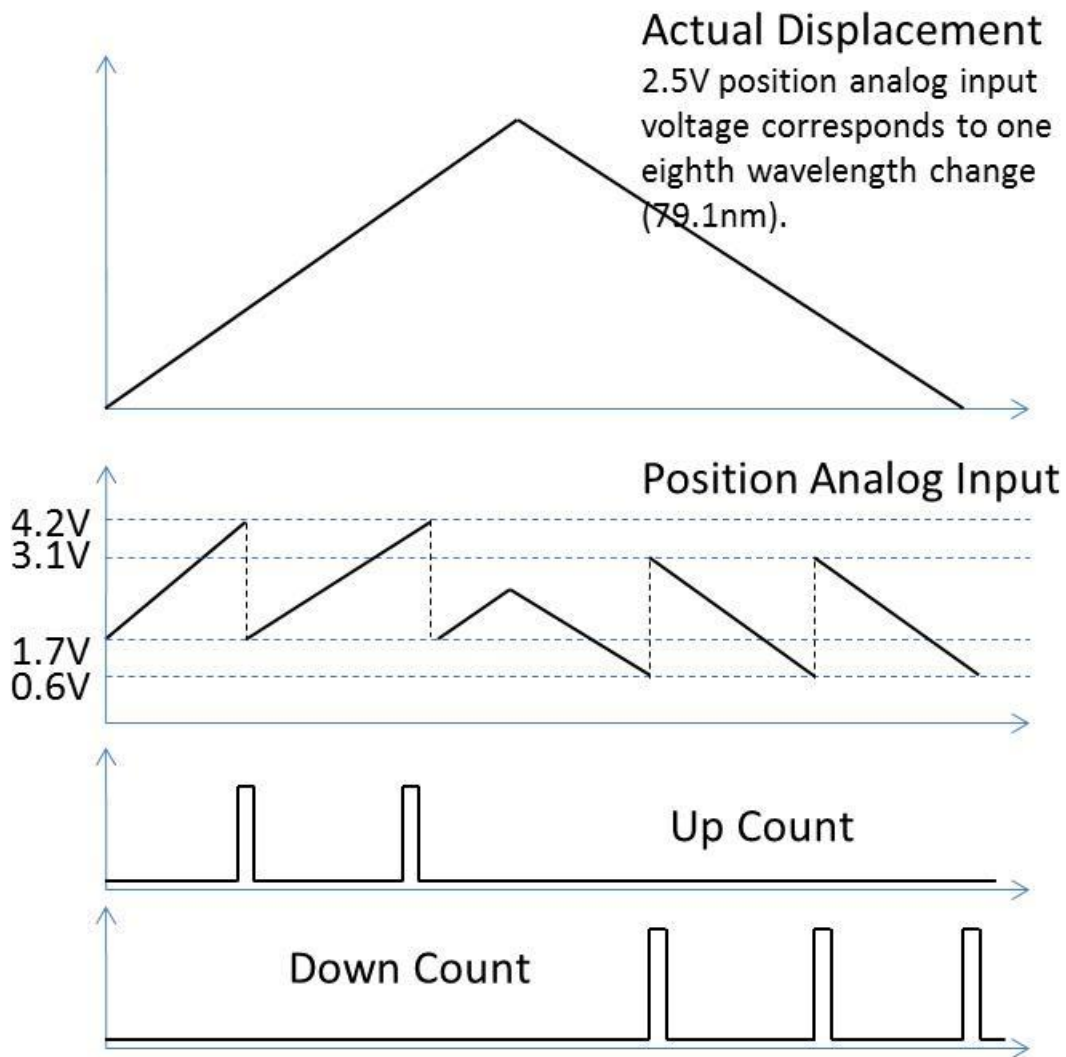


Figure 2. Example of laser encoder system output signals. Position analog input moves between 1.7V and 4.2V when the actual displacement increases. It moves between 0.6V and 3.1V when the actual displacement decreases. Up count becomes true when analog input reaches its maximum. Down count becomes true when analog input reaches its minimum.

