

Shape-Based Optimal Estimation and Design of Curve Evolution Processes with Application to Plasma Etching

Jordan M. Berg and Nan Zhou
Department of Mechanical Engineering
Texas Tech University
Lubbock, TX 79409-1021
jordan.berg@ttu.edu

March 20, 2000

Abstract

This paper considers the problem of determining a finite number of discrete parameters appearing in a nonlinear partial differential equation describing a curve evolution process. The method is applied to the plasma etching of thin films for semiconductor manufacturing. Results are obtained within the mathematical framework of level set methods. Here, the evolution of the curve under study is captured through the evolution of a level set function. The underlying physics of the process are completely contained in a scalar function called the speed function. The degree of difficulty of treating the evolution equation depends on the functional dependencies of the speed function. This paper presents optimal estimation and design techniques based on analytical gradient computations for a class of position and orientation dependent speed functions. The technique is demonstrated on a plasma etching model taken from the literature. Only simulation results are presented here, but the model under study has been shown to reproduce experimental data with reasonable accuracy. In the estimation problem, parameters in the model are fit to best match the feature shape measured in experiments. In the optimal design problem, parameter values are selected to most closely attain a desired feature shape.

1 Introduction

Although processes in which shape is the controlled output or measured quantity are of considerable economic importance, there are relatively few tools available for their estimation and control. Etching and deposition of surface features for fabrication of microelectronic devices are an example of such processes. This application is the subject of the present work, though the techniques described are general. Here the feature shape at the end of the etch or deposition can be considered to be the output of the process. An estimation problem might involve inferring information about the underlying physics from the observed shape. A control problem might consist of determining optimal values for user-specified process settings in order to achieve a desired shape. In order to proceed with either objective, a

model is needed relating evolution of the shape to the process variables. Although there are extreme approaches to modeling of these processes, ranging from purely data-based methods on one hand, to extremely large scale simulation with no adjustable parameters on the other, this paper is concerned with techniques that combine functional models of the process with adjustable parameters that must be fit to data. This methodology is sometimes called “semi-empirical” or “physics-based” modeling.

In such an approach, broad knowledge of the process physics is used to construct a low-order model. A sampling of models may be found in [14, 15, 49, 23, 34, 36, 50, 53, 54]. These models are well-suited to real-time or iterative applications, but their usefulness depends on the ability of the process engineer to choose the parameter values correctly. The values may be selected based on surface evolution data taken from scanning electron micrographs of feature cross-sections. For the most part, this calibration process is *ad hoc*, and relies heavily on the modeler’s expert knowledge of the system; for examples see [19, 49]. To our knowledge, only one study has investigated methods to optimize this process systematically [14]. That work, while successful, used a non-geometric cost function that requires the user to select points in one-to-one correspondence on the actual and estimated surfaces. This selection introduces an arbitrary component of unknown significance into the procedure, and places an undesirable burden on the user. In the present work we apply a coordinate free cost function that eliminates this arbitrariness. Our ultimate goal is a fully automated procedure for fitting parameter values in process models, or selecting process control settings, that will be meaningful and functional over a wide range of morphologies. The key to success is selecting a cost function that does not depend on specifics of the nominal feature. Such a cost function may be obtained based on a *metric* for the space of shapes. Several choices of metrics have been proposed for the formalization of shape identification, and visual processing in general [37]. The one used here is the *template metric*, which is discussed further below.

This work uses level set methods to describe the feature evolution. Level sets are a robust and flexible mathematical framework for the representation of evolving curves and surfaces. Their chief advantages are the ability to transparently handle topological transitions such as splitting and merging, and their relatively tractable behavior even in the event of corner formation. A property of the first-order nonlinear partial differential equations (PDEs) governing many of the processes we consider is that corners can develop even when the initial surface is smooth. Sethian and Adalsteinsson apply level set methods to the simulation of feature development in a variety of semiconductor manufacturing applications [1, 45, 43], using numerical methods based on hyperbolic conservation laws [31, 38]. Theirs is by no means the only work in this area. Application of a variety of techniques to simulation of thin film patterning may be found in [15, 49, 22, 24, 28, 29, 27, 34, 35, 41, 46]. Further comparison of the various methods is beyond the scope of this paper. It seems clear that, in the right hands, each can be made to work reasonably well, and that each has its own strengths and weaknesses. Level set methods are a good fit with the optimization techniques presented here.

Numerous papers have appeared on real-time control of plasma etching and deposition [13, 21, 25, 32, 33, 40, 48, 52], but relatively few address events at the wafer surface, focusing instead on the bulk properties of the plasma. Those that do consider the wafer surface

typically consider only planar processes [16, 30, 51]. Earlier work of the first author and co-workers focused on developing techniques for *real-time* feature-level estimation and control of plasma etching [6, 7, 8]. The development of the gradient formulas used in the present work was begun in [4], and a limited application of these results was made to simple models of photolithographic development and plasma etching in [3]. The present work presents a complete scheme for computing gradients, and applies the results to a realistic plasma etching model, including such necessary effects as shadowing and non-uniform ion energy distribution. The model used was presented in a series of papers by Jurgensen, Shaqfeh, and co-workers [26, 46, 47]. They initially obtain a sputtering model that successfully reproduced experimental results from oxygen plasma ion-assisted etching (IAE) of silicon [46]. Ultimately they included isotropic etch and radical re-emission terms, producing a model that successfully matched sulfur hexafluoride IAE of silicon over a wide range of process parameters. The current paper incorporates the sputtering model and an isotropic term, but neglects re-emission. This is equivalent to the model of [47] with sticking coefficient equal to zero.

We also mention a large body of work on shape optimization, such as [2, 11, 17, 39, 42]. We note in particular the use of level set functions in [42], and the use of the template metric in [2]. The problem considered by these researchers is that of finding an optimal shape to minimize a given functional. Much of the challenge in this problem arises from the need to parametrize the space of all admissible shapes. In the present paper that difficulty is absent, since a model of the underlying process is considered known. Thus only those shapes which can arise from that model are considered. Work of more direct relevance to this paper has been reported in [9, 10, 55, 56]. Here the shapes are captured by relatively few parameters, such as spline coefficients, and a functional minimized with respect to those coefficients. However, the goals and methods of those works are quite different from the present paper. In the case of [9, 10], the functional to be minimized is not itself based on shape, as it is here. The work in [55, 56] addresses metalforming and the optimal design of die shapes, and is concerned with shape-based functionals. Notably, the template metric is used to characterize the distance between the obtained and desired parts. However, the technical issues addressed in those papers are driven by the use of the finite element method, and the equations of plastic deformation. In the present work finite differencing schemes are used, and the underlying PDE is the level set evolution equation.

2 Level Sets

In this paper we restrict our attention to curves in the plane. Most aspects of our approach carry over to three dimensions directly, however, a number of important problems may be treated effectively in two dimensions.

In the level set formulation an oriented curve evolving in time, $\mathcal{C}(t)$, is represented by the zero level set (ZLS) of a level set function (LSF) $\Phi(x, t)$. That is, $\mathcal{C}(t) = \{x \in R^2 : \Phi(x) = 0\}$. The orientation is recovered by defining the points at which the LSF is negative as the interior, and requiring that the unit normal point outward. $\mathcal{C}(t)$ evolves in time according to $\mathcal{C}_t = \tilde{\alpha}(s, t, \dots)\tau + \tilde{\beta}(s, t, \dots)\nu$, where τ and ν are, respectively, the unit tangent and outward

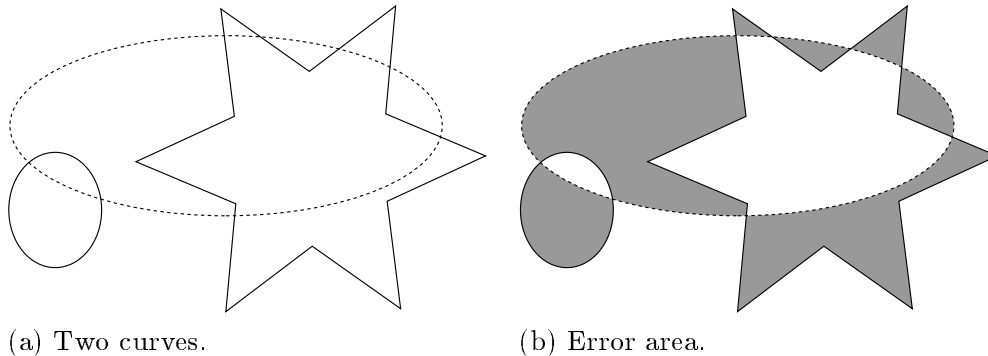


Figure 1: Two curves. One (dashed) is a large ellipse. The other (solid) consists of two simply connected components, a star and a small ellipse. The distance between the curves is found by summing the areas of the shaded regions.

pointing unit normal vectors. Now the evolution of the LSF corresponding to $\mathcal{C}(t)$ is governed by the PDE $\Phi_t + \beta(x, t, \dots)\|\nabla\Phi\| = 0$. Note that the component of the velocity along the tangent plays no part in the evolution of the LSF. Also note an important distinction between β and $\tilde{\beta}$. Namely, $\tilde{\beta}$ is defined only on $\mathcal{C}(t)$ itself, but β is a function on the entire plane. Of course on $\mathcal{C}(t)$, β must agree with $\tilde{\beta}$. For a derivation of this equation, and for more detail on its meaning when the LSF fails to be differentiable in space, see [43] and the references therein.

