Stony Brook University

OFFICIAL COPY

The official electronic file of this thesis or dissertation is maintained by the University Libraries on behalf of The Graduate School at Stony Brook University.

© All Rights Reserved by Author.

Automation Enhancement of Multilayer Laue Lenses

A Thesis Presented

by

Kenneth Robert Lauer

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Master of Science

in

Computer Engineering

Stony Brook University

December 2010

Stony Brook University

The Graduate School

Kenneth Robert Lauer

We, the thesis committee for the above candidate for the

Master of Science degree, hereby recommend

acceptance of this thesis.

Yuanyuan Yang – Thesis Advisor
Graduate Program Director
Department of Electrical and Computer Engineering

John Murray
Associate Professor
Department of Electrical and Computer Engineering

Ray Conley
Optics Fabrication Group Leader
Photon Sciences Directorate, Brookhaven National Laboratory

This thesis is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Thesis

Automation Enhancement of Multilayer Laue Lenses

by

Kenneth Robert Lauer

Master of Science

in

Computer Engineering

Stony Brook University

2010

X-ray optics fabrication at Brookhaven National Laboratory has been facilitated by a new, state of the art magnetron sputtering physical deposition system. With its nine magnetron sputtering cathodes and substrate carrier that moves on a linear rail via a UHV brushless linear servo motor, the system is capable of accurately depositing the many thousands of layers necessary for multilayer Laue lenses. I have engineered a versatile and automated control program from scratch for the base system and many subsystems. Its main features include a custom scripting language, a fully customizable graphical user interface, wireless and remote control, and a terminal-based interface. This control system has already been successfully used in the creation of many types of x-ray optics, including several thousand layer multilayer Laue lenses.

Before reaching the point at which a deposition can be run, stencil-like masks for the sputtering cathodes must be created to ensure the proper distribution of sputtered atoms. Quality of multilayer Laue lenses can also be difficult to measure, given the size of the thin film layers. I employ my knowledge of software and algorithms to further ease these previously painstaking processes with custom programs. Additionally, I will give an overview of an x-ray optic simulator package I helped develop during the summer of 2010. In the interest of keeping my software free and open, I have worked mostly with the multiplatform Python and the PyQt application framework, utilizing C and C++ where necessary.

Table of Contents

List of Figures vi
List of Tables ix
Chapter 1 Introduction to Multilayers 1
Types of MLLs 1
Facilitating the Development of MLLs 2
References 3
Chapter 2 The Large Deposition System 4
Subsystems 6
Preparation for Deposition 7
Growth Rate Calculation 9
Deposition 12
Control System 13
Services 13
Information Server 14
Overview Clients 14
Plots 16
LabJack Extensibility 17
Scripting Language 19
Additional Control System Features 21
References 25
Chapter 3 Masks 26
Mask Basics 27
Linear Deposition Mask Calculation 29
Radial Deposition Mask Calculation 30
The MaskCalc Program 32
References 33
Chapter 4 Multi-purpose Sensor Boards 34
The Main Board 34
Temperature Sensors 36
Flow Sensors 38
PC Software 38

Chapter 5 Rotary Deposition System 39
The Original Control System 41
The Improved Control System 42
Chapter 6 Quality Analysis 43
Pre-processing 44
Image Processing 45
Actual Implementation 50
The Zone Plate Digitizer Program 51
Typical Usage 51
Plotted Results 52
References 53
Chapter 7 X-ray Optics Simulator 54
Software 54
Beamline Optics 56
Additional Tools 59
References 60
References 61

List of Figures

Figure 1-1. A flat multilayer Laue lens. ... 1
Figure 1-2. Different types of MLLs. ... 2
Figure 2-1. The main chamber of the large deposition system. ... 4
Figure 2-2. An open gun door (left) and the main electrical cabinet of the large deposition system (right). ... 5
Figure 2-3. Current subsystems of the large deposition system. ... 5
Figure 2-4. The Moxa NPort 5610 8-port Ethernet RS-232 serial device server. ... 6
Figure 2-5. Process for determining the growth rate characteristic of a set of magnetron sputtering cathodes. ... 7
Figure 2-6. A magnetron sputtering cathode without (left) and with (right) a target and cover installed. ... 8
Figure 2-7. A mask holder for the large deposition system. ... 9
Figure 2-8. Looking down the far end of the main chamber. ... 12
Figure 2-9. The orientation of the sample carrier (left) and the magnetron sputtering cathode (right). ... 12
Figure 2-10. The control system console (left) and process variable listing (right). ... 13
Figure 2-11. The main overview window. ... 15
Figure 2-12. The procedure for adding a process variable to an overview image. ... 16
Figure 2-13. A simple process variable plot. ... 16
Figure 2-14. The LabJack UE9 Pro data acquisition and measurement device. ... 17
Figure 2-15. An example input signal conversion. ... 18
Figure 2-16. Console-based LabJack service status screen, showing the device type, serial number, and the associated process variables. ... 19
Figure 2-17. The control system script window. ... 20
Figure 2-18. Pinned scripts in the control system (clipped from the full control system window). ... 21
Figure 2-19. Monitor mode example. ... 22
Figure 2-20. The wireless controller itself. ... 23
Figure 2-21. The wireless control program with configurable commands. ... 24
Figure 2-22. The control system log viewer. ... 25
Figure 3-1. A mask for a rotary deposition system at Advanced Photon Source, fastened on top of a cylinder situated around a magnetron gun. ... 26
Figure 3-2. Normalized distribution of a magnetron sputtering source by formula. The central red peak indicates that the deposition concentration is in the middle. ... 27
Figure 3-4. Laterally graded profile (left) and its corresponding mask for a linear deposition system (right). ... 28

Figure 3-3. Model (left) and uniform thickness profile mask silhouette (right). 28
Figure 3-5. A top view of radial deposition system sample carrier, showing the relative path in red of a single point on a stationary magnetron gun as the carrier rotates. 31
Figure 3-6. A symmetrical set of sweeping curves. 31
Figure 3-7. The MaskCalc application. 32
Figure 3-8. The MaskCalc expression editor. 32
Figure 3-9. A successful fit of a simple uniform profile, utilizing the original magnetron modeling formula.. 33
Figure 4-1. The main board schematic, with annotations for the temperature sensor RJ45 connector.. 35
Figure 4-2. The main board layout. 35
Figure 4-3. The fabricated main board (left) and an assembled main board (right). 36
Figure 4-4. The temperature sensor board schematic (left) and single-sided layout (right), showing two RJ45 connectors for daisy chaining support. 36
Figure 4-5. The temperature sensor board in use, mounted on a metal coupling of a water supply line. 37
Figure 4-6. The command-line based sensor service status screen. 38
Figure 5-1. The “Myron” rotary deposition system (left) and the inside of its vacuum chamber (right). 39
Figure 5-2. The rotary deposition system hardware. 40
Figure 5-3. A radial deposition system with two magnetron guns 180° apart, indicating the rotation of the sample carrier in relation to the sputtering sources. 40
Figure 5-4. The improved control system for the rotary deposition system. 41
Figure 6-1. A clean electron micrograph of an MLL section; WSi_2 is light and Si is dark.. 44
Figure 6-2. 20x20 pixel cross aperture. The POI is colored red in the center. 45
Figure 6-3. A small processed ROI. 46
Figure 6-4. Poor region selection and improper rotation lead to different layer counts. 46
Figure 6-5. Image digitization algorithm. (continued on the following page)................. 47
Figure 6-6. Layer width calculation algorithm. 49
Figure 6-7. The Zone Plate Digitizer Application. 51
Figure 6-8. Isolating a single layer to find the angle of rotation (θ). 51
Figure 6-9. A good result from the algorithm. 52
Figure 6-10. Individual electron micrographs stitched together into a graph of inverse d-spacing, with contributions from each image in different colors. 53
Figure 7-1. The source and its properties shown in the simulator. 55
Figure 7-2. A plot of the intensity, phase, and angular spectrum of a wave field some distance from a simple source component. 56

Figure 7-3. The source component (left) and single slit component (right) 57
Figure 7-4. The double slit component (left) and the Fresnel zone plate component (right). 57
Figure 7-5. The double crystal monochromator component (left) and the multilayer component (right). 57
Figure 7-6. The multilayer designer. 58
Figure 7-7. A reflectivity versus angle plot of a 2000 layer multilayer. 58
Figure 7-8. A sigma displacement versus position plot for a 2000 layer multilayer. 59
Figure 7-9. The Bragg diffraction utility. 60

List of Tables

Table 2-1. Passes over specific targets for determining growth rates. 10
Table 2-2. A sample information server session, where commands are in bold. 14

Acknowledgements

I owe my gratitude to the members of the Optics Fabrication group of Brookhaven National Laboratory; for without their help and research, this thesis would not have been realized. To my parents and my sister, I give my love and thanks for their continual support in everything I do. I would also like to recognize Nathalie Bouet and Ray Conley for contributing and allowing free usage of the MLL and SEM images.

Chapter 1

Introduction to Multilayers

X-ray multilayers come in essentially two types: reflective multilayers and transmission multilayers. The reflective version has seen world-wide use in laboratory x-ray sources, synchrotron facilities, space telescopes, and satellites. Transmission multilayers are only recently coming into use as x-ray gratings, for electron microscope characterization, and as focusing optics called "multilayer Laue lenses" (hereafter "MLLs").

In order to focus x-rays achromatically, a zone plate is typically employed. A zone plate is fabricated in one of several slightly different recipes where a photoresist is patterned using e-beam lithography to write concentric rings with widths as small as 15nm. The photoresist is then developed, and a metal (typically gold) is electroplated in the space left behind.

Since the two most important figures of merit for a zone plate are what the smallest line width is and how deep the electroplated metal is, the aspect ratio is at most 1:100. MLLs overcome both of these basic limitations as sputtered layers can be accurately fabricated to be less than 1 nm thick. By sectioning an as-grown sample, the aspect ratio is then arbitrary (1:30,000 for example).

Figure 1-1. A flat multilayer Laue lens.

Types of MLLs

The first MLLs were fabricated at the Advanced Photon Source, and this work has since transferred to NSLS-II under the lead of Ray Conley. The most basic MLL is a tilted half-structure, where only one half of an MLL is grown, and is then tilted in the x-ray beam so that the outermost layers are at the Bragg angle and thus provide a higher efficiency. A flat full-structure may also be realized, providing a higher numerical aperture. This means that more of the optic is grown, so there is a larger area of the

beam to grab and focus – but the downside is that none of the layers are at the Bragg angle. A later variant on these two types is the wedged MLL, where the layers are grown with both a depth gradient and a lateral thickness gradient so that each layer is at the correct Bragg angle. This produces, in theory, the smallest focus and the best efficiency but is a chromatic optic and is more complicated to grow.

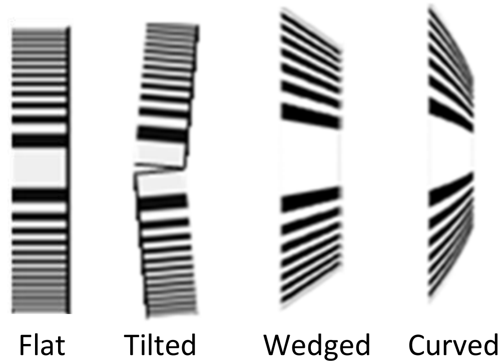


Figure 1-2. Different types of MLLs.

Facilitating the Development of MLLs

In the following chapters, I will detail the work I have undertaken as a Stony Brook University Research Assistant, sponsored by the Optics Fabrication Group at Brookhaven National Laboratory. I have worked to make the planning, simulation, deposition, and quality evaluation of multilayers as easy as possible for the scientists and technicians alike.

Most of the effort has revolved around the large deposition system of Chapter 2, which was designed with the explicit purpose of creating many thousand layered multilayers. The predecessor to this system, described in Chapter 5, helped clarify the needs of the large deposition system's new control system. Other projects, such as the mask generator of Chapter 3 and the multi-purpose sensor boards of Chapter 4, have stemmed from the needs of both of these deposition systems. Chapter 6 describes a method of determining the quality of MLLs. And finally, Chapter 7 outlines a related project that can help theoretical physicists run virtual beamline experiments.

References

"Multilayer Laue Lens Growth at NSLS-II", R. Conley, N. Bouet, K. Lauer, M. Carlucci-Dayton, J. Biancarosa, L. Boas, J. Drannbauer, J. Feraca, L. Rosenbaum, Submitted to XRM-2010 proceedings (2010)

"Nanofocusing of Hard X-rays with Multilayer Laue Lenses", A. Macrander, H. Yan, H.C. Kang, J. Maser, C. Liu, R. Conley, G.B. Stephenson, Handbook of Optics, 3rd Edition, Optical Society of America , Volume V, Chapter 42, (2010)

"Nanometer linear focusing of hard x-rays by a multilayer Laue lens", H.C. Kang, J. Maser, G.B. Stephenson, C. Liu, R. Conley, A.T. Macrander, S. Vogt, Phys. Rev. Lett. 96, pp. 127401, (2006).

"Small d-spacing WSi2/Si narrow bandpass multilayers", C. Liu, R. Conley, A.T. Macrander, T. Graber, C. Morawe, C. Borel, E.M. Dufresne, Proc. SPIE 5537, pp. 154-160, (2004).

Chapter 2

The Large Deposition System

The Optics Fabrication Group at Brookhaven National Laboratory has the goal of reliably fabricating many thousand layer MLL structures. Since such a feat had not been achieved with currently available deposition systems, the large deposition system was designed. The simple naming scheme is merely a size comparison with the small rotary deposition system of Chapter 5.

The system is based around a 7 meter long chamber, as shown in Figure 2-1, which is capable of being pumped down to high vacuum utilizing roughing vacuum pumps and cryopumps. Two 1.5m doors (Figure 2-2, left) house four magnetron sputtering cathodes each, and a ninth cathode is located at the end of the chamber.

A sample carrier moves one-dimensionally along the length of the chamber by means of a UHV brushless linear servo motor. The load-lock chamber, pictured on the right side of Figure 2-1, is separated from the main chamber by a gate valve, such that the main chamber can remain at vacuum while the samples are being set up for or removed after a deposition.

There are many electronic subsystems, some hosted from different computers, that are fully integrated into the deposition process. This chapter will touch briefly on these subsystems and detail the control system of the machine.

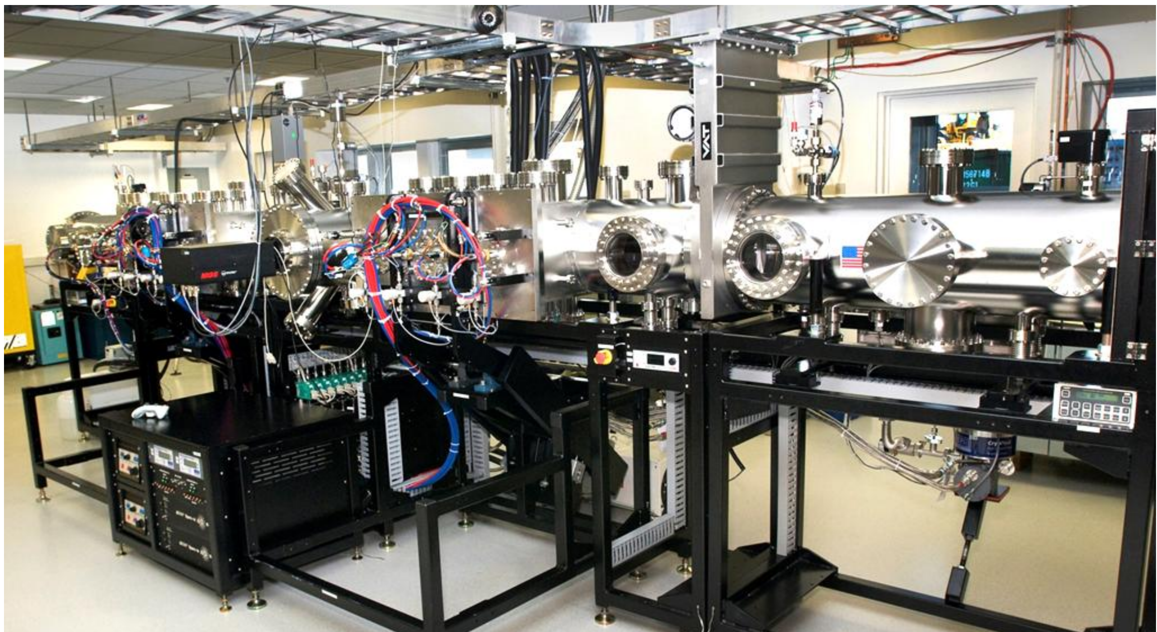


Figure 2-1. The main chamber of the large deposition system.

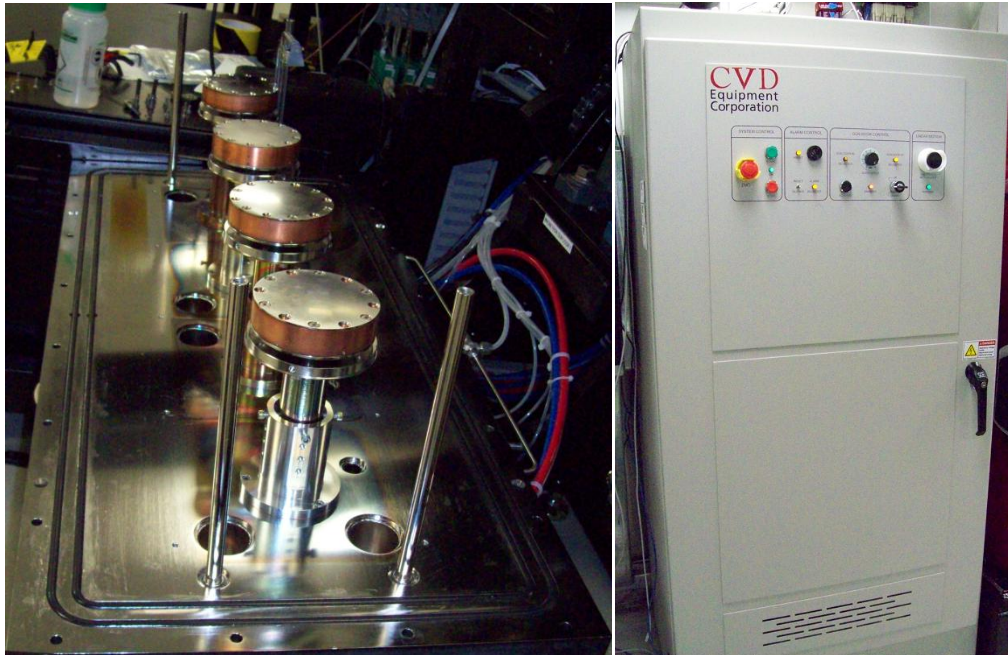


Figure 2-2. An open gun door (left) and the main electrical cabinet of the large deposition system (right).

