



HAL
open science

In-line etching process control using dynamic scatterometry

S. Soulan, M. Besacier, T. Leveder, P. Schiavone

► **To cite this version:**

S. Soulan, M. Besacier, T. Leveder, P. Schiavone. In-line etching process control using dynamic scatterometry. Modeling Aspects in Optical Metrology, 2007, Munich, Germany. pp.11111, 10.1117/12.726197. hal-00168934

HAL Id: hal-00168934

<https://hal.science/hal-00168934>

Submitted on 30 Aug 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

In-line etching process control using dynamic scatterometry

Sébastien SOULAN^a, Maxime BESACIER^a, Tanguy LEVEDER^b, Patrick SCHIAVONE^a

^aLTM - CNRS, 17 av. des martyrs, 38054 Grenoble Cedex, France

^bCEA - LETI, 17 av. des martyrs, 38054 Grenoble Cedex, France

ABSTRACT

In-line process control in microelectronics manufacturing requires real-time and non-invasive monitoring techniques. Among the different metrology techniques, scatterometry, based on the analysis of ellipsometric signatures (i.e stokes coefficients vs. wavelength) of the light scattered by a patterned structures, seems to be well adapted. Traditionally, the problem of defining the shape and computing the signature is dealt with modal methods and is called *direct problem*. On the opposite, the *inverse problem* allows to find the grating shape thanks to an experimental signature acquisition, and can not be solved as easily. Different classes of algorithms have been introduced (evolutionary, simplex, etc.) to address this problem, but the method of library searching seems to be the most attractive technique for industry. This technique has many advantages that will be presented in this article, however the main limitation in real-time context comes from the short data acquisition time for different wavelengths. Indeed, the lack of data leads to the method failure and several database patterns can match the experimental data. In this article, a technique for real time reconstruction of grating shape variation using dynamic scatterometry is presented. The different tools to realize this reconstruction, such as Modal Method by Fourier Expansion, regularization technique and specific software and hardware architectures are then introduced. Results issued from dynamic experiments will finally illustrate this paper.

Keywords: Scatterometry, Real time, Process monitoring, Regularization, Tikhonov

1. INTRODUCTION

A part of future advances in microelectronics is conditioned by the capability for industry to monitor the manufacturing process of their chips. Since the dimensions of transistors become smaller and smaller, an important effort has to be made in the field of metrology : CD (*Critical Dimension*) metrology techniques must be able to characterize very short dimensions with a good accuracy. AFM (*Atomic Force Microscopy*) and CDSEM (*CD Scanning Electron Microscopy*) techniques are suitable to reach these accuracies but are invasive or time consuming. Actually, they are not well adapted to the industry requirements.

Another path in the metrology improvement is the development of a technique able to follow the evolution of a pattern during a dynamic process. It means that the metrology tool must be placed in situ, the time of acquisition of data must be very fast and this technique must be non invasive. This objective is not a simple task but first experimental tests can be carried out for simple processes (as described in this paper).

Based on the ellipsometric measurement of light diffracted by a grating, the scatterometry technique appears to be promising for in line process monitoring.

1.1. Principle of scatterometry

Scatterometry stands for *metrology of scattered light* : a ray of polarized light is sent on a periodic structure (the grating) and the polarization state of the scattered ray is collected (Fig. 1). Here we collect only the zero-th order and the metrology device (the scatterometer) is kept static when the incident wavelength is varying.

Changes in polarization state are symbolized by ρ or ψ and Δ and these parameters are function of the grating shape geometry, the periodicity and the materials.

$$\rho = \frac{r^{TE}}{r^{TM}} = \tan(\Psi)e^{j\Delta} \quad (1)$$

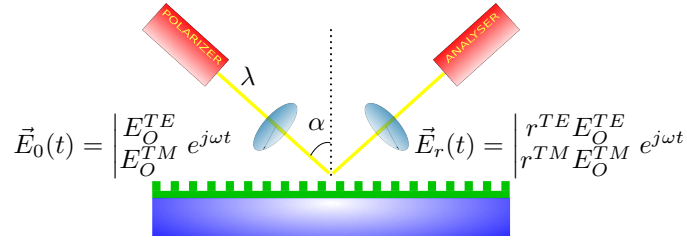


Figure 1. Principles of scatterometry

Practically, we use the two signals S_1 and S_2 .

$$S_1 = \sin(2\Psi)\cos(\Delta) \quad (2)$$

$$S_2 = \cos(2\Psi) \quad (3)$$

This technique needs only a single ray of light, so that it can then be considered as non-invasive and thus, perfectly compliant with in-line process-monitoring.

Notice that here we deal with *spectroscopic* scatterometry : the incident angle is constant meanwhile we make the wavelength λ vary.

