Integrating nanotechnology into undergraduate experience, a web-based approach

Timothy Chang, Puttiphong Jaroonsiriphan, [1] and Xuemei Sun [2]

Abstract

This paper describes a web-based distance experiment on the control of a two-degree-of-freedom (DOF) monolithic piezoelectric actuator. This actuator is part of a six DOF manipulator [1] capable of linear resolution to 2 nanometers and angular resolution to 1 arc-second [2]. The LabVIEW/Datasocket approach has been adopted in this implementation to enhance streaming of live data as well as to facilitate the interaction among the various system modules such as PC platform, digital signal processor board, client/server interface, and the nanopositioner. Besides LabVIEW based client program which requires the remote PC to have LabVIEW installation, a web-browser based client plugin has also been generated for students/users who do not have programming experience with LabVIEW. Test results for multi-client access have been very favorable. Reliable real time control bandwidth of the two DOF nanopositioner was detected at about 3KHz, adequate for undergraduate control experiments. The same client/server interface can also be applied to a wide range of experiments for 24/7 and global access.

1 Introduction

It is well recognized that nanotechnology is a strategically important area with significant growth potential. Therefore, it is of education interest to expand the study of nanotechnology from advanced graduate level to senior undergraduate level. The challenge with offering nanotechnology to undergraduate students is not so much the course content or lecture design, but rather, it has to do with the experimental issues. Nanotechnology experiments are delicate, limited in availability, and expensive to set up and maintain. The use of web-based approach circumvents these drawbacks and enables the experiment to be run securely, safely, and on a 24/7 basis. In this paper, the experience in offering a nanopositioner control experiment to a senior undergraduate class is reported. The nanopositioner is a monolithic piezoelectric device capable of six degree-of-freedom and resolution to about 1 nanometer. The experimental unit has two DOF (x and y) and is driven by two high voltage amplifiers (+/- 550 Volts) from a TMS320C31 based digital signal processor board. Two capacitance sensors are used to detect the positioner displacement for frequency up to 3kHz. Concerns for safety of the students (since high voltage drives are used)and the hardware (delicate and expensive) have been the main reasons for restricting direct student access. In this project, the students access the experiment through LabVIEW Datasocket. The server computer houses the TMS320C31 digital signal processor board and is responsible for client interface and running of the DSP functions such a code compilation, execution, and data transfer. Since the client access is restricted to code operation in the DSP, security of the PC is increased. Furthermore, a number of safeguards such as actuation level, slew rate, etc. have been set in to minimize the possible damage to the experiment.

2 Description Of The 2 DOF Actuator

The nanopositioner described in this paper is a monolithic design with very low component count and therefore, can be constructed with significantly lower cost. Applications of this actuator include: optoelectronics assembly, optical fiber alignment, and semiconductor processing. Preliminary test results from the 2 DOF prototypes indicate the following performance values:

resolution	2 nm
rise time	1.5μm/ms
range	50 μm
load	1 kg
input voltage	500V
angular coupling between x, y axes	$< 1°$
resonance	$>$ 500Hz

Refer to Figure 1, the 2 DOF actuator consists of an interlaced stack of monolithic PZT cruciforms and inert baseplates. Motions in the x, y-plane are generated by applying voltages to the electrodes on the PZT cruciforms. The actuator is constructed from piezoelectric ceramic such as Lead Zirconate/Lead Titanate (PZT or LZT) which exhibits microstrain upon proper poling and electrical excitation. With its monolithic structure, a total of three DOF can be designed into a single crucifrom actuator (x, y, and theta axes. The static displacement of an actuator with length L, width W and thickness T varies according to the following formulas:

$$\Delta x = \frac{d_{31} L}{T} n_b V \quad (1)$$

$$\Delta y = \frac{d_{31} W}{T} n_b V \quad (2)$$

[1] Timothy Chang and Puttiphong Jaroonsiriphan are with the Department of Electrical & Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102. E-mail: changtn@adm.njit.edu .

