Multi-Echelon Inventory Planning System I: A Multiple Agent Hybrid Control Implementation

Wolf Kohn[1], Jeffrey B. Remmel[2]

Abstract

This is the first of two papers in which we describe a hybrid systems approach to supply chain management problems. We provide an outline of a design procedure of a Multiple Agent Hybrid Control Architecture (MAHCA) network for the deployment of a reactive, agent based planner for a single product multi-echelon inventory system. The objective of the system is to generate a near optimal distributed stocking plan policy. This policy approximates a centralized stocking policy which minimizes the expected value of a suitably defined cost functional. In this paper, we shall describe the basic model for any agent that controls a single node in the network. In the companion paper [22], we shall describe a continualization procedure which will reduce the agent optimization problem presented in this paper to a more standard non-linear optimal control problem that has the appropriate form to be solved by a MAHCA system.

1. Introduction

This is the first of two papers in which we describe a hybrid systems approach to supply chain management problems. In these two papers, we will provide an outline of a design procedure of a Multiple Agent Hybrid Control Architecture (MAHCA) network for the deployment of a reactive, agent based planner for a single product, multi-echelon inventory system. The objective of the system is to generate a near optimal distributed stocking plan policy. We will generate a policy that approximates a centralized stocking policy which minimizes the expected value of a suitably defined cost functional. In this paper, we shall describe the basic model of an agent controlling a single node in the distribution network plus a general criterion for the whole network based on a rule based synchronization procedure. In paper 2, we shall describe a continualization procedure which will reduce the agent optimization problem described in the paper to a more standard non-linear optimal control problem that has the appropriate form to be solved by a MAHCA system.

We shall model a distribution network for a single product p as a directed graph $G = (V, A)$. Each vertex j in the set V represents a distribution center (DC). The existence of an edge (i, j) implies that the DC represented by i acts a source of supply for the DC represented by j. When a DC can receive supply from multiple sources, we assume that there is a prespecified order in which the sources will be polled for supply. The transportation time between any two DC's i and j with $(i, j) \in A$ is one time period. We allow one infinite supply source that directly supplies several nodes in the network. We assume that there is a transportation lead-time of $l_j \geq 1$ time periods from the infinite source external supplier to each node j directly connected to the infinite source. We assume that the demand for p at vertex v in time period t is a stochastic process whose cumulative density function (c.d.f.) is given by the function $F_i^t(.)$. For any vertex i, we assume that the demand for p at vertex i is greater than zero in every time period and that the demands for any two nodes are independent. We assume that the cost of holding inventory at DC i is h_i dollars per unit for each time period and that the penalty for not satisfying demand is s_i dollars per unit for each time period.

2. The Multiple Agent Hybrid Control Architecture

The general framework in which MAHCA operates is shown in Figure 1. The framework is composed of three entities: a distributed process under control or planning, a collection of control/planning agents and a communication network for agent synchronization. In the system of this paper, the process is the multi-echelon system, an agent is the device implementing the planning policy for each DC i, and the agent network is just the network dictated by the network of distribution centers.

[1] Hynomics Corporation, 10632 NE 37th Circle, Bldg 23,Kirkland, WA 980333-7921: e-mail: wk@hybrithms.com

[2] Department of Mathematics and Computer Science, University of California at San Diego, San Diego, CA 92093: e-mail: jremmel@ucsd.edu