1.2. Direct and inverse problems

Solving the *direct problem* is computing the changes in polarization state (S_1 and S_2) for a well defined grating shape : this is usually performed using a rigorous electromagnetic code which is presented in section 2. The plot of S_1 and S_2 vs. λ is a *scatterometry signature*. Such a signature is shown in figure 2.

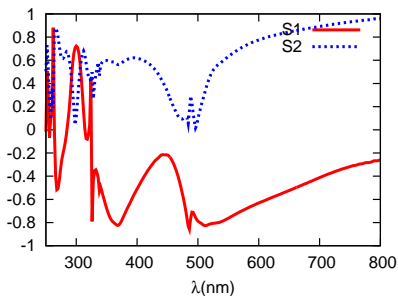


Figure 2. Scatterometry signature

The inverse problem, far more complex, is to guess the grating shape from a signature. This signature can be acquired from a scatterometry device (ellipsometer) for example. Different classes of optimization algorithms have been used (evolutionary, simplex, etc.),¹ but the method of library searching is practically the most commonly used in the industry. Moreover, this method is real-time compliant as the solving time is perfectly predictable.

The work presented in this article deals with improvements made on the method of library for the case of dynamic scatterometry. A signature library is built-up using different combinations of profile parameters (CD, height, sidewall angle, etc.) and the experimental signature is compared with the library for the best match. Although generating the library requires a lot of computation time, solving the inverse problem can be fast because pattern matching is reduced to a simple numerical database search. However, in the real time context the constraint is a poor resolution in wavelength. Indeed, in order to follow in real time the process evolution, a high frequency data acquisition is required, that makes the device to yield poorly-resolved and

imprecise signatures. This poor resolution is problematic : more than one database signature can match to the experimental data.

In this paper, we present a technique which reconstruct the profile shape variation using frequently acquired signatures despite their low resolution. The main idea is to achieve a first and complete signature measurement for the whole scale of wavelength at $t = 0$ before processing, and partial signatures at next steps to intelligently refine the library search using Tikhonov time regularization. This technique assumes that only a smooth variation of the shape over time is realistic. This approach is implemented as a general C++ architecture dedicated to scatterometry. The system integrates multiprocessing (shared memory) capability for high-speed signature generation and the use of the Graphic Processing Unit (GPU, video card processor) as a numerical database accelerator for the library-based inverse problem. Due to the inherent vector architecture of the GPU, the entire library search operation can be made in only a few clock cycles.

In a short first part, a brief description of the tools used to solve the direct problem are presented. The second part deals with the real time regularization principles and the two next parts (third and fourth) describes the software and the hardware tools. Finally, first results are shown with a simulated example followed by an experiment where the patterned resist shape evolution is followed while reflowing.

2. MMFE FOR THE DIRECT PROBLEM

As mentioned in the introduction part, the direct problem is used to establish simulated signatures from a given topography. It allows building the data library with parameter variations. Some modelling tools are suitable to address the direct problem. In this work, the Modal Method by Fourier Expansion is chosen. A full description of this method also known as R.C.W.A. (Rigorous Coupled Wave Analysis) can be found elsewhere² and will not be detailed here. Only a brief description is given.

Modal Method by Fourier Expansion MMFE is based on a simple approach where Maxwell equations are solved thanks to a matrix system. TE or TM polarization are treated using the same formalism and an improved version allows to obtain good results for TM polarization in terms of convergence speed. This method implicitly considers periodic structures. The solution of Maxwell equations is reduced to the solution of an algebraic eigenvalue problem in the discrete Fourier space. The derivation of the matrix operator involves two steps: the electromagnetic field is expanded into Floquet-Fourier series, the permittivity of the materials which appears in the Maxwell equation is here a periodic function and is expanded into Fourier series.

The topography of the structure which is studied is cut in different layers. These different layers are linked by a S matrix system which accounts for the boundary condition providing a good convergence even for thick or absorbing layers. In this rigorous method, the only approximation is due to the number of modes retained for the truncation of the modal expansion. We must remark that m must not be chosen too large : over $m = 10$, computation time is prohibitive and systematic computing errors make the precision fall down.

3. REAL-TIME REGULARIZATION

Dynamic scatterometry can not be reduced to a succession of static problems : a high frequency acquisition and a short acquisition time (regarding to the process characteristic time) are required to follow the shape variation. Thus, standard ellipsometers that usually gives full defined signatures in seconds are not adapted. However, we can use multi-channel ellipsometers : ellipsometric signals can be acquired quickly but for only a reduced set of wavelength. The number of wavelength being small, a mere static scatterometry resolution is not sufficient. Nevertheless, the grating shape can be reconstructed taking into account the whole set of partial signatures over time and the regularity (smoothness) of the shape parameters evolution (height, CD, side wall angle, etc.).