---

1. Koyo DirectLOGIC programmable logic controller (PLC)
2. Vacuum pumps (roughing and cryopumps)
3. Galil DMC-4020 linear motion controller
4. Gate valves
5. kSA Multi-beam Optical Sensor (MOS)
6. SCR (Silicon Controlled Rectifier) lamps
7. Temperature sensors
8. Water flow sensors
9. LabJack data acquisition devices
10. UHV stepper motor
11. Quartz Crystal Rate Monitor (QCM)
12. ION 1500 DC sputtering power supply
13. Two MDX 1000 sputtering power supplies
14. Moxa NPort 5610 8-port Ethernet RS-232 serial device server
15. Moxa NPort 5650 16-port Ethernet RS-232/422/485 serial device server

---

Figure 2-3. Current subsystems of the large deposition system.

Subsystems

A Koyo DirectLOGIC programmable logic controller (or PLC) handles the basic safety interlocks for the whole system. For example, it protects against opening the doors while the system is at vacuum. These interlocks are defined in a rule-based language called ladder logic in which each rule is defined on a different ladder rung.

The PLC communicates with several other hardware subsystems, including one of the gate valves, the gun door motors, and various other sensors in the system. This basic functionality was provided by the manufacturer of the system and has been retained in the custom control system. With the addition of new hardware, excepting where safety interlocks were necessary, interfacing directly with the new control system has become the method of choice. Time and hardware constraints were two main reasons for this.

Several of the subsystems are connected through the Moxa Ethernet serial device servers. These devices allow for the system to be extended with many RS-232 and RS-485 serial devices, which can then be easily communicated with over the local network. The Quartz Crystal Rate Monitor (QCM), which measures the rate of the deposition by the frequency shift of a crystal oscillator near one of the sputtering cathodes, is connected in this fashion. Though the two original MDX 1000 sputtering power supplies are interfaced through the PLC, the ION 1500 DC sputtering power supply, which is a relatively recent addition to the system, supports RS-232 communication and as such is connected through a Moxa device server. More complicated subsystems of note will be further described in the Control System section of this chapter.

Figure 2-4. The Moxa NPort 5610 8-port Ethernet RS-232 serial device server.

1. Mount the desired target material on the magnetron sputtering cathodes.
2. Prepare and install the mask holder.
3. Seal and pump down the main chamber.
4. Load the sample carrier with two substrates per set of cathodes.
5. Load the sample carrier into the system via the load-lock chamber.
6. Initialize the linear motor by executing a homing script.
7. At the proper base pressure, throttle down the vacuum pumps and inject a controlled amount of process gas.
8. Ready new targets for deposition through a burn-in process.
9. Grow two 20 bilayer multilayers with different material ratios.
10. Remove the sample carrier via the load-lock chamber.
11. Utilize X-ray diffraction to calculate the d-spacing.

Figure 2-5. Process for determining the growth rate characteristic of a set of magnetron sputtering cathodes.

Preparation for Deposition

Before being able to create any multilayer – MLL or otherwise – it is necessary to determine the characteristics of the magnetron sputtering cathodes, or magnetron guns. The next section of this chapter will describe that process with the large deposition system.

Before the chamber can be sealed and pumped down, the magnetron sputtering cathodes must have targets installed. For MLLs, this usually means one cathode is WSi_2 and the other is pure silicon. Given that this will be in a vacuum system, care must be taken to not introduce contaminants such as oil from skin. A thermal compound is spread onto the cathode evenly, and a target is situated upon it, and finally a cover is affixed to hold the target in place.

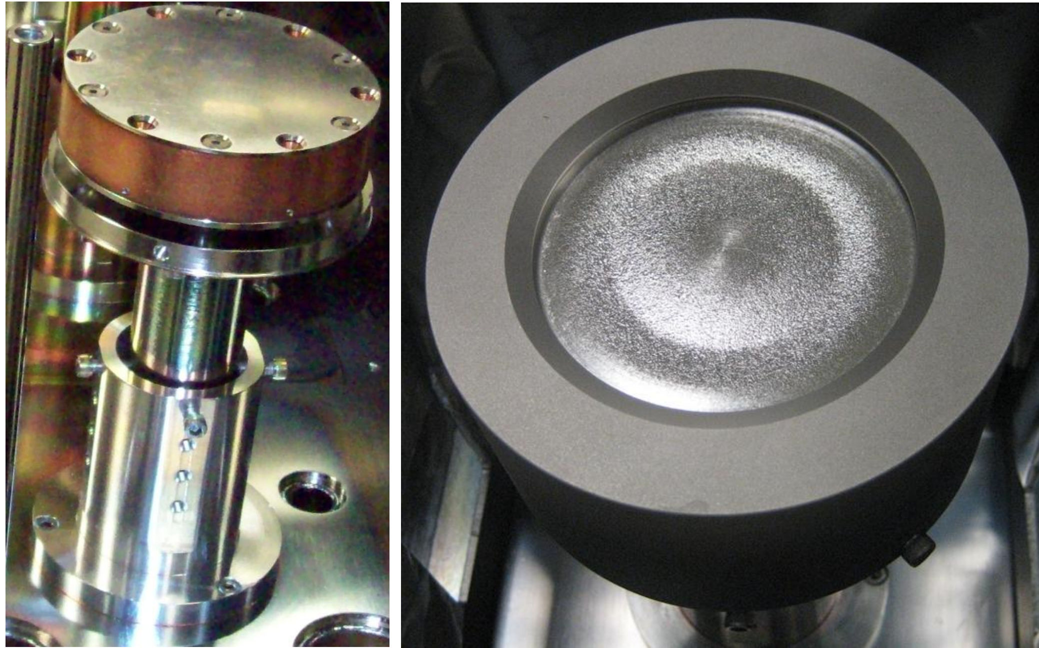


Figure 2-6. A magnetron sputtering cathode without (left) and with (right) a target and cover installed.

Mask holders (pictured in Figure 2-7), cleaned and prepared with new aluminum foil, can then be fastened around each set of four magnetron guns. The aluminum foil is very important, as it grabs the stray flux without flaking. Should a flake fall into the plasma, the flake perturbs the growth and the entire process must be restarted from the beginning. The foil also serves an additional purpose by keeping the inside of the vacuum chamber moderately clean from one deposition to the next by stopping stray atoms or molecules of the target material before they hit elsewhere in the chamber.

The main chamber can then be sealed and then be evacuated with a screw-type roughing pump. After pumping down to a medium vacuum in the 10^{-3} Torr range, 4 cryopumps each evacuating the chamber at 3000 L/s and operating at 10°K can then further pump the system down to a high vacuum of around 10^{-7} to 10^{-8} Torr.

The chamber is then ready for the process gas to be injected. Utilizing mass-flow controllers (MFCs), a controlled amount of gas is introduced to the system. The process gas is one of the noble gases and is most often argon, but sometimes helium or neon are used. The gas pressure is maintained stably below the atom clustering regime. A predefined amount of gas (typically 4-8 SCCM per cathode) is injected into each cathode directly through the dark space shield. For pressure control, one MFC operates in feedback mode with the pressure. Once the pressure has stabilized after roughly a few minutes to the desired pressure (usually 1 to 5 mTorr), the conduction cathodes are then burnt in for an hour each at around 170 to 250 W. In order to not dirty the

chamber before a long deposition, a substrate-free area of the sample carrier is positioned in front of the cathode being burnt in.

While the main chamber is being pumped down, the substrates can be prepared. This is made possible by the load-lock chamber, as the sample carrier can be brought in an out of the machine even if the main chamber is under high vacuum.

Figure 2-7. A mask holder for the large deposition system.

Growth Rate Calculation

With the linear motion of the sample carrier, the deposited film thickness depends on the number of passes it makes past the stationary sputtering source and the speed of those passes. In order to determine the growth rate of a pair of cathodes, two substrates are used to grow separate 20 bilayer multilayers.

If the material system used has chemical mixing at the interface, as in the case of Mo and Si, or W and Si, the growth versus time is not exactly linear since the first few angstroms of each material are consumed into a silicide material at the interface and this must be accounted for. For materials with little or no chemical mixing at the interface, such as the frequently used combination of WSi_2 and Si, the calculations are greatly simplified.

	Multilayer 1	Multilayer 2
WSi_2 target	3	6
Si target	5	5

Table 2-1. Passes over specific targets for determining growth rates.

Take, for example, the two multilayers grown according to Table 2-1. Both samples share the same number of passes at the same speed over the silicon target. The passes over the WSi_2 target from one are doubled from the first to the second sample. This means that the thickness of the WSi_2 layers in the second multilayer is also double that of the first multilayer.

Utilizing x-ray reflectivity measurements and viewing reflectivity versus angle plots, the angle at which the reflectivity is at its peak for each multilayer indicates the d-spacing. If, for this example, those measurements yield d-spacings of 110 Å and 164 Å, respectively, this relationship can then be expressed as

$$3\mathrm{A}_{20} + 5\mathrm{B}_{20} = \mathrm{M}_1 = 110\ \text{Å} \quad \text{(I)}$$
$$6\mathrm{A}_{20} + 5\mathrm{B}_{20} = \mathrm{M}_2 = 164\ \text{Å} \quad \text{(II)}$$

where A_i is the per-pass growth rate for *i* layers of WSi_2 and B_i is that of Si. Clearly the additional 54 Å come from the additional passes over the WSi_2 target – that is,

$$\mathrm{A}_{20} = \frac{54}{3} = 18\ \text{Å}$$

and $5\mathrm{B}_{20}$ can then be calculated from 2*(I) – (II), or

$$5\mathrm{B}_{20} = 2(110) - 164 = 56\ \text{Å}\ .$$

Thus we have the single-layer per-pass growth rates

$$A_1 = \frac{A_{20}}{20} = 0.9\ \text{Å}$$

and

$$B_1 = \frac{5B_{20}}{5(20)} = 0.56\ \text{Å}\ .$$

Generically, for a *k* bilayer structure, with *x* passes over the first material and *y* passes over the second, and doubling the passes over the first material for the second bilayer,

$$\begin{pmatrix} x & y \\ 2x & y \end{pmatrix} \begin{bmatrix} A_k \\ B_k \end{bmatrix} = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}.$$

Solving for $\begin{bmatrix} A_k \\ B_k \end{bmatrix}$ by taking the inverse of $\begin{pmatrix} x & y \\ 2x & y \end{pmatrix}$, we have

$$\begin{bmatrix} A_k \\ B_k \end{bmatrix} = \begin{bmatrix} \frac{1}{x}(M_2 - M_1) \\ \frac{1}{y}(2M_1 - M_2) \end{bmatrix}$$

and finally

$$\begin{bmatrix} A_1 \\ B_1 \end{bmatrix} = \begin{bmatrix} \frac{A_k}{k} \\ \frac{B_k}{k} \end{bmatrix} = \begin{bmatrix} \frac{1}{kx}(M_2 - M_1) \\ \frac{1}{ky}(2M_1 - M_2) \end{bmatrix}.$$

Deposition

A well-defined formula exists for growth rate decay, or erosion of the targets, which also affects the accuracy of the deposited film thicknesses. With drift compensation calculations and the originally calculated growth rate from the previous section, it is now possible to accurately deposit films of a specific thickness.

For a multilayer Laue lens, the individual layer thicknesses are then determined by the Fresnel zone plate formula. A script for the control system is then generated to control the whole deposition process. The following sections will further describe the control system.

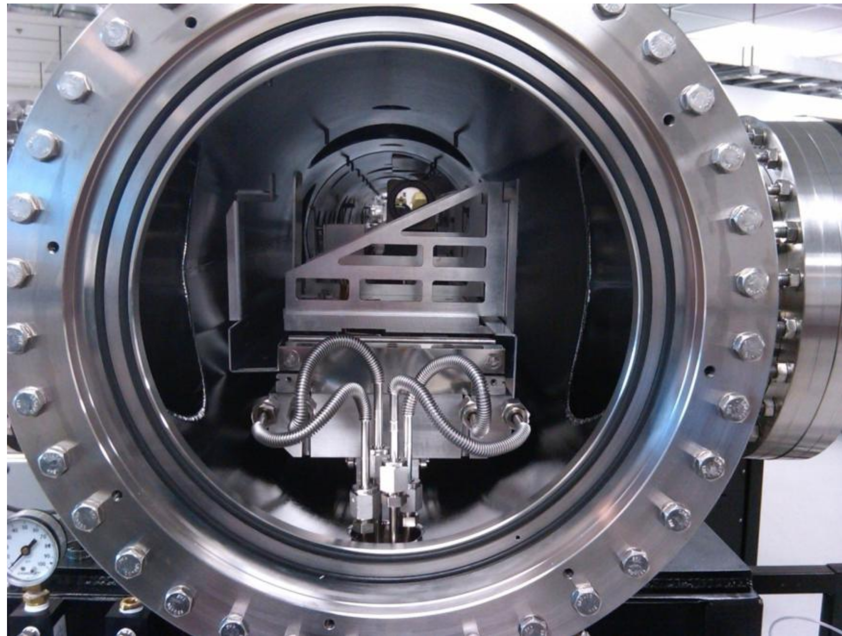


Figure 2-8. Looking down the far end of the main chamber.

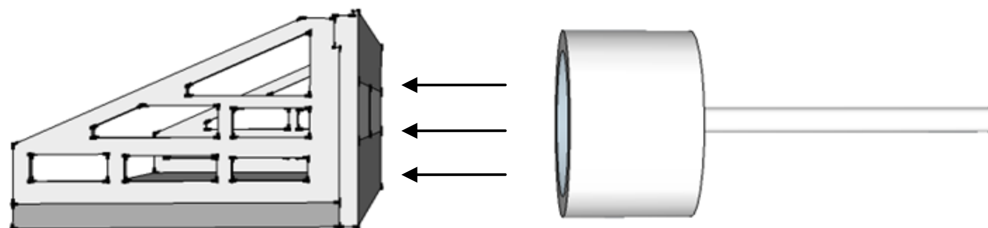


Figure 2-9. The orientation of the sample carrier (left) and the magnetron sputtering cathode (right).

Control System

The control system was designed in platform independent Python[1]. The choice of Python brought with it a wealth of freely available modules for many purposes, including multiplatform windowing systems, scientific applications, and networking. The native GUI widgets were made available by Riverbank Computing's PyQt[2], which is a Python binding to Nokia's Qt application framework. Plots were initially made possible with Matplotlib[3], but a move was made to Enthought's Chaco[4] later in the project. Interfacing with all remote subsystems through the local network was made possible by the comprehensive Twisted[5] networking engine.

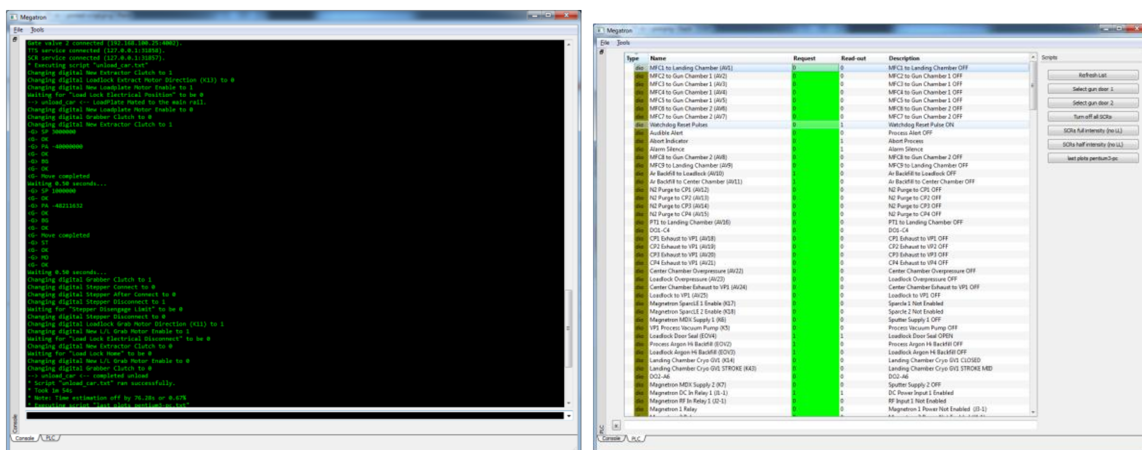


Figure 2-10. The control system console (left) and process variable listing (right).

Services

Since nearly every subsystem is controlled in an application-specific manner, there was a necessity to separate the functionality of each of those into individual "services". Services can be controlled through the main program, though their implementation may be in a separate process or on another computer. Each of these services has its own process variable (PV) that can be easily toggled to disable or restart the service or process.

Communication between the main control system and the services is done through a simple asynchronous message passing TCP/IP-based protocol. Services can access process variables from the main control program through its information server, described in the next section.

---

[1] Python 2.6.x and 2.7.x; http://www.python.org/
[2] PyQt4 v4.8.x; http://www.riverbankcomputing.co.uk/software/pyqt/intro
[3] Matplotlib 1.0.0; http://matplotlib.sourceforge.net/
[4] Enthought Chaco 3.3.x; http://code.enthought.com/projects/chaco/
[5] Twisted 10.1.0; http://twistedmatrix.com/

Information Server

An information server runs on the main control system to allow for separate programs to get access to the status of the whole system. This includes process variables, script information, and so on. There is also the possibility to change system parameters remotely with simple password-based authentication. The network which the control system runs on is private; so the risk of attacks with this basic type of authentication is low.

A non-authenticated connection to the information server is allowed read-only access to the parameters of the system. An example of a program that would not require authentication is a remote plotting application. A simple, human-readable version of a message-passing session is shown in Table 2-2. An authenticated connection might be from an overview client, which is described in the next section.

Client	Server
(Attempts to connect)	(Accepts connection)
SetPVs "Argon 1 (MFC1)", ...	**OK** (Stores per connection list of PVs)
Poll	**Status** "Argon 1 (MFC1)", 1.0 SCCM, ...
Poll	**Status** "Argon 1 (MFC1)", 1.2 SCCM, ...
(Closes connection)	

Table 2-2. A sample information server session, where commands are in bold.

Overview Clients

At its core, the control system is based around the console window, scripting language, and the editable list of process variables. In addition to these are "overviews" , which are customizable, graphical user interface (GUI) clients. They can be run on the same machine as the control system or remotely, assuming network connectivity exists.

Selection from a list of background images determines which part of the system the user desires to look at. For example, the main chamber and general overview of the whole system are shown in Figure 2-11. The right-hand portion of the window shows the status of various subsystems and process variables.

