Hybrid Automata in the Context of Additive Manufacturing

J.C. Boudreaux
Guest Researcher
National Institute of Standards and Technology
U.S. Department of Commerce

REVIEWED

Abstract. To maintain the forward momentum of additive manufacturing technology, it is necessary to thoroughly evaluate new and potentially useful technological developments in this field. One such development is the intense interest being directed to the field of *hybrid automata (HA).* Hybrid automata combine both the discrete processing behavior of finite automata as well as the continuous, or *flow*, behavior of dynamical systems. At this point, some important results on hybrid automata have been obtained, but many open questions remain, including those concerning the *decidability* of HS operational procedures. (Recall that decidability is directed to a decision problem, that is, a definite true-or-false response given by an effective procedure.) Some important decidability results for HAs have been obtained. For example, in [Henzinger *et al.*1998] the *reachability* problem for timed automata (an HA class) has been convincingly shown to be decidable. However, it should also be noted that subtle and difficult issues have been identified, *e.g.*, [Fraenzle 1999], [Asarin, Collins, 2005]. This paper will provide a summary review of the operational features of HAs as they might pertain to additive manufacturing, and then briefly consider the following technical issues: (i) are the classical models of the real numbers best suited to deal with the necessarily approximate measures of physical systems or would *non-standard analysis* of [Robinson 1996] be a better fit; and (ii) would the introduction of *"noisy semantics"* and *finite arithmetic precision*, following [Freidlin, Wentzell 1984], be a better work around?

I. Introduction

Additive Manufacturing (AM) has had significant impact on the production of engineering prototypes and more recently on the manufacture of functional parts, especially the direct fabrication of molds, dies, and other tooling. As now practiced, this style of manufacturing is based on additive manufacturing processes which may resolved into the following steps: first, to create a *solid geometric model* of the part to be built and to select a part orientation; second, to form an ordered sequence of *planar slices* which in effect decompose the solid into a sequence of thin cross-sectional *polyhedral* layers; and then to fabricate the part by producing the polyhedral designs by applying such methods as stereolithography, selective laser sintering, laminated object manufacturing, or fused metal/ceramic deposition [7], [8].

Within the now-dominant AM paradigm, *parametric design* is based on the premise that an inventory of existing designs can be used to create *design templates* such that (1) the identified functional connections are coded by *embedded parametric expressions*, and (2) novel variant designs may be generated by assigning reasonable values to the parameters and then *evaluating* the resulting expressions in a precisely controlled manner. This program has proven to be significantly more difficult than anyone had expected. This paper is an attempt to construct a qualitative parametric engineering model of the product design process in the context of discrete part manufacturing. It seems natural to suppose that *engineered artifacts* may be resolved into components parts that can be usefully represented as *compact connected 3-manifolds with boundary* where the boundary consists of a possibly disconnected set of compact smooth 2-manifolds (*topological surfaces*). This style of manufacturing is based on additive manufacturing processes which may resolved into the following steps: create a *solid geometric model* of the part, defining surface boundaries and identifying interior (material) regions from the exterior (void) regions; select a part orientation and then form *planar slices*, decomposing the solid into a sequence of thin cross-sectional *polyhedral* layers; and *fabricate* the part by producing all of the polyhedra by any of several methods, including: stereolithography, selective laser sintering, laminated object manufacturing, fused deposition modeling, and three dimensional printing. In classical manufacturing work is applied to the surface of the part either through deformation process or by material removal. But in the present context the most useful characteristic of AM is that it reduces the 3-space shape of compact objects to a sequence of 2-space slices (and possibly some mild "skinning" conditions). [8]

A. Geometric AM Models and Applied Topology.

But there is still more to be observed here: the stack of planar slices, properly understood, is of very great importance in its own right. Not only does it allow manufacturers to produce that one specific part, but also, suitably transformed, it has the potential to produce *infinitely many geometrically distinct parts*, every one of which is *topologically identical* to the single specific part with which we began. This important notion should be more precisely defined. [19]

A topological space is a pair (*S*, O) where *S* is an unempty set of points and O is a set of subsets of *S*, called the *open sets*, which satisfies the following conditions: O contains both *S* and the empty set ∅, and is closed under finite intersections and arbitrary unions. A *closed set* is the set-theoretical complement of an open set. The collection C of all closed sets satisfies the *dual* conditions: C contains both ∅ and *S*, and is closed under finite unions and arbitrary intersections. Let *Y* be an arbitrary subset of *S* and *p* be a point, then *p* is a *limit point* of *Y* only if every open set of (*S*, O) contains a point of *Y* distinct from *p*. That is, elements of *Y* get arbitrarily close to *p.* Now suppose that *S* and *T* are topological spaces, then a function *f* from *S* to *T* is said to be *continuous*, or a *map*, provided it is an *into* function and for every subset *X* of *S* and point *p*, if *p* is a limit point of *X*, then *f(p)* is a limit point of *f(*X*)*, the image of X under *f*. Equivalently, *f* is continuous if it is *into* and the *inverse* image under *f* of every open set of *T* is an open set of *S*. Given this preliminary discussion, the fundamental meaning of *topological identity* is finally made clear: a function from spaces *S* to T is a *homeomorphism* provided it is a one-to-one onto function (a *bijection*) and both the function and its inverse are continuous.