The principle of regularization can be summarized as a balance for the choice of the best parameters. This balance is made between, on one hand, the proximity of each acquired signature to the chosen simulated signature in the library and on an other hand, the regularity of the parameters variation over time.

Practically, we proceed as such :

- at $t = 0$, i.e. before the dynamic process, the grating shape is supposed to be perfectly known. Static scatterometry is used and a high resolution signature is acquired. There is no problem of computing time at this stage.
- then, at each subsequent time step, a low-resolution (and low-precision) signature is acquired. This implies bad accuracy in the database look up. The technique of regularization then proceeds in two steps:
 1. We do not pick up only the Nearest Neighbor (NN) of the acquired signatures in the database but the 4 nearest neighbours. So, even if the first NN is far to be the good solution (it can be an artifact), we assume that a better solution can be present in the 4 NN. In fact, the algorithm selects at each time step (and relatively to the previous time steps) the best set of parameters.
 2. A standard time-based Tikhonov regularization³ is then computed : this minimization problem is solved independently for each parameter describing the grating shape (height, Critical Dimension, sidewall angle...)

$$\min_X \left(|X - X_{exp}|^2 + \beta \int_{window} \left(\frac{\partial X}{\partial t} \right)^2 dt \right) \quad (4)$$

The goal of such two-step regularization is to have, for each parameter describing the grating shape, a smooth and realistic variation. Indeed, the minimization problem above (equation 4) consists in finding the appropriate weighting between the proximity to the acquired data X_{exp} (first term) and the smoothness of the parameter (second term).

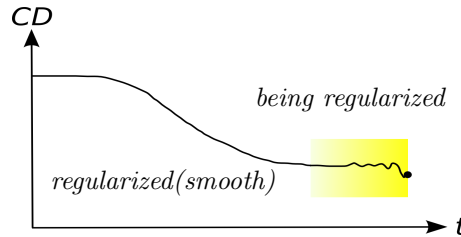


Figure 3. Regularization is made inside a window sliding on time

As shown in figure 3, the regularization has to be made into a time-sliding window in order to limit computing resource. Indeed, a $N \times N$ linear system has to be solved for each parameter regularized within a N timesteps window.

The advantage of using the principle of Tikhonov (i.e. balance with regularity) is that we only have two parameters to tune : one for the library search method and a second, β , for the minimization problem above.

4. A NEW A.P.I. DEDICATED TO SCATTEROMETRY

4.1. Introducing libScattero

A part of our work for scatterometry techniques development is made in computing science; there was a need for a coherent software that will allow us to efficiently acquire, simulate and exploit scatterometry data. Thus, we have developed a generalist framework fully dedicated to scatterometry. This software is designed to :

1. to provide a set of algorithms for:
 - an easy modeling of a grating shapes (and grating shape variations)
 - direct problem computation using our MMFE engine⁴ (computation of signatures and libraries)

- inverse problem solving in static and dynamic context, i.e. real time pattern shape reconstruction.
2. have an object-oriented architecture which make it easy to add new algorithms and to extend features.
 3. be user-friendly : A coherent and full-documented A.P.I. (Application Programming Interface) for C++ and Python is provided and a Graphical User Interface is planned.
 4. use, whenever it is possible, the most advanced high-performance computing technology.

4.2. Architecture

For a user point of view, `libScattero` is a set of six classes originally coded in C++. To this object-oriented architecture we must add the MMFE core, coded in pure Fortran 77, which is heavily called for signature computation.

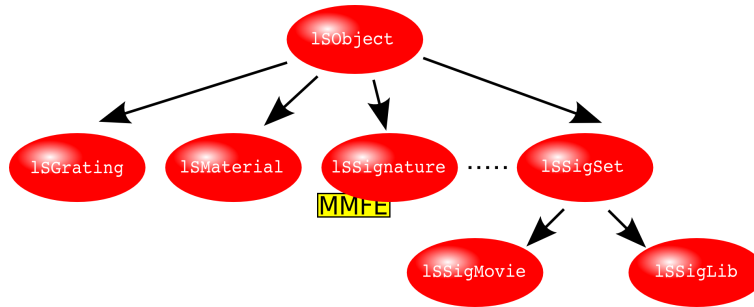


Figure 4. `libScattero` object-oriented architecture

Without entering into details, the role of these seven classes are :

<code>lSObject</code>	Main class, provides consistency to <code>libScattero</code> architecture
<code>lSMaterial</code>	Related to a material : name, optical index (depending on wavelength and/or temperature), XML data file management.
<code>lSGrating</code>	Related to the modelling process of a grating : geometry (parametrized data), materials collection. This class includes the methods for running the MMFE engine to compute a single signature or a full library.
<code>lSSignature</code>	Related to an ellipsometric signature.
<code>lSSigSet</code>	Data management (memory, files) of a big set of signatures.
<code>lSSigMovie</code>	(inherits <code>lSSigSet</code>) During experiment time, an object of this class can collect data and solve inverse problem dynamically. This class contains the real time regularization algorithms.
<code>lSSigLib</code>	(inherits <code>lSSigSet</code>) Stores a library of signatures. GPU computation techniques are integrated therein.

