Multiple Agent Hybrid Control for Manufacturing Systems

Wolf Kohn[*]
Sagent Corporation, 11202 SE 8th Street #J140
Bellevue, WA 98004-6420
e-mail: wk@sagent.com
Jeffrey B. Remmel[†]
Sagent Corporation and Department of Mathematics, University of California
La Jolla, CA 92093
e-mail: jremmel@ucsd.edu
Anil Nerode[‡]
Sagent Corporation and Mathematical Sciences Institute, Cornell University
Ithaca, NY 14853
e-mail: anil@math.cornell.edu
John James[§]
Sagent Corporation
e-mail: james@sagenteast.com

1 Introduction

Modeling and simulation support for virtual manufacturing requires support for the full range of analytical activities performed during an acquision process. During system *Concept Exploration and Development*, virtual prototypes can assist in developing a system produciton strategy, performing industrial base assessments, performing producibility engineering and planning, analyzing material/process capabilities, creating a tooling concept, assessing production risk, and estimating manufacturing costs. During system *Demonstration/Validation*, virtual prototypes can assist in constructing manufacturing process models, CAD/CAM models and factory simulation models which comprise the virtual manufacturing environment. The virtual manufacturing environment can aid in developing the preliminary manufacturing plan, delivering producibility estimates, conducting material/process tradeoffs, resolving production risk, and performing QA planning. During system *Engineering and Manufacturing Development*, the virtual prototypes and virtual manufacturing environment can be used with live-production demonstrations in developing a final manufacturing plan, completing MANTECH development, performing production readiness reviews, performing tooling design, completing detailed production planning, performing process proofing, and performing long lead procurement. During system *Production and Deployment*, the virtual prototypes and virtual manufacturing environment can be used to help execute the manufacturing plan and make decisions concerning

---

[*] Research supported by SDIO contract DAA H-04-93-C-0113, Dept. of Commerce Agreement 70-NANB5H1164.

[†] Research supported by U.S. Army Research Office contract DAAL03-91-C-0027, SDIO contract DAA H-04-93-C-0113, Dept. of Commerce Agreement 70-NANB5H1164 and NSF grant DMS-9306427.

[‡] Research supported by U.S. Army Research Office contract DAAL03-91-C-0027 and SDIO contract DAA H-04-93-C-0113, Dept. of Commerce Agreement 70-NANB5H1164.

[§] Research supported by SDIO contract DAA H-04-93-C-0113, Dept. of Commerce Agreement 70-NANB5H1164.

production, second sources and spares. During system *Operation and Support*, the virtual prototypes and virtual manufacturing environment can be used to assess production and process advances.

In this paper, we discuss how our generic hybrid control architecture, Multiple Agent Hybrid Contol Architecture (MAHCA), can be implemented to capture the complexity of the manufacturing process in an incremental fashion. We provide an example of how a MAHCA network of agents can be used to solve a planning, scheduling and control problem.

2 Current Industry Status

Implementation of factory planning, scheduling and control requires synchronization of complex events and continuous processes. The quality of factory schedules in smoothly synchronizing these events and process determines return on investment for factory components. Improved utilization would affect quality and increase return on investment. The quality of factory schedules also impacts customer satisfaction through on-time delivery of goods. Raw materials (stock) are transformed into finished products via operations performed at a number of work stations. The manufacturing plan for a product constrains the path of the raw stock through the work stations and the scheduling system must assign this stock to an actual sequence of work stations where the operations prescribed in its plan are undertaken. A good example, having several dozen work cells, is a Floor Panels Factory. It would contain a work cell of one or more workstations where an "align-panel-with-support-frame" function would be performed.

The scheduler must pay attention to a number of factors:

Achievable Can the plan called for actually work in this factory? Can it interface between multiple jobs – Hundreds of parts are undergoing some work at any moment? Do they interfere with each other? Do two parts require the same tool at the same time? Is any work piece likely to be ruined because it spoiled, or because the fixture required to hold it was busy?

Material Is the required stock available to start this job?

