On Consistent Symbolic Representations of General Dynamic Systems

Mieczyslaw M. Kokar, *Member, IEEE*

***Abstract*—This paper deals with the issue of consistent symbolic (qualitative) representation of continuous dynamic systems. Consistency means here that the results of reasoning with the qualitative representation hold in the underlying (quantitative) dynamic system. In the formalization proposed in this paper, the quantitative structure is represented using the notion of a *general dynamic system* (GDS). The qualitative counterpart (QDS), is represented by a finite-state automaton structure. The two representational substructures are related through functions, called *qualitative abstractions of dynamic systems*. Qualitative abstractions associate inputs, states and outputs of the QDS with partitions of appropriate GDS spaces. The paper shows how to establish such consistent partitions, given a partitioning of the system's output. To represent borders of these partitions, the notion of *critical hypersurfaces* is introduced. One of the main ideas that provides consistency is the interpretation of qualitative input events as elements of the partition of the Cartesian product of input, initial state and time sets. An example of a consistent qualitative/quantitative representation of a simple dynamic system, and of reasoning using such a representation, is provided.**

I. Introduction

THIS paper deals with an integration of symbolic (qualitative) and quantitative representations in the domain of continuous dynamic systems. Examples of goals of reasoning within the qualitative structure are: (1) determine the region (qualitative output) to which the output of the system belongs given a region (qualitative state) to which the current state belongs, (2) determine to which of the regions in the state space the state will belong (next qualitative state) given the region in the Cartesian product of input, initial state and time sets, (3) determine the class of input processes that would cause the system to switch to a particular region in the state space. The question that we are investigating in this paper is how to partition a quantitative dynamic system so that the results of reasoning within the qualitative structure (i.e., answers to the above three questions) always hold in the underlying quantitative dynamic system. We call such a representational structure *consistent*. We also require that the symbolic structure be finite.

The understanding of consistency that we follow in this paper is borrowed from logic (cf. [1]), where consistency refers to a theory that has a model. In other words, a consistent theory is such that all its sentences hold in the model under all possible interpretations. In our formalization of this problem, the quantitative structure is represented in terms of a *general dynamic system* (GDS). The qualitative counterpart, called *Qualitative Dynamic System* (QDS), is represented by a finite-state automaton structure. The two representational substructures are related through *abstraction functions*, called here *qualitative abstractions of dynamic systems*, which map the GDS onto the QDS. Interpretation of the QDS is through the inverse of the abstraction function. The whole quantitative/qualitative structure will be referred to as the Q^2 structure. Within our framework, the GDS is a model for the QDS, and therefore for the representation to be consistent, every sentence of the QDS must hold in the GDS under all possible interpretations.

This same notion of consistency is sometimes referred to as *soundness* (cf. [2]). In qualitative reasoning, on the other hand, the sentences derived through qualitative reasoning that do not hold in the underlying dynamic system are called "spurious behaviors." Therefore, the goal of this paper, in qualitative reasoning terms, is related to eliminating such spurious behaviors.

The typical approach to representing continuous dynamic systems with finite qualitative data structures is to partition the system state, input, and output variables into subsets (intervals), assign symbolic labels to the dividing points and intervals, and then reason about the system using these labels. Such partitioning of the variables can be interpreted as partitioning the Cartesian products of the variables into subsets. Each of the subsets is supposed to represent a class of points that can be termed "qualitatively equivalent." Such an approach can be classified as *abstraction*. If conclusions obtained apply to whole subsets represented by particular labels, then the representation and the reasoning procedure is *consistent*, otherwise it is not. Two popular means (especially within the AI community) used for subdividing the system space are *fuzzy sets* and *landmark points*. These two approaches have one thing in common: the basic shape of a partition is an interval in one-dimensional space, a rectangle in two-dimensional space, a rectangular parallelepiped (a *box*) in three dimensions, and a *hyperbox* in a higher-dimension space. When dividing the space by landmark points, the borders between the boxes are crisp, while in the fuzzy-set approach these borders are fuzzy. But the main shape in both cases is that of a box.

Unfortunately, reasoning with boxes is not consistent since such a representation of a dynamic system cannot guarantee that its results will hold in the underlying quantitative system. This means that if the symbolic reasoner derives a conclusion

Manuscript received December 28, 1992; revised February 13, 1994, and October 7, 1994. This work was funded in part by NSF Grant IRI-8915057, and was presented at the 1992 IEEE Symposium on Intelligent Control, Glasgow, Scotland.

The author is with the Department of Industrial Engineering and Information Systems, Northeastern University, Boston, MA 02116 USA.

IEEE Log Number 9411909.

that the next state will be within a given box, it may or may not be true. For instance, most AI-based qualitative reasoning programs have this feature [3].

In this paper, we show an example of how states, inputs and outputs of a finite state automaton can be assigned to subsets in the spaces of the quantitative dynamic system so that the representation is consistent, and therefore the results of the qualitative reasoning hold in the quantitative system. We could not find formal or experimental evidence that such a goal can be achieved by representing regions of qualitatively equivalent points with hyperboxes in the Cartesian product of the original system variables. Hyperboxes, however, can serve as approximations of the real shapes of qualitatively distinct subsets. Our example starts with an explicit interval partition of the output variable, inducing a partition of the state space through the inverse of the output function, which in turn induces a partition of the input-process plus initial-state space through the inverse of the state transition function.