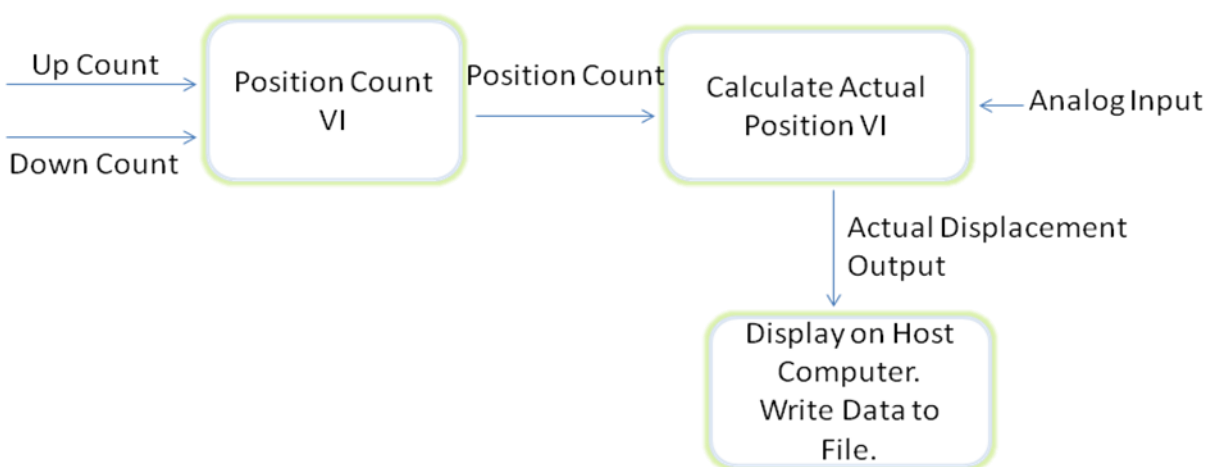


Figure 3a. Block diagram of the Data Reconstruction LabView FPGA Module. Position Count and Calculate Actual Position are performed on FPGA level, while data logging is conducted on the host computer level.