The process of parameter identification will require that we find the parameter values that give, in some sense, the closest match to an observed evolution. To make this rigorous, we must define a metric for LSFs that formalizes the idea of “distance.” A number of appropriate metrics are discussed by Mumford [37]. All these are of potential interest, but here we consider only the *template metric*. Figure 1(a) shows two closed curves, one consisting of a single connected component, the other consisting of two connected components. In all cases “inside” is taken to be the bounded component of the plane defined by the curves. Since the two curves are not identical, there should be a positive distance between them. We define this distance to be the area of the region between the two curves. This area is shaded light gray in Fig. 1(b). That is, given a pair of simple closed curves, \mathcal{C}_1 and \mathcal{C}_2 , let the distance from \mathcal{C}_1 to \mathcal{C}_2 , denoted $\rho(\mathcal{C}_1, \mathcal{C}_2)$, be the total area of points enclosed by either one curve or the other, but not both. In three dimensions or higher area is replaced by volume, etc. The definition is extended to any curve defined by a LSF on a compact domain by including appropriate portions of the boundary.

It is straightforward to compute the template metric when the estimate and the measurement curve are both characterized by LSFs. One way to make this calculation is to multiply the two LSFs pointwise. The result is a new LSF, which we call the *product LSF*. The ZLS of the product LSF defines a curve, and the area of the interior of this curve is exactly the error area. While forming the product LSF is a convenient way of visualizing the computation

of the error area, the actual method implemented is slightly different. To find the distance between the zero level sets of level set functions Φ and Ψ , we define $\Theta = \max(\underline{\theta}, -\bar{\theta})$, where, $\underline{\theta} = \min(\Phi, \Psi)$ and $\bar{\theta} = \max(\Phi, \Psi)$. Now it may be shown [4] that the desired distance is the area of the interior of Θ . We refer to Θ as the symmetric difference LSF. The symmetric difference method offers computational advantages over the product method [4]. Given Θ we compute the error area by generating \mathcal{C}_3 —the contour corresponding to the ZLS of $\Theta(x)$ (note that the contour is now composed of *oriented curves* and is no longer just a set)—and applying Green’s theorem, to obtain $\rho = \frac{1}{2} \int_{\mathcal{C}_3} \langle \mathcal{C}_3, \nu \rangle ds$, where $\langle \cdot, \cdot \rangle$ is the vector inner product, ν is the outward pointing unit normal to \mathcal{C}_3 , and s is the arc length parameter [18]. When convenient we use the notation $x \cdot y$ in place of $\langle x, y \rangle$.

Consider for a moment the following problem: Given an *objective curve*, \mathcal{M} , which is the ZLS of a LSF $\Phi(x)$, and a parametrized family of level set functions, $\Psi(x; \lambda)$ with ZLS $\mathcal{L}(\lambda)$, find the value of the parameter vector λ such that $\mathcal{L}(\lambda)$ is closest to \mathcal{M} . Denote the distance between the parameter-dependent curve defined by $\Psi(x; \lambda)$ and the objective curve defined by $\Phi(x)$ by $\rho(\lambda)$. Then $\rho(\lambda) = (1/2) \int_{\mathcal{E}} \langle \mathcal{E}, \nu \rangle ds$, where \mathcal{E} is the ZLS of $\Theta(x; \lambda)$, defined as in the previous section. In order to apply gradient-based optimization methods, we must compute ρ_i , where subscript i denotes partial differentiation with respect to λ_i . This problem was addressed in [4]. Briefly summarizing that work, we have for the smooth segments of the curve $\rho_i(\lambda) = \int_{\mathcal{E}} \langle \mathcal{E}_i, \nu \rangle ds$. To obtain an expression for $\langle \mathcal{E}_i, \nu \rangle$, consider the change in \mathcal{E} corresponding to a change in the i -th parameter λ_i . We write $\Gamma(X(s; \lambda); \lambda) = 0$. So, $\Gamma_i + \langle \nabla \Gamma, \mathcal{E}_i \rangle = 0$. In fact, differentiation with respect to a parameter gives the same form found in the evolution equation, namely $\Gamma_i + \langle \mathcal{E}_i, \nu \rangle \|\nabla \Gamma\| = 0$. This time, however, the terms are arranged as $\langle \mathcal{E}_i, \nu \rangle = -\Gamma_i / \|\nabla \Gamma\|$. Making this substitution, we eventually obtain the following expression [4], which may also be shown to hold when there are corners in \mathcal{L} :

$$\rho_i = - \int_{\mathcal{L}} \text{sign}(\Phi) \Psi_i / \|\nabla \Psi\| ds. \tag{1}$$

3 Parametrized Speed Functions

We return now to the problem of interest; that of estimating parameters in the speed function. The metric determining the distance between the obtained and target shapes is now $\rho(\lambda) := \rho(\Psi(T, x; \lambda), \Phi(T, x))$, with T a specified time, and $\Psi(T, x; \lambda)$ the solution at time $t = T$ to the evolution equation $\Psi_t + \beta(\cdot; \lambda) \|\nabla \Psi\| = 0$, subject to $\Psi(0, x; \lambda) = \Psi_0(x)$. Note that the λ dependence of Ψ is no longer due to an explicit parametrization, but arises from the variation of β with λ in the evolution equation. Thus in the event that analytic gradients are desired, they must be determined somehow from knowledge of β . One way to accomplish this is to differentiate both sides of the evolution equation with respect to the i -th parameter. The result is a PDE for the gradient, coupled to the original evolution equation. This type of *sensitivity equation* approach is described further in [9, 10].

We now restrict our attention to the plane, and consider speed functions of the form

$$\beta = \mu_1 \beta_0(x, y; \Lambda) + \mu_2 V(x, y; \Lambda) \cdot \nu \tag{2}$$

This restriction is made for convenience in the choice of algorithms for the forward solution of the evolution equation. The form (2) is sufficiently general to handle the plasma etching process considered below. The first term in (2) is an isotropic component, where the parameter-dependent isotropic etch rate β_0 is allowed to vary with position. In the second term, a parameter-dependent vector V forms an inner product with ν , the outward-pointing unit normal to the surface. Thus this term represents an orientation dependent component, where the orientation dependence can vary with position. With an eye towards our eventual chosen application, we refer to this term as the *sputtering* component. Note that the parameter vector λ has been partitioned into the parameters μ_1 and μ_2 , which determine the relative contributions of the isotropic and sputtering components, and the process parameters Λ . Recall that the outward-pointing unit normal is given by $\nabla\Psi/\|\nabla\Psi\|$. Thus the evolution equation corresponding to speed function (2) is $\Psi_t + \mu_1\beta_0(x, y; \Lambda)\|\nabla\Psi\| + \mu_2V(x, y; \Lambda)\cdot\nabla\Psi = 0$. This is a Hamilton-Jacobi equation with Hamiltonian $\mu_1\beta_0\|\nabla\Psi\| + \mu_2V\cdot\nabla\Psi$, which may be shown to be convex. The third component of this equation is a linear advection term, in which the initial data “flows” with velocity μ_2V . To treat the non-convex case the numerical update scheme must be adjusted, as in [43].

Now, take the derivative of the LSF evolution equation, exchange the order of differentiation, and denote sensitivity with respect to the μ or Λ by $S^{(q)}$. Then

$$S_t^{(\mu_1)} + (\mu_1\beta_0\nu + \mu_2V)\cdot\nabla S^{(\mu_1)} = -\beta_0\|\nabla\Psi\| \quad (3)$$

$$S_t^{(\mu_2)} + (\mu_1\beta_0\nu + \mu_2V)\cdot\nabla S^{(\mu_2)} = -V\cdot\nabla\Psi \quad (4)$$

$$S_t^{(\Lambda_i)} + (\mu_1\beta_0\nu + \mu_2V)\cdot\nabla S^{(\Lambda_i)} = -(\mu_1\beta_{0i}\|\nabla\Psi\| + \mu_2V_i\cdot\nabla\Psi) \quad (5)$$

where the subscript i denotes partial differentiation with respect to parameter Λ_i . In deriving these equations, repeated use is made of the relation $\|\nabla\Psi\|_q = (1/\|\nabla\Psi\|)(\nabla\Psi\cdot\nabla\Psi_q) = \nu\cdot\nabla S^{(q)}$. These are first-order linear equations, in which the sensitivity is advected with velocity $(\mu_1\beta_0\nu + \mu_2V)$ and sensitivity is “created” according to the various right-hand side reaction terms.

Solution of these sensitivity equations requires consideration of initial conditions and boundary conditions. The initial conditions are easily supplied. At $t = 0$ the level set function Ψ is exactly known, therefore all the sensitivities must be zero. The issue of boundary conditions is more interesting. Boundary conditions become necessary when the advection direction points into the computational domain from a boundary, since then the domain of dependence of the solution on the boundary lies outside the computational domain. Numerically this derivative should be computed using upwinding, which requires the use of a function value outside the computational domain. In theory one should specify the value of the function at those points. However, the nature of this problem allows a different approach, which we now describe. Note that these considerations hold true for both the nonlinear level set function evolution equation and the sensitivity equations.