4.3. Using `libScattero`

We will demonstrate the use of this framework to solve a dynamic inverse problem whose results are presented in part 6.1 ; the grating shape is modeled by its height (H), its width(CD) and the sidewall angle (ANGLE).

1. Library computation To produce a 1024 signatures library calculated with a grating decomposed in 18 slices (20, the first argument of `INIT` includes substrate and air), no more than the following script-file is necessary:

```

# Parameters
SET L 400
SET CD 160.32 1.28 201
SET H 160.32 1.28 201
SET ANGLE 9.5 0.7 20.1
# Building instructions
INIT 20 L CD H
SLOPE ANGLE

```

In a C++ code, to run the computation, you just have to load the `lSMaterials` objects and to call the `lSGrating::computeLibrary("your_file",...)` method (passing a scriptfile name in argument).

2. Solving the dynamic inverse problem While a movie is being acquired from the ellipsometer, the data is managed by the `lSMovie` class. Solving the inverse problem using the GPU for instance (see 5) is made using the methods :

```

lSSigMovie::set_method(LIB_GPU4)           enables the GPU for solving inverse problem
lSSigMovie::attach_lib(my_lssiglib_object) in fact, loads the GPU with a signature library
lSSigMovie::regularize(...)              regularizes in the sliding window

```

Then, the full regularized data is accessible as a member of your `lSSigMovie` object.

4.4. Conclusion on libScattero

The software framework `libScattero` is still under development but it is nevertheless perfectly usable for our researches in scatterometry. However, some efforts remains to do in the development of an efficient graphical user interface and on software packaging for distribution.

The next part of this paper will deal with a particular point of our software development : the research in high-performance computing (the fourth goal of `libScattero`) using the Graphics Processing Units.

5. HIGH PERFORMANCE COMPUTING

We have said that the library search method is a good method to solve the inverse problem taking into account the "real-time" constraint. But the libraries used for this method are often large : about half a GByte is a common size in industrial problems. Such a large quantity of data implies two challenges :

- The computing time of this library should be improved at maximum; on a single (but powerfull : Xeon 3GHz) processor, such a computation can last several weeks.
- Once computed, dealing with this data for quick library look-up is also a challenge : finding the Nearest Neighbor of an acquired signature inside a huge signature library should be made fast enough comparing to acquisition frequency of experimental data.

For the first point, we are working to make the code that computes libraries run on multiprocessor systems. Since a library is a collection of signatures that can be computed independently, the parallel efficiency is close to 100%. We have verified that on a two dual-core Intel Xeon computer, using an OpenMP library.

For the second point, we use as a co-processor the GPU⁵(Graphics processing Unit) of a graphic card. Computer game industry has a growing need in 3D graphics capability and nowadays, video game picture rendering is made by a special stream processor, very powerfull but very different from standard CPUs. Indeed, GPUs are said to be stream processors or S.I.M.D. (*Single Instruction, Multiple Data*) processors because of their ability to compute a whole set of data (like a picture) with a single instruction. Contrary to CPUs, who, in their simplest form compute one single data by instruction. (figure 5)

Using this vector processor is a bit tricky but can be very efficient for certain classes of problems, the vector architecture however makes it perfect for library look up. In summary, the main advantages are :

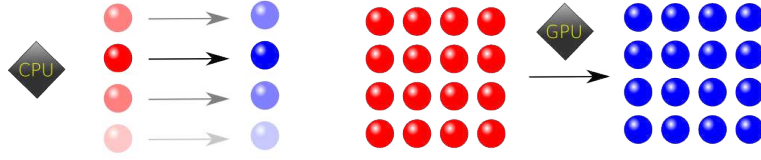


Figure 5. CPUs. vs. GPUs (S.I.M.D.)

- For the problem of large library look-up, GPU is far faster than CPU. The table 1 shows the computation speed-up using GPUs over CPUs for different number of wavelength and signatures. These results are obtained with the NVidia GeForce 8800GTX GPU.⁶ For large signature libraries (400 - 700 MB), GPUs make the work about 10 times faster and sometimes more.