Then, with the topological sketch in place, we can fashion at will an infinite variety of topologically identical geometric constructions in two steps: first, we can arbitrarily thicken all (or some) of the initial planar slices, while preserving all structural elements (such as holes or voids); and second, without tearing or puncturing through the thickened slices, we can reshape them at will, just as we would reshape wet clay … well almost!

B. Resemblances (1): Morse Theory.

We are not quite finished yet! What needs to be added to this story is some structuring principle which allows all of the infinitely many possible geometric constructions to at least *resemble* one another. Suppose that the geometry of our first part was a child's doll, say Raggedy Ann, then what we would like to have is a mechanism in place which would ensure that resemblance to Raggedy Ann would be at least roughly approximated. One way to do this is to fix the doll in a plausible fixed position, and then to scan her from just above her head (the initial scan point) to just below her shoes (the final scan point) while noting for each scan the distance from the initial scan. Of course, the results would be an ordered set (top-down) of planar slices! [6]

Again, we'll need a bit more topology [6], [7], [19]. An *n-dimensional manifold (n-manifold) M* is a separable Hausdorff topological space such that for every point *p*, there is a neighborhood of *p*, which is homeomorphic to Euclidean *n*-space or to Euclidean *n*-halfspace. Let *U* be an open set for some point p of *M* and let *h* be a homeomorphism from *U* into *n*-space, or equivalently, any open n-ball, then the pair (*U*, *h*) is called a *chart of dimension n.* An *atlas* A on *M* is a collection of charts whose open sets cover *M*. Suppose that *(U, h)* and *(V, k)* are charts whose open sets have a nonempty intersection, say *W*, then *h* and *k* given rise to a homeomorphism on the *n*-space images of *W* , specifically, $(kh^{-1})\colon h(W) \rightarrow k(W)$, and which establishes functional relationships between the *n*-tuples in the respective images of *W*. If all of the functions so established are differentiable (in class C^k or C^∞) then *M* is said to be a *differentiable (smooth) manifold*.

The collection of points (if any) with neighborhoods homeomorphic to the *n*-halfspace define the *boundary* of the manifold, ∂M. Then given two topological spaces X and Y, let $f,g\colon X \rightarrow Y$ be continuous functions. Then *f* and *g* are *homotopic* iff there is a continuous mapping $h\colon X \times [0,1] \rightarrow Y$ such that for all *x* in *X*, $h(x,0) = f(x)$ and $h(x,1) = g(x)$. The mapping *h* is a *homotopy between f and g* and the product space $X \times [0,1]$ is the *homotopy cylinder*. The existence of a homotopy between functions establishes that the functions can be continuously deformed into one another.

The final piece that needs to be added to this story is that there needs to be a closer connection between the topological structure of the compositions that we're proposing to manufacture and the *critical points* of smooth functions on the manifold (points at which all partial derivatives of the function are null). Critical points consist of *maxima*, *minima*, and *saddle points.* First, assume that *M* is a nice 3-mainfold and that ∂M, the boundary of *M*, is the disconnected union (possibly empty) of nice manifolds *V* and *W*. Intuitively, think of V and W as planes which sandwich *M* between them. Then let *f* be a smooth mapping of *M* into a real closed interval *[a,b]* such that $f^{-1}(a) = V$ and $f^{-1}(b) = W$. Second, generate slices through *M*, that is, if *x* in *[a,b]*, then $f^{-1}(x)$ is such slice. Not all such functions are useful. What we want to do is to restrict our attention to *Morse functions* whose critical points on *M* are *non-degenerate* (the Hessian matrix of partial second derivatives is non-singular at the point). On nice manifolds, these functions are plentiful. [6], [19]

The point of this exercise is to use *f* to scan *M* to build a CW-complex that has the same homotopy type as *M*. The cell complex is built up by a sequence of cell attachments, which is completely determined by the sequence of critical points. As *f* slides between critical points, the topology of the manifold does not change, that is, no cells need to be attached. When *f* reaches a critical point, the topology of the manifold changes in way that requires the cell complex to be updated by attaching a k-cell, where k is the *index of the critical point* (the number of negative eigenvalues of the Hessian matrix). Since nice manifolds are compact, there are at most finitely many critical points, which implies that the sequence of critical points p_i and the associated critical slices $f^{-1}(x_i)$ completely encode the intrinsic topology of M.