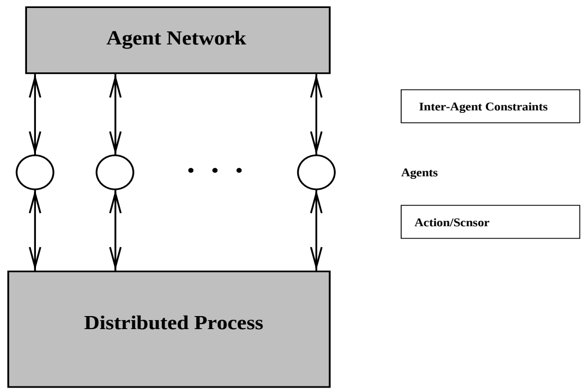

Figure 1: MACHA Framework

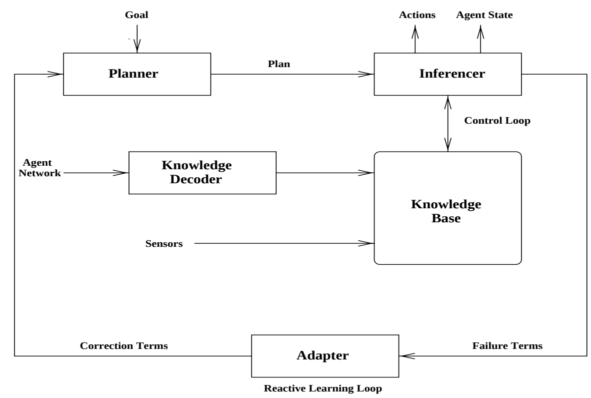

Figure 2: Agent Architecture

Each agent in the framework interacts with the process via its own sensors and actuators. An agent generates control actions that it computes in real time. The agent's control law generates the control actions. An agent's control law is constructed in real time and reflects the desired behavior of the process in accordance with stored requirements and as a function of observed sensory and inter-agent data flowing through the communication network. The data flowing through the communications network consists of inter-agent constraints used to maintain synchronization and to re-acquire it if events in the process cause its loss.

Functional Elements of MAHCA MAHCA is a software system for the autonomous synchronization and control of distributed real-time processes. From an operational point of view, the model shown in Figure 1 can represent both the distributed process under control and the MAHCA system carrying out the control.

Each of the circles represents an agent. An agent is a logic device that carries out prespecified synchronization and/or control functions. The action of each agent is a function of three information items:
(a) **Sensory Data**: On-line status data flowing from the process to the agent.
(b) **Active Knowledge**: Selected information data encoded in the agent's Knowledge Base.
(c) **Inter-agent constraints**: On-line status information from other agents via the prespecified logic network.
An agent carries out its control and synchronization functions by issuing command actions to the process and constraint data to the other agents.

The framework proposed in Figure 1 is very general. It is adequate for the representation of many dynamic distributed processes as well as their control and synchronization activities. For example, another typical supply chain management problem is to formulate an optimal schedule for a discrete multi-component manufacturing process. The process is carried out by an assembly line composed of assembly stations and product transportation subsystems. Each assembly station performs a partial assembly task on the incoming items which are then directed by an appropriate transportation subsystem to the station responsible for carrying the next stage in the assembly. In this scenario each assembly station and transportation subsystem carries out its tasks under the command or supervision of an assigned agent. The agent knows about the dynamics, constraints and operating rules of its station from encoded knowledge in its knowledge base. It knows about the current status of the station from the sensory information. It acquires and receives synchronization information from the other agents in the form of imposed constraints on the actions it can select.

An agent's functionality is implemented through an architecture called the Multiple Agent Hybrid Control Architecture (MAHCA). The architecture of a MAHCA agent is shown in Figure 2. The agent architecture operates via two interacting asynchronous loops: the *control loop* and the *reactive learning loop*. The control loop generates control actions and the agent's state as a function of its current knowledge to satisfy an internally generated plan. The reactive learning loop modifies the agent's plan as a function of observed agent behavior. In our current prototype, these two loops are implemented via five interacting modules: a *Planner*, an *Inferencer*, a *Knowledge Base*, an *Adapter*, and a *Knowledge Decoder*. The functionality of each module in the architecture is the following:

Planner: Constructs and repairs the agent's optimization criterion.
Inferencer: Determines whether there is a nonempty solution set for the agent's optimization