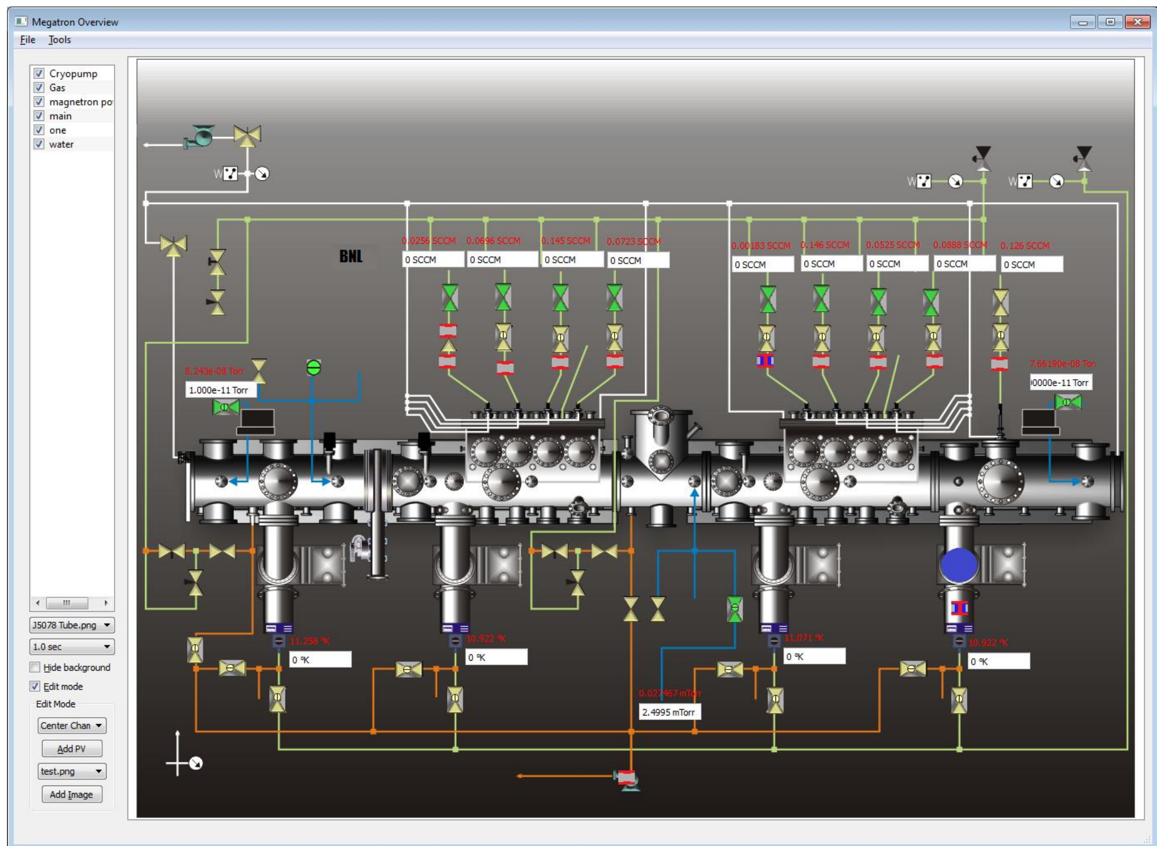


Figure 2-11. The main overview window.

In an overview, there is both a view mode and an edit mode. While in the view mode, the system can be monitored, but the configuration of the layout cannot be changed. While in edit mode, push buttons and text box user interface widgets can be placed and customized by selecting the associated process variable and pressing "Add PV". The full procedure is listed in Figure 2-12. Push buttons which execute commands are also an option. The widgets can be moved around by clicking and dragging, and further customized by right clicking and setting additional parameters. Through the pop-up menu, parameters such as text color, button style and images, and significant digits can be easily configured.

1. Select the background image which contains the related hardware.
2. Change to edit mode.
3. Select the process variable to be added, and click “Add PV”.
4. Drag the newly added widget and place it on the overview.
5. Right click on the widget and set the properties.
    a. For digital process variables, change the button style to reflect the type of control the user has over it.
    b. For analog process variables, change the number of significant digits to display to a reasonable amount, and adjust the text color if necessary.
    c. Change the associated script to be executed upon clicking, if necessary.
6. Return to the view mode by deselecting edit mode.
7. If possible, test the widget to ensure its functionality is as desired.

Figure 2-12. The procedure for adding a process variable to an overview image.

Each of these widgets is contained in a group. The visibility of the groups can be toggled so that only certain aspects of the system are visible at a time, which can help when one only wants to see the water-related process variables, for example. Since the button images can also contain a transparent color, the z-order of the widgets also becomes important. The z-order of individual widgets and also whole groups can be modified by the same pop-up menu that allows configuration of other parameters.

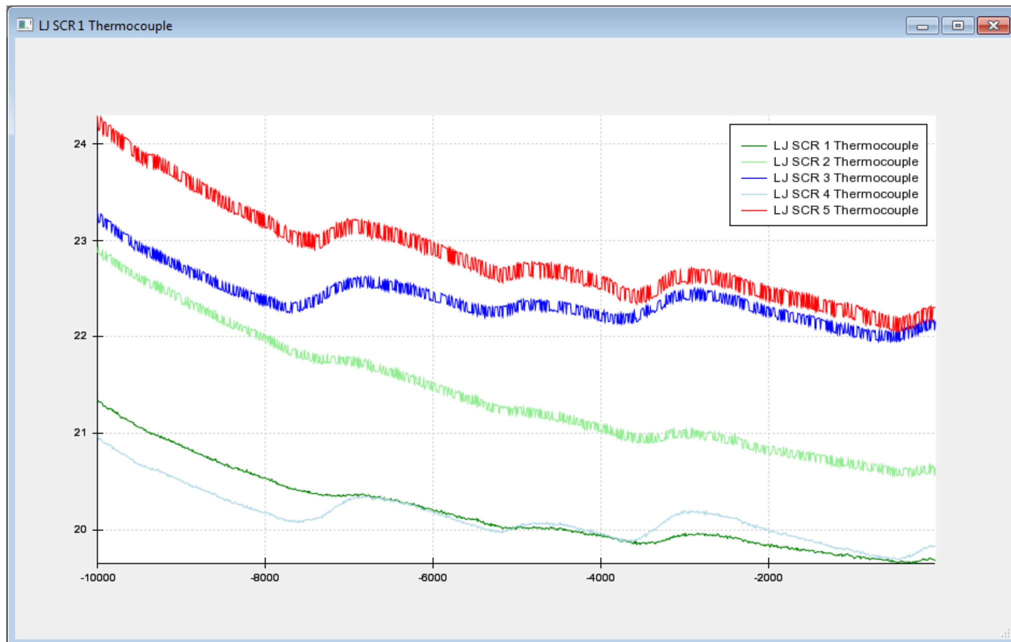


Figure 2-13. A simple process variable plot.

Plots

Similar to the overview clients of the previous section, plots can also be shown either locally or remotely for real-time viewing of process variable status. Single or multiple process variables can be viewed on the same plot. The plots can be panned and zoomed, and the rate at which the plot is updated can be configured as the user desires. Logarithmic scaling, which is standard for viewing vacuum pressures, can also be toggled.

LabJack Extensibility

As there is the potential for additional devices to be added to the system in the future, a simple option to extend upon the functionality of the control system was added. The low-cost LabJack data acquisition devices were chosen for this purpose. Any feasible number of these devices can be added to the system, and additional process variables can be mapped to each pin with minimal effort.

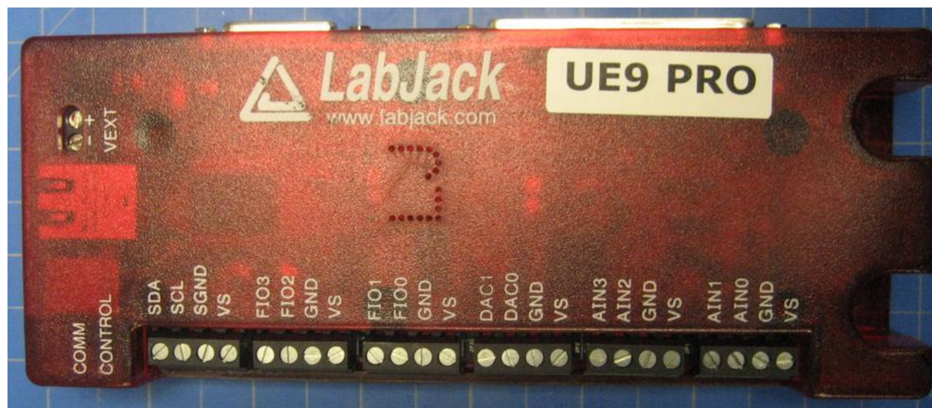


Figure 2-14. The LabJack UE9 Pro data acquisition and measurement device.

The procedure for adding an additional process variable is straightforward. First, the user must identify the requirements of the device they intend to add. If the device is a simple sensor, it usually is a matter of finding an available pin with digital or analog input capabilities. For a device that can be toggled or one that has a setpoint, a digital or analog out should suffice. After a process variable name is decided to describe the functionality appropriately, the unique serial number of the device and the pin name itself are then all that is necessary to uniquely describe the pin. Finally, adding this information to the appropriate process variable spreadsheet allows it to be accessible by the control system.

An additional option for any analog input on the LabJack service allows for proper reading of K-type thermocouples. Temperature measurements can be done in

vacuum with these thermocouples. The data acquisition hardware is then situated outside of the chamber and is connected to the thermocouple via a feedthrough. Proportional-integral-derivative (PID) loops with parameters from the main process variable list can also run independently on the LabJack service.

In the case of a general analog input or output, it is often more convenient to deal with the appropriate units and not just voltages. To this end, there is the option to automatically convert a voltage to a value with the proper units for an analog input or vice versa with an analog output. Though at times it is necessary to use a lookup table, this conversion can often be done as a simple linear function.

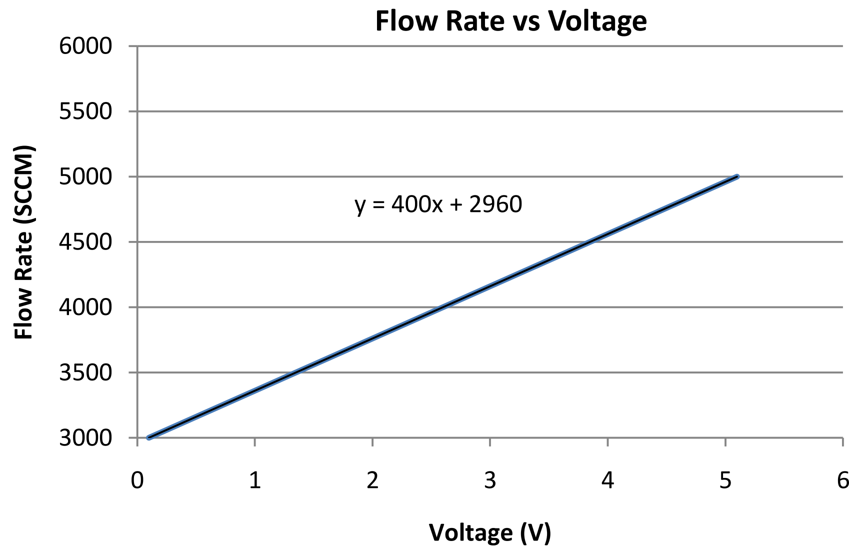


Figure 2-15. An example input signal conversion.

Take, for example, that there is a flow rate analog input (with respect to the control system) from a mass-flow controller. A data sheet indicates that an input signal of 0.1 V to 5.1 V represents from 3000 SCCM to 5000 SCCM. For such a simple example it is easy to calculate the relationship based on the point-slope form of a straight line. Microsoft Excel is also capable of doing the work for us with a trend line, as shown in Figure 2-15. This formula can be almost directly input into the configuration, changing *x* to *volts* (i.e., 400 * volts + 2960) and the LabJack service would evaluate this expression as new voltages were read. The evaluation is not limited to the simple linear cases – a quadratic or cubic function could just as easily be utilized.

```
Labjack u6 (360005813)
LJ SCR 1 Thermocouple 22.1187439403   C
LJ SCR 2 Thermocouple 23.6235858808   C
LJ SCR 3 Thermocouple 23.2503195223   C
LJ SCR 4 Thermocouple 23.6679411417   C
LJ SCR 5 Thermocouple 24.6760055538   C

Labjack u6 (360005430)
LJ Argon flow signal (MFC) 0.2763300575 SCCM
LJ Argon pressure 17.167142343 PSIG
LJ Argon purge 0 write
LJ Argon setpoint (MFC) 0.0 SCCM=0.00 V write
LJ Argon valve 0 write
LJ Baratron 2 (50mTorr) 50.5478731468 mTorr
LJ Capacitance monometer (100 Torr) 130.64253507 Torr
LJ FIO4 0 write
LJ Neon flow signal (MFC) 0.3900201383 SCCM
LJ Neon purge 0 write
LJ Neon setpoint (MFC) 0 SCCM=0.00 V write
LJ Neon valve 0 write
LJ Nitrogen pressure 1646.10914587 PSIG
```

Figure 2-16. Console-based LabJack service status screen, showing the device type, serial number, and the associated process variables.

Scripting Language

Initially, the control system allowed script or plug-in support through additional Python modules. Though this type of script offers the maximum amount of flexibility, it was too verbose and complicated for use by technicians or scientists without programming background. This led to the development of a simple, synchronous scripting language that is based around the available process variables (PVs) in the system. Commands, which are accessible by the scripts and also by the console, follow the format

```
command_name [“argument1”[, “argument2”[, “argument3” …]]]
```

where the quotation marks are only required when an argument contains spaces.

These scripts are stored in simple text files, which are shown in the scripts window of Figure 2-17. When the program detects a new script or a modified script (based on the reported modification time stamp), it proceeds to parse and ensure that the command format is strictly followed. Further checking is done to guarantee that each individual line of code can be executed, including checking per-command parameters. For example, a simple command to wait a specified number of seconds requires a floating point or integral value. Should the parsed value turn out to be a string, the script would be flagged as having an error and would not be allowed to run.

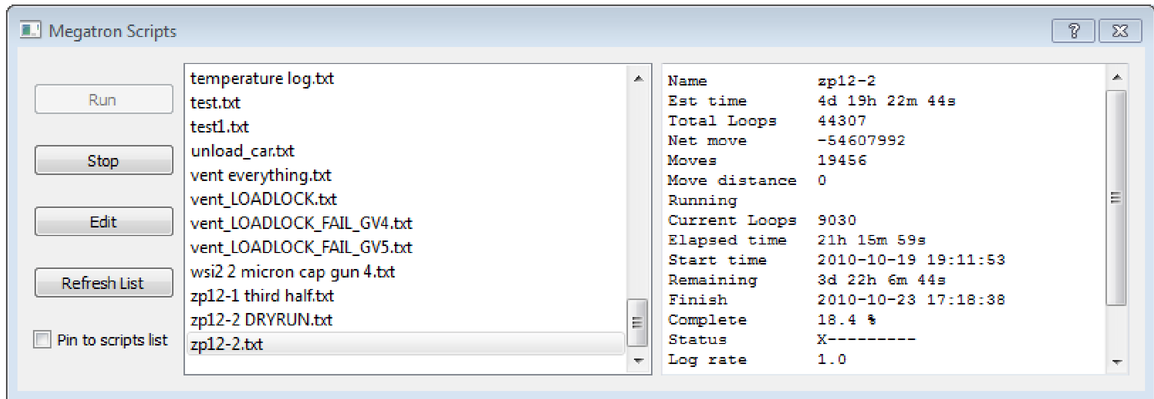


Figure 2-17. The control system script window.

During the command checking phase, a simple simulation is also occurring. The simulation tracks the motion of the linear stage, calculating how long those movements would take based on the current acceleration, deceleration, and velocity settings. Summing the individual movement times and factoring the time it takes to execute the other commands, an estimated total execution time is calculated. Should the script be selected to run, a remaining execution time and an estimated completion date and time are displayed.

These basic safeguards help protect against some careless mistakes that occur when writing scripts by hand. However, they do not protect against unforeseen events that can occur during the execution of a synchronous script. To this end, there are scripting language commands that can optionally halt the current script or trigger the execution of a recovery script.

Assuming there is no resource (i.e., subsystem) conflict, multiple scripts can be run simultaneously from the console, from the script window, and also from the pinned script list. They can also be paused and resumed as required. The pinned script list is mostly meant for frequently executed utility scripts. For example, the large deposition system is set up such that only one door to the main chamber can be raised or lowered at the same time. Which door is selected is controlled by the PLC, but the actual raising and lowering of the selected door is performed by physically pressing a button on the electrical cabinet of the system itself. Thus, a short door selection script was written to communicate with the PLC, and it was pinned to the script list. Instead of manually changing process variables, the change is now effected by a single mouse click.

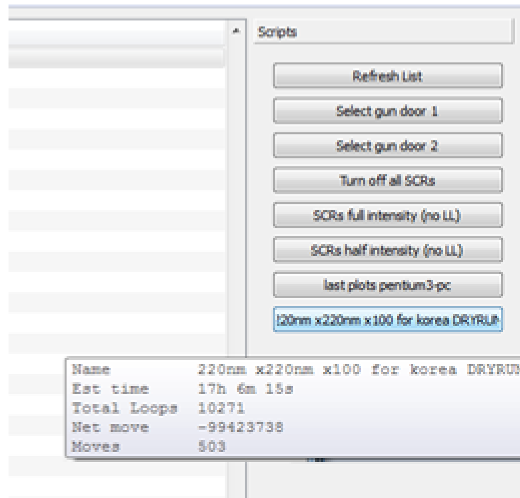


Figure 2-18. Pinned scripts in the control system (clipped from the full control system window).

The synchronous nature of the scripts needs to be retained for every subsystem. This means that a command should only be considered completed when the subsystem reports that it is complete. Take, for example, the Galil motion controller. Every command that is detected to be related to the Galil motion controller is automatically passed through from the scripting language. The Galil service waits for a reply from the hardware, and only once it has been received does it return to the interpreter. In the case of a movement command, it only returns once it has been detected that the Galil is no longer moving.

Additional Control System Features

In addition to the major features which make up the core of the control system, there are some additional elements that make using and debugging issues with the deposition system significantly easier.

Monitoring and Text-to-Speech

In a system with hundreds of process variables, debugging certain procedures can prove troublesome. In the case of analog process variables, it is logical to simply add a plot to view their behavior. Digital input values might toggle on and off before they are even noticed. So, a monitoring mode was added to the system so that during debugging operations, changes to digital values would be reported directly.

Take, for example, a case of swapped limit sensor wires. Unsure of which digital input refers to which sensor, the two can be added to the system and the monitor mode

enabled. Physically toggling the state of each limit switch can then allow one to determine the proper configuration.

With the size of the deposition laboratory and the location of the control system computers, it is not always feasible to look and see which values are changing from the opposite end of the room. In addition to the monitor mode, there is also a text-to-speech (TTS) service that relays the same information audibly. Notifications of status and other errors can be automatically sent by e-mail from a scripting language command.

```
> monitor
Monitor mode enabled.

Digital in "Sensor 1" changed to 1
Digital in "Sensor 1" changed to 0

Digital in "Sensor 2" changed to 1
Digital in "Sensor 2" changed to 0

> monitor
Monitor mode disabled.
```

Figure 2-19. Monitor mode example.

Direct Device Communication

Although every effort was made to utilize the required functionality of all of the subsystems, there is always a possibility that additional control over the devices might be required in the future. Direct communication with supported devices was added to address this issue.

This means that if, for example, an additional parameter for the ION 1500 DC sputtering power supply needs to be specified during a deposition, it can be done through a simple command such as

```
> ION SET_VALUE 1.0
```

and the control system will take the command, interpret it as necessary, create a compatible packet and send it directly to the ION 1500 without interfering with its normal operation.

Wireless Control

Before the large deposition system could be used for creating accurate multilayers, its linear rail, on which the sample carrier is situated, had to first be leveled. This was a very tedious month-long process which required the use of an autocollimator, a device which sends a collimated beam of light and analyzes the reflected beam from a mirror to ascertain a very precise angular difference. In this particular application, the autocollimator was situated at the end of the deposition system and the mirror was on the sample carrier.