$N_{signature} \backslash N_{\lambda}$	5	10	15	25	50	100	150	200	250
16384						1.18	1.48	1.7	1.9
65536				1.5	2.61	4.31	5.64	6.41	7.18
262144	1.08	1.84	2.60	3.95	6.11	10.21	16.37		
1048576	1.70	2.92	4.04	6.18					

Table 1. Computation speed up using GPUs vs. using CPUs.

- The power of GPU is increasing much faster than CPU's. (see figure 6)

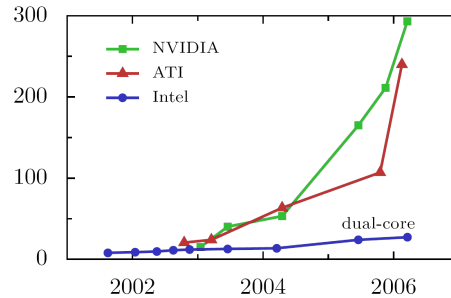


Figure 6. GPUs power in here GFLOPS increases far faster than CPUs power.⁵

- The main CPU and memory are discharged from such computation and can be used for Tikhonov regularization.
- A graphic card with the most powerful GPU of the moment only cost 700\$...

GPU processors are made for graphics. If we want to compute data with, we need to consider the signature database as a collection of big pictures in video memory. And computing in database (for instance, the calculation of the distance between the acquired signature and the pictures in the database) can be seen as a visual effect to be applied to the plan where a picture is mapped. The visual effect is made by a lot of sub-processors inside the GPU core called *shaders*. In recent GPUs, shaders are programmable and we use this to develop a technique for multi-NN search. This technique is based on the work of Benjamin Bustos et al.⁷

An important point is that using GPUs allow us to reconstruct grating shape *in time* ; fast enough compared to scatterometer acquisition frequency. Using GPUs is a way to manage 10 or 20 signature frames per second while regularizing.

6. RESULTS AND COMMENTS

Here we show two examples of application for dynamic scatterometry where real-time regularization is used. The first one is for demonstration purpose : we solve the inverse problem on simulated data (these simulated data are actually solution of the direct problem). The second example uses a real set of scatterometry signatures coming from a 32 way ellipsometer. The process we want to monitor is a reflow of patterned resist.

6.1. Results for simulated data

In this simulation, we want to monitor a grating shape whose geometry is modeled by three parameters : the height H , the CD (*critical dimension*, the width) and α the side wall angle (figure 7).

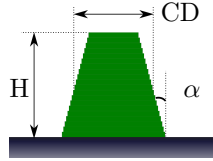


Figure 7. grating shape modeling with 3 parameters

Simulated movie Using our MMFE engine,⁴ we simulate an "ellipsometric movie" of 100 frames (the process control lasts 10s and we use 10 frames per second). During this movie, the values of the three parameters are supposed to vary as described in the figure 8 :

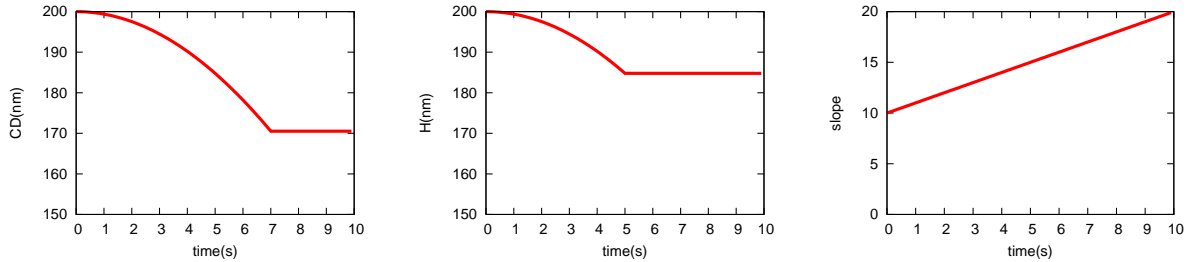


Figure 8. The 3 parameters evolution over time

Voluntarily, the CD and H parameters variations includes successively a smooth part, a corner and then a flat part. This is a way to assess the efficiency of regularization.

For the signature computation using the MMFE code, each grating is modeled as a stack of 10 resist layers, lying on a silicon substrate. The angle of incidence of the light to be diffracted is set to 70° . The truncation order for Fourier expansion is set to 10. These settings will be the same for the library computation.

Once the movie is generated, we add a noise to each frame (to each signature),

$$\begin{aligned} S'_1 &= (1 + \theta_1)S_1 \\ S'_2 &= (1 + \theta_2)S_2 \end{aligned}$$

where $\theta_1, \theta_2 \in [0; 0.1]$ (5%) in order to take into account all the uncertainties due to modelization, measure and computation. An example of a noisy signature is shown in figure 9.

