Enterprise Dynamics Via Non-Equilibrium Membrane Models

Wolf Kohn

Hynomics Corp., 10632 NE 37th Circle, Building 23, Kirkland, WA 98033-7021 & wk@hynomics.com

Vladimir Brayman

Department of Electrical Engineering, University of Washington, Campus Box 352500, Seattle, WA 98195-2500 & vbrayman@ee.washington.edu

James A. Ritcey

Department of Electrical Engineering, University of Washington, Campus Box 352500, Seattle, WA 98195-2500 & ritcey@ee.washington.edu

(Received 2000)

Abstract. This paper describes a distributed dynamic model of enterprise systems via a network of elements which are abstractions of biological membranes. Membrane characteristics such as active sites controlling the flow of substances correspond to local feedback laws in the elements of the supply chain of the enterprise. Flow conservation and chemical reactions of substances across the membrane are abstracted to represent component flow interaction in the supply chain. The model characteristics are illustrated with a simulation example. This model methodology is completely encodable. It provides a blueprint for highly automated model generation of enterprise systems, and for on-line generation of continuous repair implementations of planning, scheduling and execution applications. The proposed embedded distributed control system allows for the realization of diverse optimization strategies because a given criterion is approximated by a generic criterion via the penalty method. The control system also satisfies network element constraints and inter-element synchronization requirements.

1. Introduction

In November of 1993, when we first considered the application of Networks of Hybrid Control agents to economics problems [1], we were at the stage of ideas. In that work we modeled the economic activity of a firm as an entity which is simultaneously playing several different types of classical economic games, referred to as *base* games. The selected base games were min-max, Pareto, Stackelburg, Team, and Nash. We showed that the hybrid systems theory, applied to this game situation, coupled with the underlying mathematical theory of our hybrid agents [2], allows the generation of an optimization policy that corresponds to the chattering combination of the policies optimizing the base games.

We discovered that this modeling approach was deficient in several ways. Particularly, it did not provide for adequate representation of structural and dynamic *constraints;* such as capacity and flow constraints, a crucial element in any dynamic modeling schema for enterprise systems.

In this paper, we will present a new modeling schema for enterprise systems based on an observation that the dynamics of the *supply chain* of the enterprise can be characterized with formalisms coming from the *non-equilibrium thermodynamics* of open systems. Biological membranes [16], [17], [18] provide a very general model for open thermodynamics systems in which the mass flow of each species is tightly regulated by a built-in feedback control system [15]. This aspect appears also in each of the nodes of supply chain models. The dynamics of supply chain elements and the dynamics of the species flow across biological membranes can be derived from the mass and flow (momentum) *conservation principles.* Models of interaction among species (chemical reactions) flowing across membranes are obtained from a combination of formal principles from micro-kinetics theory [19] and phenomenological mechanisms. We will show that these principles and mechanisms provide models for goods-and-services interactions at the nodes of supply chain systems.

Our particular modeling focus is on those aspects of the enterprise that involve the generation of policies for the *flows* of goods or services throughout the enterprise and its market environment. These policies may be part of a planning, scheduling, or execution formalism. For example, the proposed model may be providing the dynamic constraint for an application computing a feedback law [10] that corresponds to the criterion of maximizing profit of the enterprise.

In this paper, we model the supply chain of an enterprise as a controlled network of *membrane elements* in which goods, capital, information, and expertise flow in or out, controlled by tuned active sites that are coupled with each other.

The principles that we used to formulate our model are based on the conservation principles and the entropy model that couples to each flow a corresponding generalized control force [11].

Many of the flows or forces characterizing enterprise dynamics take values in discrete domains. Furthermore the time scale may be measured in discrete steps. We have developed a formalism, called *continualization* [3],[4]. This formalism permits us to approximate the classical highly combinatorial models and strategies of supply chain systems with models expressed by *smooth differential systems.* Thus, discrete-valued flows and model states are approximated by continuous functions in our model. The formalism also provides several quantization mechanisms for reverting to discrete domains, so that results from simulations or optimization exercises can be expressed in their native domains.

The paper establishes a modeling format for the enterprise by building an annotated directed graph whose nodes represent enterprise activities and whose edges represent flows. This graph is the blueprint for the enterprise membrane network.

The proposed modeling strategy will be illustrated with a simple example.

For the purposes of expediency and to keep the description of our modeling approach short, we will omit the proofs of some of the mathematical statements and, instead, we will give references in each case.

The rest of the paper is organized into 6 sections. Section 2. discusses a model of the supply chain as a network of membranes. Section 3. presents the dynamic model of a membrane element. Section 4. defines the flow control mechanism for each membrane's elements and its synchronization in the network. Section 5. illustrates our modeling strategy with an example extracted from a case study [12] of a real enterprise system. Section 6. describes simulation results which are used to illustrate general characteristics of our modeling approach. Finally, Section 7. provides a summary of our approach, presents some issues for future research and draws conclusions.

2. Supply Chain as a Network of Membranes

In this section we provide an overview of classical elements of supply chain models and their representation as networks of membranes. Our objective here is to provide for the notation that we need in the next sections to introduce our model. The interested reader may find more information about supply chain systems and their modeling in [12], [13].

Let a supply chain be represented by a directed graph $G(\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is a set of nodes and $\mathcal{A}$ is a set of arcs. The arcs are of two types: flow arcs and information arcs. See Figure 1.

In G, each node represents a supply chain process element: supplier, plant, distribution center, or customer. In Figure 1 the solid line arcs represent interdependencies between process elements. We assume that each process element can process and store both materials and information. Moreover, each process element is equipped with some decision-making capabilities. In our modeling schema, the dynamics of each process element are modeled by a membrane whose dynamics is defined in Section 3. The dash line arcs in Figure 1 represent information flow in the direction of the arrows. We note that for each flow arc connecting a source node with a destination node, there is an information arc connecting the destination node to the source node.

Supply chain management focuses on coordination of material and information flows in a network of manufacturing and distribution facilities [14].

Our modeling schema assumes a criterion for the operation of the supply chain which is incorporated into the dynamics of each node. To accomplish this, we have to distribute the criterion across the nodes to generate local feedback laws (a feedback law is an operator that maps the internal state space of a node into the input flow space of this node; see Section 4.). We note that this local coupling

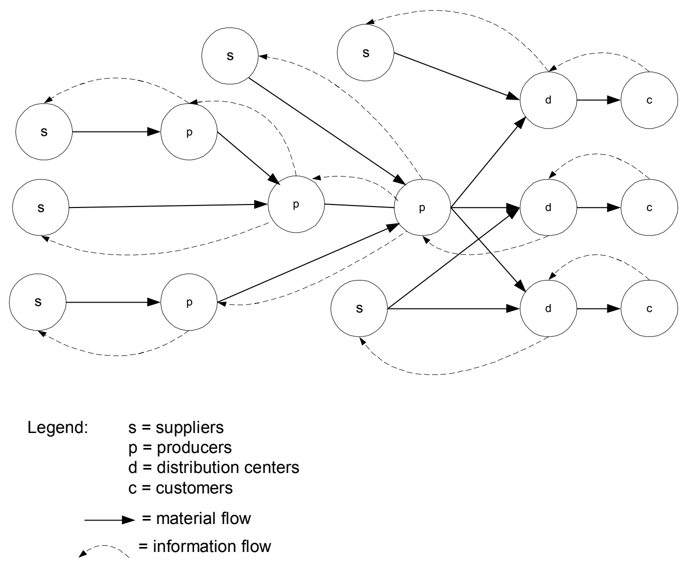

Fig. 1: Supply Chain Graph

of the criterion and the dynamics is part of the membrane formalism [15]. This property is also characteristic of supply chain systems.