Height variations along even very small stretches could induce vertical movement of the samples. This necessitated moving the sample carrier at a slow speed and recording the autocollimator data. Initially, the only tool at our disposal for moving the sample carrier was through a command line interface by specifying acceleration, deceleration, speed, and relative distance. After manually fine-tuning certain sections of the rail, stopping and restarting the motion became even more tiresome.

This was the starting point for an additional utility, the wireless control program. It makes use of the ubiquitous Xbox 360 wireless console game controller for a familiar style of input. Two two-dimensional analog control sticks allow for careful control over the speed and direction of the car, and additional buttons can send other commands.

Figure 2-20. The wireless controller itself.

As shown in Figure 2-21, there are a list of selectable commands that can be changed. These commands, when selected, are displayed as a superimposed on-screen display (OSD) which is visible throughout the deposition lab. A press of a button then sends it to do the Galil motion controller. The acceleration and deceleration values for the movement commands are also user-configurable. When not in use, the program can be minimized to a small icon in the system tray on the task bar.

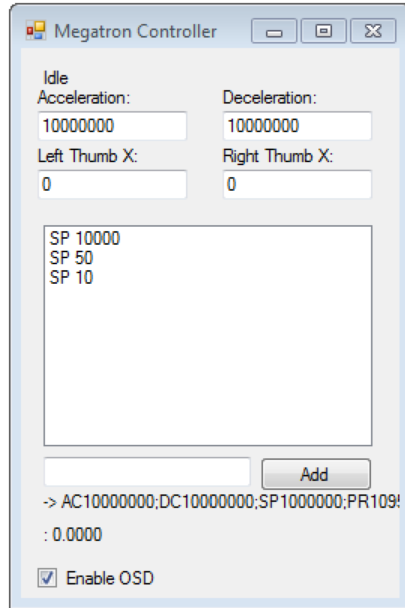


Figure 2-21. The wireless control program with configurable commands.

This simple project, developed in roughly a day, is still used extensively in the daily workings of the large deposition system. The encoder positions of various important spots in the chamber can be easily determined by using the program, and they can then be input into a control script.

Logs and the Log Viewer

In the scripting language, there is a command, *LOG*, that flags the specified process variables for periodic logging. First, those process variables would be specified. Then an additional command, *LOGRATE*, would be used to set the rate at which a snapshot of the process variables should be taken. Once the interpreter sees this command, it opens a new log file and outputs at the specified rate the values of the flagged process variables throughout the duration of the script. These logs can then be viewed with the log viewer, as shown in Figure 2-22.

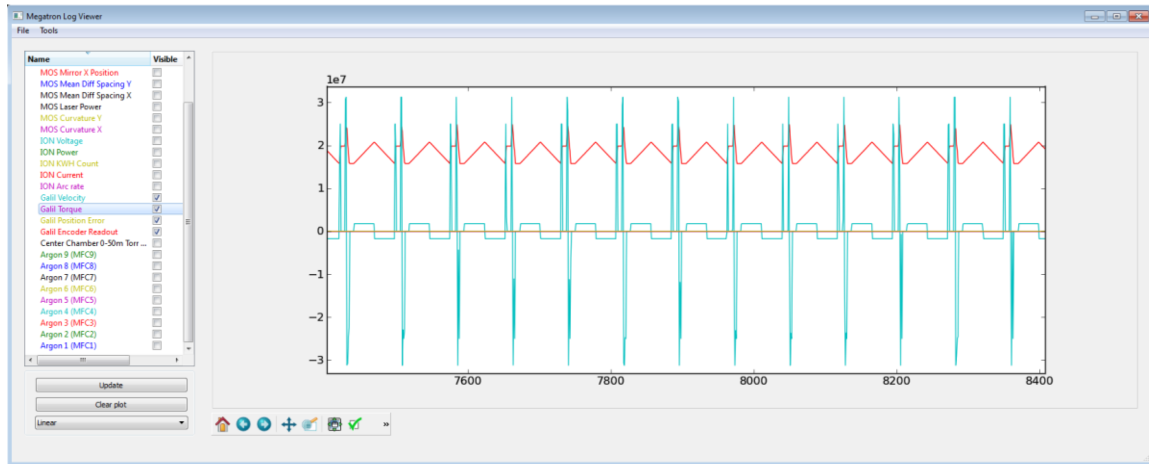


Figure 2-22. The control system log viewer.

References

"Matplotlib: A 2D Graphics Environment", J. D. Hunter, Computing in Science & Engineering, Vol. 9, No. 3. (2007), pp. 90-95

"Multilayer Laue Lens Growth at NSLS-II", R. Conley, N. Bouet, K. Lauer, M. Carlucci-Dayton, J. Biancarosa, L. Boas, J. Drannbauer, J. Feraca, L. Rosenbaum, Submitted to XRM-2010 proceedings (2010)

"Nanofocusing of Hard X-rays with Multilayer Laue Lenses", A. Macrander, H. Yan, H.C. Kang, J. Maser, C. Liu, R. Conley, G.B. Stephenson, Handbook of Optics, 3rd Edition, Optical Society of America , Volume V, Chapter 42, (2010)

"Nanometer linear focusing of hard x-rays by a multilayer Laue lens", H.C. Kang, J. Maser, G.B. Stephenson, C. Liu, R. Conley, A.T. Macrander, S. Vogt, Phys. Rev. Lett. 96, pp. 127401, (2006).

"Small d-spacing WSi2/Si narrow bandpass multilayers", C. Liu, R. Conley, A.T. Macrander, T. Graber, C. Morawe, C. Borel, E.M. Dufresne, Proc. SPIE 5537, pp. 154-160, (2004).

"The NSLS-II multilayer Laue lens deposition system", R. Conley, N. Bouet, J. Biancarosa, Q. Shen, L. Boas, J. Feraca, L. Rosenbaum, Proc. SPIE Vol. 7448 74480U., (2009)

Chapter 3

Masks

When unobstructed, a magnetron gun deposits a non-uniform thin film layer of the target material on a substrate with thickness generally proportional to the time that the connected power supply is enabled. Though the proportionality is desired, the non-uniformity is of huge detriment to the creation of MLLs; they require either absolute film thickness uniformity or an engineered thickness gradient, in the case of a wedged MLL.

To rectify this issue, stencil-like masks are introduced into the process. The shape of the mask opening is determined by the desired profile coating on the substrate, properties of the magnetron gun itself, and constraints imposed by the system. The desired profile coating depends entirely on the type of optic being created. For a flat MLL, the desired profile would be uniform; whereas for a wedged MLL, the desired profile would be linearly increasing (or decreasing, depending on the perspective). Constraints on the system include the direction of motion of the substrate carrier – whether rotary or linear, whether or not the resulting mask should be symmetric, and also physical limitations such as the maximum size a mask or its opening can be. Symmetric masks are desirable in radial systems because they remove the issue of orientation for the machine technician.

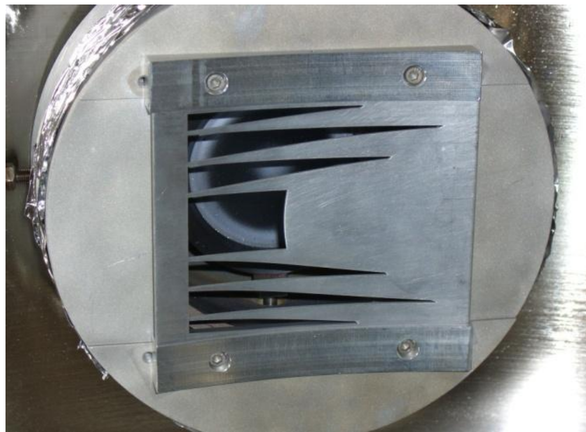


Figure 3-1. A mask for a rotary deposition system at Advanced Photon Source, fastened on top of a cylinder situated around a magnetron gun.

Mask Basics

The thickness distribution of a magnetron sputtering source can be approximated by one of many formulas for different systems. These formulas are generally based on the convolution of quasi-nth order Gaussian distributions from an infinite number of point sources situated in a ring geometry, with additional correcting factors for different systems. However, it is most accurate to grow a thin film with a stationary substrate while not using a mask and then measuring the deposited thicknesses at small increments with an ellipsometer.

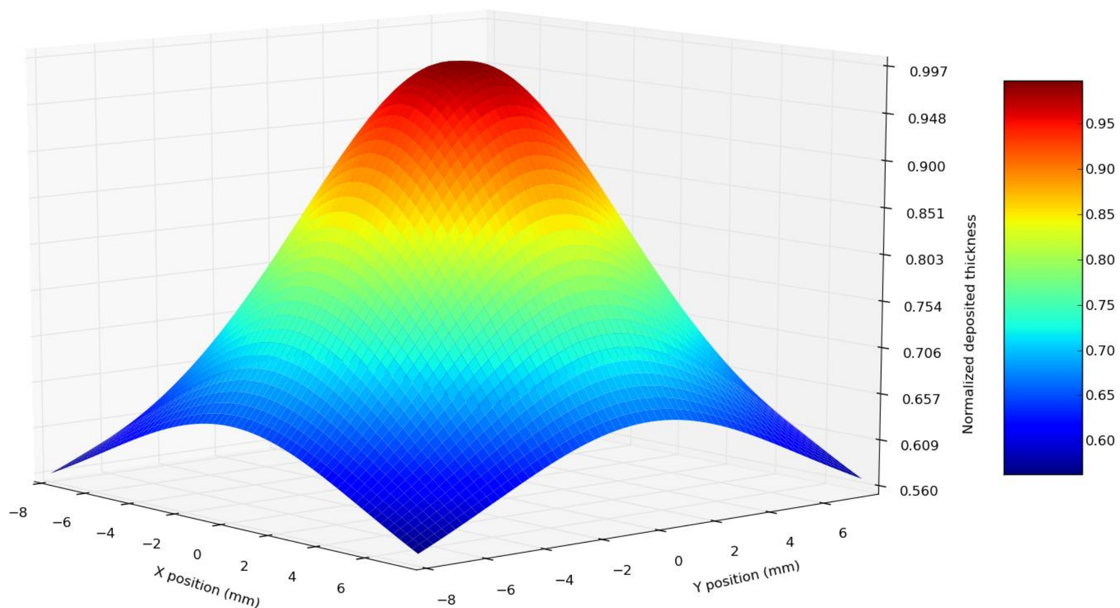


Figure 3-2. Normalized distribution of a magnetron sputtering source by formula. The central red peak indicates that the deposition concentration is in the middle.

Take, for example, that a uniform profile coating is desired in a linear deposition system. That is, the thickness of the deposited material on the substrate should be ideally constant when making lateral passes at a constant velocity over a magnetron sputtering source. Given that the film thickness distribution can be viewed as something Gaussian-like, to obtain a uniform thickness, one would expect that something needs to block out at least part of the thickest area to make up for the significantly lower growth rate off to the sides. In actuality, the resulting mask is something that looks like an inverted version of the original curve. The mask outline in Figure 3-3 shows this shape. Black represents the mask material; white is open. In order to most efficiently use the mask shape, the left and right points of the mask reach to the maximum specified mask opening.

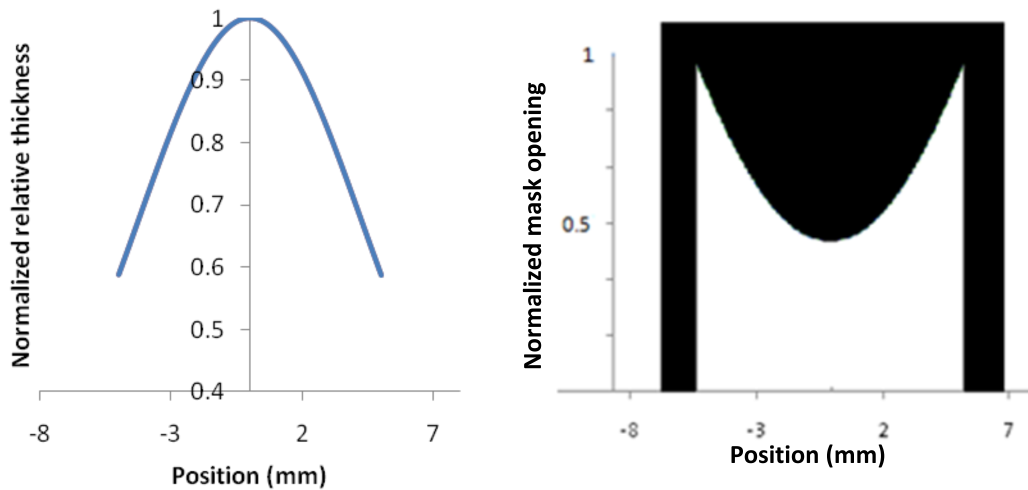


Figure 3-3. Model (left) and uniform thickness profile mask silhouette (right).

The next simplest example would be a laterally graded profile. The profile itself is essentially linear until a saturation point. The resulting mask in Figure 3-4 was calculated with 64mm as its maximum opening size. The mask shows an overall increase in the opening size from left to right, with a short dip near the center to make up for the curvature of the profile versus the thickness distribution. Finally, it continues up to the maximum possible opening size linearly, as to keep the same thickness the mask opening must necessarily increase. A silhouette as in the previous figure could be shown by filling in the regions outside of the curve and below the x-axis.

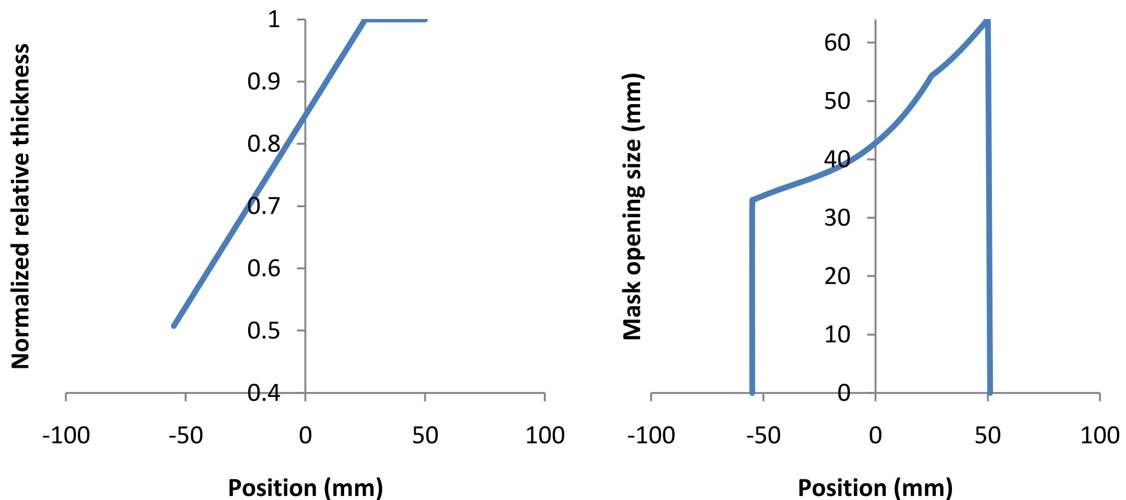


Figure 3-4. Laterally graded profile (left) and its corresponding mask for a linear deposition system (right).

Linear Deposition Mask Calculation

There are two major distinctions that determine which algorithm will be employed and how the magnetron data will be described. The basic concept behind the algorithms is the same, but the internals change depending on the movement of the sample carrier in relation to the sputtering source and, in the radial case, whether or not a symmetric mask is desired.

The magnetron data is described either by formula or ellipsometry data. For formula-based approximations, any position in the domain can be evaluated. The formulas should be designed to have a normalized maximum of 1.0. As for ellipsometry data, it is in discrete steps and is of physical height therefore necessitating normalization and 2D interpolation.

For any horizontal position x along the mask and a vertical mask opening of y, we define a numerical weight for a linear deposition system as

$$weight(x, y) \equiv \Delta x \int_{y_0}^{y_0+y} thickness(x, \gamma)\, d\gamma$$

where $y_0$is an offset indicating where the mask should be located in relation to the center of the magnetron gun, Δx is the step size between horizontal positions, and $thickness(x, y)$ is the approximated normalized thickness according to the magnetron modeling data.

The fitting process begins by taking the maximum mask opening, the magnetron deposition thickness modeling data, and the thickness profile as its input. In many cases, at least one position where the mask opening is maximal corresponds to a point where the desired thickness is also at maximum. Based on this, the algorithm sorts the input desired profile thickness in descending order and attempts to fit the whole mask based on this single point. For a test position x_{test}, input maximum mask opening of y_{max}, and corresponding desired relative thickness of τ_{test}we define

$$w_{test} \equiv weight(x_{test}, y_{max}) .$$

For each remaining position x with desired relative thickness τ_x,

$$w_{desired}(x) \equiv \frac{\tau_x}{\tau_{test}} w_{test}$$

and the minimal y that satisfies

$$w_{desired}(x) \leq weight(x, y)$$

is set to be the opening size for the horizontal position x. If this y is greater than y_{max}, this indicates a failure to fit the mask outline curve properly, and the algorithm should move onto the subsequent x_{test}. If all of the opening sizes are valid, a valid mask can be fully described by the horizontal positions and their corresponding opening sizes.

It is noteworthy that $\frac{\tau_x}{\tau_{test}} \geq 1$ and thus $w_{desired}(x) \geq w_{test}$ is itself not indicative of a failure condition. This is due to the fact that depending on the horizontal position, the magnetron modeling data could vary significantly.

Radial Deposition Mask Calculation

In the radial case, it becomes necessary to modify the weight function previously defined to take into account the arc path a single point on the sample carrier would follow over a magnetron sputtering source. Accordingly, we move to polar coordinates and define

$$weight_{radial}(r,\theta) \equiv \int_{\theta_{\min}(\mathrm{r})}^{\theta} thickness(r - R_{center}, \sigma)\, d\sigma$$

where the radius r is defined as the position on the sample carrier relative to the center of its axis of rotation, R_{center} is the radius to the center of the mask, the $thickness$ function is redefined to take polar coordinates, and $\theta_{min}(r)$ is the angle determined by the bottom of the mask at radius r. With this radial weight function in place, a mask can be fit in a similar manner as the linear case.

Using the radial weight function would likely not result in a symmetric mask. Fortunately, a simple modification yields a symmetric sweeps shown in Figure 3-6, which is described by

$$weight_{symmetric}(r,\theta) \equiv$$
$$\int_0^{\theta} thickness(r - R_{center}, \sigma) d\sigma \; + \int_0^{-\theta} thickness(r - R_{center}, \sigma) d\sigma \; .$$

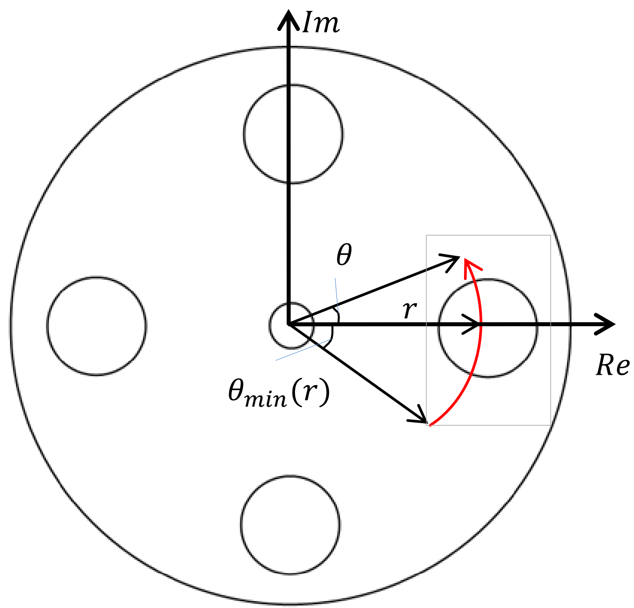


Figure 3-5. A top view of radial deposition system sample carrier, showing the relative path in red of a single point on a stationary magnetron gun as the carrier rotates.