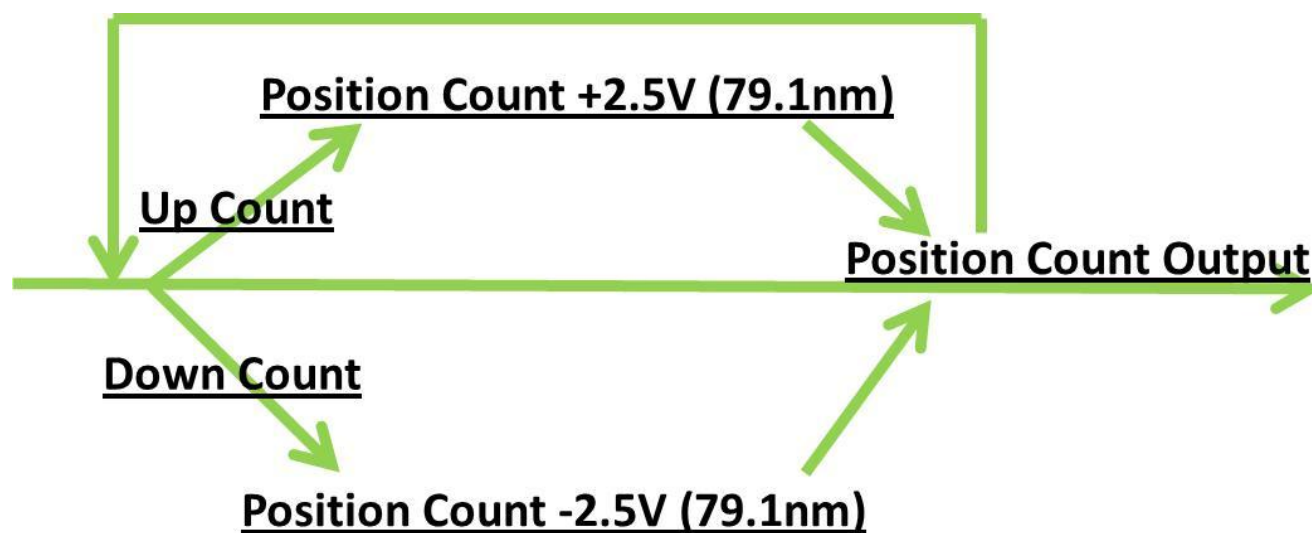


Figure 3b. Position Count VI Logic. Inputs are up count and down count digital signals. Output is position count.

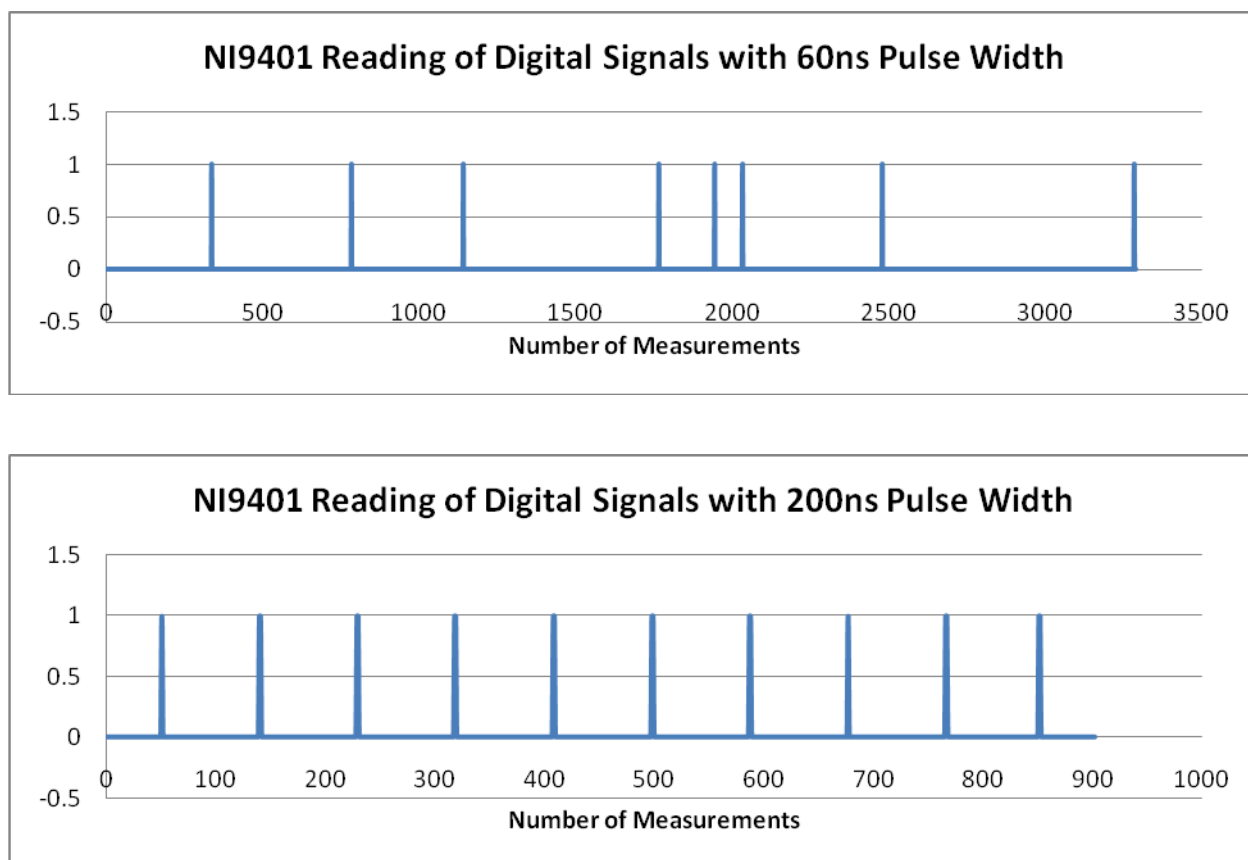


Figure 4. NI9401 digital module reading of 1Hz digital signals with 60ns and 200ns pulse widths. NI9401 module has a maximum switching frequency of 10MHz. When the pulse width is smaller than 100ns, the signals cannot be read distinctly.

