

TRAJECTORY MORPHING

APPLIED TO EPITAXIAL THIN FILM GROWTH

A. P. Engelmann, David G. Meyer*, and John Hauser

Nonlinear and Real-Time Control Laboratory

Dept. of Electrical and Computer Engineering

University of Colorado, Boulder, CO 80309-0425

engelman@wonko.colorado.edu dgm2r@wonko.colorado.edu hauser@talon.colorado.edu

Russell Caflich

Mathematics Department

University of California

Los Angeles, CA 90095-1555

caflich@math.ucla.edu

September 15, 1998

Submitted to 1999 ACC

Abstract

This paper presents results showing that trajectory morphing may be used to explore the trajectory space of an epitaxial thin film growth system. Also, a weighted optimization scheme is presented which results in a morphing algorithm that can place increased emphasis on the state portion of the system trajectory.

1 Introduction

Epitaxial thin film growth occurs when a thin film is constructed in a layer by layer fashion. One example of such a method is Molecular Beam Epitaxy (MBE). In MBE, thin film material (e.g. semiconductor

*Corresponding author. Research supported in part by the National Science Foundation under DMS-96158-54, by the National Institute of Standards under contract number 50SBNBC8517, and by the Defense Advanced Research Projects Agency under contract numbers MDA972-93-H-0005 and MDA972-95-1-0016.

material) is heated in a crucible which resides in a vacuum chamber. The heated material evaporates, exits the crucible, and lands on a substrate. A thin film grows on the substrate in a layer by layer manner. Control of epitaxial thin film growth has consisted mostly of simple single-loop feedback of operating variables. For example, PID regulation of a crucible temperature. No attempt at directly regulating critical crystal properties has been made.

Operators wishing to grow a semiconductor crystal have relied on their own expertise and experience to schedule setpoints of incident flux (crucible temperature) and substrate temperature. The setpoints are regulated with PID control. As desired growths become more complicated, there is naturally a need to more directly regulate crystal properties.

Several recent developments have made it possible to begin exploring in formal ways the trajectory space of epitaxial growth during MBE. First, an ODE model has been developed which captures the essential features of multi-layer epitaxial growth [3] during MBE. Other PDE models have been developed [1] [2], but the new model allows application of control theory for systems described by ODE's. The ODE model includes important growth phenomena such as island growth and coalescence. Additionally, a model output, the step edge density, closely correlates with an actual sensor signal, RHEED (Reflection High-Energy Electron Diffraction). Second, a new method for exploring the trajectory space of a nonlinear system has been developed, **trajectory morphing** [6]. Trajectory morphing employs a simplified system model to come up with a desirable approximate trajectory; then projection and homotopy is used to “morph” the desirable approximate trajectory into a desirable true trajectory for the full model.

This paper presents results on applying trajectory morphing to epitaxial thin film growth as modeled by [3]. Section 2 outlines the growth model. We have derived a simplified growth model, suitable for use in morphing, and this is also presented. The fundamentals of trajectory morphing are reviewed in Section 3. Also in Section 3 an enhancement to trajectory morphing using weighted optimization is derived. Section 4 presents results from a simulation study applying trajectory morphing to the thin film growth system.

2 Multi-Layer Epitaxial Growth Model

This section presents an overview of the *Multi-Layer Epitaxial Growth Model* (MLEGM) [3] for thin films. Each layer of the thin film is modeled as one atom thick. Atoms on each layer that are grouped with other atoms form an island. Atoms that are not attached to an island are free and are called adatoms. The model contains three states for each layer of growth simulated. The states for the k^{th} layer are ρ_k , the adatom density (adatoms per unit area); ψ_k , the coverage (area coverage by islands); and n_k , the island density (average number of islands per unit area). The ρ_k and ψ_k states have been normalized to number of atoms per lattice site so their maximum values are one. The model assumes

that adatoms move by standard diffusive lattice-site hopping; when two come together they bond into an island. As other adatoms diffuse to an island, they attach and consequently the island grows. Atoms on top of an island are considered to be on the next layer. The diffusive mobility of an island is taken to be zero.