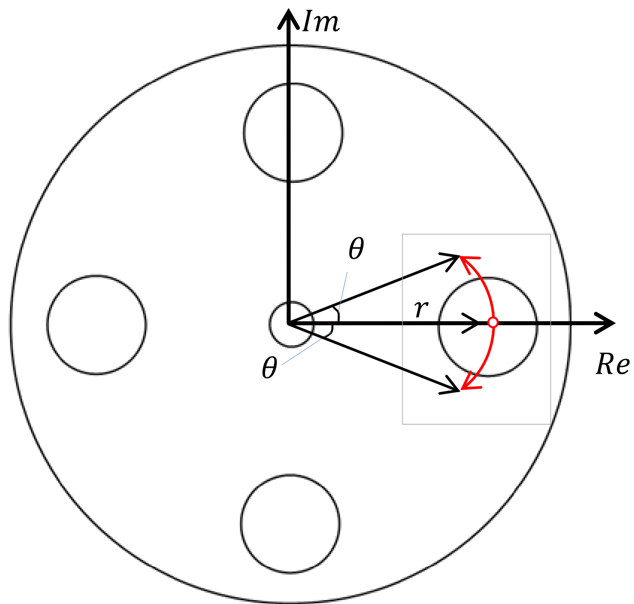


Figure 3-6. A symmetrical set of sweeping curves.

The MaskCalc Program

All of the previous algorithms are brought together in MaskCalc, a program which I developed over the fall of 2009. It is written in C# and creates masks for both linear and radial deposition systems, with the option for generating symmetric ones. It allows for custom magnetron modeling data in the form of formula input or ellipsometry data. It then requires the deposition system parameters and a user-defined desired profile coating before being able to generate a mask.

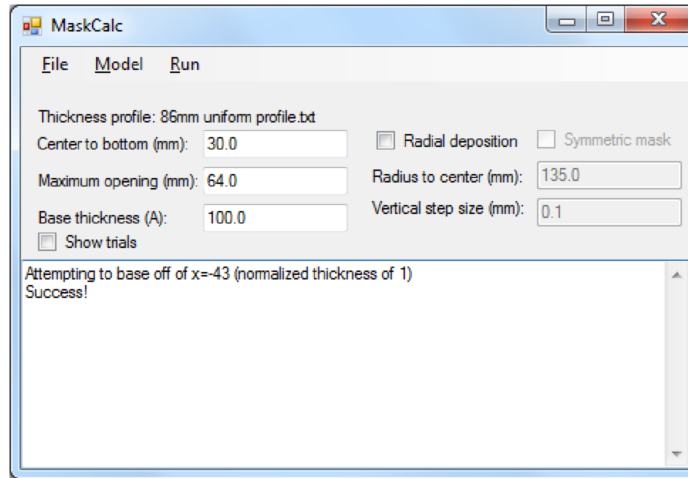


Figure 3-7. The MaskCalc application.

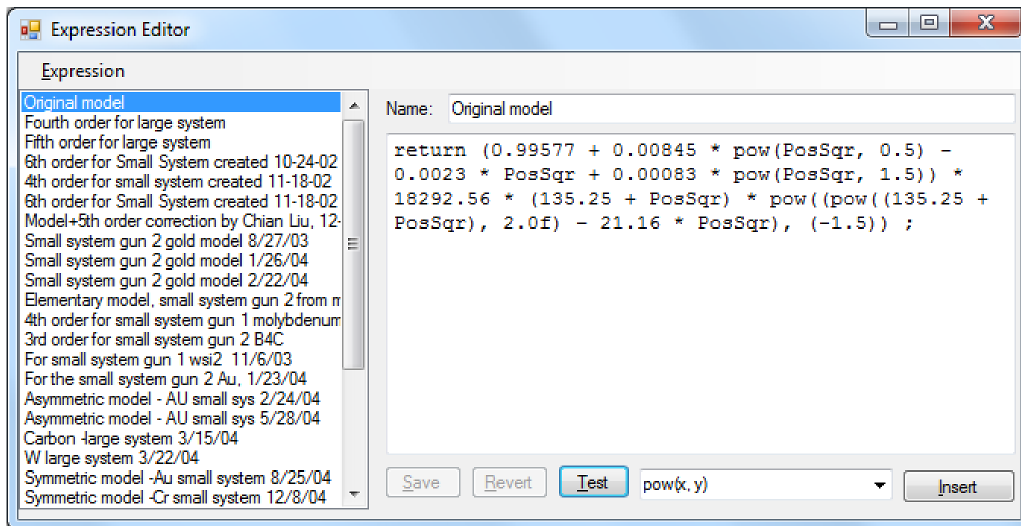


Figure 3-8. The MaskCalc expression editor.

In Figure 3-7, the main program is shown. Here, the system parameters are defined. The expression editor of Figure 3-8 can be opened by changing to the formula model. The expressions are dynamically compiled portions of C# source code for

efficiency and maximum control over the model. Simple two dimensional plots of these models, along with the resulting masks, are possible with the software.

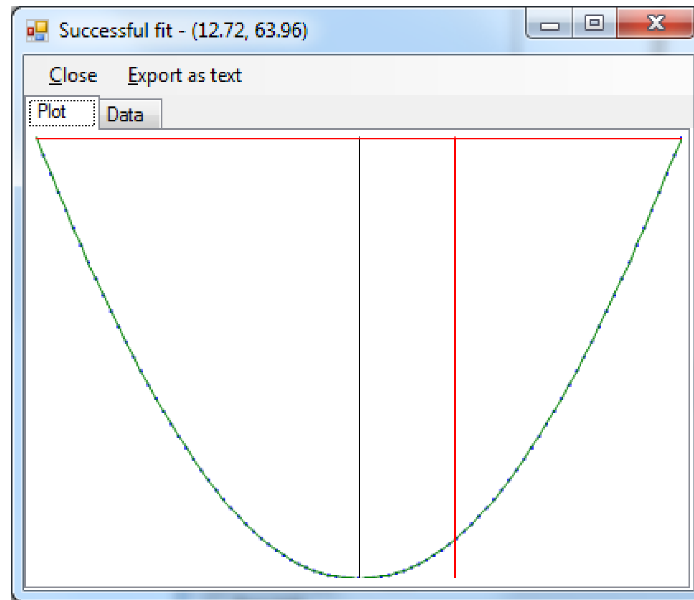


Figure 3-9. A successful fit of a simple uniform profile, utilizing the original magnetron modeling formula.

References

"Profile coating and its application for Kirkpatrick-Baez mirrors", C. Liu, L. Assoufid, R. Conley, A.T. Macrander, G.E. Ice, J. Z. Tischler, Opt. Eng, 42, pp. 3622-3628 (2003).

"Profile coatings and their applications", C. Liu, R. Conley, A. T. Macrander, L. Assoufid, J. Vac. Sci. Tech. 21, pp. 1579-1584, (2003).

"Pt KB mirrors for hard X-ray microfocusing system made by profile coating", B. Shi, C. Liu, J. Qian, W. Liu, L. Assoufid, R. Conley, A. T. Macrander, Proc. SPIE [7802-15], Accepted, (2010)

"Wedged Multilayer Laue Lens", R. Conley, C. Liu, J. Qian, C. Kewish, A.T. Macrander, H. Yan, H.C. Kang, J. Maser, G.B. Stephenson, Rev. Sci. Instrum. 79 053104, (2008)

Chapter 4

Multi-purpose Sensor Boards

Throughout the deposition process, proper magnetron sputtering gun functionality must remain uninterrupted. Temperatures rise significantly when the sputtering power supplies are sourcing, at times, 1kW or more. Ensuring the sufficient supply of chilled water to each gun is therefore essential. Also, during the coating of precision multilayer optics, the growth rate changes slightly as the target surfaces heat up before reaching an equilibrium temperature. By monitoring the temperature rise and the following time for stabilization, it may be possible to estimate and therefore compensate for growth rate change during each deposited layer.

The large deposition system had water lines for cooling the targets from the time it was first fully set up. However, there were no water flow or temperature gauges installed. I took this as a learning opportunity to design a microcontroller-based multi-purpose sensor and data acquisition system from scratch.

The Main Board

I first created a main board that could potentially connect to separate application-specific boards through an 8P8C (8 Position 8 Conductor, commonly referred to as RJ45) connector. This main board is centered around a Microchip PIC24F family microcontroller and connects to a data acquisition server computer as a full-speed Human Interface Device (HID) via USB. With USB support comes plug and play versatility, allowing users to hot swap the devices or add new ones as necessary.

Each main board has its own microcontroller embedded with a unique device descriptor, including a global vendor ID and a unique product ID, which allows the PC to differentiate between all of the devices when connected. The peripheral boards could then be sourced by the main board, as the power and ground signals are passed along the connector.

I first created a prototype, and then documented it with a schematic. From there, I laid the boards out, ordered the necessary parts, and sent the boards for fabrication at the Instrumentation Division at Brookhaven National Laboratory. Finally, they were assembled in the deposition laboratory and installed soon after.

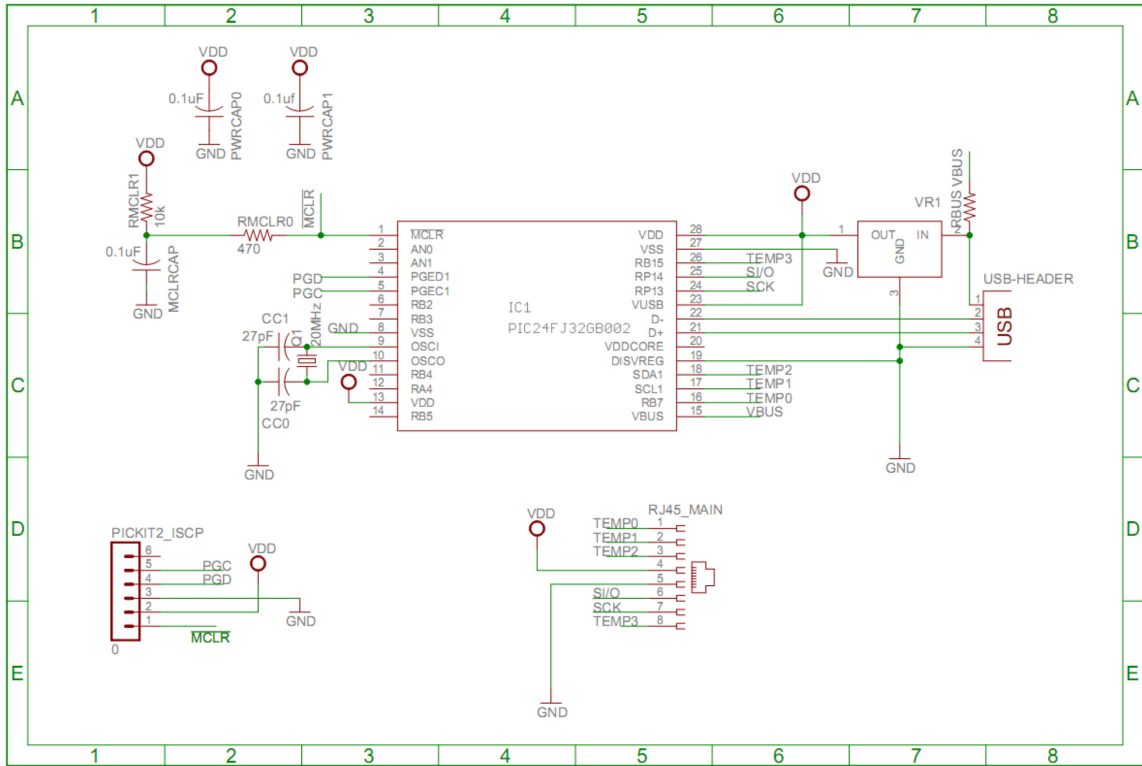


Figure 4-1. The main board schematic, with annotations for the temperature sensor RJ45 connector.

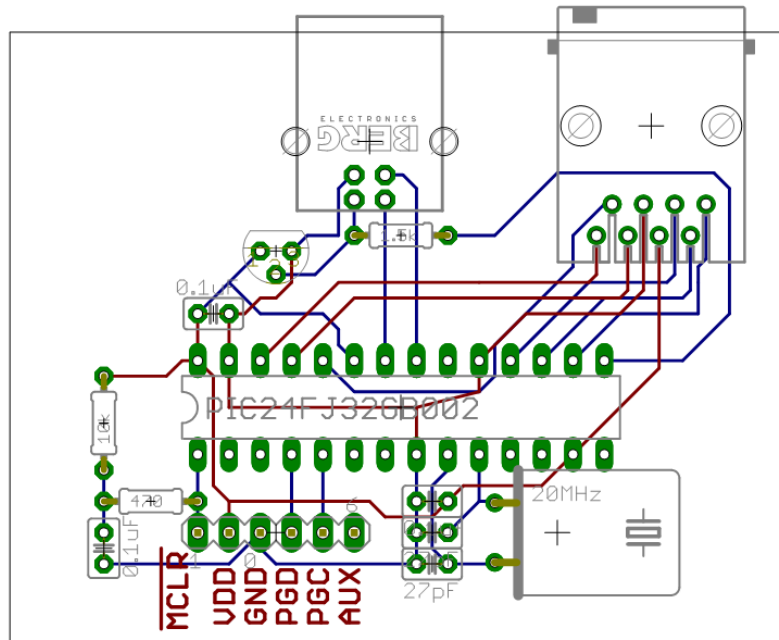


Figure 4-2. The main board layout.

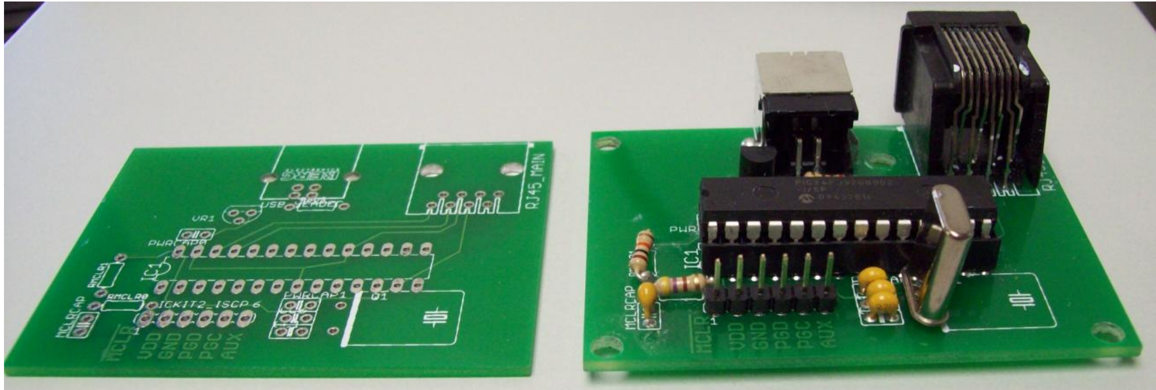


Figure 4-3. The fabricated main board (left) and an assembled main board (right).

Temperature Sensors

The temperature sensor boards were built as peripherals for the main board, and as such connect with readily available Ethernet cables. With the intention of measuring magnetron sputtering gun water supply temperature, the board itself could not be a single rigid one with several sensor integrated circuits (ICs) on it. So, the sensor boards were made to be small, containing only one temperature sensing IC, and to support daisy chaining.

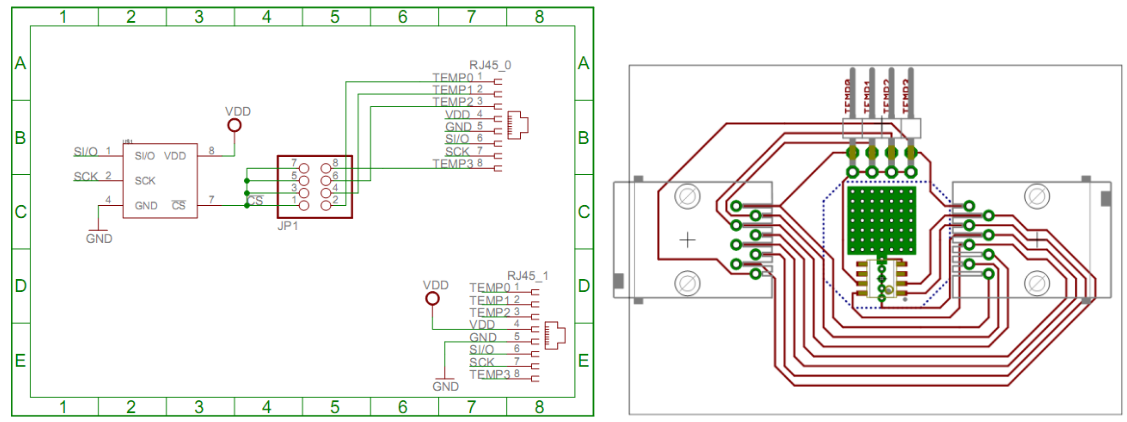


Figure 4-4. The temperature sensor board schematic (left) and single-sided layout (right), showing two RJ45 connectors for daisy chaining support.

The temperature sensing IC chosen was the low-cost Microchip TC77-3.3MOA, which features an internal band-gap type temperature sensor with digital Serial Peripheral Interface (SPI) compatibility, 0.5°C accuracy and a 12-bit Sigma-Delta ADC with 0.0625°C for the least significant bit. The temperature sensor measures the temperature of the attached PCB, and so a relatively large ground plane was added to the layout of the sensor board. With this, the board can be fastened to the object to

monitor with some thermally conductive paste in-between for effective temperature monitoring. As in Figure 4-5, the temperature sensors have been successfully mounted on metal couplings of the magnetron return chilled water supply lines.

With 8 conductors on the Ethernet cable, subtracting two for VDD and ground, it was then possible to have up to six signals passed between the daisy chained sensor boards. Subtracting another two for SPI-based communication (an I/O and a clock line), this left room for individual chip-enable selection of four temperature sensor ICs without additional logic. This certainly could have been improved with the addition of a multiplexer or even another connector from the main board.

The PIC firmware, which had basic USB support to start with, only needed to be customized for this sensing application. SPI is simple enough with PIC microcontrollers that only a short pulse on the chip-enable of a TC77 makes it send back its temperature data serially. Regularly acquiring this temperature data from each of the connected sensors, it then relays this information to the computer through USB when polled.

Finally, to allow interchangeable temperature sensor boards without individually addressable temperature sensor integrated circuits, a dual row 8-pin 0.1" header was added. With a standard 0.1" jumper, the sensor boards could be differentiated, even though each board was fabricated identically.