The border ofs the partitions in our approach are called *critical hypersurfaces*, while functions that map a quantitative system onto a qualitative representation are called *qualitative abstraction functions*. Critical hypersurfaces and abstraction functions are equivalent. We shall demonstrate the establishment of *critical hypersurfaces* which partition the GDS space, given a partition of the system's output. Critical hypersurfaces subdivide the system space into subsets whose shapes, typically, are not boxes.

While the basic aim of the paper is to show that such an approach is feasible, we do not claim that finding such partitions is an easy task. In the example of this paper we show how to find critical hypersurfaces from the known quantitative model of the dynamic system. Not always is such a model available. But knowing that they exist, we can search for such hypersurfaces using other methods. One such method can be machine learning.

In many practical situations it might be impossible to find such partitions, and therefore, it would be necessary to use less accurate methods, like those proposed in the boxes approach. Nevertheless, it is important to understand that the "boxes approach" is not optimal, and that whenever we use it, we are making a tradeoff between the consistency of reasoning and the lack of precise knowledge. This paper shows directions for the rationalization of such a tradeoff. It is worth mentioning that such precise hypersurfaces might not even be the ultimate solution. When the system we are modeling is subject to noise, a mechanism for dealing with uncertainty due to noise must be incorporated into our modeling or control system. This mechanism may be fuzzy sets. But the fuzziness would be *around the hypersurfaces*, instead of *around the walls of boxes*. This can be accomplished by treating these hypersurfaces as either random or fuzzy variables.

The main contribution of this paper is the idea of consistent partitioning of three system spaces, i.e., the output space, the state space, and the input event space (Cartesian product of the input set, (initial) state space, and time). This partitioning allows us to construct qualitative reasoning systems whose inferences are guaranteed to hold in the underlying (deterministic) continuous dynamic system.

In the following section, we will present a short overview of some of the approaches to combined qualitative/quantitative modeling of dynamic systems. Our goal is to give some hints on the approaches to qualitative/quantitative reasoning known in the literature, rather than to provide a complete history of the field. In subsequent sections we will introduce the necessary definitions of: "consistency of qualitative reasoning," "General Dynamic System" (GDS), "Qualitative Dynamic System" (QDS), "qualitative abstraction function," and "consistency postulates." We will then demonstrate some of the consequences of accepting such postulates. This is followed by the introduction of the notion of "critical hypersurface." We also give an example of a simple dynamic system and describe all the critical hypersurfaces that partition the space of this system. In the section that follows we will present an example of integrated qualitative/quantitative reasoning about this system. This reasoning uses the Q^2 representation of this dynamic system. We conclude the paper with an overview of research questions that should be investigated in relation to the Q^2 representation.

II. Related Research

Modeling and simulation of dynamic systems has always been an important issue in both science and engineering. General methodologies of modeling and simulation of dynamic systems have been studied extensively in systems science and general systems theory (cf. [4], [5]), control (cf. [6]), economics [7]–[9], and artificial intelligence (cf. [10]). Differences between methods used in all of these disciplines are, to a great extent, due to different treatments of the issues of granularity and consistency of reasoning.

The systems science approach is based on the mathematical formulation of the general dynamic system. Most of the effort in systems science has focused upon building quantitative models and automatic quantitative simulation. The main thrust of the systems science research has been to provide "fine granularity" models that would give accurate descriptions of physical systems, thereby guaranteeing consistency. In the systems science approach, a model of a dynamic system is represented by a *state transition function* and an *output function*. Future behaviors are generated through quantitative simulation which "executes" a simulation model (state transition and output functions), typically at fixed time steps, to obtain quantitative values of state and/or output variables. This type of system model is useful for predicting (simulating) the *next* state, given the *current* state, since the state transition function is given explicitly. It is quite inefficient, however, for drawing global conclusions, e.g., which of the states and which of the inputs will take the system into a particular state.

In qualitative physics (QP), simulation of behavior of dynamic systems has been one of the major concerns. The main thrusts of the qualitative physics approach to simulation were the compatibility with the reasoning techniques used by humans and automatic generation of qualitative representations. Qualitative simulation uses qualitative models in which collections of quantitatively different inputs, states, and outputs are lumped into qualitative inputs, states, and outputs, respectively. Different qualitative points are described by qualitative variables. In earlier work (cf. [11]), these variables would take on values from a three-element set, $\{-1, 0, 1\}$, or equivalently $\{(-\infty, 0), (0), (0, \infty)$. In QSIM [12], the variables were assigned values of either landmark points or intervals, and therefore the space partitions could be interpreted as boxes. These methods, as mentioned earlier, do not guarantee consistency. In the qualitative reasoning terminology, they can generate spurious behaviors.

