Feedback Process Control for Nano-manufacturing of Semiconductor Circuits

W. Kohn* and Z. B. Zabinsky**
*Atigeo Corporation, Bellevue WA, and
CUNY Graduate Center, New York NY, wolf.kohn@atigeo.com
**Atigeo Corporation, Bellevue WA and
University of Washington, Seattle WA, zelda@u.washington.edu

ABSTRACT

A feedback process control realizes 3-D circuits with a four nanometer footprint using a self-assembly nano-manufacturing facility including actuator controls. At this footprint level, quantum effects are significant. Our technology for quantizing the design of a digital circuit into a 3-D implementation using master equation models is based on a quantized Hamiltonian that is tracked by the physical Hamiltonian to deploy the circuit design. This quantum hybrid controller enables transformation of a desired circuit (expressed in logic and continuous rules) into manufacturing processes at quantum levels that includes interdependent multi-stage processes. It achieves performance criteria, such as maximize yield, maximize signal-to-noise ratio, and minimize entropy.

Keywords: process control, IoT, scaffolding, self-assembly, quantum lithography

1 INTRODUCTION

A nano-manufacturing facility requires high levels of precision due to the strong quantum effects for devices at a 2 to 4 nanometer footprint. We develop a feedback process control to realize a 3-D scaffolding approach. Our technology uses master equation models that are based on a quantized Hamiltonian that is tracked by the Hamiltonian of a modified ink-jet based printer to deploy the circuit design.

We develop a quantum hybrid controller for manufacturing processes at the mesoscopic and quantum levels that includes interdependent multi-stage processes (e.g., quantum lithography, molecular epitaxial deposition, vertical dots, quantum wells, quantum wires, etc.). The quantum hybrid controller is able to transform a desired circuit (expressed in logic and continuous rules) into manufacturing processes that will realize the nano-circuit and achieve performance criteria (such as maximize yield, maximize signal-to-noise ratio, minimize energy consumption, minimize dissipation, etc.). The proposed system will include a language for the rules that describe the specifications of the circuit, an interface with the manufacturing processes to monitor and prescribe the sequencing, and a design dashboard. The quantum hybrid controller works with Hamiltonian operators to represent the system to achieve overall manufacturing synchronization and control. The quantum hybrid controller will enable automation of nano-manufacturing of nano-circuits. This will facilitate manufacturing of IoT systems.

The manufacturing process that we will follow is based on self-assembly and will be discussed in a separate paper.

2 TECHNICAL OVERVIEW

This section summarizes our approach for the design and implementation of switching functions in a multi-tower realization of these functions using a quantum representation that allows the analysis and optimization of the design.

2.1 Scaffolding Implementation

Figure 1 illustrates the contrast between standard semiconductor manufacturing and the proposed scaffolding approach. In the scaffolding approach, a barrier layer may be added to an individual tower as opposed to a complete layer. Wires and tunneling bridges will be added, according to the specification from the quantum hybrid controller [11].

The quantum hybrid controller will enable automation of nano-manufacturing of nano-circuits. Once the "blueprint" for the desired circuit is encoded, the controller interacts with the manufacturing processes to achieve the desired behavior with desired performance metrics.

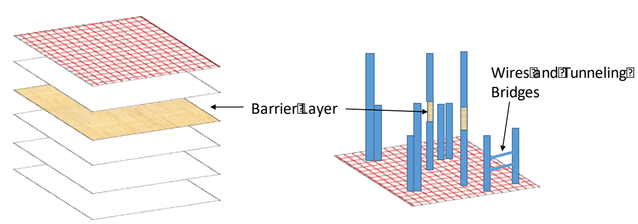

Figure 1: Standard and scaffolding semiconductor circuits.

The optimal control system implements the process control for manufacturing IoT systems represented by the switching functions. This manufacturing process control is based on a quantum representation of the switching functions [10].