The decomposition procedure will not be discussed in this paper. We will present it in a future paper currently in preparation.

For the purposes of illustration of our model, in this paper we are assuming that the local *distributed* operating criterion at each node is the one that maximizes the difference between revenue and cost (profit) at the node. However, other criteria may be considered with appropriate modifications.

The dynamic model of each node of the graph, at the chosen level of aggregation, is determined by the dynamic relations between the input and the output flows to the node. The model of this dynamics is a membrane element (ME) whose internal structure will be presented in the next section.

Figure 2 shows the input and output flows of a membrane element. The inputs corresponding to goods or services are denoted by J_1^j, $j = 1, ..., m$. The inflow of money is denoted by $\dot{M}_1$, and the inflow of information by I_1. The output flows of goods or services are denoted by J_2^i, $i = 1, ..., n$, the outflow of money is denoted by $\dot{M}_2$, and the outflow of information I_2. The function of the membrane element is to orchestrate the transformation from input to output flows such that the local pre-defined criterion is extremized. A unique feature of our model is that the *capacity constraints* and *flow constraints* are embedded into the model.

Now we will provide some brief descriptions of the different flows characterizing

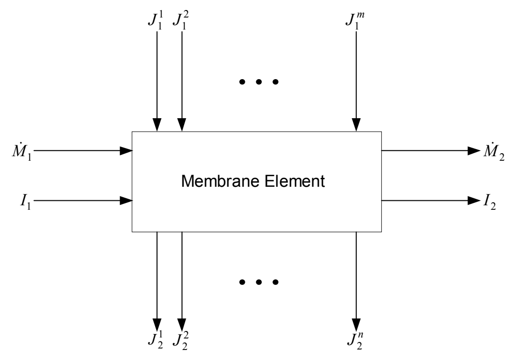

Fig. 2: Input-Output Model of a Membrane Element

the input-output behavior of a membrane element.

Flow of Goods or Services Input and output flows of goods and services model the transformation function of the corresponding node in the enterprise process. Therefore, in general they are not the same. For example, ME can take as an input raw materials, process them, possibly creating some intermediate products, and output partially finished goods. We assume available a description of the dynamics and constraints of the transformation. At the chosen level of abstraction, this description should have enough details so that our continualization procedures [3] can generate the canonical model for the node. See Section 3.

Flow of Money The inflow of money, $\dot{M}_1$, consists of the financial inflows (e.g. investments, loans, interests, etc.) and the money earned by selling goods or commodities. The outflow of money, $\dot{M}_2$, consists of the financial outflow (e.g. taxes, losses on investments, payments not related directly to the production, etc.), production costs (salaries, usage and rental of equipment, etc.), and cost of materials purchased.

Information Flow The inflow of information consists of the information about the market prices of commodities, information about the demand for a certain commodity, and a know-how information associated with R&D and training of the work-force. The outflow of information is usually associated with the attempts to increase the market value of the products, e.g. advertisements.

Capacity Constraints We assume that there are capacity constraints C_j associated with the output flows J_2^j.

Storage Constraints We assume that there is a finite volume available for storage of intermediate and/or final products.

Criterion Profit maximization (as an example)

3. Membrane Dynamics

In general, the internal dynamics of a membrane element (ME) is characterized by three coupled mechanisms: the flow mechanism, the production process, and the control system. See Figure 3. The flow mechanism models the transport of goods or services, the flow and storage capacities of the membrane element for each species involved, and, in general, the flow of information across the membrane element.

The production process mechanism models the interaction of species inside the membrane element. We use the micro-kinetics *formalism of chemical reactions* to express the interaction between goods or services inside the membrane element. On a purely empirical basis, this mechanism has shown to produce the families of dynamic models whose parameters can be adjusted to represent accurately the flow, velocity of money, and storage dynamics of both discrete and continuous supply chain elements. In other words, the analogy of interaction between goods and services in our membrane elements and chemical species in biological membranes is fruitful [6]. Furthermore, this formalism provides for 'rate' parameters which can be adjusted to realistically approximate capacity and flow rate constraints.

The control system regulates the flow across the membrane element as a function of the propagated demand and the input flow. We will discuss this system in Section 4.

In this section we will describe the membrane modeling formalism applied to the production process mechanism of an ME.

By selecting the number of chemical reaction steps, the proposed formalism allows the modeler of supply chain systems to express any desired level of aggregation. This, of course, is also true in expressing models of chemical reactions in the biological environment. We characterize the production process mechanism by *production rules* of the form:

$$a_1^j x_1 \circledast a_2^j x_2 \circledast \ldots \circledast a_n^j x_n \xrightarrow{k_j} b_1^j x_1 \circledast b_2^j x_2 \circledast \ldots \circledast b_n^j x_n, \; j = 1, \ldots, r \qquad (1)$$

where $x_1, \ldots, x_n$ are the labels of the goods , services, or money flowing through the element, a_i^j, b_i^j, $i = 1, ..., n$, $j = 1, ..., r$ are input and output stoichiometric coefficients respectively representing the number of units of each item that enter in the j-th step of the process. In particular, if item x_i does not appear as an input in the j-th step, the coefficient a_i^j is equal to zero; similarly, if it does not appear in the output of the step, $b_i^j = 0$.

Finally, the flow rate constants, k_j, $j = 1, \ldots, r$, are parameters that characterize flow and storage capacity constraints of the underlined process.

A remark about the notation: we use the same symbol, x_i, to indicate both a label of a species i and its concentration (inventory level). We will call a vector $\mathbf{x}(t) = (x_1(t), ..., x_n(t))$ of all inventory levels of species in ME the state vector. We also define partition (not necessarily disjoint) of the index set $\{1, 2, \ldots, n\}$ into three subsets: S_i corresponds to the set of the input states, S_s corresponds to the set of the internal states, and S_o corresponds to the set of the output states.