For both the level set function and the sensitivities, the values obtained must be precise only in a small portion of the domain. In the case of the LSF, this is in a narrow band around the zero level set. In the case of the sensitivities, this is the same region, that is, where the LSF

is zero. This can be seen from (1), where the gradient of the metric is computed using a line integral involving the sensitivity over the zero level contour. Because errors associated with the boundary may propagate into the computational domain at the advection velocity, they may be prevented from influencing the solution by moving the boundary away from the free surface. This is not possible when the free surface intersects the boundary. In the examples that follow, we specify value periodic boundary conditions on boundaries that intersect the free surface. On boundaries away from the free surface, we instead hold the normal derivative constant. Numerically, we set the derivative normal to the surface equal to its *inward* finite difference value, even when the direction of advection would otherwise call for an *outward* difference approximation.

4 Numerical Implementation and Analytic Gradients

The curve evolution problems considered below are solved using the convex, first-order, scheme presented in [43]. Plasma etching models involving non-convex sputtering yields will require the use of a non-convex scheme instead [43]. The update law for the LSF, Ψ , is

$$\begin{aligned} \Psi_{ij}^{n+1} = & \Psi_{ij}^n - \Delta t \{ \max(B_{ij}, 0) \nabla^+ + \min(B_{ij}, 0) \nabla^- + \max(U_{ij}, 0) D^{-x} \Psi_{ij}^n + \dots \\ & \min(U_{ij}, 0) D^{+x} \Psi_{ij}^n + \max(W_{ij}, 0) D^{-y} \Psi_{ij}^n + \min(W_{ij}, 0) D^{+y} \Psi_{ij}^n \} \end{aligned} \quad (6)$$

where D^{-x} and D^{+x} (D^{-y} and D^{+y}) are the backward and forward differences in the x (respectively, y) direction, $B = \mu_1 \beta_0$, $U = \mu_2 [V]_x$ ($W = \mu_2 [V]_y$) is the x (respectively, y) component of $\mu_2 V$, and

$$\begin{aligned} \nabla^+ = & \{ \max(D^{-x} \Psi_{ij}^n, 0)^2 + \min(D^{+x} \Psi_{ij}^n, 0)^2 + \dots \\ & \max(D^{-y} \Psi_{ij}^n, 0)^2 + \min(D^{+y} \Psi_{ij}^n, 0)^2 \}^{1/2} \end{aligned} \quad (7)$$

$$\begin{aligned} \nabla^- = & \{ \max(D^{+x} \Psi_{ij}^n, 0)^2 + \min(D^{-x} \Psi_{ij}^n, 0)^2 + \dots \\ & \max(D^{+y} \Psi_{ij}^n, 0)^2 + \min(D^{-y} \Psi_{ij}^n, 0)^2 \}^{1/2} \end{aligned} \quad (8)$$

Here we have used the notation $[V]_x$ and $[V]_y$ to refer to the x and y components of the vector V . The ∇^+ and ∇^- terms represent the approximation to the isotropic term developed by Osher and Sethian [38], and the remaining terms are simply upwind approximations to the linear advection term. We obtain numerical update laws for the sensitivity equations by formally differentiating (6) with respect to the parameters μ and Λ . The sensitivity equations all have the same left-hand side, which is approximated by

$$(\mu_1 \beta_0 \nu + \mu_2 V) \cdot \nabla S(x, y, t) \approx [D1 \ D2 \ D3 \ D4] \cdot \left[D^{-x} S_{ij}^n \quad D^{+x} S_{ij}^n \quad D^{-y} S_{ij}^n \quad D^{+y} S_{ij}^n \right]^T \quad (9)$$

where

$$D1 = \frac{1}{\nabla^+} \max(F_{ij}, 0) \max(D^{-x} \Psi_{ij}^n, 0) + \dots$$

$$\frac{1}{\nabla^-} \min(F_{ij}, 0) \max(D^{-x} \Psi_{ij}^n, 0) + \max(U_{ij}, 0) \quad (10)$$

$$D2 = \frac{1}{\nabla^+} \max(F_{ij}, 0) \max(D^{+x} \Psi_{ij}^n, 0) + \dots$$

$$\frac{1}{\nabla^-} \min(F_{ij}, 0) \max(D^{+x} \Psi_{ij}^n, 0) + \min(U_{ij}, 0) \quad (11)$$

$$D3 = \frac{1}{\nabla^+} \max(F_{ij}, 0) \max(D^{-y} \Psi_{ij}^n, 0) + \dots$$

$$\frac{1}{\nabla^-} \min(F_{ij}, 0) \max(D^{-y} \Psi_{ij}^n, 0) + \max(W_{ij}, 0) \quad (12)$$

$$D4 = \frac{1}{\nabla^+} \max(F_{ij}, 0) \max(D^{+y} \Psi_{ij}^n, 0) + \dots$$

$$\frac{1}{\nabla^-} \min(F_{ij}, 0) \max(D^{+y} \Psi_{ij}^n, 0) + \min(W_{ij}, 0) \quad (13)$$

On the right-hand sides we have,

$$\beta_0 \|\nabla \Psi\| \approx \begin{cases} \beta_{0,ij} \nabla^+; & B_{ij} \geq 0 \\ \beta_{0,ij} \nabla^-; & B_{ij} < 0 \end{cases} \quad (14)$$

$$V \cdot \nabla \Psi \approx \begin{cases} V_{ij}^x D^{-x} \Psi_{ij}^n; & U_{ij} \geq 0 \\ V_{ij}^x D^{+x} \Psi_{ij}^n; & U_{ij} < 0 \end{cases} + \begin{cases} V_{ij}^y D^{-y} \Psi_{ij}^n; & W_{ij} \geq 0 \\ V_{ij}^y D^{+y} \Psi_{ij}^n; & W_{ij} < 0 \end{cases} \quad (15)$$

$$\mu_1 \beta_{0i} \|\nabla \Psi\| + \mu_2 V_i \cdot \nabla \Psi \approx \begin{cases} \mu_1 \beta_{0,ij} \nabla^+; & B_{ij} \geq 0 \\ \mu_1 \beta_{0,ij} \nabla^-; & B_{ij} < 0 \end{cases} + \begin{cases} \mu_2 V_{i,ij}^x D^{-x} \Psi_{ij}^n; & U_{ij} \geq 0 \\ \mu_2 V_{i,ij}^x D^{+x} \Psi_{ij}^n; & U_{ij} < 0 \end{cases} + \dots$$

$$\begin{cases} \mu_2 V_{i,ij}^x D^{-x} \Psi_{ij}^n; & U_{ij} \geq 0 \\ \mu_2 V_{i,ij}^x D^{+x} \Psi_{ij}^n; & U_{ij} < 0 \end{cases} + \begin{cases} \mu_2 V_{i,ij}^y D^{-y} \Psi_{ij}^n; & W_{ij} \geq 0 \\ \mu_2 V_{i,ij}^y D^{+y} \Psi_{ij}^n; & W_{ij} < 0 \end{cases} \quad (16)$$

where β_{0i} is the partial derivative of β_0 , with respect to the i -th parameter Λ_i , and V_i^x, V_i^y are the partial derivatives of $[V]_x, [V]_y$ with respect to the i -th parameter Λ_i .

5 Example: An Ion-Assisted Etch Model

We now present an example of parameter estimation and process optimization for an ion-assisted etch. Here, a mask has been laid down and patterned, and the objective is to remove the material under the mask, down to a prescribed depth. This removal is accomplished through a combination of chemical reactions by neutral radicals and mechanical sputtering by charged ions. The wafer surface is modeled by three layers. At the lowest level is an inert planar substrate. Above this is the initially planar ‘‘active’’ layer, which is to be etched. At the top is the patterned mask, assumed here to be completely inert. The wafer surface is exposed to a plasma, which is composed of chemically active, but electrically neutral, radicals, and chemically inactive ions. The ions are accelerated by the sheath potential, and after various momentum and charge interactions in the sheath, strike the exposed surface. The energy of impact varies over the points of the free surface, and is a function of position

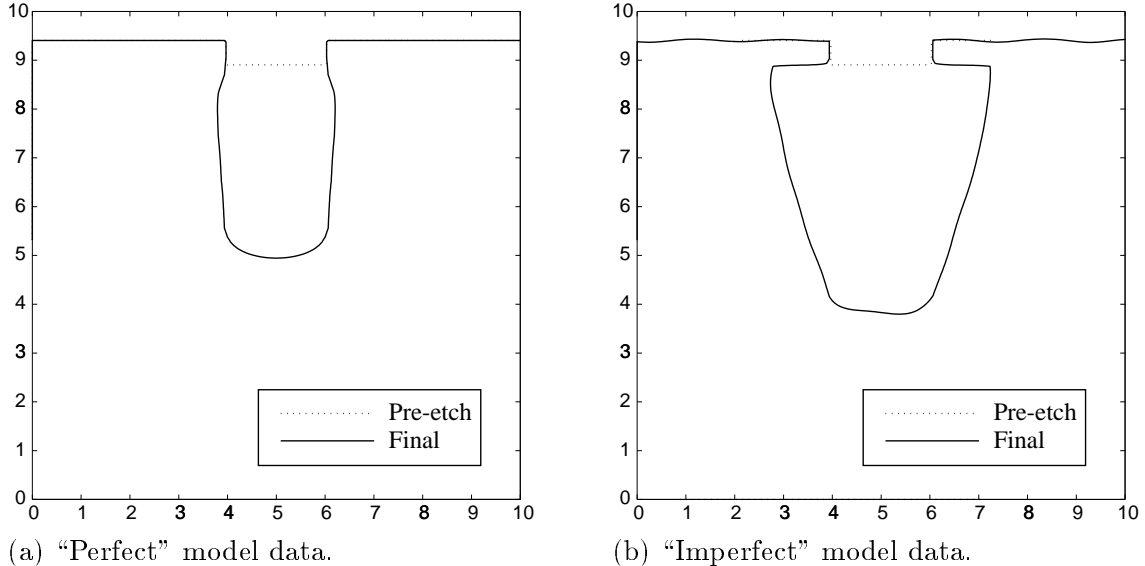


Figure 2: Numerical simulation of trench etching with and without model and measurement errors.