C. Resemblances (2): Reeb Graphs.

Once a Morse function f has been defined on M, a *Reeb graph* can be used to generate a *skeleton* of M [6], [19]. A quotient space is formed on $M \times \boldsymbol{R}$ by identifying points, which belong to the same connected component of a slice. That is, if p and q points, then $(p, f(p))$ is identified with $(q, f(q))$ if $f(p) = (f(q)$ and p and q are in the same connected component of $f^{-1}(f(q))$. The nodes of the Reeb graph will be the critical points of M and the edges will be given by the connected components between critical slices. From the perspective of Morse theory**,** the most interesting slicings are those which (1) begin and end with minimal patches and which (2) mark the *critical regions* of the *3*-manifold under study, that is, regions in which either the number of components or their topological properties change. An example of a change of the second kind is the first appearance of a new boundary patch (or void surrounded by boundary elements) in a patch whose earlier stages had none. Let slicings of this kind be called *Morse slicings*. In general, *3*-manifolds admit of many Morse slicings.

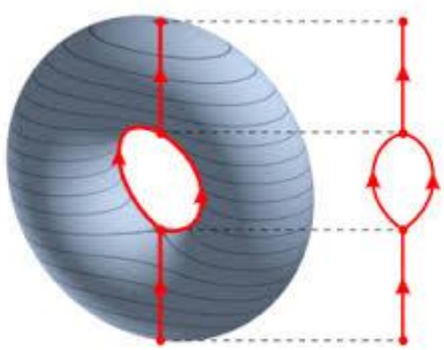

Fig. 1 : Reeb graph of torus

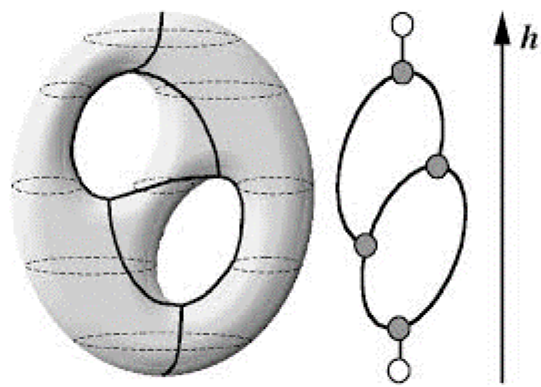

Fig. 2 : Reeb graph of 2-torus

In effect, what Reeb graphs really do is to create *stratified structures*, that is, a single component at the first (or top) level, which is called the *root* of the structure and which may be refined into a branching networks of components (the descendants) at the lower levels. The condition which must be satisfied is that every descendant node must be linked to the parent node by a unique sequence of links. Such organization structures are called *rooted trees*. Stratification also leaves the parent's boundary intact, allowing it to become a container of the boundaries the children.

Boundaries drawn in this manner impose a nest structure on the product: at the highest level is the product itself, followed by its immediate descendants, and so on, until all of the boundaries have been accounted for. In the artifact-as-network model, stratification boundaries mark components, which can then be identified by symbolic "tags" or labels. This symbolic structure permits stratified product networks to form a framework with respect to which the *design features* of components are defined. Features of one component may be either synthesized from those of its descendant components in a defined stratification or inherited by virtue of its links to other (in some cases, all other) components.

Given a 3-manifold K and a Morse slicing $(X_1,...,X_m)$, it is possible to construct a combinatorial variant of a Reeb graph. Under the assumption that the slicing has already been reduced by eliminating consecutive pairs of homeomorphic slices, then

(1) the vertices of the Reeb graph are components of slices; and
(2) the arrows of the Reeb graph link each component of slice X_i to all of the components of slice X_{i+1} which descend from it.

As in the smooth case, Reeb graphs concisely encode important topological properties of *3*-manifolds: the vertices encode the components of slices, and the arrows encode the *3*-manifolds (like cylindrical solids) which link a component of slice X_i with all of the components of slice X_{i+1} from which descend from it. The *3*-manifolds paired with Reeb graph arrows may have voids and tunnels. However, Reeb graphs are less successful in capturing properties affecting the geometric realizations of *3*-manifolds.

For example, there are many ways to embed an encoded Reeb graph into *3*-space which are not geometrically equivalent (isotopic) to one another. This has to do with the mathematics of knots in *3*-space. The multiplicity of these relational ties goes a long way in explaining the observed conceptual diversity among feature categories.

