Simulation of Semiconductor Manufacturing Supply-Chain Systems with DEVS, MPC, and KIB

Dongping Huang, Hessam Sarjoughian[1], *Member, IEEE* Wenlin Wang, *Member, IEEE* Gary Godding,
Daniel Rivera, *Senior Member, IEEE* Karl Kempf, *Member, IEEE*, and Hans Mittelmann

Abstract

The dynamics of high-volume, discrete-parts semiconductor manufacturing supply-chain systems can be described using a combination of Discrete EVent System Specification (DEVS) and Model Predictive Control (MPC) modeling approaches. To rigorously describe the interactions between the discrete process model and its controller, another model called Knowledge Interchange Broker (KIB) is used. A robust and scalable testbed supporting DEVS-based manufacturing process modeling, MPC-based controller design, and the KIB$_{DEVS/MPC}$ interaction model is developed. A suite of experiments have been devised and simulated using this testbed. The flexibility of this approach for modeling, simulating, and evaluating stochastic discrete process models under alternative control schemes is detailed. The testbed illustrates the benefits and challenges associated with developing and using realistic manufacturing process models and process control policies. The simulation environment shows the importance of explicitly defining and exposing the interactions between the manufacturing and control subsystems of complex semiconductor supply-chain systems.

Index Terms

Discrete Event System Specification, Hybrid Simulation Testbed, Knowledge Interchange Broker, Model Composability, Model Predictive Control, Optimization, Semiconductor Supply-Chain Manufacturing

I. INTRODUCTION

Some best-in-class companies have already achieved 5-6% cost reduction by employing effective supply-chain management solutions [1], [2]. With the current scale of international supply-demand networks, a 5-6% difference between near-optimal and non-optimal supply-chain management can be worth hundreds of millions of dollars per year [3]. However, rigorously describing the complexity of supply-chain system dynamics and achieving even greater efficiency are needed [4], [5], [3], [6], [7], [8], [9]. Indeed, the complexity of supply-chain systems and the difficulties they pose in reducing cost and achieving higher efficiency are widely recognized. A key enabling capability is to develop a robust simulation-based testbed for analyzing and designing the complex interactions taking place inside semiconductor supply-chain systems.

A discrete supply-chain system – extending from suppliers through manufacturing and ending with customers – can be viewed to consist of two interacting manufacturing and controller subsystems. The roles of the subsystems in the semiconductor supply-chain system are illustrated in Figure 1. The chain begins with the manufacturing subsystem, which receives commands and sends its status from/to the control subsystem. For example, $inventory_{1,k}$ may be commanded to release a number of its inventory holdings to $factory_{1,k}$ given *present* variability in the discrete process and *future* variability in supply and demand. The manufacturing subsystem responds to inventory release commands and sends manufactured products according to factory rules such as maximum inventory holdings. The control subsystem receives material release target goals and process status updates and sends predicted inventory and work-in-progress trajectories to the manufacturing subsystem. The objective of the controller is to support timely, agile responses defined by short-term inventory goals and long-term supply and demand expectations.

D. Huang, H. Sarjoughian and G. Godding are with the Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85281; W. Wang is with the IT Manufacturing Systems, Freescale Semiconductor; D. Rivera is with the Department of Chemical Engineering Department, Arizona State University, Tempe, AZ; K. Kempf is with the Decision Technologies, Intel Corporation, Chandler, AZ; and H. Mittelmann is with the Mathematics and Statistics Department, Arizona State University, Tempe, AZ.

[1] Corresponding author.

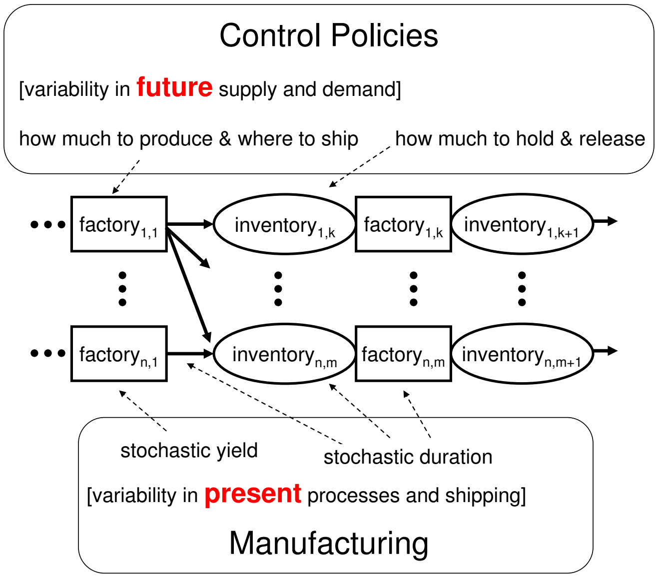

Fig. 1. Semiconductor manufacturing supply-chain system

Discrete-event simulation models for the manufacturing subsystem are needed. The models must characterize the inherent stochastic, nonlinear dynamics of factories and inventories —the basic elements of a manufacturing enterprise. Similarly, discrete-time dynamical models are required for generating control policies. Given the intricacies involved in manufacturing processes and (short- and long-term) control policies, their *interactions* must be handled in a systematic and principled fashion. Appropriate formulation of interactions between manufacturing and control subsystems should 1) produce accurate assignment of factory capacities that provide the right product at the right time to the intended customer, 2) reduce wasteful product capacity that may later be discarded, and 3) eliminate activities that increase throughput time, such as excessive changes in factory starts and setups, as well as frequent reprioritization of work-in-progress. The result is reduction in manufacturing, shipment, and management costs across the supply-chain enterprise —i.e., generating more revenue and improving customer satisfaction.

Having discrete-event process models and optimized control policies is necessary but insufficient for understanding the overall complexity of supply-chain systems (refer to Figure 1). This is due to the inherent properties of the semiconductor manufacturing processes (stochastic throughput time, different types of products, short product shelf-life, lower cost, and variability in availability of resources such as inventories and transportation) and the limitations in obtaining optimized plans (partial knowledge of future demand or supply). Hence, to tackle the kinds of complexity that arise in enterprise systems, it is crucial to explicitly and systematically account for the interactions taking place between the process and control subsystems. Only then is it possible to support interaction complexity and scalability between the discrete processes and control policies. Such a capability is essential for developing a robust simulation testbed for semiconductor supply-chain systems.

Toward building such a testbed, general and specialized modeling and simulation frameworks have been developed for analyzing and designing realistic discrete manufacturing processes [10]. Alternative approaches are commonly used for simulation-based experimentation of large-scale, complex semiconductor manufacturing processes. Models of discrete processes can range from linear feed-forward workflows to stochastic non-linear discrete processes with feedbacks [11]. To efficiently operate discrete processes under short- and long-term scenarios, optimization-based control models have been developed [12].

As mentioned above, to synthesize disparate discrete-event process models with model-based controllers, it is desirable to systematically support their integration. Advances in simulation interoperability and software engineering-e.g., HLA [13], agent-based modeling [5], and service-oriented architecture [14]—support combining different models as independent components. These methods and their supporting technologies employ programming and interoperability techniques to allow a simulator and a controller to exchange information. However, they lack model composability concepts. A desirable modeling framework needs to support *(i)* explicit modeling of the interactions between disparate models, *(ii)* scalable specification of the supply-chain systems, *(iii)* describing domain-specific knowledge founded on formal, general-purpose modeling constructs, *(iv)* realization of a robust simulation testbed, and *(v)* flexible and configurable experimentations.