and orientation. We consider that two mechanisms may be at work. The first depends only on the radical concentration, which we assume to be homogeneous. This gives rise to an isotropic etching component. The second is associated with the incident ion energy, and may be due to mechanical sputtering or to enhancement of the chemical etch by the ion collision energy. We apply the following speed function:

$$\mu_1\beta_0(x, y) + \mu_2V(x, y; \Lambda) \cdot \nu = 0 \quad (17)$$

This speed function is of the form (2), with β_0 and V spatially varying, but with β_0 independent of parameters. Here the full estimation problem is to recover the coefficients μ_1 and μ_2 governing the relative contributions of the isotropic and sputtering terms, respectively, and the parameters Λ that determine the vector field V . In this case V has the physical meaning of a normalized energy flux. The spatial dependence of the isotropic component is simple. The isotropic etch rate in the mask and substrate is zero, while in the active layer it is nominally one, scaled by μ_1 . For a more complex spatial dependency see [3], where a process model for lithographic development was analyzed. This case included dependence of μ_1 on up to three parameters.

The details of the process model for the sputtering component are adapted from [26, 46, 47]. That model incorporates simplified versions of shadowing effects and scattering in the plasma sheath. The components of the flux vector, for etching of an infinite trench, are given at any point by

$$[V]_x = \int_{\Omega} j(\theta) \bar{E}(\theta) \sin^2 \theta \cos \phi \, d\theta d\phi \quad (18)$$

$$[V]_y = \int_{\Omega} j(\theta) \bar{E}(\theta) \cos \theta \, d\theta d\phi. \quad (19)$$

Here the integral is taken over Ω , which is the solid angle over which the plasma is visible. The function $j(\theta)$ is the flux of particles moving with angle θ to the local vertical, and $\bar{E}(\theta)$ is the average energy of those particles. Note that all properties are assumed to be symmetric with respect to the local vertical, that is, independent of the azimuth angle, ϕ . Following [46], we set $\bar{E}(\theta) = \cos^2(\theta)$. Explicit computation of the flux vector is generally not possible, due to the term $j(\theta)$. Instead, write (18) and (19) as

$$[V]_x = \int_0^{\pi/2} j(\theta) X(\theta) \sin \theta \cos \theta \, d\theta \quad (20)$$

$$[V]_y = \int_0^{\pi/2} j(\theta) Y(\theta) \sin \theta \cos \theta \, d\theta \quad (21)$$

where the integration over ϕ is carried out following [46], and the functions $X(\theta)$ and $Y(\theta)$ contain all dependence on the geometry of the trench, while $j(\theta)$ contains all dependence on the plasma model. For the infinite trench, $X(\theta)$ and $Y(\theta)$ can be computed in closed form, and are given by,

$$X(\theta) = 2(\sin \phi_b - \sin \phi_a) \sin \theta \cos \theta \quad (22)$$

$$Y(\theta) = 2(\pi - \phi_b - \phi_a) \cos^2 \theta \quad (23)$$

ϕ_a and ϕ_b are the azimuthal visibility limits, for which formulas are given in [46]. The integrals (20) and (21) are approximated using the identity

$$\int_0^{\theta} j(\theta) \sin \theta \cos \theta \, d\theta = \frac{J}{2\pi} \exp(-\Lambda \bar{\sigma}_{\theta}). \quad (24)$$

Here $\Lambda := 25.8 E_{\text{lab}}^{-0.251} \delta D$, is a *single* parameter, with δ the number density of the plasma, D the effective thickness of the scattering region, E_{lab} the ion energy in the laboratory frame, and J is the total particle flux. The function $\bar{\sigma}_{\theta}$ is the normalized total cross section for collision at angles greater than θ , and may [26] be approximated by

$$\bar{\sigma}_{\theta} \approx \begin{cases} \theta^{-0.251} (1 - \theta/1.3); & 0 < \theta \leq 1.3 \\ 0; & 1.3 < \theta \leq \pi/2 \end{cases} \quad (25)$$

Then we have in an interval of width Δ_i centered on θ_i ,

$$\chi_i = \int_{\theta_i - \Delta_i/2}^{\theta_i + \Delta_i/2} j(\theta) \sin \theta \cos \theta \, d\theta = \frac{J}{2\pi} \{ \exp(-\Lambda \bar{\sigma}_{\theta_i + \Delta_i/2}) - \exp(-\Lambda \bar{\sigma}_{\theta_i - \Delta_i/2}) \}. \quad (26)$$

By evaluating the right hand side at intervals using (25), we generate a histogram. The histogram is used to evaluate (20) and (21) by assuming that $X(\theta)$ and $Y(\theta)$ are constant in the interval $[\theta_i - \Delta_i/2, \theta_i + \Delta_i/2]$. Then

$$[\bar{V}]_x \approx \sum_i X(\theta_i) \chi_i \quad (27)$$

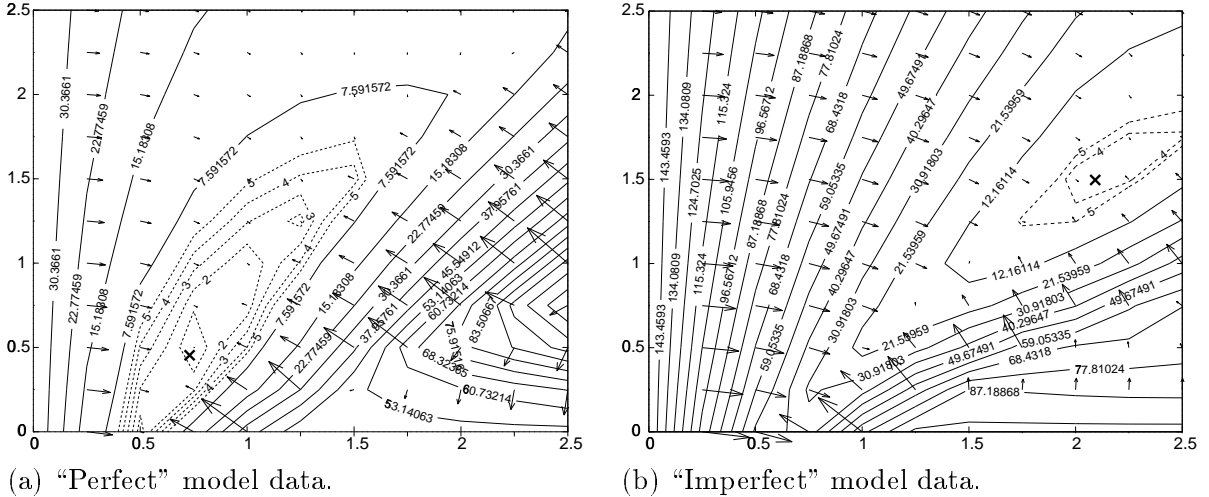


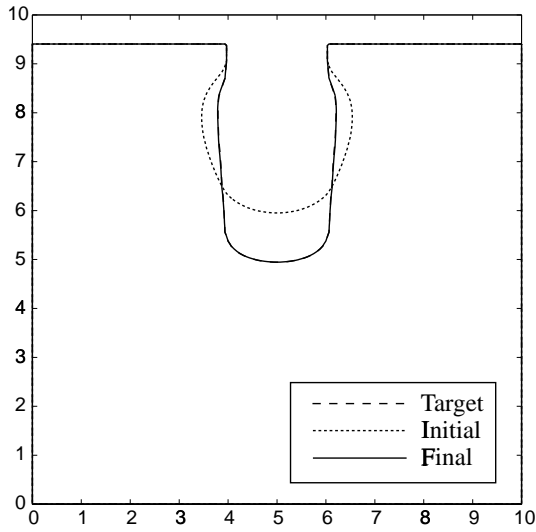
Figure 3: Cost contours and analytically computed gradients for the two parameter estimation problem, (a) with and (b) without simulated measurement errors. Contours are evenly spaced, except dashed contours, which have been added for clarity. Apparent local minima in (a) are artifacts of coarse gridding. In (a), ‘X’ marks true solution. In (b), ‘X’ marks solution found by optimization scheme.

$$[\bar{V}]_y \approx \sum_i Y(\theta_i) \chi_i \quad (28)$$

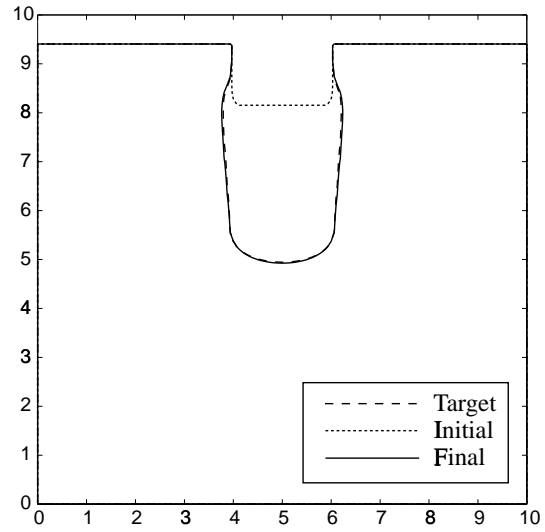
Here the overbar indicates that the constant term $J/2\pi$ has been absorbed into the parameter μ_2 in (17). Best numerical results were obtained by using a logscale for Δ_i , thus increasing resolution at small θ where $\bar{\sigma}_\theta$ changes most rapidly. For further details of these formulas, and physical interpretation of the values, see [26, 46].