Improvements Can the order of work be changed to eliminate the need to reconfigure a workstation?

Safety Does the schedule allow for required operator breaks, shift changes? Is the schedule safe?

Even without these subjective factors, the general scheduling problem cannot be solved by classical methods. To compound the difficulty of achieving a solution, schedules are rarely used to completion. A machine breaks, stock fails to arrive, a tools breaks; a change in the schedule is required.

Before a schedule change, one must consider that many workpieces are already flowing through the factory. The cost to restart this work is usually too high to consider. Thus, the scheduler must produce a new schedule quickly while taking new, possibly not previously anticipated, constraints into account.

The scheduler for a factory must produce the schedule by evaluating throughput, tardiness, work-in-progress, machine wear, and other metrics. Each of these input-output criteria can be measured as a number. These numbers are global criteria (apply to the factory as a whole), and, since the criteria are not associated with a particular step in the manufacturing process, treat the factory as a black box (i.e., to be analyzed by input-output considerations).

Contrast this with the problem of deciding what to do next at an individual work cell. The logical decisions discussed above, normally captured as events occurring at discrete instants in time, must be compatible with the laws of physics which govern the continuous operation of motors, conveyor belts, sensors, actuators and so forth. If the factory is equipped with a single super computer and a very fast, unfailing sensor environment, then one program can be tasked with deciding for every work cell what it should do next, in complete detail. If communication lines to that computer fail, the factory stops. If its sensors lie, the factory may not stop, which can be worse. If simultaneous events overload the supercomputer, everyone waits. If small amounts of another product need to be produced between production runs of the current product, the plant must be reconfigured.

These considerations argue for a multiple-agent approach with coordination among work cells to achieve reasonable performance. Technical details of the network are discussed in Section 3. The mathematical foundations of MAHCA, the Kohn-Nerode hybrid control extraction procedure, is outlined in section 4. While a high degree of autonomy among agents is needed to support a "divide-and-conquer" approach to meeting the complexity of producing workable plant schedules, it is possible for one controller to degrade factory performance by shuffling parts between two orders in a looping fashion, causing downstream workcells undue overhead in reconfiguring. Distribution of control is needed, so that work cells can make autonomous decisions, using advice from neighboring work cells, and performance criteria (goals) from a factory-wide goal-setting agent. Factory plans are made by considering certain constraints, normally in the context of a nominal manufacturing scenario. However, factory execution occurs in the context of actual decisions and events, which can deviate from the nominal scenario. In a general sense this is true for any application that spans the boundary between planning, where limited experimentation can be conducted, and situated activity, which has a much larger set of possible outcomes. The concept of an agent must appear on both sides of this boundary. Multiple-agent, declarative control promises to substationally contribute to analysis and resolution of factory planning, schedule and control.

3 The Multiple Agent Hybrid Control Architecture

In this section, we describe the main operational and functional characteristics an agent in a MAHCA network. As we mentioned in the introduction, our Multiple Agent Hybrid Control Architecture is implemented as a distributed system composed of agents and a communication network which we call the logic communication network. The architecture realizing this system operates as an on-line distributed theorem prover. At any update time, each active agent generates control actions as side effects of proving an existentially quantified subtheorem (lemma) which encodes the model of the plant as viewed by the agent. The conjunction of lemmas at each instant of time, encodes the desired behavior of the entire network. Each agent of MAHCA, consists of five modules: a Planner, a Dynamic Knowledge Base, a Deductive Inferencer, an Adapter and a Knowledge Decoder. We briefly overview the functionality of an agent in terms of its modules.