[2] Xuemei Sun is with Rochester Hills Software Center, Motorola AIEG 1840 Enterprise Dri., Rochester Hills, MI 48309.

where the piezoelectric constant d_{31} is expressed in m/V. Δx, Δy are the relative displacement in the x, y directions and V is the applied voltage. n_b is the number of PZT cruciforms. It should be mentioned that the cruciform operates in the d_{31} mode so stacking of elements must be interlaced with an inert baseplate to enable addition of the individual displacements. The theta angular motion is realized by etching proper electrode patterns on the cruciform [1]. The pressure produced by the deformation is given by:

$$P_x = P_y = \frac{d_{31} Y_{11}^E}{T} V \tag{3}$$

where Y_{11}^E is the Young's modulus of the material. With soft ceramic, typical values of d_{31} and Y_{11}^E are: -250×10^{-12} m/V and 6×10^{10} N/m^2. Maximum applied voltage is about 1kV per millimeter thickness. An actuator constructed from ten 1cm side and 0.5 mm thick baseplates produces a range of 25 microns and a force of 75N under 500V peak excitation in the x, y plane. Figure 1 illustrates the geometry of a single piezoelectric baseplate and its corresponding x axis motion under differential voltage excitation at its electrodes. Similar motion can be generated for y axis.

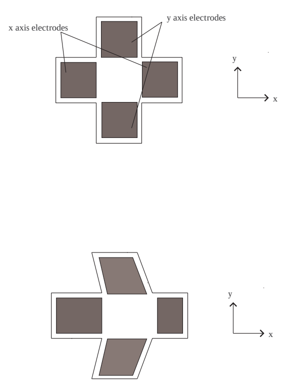

Figure 1: Cruciform baseplate and its motion

The corresponding minimal order analytic realization, modeling dynamics up to the first resonance, is given by (4) and (5):

$$\begin{aligned} &\frac{d^2x}{dt^2} + 1.0502 \times 10^3 \frac{dx}{dt} + (1.3511 \times 10^7)x \\ &-(9.4572 \times 10^5)x^2 + (4.4607 \times 10^4)x^3 \\ &= (1.3984 \times 10^5)u_x \end{aligned} \tag{4}$$

$$\begin{aligned} &\frac{d^2y}{dt^2} + 7.742 \times 10^2 \frac{dy}{dt} + (1.0572 \times 10^7)y \\ &-(7.4200 \times 10^5)y^2 + (2.9927 \times 10^4)y^3 \\ &= (1.4244 \times 10^5)u_y \end{aligned} \tag{5}$$

where x, y are displacements expressed in microns and u_x, u_y are drive voltages on the PZT. It is observed that the first resonances for the x and y axes occur at 585 Hz and 518 Hz respectively. These resonances have a Q factor of about 5-7. Characteristics of the two axes are quite similar with low mutual interaction due to the orthogonal structure. Test results indicate that as each axis is traversed through its full range of motion, the orthogonal member undergoes less than 1 degree of misalignment. The difference in the resonant frequencies is caused by fabrication tolerance and the way the capacitance probe target is mounted.

Experimental setup of the test system is shown in Figure 2. It consists of the following: two TREK high voltage amplifiers, two ADP Capacitance sensors, a TMS320C31 based digital signal processor board with four, 14-bit input channels and two 12-bit output channels.

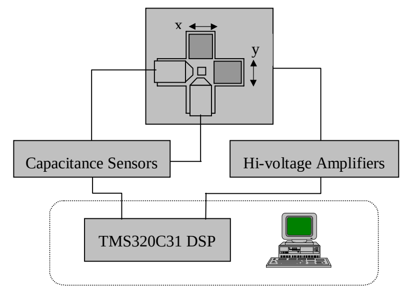

Figure 2: Experimental setup

3 Web Access with Datasocket

The National Instruments' LabVIEW (Laboratory Virtual Instrument Engineering Workbench) has been a popular software for academic and industrial control experiments. It features graphics user interface and drivers for controlling real time processes. DataSocket, included with LabVIEW, is a new Internet programming technology that simplifies live data exchange among computers connected through a network [9], [10], [11]. DataSocket is designed specifically for sharing, subscribing, and publishing live data in measurement and automation applications. In particular, datasocket is designed for live data transfer to multiple clients where a URL is used by users to connect to a data source location in the DataSocket server so as to control or share data with DataSocket applications. Although DataSocket handles connections to various types of data sources and existing schemes such as http, ftp, file, and opc, server communicates and exchanges data between two applications using the dstp protocol (DataSocket Transfer Protocol). The three components in a Datasocket subsystem are: 1. the DataSocket Server, 2. the writer (the digital signal processing board for this application), and 3. the reader (remote client). The advantages of maintaining the DataSocket Server on a separate machine other than the writer are increased robustness, scalability, and security. According to National Instruments [10], DataSocket simplifies network (TCP/IP) programming by automatically managing connections to clients and automatically converting measurement data, regardless of the types of data used or the source of the data, to and from the stream of bytes sent across the network. This automatic conversion eliminates the need to write complicated parsing code and network complexity. Therefore, users do not have to write the parsing code. Also, users can switch between data sources just by changing a simple URL. DataSocket is an ActiveX control, thus users can use it to develop data-sharing applications in ActiveX containers such as Visual Basic, and Visual C++. This aspect simplifies the use of DataSocket to broadcast live data. There are two modes of client/server operations under the Datasocket setup:

1. LabVIEW based client program where a VI program runs within the LabVIEW environment in the client computer. This is the preferred mode for research activities as the user can interactively expand on the data processing capability by adding more VIs to the system block diagram. However, this mode of operation necessitates the client to have LabVIEW software and G-programming experience.
2. Web Browser based client program where a plugin to the web browser can be download by the user and the virtual instrument panels are activated within the web browser. This way, the user can operate the experiment in real time with minimum learning time. However, the interface configuration is fixed and cannot be modified. This is the preferred mode for undergraduate students who are more limited in time, resource, and programming experiences. This mode requires the experiment designer to create a DataSocket reader component (ActiveX control), using ComponentWorks and Visual Basic codes to manipulate those controls, and then insert this component into a web page.

Finally, to protect from undesirable users taht could misuse this application, DataSocket Server Manager can handle this problem by configuring the DataSocket server setting. Users can specify which machines can create items, write items, or read items; including access permission, number of connection and predefined data items stored in the server. Users can connect a total of 1000 readers and writers to one DataSocket Server. The overall system block diagram is given in Figure 3 below:

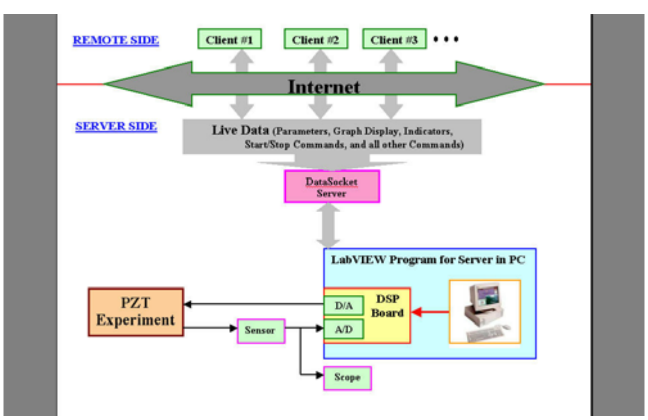

Figure 3: System Block Diagram

4 Synthesis of Datasocket VI and plugin

To create DataSocket client Web page, users have to create a DataSocket reader component with which users connect to the server. Users create that component with ActiveX controls and Visual Basic codes to manipulate those controls and save this component as an ActiveX control. After making the ActiveX control, it is plugged into a Web page.

The necessary software tools to build ActiveX control include:

1. ComponentWorks 2.0 DataSocket and User Interface (UI) controls
2. Visual Basic 5.0 or higher (I use version 6.0)
3. Recent versions of Netscape or Internet Explorer.

The goal is creating the DSReader component using ActiveX controls in Visual Basic, and design a component that connects to the DataSocket Server, which reads live data from the server, and displays that data on a graph.

The procedure is listed below:

1. Loading ActiveX Controls by opening ActiveX control, then Selecting National Instruments CW DataSocket and National Instruments CW UI.
2. Place all control components including Graph, Control buttons, CWDataSockets, Textbox, etc. onto the control form.
3. Turn on the server station on and load program to DSP board.
4. Disable all parameter-control buttons until the client presses "Run Client"
5. After a client press "Run Client" button, the CWDataSockets are automatically connect to DataSocket Server specified by IP address and item names, then the program does as the following:
   (a) CWDataSocket 1 is in the "Write Mode" for writing and sending data (all data in attributes) to Server.
   (b) CWDataSocket 2 is in the "ReadAutoUpdate Mode" for reading data (parameters) from Server.
   (c) CWDataSocket 3 is in the "ReadAutoUpdate Mode" for reading data (graph data and all data in attributes) that are generated from server, and then these data are display on the Client side.
   (d) CWDataSocket 4 is in the "Write Mode" for sending Start/Stop command to control the server.
6. After all parameter-control buttons are enabled, clients can control the server with indicators and digital display shown during run time.
7. In case that many clients are logging on to the server at the same time, the first client will be absolutely the first priority to take control until he presses "Stop Client"