We consider optimal estimation of model parameters from knowledge of the initial and final feature shape. The model is assumed to be as given above. Thus the parameters are μ_1 , μ_2 , and Λ . Some of the quantities entering into the parameters are difficult or expensive to measure precisely. One can think of the parameter estimation problem as an experiment to determine, if not these values themselves, at least their effect on the evolution of the feature shape.

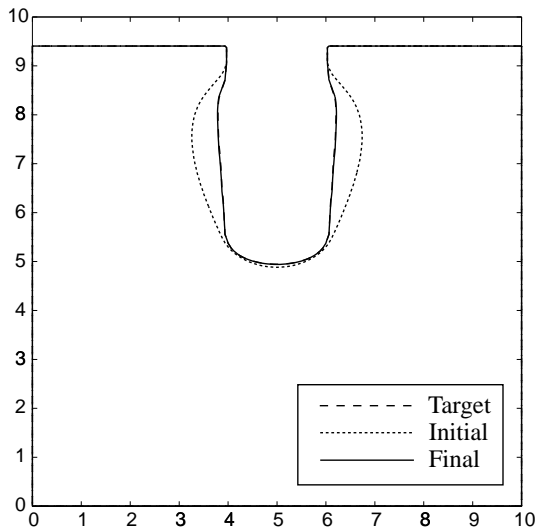
As noted in [4], the template metric does not necessarily give a smooth cost function near the minimizer. For that reason the cost function chosen is $J = (1/2)\rho(\lambda)^2$. The gradient is then $\nabla J = \rho \nabla \rho$. This provides some smoothing when the optimal ρ is small, and allows us to use Hessian-based nonlinear programming methods such as sequential quadratic programming (SQP). The implementation of SQP used for this study was the Matlab Optimization Toolbox routine `fmincon` [12]. For a general overview of SQP see [20]. This routine incorporates general nonlinear constraints. Here only upper and lower bounds were applied. Lower bounds on the parameters are easily chosen; physical considerations dictate that the plasma parameter Λ can never be negative, and validity of the process model cannot be guaranteed if the rates



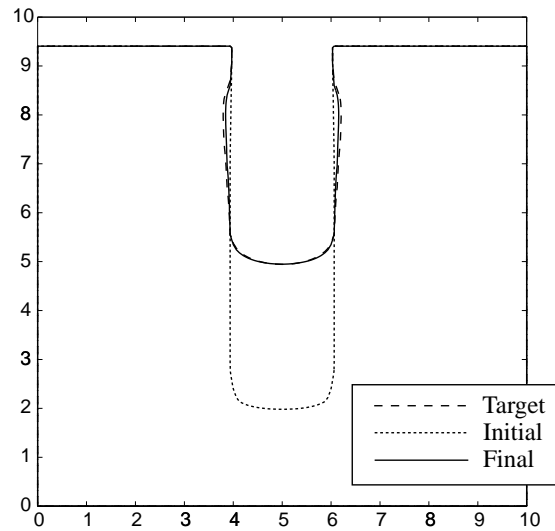
(a) Initial guess (1.0, 1.5).



(b) Initial guess (0.1, 0.0).

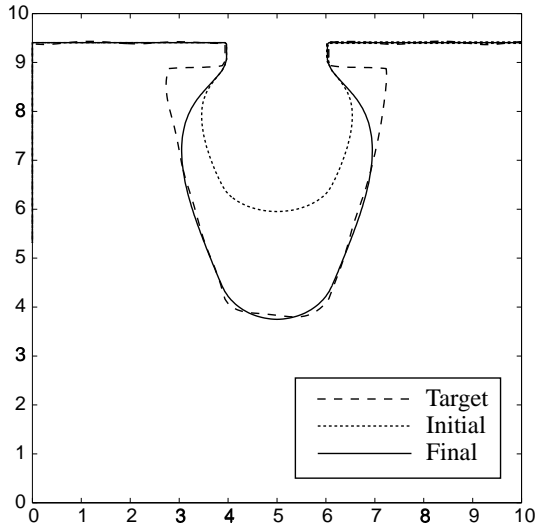


(c) Initial guess (1.5, 1.5).

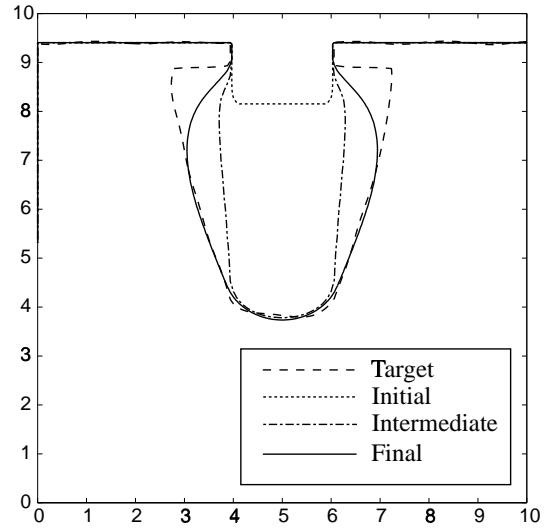


(d) Initial guess (1.0, 0.1).

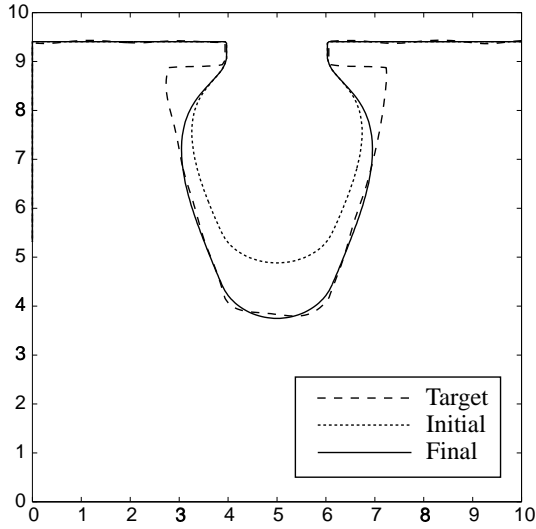
Figure 4: Estimation of two model parameters with a perfect model.



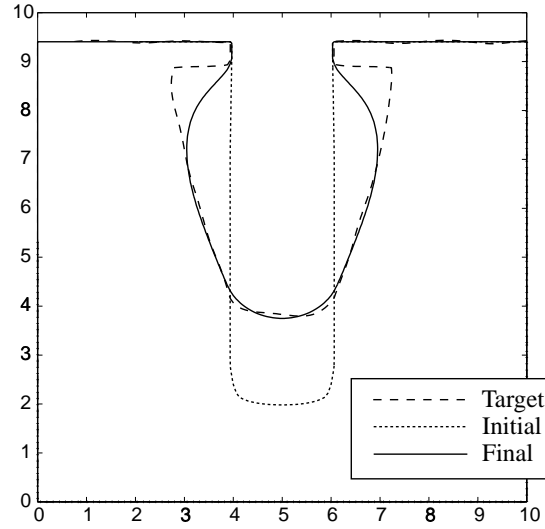
(a) Initial guess (1.0, 1.5).



(b) Initial guess (0.1, 0.0).



(c) Initial guess (1.5, 1.5).



(d) Initial guess (1.0, 0.1).

Figure 5: Estimation of two model parameters with an imperfect model. Case (b) required a constrained intermediate step.

μ_1 and μ_2 become negative. Thus the lower bounds for all three parameters were set to zero. No upper bound was placed on Λ , but μ_1 and μ_2 were limited to prevent numerical instability due to violation of the Courant-Friedrichs-Lewy condition, and to reduce the possibility that the estimated surface would extend beyond the computational domain. The problem was scaled so that constraining the sum of μ_1 and μ_2 to be less than one would guarantee that the final feature would lie entirely within the computational domain. The actual upper bounds on μ_1 and μ_2 were usually set well above this level, but on occasion it was necessary to reduce them.