To model the interactions between models that are described in disparate modeling formalisms, the concept of the Knowledge Interchange Broker, a multi-formalism modeling framework, has been introduced and applied to different domains [15], [16], [17], [18], [19], [20]. A Knowledge Interchange Broker (KIB) specifies input/output mappings, synchronization, concurrency, and timing properties for models described in multiple modeling formalisms [21]. The conceptual basis of the KIB is that disparities between different syntax and semantics need to be accounted for with a distinct model, thus enabling independent modeling of data and control interactions. In particular, rather than relying on software-based middleware concepts and often ill-defined techniques, interaction among models is specified as a pair that achieves *(i)* model composability and *(ii)* execution interoperability. The KIB, viewed as a model and an execution pair, enables two important activities-model validation and execution verification —which are necessary for building and conducting experiments.

To have a testbed for semiconductor supply-chain systems that satisfies the above requirements, a hybrid DEVS/MPC with KIB$_{DEVS/MPC}$ has been developed [17], [18]. This environment supports the Discrete Event System Specification (DEVS) [22] and Model Predictive Control (MPC) [23] approaches. The testbed is implemented using DEVSJAVA [25] as the discrete-event simulation tool and SIMULINK [24] with an efficient MATLAB-QP [25] as the MPC tool.

In this paper, we present the DEVS/MPC testbed and show the benefits of using the KIB in developing the testbed and carrying out experiments. Closely related work is summarized in the remainder of this section. Detailed descriptions of DEVS-based manufacturing process models and MPC-based control policies are presented in Sections 2 and 3, respectively. In Section 4, the KIB$_{DEVS/MPC}$ for the DEVS and MPC is presented. Section 4 focuses on the testbed capabilities to support (a) formulating complex interactions between the semiconductor manufacturing processes and predictive model-based control and (b) analyzing their combined dynamics. The design of the experiments, simulation scenarios, and analysis of the DEVS/MPC models are presented in Section 5. Conclusion and future work are presented in Section 6.

A. Related Work

The study of complex systems in terms of modeling their parts and integrating them has been the subject of research across different disciplines and application domains (e.g., [26], [27], [28], [29]). In the area of manufacturing and semiconductor supply-chain systems, some approaches and testbeds have been proposed for developing strategic plans that can effectively operate complex supply-chains [8], [30], [31], [32]. Strategic planning systems employing deterministic LP are useful, but they cannot account directly for the unavoidable variability of supply and demand. Mathematical optimization and in particular linear programming (LP) optimizers are commonly used [33], [34] to handle variability, and with recent progress in multi-echelon inventory theory, Dynamic Programming optimizers can be used for strategic plan construction [35]. Given target service levels, estimates of future supply and demand uncertainty, and historical forecast bias and error, these inventory algorithms compute safety stock positions and targets to be used as input to the LP optimizers. This safety stock is intended to buffer the expected variability in both supply and demand while executing the LP-generated multi-period plan.

The manufacturing subsystem is both the source of data and the target for the control subsystem (refer to Figure 1). While discrete event simulation of manufacturing processes is well established, its relationship with control remains only partially understood [36], [6]. Furthermore, the current state-of-the-art tools may only support ad-hoc supply-chain simulation modeling of modest complexity and scale (e.g., [10]). This is because these and many other approaches do not support defining the interactions between manufacturing processes and controllers based on model composability principles.

To address the limitation of controlling manufacturing dynamics using decision planning alone, tactical (short-term) control policies, in conjunction with (long-term) decision plans have been shown to handle the stochastic dynamics of manufacturing processes [3], [37], [12]. This approach has been developed and primarily tested under fluid assumptions (discretized continuous-time models). It aims to deal with the inevitable supply and demand variability that changes minute to minute, hour by hour, or day after day [3]. A manufacturing simulation model and one MPC model were developed in the SIMULINK/MATLAB environment [37]. The MPC was used with discrete-time manufacturing models to provide fine grain (daily) control, which surpasses the common planning-with-safety-stock approach. The interactions between the simulation model and the MPC were described using the

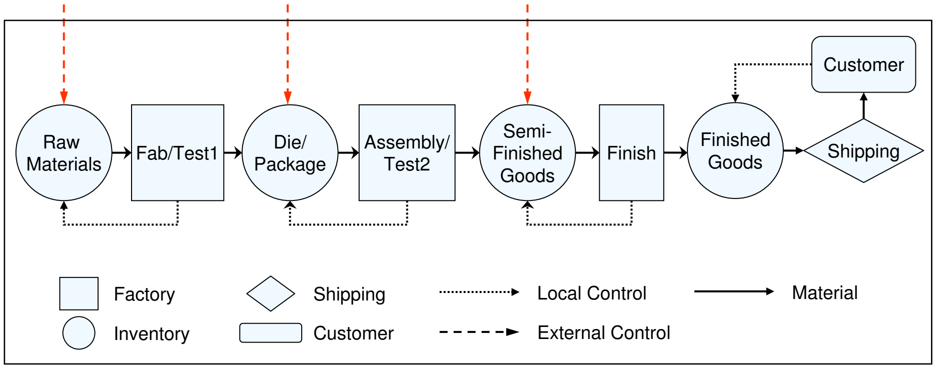

Fig. 2. Semiconductor Manufacturing Process Network

MATLAB macro programming language. With this approach, models of manufacturing processes are discrete-time and thus the role of MPC under the discrete-event setting cannot be evaluated. Furthermore, the integration of manufacturing and control is not grounded in the model composability concepts and methods mentioned above.

Another approach, aimed at a testbed for evaluating supply-chain decision control models, is under development [38]. It uses the DEVS framework and extends the software design of DEVSJAVA to allow DEVS and LP models to interoperate. The LP model is wrapped as an atomic DEVS model, which allows its execution in terms of exchanging input and output events. DEVS coupling and simulation protocols are used for exchanging input and output data between process and decision models. In this approach, input and output transformations have to be carried out inside the DEVS and LP models. Since the interactions must be divided between the simulation and optimization models, the modeling of interactions is constrained given the expressiveness supported by the DEVS and LP formalisms, scalability, and the need to rely on programming constructs. A key consequence of such an approach is the lack of robustness of the testbed, which directly impacts reusability and scalability of the process and optimization models and, more critically, their interactions. Other existing simulation-based approaches for supply-chain modeling and decision control assessment [5], [39] have similar kinds of shortcomings.

II. DEVS MANUFACTURING MODEL

To describe semiconductor supply-chain manufacturing networks, the manufacturing and assembly processes (factories), the intermediate inventory holding places, the logistics, and the customers must be modeled. The SCM model developed and used in this study is illustrated in Figure 2 [11], [18]. Raw silicon flows into the fabrication plant, wafers then flow to the assembly process where die are attached to packages, then product flows to the finish step where final configurations are made, and finally ends with the finished goods being sent to a customer. External instructions into the model specify how much and what type of product to release out of the inventory holding points into the next process step.

The semiconductor network is modeled with four types of entities: *factories*, *inventories*, *shipping*, and *customers*. Factory models can change the physical characteristics of the material flowing through it. Inventory models can hold and release material on command. Shipping can delay arrival of material to the next entity. Customers can generate orders, generate future order forecasts, consume material, and track order fulfillment. The entities are connected with 3 types of flows. Material flows model the actual physical entities flowing through the manufacturing network. Local control models the commands that are sent between entities internal to the simulation. External control models the commands that are generated outside the simulation. The material flow is modeled as discrete lots of wafers or units. A lot contains a batch of wafer or units. Before the assembly test process, a lot can contain up to 25 wafers of material. One wafer contains many die on a single piece of silicon. At the assembly step the wafers are cut up into individual die. At that time, lots contain quantities of individual die or units. The size of die unit lots is a configurable parameter to the simulation. Details of the functionality and available states for each of the entities are described next.

A. *Factory Model*

Factory (or process) models change the physical characteristics of material flowing through it. They can model stochastic throughput times and yield loss. A throughput time can be drawn from a distribution and applied to each incoming lot. The random number determines how long it will take the lot to finish the process. Similarly, yield percentages can be drawn from different distributions to determine yield losses or bin splits (e.g. speed or current leakage).