8. As soon as, the first client presses "Stop Client", all CWDataSockets are disconnected. This action causes Server stop running and all the rest of clients are standstill.
9. In case that, clients need to set priority to control server, all clients have to stop program and connect to the DataSocket Server again. Therefore, whoever logs on first will be the first priority to control the server.
10. In ComponentWorks application with Visual Basic, it is possible to publish or subscribe data through DataSocket Server directly without creating any Visual Basic codes. This property is called "Bindings", which a programmer just identifies URL/item name and values/data to publish or subscribe for each component. The other component(s) communicating with that component have to be specified the same URL and item name. After the program is running, this handshaking will take effect automatically. Therefore, some procedures that publish or subscribe data the server are not shown in the Visual Basic Code.
11. After create Visual Basic codes, programmer has to generate the ActiveX control, whose file extension is .OCX
12. Creating the HTML File by using the Package and Deployment Wizard (Visual Basic 6) will generate an Internet distribution package, which contains a cabinet (.cab) file and the HTML file. The cabinet file contains information about the DSReader component, and it is automatically downloaded, expanded, and installed by Internet Explorer so that clients on other computers can view the program component with their Web browser.
13. There are three files associated with the Visual Basic project: .vbp (project file), .ctl (DSReader component), and .vbw (workspace file). Programmer needs these files to modify the program component in Visual Basic. After modifying the Visual Basic project, programmer needs to remake the .ocx file, which saves the program component as an ActiveX control that can be embedded in an HTML (.htm) file.
14. After programmer makes the .ocx file, he can delete previous .cab and .htm files and use either the Application

5 Results

Some typical screens of the client/server are shown here. The server socket panel and server socket block digram are shown in Figures 4 and 5 respectively. The screen design uses standard VIs from LabVIEW and is completely reconfigurable. Even though the block diagram seems complex, it is capable of running at sufficient bandwidth.

The client datasocket panel (for LabVIEW users) and client datasocket diagram are shown in Figures 6 and 7. This system is also completely reconfigurable as long as the interface structure is not altered.

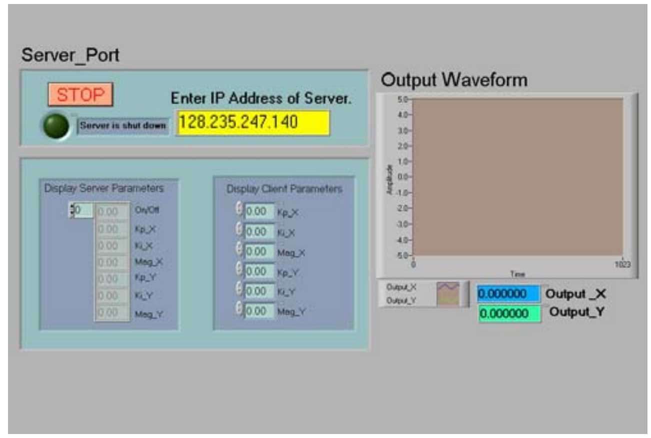

Figure 4: Server Socket Panel

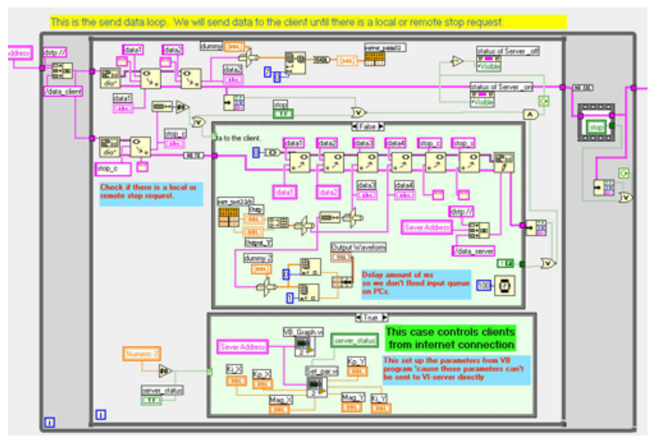

Figure 5: Server Socket Block Diagram

Finally, for the non LabVIEW users, the web browser based client is shown in Figure 8. This interface is fixed and cannot be easily modified. This restriction is actually advantageous for undergraduate applications due to the large number of student-access and the high potential for "hacking".