The differential equations describing how the states change are:

$$\begin{aligned}
 \frac{d}{dt}(\rho_k(\psi_{k-1} - \psi_k)) &= J(\psi_{k-1} - \psi_k) - q_k f_k^- \\
 &\quad - q_{k-1} f_{k-1}^+ - 2m_k \\
 \frac{d}{dt}\psi_k &= q_k v_k + 2m_k \\
 \frac{d}{dt}n_k &= m_k - c_k
 \end{aligned} \tag{1}$$

with constitutive equations:

$$\begin{aligned}
 q_k &= \sqrt{\psi_k n_k} \\
 v_k &= f_k^- + f_k^+ \\
 f_k^+ &= D_+ \rho_{k+1} \left(\frac{1}{r_{k+1}} + \frac{1}{b_k} \right) (\psi_k - \psi_{k-1}) \\
 f_k^- &= D_- \rho_k \left(\frac{1}{r_k} + \frac{1}{b_k} \right) (\psi_{k-1} - \psi_k) \\
 m_k &= D_0 (\psi_{k-1} - \psi_k) \rho_k^2 \\
 c_k &= 2v_k n_k / r_k \\
 b_k &= \sqrt{\psi_k / n_k} \\
 r_k &= b_k (\sqrt{\psi_{k-1} / \psi_k} - 1)
 \end{aligned} \tag{2}$$

Terms in (1), with respect to the k^{th} layer, are: q_k the island step edge density, v_k the island step edge velocity, f_k^+ the adatom flux (up flow) from the k layer to the $k + 1$ layer, f_k^- the adatom flux (down flow) from the k layer to the $k - 1$ layer, m_k the nucleation rate of new islands and c_k the coalescence rate of islands. Also, b_k is the average island diameter and r_k is the average inter-island distance.

The first differential equation in (1) expresses conservation mass: Adatoms arriving on the k^{th} layer must either jump up, down, nucleate, or add to ρ_k . The second equation in (1) equates the rate of change of layer coverage to the sum of a nucleation contribution and growth by attachment (advancing step edges at velocity v_k). The last equation simply says the rate of change of island number density the difference between nucleation and coalescence. The constitutive relations in (2) are more difficult to concisely explain. A full length paper on the model is presently being prepared.

The control are J , the incident adatom flux, and D_0 , the diffusive mobility coefficient which is physically controlled by adjusting the substrate temperature. The terms D_+ and D_- refer to different diffusion coefficients for the upward and downward flux of adatoms since diffusion at a step edge can be distinct from diffusion along a terrace. For simplicity we took $D_0 = D_- = D_+ = D$, though this is not necessary for the methods to work. The model output is $q = \sum_{k=1}^N q_k$ where N is the number of layers. As mentioned, this output corresponds well to an actual RHEED sensor signal.

Figure 1 shows model states and output for a four layer simulated growth. The inputs were $J = 1/4$, $D = 25 \times 10^3$ resulting in $D/J = 10^5$, which is characteristic of MBE growth. The output exhibits

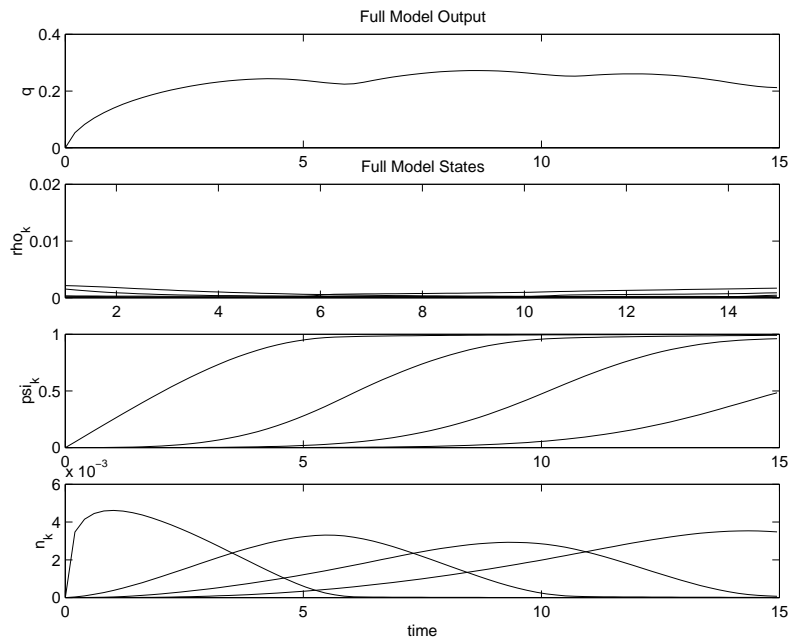


Figure 1: Full Model Output and States for Constant Controls

characteristic oscillatory behavior. The coverage states, ψ_k , are monotonically increasing and show good layer-by-layer growth. The island densities n_k , small at layer start, are maximum near the middle of the layer's growth then drop off to zero at layer end. It is noted that all of the states have positive values. A negative adatom density, coverage, or island density is not clearly physical.