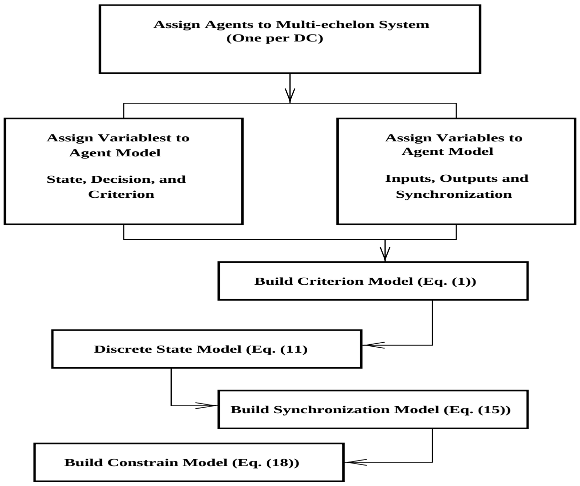

Figure 3: Model Building

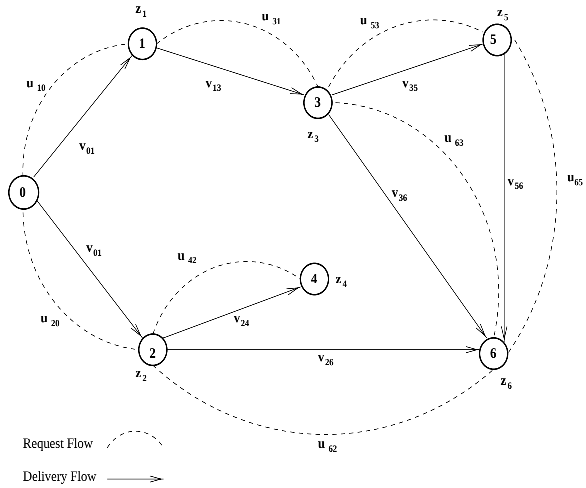

Figure 4: Supply Chain of Multi-Echelon System

problem. If there is such a solution set, the Planner infers the appropriate control actions, new state information and inter-agent information.
Adapter: Repairs failure terms and computes correction terms.
Knowledge Base: Stores and updates the agent's knowledge.
Knowledge Decoder: Receives and translates data from the other agents.

3. Constructing a MAHCA Model for the Distribution Network

In this section, we describe the formal construction of a dynamic model for each distribution center and a model for their interaction. This activity involves the formulation of flow, rules for material uncertainty and optimality conservation, and rules characterizing flow and material constraints. These rules are used to construct an iterative model representing the dynamics of the relevant supply chain or network. Figure 3, gives a flow chart that describes the model building procedure of this paper and its companion paper [22]. The main purpose of this paper is formulate our model as a collection of sub-models that correspond to the dynamics and criteria at each distribution center. This formulation is an input to our continualization process which will be described in the companion paper [22]. Our continualization procedure will generate the domain dynamics model and synchronization constraints used by the decision agents of a MAHCA implementation.

The directed graph of a simple one-product multi-echelon system is shown in the Figure 4. We shall distinguish three types of distribution centers in our example, namely, the infinite source DC ($i = 0$), the nodes directly connected to the infinite source called input DC's ($i = 1, 2$) and the regular DC's ($i = 3, 4, 5, 6$).

Next we list the model variables for a generic distribution center termed the i-th distribution center where i is the label of the node representing the center in the graph of the system.