2.2 Quantum Hybrid Process Controller

The central approach in a quantum hybrid control is to express the switching functions of the design with a Hamiltonian representation. The switching functions and associated rules describing the desired behavior of the circuit will include: rules for physical principles, such as conservation of mass; rules that capture the tunneling effect and cross-talk; and soft rules for quantifying behavior and performance criteria. We construct the Hamiltonian from the switching functions and the rules using a Cooperative Distributed Inferencer (CDI) [2].

Figure 2 illustrates the computational paradigm, composed of seven sequential steps. Each **switching function**, corresponding to a tower in the scaffolding, is either a map of the form

$$f : S_1 \times \cdots S_N \to D \tag{1}$$

where $S_i, i = 1,\ldots,N$ and D are finite subsets of the natural numbers expressed in binary encoding, or a solution of an iterative process (finite state machine) of the form

$$y_{n+1}^i = f^i\left(y_n^1,\ldots,y_n^k\right), i = 1,\ldots k \text{ where } f^i : S_i \to S. \tag{2}$$

For both cases, we can express f by the encoding

$$F : \left[0,1,\ldots,p^{N-1}\right] \to \left[0,1,\ldots,p\right] \text{ where } x = \sum_{s=0}^{N-1} n_s \bullet p^{N-1-s}$$

$$\text{and } F(x) = \begin{cases} f\left(n_0,\ldots,n_{N-1}\right) & \text{if defined} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The functions are application dependent, but they can always be expanded in terms of discrete orthogonal polynomials [6] and [7].

The **spectral continualization** step consists primarily of *completing* the encoding of F with a step function Φ as

$$\begin{gathered} \Phi : \left[0, p^{N-1}\right) \to \left[0, p^{N-1}\right) \text{ and} \\ \Phi(y) = F(x), \text{ for each } y, y \in [x, x+1), x \in \left\{0,\ldots,p^{N-1}\right\}. \end{gathered} \tag{4}$$

With some needed modification, this continualization process can be extended to discrete functions that are specified as solutions of iterations of the form

$$Z_{k+1} = g\left(Z_k, k\right) \text{ with } Z_k \in S^N, S = \left\{0,\ldots,N \mid N \text{ finite}\right\} \tag{5}$$

for each k, k an integer.

Modifying the methods of Kushner and Clark [1], the encoding and continualization of a process leads to a differential equation of the form

$$\dot{x}(t) = G\left(x(t),t\right) + B(t) \tag{6}$$

with a sampling rule of the form

$$G(\bullet,t) = \text{continualized version of } g(\bullet,k), \tag{7}$$

where $t \in \left[k\Delta,(k+1)\Delta\right), k \in N$, and $B(t)$ is an error correction term to account for the finite approximation of the continuous model.

The **Hilbert expansion** consists of selecting a basis of discrete polynomials to express the function Φ in the Hilbert space via an orthogonal expansion,

$$\Phi(y) = \sum_{y \in [0,p]^N} \beta_k \bullet \phi^k(y) \tag{8}$$

$$\beta_k = \left(\int_0^{p^N} \Phi(x) \bullet \bar{\phi}^k(x)\,dx\right) / \left(\int_0^{p^N} \phi^k(x) \bullet \bar{\phi}^k(x)\,dx\right) \tag{9}$$

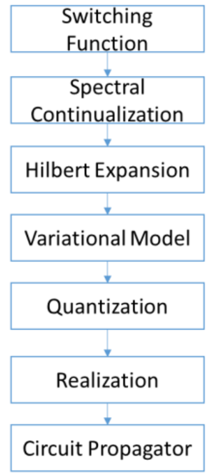

Figure 2. Design Procedure

This quantum hybrid *controller* is distributed to match the scaffolding implementation. The specific architecture will be discussed in the section on scaffolding architecture. In this architecture, each vertical device (e.g., tower) in the circuit corresponds to an agent.