Again, *functional features* are inherited from above by considering the constraints imposed by the component's children. Specifically, the functional capabilities and operational behavior of components may be modeled by *transfer functions* which, given time-sequences of inputs (and possibly information about the current *state* of the component), produce time-sequences of outputs (and possibly a new state) after a specified time lag. *Mating* features are both synthesized from below and inherited from above because of the requirements imposed by assembly and joining processes. *Form* features are those through which all of the (often) contending requirements of the part are finally reconciled and by which the connectivity-induced constraints are resolved.

II. On the Architecture of AM: Two Recent Reports

The "planar slice" AM architecture remains a central paradigm and undoubtedly will continue to develop as it has done now for a quarter of a century. A very interesting example of current developments AM was described in a recent report in *Science.* Zheng *et al.* [21] have reported a way to build ultralight and ultrastiff mechanical materials by constructing a class of cellular materials, thereby mimicking natural foamlike structures, such as trabecular bone, plant parenchyma, and sponge, each of which combines low weight with superior mechanical properties. By reducing the degree of random porosity occurring in natural materials, they proposed to develop a class of materials that contains micro- and nanoscale building blocks arranged in an ordered hierarchy. The authors report that there were several proposals for the unit cells, including an octet-truss (a stretch-dominated structure) and a tetrakaidecahedron (a bend-dominated structure). The critical feature sizes of the unit cells ranged from ~20μm to ~40nm. The fabrication of the microlattices was enabled by *projection microstereolithography*, a layer-by-layer AM process which they describe as able to build "arbitrary three dimensional structures." The authors report that the specific stiffness of the Ni-P stretch-dominated metallic lattice stays very nearly constant, measured at 1.8×10^6 m^2/s^2 and 2.1×10^6 m^2/s^2 at densities of 14 mg/cm^3 and 40 mg/cm^3, respectively. The article concludes with the claim that "fabricating ordered lattice structures at these lengths scales brings them into the regime in which it becomes possible to design microstuctured functional materials with superior bulk-scale properties."

This interesting report is one example among many of the vigor of the AM engineering community. In thinking about this more carefully, and especially by happening across the *Biomedical Engineering Society (BMES)*, it struck me, as it has obviously struck many others, that a strongly supported collaboration between BMES (and similar groups) and the AM community would be mutually beneficial. The 2014 paper by Kamm and Bashir [16], entitled "Creating Living Cellular Machines," begins with the following comments: " One of the greatest opportunities lies in the potential to understand populations of multiple cell types and their interactions. … Yet, it can be argued that a tremendous gap exists between understanding processes at the level of a single cell and the behavior of large-scale tissues, *i.e.* how he local rules of interaction result in global functionalities and diverse prototypes. This is an issue involving complex systems of multiple interacting components that could fruitfully draw upon considerable advances in the engineering realms of forward design and manufacturing of large, complex systems." [16, p445] This paper suggests that a significant direction of the future growth for AM technology could be enhanced by investigating "soft" architectures, which are likely to dominate the biological approaches. In applications of this sort the analogue of the granular "planar slice" would be "soft" polymers, hydrogels, or other biomaterials. I presume that the introduction living cellular material would be accomplished by an injection process, and that this could result in local but significant deformation of the "soft" material. In this case, it also seems that mathematically useful description of the process is one which allows a deformable topological process, perhaps, a structural approach that jointly handles both the *temporal* and the *metric properties* of the space. This task is made no easier when such added features as cell motility and intercellular chemical diffusion between the introduced samples are taken into account.

III. Hybrid Automata

A. Toward a Definition of Hybrid Automata.

In section **IV** below, I intend to sketch out the view that this interesting class of *hybrid automata*, whose defining characteristic is that they are designed to function in both *discrete* and also in *continuous* modes, are plausible candidates for the role of supervisory controllers in the context of additive manufacturing. I put this proposal forward tentatively in the hope that the technical issues which would need to be resolve to accomplish this goal are still under very active discussion. [4], [13], [15]

The general notion of *hybrid automata (HAs)*is that they are able to display both *discrete* (or *jump*) as well as *continuous* (or *flow*) dynamics. The technical definition of hybrid automata begins by introducing *directed graphs*, usually abbreviated to *digraphs*. More precisely, a digraph is a structure **(*V*, *E*)** in which ***V*** an unempty set of vertices, which are also called nodes, and ***E*** is a possibly empty set of edges. It is also understood that the vertex set and the edge set of a digraph are mutually disjoint. It will be assumed that the edges have a definite *direction*, that is, every edge points *from* one vertex *to* another. If ***v*** and ***w*** are connected vertices, then edges between them may have one of two possible directions: $\boldsymbol{v} \rightarrow \boldsymbol{w}$ or $\boldsymbol{v} \leftarrow \boldsymbol{w}$, and this means that every edge is oriented. [2], [3], [5], [9]

The definition of digraphs given above admits a very large class of structures in terms of which the inter-component relational ties can be explicitly accounted for. For example, digraphs may have many (even infinitely many) distinct *parallel* and *antiparallel* edges between the same two vertices. The first step is straightforward: the *vertices* corresponding to assembled components are symbolically "tagged" and the arrows consist of both organization links that connect parents with their children and also a complex web of links between components.