The basic architecture of an estimation agent is pictured in figure 1. The agent consists of 5 modules with the following functionality:

1. **Planner** The Planner constructs and repairs the agent state optimization criteria which we refer to as the Lagrangian associated with the agent. In particular, the Planner generates a statement representing the desired model of the plant as an existentially quantified logic expression herein refered to as the Plant Statement.
2. **Inferencer** The Inferencer determines whether there is a solution for the agent's relaxed variational optimization problem which is a near optimal trajectory where the agent's Lagrangian is used as a cost function. If there is such a solution, the agent infers control actions and sends data to the other agents. Otherwise it infers failure terms and a new state for the agent and reports the failure to the other agents. In particular, the Inferencer determines whether the Plant Statement is a theorem in the theory currently active in the Knowledge Base. If the Plant Statement logically follows from the current status of the Knowledge Base, the inferencer generates, as a side effect of proving this Plant Statement to be true, the current control signals. If the Plant Statement does not logically follow from the current status of the Knowledge Base, that is, the desired behavior is not realizable, the inferencer transmits the failed terms to the Adapter module for replacement or modification.
3. **Adapter** The Adapter repairs failure terms and constructs correction terms.

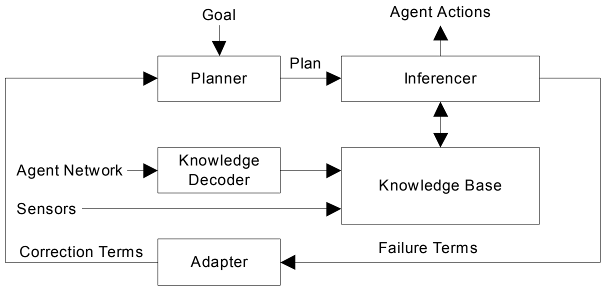

Figure 1: MAHCA Agent Architecture

4. **Knowledge Base** The Knowledge Base stores and updates the agent's plant model and constraints. The Knowledge Base also stores the requirements of operations or processes of the plant. It also encodes system constraints, inter-agent protocols and constraints, sensory data, operational and logic principles and a set of primitive inference operations defined in the domain of equational terms.

5. **Knowledge Decoder** The Knowledge Decoder receives and translates the other agent's data.

4 The Kohn-Nerode Extraction Procedure

Recall that a simple hybrid system consists of a continuous plant which interacts with a control automaton at times Δn. A hybrid system has a hybrid state, the simultaneous dynamical state of all plants and digital control devices. Properly construed, the hybrid states will form a differentiable manifold which we call the *carrier manifold* of the system. To incorporate the digital states as certain coordinates of points of the carrier manifold, we "continualize" the digital states. That is, we view the digital states as finite real valued piecewise constant functions of continuous time and then we take smooth approximations to them. This also allows us to consider logical and differential or variational constraints on the same footing, each restricting the points allowed on the carrier manifold. In fact, all logical or discontinuous features can be continualized without practical side-effects.

Every constraint of the system, digital or continuous, is incorporated into the definition of what points are on the carrier manifold. Lagrange constraints are regarded as part of the definition of the manifold as well, being restrictions on what points are on the manifold. We then look for a feedback function $F(x) = y$ of points x on the manifold. For the purpose of this paper, the value of $F(x) = y$ will be a vector in the tangent space of the manifold such that $y = \dot{x}$. Thus we can think of $F(x)$ as telling us what direction to steer when in state x to get to the goal which in this case is to provide an unbiased estimate of the plant state. The goal will always be of the form to end in state x_1 or in a neighborhood of x_1, representing the desired estimate, steering in the prespecified direction y_1.

Next we shall briefly review the highlights of that extraction procedure [14, 15, 16, 17].

First, corresponding to an autonomous problem,

$$\dot{x} = F(x, t),$$

we formulate a non-negative Lagrangian $L(x, \dot{x})$ where we have included time as an extra variable $t = x_{n+1}$ with the constraint that $\dot{x}_{n+1} = 1$.

We assume that the Lagrangian $L(x, \dot{x})$ is positive definite in $\dot{x}$ along an extremal tube. That is,

$$(g_{ij}(x, \dot{x}))\big|_* = \Big(\frac{\partial^2(L(x, \dot{x}))}{\partial \dot{x}_j \dot{x}_i}\Big)\Big|_*$$

is a positive definite matrix for each fixed x where (*) restricted to an extremal tube of trajectories which are solutions to $\dot{x} = F(x, t)$.