The output states available for external models are work-in-progress (*WIP*) and what material left in the last time period. The material leaving is referred to as actual outs (*AO*). The WIP can be reported in buckets. For example, if you configure WIP to be reported in 2 buckets, the output would be two values, the WIP in the front and back halves of the factory. The factory models are able to report their available capacity to other local simulation entities – e.g., see the arrow from the Finish factory to the semi-finished goods inventory shown in Figure 2 [17]. The available capacity is the maximum that can be started into the factory at any given time period.

In our semiconductor model above, there are 3 types of factory models. They are Fab/Test_1, the Assembly/Test_2, and the Finish. The Fab/Test_1 models the facility that fabricates circuitry onto raw silicon and performs the initial testing of the die on wafer. The Fab/Test_1 includes both throughput and yield distributions. Lots flowing through this process take varying amounts of time, and the sizes are variable due to the per lot yield losses. The Assembly/Test_2 entity characterizes the cutting of wafers, the assembly of die with a package, and the final test. This process consumes a die and a package. A single product flowing into this model can be split into multiple output products depending on the performance distribution measured by the test step. The Finish process sets the final configuration for the assembled material. All factories shown in Figure 2 have stochastic throughput times.

B. *Inventory Model*

The inventory model provides a holding place for material. Material that flows into the inventory will stay there until it is released. Releases can be generated from either external or local control commands. A release command has three parameters, what product to release, how much to release, and where it should be released to. Release messages can be configured to be queued up if they are not fulfilled. In there is not enough inventory to fill a release, or if the output is capacity constrained, the release can happen in the future when the constraining condition does not exist anymore. The inventory can receive a local control message specifying the maximum capacity of the connected entity. For the model in Figure 2, the local control message is used to control how much can be released into the factories. The factory model reports its maximum available capacity to the inventory; the inventory in turn does not allow the maximum to be exceeded. Each of the Raw Materials, Die/Package, Semi Finished Goods, and Finished Goods inventories has two externally available states, the current beginning on hand (*BOH*) inventory, and the actual amount released out (*AO*) in the previous time period. Inventories can hold different kinds and quantities of products.

C. *Shipping Model*

Shipping models can provide a stochastic delay for materials flowing between entities. The Shipping model is used to characterize air, land, or sea transportation and the associated customs delay. The shipping can have stochastic throughput times and yield losses. Yield losses are used to model damage and theft. The output states available for shipping entities are what is in transit, and how much actually shipped in previous time period. The in transit data can be reported in time buckets similar to WIP for the process model.

D. *Customer Model*

The Customer model can generate orders, future order forecasts, and track order fulfillment. The current orders are what is currently due and any unfilled orders from the past. The future order forecasts can be output externally as multi-dimensional matrices. For each product the customer has orders for, it can specify a vector of future date/quantity values. Forecast errors can be simulated using distributions. Order fulfillment is tracked by how many orders are filled on time or late. The supply network revenue and customer service levels can be measured from customer logs. For the model in Figure 2, the current actual orders are fed into the finished goods warehouse as release commands. The forecast is sent to the external controller. The controller objective is to manage the inventory releases, getting product to the appropriated holding points in time to maximize customer service levels while minimizing manufacturing costs.

III. MPC CONTROLLER MODEL

Controlling the inherent non-linearity and stochasticity of supply-chain system operations in semiconductor manufacturing is a fundamental goal of this work. This is necessary since a manufacturing network has modes of operations that need to be controlled based on periodic (e.g., hourly/daily) manufacturing process cycles to some other periodic (e.g., weekly/monthly) decision policies in the presence of large and partially unpredictable demand changes. Model Predictive Control (MPC) is a technique arising from the chemical process industries that has been demonstrated to accomplish effective constrained control of stochastic multivariable systems through an optimization-based formulation that incorporates feedback and feed-forward decision-making. As a tactical controller for semiconductor manufacturing supply-chain operations, MPC provides robust control and enhanced performance in the presence of significant supply and demand variability and forecasting errors while enforcing constraints on inventory levels and production and transportation capabilities [12], [37]. In the approach described in [12] and [37], a deterministic linear discrete-time model serves as a predictive model for the complex, stochastic, discrete-event model of the manufacturing process. The predictive model has a homomorphic relation to the DEVS process model [17], [18]. The discrete-time factory and inventory models are denoted as $M_{10}, M_{20}, M_{30}, I_{10}, I_{20}$, and I_{30}. For example, the factory responsible for manufacturing products for the "Finished Goods inventory" is modeled as "finish" with its simplified discrete-time model as "M_{30} finish". The MPC uses the DEVS models to represent the real manufacturing processes (i.e., representing the TPT-load function) and the simplified discrete-time models (i.e., representing a single-value TPT-load as opposed to a multi-value TPT-load [18]) for predicting future inventories which are used by the optimization model. The MPC design in combination with the simplified manufacturing process and detailed optimization must handle stochasticity and uncertainty of the system for some specified time horizon. The optimizer has a set of constraints and an objective function. The predictive (i.e., controller) model is based on the mass conservation relationships among the inventory, factory, and transportation models. For example, the mass conservation relationship between Die/Package inventory level (I_{10}) and Fab/Test1 WIP (M_{10}) are modeled shown below:

$$\begin{aligned} I_{10}(k+1) &= I_{10}(k) + Y_1 C_1(k-\theta_1) - C_2(k) \\ WIP_{10}(k+1) &= WIP_{10}(k) + C_1(k-\theta_1) - Y_1 C_1(k-\theta_1) \end{aligned}$$

The variables θ_1 and Y_1 represent the nominal (single-value) throughput time and yield for the Fab/Test1 node, while C_1 and C_2 represent the daily starts that constitute inflow and outflow streams for I_{10} and M_{10}. Similar relationships are provided for other nodes of the manufacturing process network.

For a given node topology of a semiconductor manufacturing process, the MPC policy manipulates the daily starts of the factories to satisfy the customer demand (both forecasted and unforecasted) while maintaining the inventories at desired levels. The MPC formulation used in this work corresponds to the algorithm developed in [37] and [40]. The general scheme, including its integration with the DEVS simulation model, is described as follows:

1) At initialization, the inventory set-point trajectories are specified. MPC model attributes such as average TPT and yield for each factory model are assigned. The resulting equations for the predictive model are organized into a linear discrete-time state-space model according to:

$$\begin{aligned} x(k) &= A\, x(k-1) + B_u\, u(k-1) + B_d\, d(k-1) + B_{d'}\, d'(k-1) \quad (1) \\ y(k) &= C\, x(k) + D_{d'}\, d'(k) + v(k) \quad (2) \end{aligned}$$

The input vectors u, d and d' represent manipulated variables, measured disturbances and unmeasured disturbances, respectively. The manipulated variable vector u physically corresponds to the starts in the manufacturing nodes of the supply-chain, while d represents the forecasted customer demand, which is treated as a disturbance signal with anticipation. y, the vector of measured inventory levels, is the controlled variable, while $v(t)$ is a vector of measurement noise signals. d', the unforecasted demand, is a stochastic signal which can be further described by its own state-space model.

The distribution of some stochastic and nonlinear behaviors, such as distribution of the TPT and yield, are assigned in at initialization. The TPT and Yeild state variables of the DEVS factory models are chosen at the start of the simulation (see Section V).