Systems theory and control have also dealt with qualitative properties of systems and qualitative models, although the qualitative properties considered in systems science were different. The system scientists' main interest was in the system being *stable/unstable, observable/nonobservable*, and *controllable/noncontrollable*. This line of interest was also followed by some AI researchers. Yip [13], Sacks [14] and Zhao [15] investigated the issue of partitioning the state space into regions defined by the asymptotic properties of the trajectories of the system behavior. This kind of partitioning consists of open regions delineated by "bounding trajectories." Although in the terminology of this paper, these bounding trajectories are examples of hypersurfaces, they differ in a number of ways. The main difference is that in the cited work, only the partitioning of the state space has been addressed. The input process was not included, and therefore the issue of consistency of partitioning the different system spaces was nonexistent. The input process is especially important for considering not only free evolution of the dynamic system in time, but for analyzing the system behavior when an input control process is applied. Also, [13] and [14] dealt with time-varying aspects of dynamic systems (bifurcations), which are outside of the scope of this paper.

Qualitative analysis of dynamic systems has also been of interest to economists who studied qualitative properties (positive and negative influences on the variables) of economic models [7]–[9]. These approaches were also interval-based, and thus did not guarantee consistency.

There is a number of research efforts which, in order to avoid violating the consistency of reasoning, employ an approach based on dynamically varying granularity over time (refining the partitions by using smaller boxes) (cf. [16]). Although this direction is closely aligned with the goals of this paper, it differs significantly because it does not guarantee a stable symbolic representation since it gradually refines the accuracies of the partitions. Our aim is to search for finite partitions that are stable over time and thus do not need to be refined.

One way of avoiding problems with consistency is to use representations that have imprecision as an inherent property; in this kind of representation, probabilistic or fuzzy-set based methods are used. Therefore, the problem posed in this paper cannot be formulated as a direct translation. For instance, Saridis [17] has proposed to view different levels of abstraction as having different levels of "intelligence" and "precision." The three layers of the Saridis' design reflect "increasing intelligence with decreasing precision." Antsaklis *et al.* [18] proposed to extend this model to an arbitrary number of layers.

In the control literature, landmark points are considered as generators of *discrete events*. The systems that deal with discrete events are called *discrete event dynamic systems* (DEDS). One of the formalizations of such systems is due to Zeigler [19]. This formalization has been investigated by many researchers (cf. [20]–[25]). In this approach, a continuous-time continuous-state system is represented by a formal system, called DEVS, whose distinguishing features are that it has two state transition functions (internal and external), and has a *time advance* function which represents a (continuous) time interval during which the system remains in the same state partition if an external event does not occur. In this formalization, the consistency of the representation is ensured through using quantitative representation of the time. The DEVS representation, although very close in the goals, differs from the Q^2 representation presented in this paper in several respects. One of our goals was to use a standard automaton representation to capture the qualitative aspects of a dynamic system. The DEVS representation is not an automaton; it is a more complex mathematical structure. It includes continuous time, and therefore it is not a purely qualitative representation. And finally, it does not represent explicitly what we call *qualitative input events*, i.e., equivalence classes in the Cartesian product of inputs, initial system states, and time.

This paper is also closely related to the research on *hybrid control systems* (cf. [26]–[29]). A hybrid system consists of [29] a continuous dynamical system (plant), a discrete controller (automaton), and two interfaces: generator (translates plant states into discrete events), and actuator (converts controller symbols into continuous plant inputs). This paper deals with a similar framework, except for the actuator part. The operation of the generator is based upon the principle of *boundaries*, which are the critical hypersurfaces in the state space, according to our terminology. These boundaries are detected by sensors. Typically, these are threshold sensors, and therefore the state space is partitioned into boxes. The emphasis in the control approach is, however, on the controllability of such continuous plants rather than on the predictability of their behaviors. In the control paradigm, the imprecision of the plant model can be compensated by the feedback control loop, and therefore the issue of consistency (or predictability of the plant's behavior under specific inputs) has not been addressed.

The notion of *critical hypersurface* was first introduced into qualitative reasoning by this author in [30]. Chiu [31] used the name "domain map" for the subdivision of the parameter space into qualitative regions. He used algebraic reasoning to determine domain maps. Hypersurfaces were also recognized by Dangelmaier as important in robotics control, where the robot manipulator's workspace was partitioned into qualitative regions delimited by "critical curves and surfaces" [32]. In all of these approaches, the notion of a critical hypersurface was not fully formalized. It was introduced using examples from particular domains. In this paper we define this notion for a more general class of systems: general dynamic systems (GDS) [4]. None of the previous approaches made a clear connection to the theory of dynamic systems.

In the systems science/control literature the notion of the hypersurfaces has been addressed (cf. [19]–[25]). However, as we mentioned earlier, it was limited to the system state spaces only, and consequently, consistent partitioning of the system spaces was not an issue.