The **variational model** is constructed by an inverse algorithm that generates the Lagrangian of the IoT system, as detailed in [3]. The action integral defined by the Lagrangian L, generates the same trajectories as (6). We convert the problem to a variational problem because the associated Lagrangian, and corresponding Hamiltonian, encodes the switching functions of the IoT we want to realize. Specifically, the variational problem is defined as

$$\begin{gathered} \min_x \int L(x,\dot{x},t)\,dt \text{ where } L \text{ is generated by} \\ \text{the following linear hyperbolic equation} \\ q_t + G_x q + \dot{x} q_x + \left(\dot{x} G_x + G_t + B_t\right) q_{\dot{x}} = 0 \\ \text{and } q(x,\dot{x},t) = L_{\dot{x}\dot{x}}(x,\dot{x},t). \end{gathered} \tag{10}$$

Equation (10) can be solved analytically by the method of characteristics and a quadrature integration [3].

The **quantization** of the Lagrangian is carried out by converting the Lagrangian to a Hamiltonian function [9]. Specifically, $p = L_{\dot{x}}$ is the generalized momentum of the system.The canonical commutation relations $\left[x_k, p_l\right] = i\delta_{k,l}$ are satisfied for $k,l = 1,\ldots,N$. The Hamiltonian is

$$H(x,p) = \sum p_k \bullet \dot{x}_k - L, \text{ Legendre transformation.} \tag{11}$$

The quantization of (11) is carried out by constructing $\hat{H}\left(\hat{x}, \frac{1}{i}\frac{\partial}{\partial x}\right)$ by replacing x with $\hat{x}$, and p with $\frac{1}{i}\frac{\partial}{\partial x}$ in $H(x,p)$.

The evolution equation of the IoT system is completely specified by the corresponding Schrodinger equation,

$$i\hbar\frac{\partial}{\partial t} = \hat{H}\left(\hat{x}, \frac{1}{i}\frac{\partial}{\partial x}\right) \tag{12}$$

TechConnect Briefs 2017, TechConnect.org, ISBN 978-0-9975117-8-9

The Hamiltonian as defined in (11) is a model of how the circuits in the design will operate in the quantum domain. It must satisfy the condition

$$\left|\hat{H} - H_0 + H_f\right| \le \varepsilon \tag{13}$$

where H_0 is a model of the physics of the circuits, and H_f is the Hamiltonian associated with the switching function to be computed [8].

The **realization** step is to construct H_f so that the discrete spectrum of the total Hamiltonian $H_0 + H_f$ has to satisfy

$$\left(H_0 + H_f\right)\left|\Psi_k\right\rangle = \lambda_k \left|\Psi_k\right\rangle \tag{14}$$

where $\{\lambda_k, k = 1, \ldots, N\}$ is the set of eigenvalues of $H_0 + H_f$ with corresponding eigenvectors $\{\Psi_k, k = 1, \ldots, N\}$.

The central realization step is to deploy H_f that satisfies

$$\left|\lambda_k - \beta_k\right| \le \varepsilon / N \text{ and}$$
$$\int_0^\infty \left|\phi^k(x) - \Psi_k(x)\right| dx \le \varepsilon \tag{15}$$

Note that, by the continualization process of the switching function, the computational problem has been reduced to computing functions with scalar arguments that approximate the desired switching function, as in (15).

The **circuit propagator** based on the design criterion in (15) is a self-assembly quantum implemention of the transition between the discrete states of the spectrum.

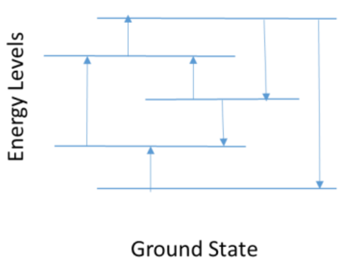

Figure 3. State transitions between eigenvectors. Up arrows are field driven; down arrows are spontaneous.

This transition is realized in a scaffolding implementation where the components of each scaffolding step have a representation illustrated in Figures 4 and 5. In Figure 4, each column of computational elements corresponds to a tower in Figure 1.