We consider both the estimation of two parameters, namely μ_2 and Λ , and estimation of all three parameters. The primary purpose of the two parameter problem is to visualize the resulting cost function, and to show that the computed gradients are reasonable. For the two parameter estimation case, the estimated value of μ_1 was fixed at zero. One case was run in which the data was generated by the process model with $\mu_1 = 0$, $\Lambda = 0.454$ and $\mu_2 = 0.73$. Here the model can perfectly capture the observed measurement, and the optimal value of the cost function is zero. We refer to this as the “perfect” model case. In the second case, to simulate modeling errors, measurement errors, and other inaccuracies, the truth model was given a significant isotropic component, $\mu_1 = 0.15$, and the resulting data was then corrupted by a level set function consisting of several sinusoids with random phasing. The remaining parameters are again $\Lambda = 0.454$, and $\mu_2 = 0.73$. We refer to this as the “imperfect” model. Figure 2 shows the results of an 8 time unit etching with a two length unit unmasked aperture, and a 0.5 length unit thick mask, for both the perfect and imperfect truth models. Figure 3 shows cost contours and gradients for the two parameter estimation problem corresponding to the perfect and imperfect models. The results of several numerical experiments are summarized in Table 1. Figure 4 shows the resulting estimated shapes for the perfect model case using four different initial guesses at the parameter values. Figure 5 shows the resulting estimated shapes for the imperfect model case, and the same initial guesses.

The analytically computed gradients are consistent with the cost contours, and the results of the optimization scheme are generally quite good, with the following caveats: 1) The only sensitivity to μ_2 for $\Lambda = 0$ is due to the position of the trench bottom. But for large values of μ_2 the trench extends below the bottom of the computational domain. Thus in these cases the only parameter dependence is on Λ , and increasing Λ causes the sidewalls of the trench to move outward, which may *increase* the value of the cost function. This effect can be seen in the cost contours corresponding to the perfect model, and is present, though not visible, along portions of the boundary in the imperfect model case. The result is that if the minimization routine ends up in certain regions of parameter space, it converges to the boundary. When this occurred, as noted in the table and figures, the constraints were adjusted to keep the solution in the computational domain. We remark that it will probably be necessary to use multiple starting points with the algorithm as presently formulated. In the present case, since most start points give good results, this would be enough to successfully recover the desired parameter values. 2) The convergence criteria, which require that the gradient be small, was never satisfied in the “imperfect model” cases, resulting in many unnecessary iterations. We

Table 1: Results of Model Parameter Estimation

N_p	P/I	λ_0	λ_u	N_{FE}	λ^*	J^*
2	P	(1.0, 1.5)	(2.5, ∞)	67	(0.738, 0.469)	$2.7e-7$
2	P	(0.1, 0.0)	(2.5, ∞)	27	(0.763, 0.509)	$2.1e-4$
2	P	(1.5, 1.5)	(2.5, ∞)	86	(0.741, 0.474)	$8.0e-6$
2	P	(1.0, 0.1)	(1.0, ∞)	27	(0.604, 0.201)	$8.6e-2$
3	P	(0.0, 1.0, 0.0)	(2.5, 2.5, ∞)	155	(0.149, 0.759, 0.496)	$1.9e-5$
2	I	(1.0, 1.5)	(2.5, ∞)	100	(2.09, 1.49)	1.81
2	I	(0.1, 0.0)	(1.0, ∞)	40	(1.00, 0.494)	17.65
2	I	(1.0, 0.4943)	(2.5, ∞)	100	(2.09, 1.48)	1.814
2	I	(1.5, 1.5)	(2.5, ∞)	100	(2.10, 1.50)	1.8162
2	I	(1.0, 0.1)	(2.5, ∞)	100	(2.10, 1.50)	1.8158
3	I	(0.0, 1.0, 0.0)	(2.5, 2.5, ∞)	155	(0.154, 0.716, 0.443)	$2.3e-3$

Legend: N_p = number of parameters, P/I = perfect/imperfect model, λ_0 = initial guess at parameters, N_{FE} = number of function evaluations (maximum of 100 for $N_p = 2$, 150 for $N_p = 3$), λ^* = optimal model parameter estimate, J^* = cost function at optimum. For two parameter case, “perfect” truth model is (0.73, 0.454) and “imperfect” truth model is (0.15, 0.73, 0.454) plus sinusoidal error terms. For three parameter case, “perfect” truth model is (0.15, 0.73, 0.454) and “imperfect” truth model adds sinusoidal error terms.

attribute this problem to the lack of smoothness near the optimum, which is not improved in these cases due to the relatively large residual. However the solutions obtained in these cases appear to be very good. In future versions, the convergence criteria will be modified to account for this effect. 3) In the perfect model case, the initial condition (1.0, 0.1) gives noticeably worse results than the others. The reason for this is as yet unclear, but the figure shows that the shape obtained is still quite close.

Model parameter estimation was performed for the three parameter case using a truth model of (0.15, 0.73, 0.454) and an initial guess of (0.0, 1.1, 0.0). Data and the resulting modeled shapes are shown in Fig. 6, with and without the same sinusoidal measurement error terms as in the two parameter imperfect model case. In the three parameter problem, however, no modeling error terms are included. Numerical results are included in Table 1. Both the numerical and graphical results appear very good.

Next we consider optimal design. Here the target shape is a *desired* surface feature, and we view the parameters as control knobs which may be adjusted to produce this result. In the model being used these values are related to pressure, bias voltage, gas pressures, and other quantities that may indeed be specified by the process designer. The objective is to choose the settings that approximate the desired shape with minimal error. The desired shape is known exactly, so measurement noise is not a factor. The same optimization settings were used here as in the model parameter estimation study.

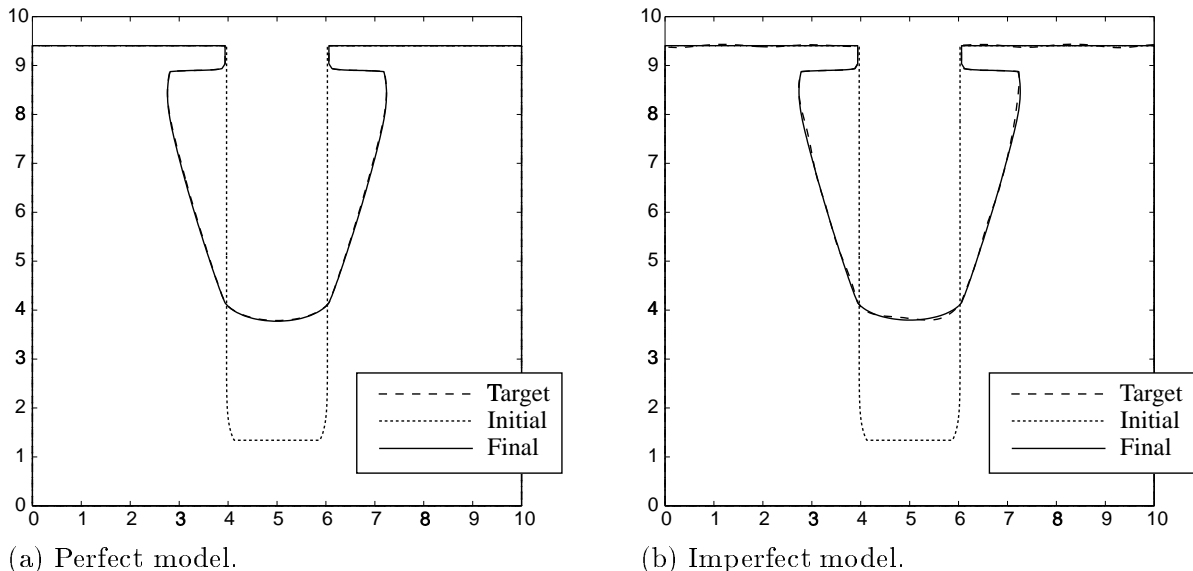
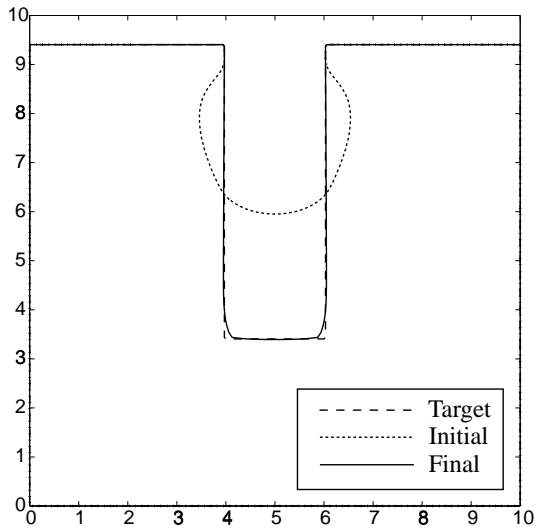


Figure 6: Estimation of three model parameters.

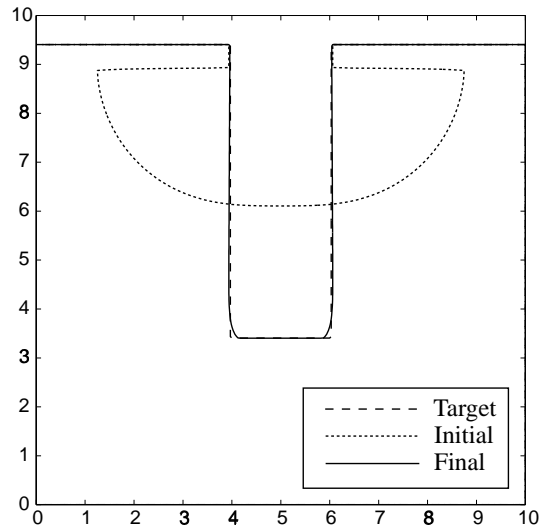
Typically the most desirable trench topography is straight sidewalls and a flat bottom. In this case we know the solution, namely $\mu_1 = \Lambda = 0$, with μ_2 chosen to achieve the desired etch rate. The case was run to obtain design settings for both two parameters (μ_2 and Λ) and three parameters (μ_1 , μ_2 and Λ). The results are shown in Fig. 7. For the two parameter case, μ_1 is fixed at zero. The initial guess was $(1.0, 1.5)$. The final result is $(0.751, 0.033)$. Despite Λ not being exactly zero, the small value obtained gives a good match to the desired shape. For the three parameter case the initial guess was $(0.35, 0.0, 1.0)$. The final result was $(0.0048, 0.724, 0.0002)$, again giving a good match to the desired profile. A somewhat more involved topography is a channel with sloping sidewalls. Again the case was run for both two and three parameters. The results are shown in Fig. 8. The two parameter design does poorly. This is not unexpected, since the target design contains significant undercutting of the mask. On the other hand, the three parameter design, which includes an isotropic component capable of producing the desired undercut, does a nearly perfect job of achieving the target shape.