The interested reader is referred to [3] for full details on the derivation of the model.

2.1 Simplified Model

To apply trajectory morphing, one needs a simple model of the system to be morphed. The need for the simple model is reviewed in Section 3. The full MLEGM model presented in the previous section is highly coupled and nonlinear. It is highly non-obvious how to cooperatively use J and D to steer the system in a specified manner. We had hoped to find a physically plausible simplified model that was also differentially flat so as to make the generation of approximate trajectories rather trivial. Unfortunately, we could not accomplish that. Instead simulation, and order of magnitude analysis, was used to drop terms, and develop approximations to terms, in the MLEGM.

The following approximations of (2) were made:

$$\begin{aligned}
 f_{k-0}^+ &= D_+ \rho_{k+1} \left(\frac{1}{r_{k+1}} + 1 \right) (\psi_k - \psi_{k-1}) \\
 f_{0k}^- &= D_- \rho_k \left(\frac{1}{r_k} + 1 \right) (\psi_{k-1} - \psi_k) \\
 v_{0k} &= f_{0k}^- + f_{0k-1}^+ \\
 m_{0k} &= 0.01 D_0 (\psi_{k-1} - \psi_k) \rho_k \\
 c_{0k} &= 2 v_{0k} n_k / r_k
 \end{aligned} \tag{3}$$

where the 0 subscript denotes terms for the simple model. Figure 2 shows the output and states for the simple model with $J = 1/4$, $D = 25 \times 10^3$. Comparing Figures 1 and 2 reveals that, while their are

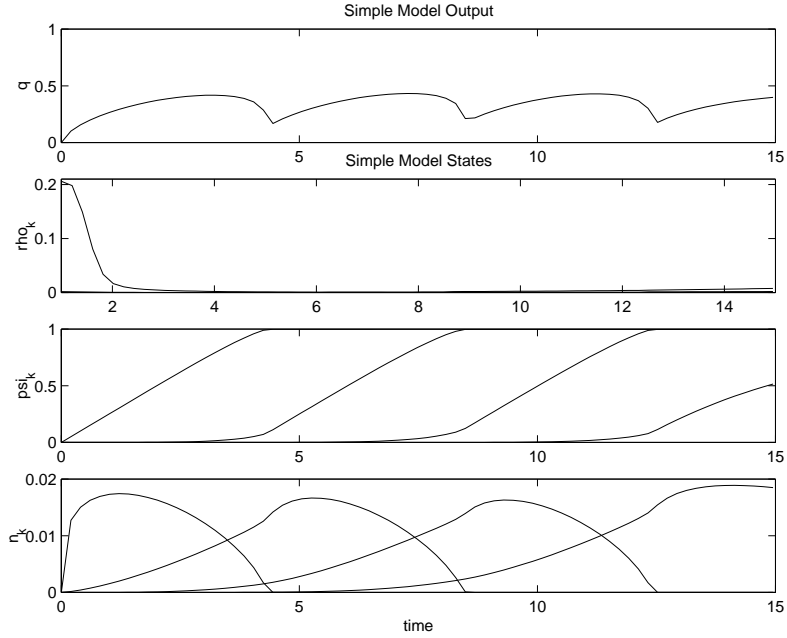


Figure 2: Simple Model Output and States for Constant Controls

inevitable quantitative differences, the simplified model behaves in manner qualitatively similar to the MLEGM. This is important.

3 Trajectory Morphing

Trajectory morphing [6] is a method for exploring the trajectory space of a nonlinear system. Trajectories of a nonlinear system can be defined as follows. Let f be a C^r , $r \geq 1$, mapping from $\mathbb{R}^n \times \mathbb{R}^m$ to \mathbb{R}^n and consider

$$\Sigma : \dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0 \tag{4}$$

Let \mathcal{X} and \mathcal{U} be the space of measurable functions from \mathbb{R}^+ to \mathbb{R}^n and \mathbb{R}^m respectively. A curve $(x(\cdot), u(\cdot)) \in \mathcal{X} \times \mathcal{U}$ satisfying 4 is a *trajectory* of Σ . Recent work has shown that the set of exponentially

stabilizable trajectories has a manifold structure [7].