III. General Dynamic Systems

For the sake of self containment, we present the general systems theory's definition of a "general dynamic system" [4]–[6], [33]. This notion is rigorously defined as a mathematical object, and therefore, any general claim related to such an object can be precisely understood and verified.[1]

Definition 1: A *general dynamic system*, $\mathcal{S}$, is an 8-tuple

$$\mathcal{S} = (T, X, W, Q, P, F, g, \leq),$$

where

1) T is a *time set* with an order relation $\leq$ on it; we can write it as: $\leq \subset T \times T$.
2) X and W describe the *input set* and the *output set*, respectively.
3) Q describes the *inner states* q of the dynamic system.
4) P (input processes) consists of time functions $p: T \to X$ together with an inner *splicing* (or *concatenation*) on them. The splicing operation, $\perp_t$, for any two functions p_1 and p_2 is defined as

$$p = p_1 \perp_t p_2 = \begin{cases} p_1(\tau), & \tau < t \\ p_2(\tau), & \tau \geq t. \end{cases}$$

   P is closed under splicing.
5) F is a *global state transition function* $F: T \times T \times Q \times P \to Q$. For time-invariant systems $F: T \times Q \times P \to Q$ and $t_0 = 0$.
   F has the following properties:
   a) (consistency) $F(t_0, t_0, q, p) = q$;
   b) (semigroup) $F(t, t_0, q, p) = F(t, t_1, f(t_1, t_0, q, p), p)$, if $t_0 < t_1 \leq t$;
   c) (causality) $F(t, t_0, q, p) = F(t, t_0, q, p_1)$, if $p(\tau) = p_1(\tau)$, for $t_0 < \tau \leq t$.
6) g is an output function $g: T \times Q \to W$. For time-invariant systems $g: Q \to W$.

In a general (multivariable) case $X = X_1 \times \cdots \times X_n$, and $Q = Q_1 \times \cdots \times Q_s$; output set W is typically considered to be one-dimensional. We will refer to the input, state, and output sets as *input space, state space*, and *output space*, respectively. When referring to an unspecified space, or to a Cartesian product of such spaces, we will use the term a *system space*.

The definition of a GDS is constructed in such a way as to reflect some general characteristics of all dynamic systems. The splicing property of the input processes gives the flexibility of choosing any control strategy for the given system at any point in time. One typical control scenario occurs when the control variable is a step function. The state space Q of a GDS is the most important element in the definition. Each state $q = (q_1, \ldots, q_s)$ represents a history of the system as far as it is necessary to compute the current output of the

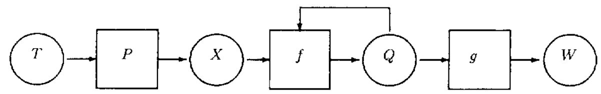

Fig. 1. The structure of a general dynamic system.

system given the current input. The state transition function $q = F(t, t_0, q_0, p)$ computes for a state q_0 at time t_0 (also called "initial state") the state q reached at time t when an input function p is applied at time t_0. The consistency property of this function ensures that state transitions are not instantaneous. The semigroup property concerns both the state and the state transition function. This property means that the system state at any particular time, t, can be computed either directly for the whole time interval $[t_0, t)$, or indirectly, by performing two computations: for $[t_0, t_1]$ and $[t_1, t]$. More generally, it allows us to decompose the computation of a state into finitely many steps. The causality property means that the value of the new state depends only on the input function p restricted to the interval $[t_0, t)$.

The state transition function F in Definition 1 is called a "global transition function." For discrete systems, where time is represented by the set of integers, it makes sense to talk about *current* and *next time*. In such a case, it is possible to replace the global transition function F with a *local transition function* f, which computes the state at the *next* time instance given the *current* state and the value of the input x at the current time.[2] Since it is possible to prove (using mathematical induction, cf. [6]), that a local transition function uniquely determines a global transition function, we can use freely either one of the two functions to define a GDS. Consequently, a local transition function f is a mapping

$$f: T \times Q \times X \to Q$$

where X are inputs applied at the current time, which, for a time-invariant system, reduces to

$$f: Q \times X \to Q.$$

For continuous-time systems, the existence of such a local transition function can be guaranteed only if the system is sufficiently "smooth." In such a case the transition function (of a time-invariant system) is expressed using differential equations,

$$\dot{Q}(t) = f(Q(t), x(t)).$$

In the rest of this paper we will use local transition functions.

The anatomy of a general dynamic system is represented in Fig. 1. The circles in this figure represent sets: T—time set, X—input set, Q—state set, and W—output set respectively. The rectangles represent functions that appear in the definition of a general dynamic system: P—input process functions, f—local state transition function, and g—output function. The state transition function has two incoming arrows: from X, the input at current time, and from Q, the state at current time.[3]

[1] Note that the use of the notion of "state" in the AI qualitative simulation literature is only loosely related to the notion of "state" in the definition of general dynamic system. For instance, input variables are often treated as state variables.

[2] In discrete systems the values of state, input and output variables are presumed to change only at the time instances and nothing changes in between.

[3] The third arrow—from T—is not shown, since the intent is to represent a time-invariant system.

IV. QUALITATIVE DYNAMIC SYSTEMS AND QUALITATIVE ABSTRACTIONS

An *abstraction*, as defined in [34], is a pair of formal systems and an effective total mapping which relates the languages of the two systems. One of the systems is called the *ground* system and the other the *abstract* system. Two of the features of abstractions are "preservation of certain desirable properties" and "throwing away details" (i.e., making the abstract system simpler and easier to handle). In this paper, we propose to use abstraction to represent an infinite-state quantitative dynamic system with a finite state automaton. This represents the feature of "throwing away details." The representation is to be provably correlated with the underlying dynamic system, meaning that the conclusions about the state transitions and outputs of the system derived within the qualitative representation must hold in the quantitative dynamic system. This represents the feature of "preserving certain desirable properties" (preserving the provability from QDS to GDS).