The transitions illustrated in Figure 5 are described by the following equations, as in [4]. The probability density operator is defined by

$$\rho(t) = \sum_j p_j \left|\Psi_j(t)\right\rangle\left\langle\Psi_j(t)\right| \tag{16}$$

and by differentiating (16) and using some algebraic manipulations with (12), we get the master equation for propagating ρ

$$i\hbar\frac{\partial}{\partial t}\rho = \left[\hat{H}, \rho\right] = \hat{H}\rho - \rho\hat{H} \tag{17}$$

The probability transition in Figure 5 realizes the master equation associated with the Hamiltonian. The device in Figure 5 is deployed in the crystal by self-assembly [11].

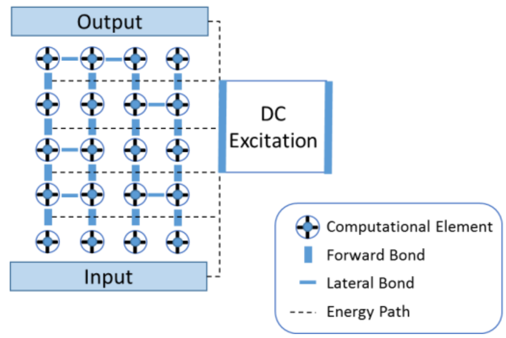

Figure 4: Quantum processor.

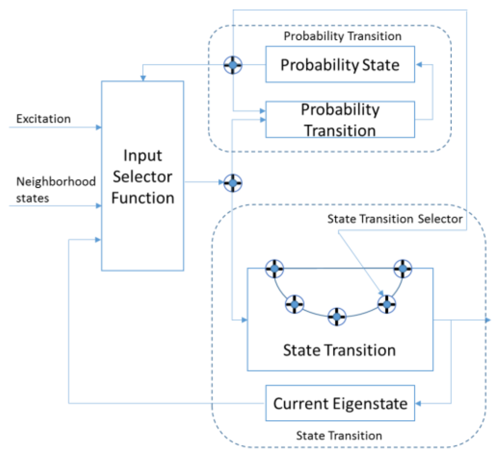

Figure 5: Computational Element.

2.3 Design Architecture

The overall design architecture is shown in Figure 6. For a given implementation, each tower in Figure 1 realizes a specific switching function. The coupling of the switching functions is realized by a "bridge", as in Figure 4. For manufacturing purposes, each tower is self-assembled by a "control agent" as shown in Figure 6.

The control law specifies how to propagate the Hamiltonian to achieve the desired behavior and performance. Figure 6 illustrates the system for an agent. The synchronization between the towers on each layer is achieved by updating the agent Hamiltonians with a Hamiltonian field that represents the overall circuit and performance characteristics. This synchronization will determine when to construct a quantum wire, for example, that achieves sharing of energy. Instead of computing all combinations of agents (which is impossible, due to the impossibility of solving an *n*-body Pareto game), we play a two-body Pareto game between each agent and the design field to update both the agent Hamiltonian and the design field Hamiltonian representing the complete circuit [5]. Since each agent contributes to the design field Hamiltonian, and is, in turn, updated by the design field Hamiltonian representing all of the agents, the agents are synchronized to achieve the overall objectives.

The sensor data provides the current state of the system, such as the current layer that is being processed and other information relating to the performance criteria. Our tomograph engine collects measurements as the "nano-manufacturing printing" occurs and converts this into potential representations that update the Hamiltonian of the circuit in construction.

This tomograph data is in the form of polynomial approximations representing potential forms of the Hamiltonian in construction. The sensory data driving the tomograph engine are sequences of images generated by an electron microscope. While a spatial resolution of one nanometer is achievable with the tomograph approach, this aspect still needs a significant amount of experimentation and research.