We define the generalized output rate of process step j as follows

$$\theta_j(x_1, ..., x_n) = k_j \prod_{i=1}^{n} x_i^{a_i^j}. \tag{2}$$

The net rate of production (or consumption) of species x_i, $\dot{x}_i(t)$, at any time t is given by

$$\dot{x}_i(t) = \begin{cases} -\sum\limits_{j\in R_i^r} a_i^j \theta_j(x_1(t), ..., x_n(t)) + \sum\limits_{j\in R_i^p} b_i^j \theta_j(x_1(t), ..., x_n(t)) + J_1^i(t), & i \in S_i \\ -\sum\limits_{j\in R_i^r} a_i^j \theta_j(x_1(t), ..., x_n(t)) + \sum\limits_{j\in R_i^p} b_i^j \theta_j(x_1(t), ..., x_n(t)), & i \in S_s \cup S_o, \end{cases} \tag{3}$$

where, by definition, R_i^r is a set of all reactions in which species i is consumed, and R_i^p is a set of all reactions in which species i is produced.

At the chosen level of aggregation, the membrane element (ME) has the architecture shown in Figure 3. For simplicity, in the figure we use the identification,

$$f_i(\mathbf{x}(t)) = -\sum_{j\in R_i^r} a_i^j \theta_j(x_1(t), ..., x_n(t)) + \sum_{j\in R_i^p} b_i^j \theta_j(x_1(t), ..., x_n(t)).$$

Then using this identification, the state dynamics is described by the following system of ODEs

$$\begin{aligned} \dot{x}_i(t) &= f_i(\mathbf{x}(t)) + J_1^i(t), \; i \in S_i \\ \dot{x}_i(t) &= f_i(\mathbf{x}(t)), \; i \in S_s \cup S_o, \end{aligned} \tag{4}$$

Note that the functions $f_i(\cdot)$ are, in general, non-linear in the state vector $\mathbf{x}$, but the dynamics is linear in the control variables, J_1^i.

The production rules (1) defining the dynamics of each membrane element are extracted from a database of the enterprise. In manufacturing processes, the necessary information to build the production rules is contained in the bill of materials.

The explicit form of equations (4) for an example will be given in Section 5.

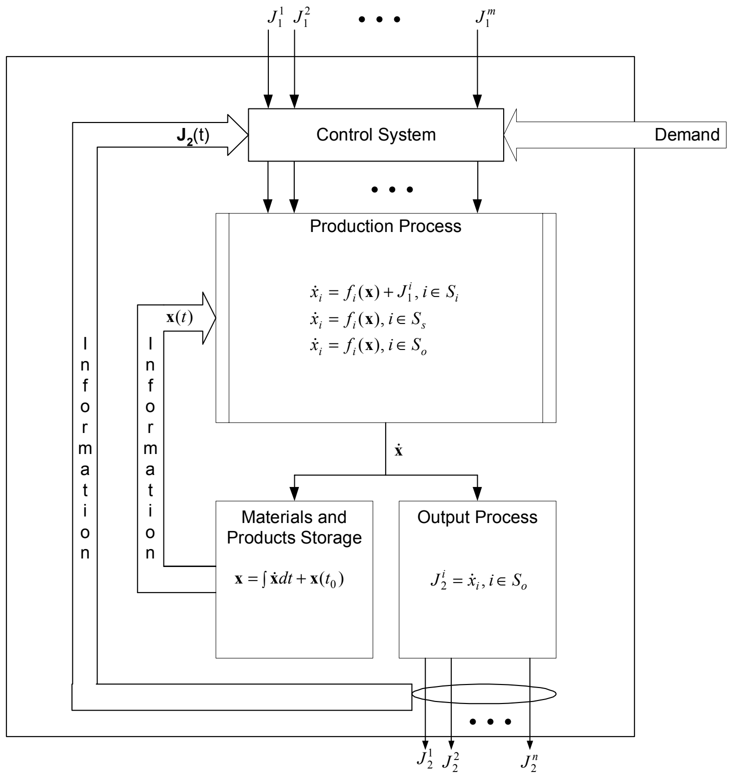

Fig. 3: Internal Dynamics of a Membrane Element

4. Control System

In a biological membrane, the embedded control system has four functions [15]: transport regulation of species, energy management, protection of the underlying cell, and transport of information (neuron membranes). The analogy that we exploit in our work is that the control system of a membrane element in our supply chain model fulfills these four functions for the enterprise system, i.e. the transportation of the goods and services, management of the flow of money, protection against oversupplies or productions beyond safety levels, and processing the flow of information.

In general, our supply chain system operates in a *continuous repair policy* mode. That is, if the supply chain model is used in the planning, scheduling, or execution environment, the underlying control system of each element will operate as a continuous repair planner, continuous repair scheduler, or continuous repair executor. In either of these three functionalities the assumption is that the supply chain dynamics fluctuates around a dynamically upgradable nominal regime. This regime is defined differently depending on an application. In a planning environment, for example, one wants to select the regime so that the input flow matches the demand with intermediate storages slightly above the safety level.

In either case, the control system operates as an incremental tracker around the defined nominal regime. We will derive now the optimal feedback control law for this system.

The dynamic equations (4) describing the production process of ME act as constraints for the optimization procedure for synthesizing the control law, to be described below.

An additional dynamic constraint, synchronization constraint, is added to the optimization formulation. The synchronization constraint implements the functionality of synchronization of an ME with other ME's connected to its inputs. This constraint takes the generic form

$$\dot{y}_i(t) = J_2^i(t) - J_1^i(t), \quad i \in S_i \tag{5}$$

These differential equations are appended to 4 and a penalty is added to the criterion defining optimality for the repair. In a paper in preparation, we introduce a learning and adaptation schema to the repair mechanism of the control system to allow the ME network to improve performance over time.

The global criterion for the ME network is approximated by defining an additional accumulation variable, $z(t)$, for each ME and a differential equation of the form:

$$\begin{aligned} &\dot{z}(t) = \Psi(\mathbf{x}(t)) - \Lambda \mathbf{J}_1(t) \quad \text{with} \\ &z(0) = 0. \end{aligned} \tag{6}$$

The right hand side of (6) is the criterion *rate* (profit maximization rate). Thus, the variable $z(t)$ represents the accumulated value of the criterion up to time t when the policy $\mathbf{J}_1(\tau), 0 \leq \tau \leq t$ is used by the ME. The function Ψ represents the "profit" associated with the inventory level. The term $\Lambda \mathbf{J}_1(t)$ represents the "cost" as a function of the number of items input to the ME. The constant vector Λ represents the "cost" per unit for each input item.

The criterion is approximated by adding a terminal penalty to the *control law design criterion* $\int_0^T L(\rho(t), \mathbf{v}(t))dt$ in equation (17). The mechanism is as follows. From (6), the criterion is given by

$$\max_{\mathbf{J}_1} \int_0^T (\Psi(\mathbf{x}(t)) - \Lambda \mathbf{J}_1(t))dt, \tag{7}$$

where T is the time horizon of the formulation.