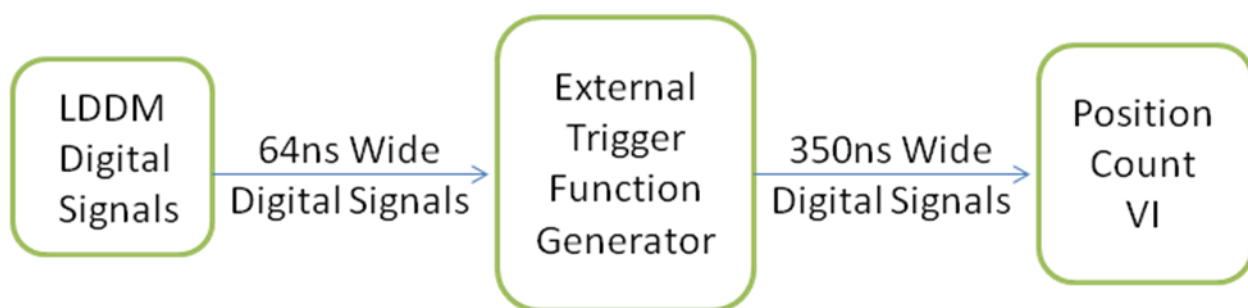


Figure 5. Block diagram of how the LDDM digital signals of 64ns pulse width are transformed to 350ns wide digital signals.

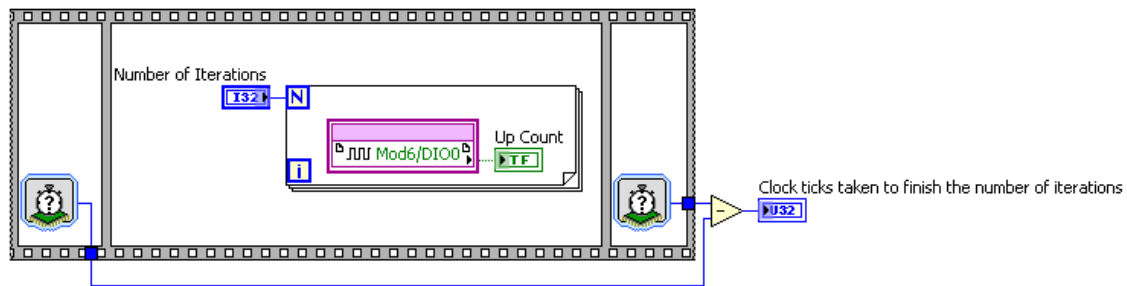


Figure 6. Block diagram of a benchmark test VI. Benchmark tests are used to measure the minimum time needed for each loop to execute once.