Among the experiments performed on the this system, the resolution test was the most successful. In this test, fine motion of the actuator under closed loop control is assessed by a series of three linearity tests where the actuator steady state outputs are measured under different excitation ranges and step sizes. The ranges of excitation are $\pm$ 0.25 μm, $\pm$ 0.1 μm, and $\pm$ 0.025 μm for the three tests. In each test, the actuator is commanded to move between the maximum and minimum limits of the range with a predefined step size (0.025, 0.01, and 0.0025 μm for the three tests). The data are averaged to yield the DC bias. Typical test data corresponding to the 0.0025 μm step size are plotted in Figure 9 where the left and right plots represent the response of the x and y axes respectively. It is observed that the controlled output is highly linear, with no trace of hysteresis. In Figure 9, slight distortion of the scale factor is noticed. This distortion is caused by ambient noise and sensor resolution.

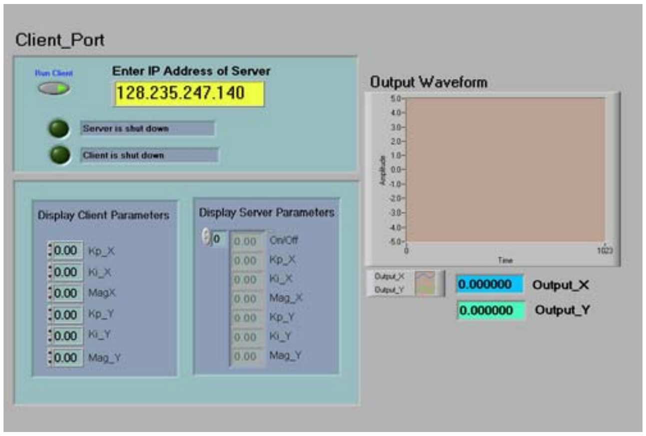

Figure 6: Client Socket Panel

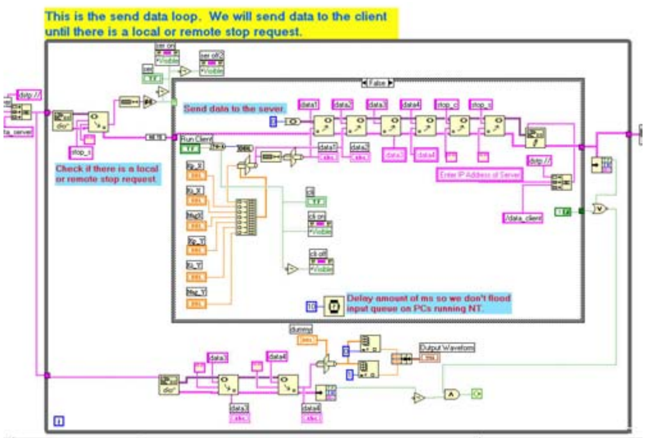

Figure 7: Client Socket Block Diagram

6 Conclusions

In this paper, web based control of a two degree-of-freedom nanopositioner is described. The device has a monolithic construction – both the x and y axes reside on the same PZT ceramic so that a number of advantages can be achieved: high bandwidth, resolution, load capacity while minimizing production cost. However, this experiment is delicate and the high voltage drives pose a potential safety problem. For these reason, undergraduate access has been excluded in the past. The use of web based control has circumvented these drawbacks and enabled the experiment to be run securely, safely, and on a 24/7 basis. The key enabling technology is the LabVIEW/Datasocket package which provides the standardized interface for live data streaming and multi-user access. Furthermore, along with ComponentWorks, a web browser plugin has been generated, allowing non-LabVIEW users to run the nanopositioner experiment remotely. Test results indicate that sufficient control and data exchange bandwidth can be achieved for multi-user access. From the class survey, student response to the experiments had been highly favorable and therefore more experiments will be deployed for the 2001 Fall semester.

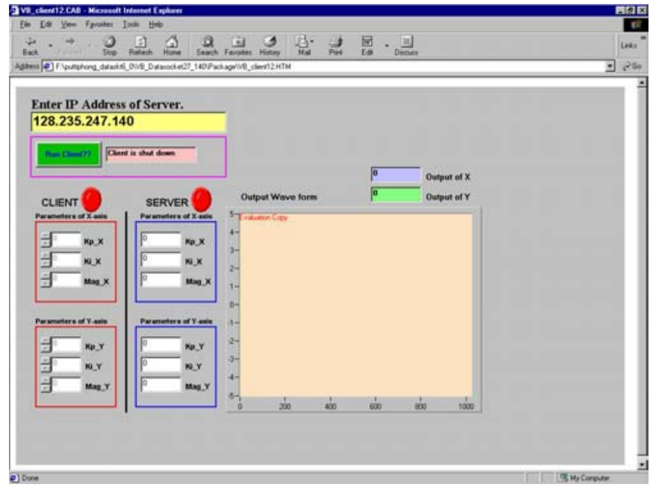

Figure 8: Client Web Panel

References

[1] T. Chang, *Piezoelectric multiple degree of freedom actuator*, U.S. Patent pending, 1998.