Thus for the optimal flow $\mathbf{J}_1(t)$, the variable $z(t)$ represents the maximum accumulated profit up to time t, while $z(T)$ represents the total optimal profit. By heavily penalizing the terminal value of $z(t)$, $z(T)$, it can be shown [21] that under mild assumptions of the smoothness of the criterion, we can approximate the flow optimal trajectory $\mathbf{J}_1(t)$, $0 \le t \le T$, using the generic criterion in (17).

Now we proceed to outline the derivation of the generic control law of the control system in each ME.

Equations (4) can be written as follows

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{D}\mathbf{J}_1(t), \tag{8}$$

where $\mathbf{x} = (x_1, ..., x_n, \mathbf{y}, z)$, $\mathbf{f}(\mathbf{x})$ is a continuous differential function, $\mathbf{D}$ is an $n \times m$ constant matrix whose entries are $D_{ij} = \begin{cases} 1, \text{ if } i \in S_i \text{ and } j = i \\ 0 \text{ otherwise} \end{cases}$, and $\mathbf{J}_1 = \left(J_1^1, ..., J_1^m\right)$ is a vector of inputs (controls).

The dynamics in our model architecture is driven by the input flows. Thus the membrane element behavior is controlled by manipulating these flows. We need a control law that makes the current input flow a function of the history of the storages and the internal and output flows for the following reasons: 1- To satisfy demand at the node in the chain modeled by the ME, and 2- To satisfy the optimality criterion (7) and synchronization constraint (5).

We also want the control law to satisfy exactly the target demand *asymptotically*. This requires an *integral control law* [6]. In order to use the linear quadratic optimization techniques and satisfy the requirements above, we will synthesize the control law by defining a linear quadratic problem in which the decision variables are the rates of change of the input flows. In order to accomplish this, we need to express the dynamics of each membrane element as a second order differential equation.

The second order model is determined below. We take the derivative with respect to t on the both sides of (8) to obtain:

$$\ddot{\mathbf{x}}(t) = \frac{\partial \mathbf{f}(\mathbf{x})(t)}{\partial \mathbf{x}} \left[\mathbf{f}(\mathbf{x}(t)) + \mathbf{D}\mathbf{J}_1(t)\right] + \mathbf{D}\dot{\mathbf{J}}_1(t). \tag{9}$$

Equation (9) is a special case of a very general class of second order systems called *sprays* [9], [8]. Sprays encode conservation principles in a special class of connections. These connections establish locally how neighboring states can be reached from the current state. The knowledge about any supply chain is completely encoded in the connections of the models of each ME. We exploit this property in our upcoming monograph [8].

We define $\mathbf{P}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{J}_1, \dot{\mathbf{J}}_1) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} [\mathbf{f}(\mathbf{x}) + \mathbf{D}\mathbf{J}_1] + \mathbf{D}\dot{\mathbf{J}}_1$. Then (9) becomes

$$\ddot{\mathbf{x}}(t) = \mathbf{P}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{J}_1, \dot{\mathbf{J}}_1). \tag{10}$$

After performing the state augmentation procedure, we get the following system that describes the linearized dynamics of (10) around the nominal regime of the ME.

$$\dot{\eta}(t) = \mathbf{A}\eta(t) + \mathbf{B}\mathbf{v}(t), \ \eta(0) = \mathbf{0}. \tag{11}$$

where the $2n + m$ vector function η is of the form:

$$\begin{bmatrix} \eta_1(t) \\ \eta_2(t) \\ \eta_3(t) \end{bmatrix}. \tag{12}$$

In (12) $\eta_1(t)$ is the variation of $\mathbf{x}(t)$, $\eta_2(t)$ is the variation of $\dot{\mathbf{x}}(t)$, and $\eta_3(t)$ is the variation of $\mathbf{J}_1(t)$. In (11), $\mathbf{v}(t)$ is the incremental "acceleration" $(units/time^2)$ of the input flow vector. For the purposes of the linear quadratic design, $\mathbf{v}(t)$ is our decision variable. However, as we will show below, the control law expresses the desired *input flow* as a function of storage, and internal and output flow quantities.

We define the variation $\xi(t)$ of the demand rate $\mathbf{d}(t)$, to the ME, from the nominal demand $\bar{\mathbf{d}}$ as follows:

$$\mathbf{d}(t) = \bar{\mathbf{d}} + \xi(t). \tag{13}$$

The demand rate $\mathbf{d}(t)$ for each ME is determined by *back-propagating* the external demand to the supply chain network. This back-propagation is a forecasting exercise that requires a *smoothing filter* [20] for each ME. This filter network is designed with an optimization approach similar to the control synthesis we are describing in this section. We will discuss the network of synchronized smoothing filters for demand back-propagation in a future paper.

The demand rate variation, $\xi(t)$, $0 \leq t \leq T$, is the target trajectory for the output flow increments trajectory, $\eta_i, \ \ i \in S_o$, for each ME.

We define the Lagrangian for the *demand-tracking problem* as follows

$$L(\rho(t), \mathbf{v}(t)) = \rho^T(t)\mathbf{Q}\rho(t) + \mathbf{v}^T(t)\mathbf{R}\mathbf{v}(t), \tag{14}$$

with

$$\begin{bmatrix} \rho_1(t) \\ \rho_2(t) \\ \rho_3(t) \end{bmatrix} = \begin{bmatrix} \eta_1(t) \\ \eta_2(t) - \mathbf{h}(t) \\ \eta_3(t) \end{bmatrix}, \tag{15}$$

where $\mathbf{h}(t)$ is defined as follows:

$$h_i(t) = \begin{cases} \xi_i(t) & i \in S_o \\ 0 & i \in S_i \cup S_s \end{cases}$$

Note $\mathbf{h}(t)$ is the variation of the current demand $\mathbf{d}(t)$ for the ME, back-propagated in time and over the membrane network from the output (finished) goods and services.

In (14), the penalty matrix $\mathbf{Q}$ is determined so that storage, flow, synchronization constraints and criterion constraints are satisfied. Many heuristic rules have been developed over the years to design these penalty matrices. However, in the enterprise applications these matrices are determined so that the Lagrangian (14) is compatible with the mass and flow conservation principles defining (8):

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_3 \end{bmatrix}. \tag{16}$$

In (14), $\mathbf{R}$ is a positive-definite symmetric matrix penalizing the rate of input flow. It is chosen so that input flow constraints are satisfied. The positive-definiteness requirement is the necessary condition for the linear quadratic optimization problem formulated below.