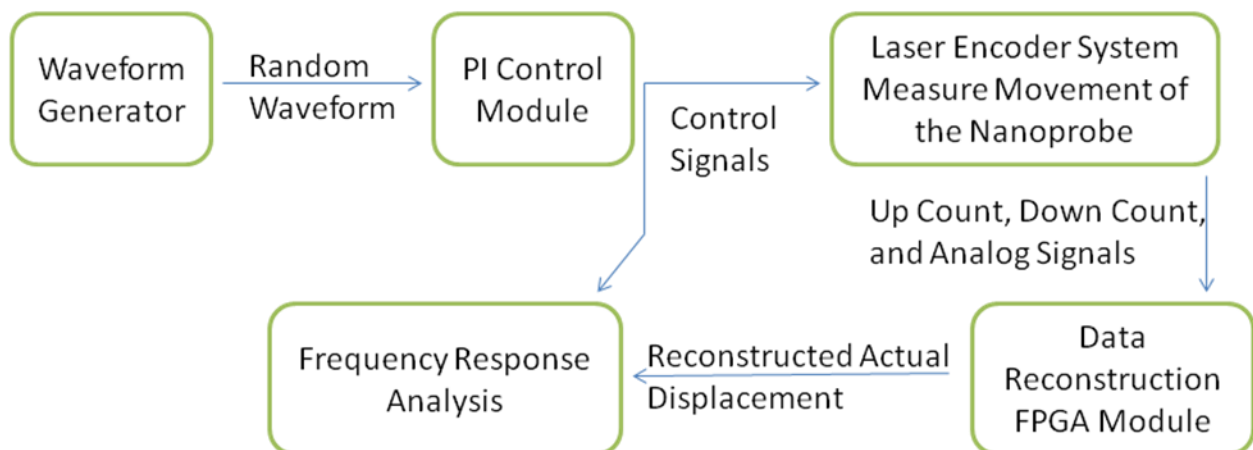


Figure 7. System Identification LabView module block diagram.