Under suitable convexity assumptions on L, we can then define a metrix gound form ds on the carrier manifold such that

$$ds^2 = \sum_{ij} g_{ij}(x, \dot{x})\big|_* dx_i dx_j \tag{1}$$

A key result of the theory is the fact that the geodesics under the metric ground form induced by L are the extremals of the corresponding parametric calculus of variations problem which is to find an admissible curve on the carrier manifold which minimizes

$$\int_{t_0}^{t_1} L(x(t), \dot{x}(t))\, dt \tag{2}$$

satisfying given endpoint conditions. That is, the geodesics are solutions to the Euler-Lagrange equation

$$\frac{d}{ds}\frac{\partial L(x,\dot{x})}{\partial \dot{x}_i} - \frac{\partial L(x,\dot{x})}{\partial x_i} = 0. \tag{3}$$

Second, the performance specification for the plant is reformulated as the requirement that the plant state trajectory has an integral within a user prescribed ϵ of its minimum. A control function of state that gives such a trajectory is called an ϵ-optimal control.

Third, we replace $L(x, u)$ by $L^*(x, u)$ which is the convexification of L with respect to u. Then an existence theorem for relaxed controls can be applied to $L^*(x, u)$ which ensures that the optimal control problem has a relaxed solution. We can approximate this relaxed solution to obtain an ϵ-optimal control for the original problem.

Fourthy, for any prespecified ϵ, we compute an ϵ-optimal control for the original problem [2] and implement ϵ-optimal control as a finite state, physically realizable, control automaton [14]. The actual control law issued by the finite control automaton at state x is a chattering control [2, 3]. The chattering is between approximations to some local extrema of $L(x, u)$.

Fifth, we repeat the derivation of the control automaton on-line, if due to unmodelled dynamics, the performance specification is not met by observing some violation of the Noether invariant conditions. Then we substitute a new control automaton for the old as required. This type of substitution represents the basic adaptive capability of the architecture.

4.1 Manufacturing Example – A Plastic Excrusion Process

Next we will illustrate some of MAHCA features with an example. The figure below illustrates MAHCA with 5 permanent agents controlling a manufacturing process for producing molded plastic products. The plant is composed of 5 functional cells.

Cell 1 performs the shipping and receiving fuctions and also includes an automated forklift which can transport materials and products through the factory. The functions of Cell 1 are controlled and scheduled by the Shipping and Receiving MAHCA agent. Note that the agent in this cell interacts with a human operator (MAHCA can perform semi-autonomous tasks).

Cell 2, the Accounting Agent which performs the accounting and plant coordination functions. This agent also performs inventory and control functions on the element of Cell 3, an automated warehouse bin.

Cell 3 is composed of a blender that mixes the raw materials and produces the resin mix used in the final products. The Blender is controlled by the Blender Control agent.

Cell 4 is composed of a molder and the Molder Control agent. This agent also performs control functions on the component of Cell 5, the packaging unit, which is also controlled by MAHCA's Packaging agent.

Let us illustrate how this process operates. Suppose that an order for plastic bottles arrives. This order becomes a goal to be satisfied by the Accounting Agent. The agent generates a command to update the factory's backlog file and the scheduler. It also sends a request for service to the shipping and receiving agent for the use of the automatic forklift to transport raw materials from the warehouse to the input bin of the blender cell. The shipping and receiving bin issues a command to update the scheduler for the automated forklift and sends a request to the accounting agent for the ordinates (in warehouse coordinates) of the bin(s) in the warehouse where the raw material is (are) located. Once this information is available, the shipping and receiving agent commands the forklift to that location. If the accounting agent determines that the raw material is not available it issues a command to purchase the raw material and it communicates this to the shipping and receiving agent which issues a command to update the delivery ledger.