To properly understand the implications of this 2-way hybrid structure, note that each vertex can be interpreted both as a place in which one or more actions are initiated by the node's immediate predecessors (if any). The interweaving of product and process data in a single structure is very useful. In effect, it supplies a single unified support system for many of the operational control methods. For example, if the digraph is interpreted in terms of the processes to be performed, then it contains the information needed to identify and initiate the actions that will be needed to produce the required effects. Hybrid automata also require that there may be a multiplicity of edges between any two vertices, that is, the digraphs which define that basic groundwork of hybrid automata are *multidigraphs.* The only edge configurations which are explicitly disallowed are *loops*, that is, edges which are directed from one vertex to itself. We now have enough basic information to proceed to a suitable definition of the functional components of formal model of computation.

(HA1) **Control Digraph**: a finite directed multidigraph **(*V*, *E*)** whose vertices are called *control modes*, or locations, and whose edges are called *control switches*.

(HA2) **Variables**:
(i) a finite set $X = \{x_1, x_2, \ldots, x_n\}$ of *real-valued variables*, where the index n is the dimension of the hybrid automata;
(ii) a finite set $\mathbf{D}X = \{\mathbf{D}x_1, \mathbf{D}x_2, \ldots, \mathbf{D}x_n\}$, which are first derivatives of the variables (with respect to time) during *continuous change*; and
(iii) a finite set ${}^+X = \{{}^+x_1, {}^+x_2, \ldots, {}^+x_n\}$ of "+" marked variables which represent values at the conclusion of *discrete change*.

(HA3) **Initial, Invariant, and Flow Conditions**: Three vertex labeling functions *init*, *inv*, and *flow* that assign to each control mode (*i.e.,* to each vertex $\boldsymbol{v}$ in the control digraph) three predicates:
(i) *init*($\boldsymbol{v}$) is a predicate whose free variables are in X which assigns *initial conditions* to each control mode;
(ii) *inv*($\boldsymbol{v}$) is a predicate whose free variables are in X which assigns *invariant condition*s to each control mode; and
(iii) *flow*($\boldsymbol{v}$) is a predicate whose free variables are in $X \cup \mathbf{D}X$ which assigns *invariant condition*s to each control mode.

The rule *(HA3)* explicitly introduces the logical notion "predicate" which is properly understood as an expression which is applied to one or more arguments, *i.e.*, either *variables* that may be assigned different values from one time to another or *constants* whose values are fixed in advance. Unlike functions, the values returned by the application of predicates are limited to a specific range of truth-values, usually limited to "true" or "false" but which can be extended to include a much larger collection of allowable values.

(HA4) **Events:** a finite set Σ of events and an edge labeling function *event:* $\boldsymbol{E} \rightarrow \Sigma$ that assigns a suitable control switch (*i.e.,* to edges in the control digraph) an event.

(HA5) **Jump Conditions**: a edge-labeling function *jump* that assigns to each control switch $\boldsymbol{e}$ in $\boldsymbol{E}$ a predicate *jump*($\boldsymbol{e}$) whose free variables are in $X \cup {}^{+}X$.

B. Putting Hybrid Automata to Work.

Having identified the components, the next step is to describe execution of hybrid automata. The first, and most obvious point, is that there are two operational modalities: a continuous modality in which the process is a flow, which may be pictured as a kind of evolution, and a modality in which the change is a discrete jump. [2], [3], [9], [11], [14], [15]

The idea is that both modalities can be properly handled by accepting variant of the familiar class of *labeled transition system* as an operational paradigm. In this case, we first identify a state space $\boldsymbol{Q}$, which may be infinite, and a subset $\boldsymbol{Q}^0$ of all initial states, then identify a (possibly infinite) set $\boldsymbol{Rel}$ of binary relations $\mathbf{r_j} \subseteq \boldsymbol{Q} \times \boldsymbol{Q}$. Each binary relation is a set of ordered pairs of states, so for any $\mathbf{r_j}$ we may distinguish *dom*($\mathbf{r_j}$), the set of all first members of the relation, and *rng*($\mathbf{r_j}$), the set of all second members of the relation.