Trajectory morphing uses projection and homotopy as tools to explore the trajectory space. Consider the trajectory tracking projection operator $\mathcal{P} : (\alpha(\cdot), \mu(\cdot)) \mapsto (x(\cdot), u(\cdot))$ defined by

$$\begin{aligned} u(t) &= \beta(t, x(t), \alpha(t), \mu(t)) \\ \dot{x}(t) &= f(x(t), u(t)) \\ x(0) &= \alpha(0) \end{aligned} \tag{5}$$

where the feedback law β satisfies the invariance condition

$$\beta(t, \alpha, \alpha, \mu) = \mu \tag{6}$$

for any $(\alpha, \mu) \in \mathbb{R}^{n \times m}$ and $\forall t$. It is shown in [6] that \mathcal{P} takes curves in an L_∞ neighborhood of $(\alpha(\cdot), \mu(\cdot))$ into system trajectories, i.e. it is a projection from the vector space of $(\alpha(\cdot), \mu(\cdot))$ curves onto the manifold of system trajectories.

Consider the optimization problem

$$\min_{(x(\cdot), u(\cdot))} \frac{1}{2} \|\mathcal{P}_\lambda(\alpha(\cdot), \mu(\cdot)) - (x_0(\cdot), u_0(\cdot))\|_2^2 \tag{7}$$

where $\mathcal{P}_\lambda : (\alpha(\cdot), \mu(\cdot)) \mapsto (x(\cdot), u(\cdot))$ is the trajectory tracking projection operator defined by

$$\begin{aligned} u(t) &= \beta(t, x(t), \alpha(t), \mu(t)) \\ \dot{x}(t) &= (1 - \lambda)f_0(x(t), u(t)) + \lambda f(x(t), u(t)) \\ x(0) &= \alpha(0) \end{aligned} \tag{8}$$

where f_0 is the simple model, f is the full model and λ is the homotopy scalar that takes values from 0 to 1. The optimization problem above minimizes the tracking error between a desired trajectory of the simple system and a trajectory projected onto the system scaled by the homotopy factor λ . Once the minimization is complete, λ can be increased and the minimization performed again resulting in a trajectory that is closer to the full system trajectories. This process can be repeated until a minimization is found between the nominal trajectory and full system trajectories ($\lambda = 1$). In summary, the process is:

1. Design feedback law β that stabilizes (x_0, u_0) for f_0 .
2. Increment λ and solve (7) using (x_0, u_0) as initial guess. Call solution (x_λ, u_λ) .
3. Simulate f_λ to make sure β stabilizes (x_λ, u_λ) . If not, redesign β .
4. Increment λ and solve (7) using (x_λ, u_λ) as an initial guess.
5. If $\lambda = 1$ stop; otherwise goto number 3.

3.1 Weighted Optimization

The trajectory morphing technique described earlier minimizes the square of the L_2 norm for the entire trajectory, a Cartesian product of a state and a control. In many control applications, a desirable

behavior focuses more on the state portion of a trajectory than the control portion. However, the objective (7) does not recognize this. In this section we present a modification of the optimization problem described in (7) which facilitates the control design process.

The results here are a modification to the derivation in [5]. Consider the following weighted trajectory functional

$$h(\xi) = \|\xi(\cdot) - \xi_*(\cdot)\|_W^2 = \int_0^T \|\xi(\tau) - \xi_*(\tau)\|_W^2 d\tau \quad (9)$$

where $\xi = (x, u)$ is a trajectory and ξ_* is some desired trajectory. Defining $a(\tau) = x(\tau) - x_*(\tau)$ and $b(\tau) = u(\tau) - u_*(\tau)$, then we define the weighted L_2 norm as

$$\|\xi(\tau) - \xi_*(\tau)\|_W^2 = a(\tau)^T W_a a(\tau) + b(\tau)^T W_b b(\tau) \quad (10)$$

The weighting matrices W_a, W_b are taken to be positive definite and are typically diagonal. Define $g(\xi) = h(\mathcal{P}(\xi))$, then

$$Dg(\xi) \cdot \zeta = Dh(\mathcal{P}(\xi)) \cdot D\mathcal{P}(\xi) \cdot \zeta \quad (11)$$

where the D operator denotes differentiation.