(1) **Inventory (state variable)**: $x_i(t)$ is the number of units of the product carried over from the previous period at node DC i at the beginning of period t. We assume $x_i(t) \geq 0\ \forall i, t$.
(2) **Ordered amount(decision variable)** $u_{ij}(t)$ is the number of units of the product ordered by DC i from DC $j \in U_i$ at the end of period t where U_i is the index set of the upstream nodes directly connected to node DC i. We assume $u_{ij}(t) \geq 0\ \forall i, j, t$.
(3) **Sent amount (decision variable)** $v_{ij}(t)$ is the number of units of the product sent from DC i to DC $j \in V_i$ at the end of period t where V_i is the index set of the downstream nodes directly connected to node DC i. We assume $0 \leq v_{ij}(t) \leq u_{ji}(t)$, $\forall i, t \forall j \in V_i$.
(4) **External demand (auxiliary variable)** $y_i(t)$ is the number of units of product delivered by DC i to meet the external demand $d_i(t)$. We assume $0 \leq y_i(t) \leq d_i(t)\ \forall i, t$.
(5) **Total demand (auxiliary variable)** $D_i(t) = d_i(t) + \sum_{j \in V_j} u_{ji}(t-1)$, $\forall i$.
(6) **Total availability (auxiliary variable)** $S_i(t) = x_i(t) + \sum_{j \in U_i} v_{ji}(t - \omega_{ji})$, $\forall i > 0$ and $S_0(t) = \infty$. Here ω_{ij} is the transportation lead-time from DC j to DC i. Thus for $j > 0, i = 3, \ldots, 6$, $\omega_{ji} = 1$ and for $j = 0, i = 1, 2$, $\omega_{ji} = l_i$.
(7) **Total amount sent (auxiliary variable)** $M_i(t) = y_i(t) + \sum_{j \in V_i} v_{ij}(t)$. We assume that $M_i(t) \leq$

$S_i(t)\ \forall i$.

For each DC i, the cost criterion is given by

$$min_{\substack{u_{ij}, j \in U_i \\ v_{ij}, j \in V_i}} E\{\sum_{t=0,1,\ldots} s_i max\{0, d_i(t) - y_i(t)\} + h_i x_i(t)\} \quad (1)$$

where

$$y_i(t) = min(d_i(t), S_i(t) - \sum_{j \in V_i} v_{ij}(t)) \quad (2)$$

Here E is the expectation operator defined with respect to the corresponding demand distribution. The storage dynamics for each distribution center is given by

$$x_i(t+1) = S_i(t) - M_i(t). \quad (3)$$

We can simplify our formulas by introducing an new function $z_i(t)$ called *excess demand* function which is difference between total demand and availability, that is,

$$\begin{aligned} z_i(t) &= D_i(t) - S_i(t) \\ &= d_i(t) + \sum_{j \in V_i} u_{ji}(t-1) \\ &\quad -x_i(t) - \sum_{j \in U_i} v_{ji}(t-1). \end{aligned} \quad (4)$$

We can rewrite (1) and (3) in terms of $z_i(t)$ as follows.

$$\begin{aligned} &min_{\substack{u_{ij}, j \in U_i \\ v_{ij}, j \in V_i}} E(\sum_{t=0,1,\ldots} s_i max\{0, z_i(t) \\ &- \sum_{j \in V_i} (v_{ij}(t) - u_{ji}(t))\} + h_i[d_i(t) \\ &-(z_i(t) + \sum_{j \in V_i} (v_{ij}(t-1) - u_{ji}(t-1)))]) \end{aligned} \quad (5)$$

$$\begin{aligned} &x_i(t+1) = \\ &max\{0, -[z_i(t) + \sum_{j \in V_i} v_{ij}(t) - u_{ji}(t-1)]\}. \end{aligned} \quad (6)$$

From (4) and (6), we can obtain the following recursion for the excess demand dynamics.

$$\begin{aligned} &z_i(t+1) = d_i(t+1) + \sum_{j \in V_i} u_{ji}(t) \\ &- \sum_{j \in U_i} v_{ji}(t - \omega_{ji} + 1) \\ &-max\{0, -[z_i(t) + \sum_{j \in V_i} v_{ij}(t) - u_{ji}(t-1)]\}. \end{aligned} \quad (7)$$

Next we define the consolidated discrete dynamic model for each DC i. This model constitutes the input data for our continualization procedure that we will describe in the companion paper [22]. To define this consolidated model, we first need to define an auxiliary variable $\xi_i(t)$, the planning decision vector $w_i(t)$, the input vector $g_i(t)$, and the maximum supply delay τ_i associated with DC i.