The solution of the following optimization problem generates the control law implemented by the ME's control system.

$$\min_v \int_0^T L(\rho(t), \mathbf{v}(t))dt + \rho^T(T)\mathbf{S}_T\rho(t) \tag{17}$$

subject to

$$\dot{\rho}(t) = \mathbf{A}\rho(t) + \mathbf{B}\mathbf{v}(t) + \phi(t), \;\; \rho(0) = \begin{bmatrix} \mathbf{0} \\ -\mathbf{h}(0) \\ \mathbf{0} \end{bmatrix}, \tag{18}$$

where $L(\rho(t), \mathbf{v}(t))$ is defined in (14) and

$$\phi(t) = \mathbf{A} \begin{bmatrix} 0 \\ \mathbf{h}(t) \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \dot{\xi}(t) \\ 0 \end{bmatrix} \tag{19}$$

The feedback control law that solves problem (17)-(19) is given by [7], [10]:

$$\mathbf{v}(t) = \mathbf{K}(t)\rho(t) - \mathbf{R}^{-1}\mathbf{B}^T\mathbf{g}(t), \tag{20}$$

where

$$\mathbf{K}(t) = -\mathbf{R}^{-1}\mathbf{B}^T\Sigma(t). \tag{21}$$

In (21), matrix function Σ is computed by solving the following matrix differential equation (Riccati):

$$\dot{\Sigma}(t) = -\Sigma(t)\mathbf{A} - \mathbf{A}^T\Sigma(t) + \Sigma(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\Sigma(t) + \mathbf{Q}, \quad \Sigma(T) = \mathbf{S}_T. \tag{22}$$

For most enterprise applications, due to the large state penalty ($\mathbf{Q}$), the solution of (22) converges to a steady state solution $\tilde{\Sigma}$ in an interval $\tau \ll [0, T]$. This is illustrated in Figure 4. So in 21 we can replace $\Sigma(t)$ with $\tilde{\Sigma}$. We can show that the error of this approximation is lower than the absolute value of the fastest eigenvalue of $\mathbf{A}$.The gain matrix function $\mathbf{K}(t)$ is replaced by the approximation $\tilde{\mathbf{K}}$:

$$\tilde{\mathbf{K}} = -\mathbf{R}^{-1}\mathbf{B}^T\tilde{\Sigma}.$$

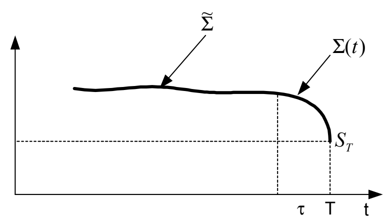

Fig. 4: Conceptual Illustration of Convergence of $\Sigma(t) \to \tilde{\Sigma}$

In (20), vector $\mathbf{g}$ is computed by solving the following vector linear differential equation

$$\dot{\mathbf{g}}(t) = \left(\Sigma(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T - \mathbf{A}^T\right)\mathbf{g}(t) - \Sigma(t)\phi(t), \quad \mathbf{g}(T) = \mathbf{0}. \tag{23}$$

which by the observation above, can be approximated by the constant coefficient equation:

$$\dot{\mathbf{g}}(t) = \left(\tilde{\Sigma}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T - \mathbf{A}^T\right)\mathbf{g}(t) - \tilde{\Sigma}\phi(t), \quad \mathbf{g}(T) = \mathbf{0}. \tag{24}$$

We wish to express the feedback control law in the original variables. We substitute (20) into (11) to get

$$\begin{bmatrix} \dot{\eta}_1(t) \\ \dot{\eta}_2(t) \\ \dot{\eta}_3(t) \end{bmatrix} = \left(\mathbf{A} + \mathbf{B}\tilde{\mathbf{K}}\right) \begin{bmatrix} \eta_1(t) \\ \eta_2(t) \\ \eta_3(t) \end{bmatrix} - \psi(t), \tag{25}$$

where

$$\psi(t) = \mathbf{B}\tilde{\mathbf{K}} \begin{bmatrix} \mathbf{0} \\ \mathbf{h}(t) \\ \mathbf{0} \end{bmatrix} + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{g}(t).$$

We solve the differential equation in (25) for $\eta_3(t)$ to obtain

$$\eta_3(t) = \exp\left(\left(\mathbf{A} + \mathbf{B}\tilde{\mathbf{K}}\right)_{33}(t - t_0)\right)\eta_3(t_0) + \int_{t_0}^{t} \exp\left(\left(\mathbf{A} + \mathbf{B}\tilde{\mathbf{K}}\right)_{33}(t - \tau)\right)[\Phi(\tau) - \psi(\tau)]\, d\tau, \tag{26}$$

where

$$\Phi(\tau) = \left(\mathbf{A} + \mathbf{B}\tilde{\mathbf{K}}\right)_{31}\eta_1(\tau) + \left(\mathbf{A} + \mathbf{B}\tilde{\mathbf{K}}\right)_{32}\eta_2(\tau). \tag{27}$$

In (26) the first term on the right-hand side corresponds to the transient of the input flow variation. We can neglect this term because the close-loop system is stable by design and the initial variation is bounded.

Then the flow control law is given by

$$\mathbf{J_1}(t) = \bar{\mathbf{J}}_1 + \int_{t_0}^{t} \exp\left(\left(\mathbf{A} + \mathbf{B}\tilde{\mathbf{K}}\right)_{33}(t - \tau)\right)[\Phi(\tau) - \psi(\tau)]\, d\tau. \tag{28}$$

The interpretation of this result is as follows. The second term on the right-hand side is the input flow variation from the nominal regime $\bar{\mathbf{J}}_1$. The integrand at each instant of time is the discounted difference of the weighted sum $\Phi(\tau)$ of the optimal level of inventory variation at time $\tau \le t$, $\left(\mathbf{A} + \mathbf{B}\tilde{\mathbf{K}}\right)_{31}\eta_1(\tau)$, and the optimal production variation at time $\tau \le t$, $\left(\mathbf{A} + \mathbf{B}\tilde{\mathbf{K}}\right)_{32}\eta_2(\tau)$, and the increment in demand variation, $\psi(\tau)$. If this difference is zero, then the optimal regime is equal to the nominal regime. On the other hand, if this difference is positive at some time t, then the optimal regime is less than nominal. Alternatively, if the increment in demand variation, $\psi(\tau)$, is larger than both the variations of the inventory level and the production rate, the optimal regime is greater than the nominal regime.

Observe also that the flow control law in (28) takes into consideration past values of the difference $[\Phi(\tau) - \psi(\tau)]$, discounting them by the factor $\left(\mathbf{A} + \mathbf{B}\tilde{\mathbf{K}}\right)_{33}(t-\tau)$. Since $\mathbf{A} + \mathbf{B}\tilde{\mathbf{K}}$ is a stable matrix, the real part of each of its eigenvalues is negative, the effect of past differences decreases as time increases. That is, for each time t the current input flow dependency on the difference between incremental flow capacity and incremental demand is monotonic in time.