The first order search direction that minimizes the functional is given by

$$g_1(\zeta) = \int_0^T D_2 \|\xi(\tau) - \xi_*(\tau)\|_W^2 \cdot \zeta + \frac{1}{2} \|\zeta(\tau)\|^2 d\tau \quad (12)$$

Computing the derivative of the weighted L_2 norm yields

$$D_2 \|\xi(\tau) - \xi_*(\tau)\|_W^2 = (a(\tau)^T W_a, b(\tau)^T W_b) \quad (13)$$

and results in the following linear quadratic optimal control problem

$$\begin{aligned} \min \int_0^T & a(\tau)^T W_a z(\tau) + b(\tau)^T W_b v(\tau) + \\ & \frac{1}{2} \|z(\tau)\|^2 + \frac{1}{2} \|v(\tau)\|^2 d\tau \end{aligned}$$

subject to

$$\dot{z}(t) = A(\xi(t))z(t) + B(\xi(t))v(t) \quad (14)$$

where the minimization is done over $v(\cdot)$ and z_0 . Standard techniques can be used to solve the problem [4] (e.g., solve the Hamilton-Jacobi equation). The optimal cost-to-go from state $z(t)$ at time t is given by

$$V(t, z(t)) = \frac{1}{2} z(t)^T P(t) z(t) + r(t)^T z(t) + c(t) \quad (15)$$

where P , r , and c satisfy the following quadratic (Riccati), linear, and scalar equations (suppressing t arguments):

$$\begin{aligned} -\dot{P} &= A^T P + P A - P B B^T P + I, \\ -\dot{r} &= (A - B B^T P)^T r + W_a a - P B W_b b, \\ -\dot{c} &= -\frac{1}{2} b^T W_b^T W_b b - \frac{1}{2} r^T B B^T r, \\ P(T) &= 0, \quad r(T) = 0, \quad c(T) = 0 \end{aligned}$$

Providing that $P(0) > 0$, the initial condition minimizing $V(0, z(0))$ is $z_0 = -P(0)^{-1} r(0)$. The steepest descent direction is then $\zeta = (z, v)$ where

$$\begin{aligned} \dot{z}(t) &= A(\xi(t)) z(t) + B(\xi(t)) v(t), \quad z(0) = z_0 \\ v(t) &= -B(\xi(t))^T P(t) z(t) - B(\xi(t))^T r(t) - W_b b(t) \end{aligned}$$

3.2 Design of Stabilizing Feedback

Morphing requires a feedback¹ which exponentially stabilizes approximate trajectories. To this end, we designed a controller using Linear Quadratic Regulator (LQR) theory (see [8]). The linearization of (1) along a desired trajectory (x^*, u^*) results in the following linear time varying system:

$$\dot{z}(t) = A(t) z(t) + B(t) v(t) \tag{16}$$

where $A(t) = D_1 f(x^*, u^*)$ and $B(t) = D_2 f(x^*, u^*)$. Stabilizing (16) may be accomplished by solving the following optimization problem:

$$\min_{u(\cdot)} \int_0^T z(t)^T Q z(t) + v(t)^T R v(t) \quad dt \tag{17}$$

where Q, R, M are positive definite matrices. The optimal solution to (17) is known to be

$$u_{opt}(t) = R^{-1} B(t)^T P(t) z(t) = K(t) z(t) \tag{18}$$

where P is the solution to the Riccati equation with final condition (suppressing t arguments)

$$\begin{aligned} -\dot{P} &= A^T P + P A + Q - P B R^{-1} B^T P \\ P(T) &= 0 \end{aligned}$$

This particular feedback design method was chosen for two reasons. First, solving the LQR problem involves selecting matrices Q and R . These matrices balance the cost of the optimization between tracking errors and control perturbations which gives the designer flexibility in the design. Secondly, the optimal control in (18) is guaranteed to stabilize the linearization (16) around the desired trajectory because $V(t, z(t)) = z(t)^T P(t) z(t)$ is a Lyapunov function for (16). The stabilization is valid for some region local to the desired trajectory.

¹A single feedback through all values of λ is not required, i.e, you can redesign the feedback at each step of the morphing process.