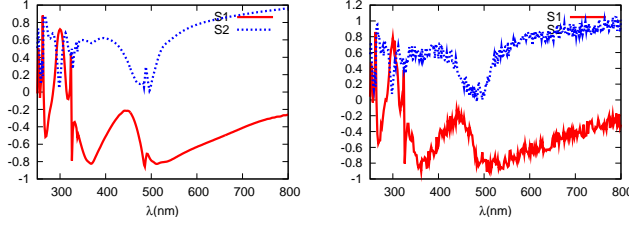


Figure 9. Example of noisy signatures compared to the original one

Library We choose to generate a signature library corresponding to the three-dimensional parameter space. Values taken by the parameters are :

$$\begin{array}{lll}
 CD = 160.32 & \text{to} & CD = 200 \quad \text{by steps of} \quad 1.28nm \\
 H = 160.32 & \text{to} & H = 200 \quad \text{by steps of} \quad 1.28nm \\
 \alpha = 9.5 & \text{to} & \alpha = 20 \quad \text{by steps of} \quad 0.7^\circ
 \end{array}$$

The choice of parameter range and density is the most tricky part in library computation : the library size in memory is proportional to the number of parameters combination. So, a good modeling work is necessary, first, to maintain the dimension of parameter space as low as possible, and then to restrict the number of parameters values.

Here we have $32 \times 32 \times 16 = 16,384$ signatures in the library. Each signature represents 16 wavelengths. Library size in memory, given by :

$$S_{Bytes} = N_{signature} \times 2N_\lambda \times \text{sizeof(float)}$$

is here of 4 MBytes.

Dynamic reconstruction The last step of this simulation is to run the program with simulated frames from the movie. You can see in figure 10 the geometrical parameters set as it has been found using step-by-step static scatterometry using a mere library search.

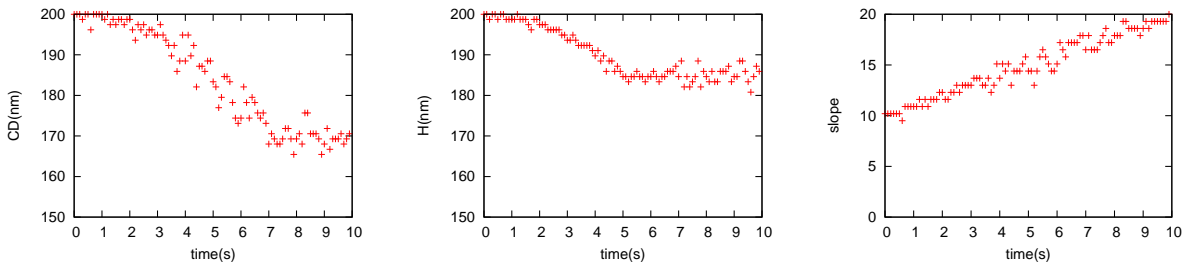


Figure 10. Parameters reconstruction using raw library search

Due to the added noise, geometrical parameters appear to be most of the time irregular and sometimes false. The best match in library search is not necessarily the real solution : when parameters density is poor (mainly because the modeling is difficult and thus, parameter space dimension is important); the solution vector \vec{X} (equation 4) of the three parameters can be close to real shape only by some of its components. It means that the raw library search method could privilege a solution with good height and CD and insignificant α instead of a more significant solution where the set of the three parameters would be closer to the reality.

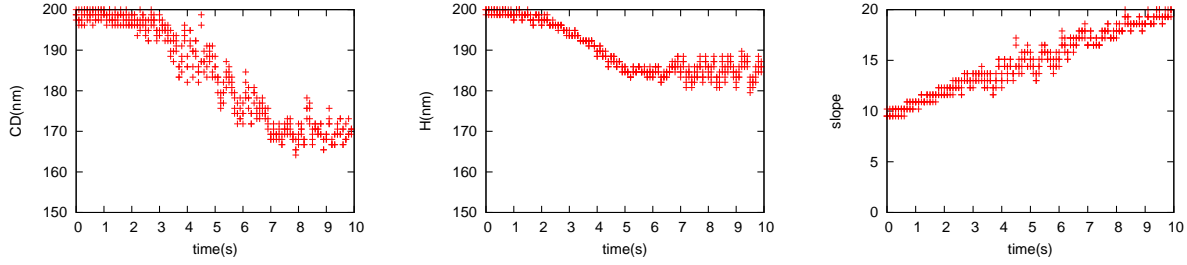


Figure 11. Parameters reconstruction using 4-NN library search

Picking-up the 4 nearest neighbors of each acquired signature (each frame) expands the field where realistic solution can be found. Such expansion is visible on figure 11.