Furthermore, a simple algebraic manipulation shows that first order perturbations in the demand variation induce first order perturbations in the input flow. This is a consequence of the stability of $\mathbf{A}+\mathbf{B}\tilde{\mathbf{K}}$ [22]. Furthermore the same result follows if the first order perturbation is in the parameters. This implies that the feedback law induces robustness in the behavior of each membrane element.

Finally, the control law (28) can be shown to have good characteristics for uncertainty management of the demand. We will discuss this in detail in a paper in preparation.

5. Illustrative Example

5.1. Problem Description

For the purposes of illustration of the procedure described in Sections 3. and 4., we use an HP DeskJet printer supply chain system adapted from a case study in [12].

The supply chain serves a high volume, high production rate facility with total enterprise cycle of about one week and a time horizon of about 7 weeks. There are three distribution centers: US DC, European DC, and Far East DC.

HP operates the three DC's in a make-to-stock mode, i.e. maintaining the target inventories at the levels of the forecasted sales plus some safety stock. As a consequence of the long transportation time, the European and Asian DC's had to keep high levels of safety stocks in order to response to fluctuations in the demand for the different versions of the printer.

5.1.1. Supply Chain

The Vancouver division is connected with its suppliers, most of which are other HP divisions, and the distribution centers, located in the US, Europe, and Asia. Oversees sales require localization, i.e. equipment of printers with the appropriate power supplies and manuals. Also they lead to a significant transportation times between the factory and DC's in Europe and Asia.

The supply chain block diagram is presented in Figure 5. As discussed in Section 2., we assign a membrane element to each node of the supply chain graph.

The factory has two manufacturing units

1. Printed circuit assembly and test
2. Final assembly and test.

All necessary raw materials and parts are supplied by the suppliers. The lead times for all suppliers are 2 days. The shipment times for the DC's are shown in

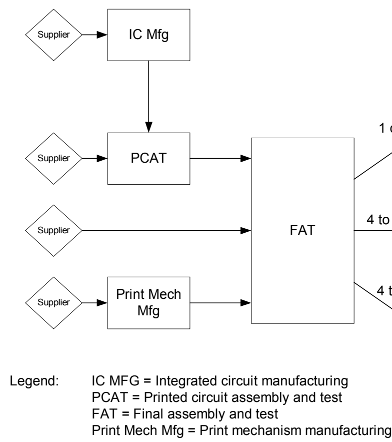

Fig. 5: Supply Chain

the figure.

5.1.2. Bill of materials

Figure 6 shows the bill of materials.

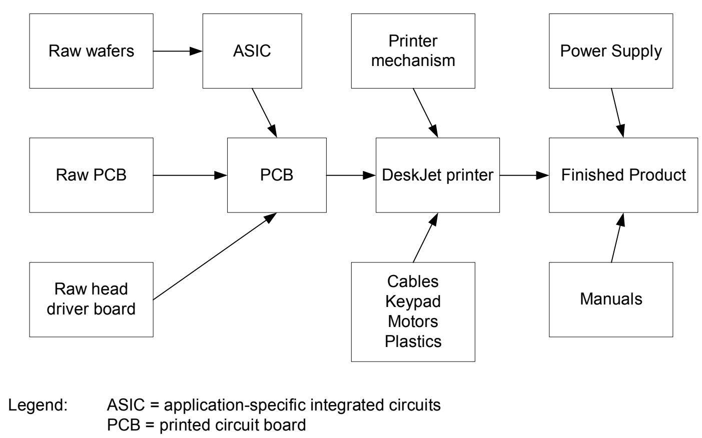

Fig. 6: Bill of Materials

5.1.3. Variable assignment

The assignment of variables is shown in Figure 7. This is carried out in general as a function of the desired level of aggregation.

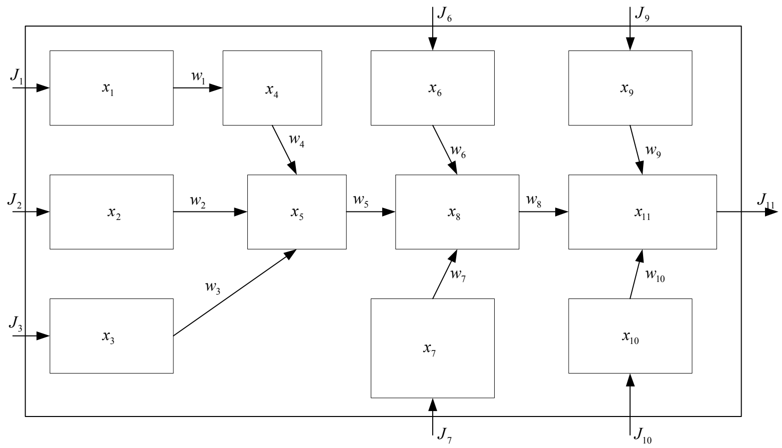

Fig. 7: Variable Assignment

5.2. Dynamic Model

From Figure 7 and under the assumption that there are storages available at each node of the supply chain, the ME controls the node availability by regulating the input flow.

We can express the bill of materials graph (Figure 6) and the production process in the following symbolic form (production rules). See (1) in Section 3.

$$a_1^1 x_1 \xrightarrow{k_1} b_4^1 x_4 \tag{29}$$

$$a_2^2 x_2 \circledast a_3^2 x_3 \circledast a_4^2 x_4 \xrightarrow{k_2} b_5^2 x_5 \tag{30}$$

$$a_5^3 x_5 \circledast a_6^3 x_6 \circledast a_7^3 x_7 \xrightarrow{k_3} b_8^3 x_8 \tag{31}$$

$$a_8^4 x_8 \circledast a_9^4 x_9 \circledast a_{10}^4 x_{10} \xrightarrow{k_4} b_{11}^4 x_{11}, \tag{32}$$

where a_i^j's and b_i^j's are coefficients defined by the bill of materials, the symbol $\circledast$ represents a "merging" operator and '$\rightarrow$' represents a process with rate functions k_i's defined by the characteristics of the process.

Because of the simplicity of the example, we implement one ME for controlling the entire supply chain.

We note that once the bill of materials graph is given, we can construct the production rules almost by inspection. The bill of material graph should be specified with the desired level of abstraction in mind. In our example the desired level of abstraction is at the sub-assembly level.

Specifically, relation (29) corresponds to manufacturing of ASIC from the raw wafers (see Figure 6). Likewise, (30) relates the printed circuit board assembly to its inputs: ASIC, raw PCB and raw head driver board. Production rule (31) represents the relation between the printer assembly and the materials needed for it: printer mechanism, PCB, cables, keypad, motors, and plastics. Finally, (32) relates the assembly of a finished product and the modules necessary for it: a printer, a power supply, and a manual.