[2] T.N. Chang and X. Sun, "Analysis and Control of Monolithic Piezoelectric Nano-actuator, IEEE Transaction on Control Systems Technology, Special Issue on Smart Materials, January, 2001.

[3] F. Charette, A. Berry, and C. Guigou, "Dynamic effects of piezoelectric actuators on the vibrational response of a plate," *Journal of Intelligent Material Systems and Structures* , pp. 513–524, 1997.

[4] G. J. Knowles and J. Murray, "Foundations of piezoelectronic theory for distributed parameter systems," *Proceedings of SPIE - The International Society for Optical Engineering Smart Structures and Materials 1997: Mathematics and Control in Smart Structures* , pp. 324–334, 1997.

[5] M. Goldfarb and N. Celanovic, "Modeling piezoelectric stack actuators for control of micromanipulation," *IEEE Control Systems Magazine* , pp. 69–79, 1997.

[6] T. Alberts, T. DuBois, and H. Pota, "Experimental verification of transfer functions for a slewing piezoelectric laminate beam," *Control Engineering Practice* , pp. 163–170, 1995.

[7] S. Agrawal, D. Tong, and K. Nagaraja, "Modeling and shape control of piezoelectric actuator embedded elastic plates," *Journal of Intelligent Material Systems and Structures* , pp. 514–521, 1994.

[8] C. Won, D. Sparks, K. Belvin, and J. Sulla, "Application of piezoelectric devices to vibration suppression. from modeling and controller designs to implementation," *AIAA Guidance, Navigation and Control Conference* , p. 1381, 1992.

[9] H. Edwards, "Building an Interactive Web Page with DataSocket," Application Note 127, June1999 Edition, Part Number 341572B-01, National Instruments.

[10] H. Edwards, "Connecting Measurement Studio User Interface ActiveX Controls to Remote Data," Application Note 151, November 2000 Edition, Part Number 341972C-01, National Instruments.

[11] "Integration the Internet into Your Measurement System," DataSocket Technical Overview, March 1999 Edition, Part Number 341680A-01, National Instruments.

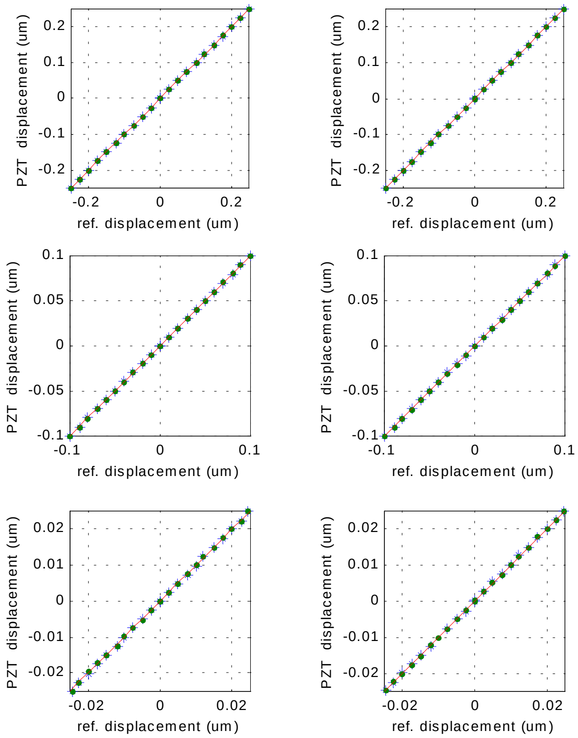

Figure 9: Resolution test with x-axis(left) and y-axis(right)

Acknowledgment

This work is supported, in part, by an NIST ATP Contract No. 70NANB5H1092 Precision Optoelectronics Assembly Consortium and New Jersey Information-Technology Opportunities for the Workforce, Education, and Research (NJ I-TOWER).