The operation of a labeled transition system is not straightforward: (i) pick any initial state in $\boldsymbol{Q}^0$, (ii) given that state, say $\boldsymbol{q_0}$, select from $\boldsymbol{Rel}$ any relation $\mathbf{r_j}$ such that $\boldsymbol{q_0}$ is a element of *dom*($\mathbf{r_j}$) and then add to the sequence of states the corresponding element of *rng*($\mathbf{r_j}$); (iii) continue (perhaps to infinity) until no further additions are possible. This procedure is only roughly described in this case and a bunch of qualifications and distinctions would need to be made.

The useful observation is that there are two sequences involved: the sequence of states from $\boldsymbol{Q}$ and also the sequence of relations from $\boldsymbol{Rel}$. Having sketched labeled transition systems in general, the next task is to focus more narrowly on hybrid systems and in particular on timed transition systems for *n-dimensional hybrid automata of the form* ($\boldsymbol{Q}$, $\boldsymbol{Q}^0$, $\boldsymbol{A}$, $\boldsymbol{Rel}$):

(HA6) The state space of an n-dimensional hybrid automaton is defined as Q, Q0 $\subseteq$ V × Rn, where Rn is an n-dimensional vector space. Thus, the elements of Q and Q0 are ordered pairs of the form (v, x), where v is a control mode (i.e., a vertex) and x is an n-dimensional vector.

(HA7) The *edge label set* of an n-dimensional hybrid automaton is defined as ***A*** = $\Sigma \cup R_{\geq 0}$, that is, the union of all of the edge labels defined in *(HS4)* and the set of all non-negative real numbers. (Recall that the set components of ***A*** are taken to be mutually disjoint.)

(HA8) (i) The ordered pair (***v***, ***x***) is an element of ***Q*** iff the closed predicate *init*(***v***)[*X* = ***x***] is true; and (ii) the ordered pair (***v***, ***x***) is an element of $\boldsymbol{Q}^0$ iff both *init*(***v***)[*X* = ***x***] and *inv*(***v***)[*X* = ***x***] are true.

(HA9) For each event η in Σ and for each $\mathbf{r_k}$ in ***Rel***, $\mathbf{r_k}$((***v***, ***x***), (***v'***, ***x'***)) iff there is a control switch ***e*** in ***E*** such that
(i) the edge ***e*** is from ***v*** to ***v'***,
(ii) the closed predicate *jump*(***e***)[$X, {}^{+}X$ = ***x***, ***x'***] is true; and
(iii) *event*(***e***) = η.

(HA10) ***init* and *inv* Rules:** (i) The ordered pair (***v***, ***x***) is an element of ***Q*** iff the closed predicate *init*(***v***)[*X* = ***x***] is true; and (ii) the ordered pair (***v***, ***x***) is an element of $\boldsymbol{Q}^0$ iff both *init*(***v***)[*X* = ***x***] and *inv*(***v***)[*X* = ***x***] are true.

(HA11) ***jump* (Discrete) Transitions:**
For each event η in Σ and for each $\mathbf{r_k}$ in ***Rel***, $\mathbf{r_k}$((***v***, ***x***), (***v'***, ***x'***)) iff there is a control switch ***e*** in ***E*** such that
(i) the edge ***e*** is from ***v*** to ***v'***,
(ii) the closed predicate *jump*(***e***)[$X, {}^{+}X$ = ***x***, ***x'***] is true; and
(iii) *event*(***e***) = η.

(HA12) ***flow* (Continuous) Transitions:**
For each nonnegative real δ and for each $\mathbf{r}_\delta$ in ***Rel*** , $\mathbf{r}_\delta$((***v***, ***x***), (***v'***, ***x'***)) iff ***v*** = ***v'*** and there is a differentiable function $f : [0, \delta] \longrightarrow R^n$, with the first derivative $\mathbf{D}f : (0, \delta) \longrightarrow R^n$,
such that:
(i) $f(0)$ = ***x*** and $f(\delta)$ = ***x'*** ;
(ii) for all reals ε in (0, δ), both *init*(***v***)[*X* = *f*(ε)] and *flow*(***v***)[*X*, **D***X* = *f*(ε), **D***f*(ε)] are true; and
(iii) the function *f* is called a *witness* for the transition $\mathbf{r}_\delta$((***v***, ***x***), (***v'***, ***x'***)).

IV. A Model for Supervisory Hybrid Automata

The discussion in the previous section has sketched in outline the basic operational structure of hybrid automata. The main effort in this section is going to be devoted to a consideration of the possible application of hybrid automata theory to the supervisory control of continuous systems. This requires reasonably well-defined linkage between a continuous plant on one end and a discrete system on the other. One obvious problem is that the ends of this linkage do not speak the same language! That is, the behavior of the plant is governed by a well-defined family of ordinary differential equations, but the computational end of this configuration, usually taken to be a *discrete event system (DES)* which is in fact an event-driven finite-state automaton. [18]

Therefore, in order to communicate at all, there needs to be a *bidirectional interface* which moves (1) the analog-to-digital signals (*i.e.,* the continuous-time output or plant-state) from the plant to the DES, and also (2) the digital-to-analog signals (*i.e.,* the discrete-time output, or DES-state) from the DES to the plant. Having this distinction available is useful, but there are still fundamental differences between the continuous realm of plant-states on the one hand and DES-states on the other.