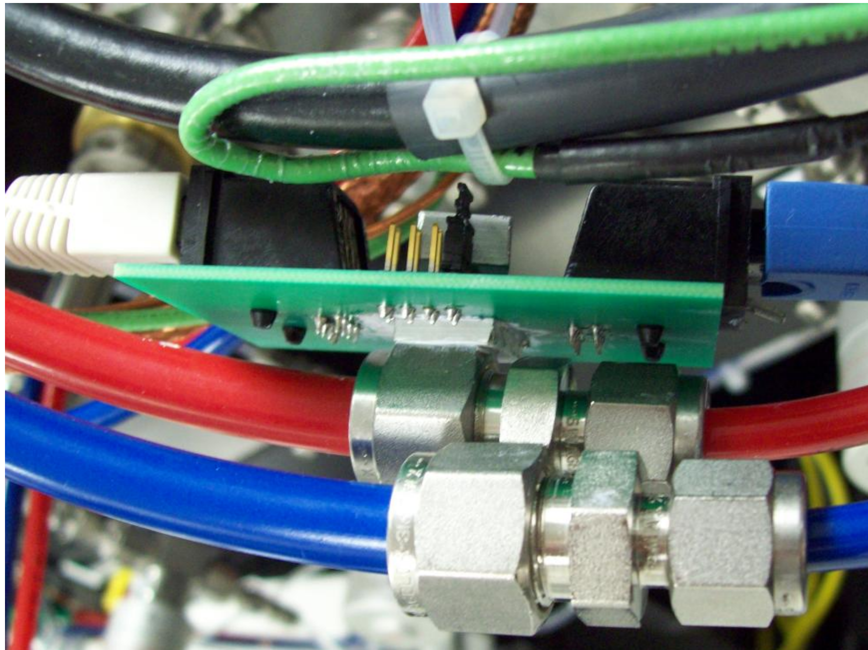


Figure 4-5. The temperature sensor board in use, mounted on a metal coupling of a water supply line.

Flow Sensors

Chosen for the task of in-line water flow sensing were Gems Sensors & Controls FT-110 turbine sensors. These sensors output 5 VDC pulses proportional to the rate of flow. This then becomes a trivial matter of pulse counting and multiplying by a fixed constant to get flow in either gallons or liters per minute.

The selection of the PIC24F microcontroller was fortunate in this instance. There are no less than five configurable timers available on the PIC24FJ32GB002, each of which can act as a synchronous counter on a customizable pin. Four of those timers are utilized for pulse counting, with the remaining one keeping track of one second intervals.

PC Software

The initial test implementation was with a custom Microsoft Visual C# application that polled the sensor boards at regular intervals and displayed the interpreted data in a normal graphical user interface. Although this application was in perfect working order, I decided to discard it and work with a more dynamic implementation in Python.

With the new implementation, a modification of a few configuration entries results in the ability to control any feasible number of temperature and flow sensors. Each device identifier is associated with a certain set of process variables in the main program. Upon polling each device, the internal database of sensor data is then updated. Whenever the main control program needs updated values, it can then simply query the sensor server.

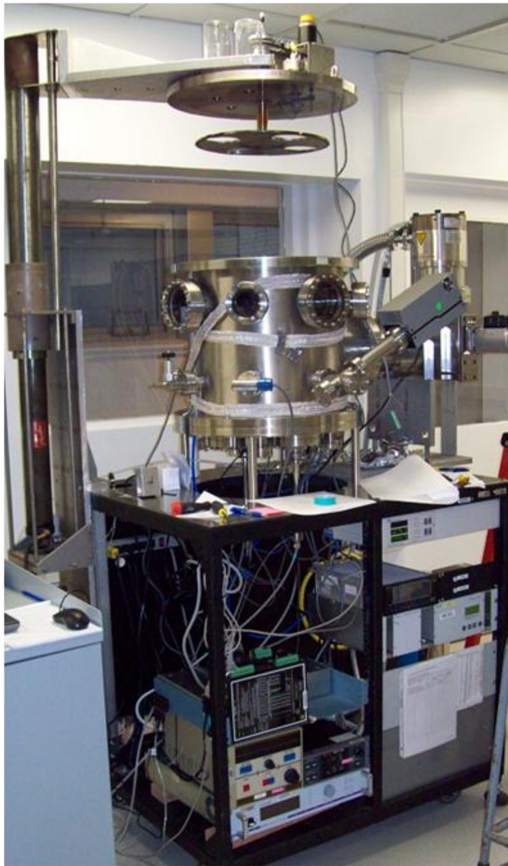
```
         Flow sensor 1 (8680)                    Flow sensor 2 (8681)                    Flow sensor 3 (8682)
     Magnetron 1 Flow: 0.66 GPM              Magnetron 5 Flow: 0.74 GPM              Magnetron 9 Flow: 0.64 GPM
     Magnetron 2 Flow: 0.63 GPM              Magnetron 6 Flow: 0 GPM                       QCM 1 Flow: 0.17 GPM
     Magnetron 3 Flow: 0.63 GPM              Magnetron 7 Flow: 0.64 GPM                    QCM 2 Flow: 0.17 GPM
     Magnetron 4 Flow: 0.66 GPM              Magnetron 8 Flow: 0.7 GPM                    Unused Flow: 0 GPM

  Temperature sensor 1 (8675)           Temperature sensor 2 (8676)          Temperature sensor 3 (8677)
Magnetron 1 Temperature 21.56 C       Magnetron 5 Temperature 21.69 C      Not connected
Magnetron 2 Temperature 21.75 C       Magnetron 6 Temperature 21.88 C
Magnetron 3 Temperature 21.13 C       Magnetron 7 Temperature 22.19 C
Magnetron 4 Temperature 22.13 C       Magnetron 8 Temperature 22.38 C
```

Figure 4-6. The command-line based sensor service status screen.

Chapter 5

Rotary Deposition System

The “Myron” rotary deposition system, named after the retired physicist who passed on the system to the Optics Fabrication Group, consists of a cylindrical vacuum chamber, a stepper motor controller, two magnetron sputtering cathodes with power supplies, a vacuum control system, and a water-cooling system for the cathodes. The addition of the stepper motor controller to the system allowed for automated depositions.



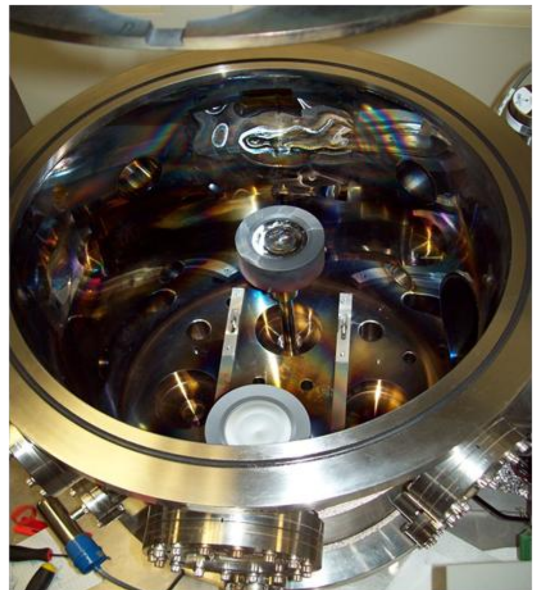


Figure 5-1. The “Myron” rotary deposition system (left) and the inside of its vacuum chamber (right).

1. Magnetron power supplies (MDX 500, MDX 1000)
2. Granville-Phillips Series 370 Stabil-Ion vacuum gauge and controller
3. MKS 146 vacuum gauge measurement and control system
4. MKS mass-flow controller
5. Water flow switches
6. Advanced Energy Cesar RF power generator
7. Pfeiffer TCM1601 Pump
8. Measurement Computing USB-1208HS-4AO for data acquisition
9. Parker Compumotor SX-6 Microstepping Drive (replacing an AllMotion EZHR23)

Figure 5-2. The rotary deposition system hardware.

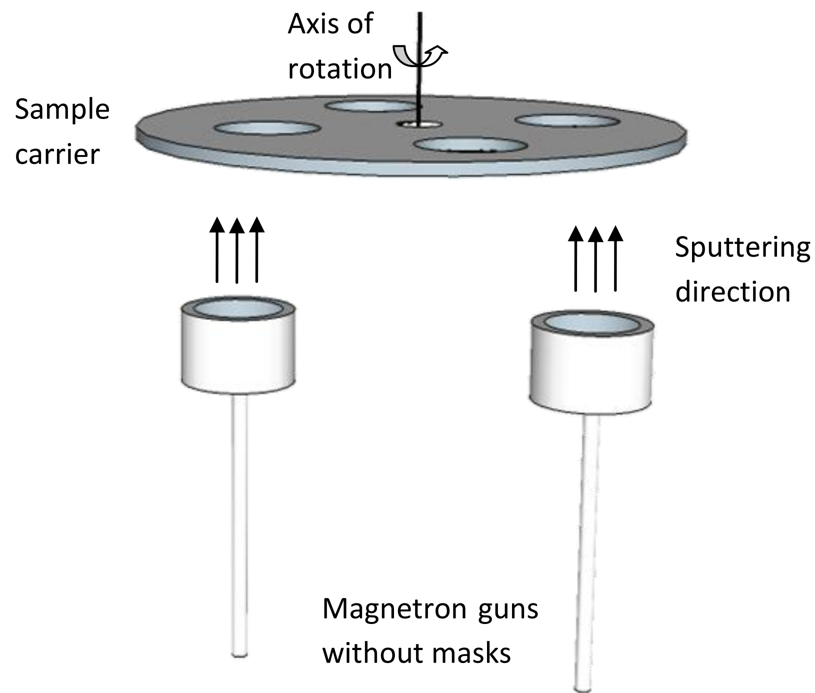


Figure 5-3. A radial deposition system with two magnetron guns 180° apart, indicating the rotation of the sample carrier in relation to the sputtering sources.

The Original Control System

A control system for a similar machine was developed in the early 2000s at the Advanced Photon Source (APS) at Argonne National Laboratory. This control system was then reused for the small deposition system at Brookhaven National Laboratory. When the stepper motor rotated the sample carrier, it appeared to struggle and not smoothly rotate the carrier. This became the number one hardware issue to resolve. As far as software, the control system worked, but it had some usability issues. Though a complete rewrite of the system would have been desirable, due to time constraints simply addressing the usability concerns was prioritized. Along the way, refactoring the source code became the next priority.

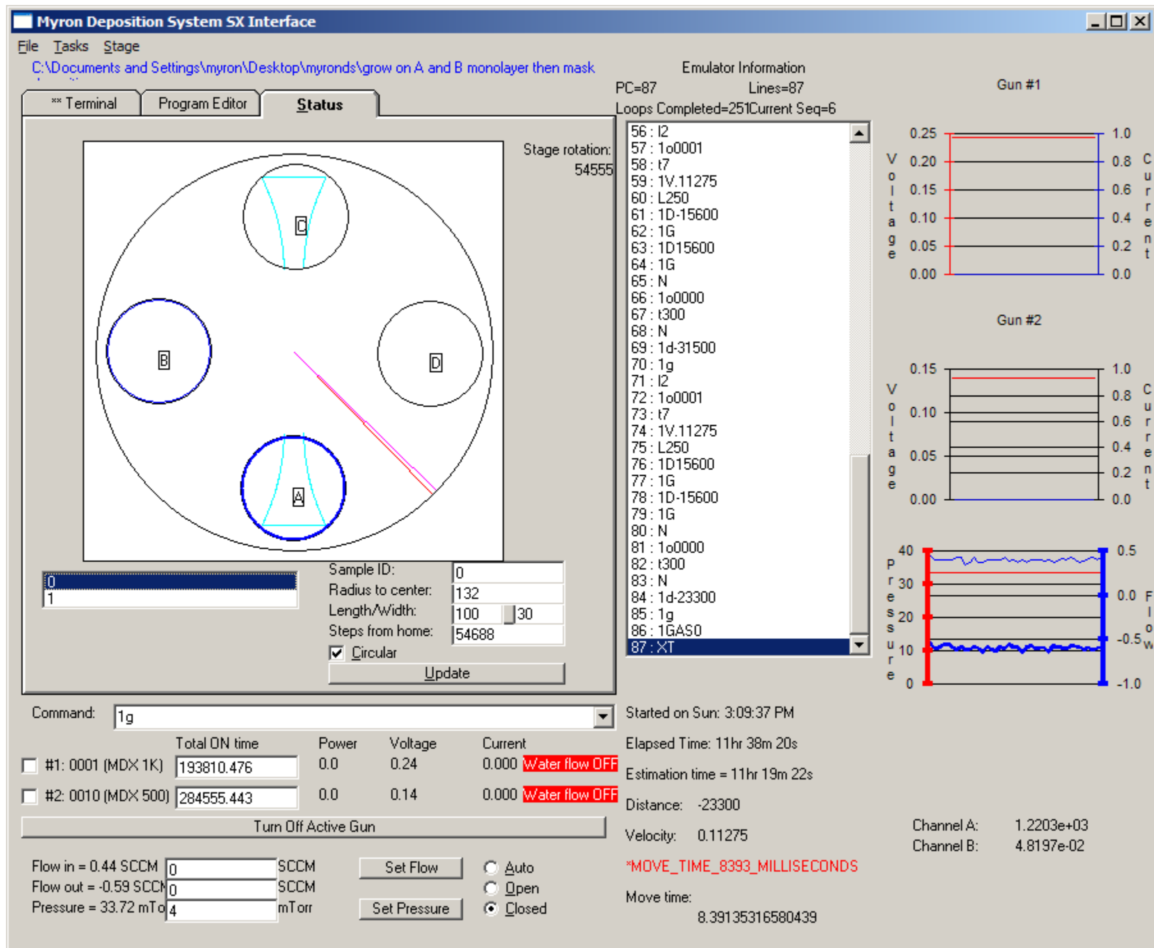


Figure 5-4. The improved control system for the rotary deposition system.

The Improved Control System

The stepper motor control issue was eventually fixed by replacing the AllMotion stepper controller with a more configurable and time-tested alternative, the Parker Compumotor SX6 drive. Integration into the system mostly meant fixing up time estimation and changing how moves were performed, in addition to basic hardware setup and communication configuration.

Once the movement issue was resolved, it was then possible to do reliable depositions. However, the concerns raised by technicians and other users had not yet been worked out. Without proper cooling, the expensive sputtering targets can overheat, crack, and become useless. Liquid flow switches, which indicate whether or not water is flowing through it by producing logic-level voltages, were then added to the magnetron gun chilled water supply lines. Monitoring them via software allowed for sputtering sources to be disabled if the sensors indicated that flow had been interrupted. Additions to the scripting language, like allowing for control of the mass flow controllers (MFCs), which control the rate of process gas flow into the system, also simplified the process of running a deposition.

Knowing the current rotation angle of the sample holder before and during the execution of a script is absolutely vital. Without an encoder to keep track of the current position of the carrier – and therefore only the potential for relative movements – it is necessary to manually move the carrier to the correct starting position. The addition of the Status tab (see Figure 5-4) helped visualize the whole process.

After having properly homed the sample carrier, the visual representation of the sample carrier shows the current rotation. In addition, it shows whether or not the magnetron sputtering cathodes are active and also the proper orientation of the masks loaded from MaskCalc (see Chapter 3 for more information about masks). The user could then also draw either circular or rectangular samples to this status window and ensure that the deposition scripts run as expected.

Chapter 6

Quality Analysis

In creating a multilayer optic, it is crucial to verify the quality and accuracy of the process undertaken. As explained in the deposition system section, utilizing quartz-crystal rate monitors help to give a rough idea of the layer thickness of an in-progress deposition. This reading is a measure of the amount of material deposited on the rate monitor itself, however, and could differ from that which reaches the substrate due to various factors.

Reflective multilayer optics such as multilayer monochromators (MLMs) can be tested with an X-ray diffractometer like the Bruker D8 Discover to visualize the angular reflectance versus angle and ensure it conforms with the desired reflectivity. However, MLLs are transmission optics and require considerably more effort to inspect. One way to ensure a correctly deposited MLL structure is to just use the optic on the beamline. While this method works well, beam time is a precious commodity that only comes for a mere week every so often. Beam time is therefore meant to be used for actual experiments and not just verifying correct optical characteristics. In this chapter, I will introduce an alternative method for measuring the quality of a MLL.

Given the nature of MLLs, it is impossible to physically check the deposited thickness. As such, we turn to scanning electron microscopy. A scanning electron microscope (SEM) allows us to view the individual layers as electron micrographs.

These grayscale micrographs work exceptionally well for multilayers of alternating electron density, since pixel intensities vary according to electron density. This is absolutely fundamental to the analysis of MLLs; as for the two common materials in current MLLs, WSi_2 appears as a light region and Si appears as a relatively dark region. Taking the image in Figure 6-1 as input, individual layer thicknesses can be determined by image processing techniques.

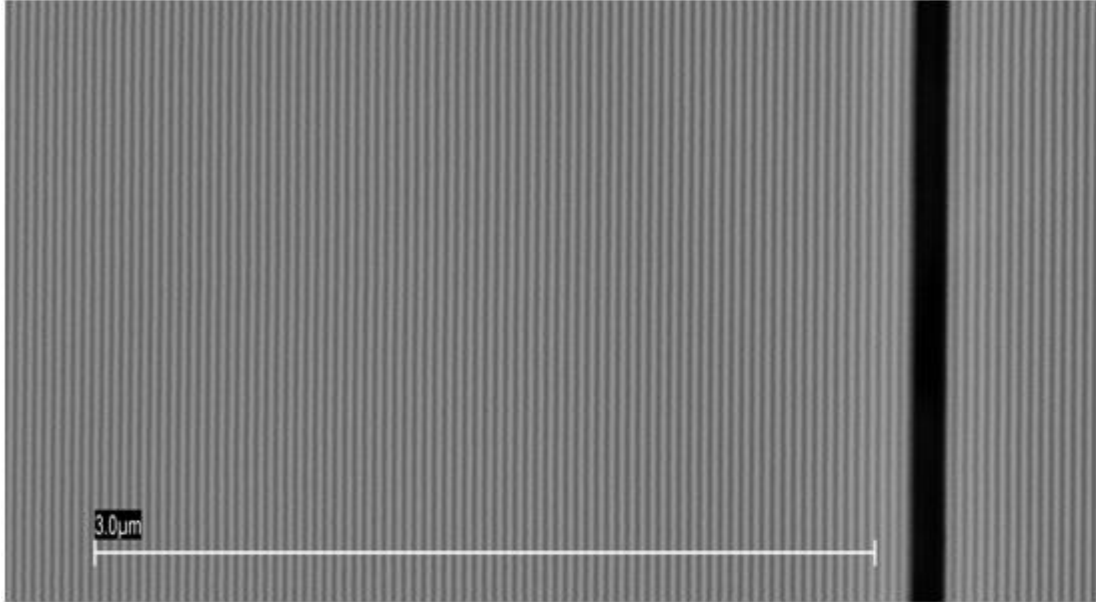


Figure 6-1. A clean electron micrograph of an MLL section; WSi_2 is light and Si is dark.

Even from a clean electron micrograph of a multilayer, it is not entirely clear where one layer starts and the next begins. The image processing method I employ is to turn the electron micrograph into a binary image. With such an image, it becomes trivial to determine layer thicknesses in pixels. With layers being roughly vertical lines, as they are in the example figures, averaging over all of the rows and multiplying by a conversion factor of angstroms per pixel, the result is the per-layer thickness.