We can see that the operations sketched above are

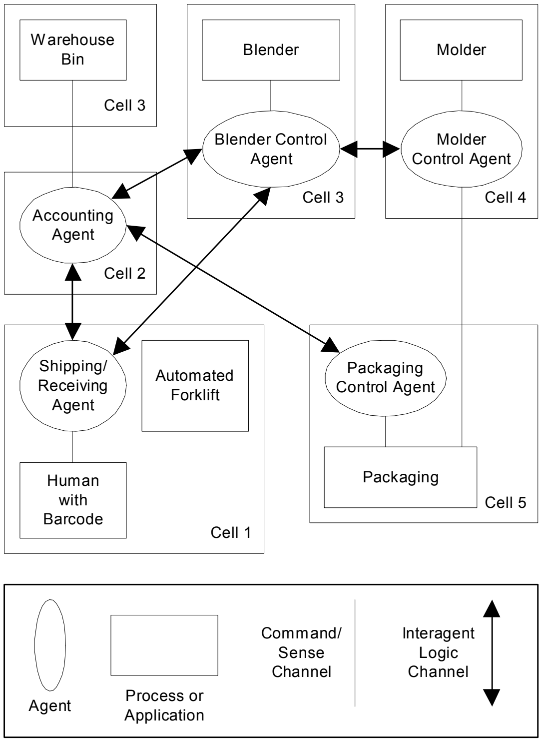

Figure 2: A Logic Network of MAHCA Agents

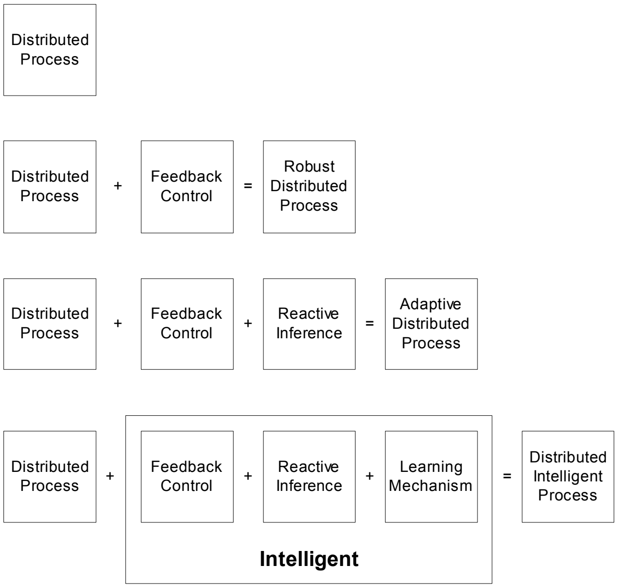

Figure 3: Control of Distributed Process

more or less the ones that human agents would carry out if in control. Agents communicate among themselves through the logic network connecting them. The information that an agent stores in its knowledge base is directly associated with the aspects of the process it is controlling and the information about the other agents with which it directly communicates.

If a permanent agent finds that its computational load is high enough that it would cause its function to fall out of schronism, it can spawn a helper, a Temporary Agent, which inherits some of the knowledge of the permanent agent. This temporary agent executes a specific function, determined by the spawning agent, in the processor selected by the spawning agent. Once that function is executed the temporary agent self-destructs.

The on-line reactive matching of current logic and continuum constraints from plant sensors with user-defined logic and continuum models is the basis for generating procedures to interface heterogeneous components of the manufacturing process, and for our belief that the resulting architecture will support incremental expansion of new components with greatly reduced requirements for expensive experimentation validation. We do not expect to fully eliminate the need for experimentation because the degree of “trust” in the newly composed architecture will depend on the rules for composition of the components. However, to the degree that the composition rules are correct, the methodology will be a formally correct composition of the components, the focus of the verification and validation effort will be raised to the component level and the results will be reusable across the confederation of components.

5 Conclusions

We have reviewed the Multiple Agent Hybrid Control Architecture. MAHCA allows for the on-line reactive matching of current logic and continuum constraints from plant sensors with user-defined logic and continuum models is the basis for generating procedures to interface heterogeneous components of the manufacturing process. The MAHCA architecture will support incremental expansion of new components with greatly reduced requirements for expensive experimentation validation. We do not expect to fully eliminate the need for experimentation because the degree of “trust” in the newly composed architecture will depend on the rules for composition of the components. However, to the degree that the composition rules are correct, our hybrid systems methodology will yield a formally correct composition of the components. This will allow the focus of the verification and validation effort will be raised to the component level and allow the construction of agents which are reusable across the confederation of components.