2) At each subsequent time interval k, the MPC algorithm receives the current inventory levels from the system simulation model. The controller calculations that determine the future starts for each factory involve the following two stages:
   - *Prediction:* Relying on the state-space model (2), the controller uses past, current, and forecasted values of inventories, starts, and demand to generate a vector $\mathcal{Y}$ of anticipated inventory levels over a prediction horizon p.

     $$\mathcal{Y} = [y(k+1)\ y(k+2) \cdots y(k+p)]^T \tag{3}$$

     The prediction algorithm is structured in this formulation to possess adjustable parameters that moderate the effects of forecasted and unforecasted demand and inventory setpoints; this is described later in this section.
   - *Optimization:* In this stage, a vector of future start changes $\Delta\mathcal{U}$ (also referred to as moves) over a move horizon m is calculated.

     $$\Delta\mathcal{U}(k) = [\Delta u(k+1)\ \Delta u(k+2) \cdots \Delta u(k+m)]^T \tag{4}$$

     This is accomplished by solving the optimization problem

     $$\min_{\Delta u(k|k) \ldots \Delta u(k+m-1|k)} J \tag{5}$$

     where the individual terms of J correspond to:

     $$\begin{aligned} J \;=\; & \overbrace{\sum_{\ell=1}^{p} ||Q_e(\ell)(\hat{y}(k+\ell|k) - r(k+\ell))||_2^2}^{\text{Keep Inventories at Inventory Planning Setpoints}} \\ + & \overbrace{\sum_{\ell=1}^{m} ||Q_{\Delta u}(\ell)(\Delta u(k+\ell-1|k))||_2^2}^{\text{Penalize Changes in Starts}} \\ + & \overbrace{\sum_{\ell=1}^{m} ||Q_u(\ell)(u(k+\ell-1|k) - u_{target}(k+\ell-1|k))||_2^2}^{\text{Maintain Starts at Strategic Planning Targets}} \end{aligned}$$

     $Q_u, Q_{\Delta u}, Q_e$ are penalty weights on the control error, move size and control signal, respectively; the selection of these weights enables the user to trade-off the ability of the algorithm to satisfy inventory setpoint targets r, adjust starts variability, and maintain starts close to strategic planning targets u_{target} that may be supplied by a higher-level strategic planning module. The problem per (5) can be solved by standard programming algorithms subject to linear inequality constraints. Meaningful constraints in the semiconductor manufacturing supply-chain problem include upper and lower limits on starts, inventories, Work-in-Progress, and their rate-of-change.
3) The starts at time k are sent to the process simulation model. Only the first set of calculated starts in the move horizon are implemented. Each inventory model then releases products to its downstream factory given its local control policy shown in Figure 2.
4) At the next time interval $k+1$, continue with step 2, using updated information to ultimately compute a new set of future starts over the move horizon. The process of updating information and recomputing an optimal set of decisions corresponds to a *receding horizon* implementation of the Model Predictive Control algorithm.

To better meet the requirements of the semiconductor supply-chain tactical control, the MPC algorithm is devised using a three-degree-of freedom formulation [37], [40]. Such functionality enables the user to independently address the performance requirements associated with meeting forecasted demand (anticipated measured disturbance rejection), inventory targets (setpoint tracking) and unforecasted demand (unmeasured disturbance rejection). The formulation includes adjustable parameters that directly influence the control system response for each performance objective in a manner that is both more intuitive and convenient than accomplished using penalty weights on the objective function. The details can be found in [37] and [40], but a brief summary follows:

Inventory Targets Tracking. Inventories are held in a supply-chain to buffer the variability from the stochasticity of the manufacturing process and customer demand. Inventory targets need to be maintained to hold enough safety stock that will handle these unexpected demand changes and avoid backorders. In most cases, the targets will be updated weekly or monthly and can be assumed as asymptotically step signals. To adjust the speed-of-response for setpoint tracking in the control system, a first-order discrete-time filter structure is used with α_i representing the tuning parameter corresponding to the i^{th} inventory target. The tracking speed for each inventory target can be adjusted independently by selecting α_i in the range $[0, 1)$. The smaller α_i, the faster the response that can be expected from the control system for tracking inventory targets.

Measured Disturbance Rejection. It was previously noted that an externally generated forecast of customer demand is used as a measured disturbance signal with anticipation in the control system. Because of the integrating nature of the dynamics in the supply-chain, asymptotically-step changes in demand result in ramp-like changes in the inventories. As a consequence, a filter structure for asymptotically ramp signals is required. The filter incorporates a user-specified value β_j (also within the range $[0, 1)$) that enables the user to independently influence the speed-of-response for each demand signal j. As with the inventory target tracking filter, the smaller the value for β_j, the faster the response will be.

Unmeasured Disturbance Rejection. The controllers' response to unforecasted demand is achieved by relying on a specially formulated observer, that, as with the forecasted demand, recognizes the integrating nature of these dynamics. The filter acts on the prediction error resulting from the difference between the predicted and actual inventory level values. γ_i corresponds to the adjustable filter gain parameter for each i^{th} inventory, which ranges between 0 and 1. As γ_i approaches zero, the controller increasingly ignores the prediction error and acts in a feedforward-only manner. The controller will attempt to compensate for all of the prediction error from the stochasticity and uncertainty if $\gamma_i = 1$; however, under these circumstances control action can be very aggressive and, consequently, the closed-loop system may be very unrobust. The tradeoffs associated with the proper selection of the filter parameter in this mode of the controller are described in [40] and further illustrated in Section V of this paper. A significant advantage of tuning with γ_i in lieu of the move suppression $Q_{\Delta u}$ is that using γ_i the user is able to influence each output variable independently; move suppression, on the other hand, by penalizing the inputs, consequently affects all of the outputs (albeit in a norm-optimal sense).

IV. KIB$_{DEVS/MPC}$ COMPOSITION MODEL

The DEVS and MPC modeling approaches complement one another and support simulation and analysis of real-world semiconductor supply-chain problems. To develop a model of a semiconductor supply-chain manufacturing network, the concept of Knowledge Interchange Broker (KIB) is used (see Section I). The KIB approach has been developed to formulate the integration of the DEVS and MPC models by modeling their interactions as shown in Figure 3. The KIB Model Specification as a separate model is used to systematically characterize the interactions between the disparate discrete-event manufacturing and optimization-based discrete-time control model. The KIB Execution Algorithm is used to execute the DEVS and MPC execution algorithms. The benefit of model composability and simulation interoperability is that the data and control described in each of the formalisms can maintain their own well-defined syntactic and semantic specifications in a neutral setting.

Depending on the domain of interest such as semiconductor manufacturing supply-chain, the general modeling constructs of the DEVS and MPC must be accounted for by the KIB in terms of a suite of specific input/output data mappings and transformations. The KIB$_{DEVS/MPC}$ model specification has been developed to ensure the correctness of the integrated structures of the DEVS and MPC models. The KIB$_{DEVS/MPC}$ execution algorithm accounts for the combined execution of the DEVS simulator and the MPC solver in such a way that it can correctly execute the DEVS and MPC model specifications. The composition specification supports simple to complex model interactions that have logically correct structures and can be executed under well-formed protocol. The generality of the DEVS, KIB, and MPC modeling approaches is used to systematically represent the semiconductor supply-chain domain knowledge into the DEVS, KIB, and MPC models. Data mappings and aggregation/disaggregation

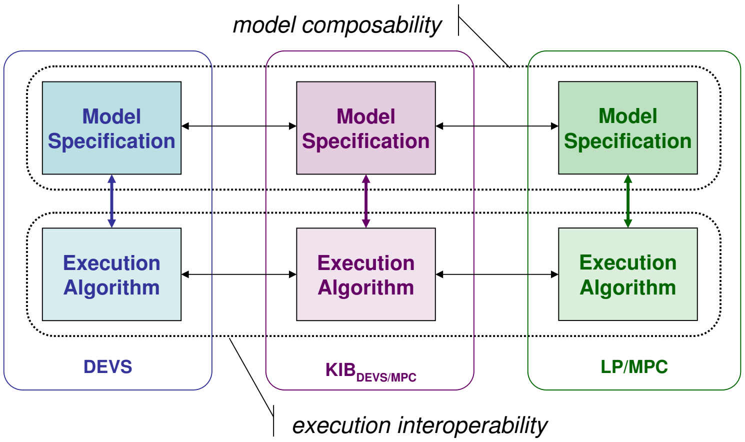

Fig. 3. The KIB Model Composability Approach

relationships with synchronized input/output data and control exchanges are handled in a scaleable setting which is key for developing and simulating large-scale SMC models.