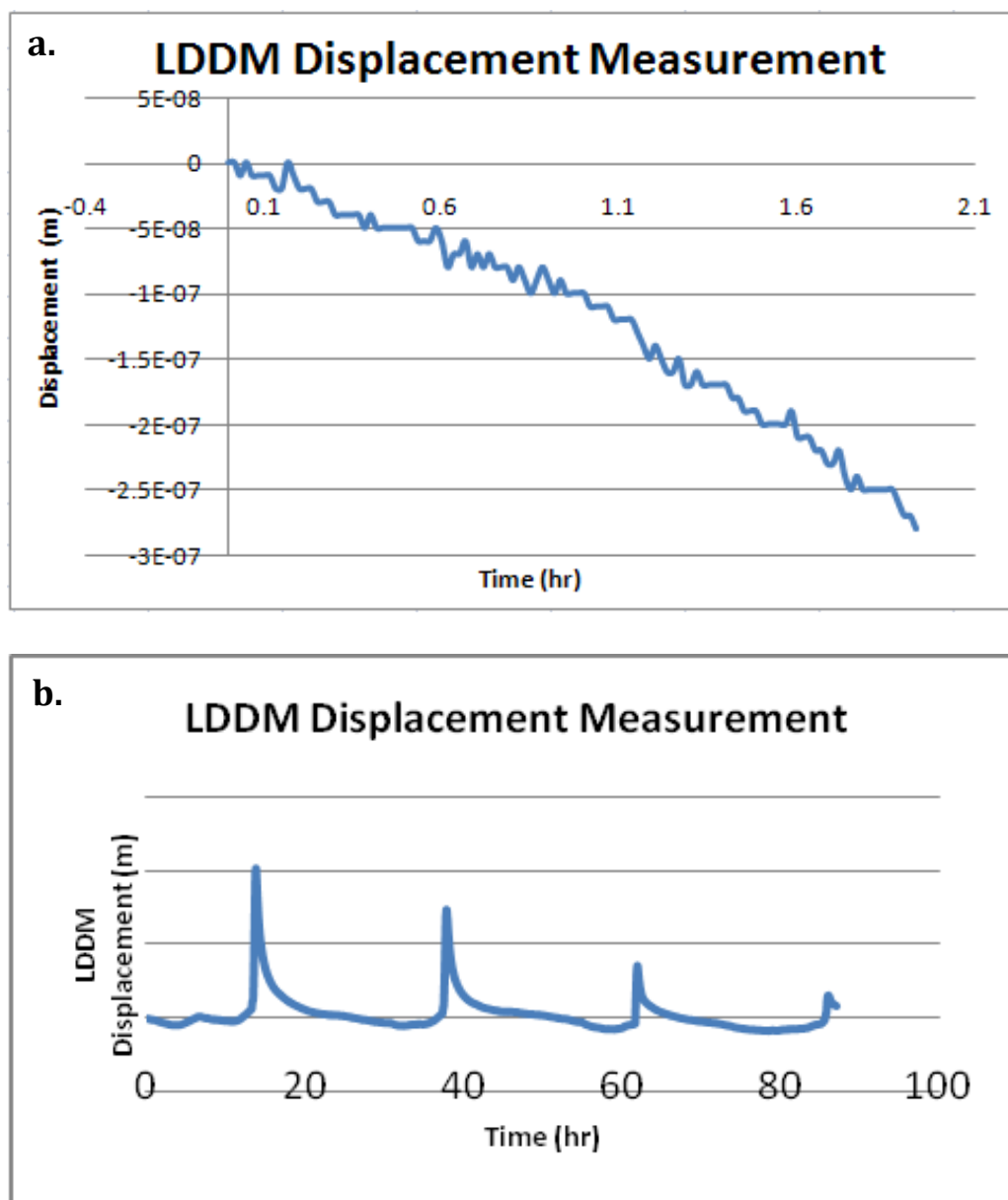


Figure 8. a. LDDM serial port displacement measurement taken at 30s intervals over two hours.
b. LDDM serial port displacement measurement, taken at 1min intervals over four days. The displacement peaks at the same time every day.

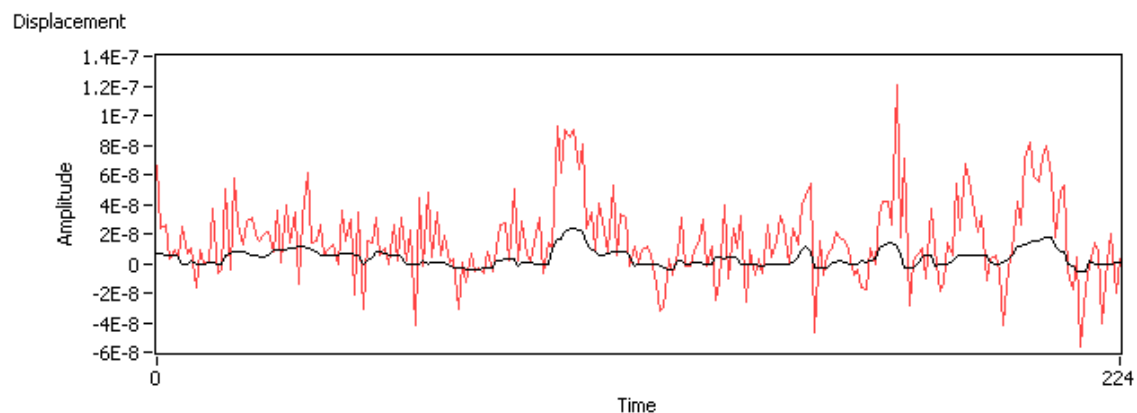


Figure 9a. LDDM serial port data (black) compared with reconstructed displacement data (red). The reconstructed data is much more accurate that we can see more high frequency measurements.

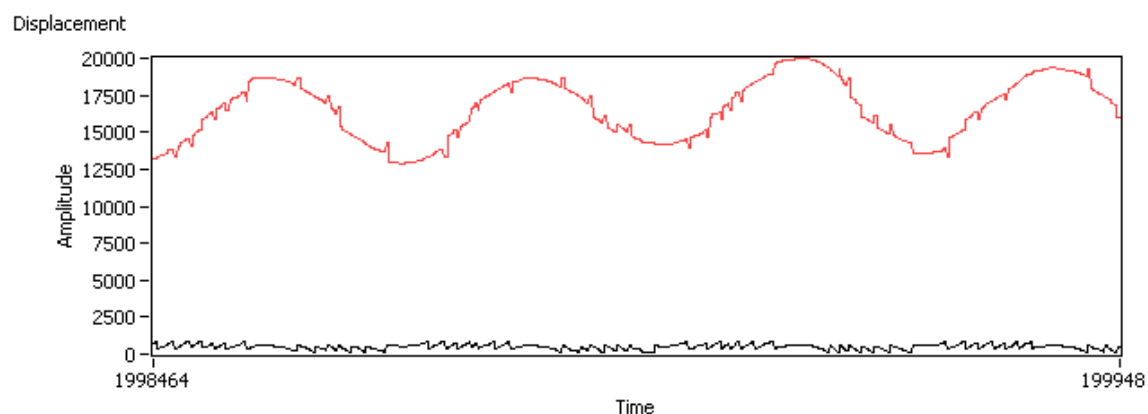


Figure 9b. Reconstructed data compared with analog input when the nanoprobe was driven by a 1V sine wave.