4 Simulation Study

In this section we present results of trajectory morphing on the MLEGM. For this study, a four layer MLEGM was chosen; hence we have 12 states. Showing all of the 12 state trajectories is cumbersome and messy. So, to simplify the presentation, the output of each trajectory of interest will be shown and the relative state and control tracking errors will be shown as a measure of the overall tracking accuracy. Tracking errors will be computed using an L_2 norm.

One complete step of the morph will be described; subsequent steps are very similar.

Following the algorithm presented in Section 3, the first step in trajectory morphing is to generate a feedback law that stabilizes a nominal (desired) trajectory. The nominal trajectory was taken to be the one resulting from applying constant controls to the simple model. This was shown in Figure 2. This nominal trajectory was then used to compute the linearization of the MLEGM along this trajectory. The linearization was then used in computing a LQR feedback law described in section 3.2. The final trajectory tracking feedback controller used for projection was

$$u(t) = \mu(t) + K(t)(\alpha(t) - x(t)) \quad (19)$$

Note that this feedback law satisfies the invariance condition in (6).

The next step in the morphing process is to solve the optimization problem in (7). The homotopy scalar was incremented to a small value, $\lambda = 0.1$. Then the initial guess for a trajectory was projected onto the $\lambda = 0.1$ system trajectories. At this point, the initial guess is simply the nominal trajectory. Figure 3 shows the nominal output and the projected output. Also shown is the state and control tracking errors. A relatively large deviation in output appears at about 9 seconds into the trajectory. At the same time, there is also a relatively large increase in state and control tracking error. The state-tracking error for the entire trajectory was 0.31 and the control-tracking error was 0.21. Figure 4 shows the projected controls; controls which stabilize the $(x_\lambda(\cdot), u_\lambda(\cdot))$ trajectory onto the $\lambda = 0.1$ system.

The optimization problem was then solved using a steepest descent method. See section 3.1 for a description. The steepest descent direction (gradient) of the functional is shown in Figures 5 and 6. Notice that some of the state and control gradient components are negative. This illustrates the need for a stabilizing feedback in the projection operator. Recall that negative states and controls do not make any sense in the MLEGM model. The projection operator ensures that the updated trajectory is projected onto real system trajectories. It is interesting to note that the control gradients shown in Figure 6 are simply inversions of the projected control. Recall that the nominal or desired control trajectories are simply constants. So the control gradient, that pushes the controls back to the nominal, is simply the opposite of the projected controls.

After calculating the steepest descent direction of the functional, a scaled version of the steepest descent was added to the current trajectory guess and then projected onto the system trajectories. The result is the morphed trajectory (again, one step shown). Figure 7 shows the morphed output and its

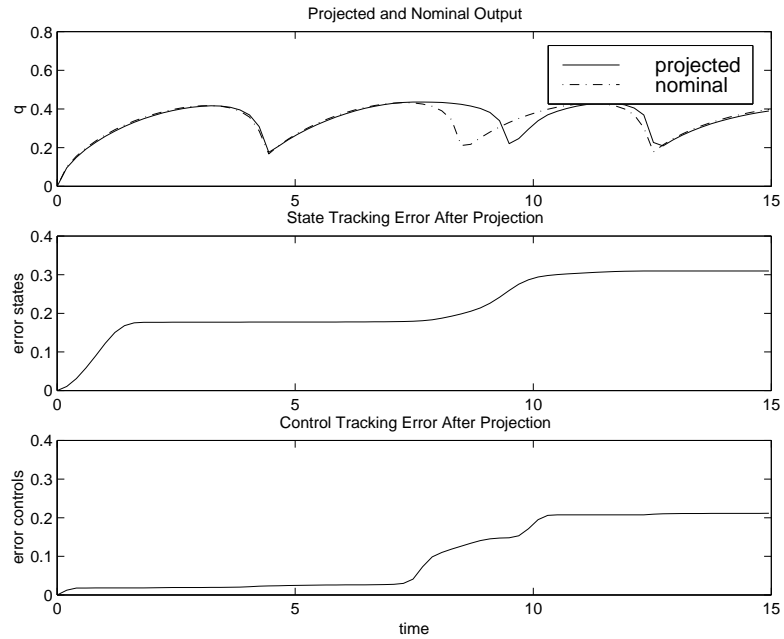


Figure 3: Projected Output and Tracking Errors

associated tracking errors. As is shown the large deviation in the middle of the output trajectory has been diminished. Although there is noticeable deviations in output tracking at more locations along the trajectory, the overall state-tracking error was reduced from 0.31 to 0.22, a reduction of %29. The overall control-tracking error also went down slightly from 0.21 to 0.19.