A model's structure and therefore its interface has an abstract specification so that it can be suitable for different kinds of systems. The DEVS model interface is defined in term of ports and messages. The data contained in messages can have primitive or complex structures. In comparison, the MPC model interface is defined in terms of vectors. The modeling and simulation tool such as DEVSJAVA and MATLAB/SIMULINK allow modelers to specialize generic message and vector types for specific application domains including semiconductor supply-chain manufacturing. The KIB$_{DEVS/MPC}$ is devised to handle structural and behavioral differences between the DEVS and MPC models. A basic concept used in the KIB is to view each model in terms of its external interface and internal execution control. A model's external interface defines what input and output a model can receive and send. A model's internal execution enforces execution of the model according to its semantics. The KIB handles the differences between the DEVS and MPC external interfaces and synchronizing their execution algorithms as described at the end of this section. Next, it is shown how the different kinds of interactions between DEVS and MPC model is handled in a systematic fashion.

The KIB$_{DEVS/MPC}$ specification is defined as a set of nodes (see Figure 4) each of which corresponds to a DEVS model component type as defined in Section 2. Each node specifies the interactions between a DEVS component model (e.g., Semi-Finished Goods inventory) and the MPC. Each node model has its own designated mapping and transformation specification. Each node has two responsibilities: (1) mapping the outputs of the DEVS model component to inputs for the MPC model and vice versa and (2) transform the outputs of the MPC model to the inputs for the DEVS model components and vice versa [17], [18], [19], [20]. Consider a synchronous interaction cycle among the DEVS Semi-Finished Goods inventory, KIB Semi-Finished Goods node, and the MPC to have the start and end times of t_k and t_{k+1}, respectively [41]. The Semi-Finished Goods node receives the BOH status message from the DEVS Semi-Finished Goods inventory at t_k and receives the input release command u_i from the MPC before t_{k+1}. The BOH status is sent as output y_i to the MPC at time t_k and the release command message is sent to the DEVS Semi-Finished Goods inventory at t_{k+1}. When a message or a vector arrives at the KIB, it undergoes a suitable mapping based on the DEVS and MPC data and input data types (e.g., one or more DEVS message types are converted to a vector type) and the data content of a message or vector is transformed as desired (e.g., partitioning the MPC manipulated variable into individual inventory release commands categorized based on the type of material, destination, and quantity [20]). The message transformation for the Semi-Finished Goods node has the following specification.

$$BOH(material, quanity) \longrightarrow y_i$$

$$u_i \longrightarrow release(material, destination, quantity)$$

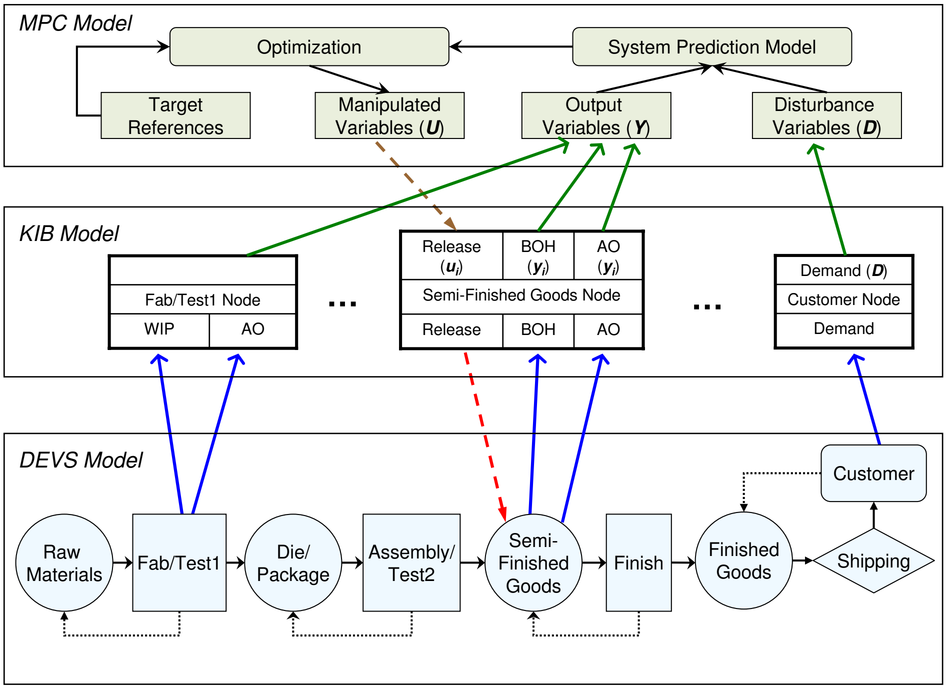

Fig. 4. Composition of the DEVS and MPC models with the KIB model

An inventory model such as Semi-Finished Goods holds one or more material types. The DEVS inventory model component sends its status (e.g., BOH) as output or receive release command as input for current time t_k or for a specific future time ($t_{k+m}, m \in N$). The status output is a collection of different kinds of material lots and each is associated with one time instance. This output has the same structure for any other inventory model (e.g., Die/Package). The KIB is specified to support the DEVS Inventory Model interface in general and specialized systematically for different kinds of inventories [11]. Similarly, the release command output generated by MPC is a vector of release commands for different types of material lots. The difference among the release commands is that the time associated with the lots to be released into factories can be the current execution time or a future time instance. The output and input structures for this inventory model are specified below. Given the differences between the input/output of the DEVS and MPC models, the Inventory Model node maps and aggregate status of the DEVS inventory model and disaggregate the release command of the MPC model [18], [20].

Input: $release\ command = \sum_n^t (material_n, destination_n, quantity_n, t)$

Output: $BOH\ status = \sum_n (material_n, quantity_n)$

where n is the number of lots and $t \in t_k, \cdots, t_{k+m}$. The specifications of the factory, shipping, and customer nodes follow the principles that are used for the Inventory Model node.

The synchronous interaction cycle between the DEVS process model and the MPC controller follows the following steps. The execution cycle consists of an initial step that initializes the variables that hold the input and output events and numerical values for the semiconductor manufacturing process model and the controller.

1) DEVS process model computes status events (messages) and sends them to the KIB,
2) KIB transforms and maps the status events to numerical status values (vectors),
3) KIB sends the numerical status values to the MPC controller,
4) MPC tactical controller computes the command numerical values and sends them to the KIB,
5) KIB transforms and maps the command numerical values to command events, and
6) KIB sends the command events to the DEVS process model.

V. EXPERIMENTS

In this section, a set of simulation experiments are described to show the dynamics of the interacting DEVS and MPC models with the KIB$_{DEVS/MPC}$ model. The correctness of the DEVS/MPC testbed for prototypical semiconductor supply-chain manufacturing (see Figure 2) with an optimization-based controller as well as their interactions has already been shown [18], [41]. Experiments were devised and simulated with predetermined inventory release commands and customer demand. Daily controller commands are defined as standard step and sinusoidal regimes. These factory starts correspond to idealistic customer demand which shows that the DEVS models correctly represent the fundamental dynamics of realistic semiconductor supply-chain manufacturing [11], [16], [17]. Customer demand profile with the average demand was set at 951 with a small variance between 939 and 968 starting from day 61 until the end of the simulation.