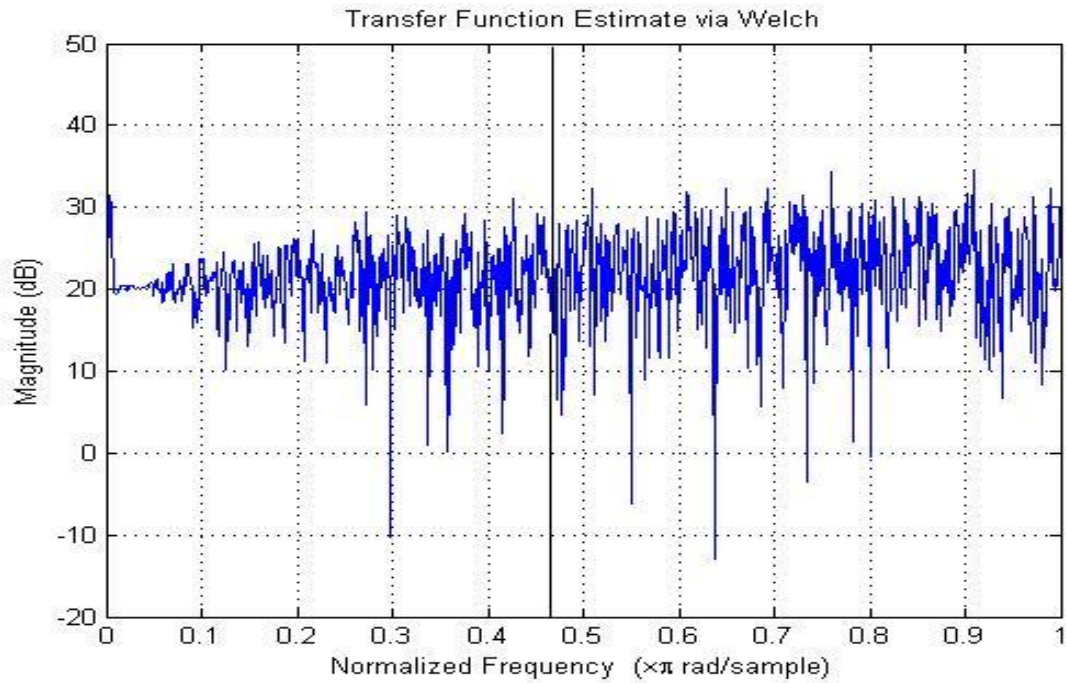


Figure 10. Transfer function for our PI piezo control system. The 0 to 12.5 kHz range is normalized. We are interested in the 0 to 5 kHz range, which is on the left of the black line.

8. Tables

Part Tested	Iteration Duration,
One digital input	$\approx 5 \text{ ticks} = 125 \text{ ns}$
One digital input with a loop timer set to 4 ticks (100ns)	$\approx 5 \text{ ticks} = 125 \text{ ns}$
Two analog inputs	$\approx 5 \text{ ticks} = 125 \text{ ns}$
Position Count VI	$\approx 18.5 \text{ ticks} = 462.5 \text{ ns}$

Table 1. Benchmark test results. Benchmark tests conducted to test the setup for digital input and position count VI calculation speed.

Calculate Actual Position Count VI	$\approx 4,000,000$ ticks = 10ms
Analog Input	$\approx 3,800,000$ ticks = 9.6ms

Table 2. Benchmark test results on analog input and Calculate Actual Position VI.