We define "qualitative dynamic system" (QDS) as a finite state automaton (FSA) which is related to the underlying general dynamic system. To establish a relationship between a GDS and a QDS, we introduce the notion of "qualitative abstraction function."

Definition 2: Let S be a set, and I be a finite set. A function $\chi\colon S \to I$ will be called a *qualitative abstraction function* if it is total and many-to-one.

A typical example of such an abstraction function is

$$\text{sign}\colon W \to \{-1, 0, 1\},$$

whose value is determined by the sign of the output W of the system (it is -1 for negative values of W, 1 for positive, and 0 for $W = 0$). In qualitative reasoning in AI, W is taken to be either an output variable or the derivative of a variable.

Definition 3: A *Qualitative Dynamic System (QDS)* is a finite state automaton (FSA)

$$\Sigma = (\Lambda, \Omega, \Theta, \phi, \gamma),$$

where

1) Λ, Ω, Θ are finite sets and ϕ, γ are functions:
   a) Λ—qualitative input event set,
   b) Θ—qualitative state set,
   c) Ω—qualitative output set,
   d) ϕ—qualitative state transition function
   $$\phi\colon \Theta \times \Lambda \to \Theta,$$
   e) γ—qualitative output function
   $$\gamma\colon \Theta \to \Omega.$$
2) Λ, Θ, and Ω are related to the underlying general dynamic system, GDS, through the *qualitative abstraction function*
   $$\chi = (\chi_{TQX}, \chi_Q, \chi_W),$$
   consisting of three qualitative abstraction functions: $\chi_{TQX}\colon T \times Q \times X \to \Lambda$, $\chi_Q\colon Q \to \Theta$, and $\chi_W\colon W \to \Omega$, where χ_{TQX}, χ_Q, and χ_W are called qualitative input event, qualitative state and qualitative output abstraction functions, respectively.

As we can see, the resulting structure is an abstraction of a ground system, GDS, into QDS, using the qualitative abstraction χ as the abstraction mapping.

The AI literature dealing with qualitative reasoning uses the terms "abstraction function" and "qualitative dynamic system" quite differently than we have, above. The main difference is that in the literature on qualitative reasoning, these direct qualitative counterparts of quantitative concepts are sought: qualitative inputs, qualitative states, qualitative outputs, and qualitative time. We propose, instead, to interpret qualitative inputs (events) in the Cartesian product of quantitative states, inputs and time.[4] This removes the main obstacle to constructing representations in which qualitative reasoning is consistent. Intuitively, we cannot abstract just qualitative inputs (meaning input values from an interval of possible values), because the qualitative behavior, i.e., both the trajectory of the system's state and the output, depend, in addition to the value of the input, on the state the system is in. Moreover, this behavior depends also on the time interval at which a given input is applied. In order to be able to predict the system's behavior, we need to have all these three pieces of information: the current state, the input, and the time interval at which the input is applied.

V. CONSISTENCY OF REASONING

As mentioned before, consistent reasoning derives only conclusions that are satisfied in the dynamic system under consideration. Our objective is to construct integrated quantitative/qualitative representations of dynamic systems in which consistency of reasoning can be achieved. In our case, we are concerned with the consistency of reasoning using abstractions. In particular, we are interested in reasoning about the GDS using the QDS representation.

Definition 4: A qualitative reasoning procedure is *consistent* if all the theorems $TH(QDS)$ of the qualitative dynamic system QDS are satisfied in the underlying general dynamic system GDS under all possible interpretations, i.e.,

$$GDS \models TH(QDS),$$

where interpretations are selected through the inverse of the abstraction function χ.

One of the simplest tasks for a qualitative/quantitative reasoner may be: given the current (quantitative) state of a GDS, find to which of the subsets of the GDS's output set W the output of the system will belong (see Fig. 2). The reasoning is as follows. First, the abstraction function χ_Q is applied to q_i, returning qualitative state θ_i. Then the qualitative output function γ is applied to this qualitative state. This returns the value of ω_i. The class of the qualitative output can be established through the inverse of χ_W applied to ω_i. The solution does not give an exact value of w since it involves qualitative reasoning; it returns a set of values, $W_i = \chi_W^{-1}(\omega_i)$. However, if this is the set to which the value

[4] For notational convenience, we will denote $T \times Q \times X$ as TQX-set (or space).

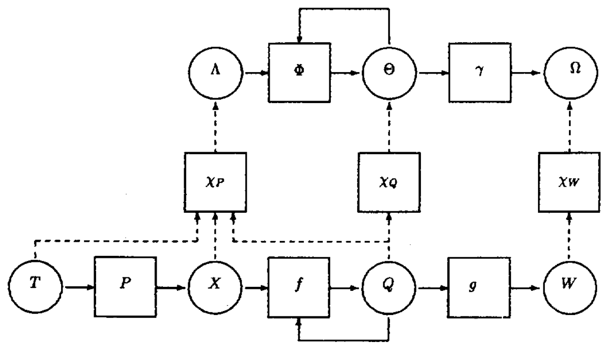

Fig. 2. Q^2: Quantitative/qualitative representation of a dynamic system.

of the output function $g(q_i)$ belongs, then the result of this reasoning is correct. If this reasoning procedure returned a different interval, say $W_j = \chi_W^{-1}(\omega_j)$ instead, it would be incorrect.