TABLE I
TPT-LOAD MODEL

Cases	Load	TPT (Day)		
	%	Min	Ave	Max
3-Level Distribution	(0 – 70]	30	32	34
	(70 – 90]	32	35	38
	(90 – 100]	35	40	45
5-Level Distribution	(0 – 70]	30	32	34
	(70 – 80]	31	34	36
	(80 – 90]	32	35	38
	(90 – 95]	34	37	42
	(95 – 100]	36	40	45

TABLE II
MANUFACTURING NETWORK MODEL CONFIGURATION

		DEVS Manufacturing Model							
			TPT Distribution (Day)			Yield Distribution (%)			
		Load	Min	Ave	Max	Min	Ave	Max	Capacity
Factory	FAB/Test$_1$	See Table I				93	95	97	70,000
	Assembly/Test$_2$	[0,100]	5	6	7	98	98.5	99	10,000
	Finish	[0,100]	1	2	3	98.5	99	99.5	5,000
	Shipping	[0,100]	1	1	1	100	100	100	2,500
Inventory		Maximum Capacity							
	Die/Package	20,000							
	Semi-Finished Goods	10,000							
	Finished Goods	10,000							
		Lot Size				Simulation Time (Day)			
	Others	50				638			

For modeling and simulation the combination of the DEVS and MPC models, the DEVS/MPC testbed is used [18], [41]. This testbed supports configuring, simulating, and analyzing the DEVS, MPC, and KIB models individually and collectively. Simulations show how well daily nonlinear and stochastic dynamics of the manufacturing process models can be controlled for given customer demands. Another profile with square input regime is devised such that the customer demand increases by 500 (53$\%$ percent variation compared with the average customer demand) from day 201 to day 400 (refer to Figures 5, 6 and 7). This profile examines realistic dynamics of the manufacturing supply-chain and the robustness of the controller given large increase and decrease in customer demand. The testbed allows determining how well the MPC can handle stochastically and nonlinearity of manufacturing processes given unanticipated changes occurring in customer demand. The tuning parameters of the MPC and resolution of the

DEVS process models (e.g., varying lot sizes) and formulation of the interactions in the KIB$_{DEVS/MPC}$ support systematic experimentation of realistic semiconductor supply-chain manufacturing.

All simulations are executed for a period of 577 days. A 5-level TPT-Load configuration for factory node is given in Table I. A set of parameters (e.g., TPT and Yield) are shown in Tables II [3], [18], [12]. A set of nominal parameters are also given in Table III for the discrete-time model and controller gains in the MPC. These parameters are *nominal TPT* and *Yield* and are consistent with the *average TPT* and *Yield* for the DEVS manufacturing process model. The *Target Points* define the desired inventory levels in the manufacturing process model. The tuning parameters α, β, and γ can be configured to control prediction error and deviation from target inventory levels. In the following experiments, the α parameter is set to zero for maximum tracking speed and the β parameter is set to zero for the fastest rejection of customer disturbance.

A. SMC DEVS/MPC Simulation Results

The DEVS/MPC testbed enables analyzing and evaluating the interactions between the control policies and the stochastic, nonlinear manufacturing process simulation. The DEVS model represent complexity and details of the supply-chain which is central to the MPC controller. The role of the MPC is to handle the prediction errors due to the differences between the actual and forecasted customer demand. Assuming the Finished Goods inventory is maintained close to the desired level, the customer demand is satisfied. Fine-grain control of factory starts can be achieved using higher resolution TPT-load levels and by varying the γ filter gain as shown in Figure 5 [18], [41]. While a filter gain greater than zero is necessary for feedback control, its value needs to be determined judicially in order to have an acceptable tradeoff between fast responses to the changes and stochasticity in the process models and preventing potential instability caused by large changes in factory starts. For example, the simulation results show the average starts for Fab/Test$_1$ vary only 0.2% when γ changes from 0.05 to 0.01. However, the maximum starts increase by 255% if γ is changed from 0.01 to 0.05.

TABLE III
MPC MODEL CONFIGURATION

MPC Model		
Factory	**Nominal TPT**	**Nominal Yield (%)**
Fab/Test1$_1$	35	95
Assembly/Test$_2$	6	98.5
Finish	2	99
Shipping	1	100
Inventory	**Target Points**	
Die/Package	5,721	
Semi-Finished Goods	2,856	
Finished Goods	1,787	
Controller	**Settings**	
α	0	0
β	0	0
γ	0.01	0.05

The robustness of the MPC is subject to the degree of nonlinearity and stochasticity of the plant model (i.e., the manufacturing process model depicted in Figure 2) and the linear, time-invariant model of the plant used in MPC. For example, the customer demand can be changed by 50% and thus cause significant nonlinearity in Fab/Test$_1$ due to the TPT-load model. Factory models are configured with large capacities (i.e., $C_{\text{Fab/Test}_1} = 70,000$, $C_{\text{assembly/Test}_2} = 10,000$, and $C_{\text{Finish}} = 5,000$) to handle large increases in customer demand. As shown in Figures 5 and 6, given a 3-level TPT-load, the Die/Package inventory has transient dynamics due to the significant change in the upstream Fab/Test$_1$ factory model. Ideally, when Fab/Test$_1$ maintains its load within specific range (e.g., ($load \in [72\%, 76\%]$), the average TPT can be kept at the average of 35 days in the process simulation model. Accordingly, such average TPT value is consistent with the corresponding nominal TPT parameter configured in the MPC model. However, due to the significant increase in customer demand, starts on Fab/Test$_1$ are increased.

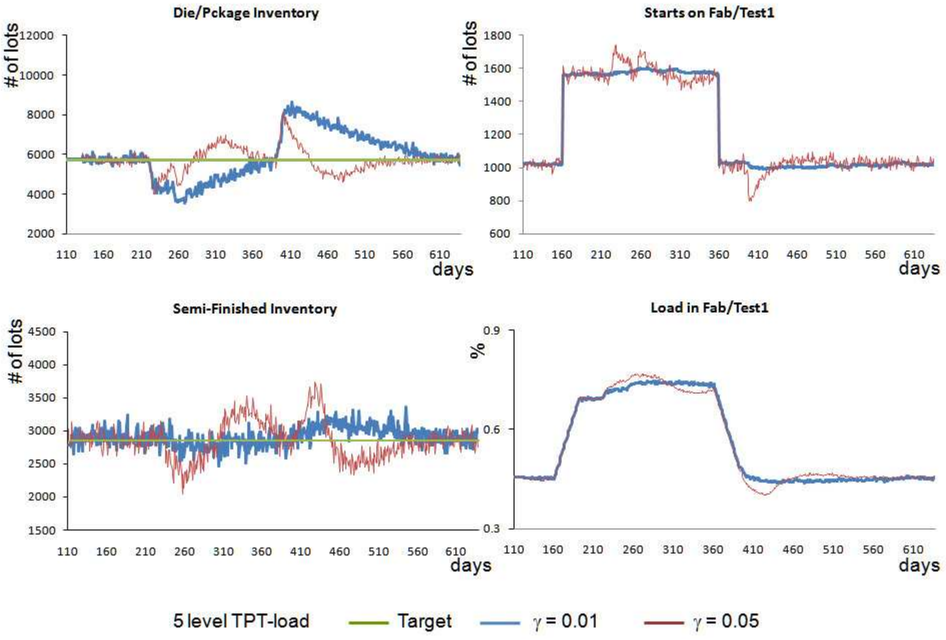

Fig. 5. Effect of varying γ on inventory and factory starts with 5 TPT-load level