Pre-processing

The algorithm I make use of assumes vertically oriented layers; this is the same perspective as if one were looking straight down the linear assembly of the large deposition machine, with the magnetron guns depositing onto the substrate in front (see Figure 2-8 and Figure 2-9). SEM images are not always taken in the same perspective. Before the algorithm can be employed, a manual rotation angle must be specified to correct for this.

Also, not all electron micrographs are as clear as shown in the examples. Many have artifacts or aberrations arising from, for example, polishing compound, improper polishing practices, dust on the sample, and SEM stage drift. Growth imperfections during the deposition process itself or plasma perturbation could also result in major issues. These imperfections usually lead to a specific region of interest (ROI) being chosen manually for processing.

Image Processing

The method I employ takes a cross-shaped aperture (pictured in Figure 6-2) over the region of interest. The horizontal and vertical slices of the pixel intensities exposed by the aperture are averaged into $\overline{I_h}$ and $\overline{I_v}$, respectively, for a single pixel of interest (POI). That is, assuming odd aperture dimensions, for any (x, y) in the ROI,

$$\overline{I_h}(x, y) = \sum_{i=x-\lfloor aperture_size_x/2 \rfloor}^{x+\lfloor aperture_size_x/2 \rfloor} image[i][y] \;/\; aperture_size_x$$

$$\overline{I_v}(x, y) = \sum_{i=y-\lfloor aperture_size_y/2 \rfloor}^{y+\lfloor aperture_size_y/2 \rfloor} image[x][i] \;/\; aperture_size_y \;.$$

Due to the inherently vertical nature of MLLs, $\overline{I_v}$ is a good indication of the current material of the layer (or graphically, the color of the column). For example, an average intensity close to that of white, would indicate a layer of WSi_2. Conversely, $\overline{I_v}$ closer to black would indicate Si.

A simple threshold for $\overline{I_v}$ is not enough to fully classify the pixel of interest or POI (i.e., the intersection of the horizontal and vertical line of the aperture). Instead, a comparison between the horizontal and vertical averages is done which takes the spatial locality of the scan into consideration. Given that pixel intensities range from 0 (black) to 255 (white) on a single byte scale, for the purposes of this explanation let's assume normalized values of 0.0 and 1.0 for black and white, respectively. Then, in the case that $\overline{I_v} > \overline{I_h}$, the POI would be colored white for WSi_2. In the other case, $\overline{I_v} \leq \overline{I_h}$, the POI would be colored black for Si.

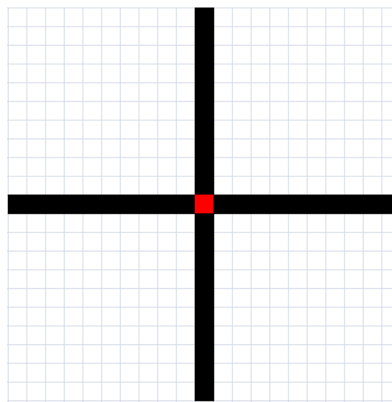


Figure 6-2. 20x20 pixel cross aperture. The POI is colored red in the center.

With a processed region of interest, as shown in Figure 6-3, per-row layer widths in pixels can be calculated straightforwardly by horizontal scans. A state variable holds the current material, which is initialized to the material of the leftmost pixel. The variable holding the current layer's width is incremented for each subsequent pixel of the same material. Upon encountering the other material, the state variable is toggled, the current layer's width is fixed, and the width of the following layer is initialized to 1. The average layer width in pixels for the whole ROI can then be calculated by averaging all of the per-row layer widths.

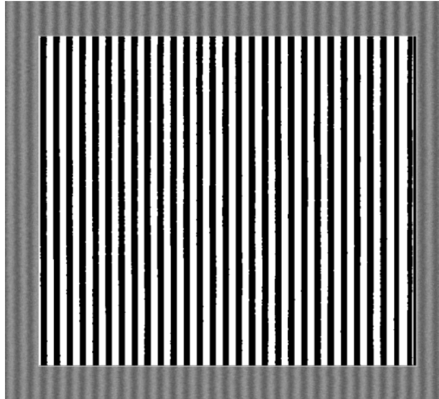


Figure 6-3. A small processed ROI.

In some cases, the detected number of layers in a row can be different from the other rows. In this case, the median number of layers detected is used, and rows not corresponding to the median are discarded. See, for example, Figure 6-4. In this figure, roughly the top half will have an additional layer compared to the bottom half. Not all occurrences of this are so simple. Aberrations can appear in different ways and can potentially appear anywhere in the image, so intelligently detecting these regions would take considerable effort. It is therefore recommended to properly select the ROI and do the necessary preprocessing to optimize the number of usable layers.

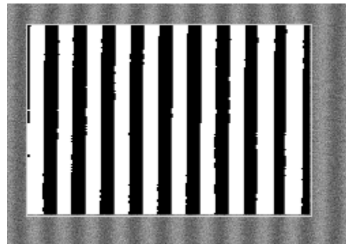


Figure 6-4. Poor region selection and improper rotation lead to different layer counts.

```
/* Process the ROI of image utilizing a square cross
   aperture into a binary image */
function digitize_image(image[], ROI, aperture_size):
    int scan_side=(int)(aperture_size / 2)

    /* For simplicity, arrays are defined with first and
         last indicies. */
    int hsum[]=array[ROI.y:ROI.y+ROI.height]
    int vsum[]=array[ROI.x:ROI.x+ROI.width]
    Image new_image[,]=allocate_image(ROI)

    /* Calculate top row's vsum and the left column's hsum.
       Note that the range of i is inclusive. */
    for each i in [-scan_side, scan_side]:
         for each x in ROI:
             vsum[x] += image[x][ROI.y+i]
         for each y in ROI:
             hsum[y] += image[ROI.x+i][y]

    /* Regarding overflow:
       Assuming 32-bit unsigned integers for the sums and
       and single byte intensity values from the images, the
       vertical and horizontal sums could only overflow if
       utilizing 2^32/2^8 = 2^24 pixels for the full scan.
       Since typical aperture sizes are low multiples of
       10x10, this is not an issue.
    */

    // Begin processing.
    for each x in ROI:
         for each y in ROI:
             // Update the vertical sum when necessary
             if x > ROI.x:
                  vsum[x]-=image[x][y-scan_side-1]
                  vsum[x]+=image[x][y+scan_side  ]

             // Update the horizontal sum when necessary
             if y > ROI.y:
                  hsum[y]-=image[x-scan_side-1][y]
                  hsum[y]+=image[x+scan_side  ][y]
```

Figure 6-5. Image digitization algorithm. (continued on the following page)

```
        if (vsum[x] > hsum[y])
            new_image[x][y]=white
        else
            new_image[x][y]=black

/* Do a bit of post-processing to fix the cases where
   there are pixels alternating vertically, as with a
   true multilayer that cannot physically happen.
*/

for each x in ROI:
    for each y in ROI:
        if new_image[x][y+1]==new_image[x][y-1]:
            new_image[x][y] =new_image[x][y-1]

/* Note that the above can be integrated with the main
   loop, but for the sake of simplicity it is shown
   here.
*/

return new_image
```

Figure 6-5. Image digitization algorithm.

```
/* From a processed binary image, calculate layer widths in
the ROI */
function get_layer_widths(image, ROI, angstroms_per_pixel):

    /* Though very unlikely, the maximum possible count of
       layers is found when there are pixels of alternating
       color on a horizontal scan. */
    int layer_widths[,]=array[ROI.width,ROI.height]
    int row_layers[]=array[ROI.y:ROI.height]
    int layer_y=0

    for each y in ROI:
        int layer_x=0
        int current_compound=image[rgn.X][y]

        for each x in ROI:
            if image[x][y]==current_compound:
                // Same compound -> same layer.
                layer_widths[layer_x][layer_y]++
            else
                // Advance to the next layer
                layer_x+=1
                layer_widths[layer_x][layer_y]=1
                current_compound=image[x][y]

        // Record the number of layers in this row
        row_layers[y]=layer_x
        layer_x+=1

    int mode_layers=find_mode(row_layers)
    int row_count=count_rows_with_n_layers(row_layers,
                                              mode_layers)
    float avg_layer_widths[]=array[mode_layers]

    for each layer in mode_layers:
        int sum=0
        for each row with layer count == mode_layers:
            sum+=layer_widths[layer][row]

        avg_layer_widths[layer]=(sum/row_count)*
                                 angstroms_per_pixel
    return avg_layer_widths
```

Figure 6-6. Layer width calculation algorithm.

Actual Implementation

The current implementation of the algorithm described previously is in fact simplified, as we assume a square aperture. The aperture size itself remains the same throughout the image; this means the horizontal and vertical averages always have the same divisor. That is, if $aperture_size \equiv aperture_size_x = aperture_size_y$, and defining

$$hsum(x, y) \equiv \sum_{i=x-\lfloor \text{aperture_size}/2 \rfloor}^{x+\lfloor \text{aperture_size}/2 \rfloor} image[i][y]$$

$$vsum(x, y) \equiv \sum_{i=y-\lfloor \text{aperture_size}/2 \rfloor}^{y+\lfloor \text{aperture_size}/2 \rfloor} image[x][i]$$

we then have:

$$\overline{I_h}(x, y) = hsum(x, y) \;/\; aperture_size$$

and

$$\overline{I_v}(x, y) = vsum(x, y) \;/\; aperture_size \;.$$

The original comparison $\overline{I_v} > \overline{I_h}$ is, by definition,

$$vsum(x, y) \;/\; aperture_size \; > hsum(x, y) \;/\; aperture_size$$

which can be clearly simplified by multiplying by $aperture_size$ to:

$$vsum(x, y) > hsum(x, y)$$

Thus, the horizontal and vertical sums themselves can be relied upon to determine the proper material for the POI. These sums are easy to store in conjunction with scrolling buffers, which means that the ROI can be digitized in time proportional to the number of pixels contained in it. For further implementation details, see the algorithms in Figure 6-5 and Figure 6-6.

The Zone Plate Digitizer Program

The layer width calculation algorithm described in the previous sections makes for the basis of the Zone Plate Digitizer program, which I developed in the fall of 2009. It is written in C# and utilizes multithreading to allow for efficient parallel processing of electron micrographs.

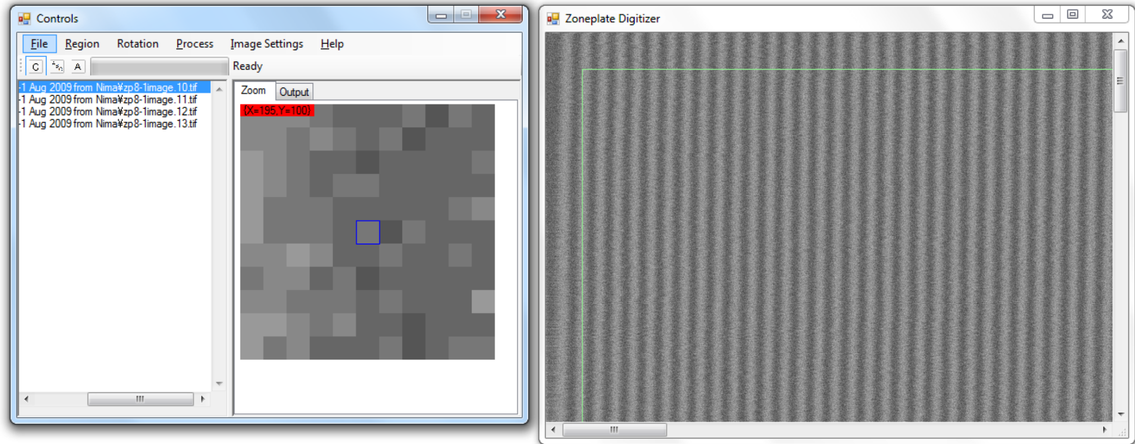


Figure 6-7. The Zone Plate Digitizer Application.

Typical Usage

A typical usage of the program involves first loading of the batches of SEM images, as in Figure 6-7. Rotated images can then be corrected by first processing an image, isolating a single layer, and selecting two points on a line defined by the layer to indicate the rotation angle.

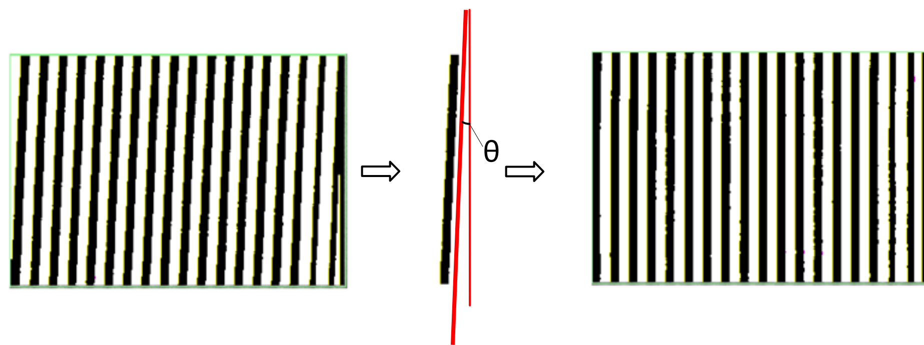


Figure 6-8. Isolating a single layer to find the angle of rotation (θ).

Once the images are properly rotated, the user should, through a similar process as the rotation, define the angstroms per pixel conversion factor. This factor can be determined by using the scale that is already present on the electron micrographs (see, for example, the 3.0µm scale in Figure 6-1).

The user can then select the region of interest globally or per-image and then begin processing. Upon getting a good result for the usable rows in each image, like in Figure 6-9, a spreadsheet for the whole project can be exported with the layer thickness data, calculations for inverse d-spacing of the multilayer, and automatically generated plots.

```
Layer 000: 4.066 pixels  59.795 angstroms
Layer 001: 9.099 pixels 133.810 angstroms
Layer 002: 9.066 pixels 133.324 angstroms
Layer 003: 9.108 pixels 133.948 angstroms
Layer 004: 9.212 pixels 135.474 angstroms
Layer 005: 9.189 pixels 135.128 angstroms

...

Layer 231:  7.901 pixels 116.190 angstroms
Layer 232: 10.906 pixels 160.377 angstroms
Layer 233:  7.538 pixels 110.849 angstroms
Layer 234: 10.929 pixels 160.724 angstroms
235 mode row layers
0 rows that deviate from mode
2000 usable rows
Elapsed times:
0.12s Image processing
0.05s Width calculation
```

Figure 6-9. A good result from the algorithm.

Plotted Results

Taking the calculated layer widths from the individual electron micrographs and effectively stitching them together, the software generates a Microsoft Excel spreadsheet and automatically generates a plot of the inverse d-spacing. This inverse d-spacing is the desired end-result and is a great indicator of how well all of the theory and processes in the planning and execution of the MLL deposition went. See Figure 6-10 for an example of the inverse d-spacing.

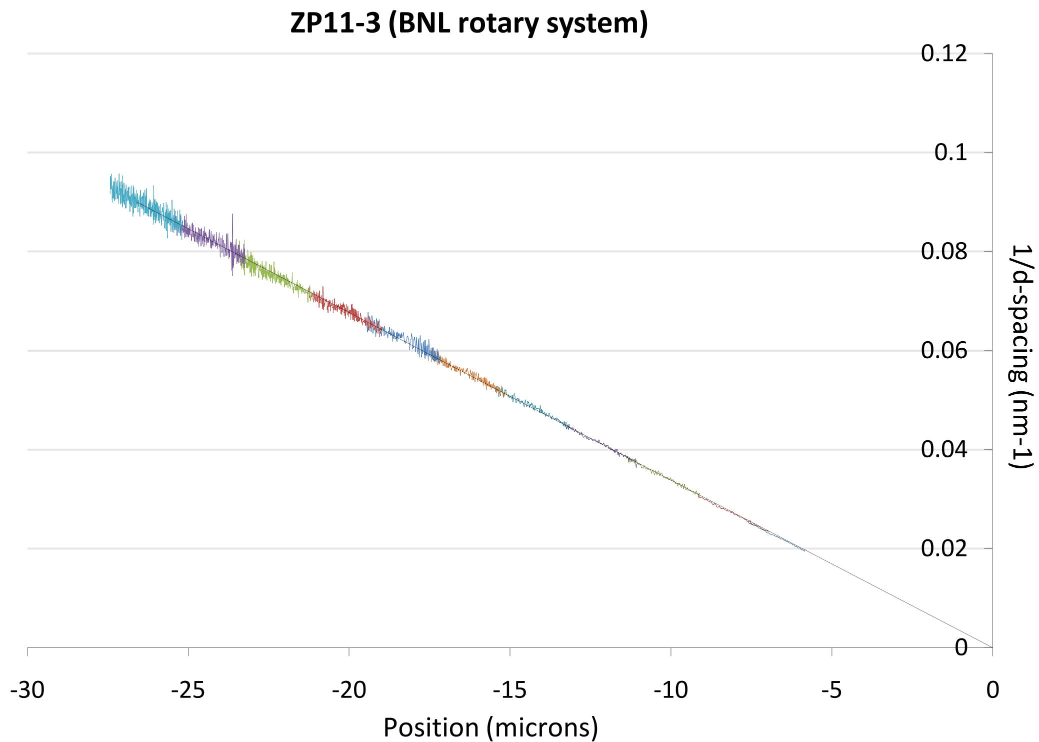


Figure 6-10. Individual electron micrographs stitched together into a graph of inverse d-spacing, with contributions from each image in different colors.

References

"Characterization of a multilayer Laue lens with imperfections", H. Yan, H.C. Kang, J. Maser, A.T. Macrander, C.M. Kewish, C. Liu, R. Conley, G.B. Stephenson, Nuclear Instruments and Methods A, Volume 582, Issue 1, (2007), pp. 126-128

"Metrology of multilayer Laue lens structures by means of scanning electron microscope imaging" N. Jahedi, R. Conley, B. Shi, J. Qian, K. Lauer, A. Macrander, Nuclear Instruments and Methods A, 616, pp. 89-92, (2010)

"Sectioning of multilayers to make a multilayer Laue lens", H.C. Kang, G.B. Stephenson, C. Liu, R. Conley, Khachatryan, M. Wieczorek, A.T. Macrander, H. Yan, J. Maser, J. Hiller, and R. Koritala, Rev. Sci. Instrum., 78 (4) (2007) 046103.

Chapter 7

X-ray Optics Simulator

Currently, there exist several good ray tracing programs that simulate x-ray optics, such as SHADOW[6]. However, ray tracing methods can be insufficient to model the performance of a beamline because ray tracing makes only geometric assumptions about the particles in transit. Recently, interest has been increasing in the significantly more accurate method of modeling the x-rays as waves.

Utilizing the wave propagation method, operations can also be significantly simplified by operating on the spectrum from the Fast Fourier Transform (FFT). The main drawback is that while the ray tracing methods are relatively straightforward to implement, wave propagation takes considerable effort and understanding of complex physics to implement. Due to the complexity, the details of the underlying mathematics and algorithms will not be covered. The credit for this vital work goes to Hanfei Yan, of the Photon Source Directorate at Brookhaven National Laboratory.