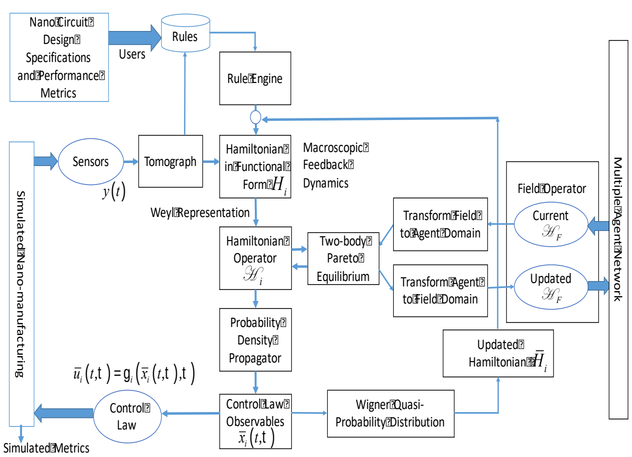

Figure 6: Quantum hybrid controller for agent *i* representing a tower in the nano-circuit.

The control variables in the quantum hybrid controller specify the devices (e.g., vertical dots, quantum wells, quantum wires, etc.) in each layer that comprise each tower, and the sequencing of the processes as each layer is constructed. The control law prescribes how each layer is constructed and is derived from the propagation of the Hamiltonian.

We propose to develop a computationally efficient algorithm for accurate and robust image processing based on a method called Speeded Up Robust Features, or SURF, for short. SURF is a local feature descriptor based on the sum of Haar wavelets around points of interest that are detected using an integer approximation of the determinant of the Hessian operator.

3 CONCLUSIONS

This paper overviews a technology for the design and implementation of programmable tower semiconductors. The design technique is based on continualizing a spectral representation of switching functions [1] and [3]. A future paper will give details of yield and complexity. The quantum representation is essential because the quantum effects at 4 nanometer footprint are dominant. The two layer device (probability and state transitions in Figure 5) represents the generic structure of each component in a tower and a bridge. Experimental work in that direction is being pursued at the present time.

REFERENCES

[1] Kushner, H. J., and Clark, D. S., "Stochastic Approximation Methods for Constrained and Unconstrained Systems," Springer-Verlag, NY, 1978.

[2] Kohn, W., Sandoval, M., Vettrivel, V., Cross, J., Knox, J., Talby, D., and Lazarus, M., "Cooperative Distributed Inferencing (CDI)" patents, US 14/746, 738, 751, 759, 777, 2015.

[3] Kohn, W., Brayman, V., Nerode, A. "Control synthesis in hybrid systems with Finsler dynamics," Houston Journal of Mathematics, Vol. 28(2) pages 353-375, 2002.

[4] Kohn, W. " Distributed Hierarchical Automata with some Applications to Genetics in Prokaryotes," PhD Dissertation, MIT EE, Cambridge, MA, 1978.

[5] Bussmann, S., Jennings, N. R., and Wooldridge, M., "Multiagent Systems for Manufacturing Control: A Design Methodology," Springer, 2004.

[6] Shen, J., Tang, T., and Wang, L. L., "Spectral Methods: Algorithms, Analysis and Applications," Springer, 2011.

[7] Barkalov, A., Titarenko, L., Kolopienczyk, M., Mielcarek, K., and Bazydlo, G., "Logic Synthesis for FPGA-Based Finite State Machines," Springer, 2016.

[8] Dereziński, J., and Gérard, C., "Mathematics of Quantization and Quantum Fields," Cambridge, 2013.

[9] Landau, R. H., Paex, M. J., and Bordeianu, C. C., "Computational Physics: Problem Solving with Python," Wiley, 2016.

[10] Abdessaied, N., and Drechsler, R., "Reversible and Quantum Circuits: Optimization and Complexity," Springer, 2016.

[11] Gronheid, R., and Nealey, P., "Directed Self-assembly of Block Co-polymers for Nano-manufacturing," Woodhead Publisher, 2016.