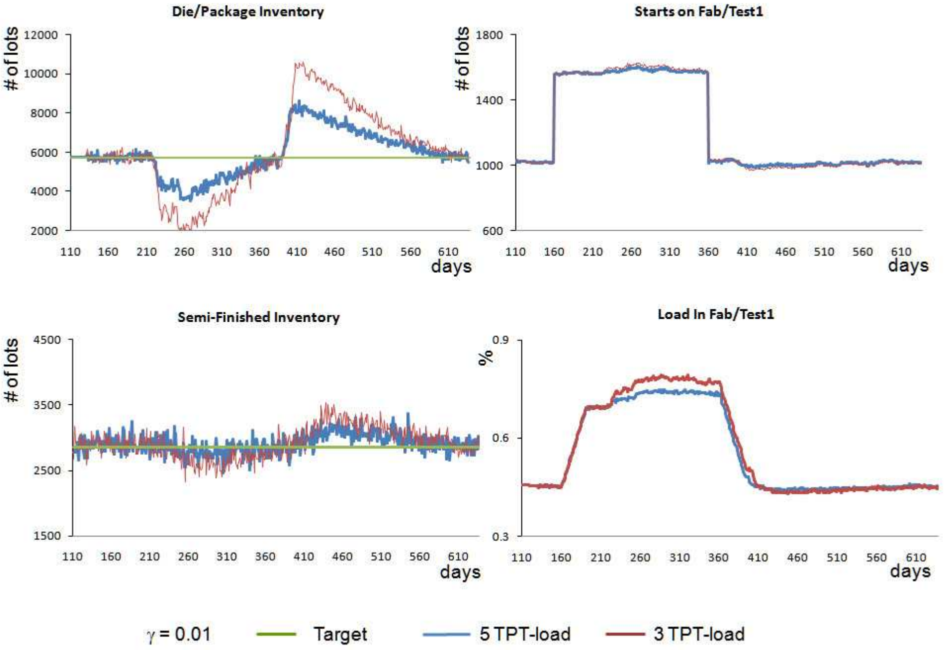

Fig. 6. Effect of varying TPT-load on inventory and factory starts with $\gamma = 0.01$

Consequently, the load in the factory model increases. Since the run-time TPT is calculated based on the load, a heavier load can cause longer delays in the Fab/Test$_1$ model. Longer delays impact the inventory level of the downstream Die/Package model. Similar transient behaviors occur when customer demand is decreased by 50% in one day.

Given that the nominal TPT value used in the MPC model is deterministic, the difference between this TPT and the average run-time TPT in the DEVS simulation model can be very large. To demonstrate the impact of large differences between nominal and actual TPTs, experiments with a 5-level TPT-load model were conducted (see Table I). Under more accurate TPT-load model, the Fab/Test1 behaves significantly better as shown in Figures 5 and 6. The 5-level TPT-load function represents more realistically the behavior of the factory model, which in turn

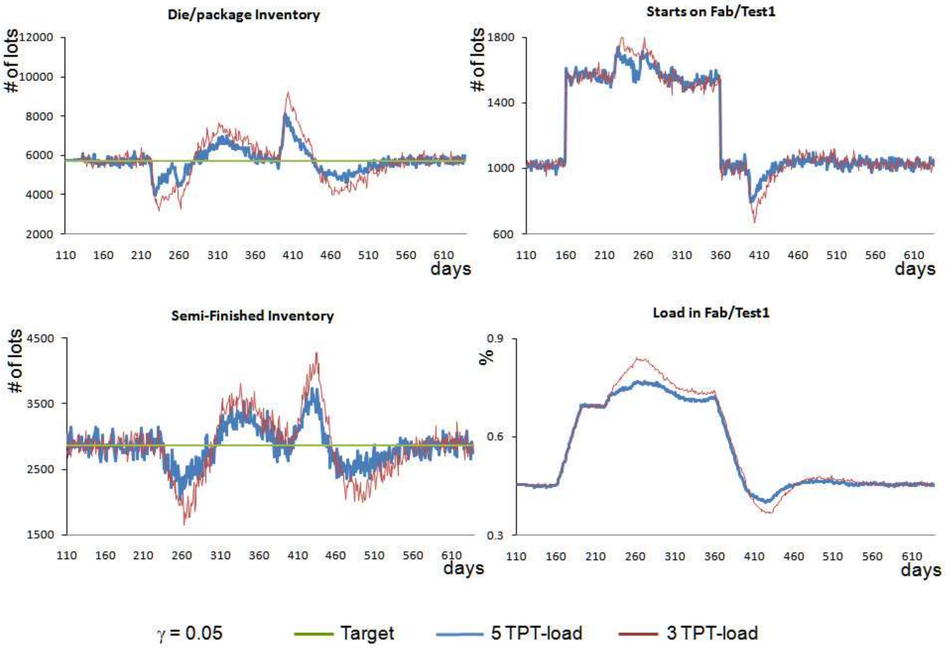

Fig. 7. Effect of varying TPT-load on inventory and factory starts with $\gamma = 0.05$

results in providing more accurate status updates to MPC. Another consideration in these experiments is the role of factory starts. The variance in the factory starts plays a major role in choosing small γ values since factory starts are smoother (and less thrashing) when γ is closer to zero and the inventories can be used to act as buffer to handle demand variability.

Comparing the three and five TPT-load level responses with $\gamma = 0.01$ and $\gamma = 0.05$ shows the higher TPT-load level has smoother nonlinear response that may agree better with actual semiconductor operations. The MPC control responds differently given varying plant dynamics. The more accurate use of the TPT in the DEVS simulation models produces better transient dynamics when there is a step change in customer demand. Both factory starts and inventories are subject to less variation when the three TPT-load level is used. But with the same range of TPT-load, the five TPT-load level gives less variability in the plant so that MPC can generate smoother starts command to track the inventory target more closely. When the five TPT-load level is used, $\gamma = 0.05$ gives better transient process on Die/Package inventory in terms of fast response and closely tracking targets. The reason is that compared to $\gamma = 0.01$, more prediction error is used by controller to calculate the starts decision to address the forecast error. The starts then can change more aggressively to bring the inventory back to the target faster. As a result, the variability on all the starts and Semi-Finished inventory is larger when using $\gamma = 0.05$ than $\gamma = 0.01$. This also demonstrates the tradeoff between the response speed and system robustness. Finally, the simulation results show the performance gains of the semiconductor supply-chain manufacturing in terms of maximum deviation from setpoint and speed of response for $\gamma = 0.05$, at the expense of a more aggressive starts profile.

VI. CONCLUSIONS

Simulation of semiconductor manufacturing systems requires modeling discrete processes combined with control policies. We have shown the importance of simulating inherently distinct manufacturing processes and control polices using the DEVS/MPC testbed. This novel testbed brings together the complementary DEVS and MPC modeling approaches using the $\text{KIB}_{DEVS/MPC}$. A capability of this testbed is independent evaluation of the manufacturing processes, control schemes, and their interactions. The experiments are grounded in a sound hybrid DEVS/MPC modeling framework that supports flexibility for observing and analyzing how discrete-event processes and control policies affect each other. Experiments revealed the impact of realistic non-linearity and stochasticity of the manufacturing dynamics and its importance in designing suitable tuning control parameters. The simulated responses show the ability of the MPC control algorithm based on a linear time-invariant model to maintain stable, robust operation under conditions of nonlinearity and uncertainty in the manufacturing plant dynamics. The simulation studies helped uncover and explain complex relationships between control policies and manufacturing processes. The hybrid DEVS/MPC framework and its testbed are suitable to be extended for distributed simulation and thus supporting large-scale, complex analysis and design of semiconductor supply-chain manufacturing systems. Another area of future research motivated by this study is to develop nonlinear fluid models of the semiconductor manufacturing supply chain as the basis for novel nonlinear MPC controllers. Complementary future research is aimed at developing and simulating manufacturing topologies with multiple controllers.

ACKNOWLEDGMENT

This research is supported by grants from Intel Research Council and in part by NSF Grant No. DMI-0432439. We would like to acknowledge Kirk Smith of the Intel Corporation and Dieter Armbruster of the Mathematics and Statistics Department at Arizona State University for fruitful discussions on the study of semiconductor supply-chain manufacturing systems.

REFERENCES

[1] ASCET, "Achieving supply chain excellence through technology," `http://www.ascet.com`, 2003.

[2] SimulationDynamics, "Importance of supply chain management," `http://www.simulationdynamics.com/Sc/SupplyChainImportance.htm`, 2003.

[3] K. Kempf, "Control-oriented approaches to supply chain management in semiconductor manufacturing," in *Proceedings of IEEE American Control Conference*, Boston, MA, USA, 2004, pp. 4563–4576.