Suppose that $V_i = \{j_1 < \ldots < j_{m_i}\}$, then

$$w_i(t) = \begin{bmatrix} w_i^1(t) \\ \vdots \\ w_i^{m_i}(t) \end{bmatrix} = \begin{bmatrix} v_{ij_1}(t) \\ \vdots \\ v_{im_i}(t) \end{bmatrix} \quad (8)$$

$$g_i(t - \tau_i) = \begin{bmatrix} \sum_{j \in V_i} u_{ji}(t - \tau_i - 1) \\ \sum_{j \in V_i} u_{ji}(t - \tau_i) \\ \sum_{j \in U_i} v_{ij}(t - \tau_i - \omega_{ij}) \end{bmatrix} \quad (9)$$

where

$$\tau_i = max_{j \in U_i}\{\tau_j + \omega_{ji}\} \quad (10)$$

and $\tau_0 = 0$. Then the excess demand dynamics can then be rewritten as the following stochastic iteration.

$$z_i(t+1) = SAT(\xi_i(t))\xi_i(t) + C_2 \cdot g_i(t - \tau_i) + d_i(t) \quad (11)$$

where

$$SAT(f) = \begin{cases} 1 \text{ if } f > 0 \\ 0 \text{ if } f \leq 0 \end{cases} \quad (12)$$

and

$$\xi_i(t) = -z_i(t) + B^i \cdot w_i(t - \tau_i) + C_1 \cdot g_i(t - \tau_i) \quad (13)$$

with $B^i = [1, \ldots, 1]$ is a row vector of length m_i, $C_1 = [1, 0, 0]$, and $C_2 = [0, 1, 1]$.

The delay τ_i introduced in the excess demand dynamic equation of DC i (11) provides a coordination mechanism between the actual flow of the product and the state of the distribution center. This is not the only possible mechanism, but it is a conservative one. It can be easily shown that this mechanism guarantees that real time policies generated from our agent procedure are feasible in the mean sense relative to the probability space induced by the demand on the state space. This mechanism, or one like it, is necessary if the MAHCA system is to be implemented as a real time planner. As we shall see below, in order to ensure inter-agent synchronization, we need additional dynamics.

Note that we have formulated optimization problem P_i solved by the planning agent for DC i to be independent of the optimization problems solved by the planning agents for the other distribution centers in the system. Thus we need to establish constraints that will ensure that the solution of P_i is compatible with the solutions of the other P_j's. Our MAHCA technology provides two ways to address this synchronization problem: (1) Real-time synchronization and

(2) Rule based synchronization. In real-time synchronization, sensory data is used to establish whether the current plan behavior coincides with the model behavior. A description of real time synchronization via a MAHCA network can be found in [11, 23]. We shall not follow this approach in this paper. Instead, we shall describe a model for rule-based synchronization associated with the multi-echelon system.

For each DC i, we define a function $\eta_i(t)$ by the following iteration.

$$\eta_i(t+1) = \eta_i(t) + \sum_{j \in U_i} (u_{ij}(t-1-\tau_i) - v_{ji}(t-\tau_i)) \quad (14)$$

where $\eta_i(0) = 0$. We note that $\eta_i(t)$ measures the cumulative discrepancy between what a DC i orders and what it receives. For optimal performance of the system, we need

$$\sum_t E(\eta_i(t)) = 0 \quad (15)$$

for all i. That is, if $\sum_t E(\eta_i(t)) > 0$, then the optimal policy for DC i introduces a negative bias on the policies of the upstream DC j's. On the other hand, if $\sum_t E(\eta_i(t)) < 0$, then the system may exhibit unbounded behavior.