6 Conclusions and Future Work

This paper constructs a framework for the recovery of parameters appearing in the level set evolution equation, with the objective of matching an observed measured curve, or sequence of curves. The key ideas implemented are a metric for quantifying the distance between estimated and measured curves, and an expression for the gradient of that metric. A minimization scheme based on these elements was used to identify parameters in a certain class

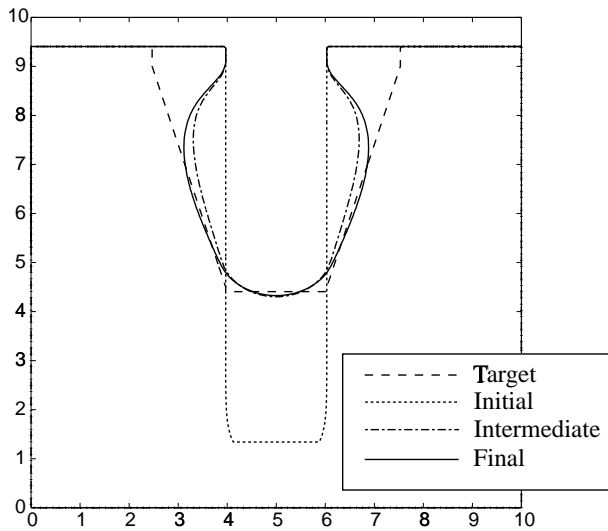


(a) Two parameters.

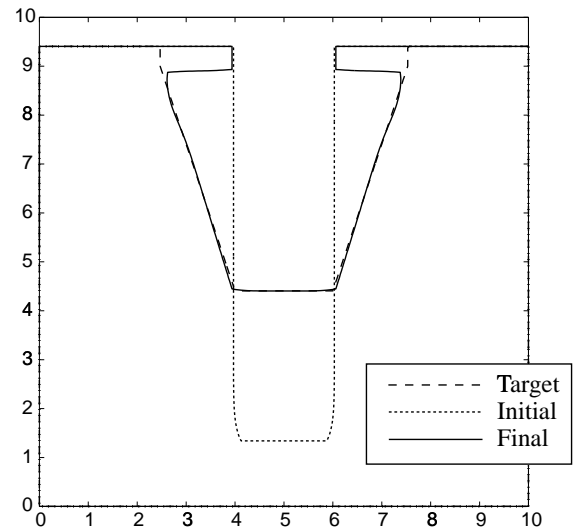


(b) Three parameters.

Figure 7: Optimal design of a straight sided trench



(a) Two parameters.



(b) Three parameters.

Figure 8: Optimal design of a sloped sided channel. Case (a) required an constrained intermediate step.

of spatially and orientation-dependent speed functions. The results show this method to be highly effective at estimating parameters in low-order models to match measured feature shapes and in selecting parameters in a design model to obtain a desired feature shape.

The most pressing need for future work is in increasing the efficiency of the algorithm. This is particularly important in view of the fact that no convexity or other desirable global properties seem to hold for these problems. Thus, it is likely that multiple trials will be needed in practice. The best way to speed up the computation would be a narrowband formulation of the sensitivity equations. This would allow each iteration to solve the evolution and sensitivity PDEs only in a narrow region of the zero level set (see [43]). The other area with the greatest potential to improve the performance of the algorithm is to better match the numerical optimization scheme to the cost function. Some directions to pursue are applying nonsmooth optimization directly to the template metric, or using a smoother metric.

Some capabilities that will be added to the process model are curvature dependent terms, integral re-emission models, and time-varying advection. These last two in particular will enable us to handle realistic deposition problems. A non-convex option will increase the class of etching that may be considered. The plasma etching model considered in the present work suggests an interesting extension. That model uses a simplified assumptions about the plasma sheath to obtain an approximate expression for the energy distribution of the incoming ions. That distribution was expressed as a histogram. A natural extension of of the present work would be to consider the histogram levels directly as unknown parameters—in other words, to solve the inverse problem relating the energy distribution of the ions to the shape of the resulting feature.

References

- [1] D. Adalsteinsson and J. A. Sethian, “A level set approach to a unified model for etching, deposition, and lithography I: algorithms and two-dimensional simulations,” *Journal of Computational Physics* **120**:128–144, 1995.
- [2] S. Andrieux, A. Ben Abda, and M. Jaou, “On Some Inverse Geometrical Problems,” in *Partial Differential Equation Methods in Control and Shape Analysis*, G. Da Prato and J.-P. Zolésio, Eds. , pp. 11–27,1997.
- [3] J. Berg, “Estimation of Parameter Values Appearing in Space and Orientation Dependent Curve Evolution Process Models,” *Proceedings of the American Control Conference*, San Diego, CA, pp. 3905–3909, 1999.
- [4] J. M. Berg, K. Holmström, “On Parameter Estimation Using Level Sets,” *SIAM Journal of Control and Optimization*, Vol. ,37,No. ,5, , pp. 1372–1393, 1999.
- [5] J. M. Berg, “Estimation of Parameters Appearing in the Level Set Evolution Equation,” *Proc. 37th IEEE CDC*, pp. 2323–2328, Tampa, Florida, 1998.

- [6] J. M. Berg, T. Higman, and A. Tannenbaum, "Real-Time Control of Etching Processes: Experimental Results," *Proceedings of the SPIE: Process, Equipment, and Materials Control in Integrated Circuits*, Vol. 3213, A. Ghanbari and A. J. Toprac, Eds., SPIE Conference on Process, Equipment, and Materials Control in Integrated Circuits, pp. 249–260, Austin, TX, October 1–2, 1997.
- [7] J. M. Berg, A. Yezzi, and A. R. Tannenbaum, "Toward Real-Time Estimation of Surface Evolution in Plasma Etching: Isotropy, Anisotropy, and Self-Calibration," *Proceedings of the 36th IEEE CDC*, San Diego, CA, pp. 860–865, 1997.
- [8] J. M. Berg, A. Yezzi, and A. R. Tannenbaum, "Curve Evolution Models for Real-Time Identification with Application to Plasma Etching," *IEEE Transactions Automatic Control*, TAC-44, No. 1, pp. 99–101, 1999.
- [9] J. Borggaard and J. Burns, "A sensitivity equation approach to shape optimization in fluid," in *Flow Control*, M.D. Gunzburger, ed., Springer-Verlag, New York, 1995.
- [10] J. Borggaard and J. Burns, "A PDE sensitivity equation method for optimal aerodynamic design," *Journal of Computational Physics*, **136** (2), pp. 366–384, 1997.
- [11] G. Bouchitte, G. Buttazzo, and P. Seppecher, "Shape Optimization Solutions via Monge-Kantorovich Equation," Publications du Laboratoire d'Analyse Non Linéaire Appliquée, Publication 97/02, 1997.
- [12] T. F. Coleman, M. A. Branch and A. Grace, *Optimization Toolbox User's Guide, Version 2*, The MathWorks, Inc., Natick, MA, 1999.
- [13] S. W. Butler, K. J. McLaughlin, T. F. Edgar, and I. Trachtenberg, "Development of techniques for real-time monitoring and control in plasma etching, II: multivariable control system analysis of manipulated, measured, and performance variables," *J. Electrochem. Soc.* **138**:9, pp. 2727–2735, 1991.
- [14] T. S. Cale, M. B. Chaara, and A. Hasper, "Estimating Local Deposition Conditions and Kinetic Parameters Using Film Profiles," *Mat. Res. Soc. Symp. Proc.*, Vol. 260, pp. 393–398, 1992.
- [15] T. S. Cale, *EVOLVE—A Low Pressure Deposition Simulator*, Center for Solid State Electronics Research, Arizona State University, 1994.
- [16] T. J. Dalton, W. T. Conner, and H. H. Sawin, "Interferometric Real-Time Measurement of Uniformity for Plasma Etching," *J. Electrochem. Soc.* Vol. 141, No. 7, pp. 1893–1900, 1994.
- [17] M. Delfour and J.-P. Zolésio, "Functional Analytic Methods in Shape Analysis," in *Boundary Control and Variation*, J.-P. Zolésio, Ed., Marcel Dekker, Inc., NY, pp. 105–139, 1994.