Let's first approach this from the perspective of the plant, and specifically the notion of a *plant-event*. As explained in [18], "a plant event is simply and occurrence in the plant… In the case of hybrid control, a plant event is defined by specifying a hypersurface that separates the plant's state space into two disjoint sets. The plant event occurs whenever the plant state trajectory crosses this hypersurface."

Though not explicitly introduced here, the inherent structure of plant space is an n-dimensional vector space through which the plant-event moves and the overarching mathematical framework is that of dynamical system theory. The authors of [18] also make it quite clear that "crossing the hypersurface" is not a simple matter to define in practice: (1) the trajectory could have a very brief excursion across the hypersurface and could even repeatedly re-cross it, and (2) the normal practice would require an unambiguously clear crossing event. The authors propose that in order for a crossing to be recognized, a set of smooth functional should be used to disambiguate the situation ([18], 1029). If all has gone to plan, then the bidirectional interface uses the *generator* subsystem to convert the continuous-time plant-state to an *asynchronous symbolic input* to the DES control system.

From the perspective of the DES, the operation of the bidirectional interface accepts symbolic inputs from the *actuator* subsystem and then produces appropriate symbolic outputs for the generator. It should be noted that behavior of the DES system is a matter of a explicit computational conversion from input symbolic-text (for the actuator) to output symbolic-text (for the generator).

At this point, the notion of a system *controller*, perhaps better called the *supervisor*, can be straightforwardly introduced. Here is the main line taken: there is in the world a physical plant whose operational style depends upon life in a bounded continuum, and also in the world there is a supervisor whose operational style depends upon life in symbolic realm of at most denumerably infinite symbolic structures, and finally there is in the world a supervisor whose operational style is to shuttle symbolic expressions from one part of itself (the actuator part) to another part of itself (the generator part) subject to the condition that this shuttling shall always and everywhere carry *finite and only finite* messages.

V. Discussion

First, the easiest generalization about AM manufacturing is that, if and when anything gets made, it will almost inevitably get made in a closed box. And the second easiest generalization about AM manufacturing is that any one closed box will have virtually no contact with whatever happens in any other closed box.

I do accept the point that the discussion of supervisory HAs, particularly in Section III.C, was a bit of a stretch, but in a larger sense what I am recommending is that AM technology be fully incorporated into the industrial engineering environment.

The commercial viability of AM and the ability of this technology to add value to such product classes as automobiles, household appliances and industrial machine tools is now widely accepted. Functional requirements and detailed specifications need to be developed. Validation procedures will be needed to assure that the functional requirements are satisfied by interlinked networks of HAs.

AM technology can offer more cost-effective and higher-quality solutions compared to traditional devices with similar functionality, or they offer possibilities that cannot be realized at all using traditional systems. Compared to the traditional devices, AM can be produced from small amounts of raw materials, use little energy and generate small volumes of waste products. But it must also be clearly understood that there are substantial technical and engineering barriers to be overcome.

Second, this paper, more perhaps than earlier ones, has brought home to me the need to take great care in the construction of important mathematical notions. I am now focusing on Non-Standard Analysis (NSA) – and I believe that this notion will be an increasingly powerful tool in the engineering tool box. This work was developed by several authors including Skolem and also Laugwitz among others, but the main systematic development of NSA was achieved by Robinson, esp. in his book *Non-Standard Analysis*, which was published in 1966. Here's a very brief summary of this mathematical idea (taken from Alberverio *et al.*[1]): the main result of NSA is that "the geometric line or continuum can support a point set richer than the standard reals. This … gives us a framework for a geometric analysis of physical phenomena on many scales …"

This book was well-received, especially by those mathematicians who had been acquainted with Model Theory. Those less familiar with this logical material were put off by such notions as ultrafilters, ultraproducts, and other seemingly alien notions. In the early days it was commonly believed that anything that could be done within NSA could also be done in analysis without using NSA techniques. However, it soon became obvious that this idea (or hope) was not true. Today NSA is showing signs of robust health, especially in the area of applied analysis and related areas of mathematics which have direct and immediate bearing on the engineering community.

There are other areas of mathematics which should also be mentioned. One very interesting example is the work of Stefan Hilger, who developed the notion of *measure chains* (also called *time scales*) in order to be able to unify the study of differential equations and difference equations into a single theory.