5 Conclusion

In conclusion, it has been shown that trajectory morphing can be successfully applied to explore the trajectory space for a epitaxial thin film growth model.

A multi-layer epitaxial thin film growth model was presented along with a simplified version used to generated a desired trajectory for the full model. The trajectory morphing technique was reviewed and shown to be successful in searching the growth model's trajectory space.

References

- [1] Caffisch, R.E., M. Gyure, B. Merriman, S. Osher, C. Ratsch, D. Vvedensky, and J. Zinck. "Island Dynamics and the Level Set Method for Epitaxial Growth", 1998.
- [2] Caffisch, R.E., W. E, M. Gyure, B. Merriman, C. Ratsch, "Kinetic Model for a Step Edge in Epitaxial Growth ", 1998.
- [3] Caffisch, R., David G. Meyer,, Personal communications, 1998.

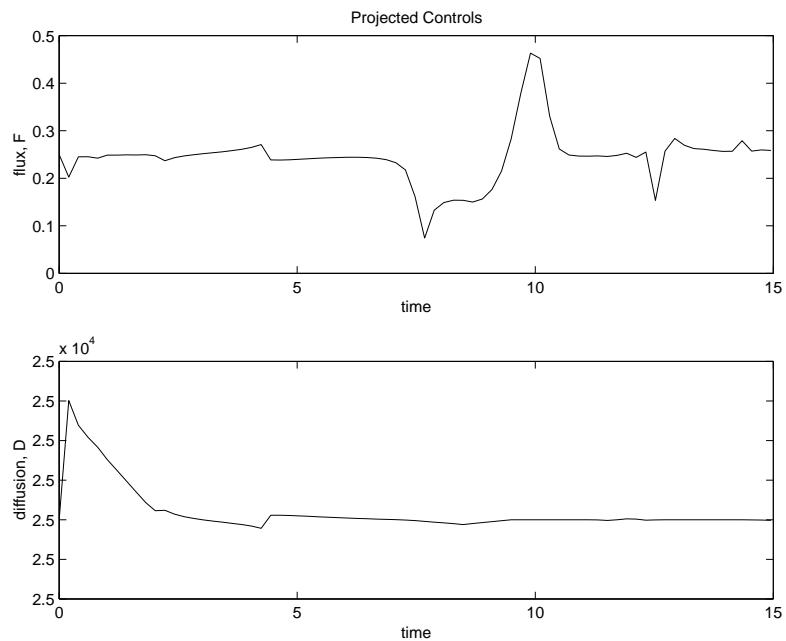


Figure 4: Projected Controls

- [4] Dorato, Peter, C. Abdallah, and V. Cerone, Linear-Quadratic Control: An Introduction, Prentice-Hall Inc., 1995.
- [5] Hauser, John, Personal notes entitled "Local Optimization of a Trajectory Functional", 1998.
- [6] Hauser, John, and David G. Meyer, "Trajectory Morphing for Nonlinear Systems", *Proc. American Control Conference*, Philadelphia, Penn., 1998, ppg 2065-2070.
- [7] Hauser, John, and David G. Meyer, "The Trajectory Manifold of a Nonlinear System", submitted to *1998 Conference on Decision and Control*.
- [8] Kailath, T., Linear Systems, Prentice-Hall Inc., 1980.