Software

With the benefits of choosing Python for the control system of the large deposition system, it was also a clear choice for this simulator. Being free and open-source multiplatform software, it will have the most potential to benefit other theoretical physicists who may not all be using the same operating system. The professional quality plots that Matplotlib offers made it the best choice for this project.

Of course, as an interpreted language, Python does not offer the greatest amount of speed when doing the large calculations that a simulator like this might require. The idea then evolved to have a Python front-end to components (i.e., x-ray optics) written in the modern dialect of C, C99, which can be compiled on many platforms with free compilers such as gcc[7] or MinGW[8]. This allows for the fast prototyping capabilities of Python and the calculation speed of the relatively low-level C. This is the direction the project took, and as such Python and C source code along with build scripts for Windows and Makefiles for Linux were created.

---

[6] See the ESRF page at http://www.esrf.eu/computing/scientific/raytracing/

[7] http://gcc.gnu.org/

[8] http://www.mingw.org/

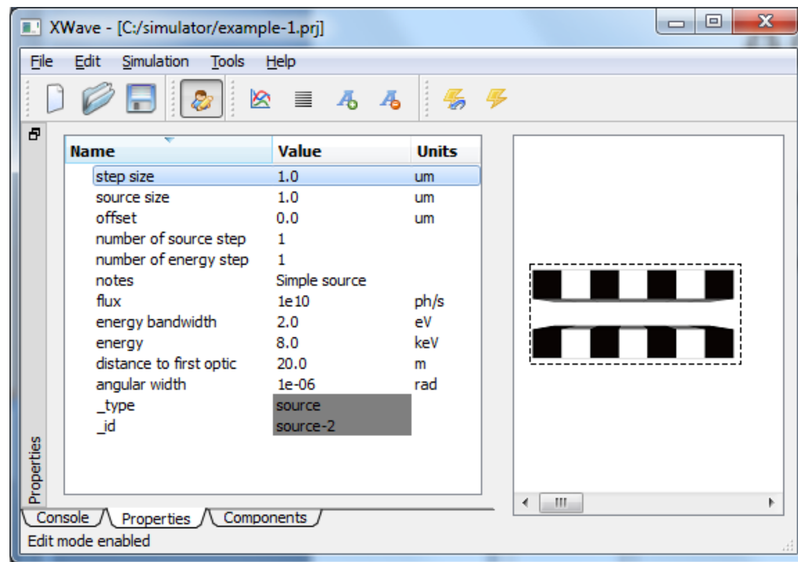


Figure 7-1. The source and its properties shown in the simulator.

Human-readable project files along with command-line parameters allow for the exchange of component information between the front-end and the components. Each component (with the exception of the source) takes, a data file as input along with either a project or in some cases just some command line parameters. The output of the component is then readable by the next component. This data is stored in a binary format and is parsable by both the components and Python, allowing for Python to generate plots with Matplotlib. From the data files, all interesting wave field data can be determined. This leads to plots of the wave field's intensity, phase, and angular spectrum.

A custom aggregated material database was created to allow for automatic calculation of the material and energy-specific complex refractive index and the complex Fourier coefficient of the susceptibility function of a crystal. This is vital for allowing different optics to be made of different materials.

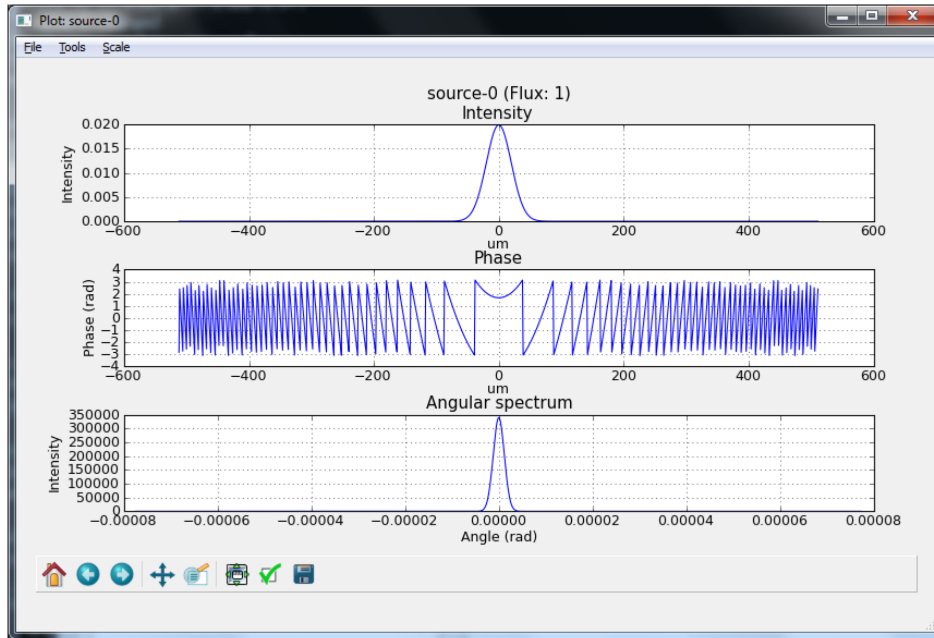


Figure 7-2. A plot of the intensity, phase, and angular spectrum of a wave field some distance from a simple source component.

Beamline Optics

Requisite for any experiment is a source. The initial source module (Figure 7-3, left) that we have developed is not meant to accurately model any real undulator in specific; it merely models a simple or extended point source. It can be configured to have an energy spread in a particular range, allow for multiple points, or both. Additionally, the size of the source and its angular width can be specified (see Figure 7-1). A simulation package like this would also not be complete without also having a freespace propagation component. The freespace component offers two modeling choices, either based on the Fresnel-Kirchoff formula or using the angular spectrum of the wave.

Conceptually and computationally, the easiest component is the single slit. It only lets a very small portion of the incoming wave through, depending on the slit opening size, its thickness, and the material with which it is made. All of these options are configurable, in addition to the simulation model – geometrical theory-based or using beam propagation methods.

Figure 7-3. The source component (left) and single slit component (right) .

Double slits (Figure 7-4, left), as the name suggests, are like two single slits. They can be used to measure coherence in an experiment. Fresnel zone plates (Figure 7-4, right), which use diffraction to focus x-rays, can also be used with configurable materials, focal length, thickness, radius, and a beam stop range.

Figure 7-4. The double slit component (left) and the Fresnel zone plate component (right).

Consisting of two single crystals, a double crystal monochromator (Figure 7-5, left) filters out a single x-ray energy. Its material, reflection indices, polarization, incidence angle, and the separation distance of the two crystals are all configurable.

Flat multilayers (Figure 7-5, right) can also be used. Any number of layers can be specified with material, thickness, sigma thickness, and sigma roughness all customizable. Additional tools for multilayers are also included with the software package, as described in the next section.

Figure 7-5. The double crystal monochromator component (left) and the multilayer component (right).

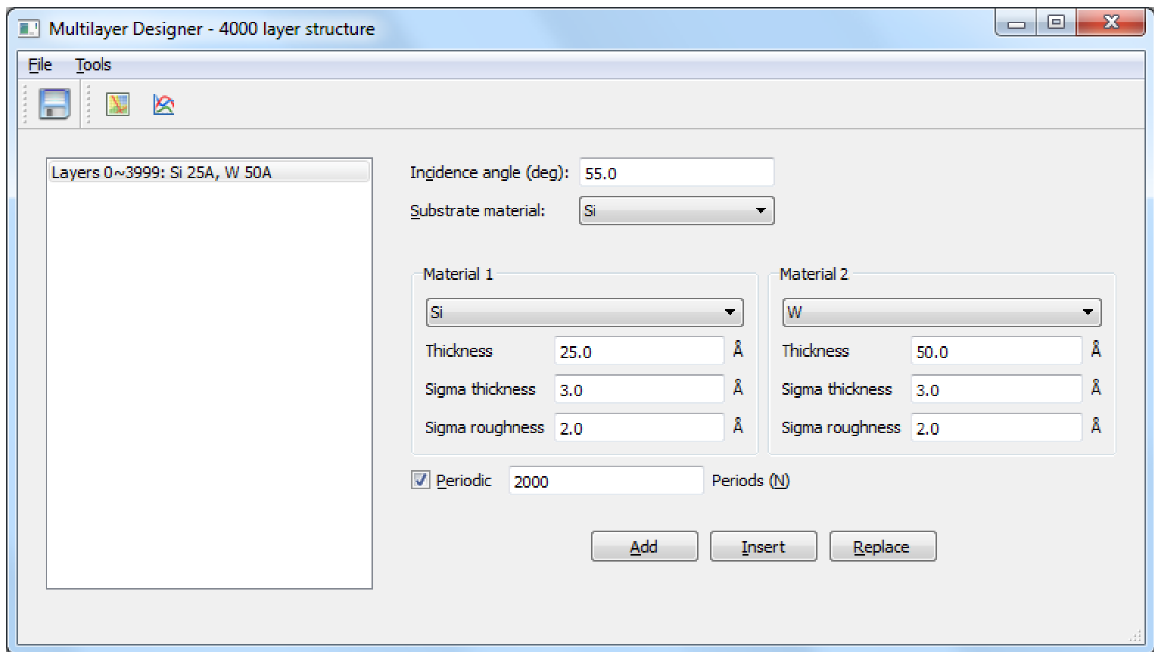


Figure 7-6. The multilayer designer.

The last component currently implemented is the Kirkpatrick-Baez (KB) mirror, which also functions as a focusing optic. Its configurable parameters are its coating material and thickness, its length, the designed grazing incident angle, the focal length, and misalignment angle.

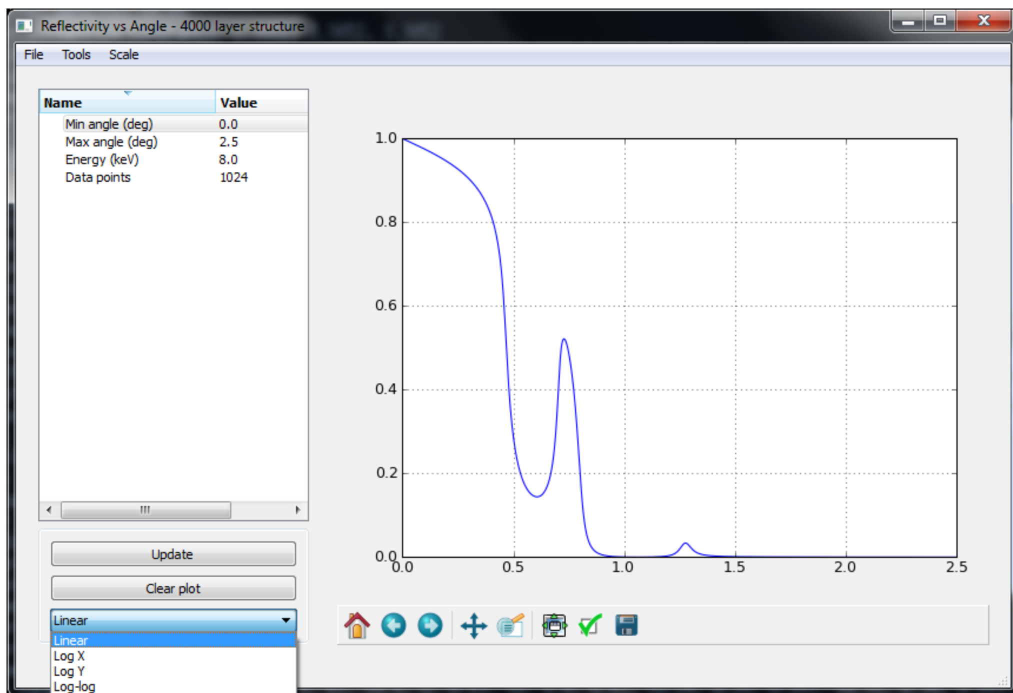


Figure 7-7. A reflectivity versus angle plot of a 2000 layer multilayer.

Additional Tools

There are two additional utilities included in the simulator package. The first is a multilayer reflectivity simulation, which shares the same interface as the multilayer designer in the simulator (Figure 7-6). Multilayers can be created outside of a project and their reflectivity versus angle (Figure 7-7) and displacement versus position (Figure 7-8) plots can be viewed. The second tool is the Bragg angle utility (Figure 7-9).

Figure 7-8. A sigma displacement versus position plot for a 2000 layer multilayer.

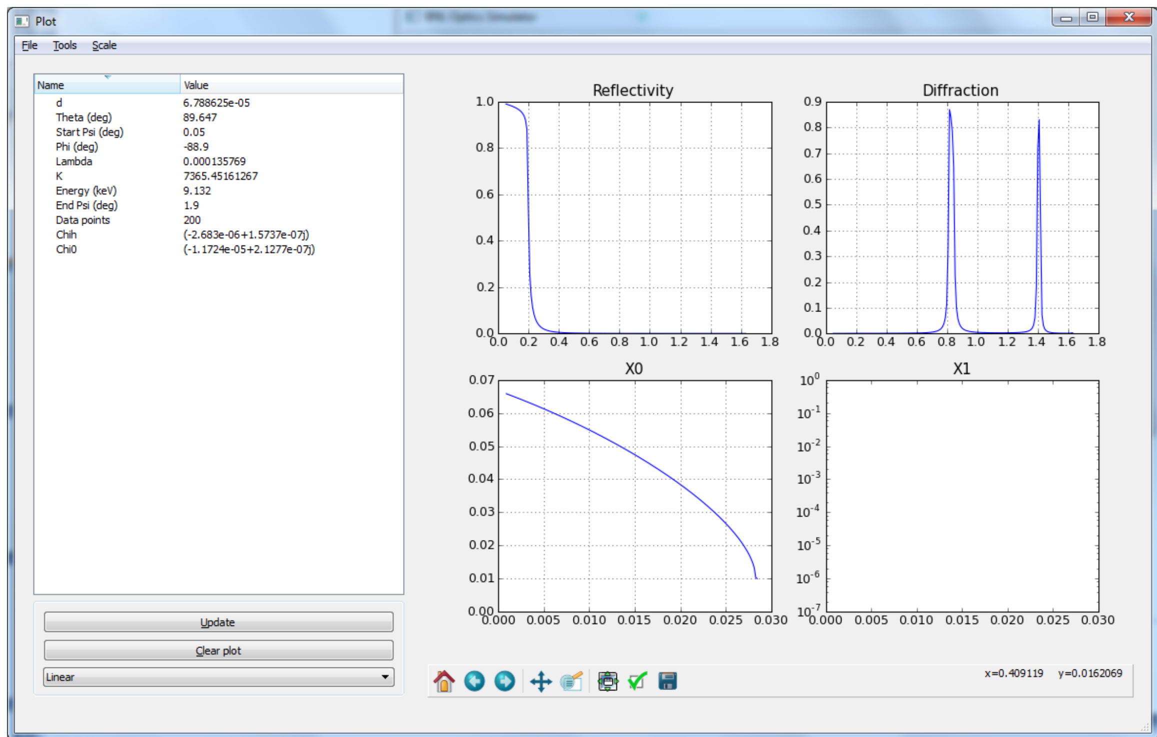


Figure 7-9. The Bragg diffraction utility.

References

"Matplotlib: A 2D Graphics Environment", J. D. Hunter, Computing in Science & Engineering, Vol. 9, No. 3. (2007), pp. 90-95

References

"Characterization of a multilayer Laue lens with imperfections", H. Yan, H.C. Kang, J. Maser, A.T. Macrander, C.M. Kewish, C. Liu, R. Conley, G.B. Stephenson, Nuclear Instruments and Methods A, Volume 582, Issue 1, (2007), pp. 126-128

"Film stress studies and the multilayer Laue lens project", C. Liu, R. Conley, A.T. Macrander, Proc. SPIE 6317, 63170J, (2006).

"Matplotlib: A 2D Graphics Environment", J. D. Hunter, Computing in Science & Engineering, Vol. 9, No. 3. (2007), pp. 90-95

"Metrology of multilayer Laue lens structures by means of scanning electron microscope imaging" N. Jahedi, R. Conley, B. Shi, J. Qian, K. Lauer, A. Macrander, Nuclear Instruments and Methods A, 616, pp. 89-92, (2010)

"Multilayer Laue Lens Growth at NSLS-II", R. Conley, N. Bouet, K. Lauer, M. Carlucci-Dayton, J. Biancarosa, L. Boas, J. Drannbauer, J. Feraca, L. Rosenbaum, Submitted to XRM-2010 proceedings (2010)

"Nanofocusing of Hard X-rays with Multilayer Laue Lenses", A. Macrander, H. Yan, H.C. Kang, J. Maser, C. Liu, R. Conley, G.B. Stephenson, Handbook of Optics, 3rd Edition, Optical Society of America , Volume V, Chapter 42, (2010)

"Nanometer linear focusing of hard x-rays by a multilayer Laue lens", H.C. Kang, J. Maser, G.B. Stephenson, C. Liu, R. Conley, A.T. Macrander, S. Vogt, Phys. Rev. Lett. 96, pp. 127401, (2006).

"Pressure-dependent transition from atoms to nanoparticles in magnetron sputtering: Effect on WSi2 film roughness and stress", L. Zhou, Y. Wang, H. Zhou, M. Li, R. L. Headrick, K. MacArthur, B. Shi, R. Conley, A. T. Macrander, Phys. Rev. B 82, 075408, (2010)

"Profile coating and its application for Kirkpatrick-Baez mirrors", C. Liu, L. Assoufid, R. Conley, A.T. Macrander, G.E. Ice, J. Z. Tischler, Opt. Eng, 42, pp. 3622-3628 (2003).

"Profile coatings and their applications", C. Liu, R. Conley, A. T. Macrander, L. Assoufid, J. Vac. Sci. Tech. 21, pp. 1579-1584, (2003).

"Pt KB mirrors for hard X-ray microfocusing system made by profile coating", B. Shi, C. Liu, J. Qian, W. Liu, L. Assoufid, R. Conley, A. T. Macrander, Proc. SPIE [7802-15], Accepted, (2010)

"Sectioning of multilayers to make a multilayer Laue lens", H.C. Kang, G.B. Stephenson, C. Liu, R. Conley, Khachatryan, M. Wieczorek, A.T. Macrander, H. Yan, J. Maser, J. Hiller, and R. Koritala, Rev. Sci. Instrum., 78 (4) (2007) 046103.

"Small d-spacing WSi2/Si narrow bandpass multilayers", C. Liu, R. Conley, A.T. Macrander, T. Graber, C. Morawe, C. Borel, E.M. Dufresne, Proc. SPIE 5537, pp. 154-160, (2004).

"The NSLS-II multilayer Laue lens deposition system", R. Conley, N. Bouet, J. Biancarosa, Q. Shen, L. Boas, J. Feraca, L. Rosenbaum, Proc. SPIE Vol. 7448 74480U., (2009)

"Transition from Atoms to Nanoparticles in Magnetron Sputtering", L. Zhou, Y. Wang, H. Zhou, M. Li, R. L. Headrick, K. MacArthur, B. Shi, R. Conley, A. T. Macrander, arXiv: 1005:2708, (2010)

"Wedged Multilayer Laue Lens", R. Conley, C. Liu, J. Qian, C. Kewish, A.T. Macrander, H. Yan, H.C. Kang, J. Maser, G.B. Stephenson, Rev. Sci. Instrum. 79 053104, (2008)