The above definition of consistency applies to formal systems in general. In logical terms, it expresses the requirement that the mapping χ be an *T^*—abstraction* [34]. To apply this definition to a specific formal system, one needs to define the language in which theorems are expressed, the inference rules, and the abstraction χ. With respect to general dynamic systems and qualitative dynamic systems as their abstractions, we propose the following constraints on the abstractions, which we call *consistency postulates.*

Definition 5: Let $\mathcal{S}$, Σ and χ represent a GDS, a QDS and an abstraction function defined as above. The pair (Σ, χ) is a consistent representational structure of a dynamic system $\mathcal{S}$ if the following consistency postulates are fulfilled $\forall q, q_0, x, t$:

1) $\gamma(\chi_Q(q)) = \chi_W(g(q))$
2) $\phi(\chi_Q(q_0), \chi_{TQX}(t, q_0, x)) = \chi_Q(f(t, q_0, x))$.

In algebraic terms, these consistency postulates state that the quantitative structure (GDS) and the qualitative structure (QDS) must be related through the homomorphism [35] defined by the abstraction χ. If we limit our theorems to those that are simple applications of the output and state transition functions (in the qualitative system) then, since χ is a homomorphism and is onto, we can be assured that the consistency is preserved. We can summarize it as a theorem (we skip the proof, since this is an instantiation of a more general proof in abstract algebra).

Theorem 1: If (Σ, χ) is a consistent representational structure of a dynamic system $\mathcal{S}$ and the theorems of the QDS are limited to statements of the form: $\theta' = \phi(\theta, \lambda)$, and $\omega = \gamma(\theta)$, with $\theta, \theta' \in \Theta$, $\omega \in \Omega$, then the reasoning using the qualitative/quantitative structure $Q^2 = (\mathcal{S}, \Sigma, \chi)$ is consistent.

This concludes our step-by-step outline of a *qualitative/quantitative* (Q^2) *representation of a dynamic system.* The resulting structure is represented in Fig. 2.

VI. Qualitative Interpretations and Admissible Partitions

Inverse images of the ranges of the qualitative abstraction functions χ_{TQX}, χ_Q and χ_W constitute partitions of their domains. These abstraction functions establish a homomorphism between two structures, a GDS and a QDS. In algebraic terms, the partitions established by the qualitative abstraction functions are called "admissible partitions," which means that the equivalence classes (elements of the partitions) are congruent [35] with respect to the two functions, f and g.

Definition 6: A partition of a space of a GDS, $S = S_1; \ldots; S_n$, defined by a qualitative dynamic system abstraction function χ, which fulfills the consistency postulates is called an *admissible partition.* An element of an admissible partition is called a *qualitative subset* of a GDS.

The function χ_W defines *qualitative output subsets,* χ_Q defines *qualitative state subsets,* and χ_{TQX} defines *qualitative input event subsets.* Note, however, that the qualitative input event subset is not merely a partition of the input set (space); it is rather a partition of the Cartesian product of the input space, state space, and time.

Admissible partitions and qualitative abstractions are closely related. This fact is useful for both constructing qualitative representations and checking for consistency of proposed qualitative representations. We can either first define partitions and then find abstraction functions, or first define abstractions and then find the partitions. Also, we have a choice of checking either the consistency postulates directly, or the congruency of a particular partition.

VII. Relations Among Qualitative Abstraction Functions

We now focus our attention on how to generate qualitative partitions[5] or, equivalently, qualitative abstractions of system spaces. The main question is whether we can partition one of the system's spaces into qualitative subsets independently of the rest of these spaces. It is natural to suspect that consistency postulates introduced in the previous section should restrict our flexibility in choosing partitions (or, equivalently, abstraction functions) for particular system spaces. Indeed, we will prove a theorem, which states that (1) a partitioning of the output space gives a unique partitioning of the state space Q, and (2) the partitioning of the state space uniquely defines the partitioning of $T \times Q \times X$.

Theorem 2: Let $W_\pi = W_1; \ldots; W_n$ be a finite partition of a GDS's output space W, given by the inverse of the abstraction function $\chi_W^{-1}\colon \Omega \to W_\pi$. Let Q_π describe a partition of Q defined as an inverse image of W_π through g, $Q_\pi = g^{-1}(W_\pi)$, and let TQX_π describe a partition of $T \times Q \times X$ defined as an inverse image of Q_π through f, $TQX_\pi = f^{-1}(Q_\pi)$.

a) Q_π is a maximal admissible partition of Q
b) TQX_π is a maximal admissible partition of $T \times Q \times X$.

Proof: To prove that Q_π is an admissible partition in a), we need to show that there exist mappings $\chi_Q\colon Q \to \Theta$ and $\gamma\colon \Theta \to \Omega$, such that the consistency postulates are fulfilled, i.e., $\gamma(\chi_Q(q)) = \chi_W(g(q))$, for every $q \in Q$. The mapping χ_Q would have to be defined in such a way that it would assign a θ_i to each class $Q_i = g^{-1}(\chi_W^{-1}(\omega_i))$, and γ would assign ω_i to θ_i. Note that as a result of such a construction process, for any two different $q_1, q_2 \in Q_i$, we have $\chi_W(g(q_1)) = \chi_W(g(q_2))$.