- [18] M. P. do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, New Jersey, 1976.
- [19] G. P. Carnevale, P. Colpani, A. Marmiroli, A. Rebora, and A. Tixier, "Optimization of a Recessed LOCOS Using a Tuned 2-D Process Simulator," *Simulation of Semiconductor Devices and Processes Vol. 6*, H. Ryssel and P. Pichler, Eds., pp. 286–289, 1995.
- [20] R. Fletcher, *Practical Methods of Optimization*, 2nd Edition, John Wiley & Sons, New York, 1987.
- [21] M. A. Gevelber, M. T. Quiñones, M. L. Bufano, M. C. Deniz, A. Stubbs, J. Weeks, K. Grunke, "CVD Dynamics for Real-Time Control: Multi-Component and Flow Modelling," *Proceedings of the ACC*, Seattle, WA, pp. 1220–1224, 1995.
- [22] J. Glimm, J. W. Grove, X. L. Li, K.-M. Shyue, Y. Zeng, and Q. Zhang, "Three-Dimensional Front Tracking," *SIAM J. Sci. Comput.*, Vol. 19, No. 3, pp. 703–727, 1998.
- [23] D. C. Gray, I. Tepermeister, and H. H. Sawin, "Phenomenological Modeling of Ion-Enhanced Surface Kinetics in Fluorine-Based Plasma Etching," *J. Vac. Sci. Technol. B*, Vol. 11, No. 4, pp. 1243–1257, 1993.
- [24] S. Hamaguchi and S. M. Rossnagel, "Simulations of Trench-Filling Profiles Under Ionized Magnetron Sputter Metal Deposition," *J. Vac. Sci. Technol. B*, Vol. 13, No. 2, pp. 183–191, 1995.
- [25] M. Hankinson, T. Vincent, K. B. Irani, and P. P. Khargonekar, "Integrated Real-Time and Run-to-Run Control of Etch Depth in Reactive Ion Etching," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 10, No. 1, pp. 121–130, 1997.
- [26] C. W. Jurgensen, "Sheath collision processes controlling the energy and directionality of surface bombardment in O₂ reactive ion etching," *J. Appl. Phys.*, **64**:2, pp. 590–597, 1988.
- [27] I. V. Katardjiev, G. Carter, M. J. Nobes, S. Berg, and H.-O. Blom, "Three-Dimensional Simulation of Surface Evolution During Growth and Erosion," *J. Vac. Sci. Technol. A.*, Vol. 12, No. 1, 1994.
- [28] I. Katardjiev, G. Carter, and M. Nobes, "The application of the Huygens principle to surface evolution in inhomogeneous, anisotropic and time-dependent systems," *J. Phys. D: Appl. Phys.* **22**, pp. 1813–1824, 1989.
- [29] I. V. Katardjiev, "A Kinematic Model of Surface Evolution During Growth and Erosion: Numerical Analysis," *J. Vac. Sci. Technol. A*, Vol. 7, No. 6, pp. 3222–3232.
- [30] T. J. Knight, D. W. Greve, X. Cheng, and B. H. Krogh, "Real-Time Multivariable Control of PECVD Silicon Nitride Film Properties," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 10, No. 1, pp. 137–146, 1997.

- [31] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser, Boston, 1992.
- [32] K. J. McLaughlin, S. W. Butler, T. F. Edgar, and I. Trachtenberg, “Development of Techniques for Real-Time Monitoring and Control in Plasma Etching, I: Response Surface Modeling of CF_4/O_2 and CF_4/H_2 Etching of Silicon and Silicon Dioxide,” *J. Electrochem. Soc.* **138**:3, pp. 789-798, 1991.
- [33] K. J. McLaughlin, T. F. Edgar, and I. Trachtenberg, “Real-Time Monitoring and Control in Plasma Etching,” *IEEE Control Systems Magazine*, pp. 3-10, April, 1991.
- [34] J. P. McVittie, J. C. Rey, A. J. Bariya, M. M. IslamRaja, L. Y. Cheng, S. Ravi, and K. C. Saraswat, “SPEEDIE: A Profile Simulator for Etching and Deposition,” *Proceedings SPIE Symposium: Advanced Techniques for Integrated Circuits Processing*, Vol. 1392, pp. 126-138, 1990.
- [35] R. Mlekus, Ch. Ledl, E. Strasser, and S. Selberherr, “Polygonal Geometry Reconstruction after Cellular Etching or Deposition Simulation,” *Simulation of Semiconductor Devices and Processes Vol. 6*, H. Ryssel and P. Pichler, Eds., pp. 50–53, 1995.
- [36] C. J. Mogab, A. C. Adams, and D. L. Flamm, “Plasma Etching of Si and SiO_2 s—The Effect of Oxygen Additions to CF_4 Plasmas,” *J. Appl. Phys.*, Vol. 49, No. 7, pp. 3796–3803, 1978.
- [37] D. Mumford, “Mathematical Theories of Shape: Do They Model Perception?” *SPIE Vol. 1570 Geometric Methods in Computer Vision*, 1991.
- [38] S.J. Osher and J.A. Sethian, “Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations,” *Journal of Computational Physics* **79**, pp. 12-49, 1988.
- [39] O. Pironeau, *Optimal Shape Design for Elliptic Systems*, Springer-Verlag, NY, 1984.
- [40] B. A. Rashap, M. E. Elta, H. Etemad, J. P. Fournier, J. S. Freudenberg, M. D. Giles, J. W. Grizzle, P. T. Kabamba, P. P. Khargonekar, S. Lafortune, J. R. Moyne, D. Teneketzis, and F. L. Terry, “Control of semiconductor manufacturing equipment: real-time feedback control of a reactive ion etcher,” *IEEE Trans. Semiconduct. Manufacturing* **8**:3, pp. 286-297, 1995.
- [41] J. C. Rey, L.-Y. Cheng, J. P. McVittie, and K. Saraswat, “Monte Carlo Low Pressure Deposition Profile Simulations,” *Journal of Vacuum Science and Technology A*, Vol. 9, No. 3, pp. 1083–1087, 1990.
- [42] F. Santosa, “A Level-Set Approach for Inverse Problems Involving Obstacles,” *ESAIM: Control, Optimisation, and Calculus of Variations*, Vol. 1, pp. 17–33, 1996.
- [43] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, second edition, Cambridge University Press, 1999.

- [44] J. A. Sethian, “A Fast Marching Level Set Method for Monotonically Advancing Fronts,” *Proc. Nat. Acad. Sci.* **93**:4, pp. 1591–1595, 1996.
- [45] J. A. Sethian and D. Adalsteinsson, “An Overview of Level Set Methods for Etching, Deposition, and Lithography Development,” *IEEE Trans. Semiconduct. Manufacturing* **10**:1, pp. 167–184, 1997.
- [46] E. Shaqfeh and C. Jurgensen, “Simulation of reactive ion etching pattern transfer,” *J. Appl. Physics*, **66**, pp. 4664–4675, 1989.
- [47] V. Singh, E. Shaqfeh, and J. McVittie, “Simulation of profile evolution in silicon reactive ion etching with re-emission and surface diffusion,” *J. Vac. Technol. B*, **10**, pp. 1091–1104, 1992.
- [48] D. Stokes and G. May, “Real-Time Control of Reactive Ion Etching Using Neural Networks,” *Proceedings of the American Control Conference*, Albuquerque, NM, pp. 1575–1578, 1997.
- [49] D. S. Taylor, M. K. Jain, and T. S. Cale, “Deposition rate dependence of step coverage of sputter deposited aluminum-(1.5%) copper films,” *J. Vac. Sci. Technol. A*, **16**:5, pp. 3123–3126, 1998.
- [50] S. Tazawa, S. Matsuo, and K. Saito, “A General Characterization and Simulation Method for Deposition and Etching Technology,” *IEEE Transactions on Semiconductor Manufacturing*, Vol. 5, No. 1, pp. 27–33, 1992.
- [51] T. L. Vincent, P. P. Khargonekar, F. L. Terry, Jr., “An Extended Kalman Filtering-Based Method of Processing Reflectometry Data for Fast *In Situ* Etch Rate Measurements,” *IEEE Transactions on Semiconductor Manufacturing*, Vol. 10, No. 1, pp. 42–51, 1997.
- [52] D. A. White, D. Boning, S. W. Butler, and G. G. Barna, “Spatial Characterization of Wafer State Using Principal Component Analysis of Optical Emission Spectra in Plasma Etch,” *IEEE Transactions on Semiconductor Manufacturing*, Vol. 10, No. 1, pp. 52–61, 1997.
- [53] E. Zawaideh and N. S. Kim, “A plasma etching model based on a generalized transport approach,” *J. Appl. Phys.*, Vol. 62, No. 15, pp. 2498–2507, 1987.
- [54] E. Zawaideh and N. S. Kim, “A generalized plasma etching model,” *J. Appl. Phys.*, Vol. 64, No. 8, pp. 4199–4207, 1988.
- [55] G. Zhao, E. Wright, and R. V. Grandhi, “Sensitivity Analysis and Preform Die Shape Design in Metal Forging,” *Advanced Materials: Development, Characterization, Processing, and Mechanical Behavior*, MD-Vol. 74, pp. 123–124, 1996.
- [56] G. Q. Zhao, R. Huff, A. Hutter, and R. V. Grandhi, “Sensitivity Analysis Based Preform Die Shape Design Using the Finite Element Method,” *Journal of materials Engineering and Performance*, Vol. 6, No. 3, pp. 303–310, 1997.