[4] J. Shapiro, *Modeling the Supply Chain*. Duxbury, 2001.

[5] J. Gjerdrum, N. Shah, and L. Papageorgiou, "A combined optimization and agent-based approach to supply chain modelling and performance assessment," *Production Planning & Control*, vol. 12, no. 1, pp. 81–88, 2001.

[6] J. W. Fowler and O. Rose, "Grand challenges in modeling and simulation of complex manufacturing systems," *Simulation Transactions*, vol. 80, no. 9, pp. 469–476, 2004.

[7] S. D. Wu, M. Erkoc, and S. Karabuk, "Managing capacity in the high-tech industry: A review of the literature," *The Engineering Economists*, vol. 50, pp. 125–158, 2005.

[8] P. Lendermann, N. Julka, B. Gan, D. Chen, L. McGinnis, and J. McGinnis, "Distributed supply chain simulation as a decision support tool for the semiconductor industry," *ACM Transactions on Modeling and Computer Simulation*, vol. 15, pp. 316–345, 2005.

[9] A. Velosa, "Semiconductor manufacturing: Boom busts, and globalization," *The Bridge, National Academy of Engineers*, vol. 35, 2005.

[10] M. Semini, H. Fauske, and J. O. Strandhagen, "Applications of discrete-event simulation to support manufacturing logistics decision-making: A survey," in *Proceedings of Winter Simulation Conference*, Monterey, CA, USA, 2006, pp. 1946–1953.

[11] R. Singh, H. S. Sarjoughian, and G. W. Godding, "Design of scalable simulation models for semiconductor manufacturing processes," in *Summer Computer Simulation Conference*. San Jose, CA: SCS, 2004, pp. 235–240.

[12] W. Wang, D. E. Rivera, and K. Kempf, "Model predictive control strategies for supply chain management in semiconductor manufacturing," *International Journal of Production Economics*, vol. 107, pp. 56–77, 2007.

[13] HLA, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Federate Interface Specification*. IEEE, 2000.

[14] S. Narayanan and S. McIlraith, "Analysis and simulation of web services," *Computer Networks*, vol. 42, pp. 675–693, 2003.

[15] H. Sarjoughian and J. Plummer, "Design and implementation of a bridge between RAP and DEVS," Computer Science and Engineering, Arizona State University, Tempe, AZ, 2002, Internal Report, Computer Science and Engineering, Arizona State University, Tempe, AZ.

[16] G. Godding, H. Sarjoughian, and K. Kempf, "Multi-formalism modeling approach for semiconductor supply/demand networks," in *Proceedings of Winter Simulation Conference*, Washington DC, USA, 2004, pp. 232–239.

[17] H. S. Sarjoughian, D. Huang, W. Wang, D. E. Rivera, K. G. Kempf, G. W. Godding, and H. D. Mittelmann, "Hybrid discrete event simulation with model predictive control for semiconductor supply-chain manufacturing," in *Proceedings of the Winter Simulation Conference*, Orlando, FL, USA, 2005, pp. 255–266.

[18] D. Huang, H. S. Sarjoughian, G. W. Godding, D. E. Rivera, and K. G. Kempf, "Flexible experimentation and analysis for hybrid DEVS and MPC models," in *Proceedings of the Winter Simulation Conference*, Monterey, CA, 2006, pp. 1863–1870.

[19] G. R. Mayer and H. S. Sarjoughian, "Complexities of simulating a hybrid agent-landscape model using multi-formalism composability," in *Proceedings of Agent Directed Simulation, Spring Simulation Multi-conference*, Norfolk, VA, USA, 2007, pp. 161–168.

[20] G. Godding, H. Sarjoughian, and K. Kempf, "Application of combined discrete-event simulation and optimization models in semiconductor enterprise manufacturing systems," in *Proceedings of Winter Simulation Conference*, Washington DC, USA, 2007, p. accepted.

[21] H. S. Sarjoughian, "Model composability," in *Proceedings of the Winter Simulation Conference*, Monterey, CA, 2006, pp. 149–158.

[22] B. Zeigler, H. Praehofer, and T. Kim, *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*, 2nd ed. Academic Press, 2000.

[23] S. Qin, "An overview of industrial model predictive control technology," `http://www.che.utexas.edu/~qin/cpcv/cpcv14.html`, 1996.

[24] Mathworks, "MATLAB/Simulink," `http://www.mathworks.com`, 2005.

[25] R. J. Vanderbei, "An interior point code for quadratic programming," *Optimization Methods and Software*, vol. 11, pp. 451–484, 1999.

[26] P. Mosterman and H. Vangheluwe, "Guest editorial: Special issues on computer automated multi-paradigm modeling," *ACM Transaction on Modeling and Computer Simulation*, vol. 4, no. 12, pp. 249–255, 2002.

[27] F. Barros and H. Sarjoughian, “Component-based modeling and simulation, guest editorial,” *Simulation: Transactions of the Society for Modeling and Simulation International*, vol. 80, pp. 319–320, 2004.

[28] J. Eker, J. Janneck, E. A. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. R. Sachs, and Y. Xiong, “Taming heterogeneity–the ptolemy approach,” *Proceedings of the IEEE*, vol. 91, pp. 127–144, 2003.

[29] P. Davis and R. Anderson, *Improving the Composability of Department of Defense Models and Simulations*. Santa Monica, CA: RAND, 2004.

[30] C. S. Chong, P. Lendermann, B. Gan, B. Duarte, J. Fowler, and T. Callarman, “Analysis of a customer demand driven semiconductor supply chain in a distributed simulation testbed,” in *Proceedings of Winter Simulation Conference*, Washington DC, USA, 2004, pp. 1902–1909.

[31] J. D. Schwartz, W. Wang, and D. E. Rivera, “Simulation-based optimization of Model Predictive Control policies for inventory management in supply chains,” *Automatica*, vol. 42, no. 8, pp. 1311–1320, 2006.

[32] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi, *Designing and Managing the Supply Chain*. New York: McGraw Hill, 2000.

[33] W. J. Hopp and M. L. Spearman, *Factory Physics: Foundations of Manufacturing Management*. New York: McGraw Hill, 1996.

[34] S. Chopra and P. Meindl, *Supply Chain Management: Strategy, Planning, and Operation*. New Jersey: Prentice-Hall, Upper Saddle River, 2001.

[35] S. C. Graves and S. P. Willems, “Optimizing strategic safety stock placement in supply chains,” *Manufacturing and Service Operations Management*, vol. 2, no. 1, pp. 68–83, 2000.

[36] S. Ramakrishnan, S. Lee, and R. A. Wysk, “Implementation of a simulation-based control architecture for supply chain interactions,” in *Proceedings of the Winter Simulation Conference*, San Diego, CA, USA, 2002, pp. 1667–1674.

[37] W. Wang, “Model predictive control strategies for supply chain management in semiconductor manufacturing,” Ph.D. dissertation, Dept. of Chemical and Materials Engineering, Arizona State University, 2006.

[38] Y. Xu and S. Sen, “A distributed computing architecture for simulation and optimization,” in *Proceedings of the Winter Simulation Conference*, Orlando, FL, USA, 2005, pp. 365–373.

[39] J. Venkateswaran and A. Jones, “Hierarchical production planning using a hybrid system dynamic-discrete event simulation architecture,” in *Proceedings of the Winter Simulation Conference*, Washington D.C., USA, 2004, pp. 1094 – 1102.

[40] W. Wang and D. E. Rivera, “Model predictive control for tactical decision-making in semiconductor manufacturing supply chain management,” *IEEE Transactions on Control Systems Technology*, in press.

[41] H. S. Sarjoughian, D. Huang, G. W. Godding, D. E. Rivera, W. Wang, K. G. Kempf, and H. D. Mittelmann, “Hybrid discrete-event process simulation and model predictive control with knowledge interchange broker,” *IEEE Transaction SMC, Part A*, submitted July, 2007.