[5]From now on, we will use the term "partition" instead of "admissible partition."

Likewise, for any two different $q_1 \in Q_1$, $q_2 \in Q_2$, where $Q_1 \neq Q_2$, we have $\chi_W(g(q_1)) \neq \chi_W(g(q_2))$.

To prove that Q_π is the maximal partition of Q, we first show that any partition of Q which does not agree with Q_π is not an admissible partition. Then we show an example of a partition that is a subpartition of Q_π but is not maximal.

Say Q'_π is another partition of Q that is not a subpartition of Q_π. This means that there exist $q_1, q_2 \in Q$ such that they belong to the same partition class Q'_1 of Q'_π, while they belong to two different classes, say Q_1 and Q_2, $Q_1 \neq Q_2$, in Q_π. As for any two elements of two different classes we would have $\chi_W(g(q_1)) \neq \chi_W(g(q_2))$ (as stated above). The abstraction χ'_Q would assign to Q'_1 a value (only one, since it is a function), say θ'_1. The function γ' would assign to θ'_1 a value, say ω'_1. Consequently, $\gamma'(\chi'_Q(q_1)) = \gamma'(\chi'_Q(q_2))$, which contradicts the fact stated above that these values should be different. The conclusion is that if Q'_π does not agree with Q_π it also implies that the consistency postulates are violated.

Note that it is possible to define a partition Q'_π of Q in such a way that each class in the partition is a subclass of one of the classes in Q_π. Then, a function γ' can be defined in such a way that it assigns the same value ω to all classes of Q_{ij} that are contained in class Q_i. In such a case, the mapping $\gamma'(\chi'_Q(Q))$ agrees with the mapping $\chi_W(g(Q))$, and therefore the consistency postulates are not violated. However, the partition Q'_π, and consequently the function γ', are more complex and redundant. This redundancy does not appear in the Q_π partition, and therefore this partition seems to be most desirable.

The b) part of the proof can be carried out in a similar way; we skip it for the sake of brevity of this presentation. □

A direct consequence of this theorem is the fact that if we are given a qualitative partition of the output space of a dynamic system, the partitioning of the rest of the system's spaces can be derived using the construction process outlined in this paper.

VIII. Critical Hypersurfaces

In the theorem of the previous section we did not make any assumptions about the functions f, g and about the topology of the initial partition of the output space W. Without such presumptions we were not able to make any statements about the form of the derived partitions. In practical applications, however, the assumption of continuity of functions is quite typical, and so is the assumption of partitioning a variable into intervals. For instance, we might limit our considerations to such systems where:

1) Functions f, g are continuous, except for a finite number of points.
2) The dynamic system is a single-output system (the output space consists of one variable).
3) The partition of this variable is given by a sequence of points (we will call them *distinguished points*[6]). This results in partitioning the output variable W into a sequence of open intervals and points $W = (-\infty, w_1) \cup w_1 \cup (w_1, w_2) \cup \cdots \cup (w_{k-1}, w_k) \cup (w_k, \infty) \cup [\text{Inf}]$.

[6]As for instance in [12].

The symbol Inf is introduced to represent an extra point that can be added to the real line in order to make it *compact* [36]. Inf represents both ∞ and $-\infty$. We consider Inf to be a distinguished point, similarly to w_i's.

The types of constraints on the system spaces and of the consequences of such constraints (like the above three assumptions) on the partitioning of the spaces of a GDS is a subject of topology (cf. [36]). From the applications point of view, other kinds of constraints might be both interesting and useful in particular domains (classes of dynamic systems). To analyze this, we would need to focus our attention on topological properties of particular dynamic systems. In this paper we just make note of some topological facts related to this subject and ask some of the relevant questions. For instance, qualitative classes of the state space Q, obtained through the application of g^{-1} to open intervals in W, are regions in Q which are *open* and *connected* [36]. This is because inverse images of open and connected sets through a continuous function are also open and connected [36]. Similarly, the application of f^{-1} to these regions results in a collection of open and connected regions in $T \times Q \times X$. The distinguished points are transformed into closed regions in Q and $T \times Q \times X$; we call them *critical hypersurfaces*. Depending on the type of the functions f and g, there might be only a few regions or an infinite number. Interesting questions include the following: What are the classes of dynamic systems for which these hypersurfaces are connected? When would we have only a finite number of them? And many others. Another possibility is to consider some "nice" functions (Kuipers [12]) limited his considerations to so called "reasonable" functions).

Definition 7: A *critical hypersurface* in Q is an image of a distinguished point in W through g^{-1}. A critical hypersurface in $T \times Q \times X$ is an image of a critical hypersurface in Q through f^{-1}.