From the model we see that $a_i^j = b_i^j = 1, \forall i, j$. Then from (29)-(32), the dynamics of each of the components is given by

$$\begin{aligned} \dot{x}_1(t) &= J_1(t) - k_1 x_1(t) \\ \dot{x}_2(t) &= J_2(t) - k_2 x_2(t) x_3(t) x_4(t) \\ \dot{x}_3(t) &= J_3(t) - k_2 x_2(t) x_3(t) x_4(t) \\ \dot{x}_4(t) &= k_1 x_1(t) - k_2 x_2(t) x_3(t) x_4(t) \\ \dot{x}_5(t) &= k_2 x_2(t) x_3(t) x_4(t) - k_3 x_6(t) x_7(t) x_5(t) \\ \dot{x}_6(t) &= J_6(t) - k_3 x_6(t) x_7(t) x_5(t) \\ \dot{x}_7(t) &= J_7(t) - k_3 x_6(t) x_7(t) x_5(t) \\ \dot{x}_8(t) &= k_3 x_6(t) x_7(t) x_5(t) - k_4 x_8(t) x_9(t) x_{10}(t) \\ \dot{x}_9(t) &= J_9(t) - k_4 x_8(t) x_9(t) x_{10}(t) \\ \dot{x}_{10}(t) &= J_{10}(t) - k_4 x_8(t) x_9(t) x_{10}(t) \\ \dot{x}_{11}(t) &= k_4 x_8(t) x_9(t) x_{10}(t) = J_{11} \end{aligned} \tag{33}$$

where $x_i(t)$ $i = 1, ..., 11$ measures the inventory level at time t . Although all the components are discrete, at this level of abstraction, we can treat the corresponding inventory variables as continuous [23]. In more detailed applications (lower level of abstraction) we would, in addition to the differential equations, provide a quantization schema for extracting the integer sample trajectories out of the continuous trajectories.

Differentiating (33) with respect to t and using (33) again, we obtain the second order equation representing the dynamics of this example, see (10):

$$\ddot{\mathbf{x}}(t) = \mathbf{P}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{J}_1, \dot{\mathbf{J}}_1), \tag{34}$$

where $\mathbf{P}$ is a real 11-dimensional vector-valued function.

After linearizing equation (34), around the nominal storage, $x_i^0 = 140747.4$ units and nominal regime $\dot{x}_i^0 = 140747.4$ units/week, we obtain as in (11):

$$\begin{bmatrix} \dot{\eta}_1(t) \\ \dot{\eta}_2(t) \\ \dot{\eta}_3(t) \end{bmatrix} = \mathbf{A} \begin{bmatrix} \eta_1(t) \\ \eta_2(t) \\ \eta_3(t) \end{bmatrix} + \mathbf{B}\mathbf{v}(t), \tag{35}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{11\times 11} & \mathbf{I}_{11\times 11} & \mathbf{0}_{11\times 7} \\ \frac{\partial \mathbf{P}}{\partial \mathbf{x}}\Big|_{(\mathbf{x}^0,\dot{\mathbf{x}}^0)} & \frac{\partial \mathbf{P}}{\partial \dot{\mathbf{x}}}\Big|_{(\mathbf{x}^0,\dot{\mathbf{x}}^0)} & \frac{\partial \mathbf{P}}{\partial \mathbf{J}_1}\Big|_{(\mathbf{x}^0,\dot{\mathbf{x}}^0)} \\ \mathbf{0}_{7\times 11} & \mathbf{0}_{7\times 11} & \mathbf{0}_{7\times 7} \end{bmatrix} \tag{36}$$

and

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{11\times 7} \\ \frac{\partial \mathbf{P}}{\partial \dot{\mathbf{J}}_1}\Big|_{(\mathbf{x}^0,\dot{\mathbf{x}}^0)} \\ \mathbf{I}_{7\times 7} \end{bmatrix}. \tag{37}$$

We note that $\dot{\eta}_3(t)$ in (35), the "acceleration" of the input flow, equals the decision variable $\mathbf{v}(t)$.

The values for the nominal point, $(x_i^0, \dot{x}_i^0)$, were obtained from the demand data given in [12] by computing the values for the economic order quantity (EOQ), the reoder point s, and order-up-to level S. Then we set x_i^0 to the average inventory level $s + \frac{1}{2}EOQ$ and $\dot{x}_i^0$ to the expected value of demand rate.

Follow the method described in (6)-(8), we introduce an additional state variable, η_{30}, such that

$$\dot{\eta}_{30}(t) = c\sum_{i=1}^{29} \eta_i(t), \tag{38}$$

where the right-hand side approximates the total inventory holding cost.

Then the quadratic optimization criterion of (17) becomes

$$\int_0^T \left(\rho^T(t)\mathbf{Q}\rho(t) + \mathbf{v}^T(t)\mathbf{R}\mathbf{v}(t)\right) dt + \rho^T(T)\mathbf{S}_T\rho(T), \tag{39}$$

where we set $\mathbf{Q} = \mathbf{I}_{30\times 30}, \mathbf{R} = \mathbf{I}_{7\times 7}$, and $\mathbf{S}_T = \begin{bmatrix} \mathbf{0}_{29\times 29} & \mathbf{0}_{29\times 1} \\ \mathbf{0}_{1\times 29} & 50 \end{bmatrix}$.

6. Simulation Results

The simulation of the proposed model (29)-(39) was carried out using Maple software package. The parameters of the model were computed using the method described in [12]. In particular, the k_i's are chosen to satisfy flow and storage capacity constraints.

The nominal regime was chosen around the capacity of the chain. In addition to this case, we performed several runs with different constraints to ascertain the theoretical properties of the model discussed in Section 4. (robustness, stability, and asymptotic conversions). We observed that these properites hold for the system on hand in a variety of operation regimes. We also determined that in order to maintain the output stability, the feedback control system is necessary. The simulations also demonstrated the flexibility of the modeling paradigm: one can run this system as a system of different networks of ME's by partioning the production rules (29)-(32) appropriately and adding relevant synchronization constraints (5) between nearest neighbor ME's .

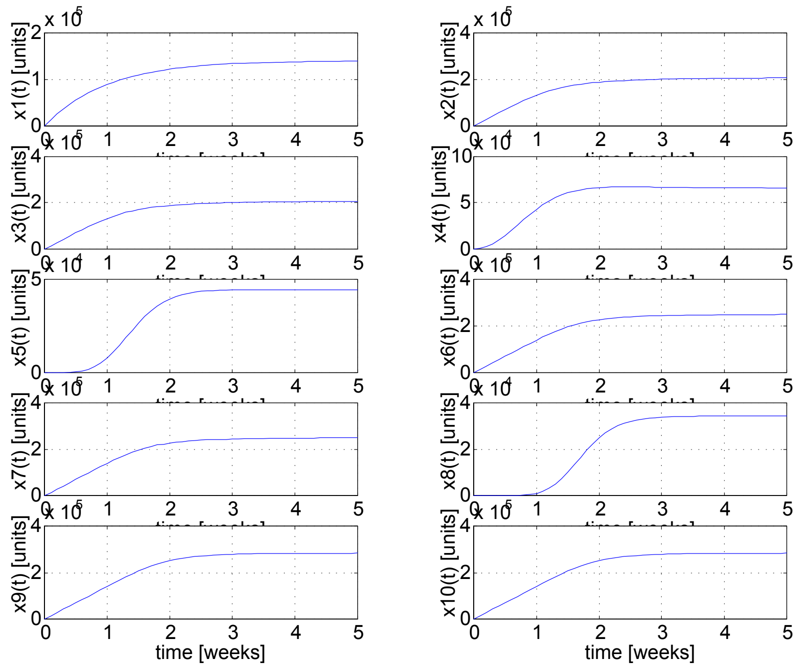

Fig. 8: Simulation Results

The plots of the simulated state trajectories are presented in Figure 8. They exhibit the strong convergence and performance properties. The system exhibits a tendency to operate at balance regime: that is after an almost linear transient the input flow is almost constant and balances the variability of the demand.