We shall implement a synchronization rule by incorporating (14) and (15) into the state equations of the model and by adding $\eta_i(t)$ to the criterion. First, rewrite the expression in (3) inside the expectation operator in terms of $\xi_i(t)$ taking care of the first and last terms in the summation. Then we add the term $\eta_i(t)$ to the summand to obtain the following expression.

$$\sum_t \{s_i(-\xi_i(t) \quad + \quad SAT(\xi_i(t))\xi_i(t)) \\ + h_i[d_i(t) + \xi_i(t)] + \eta_i(t)\} \quad (16)$$

Next we define a summing variable, $J_i(t)$ as follows.

$$J_i(t+1) = J_i(t) + (h_i - s_i\xi_i(t) \\ + SAT(\xi_i(t))\xi_i(t) + h_i d_i(t) + \eta_i(t) \quad (17)$$

with initial condition $J_i(0) = J_{i_0}$.

The criterion for the the planning agent of DC i then simply becomes

$$min_{w_i, u_{ji}: j \in U_i} E\{J_i(N)\} \quad (18)$$

where N is the time horizon of the plan.

Our final optimization problem characterizing the policies of each DC i is given by

$$min_{w_i, u_{ji}: j \in U_i} E\{J_i(N)\} \quad (19)$$

subject to the dynamic constraints (11),(14) and (17) and their corresponding initial conditions plus the structural constraints that $\forall t$, $u_{ij}(t) \geq 0$, $v_{ij}(t) \geq 0$ and $\xi_i(t) \geq 0$. We note that this formulation is characterized by a set of iterations constraining a terminal optimization problem. This is always the result of our model building operation. Notice also that once the conservation principles characterizing the chain are given the procedure is almost completely mechanical.

4. Conclusions

In this paper, we have described an Multiple Agent Hybrid Control architecture for the deployment of a reactive, agent based planner for a single product multi-echelon inventory system. We modeled the distribution network for the single product as a directed graph whose vertices are distribution centers which can supply outside demand and where a directed edge from i to j represents the fact that DC i can act as a source of supply for DC j. We assume that the demand for product p at vertex i is a stochastic process with a given a cumulative density function $F_i^t(\cdot)$.

We then described a procedure to construct a local optimization problem for each DC i in terms of the external demand $d_i(t)$ for product p at DC i at stage t, the per unit cost of storage h_i and the per unit cost for failing to meet the demand, the excess demand function $z_i(t)$ which is the difference between the total demand and total availability of p at DC i at time t, a function $\eta_i(t)$ which measures the cumulative discrepancy between what DC i orders and what it receives, and local decision variable for how many units of product are received by DC i from upstream suppliers and how many units are sent to downstream distribution centers. The local optimization criterion can be expressed in the form

$$min_{w_i, u_{ji}: j \in U_i} E\{J_i(N)\}, \quad (20)$$

where J is a simple summing variable which is a function of the variables described above and E is a the expectation operator, subject to the obvious constraints of the system. In this model, we have formulated the optimization criterion for the planning agent of DC i to be independent of the optimization problems for the agents of other distribution centers. Nevertheless, we can employ a simple mechanism for ensuring that these local optimization problems do not conflict by ensuring that the cumulative discrepancy function

$\eta_i(t)$ satisfies

$$\Sigma_t E(\eta_i(t)) = 0.$$

This paper is the first part of a two part presentation on a hybrid systems approach to supply chain management problems. In the companion paper [22], we shall describe a continualization procedure which can be applied to the optimization problem described in (20) to reduce it to a continuous optimization problem that can be handled by MAHCA agents.

References

[1] Alur, R., Henzinger, T. A., Sontag, E. D., eds. *Hybrid Systems III*, Lecture Notes in Computer Science **1066** (Springer-Verlag, Berlin, 1996).

[2] Antsaklis, P., Kohn, W., Nerode, A, and Sastry, S. eds., *Hybrid Systems II*, Lecture Notes in Computer Science **999** (Springer-Verlag, Berlin, 1995).

[3] Antsaklis, P., Kohn, W., Nerode, A, and Sastry, S. eds., *Hybrid Systems IV*, Lecture Notes in Computer Science **1273** (Springer-Verlag, Berlin, 1997).