References

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

[2] X. Ge, W. Kohn, A. Nerode, and J. B. Remmel: Algorithms for Chattering Approximations to Relaxed Optimal Control. MSI Tech. Report 95-1, Cornell University, (1995).

[3] X. Ge, W. Kohn, A. Nerode, and J. B. Remmel: Hybrid Systems: Chattering Approximations to Relaxed Controls. *Hybrid Systems III*, Lecture Notes in Computer Science, Springer-Verlag (1996), (to appear).

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

[5] Kohn, W., “A Declarative Theory for Rational Controllers” Proceedings of the 27th IEEE CDC, Vol. 1, pp 131–136, Dec. 7–9, 1988, Austin, TX.

[6] Kohn, W., “Declarative Hierarchical Controllers” Proceedings of the Workshop on Software Tools for Distributed Intelligent Control Systems, pp 141–163, Pacifica, CA, July 17–19, 1990.

[7] Kohn, W., “Declarative Control Architecture” CACM Aug 1991, Vol 34, No 8.

[8] Kohn, W., J. James, and A. Nerode, “Multiple-Agent Reactive Control of Distributed Interactive Simulations (DIS) through a Heterogeneous Network”’ Proceedings of the Army Research Office Workshop on Hybrid Systems and Distributed Interactive Simulations, Research Triangle Park, NC 28 Feb. – 1 Mar. 1994.

[9] Kohn, W., J. James, A. Nerode, and J. Chandra, “An Architecture for Incremental Construction of Distributed, Heterogeneous Systems”, Proceedings of the Workshop on Software Architectures of the International Conference on Software Engineering, Seattle, WA, April 1995.

[10] Kohn, W., J. James, A. Nerode, K. Harbison, and A. Agrawala, “A Hybrid Systems Approach to Computer-Aided Control System Design,” IEEE Control Systems, April 1995, 14–24.

[11] Kohn, W. and A. Nerode, “An Autonomous Systems Control Theory: An Overview” Proc. IEEE CACSD’92, March 17–19 Napa Ca.

[12] Kohn, W., and A. Nerode, “Multiple-Agent Hybrid Systems” Proc. IEEE CDC 1992, vol 4, pp 2956–2972.

[13] Kohn, W. and A. Nerode, “Multiple Agent Declarative Control Architecture”, Hybrid Systems, Springer Verlag, 1993, Volume 736, R. L. Grossman, A. Nerode, T. Rischel, A. Ravn, eds.

[14] Kohn W. and A. Nerode, “Models for Hybrid Systems: Automaton, Topologies, Control, Controllability, Observability”, Hybrid Systems, Springer Verlag, 1993, Volume 736, R. L. Grossman, A. Nerode, T. Rischel, A. Ravn, eds.

[15] Kohn, W. and A. Nerode, “Multiple Agent Autonomous Control: A Hybrid Systems Architecture” Logical Methods In Honor of Anil Nerode’s Sixtieth Birthday, N. C. Crossley, J. B. Remmel, M. E. Sweedler, Eds., Birkhauser, Boston, 1993.

[16] Kohn W. and Nerode A. “Multiple Agent Hybrid Control Architecture” MSI Report 93-11 Cornell U.

[17] W. Kohn, A. Nerode, and J. B. Remmel, Hybrid Systems as Finsler Manifolds: Finite State Control as Approximation to Connections. In [1] (1995).

[18] Kohn, W., Nerode, A. and Remmel, J.B., “Feedback Derivations: Near Optimal Controls for Hybrid Systems”, to appear in CESA’96.

[19] Kohn, W., Remmel, J.B., and Nerode, A., Scalable Data and Sensor Fusion via Multiple Agent Hybrid Systems, to appear in AC-IEEE Trans, special issue in Hybrid Control Systems.

[20] Young, L.C. “Optimal Control Theory” Chelsea Publishing Co., NY, 1980.