Refining again with the Tikhonov³ time-based regularization gives the figure 12

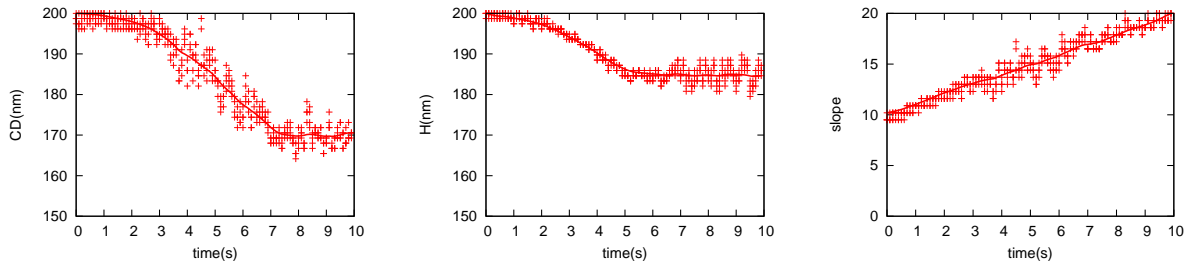


Figure 12. Parameters reconstruction once regularized

Then a comparison with "original" simulated parameters gives the figure 13 Notice that the non-regularized

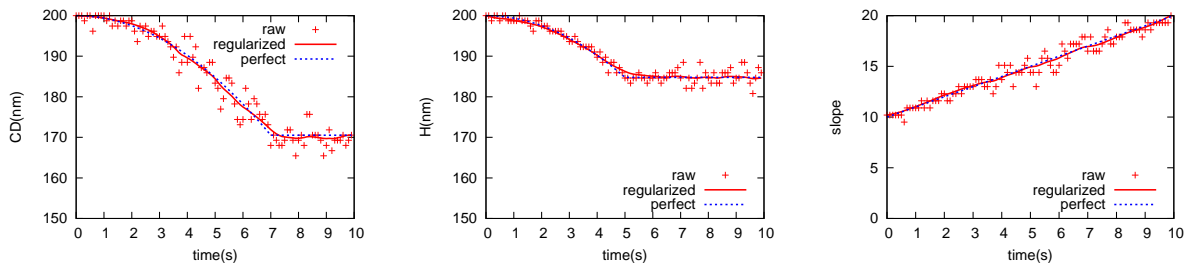


Figure 13. Parameters reconstruction : before/after comparison

data are sometimes far from the solution, it proves the need for regularization to reconstruct a realistic shape variation.

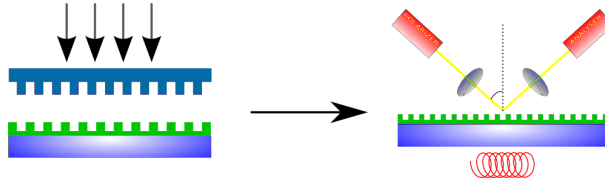
This regularization underlines a problem for realistic reconstruction when the first derivative is supposed to be discontinuous : corners are softened. This can not be avoided but can be attenuated either by taking β lower as possible (in eq. 4), or by increasing the frequency of acquisition.

We can see here the fundamental difference between the two steps of the shape reconstruction : the first one (the library search using 4 nearest neighbours) does not make regular a single parameter in itself (it can even be less regular than before) but regularizes the whole set of parameter.

6.2. Resist pattern reflow

This experiment, originally carried out for resist viscosity study is here a good way to assess the validity of our algorithms.

A nano-imprinted grating, made of resist on silicon is heated on a hot plate while scatterometry signatures are acquired. The 32 wavelengths ellipsometer is from Horiba/Jobin Yvon. The temperature is increased until reflow of the resist ($\sim 90^\circ\text{C}$). We try to watch the pattern shape variation during reflowing with an acquisition frequency of about 1 signature per second.



First, we chose to model the pattern shape using a mere sine wave : the amplitude of the sine function is used as the only variable parameter. A comparison between reconstructed profile shape and AFM measurements made in the same conditions and at different time step is shown in figure 14.

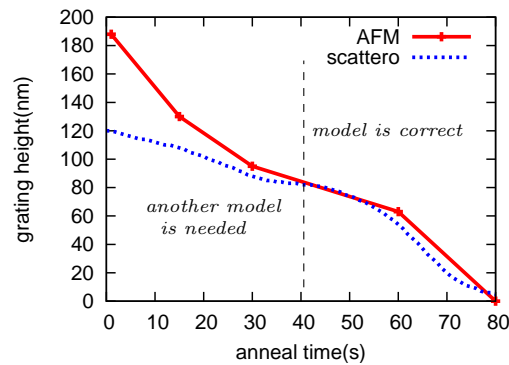


Figure 14. Comparison between AFM measurements and scatterometry with sinusoidal shape model

We can clearly see that there are two parts in figure 14 : A part, beginning at 40s where the simple sinusoidal model seems to work, and an other part, at the beginning of the reflow, where actually another model is required.