Finally, I am fortunate to have been given the opportunity in my present role at NIST to be able to explore these intriguing areas more thoroughly and more deeply than would otherwise have been possible.

References

[1] S. Alberverio, J.E. Fenstad, R. Hoegh-Krohn, and T. Lindstrom, *Nonstandard Methods in Stochastic Analysis and Mathematical Physics,* Academic Press, 1986.

[2] R. Alur, *et al.* "The Algorithmic Analysis of Hybrid Systems," *Theoretical Computer Science*, vol 138 (1995); 3-34.

[3] R, Alur, *et al.*, "Discrete Abstractions of Hybrid Systems," *Proc. IEEE,* vol 88 (2000); 971-984.

[4] E. Asarin and P. Collins, "Noisy Turing Machines," in L. Caires, *et al., Automata, Languages, and Programming,* 32nd International Colloquium; ICALP, 2005; LNCS 3580; 13 pp.

[5] E. Asarin and and G. Schneider, "Widening the boundary between decidable and undecidable hybrid systems," CONCUR '02, LNCS 2421, Brno, Czech Republic; 193-208.

[6] R. Bott, "Lectures on Morse Theory, Old and New," *Bulletin (new series) AMS,* vol 7 (1982); 331-358.

[7] J.C. Boudreaux, "Simplicial Complexes, Edge Paths, and Digraph Automata," *Proceedings IPSI-2005,* Massachusetts Institute of Technology, 2005 (abstract in *NASA Langley Research Center Scientific and Technical Aerospace Reports,* vol 44 no 7 (April 7, 2006), 95-96.)

[8] J.C. Boudreaux, "A combinatorial framework for parametric additive manufacturing," *Int, J. Rapid Manufacturing,* vol 3 (2013); 105-129.

[9] M.S. Branicky, "Introduction to Hybrid Systems," in D. Hristu-Varakelis and W.S. Levine (eds.), *Handbook of Networked and Embedded Control Systems, Birkhauser,* 2005; 91-115.

[10] M. Braverman and S. Cook, "Computing over the Reals: Foundations for Scientific Computing," *Notices of the AMS,* vol 53 (2006); 318-329.

[11] P. Collins, "Hybrid Trajectory Spaces," *Centrum voor Wiskunde en Informatica,* Amsterdam, The Netherlands; 26 May 2005; 23 pp.

[12] I. Davis, "An Introduction to Nonstandard Analysis," Internet source, publication date August 14, 2009; 25 pp.

[13] M. Fraenzle, "Analysis of Hybrid Systems: An ounce of realism can save an infinity of states," *Computer Science Logic*, *13th International Workshop, CSL '99;* 126-140.

[14] T.A. Henzinger, "The Theory of Hybrid Automata," *Proc. 11th Annual Symp. Logic in Computer Science (LICS),* IEEE Computer Society Press, 1996; 278-292.

[15] T.A. Henzinger, *et al.* "What's Decidable About Hybrid Automata?" *J. Computer and System Sciences*, vol 57 (1998); 94 – 124.

[16] R.D. Kamm and R. Bashir, "Creating Living Cellular Machines," *Annals of Biomedical Engineering*, vol 32 no. 2 (2014); 445-459.

[17] H. J. Keisler, *Foundations of Infinitesimal Calculus*, licensed under Creative Commons Attribution-Noncommercial Share Alike 3.0, June 4, 2011.

[18] X.D. Koutsoukos, P.J. Antsaklis, J.A. Stiver, and M.D. Lemon, "Supervisory Control of Hybrid Systems," *Proceedings of the IEEE*, vol (88) no. 7 (2000); 1026 -1049.

[19] J.G. Hocking and G.S. Young, *Topology*, Dover, 1988.

[20] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in C.E. Shannon and J. McCarthy (eds.), *Automata Studies*, Princeton, 1956; 329-378.

[21] X. Zheng *et al.*, "Ultralight, Ultrastiff Mechanical Metamaterials," *Science*, 20 June 2014, vol(334) 2014; 1373-1377.

Biography. Dr. Boudreaux joined the National Institute of Standards and Technology, formerly the National Bureau of Standards, in 1979, first in the Computer Science Laboratory, and then from 1984 to 1992 in the Manufacturing Engineering Laboratory. In 1992, he was appointed a program manager in the NIST Advanced Technology Program. His research has focused primarily in the following areas: models for higher-order functional systems, adaptive control, real-time error compensation systems, and intelligent manufacturing systems.

Contact information: J.C. Boudreaux, Guest Researcher, NIST/Advanced Network Technologies Division, Bldg 222, Gaithersburg MD 20899; Email: *jackb@nist.gov*.