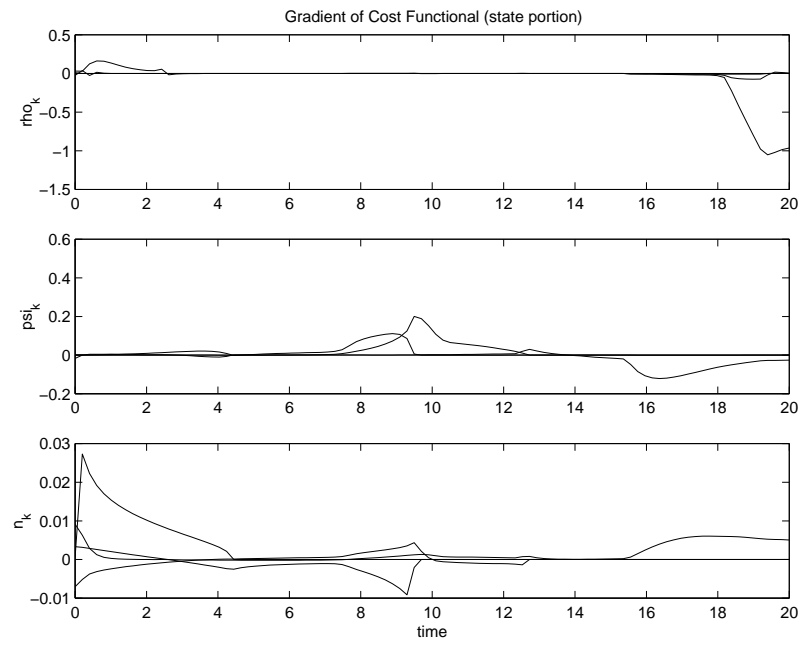


Figure 5: State Steepest Descent Direction

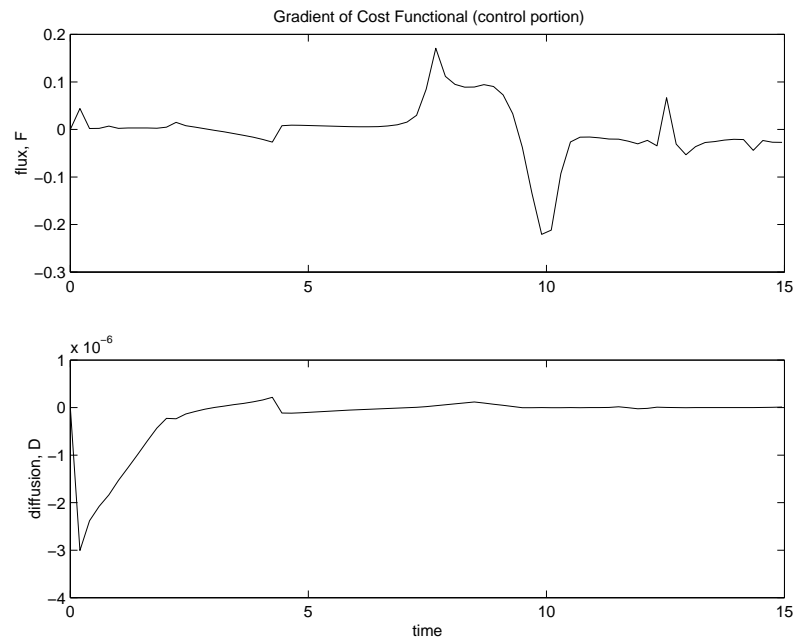


Figure 6: Control Steepest Descent Direction

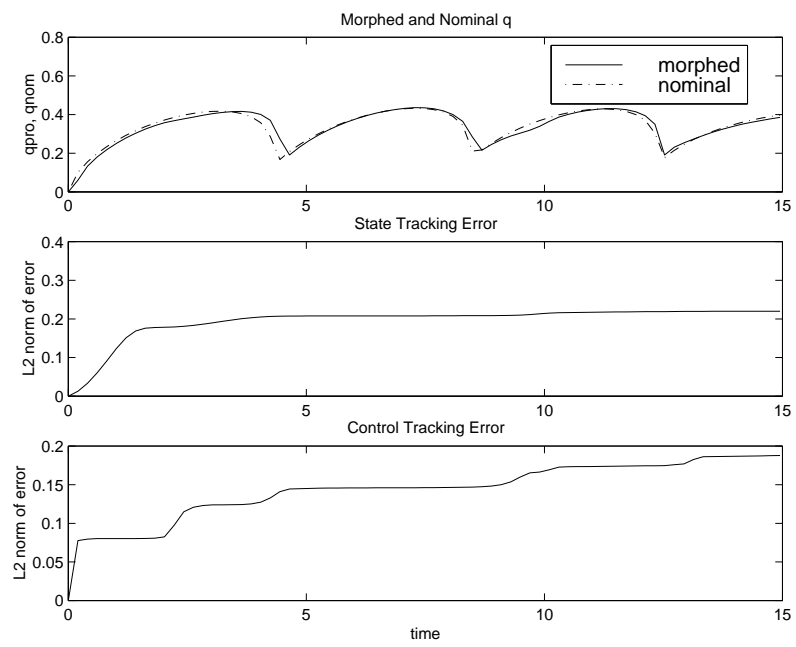


Figure 7: Morphed Output and Tracking Errors