Comparisons of these simulations with operating data show the behavior of the model is reasonable and feasible.

7. Conclusions

We have developed an automated methodology for generating dynamic distributed models of enterprise systems. In particular, our approach is ideally suited for modeling supply chain systems. The model uses the computational theory of hybrid systems [5] to transform enterprise hybrid models (discrete and continuous components) into continuous differential models. The complete structural and computational characteristics of continuous systems provides a rich environment for the representation of enterprise systems and their control (planning, scheduling, and execution). In particular, distributed continuous models rent themselves for generic effective synchronization. Furthermore, we have verified that these continuous differential paradigms scale up computationally in the linear fashion.

The mechanism of going from the production rules (1) to the differential model characterizing the dynamics (3) is a very good candidate for symbolic implementation. We have tested a prototype in Prolog.

Furthermore, by building -in a generic control system in each ME, we have produced a framework for implementing supply chain functionality (planning, scheduling, and execution) largely as a highly automated *modeling exercise.*

The ability to encode a generic optimization system and a mechanism (6) for approximating optimal behavior for a wide variety of criteria, shortens substantially the implementation time for distributed supply chain applications. This is because the *implementation is largely generic* and the specialization is achieved by encoding a criterion and constraints into this generic framework.

The generic control law presented in Section 4. is based on the linear quadratic theory. The decision variables are input flow rates. The implemented control variable is input flow. This approach leads to an integral control law that has several desirable properties, see Section 4. Perhaps, its most important characteristic is the fact that each input flow is synchronized with its source and it balances demand variation.

We have three papers in progress. The first paper discusses a schema for uncertainty management in supply chain systems in which we will demonstrate how to propagate demand backwards as discussed in Section 2. The second paper describes a synchronized smoothing filter for demand forecasting. The third paper formulates an adaptation schema in which the parameters of the model are adjusted as a function of information from the supply chain systems.

Bibliography

1. Ben Cummings, Wolf Kohn, Anil Nerode, Karl Shell, and Jeffrey B. Remmel, *MAHCA Cost/Benefit Modeling: A Multiple-Agent Hybrid Control Architecture for Cost/Benefit Analysis, A Proposed Approach to Optimal Performance while Downsizing*, HyBrithms Technical report.

2. Wolf Kohn, Anil Nerode, John James, Jeffrey Remmel, and Benjamin Cummings, *A New Approach to Generating Finite-State Control Programs for Hybrid Systems*, Proceedings of the 13-th Triennial World Congress of the International Federation of Automatic Control (IFAC'96), (1996), 461-466.

3. Wolf Kohn and Jeffrey B. Remmel, *Digital to Hybrid Program Transformations*, Proceedings of 1996 IEEE International Symposium on Intelligent Control, (1996), 342-347.

4. Wolf Kohn, Anil Nerode, and Jeffrey B. Remmel, *Continualization: A Hybrid Systems Control Technique for Computing*, Proceedings of CESA'96 IMACS Multiconference, (1996), 517-521.

5. Panos Antsaklis et al. (eds.), *Hybrid Systems V, Lecture Notes in Computer Science,* number 1567, Springer, Berlin, 1999.

6. Wolf Kohn, Vladimir Brayman, and Pawel Cholewinski, *Hynomics Optimal Plan Repair System*, Hynomics Technical report, 2000.

7. Wolf Kohn, Vladimir Brayman, and Pawel Cholewinski, *Finsler Tracker*, Hynomics Technical report, 2000.

8. Wolf Kohn, Jeffrey Remmel, and Vladimir Brayman, *Distributed Agent Control of Enterprise Systems*, to be published, July 2001.

9. Serge Lang, *Differential and Riemannian Manifolds*, Springer-Verlag, New York, 1995.

10. Arthur E. Bryson, Jr. and Yu-Chi Ho, *Applied Optimal Control*, Blaisdell Publishing Company, Waltham, Massachusetts, 1969.

11. S. R. de Groot and P. Mazur, *Non-Equilibrium Thermodynamics*, Dover, New York, 1984.

12. David Simchi-Levi, Philip Kaminsky, and Edith Simchi-Levi, *Designing and Managing the Supply Chain*, McGraw-Hill, 2000.

13. Sridhar Tayur, Ram Ganeshan, and Michael Magazine, *Quantitative Models for Supply Chain Management*, Kluwer, Boston, 1999.

14. Steve Rockhold, Hau L. Lee , and Robert Hall, *Strategic Alignment of a Global Supply Chain for Business Success* , in Global Supply Chain and Technology Management, Hau L. Lee and Shu Ming Ng eds.,POMS Series in Technology and Operations Management vol. 1, POMS, 1998.

15. Wolf Kohn, *A Model for the Neuron Cell Membrane*, Master of Science Thesis, MIT, 1974.

16. J. Dannielli and H. Davison, *The Chemical Structure of Biological Membranes,* J. Cell Comp. Physiol., 5, 495-506, London, 1935.

17. A. Katchalsky and R. Spangler, *Dynamics of Membrane Process*, Biophys. J., vol. 4, Supple 2, 9, 1969.

18. R. Rosen ed., *Foundations of Mathematical Biology*, vol.1, Subcellular Systems, Academic Press, New York, London, 1972.

19. John W. Moore and Ralph G. Pearson, *Kinetics and Mechanism*, 3rd edition, John Wiley & Sons, New York, 1981.

20. J. S. Meditch, *Stochastic Optimal Linear Estimation and Control*, McGraw-Hill, New York, 1969.

21. Elijah Polak, *Optimization: Algorithms and Consistent Approximations*, Springer, New York, 1997.

22. Jose B. Cruz Jr., *Feedback Systems,* McGraw-Hill, New York, 1972

23. Oded Maimon, Eugene Khmelnitsky, and Konstantin Kogan, *Optimal Flow Control in Manufacturing Systems*, Kluwer, Dordrecht, 1998.