[4] Crossley, J.N., Remmel, J.B., Shore, R.A. and Sweedler, M.E., *Logical Methods: In Honor of Anil Nerode's Sixtieth Birthday* (Birkhauser, Boston, 1993).

[5] Ge, X. , W. Kohn, A. Nerode, and J.B. Remmel, "Multiple agent hybrid control: carrier manifolds and chattering approximations to optimal control," *Proceedings of the 33rd IEEE Conference on Decision and Control CDC'94* (IEEE Press, 1994), 4221-4227.

[6] Ge, X., Kohn, W., Nerode, A. and Remmel, J.B., "Hybrid Systems: Chattering Approximations to Relaxed Controls", in [1] (1996); 76-100.

[7] Grossman, R.L., Nerode, A., Ravn, A., Rischel, H. eds., *Hybrid Systems*, Lecture Notes in Computer Science **736** (Springer-Verlag, Berlin, 1993).

[8] Kohn, W., "Declarative Control Architecture" *Communications of the ACM*, Aug 1991,Vol. 34, No. 8.

[9] Kohn W., Nerode, A., "Multiple-Agent Hybrid Systems", *Proceedings of the 31st IEEE Conference on Decision and Control CDC'92* (IEEE Press, 1992); 2956-2972.

[10] Kohn, W., Nerode, A., "An Autonomous Systems Control Theory: An Overview", *Proceedings of IEEE CACSD'92*, March 17-19, Napa, Ca.; 200- 220.

[11] Kohn, W.,Nerode, A. "Multiple Agent Hybrid Control Architecture", in [7] (1993); 297-315.

[12] Kohn, W., Nerode, A.., "Models For Hybrid Systems: Automata, Topologies, Controllability, Observability", in [7] (1993); 317-356.

[13] Kohn W., Nerode, A., "Multiple Agent Hybrid Control Architecture", in [4] (1993); 593-623.

[14] Kohn, W., Nerode, A., Remmel, J.B., " Hybrid Systems as Finsler Manifolds: Finite State Control as Approximation to Connections", in [2] (1995); 294-321.

[15] Kohn, W., Nerode, A., Remmel, J. B. , Yakhnis, A., "Viability in hybrid systems," *Theoretical Computer Science* **138** (1995), 141-168

[16] Kohn, W., Nerode, A., Remmel, J.B., "Feedback Derivations: Near Optimal Controls for Hybrid Systems", *Proceedings of 1996 IMACS Multiconference on Computation Engineering in Systems Applications (CESA'96)*, vol. 2, (1996), 517-521.

[17] Kohn, W., Nerode, A., Remmel, J.B., "Continualization: A Hybrid Systems Control Technique for Computing", *Proceedings of 1996 IMACS Multiconference on Computational Engineering in Systems Applications (CESA'96)*, vol. 2, (1996), 507-511.

[18] Kohn, W., Nerode, A., Remmel, J.B., "Agent Based Velocity Control of Highway Systems," in [3] (1997); 175-214.

[19] Kohn, W. and Remmel, J.B., "Digital to Hybrid Program Transformations", Proceedings of the 1996 IEEE International Symposium on Intelligent Control, 342-347.

[20] Kohn, W. and Remmel, J.B., "Hybrid Dynamic Programming, *Hybrid and Real Time Systems* (O. Mahler, ed.), Lecture Notes in Computer Science 1201, Springer-Verlag, (1997), 391-396.

[21] Kohn, W. and Remmel, J.B., "Implementing Sensor Fusion using a Cost Based Approach", Proceedings of 1997 American Conference of Control.

[22] Kohn, W. and Remmel, J.B., "Multi-Echelon Inventory Planning System II: Continualization", This volume.

[23] Kohn, W., Remmel, J. B. , Nerode, A. , "Automaton Comparison Procedure for the Verification of Hybrid Systems," *Proceedings of 5th IEEE Mediterranean Conference on Control and Systems (MED'97).*