The main reason why the two curves do not match in the beginning is that the grating shape is rectangular before the reflow. This is shown in figure 15 where the AFM measurements of samples taken at different processing time are represented.

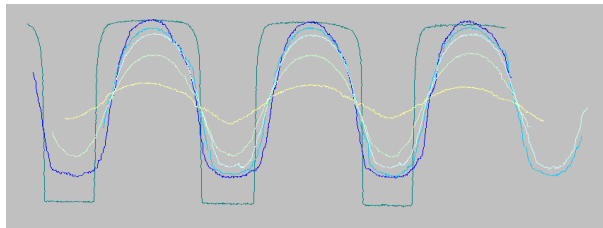


Figure 15. AFM measurements of the grating shape during reflow at different time steps

It means that a single library, based on the sinusoidal model is not sufficient to reconstruct the full shape evolution during reflowing. Another library, build for rectangular shapes with rounded corners would probably better fit the first part.

Nevertheless, this preliminary study shows that the choice of the model and the parameters associated are fundamental for the results validity ; when a good modeling is developed, standard scatterometry is efficient and dynamic shape reconstruction can work.

7. SUMMARY

Scatterometry is a metrology technique based on the polarization of the light diffracted by a grating. It is commonly used in microelectronics manufacturing for profile shape characterization : in practice, a huge set of pre-computed scatterometry signatures is compared to the acquired one (library based search method) and the best match is supposed to give the most relevant shape. In this paper we have demonstrated the capability of scatterometry for real-time process monitoring. Indeed, we have developed a way to counterbalance the bad accuracy in library search method due to by the real-time constraint. A new set of algorithms and innovative computer science techniques (GPUs) are here used to reconstruct a smooth and realistic grating shape variation over time. In the last part of this paper, we have shown, as a demonstration, two examples of such reconstruction : first with a pure simulated set of data, in which we added noise. And then, with true experimental data (resist reflow), where we finally faced the typical scatterometry problem of finding a good geometrical modeling for the grating shape.

8. CONCLUSION

The main point to conclude with is that is work is actually a "work in progress". It began with a view to provide real-time monitoring techniques for plasma etch using scatterometry. So far, our researches were made in algorithmics and computer science in order to build preliminary elements for real process monitoring. The results presented in this paper prove that our regularization scheme is working. However, an important effort remains to do mainly in these three ways :

- regularization algorithms improvement : GCV (Generalized Cross Validation) technique could be able to help us tuning the two regularization parameters.
- experimental validation : following a mere resist trimming process is the first step toward more complex plasma etching experiment.
- computer science : the computation of very large libraries is necessary but very time consuming. Thus, we are working for "parallel" solutions (clusters...).

REFERENCES

1. C. J. Raymond, M. E. Littau, A. Chuprin, and S. Ward, "Comparison of solutions to the scatterometry inverse problem," in *Metrology, Inspection, and Process Control for Microlithography XVIII. Edited by Silver, Richard M. Proceedings of the SPIE, Volume 5375, pp. 564-575 (2004).*, R. M. Silver, ed., pp. 564-575, May 2004.
2. T. K. Gaylord and M. G. Moharam, "Analysis and applications of optical diffraction by gratings," *Proceedings of the IEEE* **73**(5), pp. 894-937, 1985.
3. A. Tikhonov and V. Arsenin, *Solution of Ill-posed Problems*, 1977.
4. D. Fuard, C. Perret, V. Farys, C. Gourgon, and P. Schiavone, "Measurement of residual thickness using scatterometry," **23**(6), pp. 3069-3074, AVS, 2005.
5. I. Buck, N. Govindaraju, M. Harris, J. Kruger, A. Lefohn, D. Luebke, T. Purcell, and C. Woolley, "Gpgpu: General-purpose computation on graphics hardware," in *Course 32 at ACM SIGGRAPH*, 2004.
6. <http://www.nvidia.com/page/geforce.8800.html>.
7. S. H. Benjamin Bustos, Oliver Deussen and D. Keim, "A graphics hardware accelerated algorithm for nearest neighbor search," in *Computational Science - ICCS 2006*, P. M. S. Vassil N. Alexandrov, Geert Dick van Albada and J. Dongarra, eds., *LNCS 3994*, pp. 196-199, Springer, 2006.