All of the system's spaces can be subdivided into regions by distinguished points and critical hypersurfaces. The behavior of the system can be represented as a trajectory in each of the spaces: input process plus initial state trajectory, state space trajectory, and output space trajectory. Each of these trajectories may cross a critical hypersurface. If these critical hypersurfaces are given by distinguished points on the system's output variable, then the event of crossing a hypersurface in one of the system's spaces coincides with the crossing of the respective critical hypersurfaces in all system's spaces. Qualitative behaviors can be characterized by the trajectories, i.e., by series of qualitative inputs (events) in Λ, qualitative states in Θ, or qualitative outputs in Ω.

Points on critical hypersurfaces play a role similar to that of distinguished points in the output space. In AI literature they are known as "landmark points" [12]. Therefore critical hypersurfaces are sets of landmark points. We can view critical hypersurfaces as *distributed landmarks*.

IX. Example

To clarify the ideas presented in this paper, we consider an example of a very simple dynamic system, a ball thrown upward, which has been very extensively used throughout

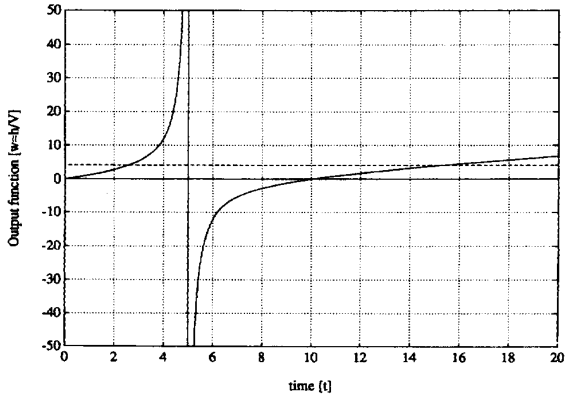

Fig. 3. The ball example: The output function, $g(t)$ for $h_0 = 0$, and $V_0 = 50$ m/s. The dashed line represents the value of the distinguished point $w^* = 4$.

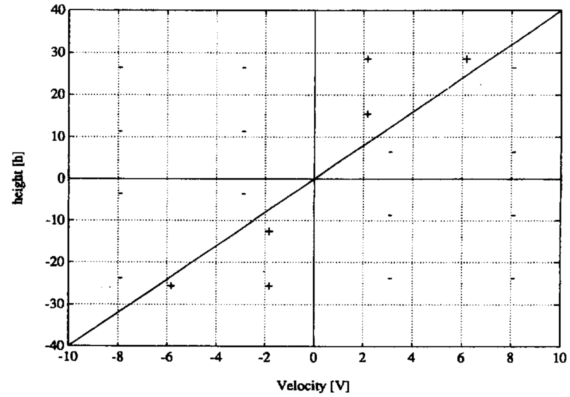

Fig. 4. The ball example: A critical hypersurface in the state space.

qualitative physics literature. Such a dynamic system is described by the differential equation

$$\ddot{h}(t) - a = 0.$$

The initial conditions are: $h(0) = h_0$ and $V(0) = V_0$, where h represents the height of the ball, t—time, V—velocity, V_0—initial velocity, h_0—initial height, a—acceleration (due to gravity).

In the terminology of the GDS, the state space Q is represented by pairs of state variables (h, V); the initial states are represented by (h_0, V_0). The input process is

$$p(t) = a = \text{const.}$$

Suppose also that the output of the system is given by

$$w = g(t) = \frac{h(t)}{V(t)}.$$

The values of this function, for $h_0 = 0$ and $V_0 = 50$ m/s are shown in Fig. 3. The dashed line in this figure represents the value of the distinguished point $w^* = 4$.

The state transition function f is given by two equations, which can be easily obtained from the differential equation

$$\begin{cases} h(t) = h_0 + V_0 t + \frac{1}{2} a t^2, \\ V(t) = V_0 + at. \end{cases}$$

Suppose that the qualitative abstraction function for the output space, χ_W, is given by the distinguished points w^* and Inf

$$\chi_W(w) = \begin{cases} \omega_{-1}, & -\infty < w < w^*, \\ \omega_0, & w = w^*, \\ \omega_1, & w^* < w < \infty, \\ \omega_\infty, & w = \text{Inf}. \end{cases}$$

The output space W is subdivided through χ_W^{-1} into

$$W = \{(-\infty, w^*),\ [w^*], (w^*, \infty), [\text{Inf}]\}.$$

The qualitative output space Ω consists of four elements

$$\Omega = \{\omega_{-1}, \omega_0, \omega_1, \omega_\infty\}.$$

The subdivision of the state space can be obtained by finding the critical hypersurfaces given by the image of w^* and Inf through g^{-1}

$$\pi_Q(h, V) = C,$$

where C is a constant parameter. The hypersurface (corresponding to w^*) is a set of pairs

$$\{(h^*, V^*)\} = \{(h, V) : g(h, V) = w^*\}.$$

In this two-dimensional case we have two critical hypersurfaces (see Fig. 4): a straight line corresponding to the distinguished point w^*, and a straight line corresponding to Inf, which is equal to the axis $V = 0$

$$\begin{cases} \frac{h}{V} = w^*, \\ V = 0. \end{cases}$$

These two hypersurfaces partition the state space Q into two groups of subsets. They are marked in Fig. 4 as: '$-$'—where the system's output w is less than w^*, and '$+$'—where the system's output is more than w^*. The system's output is equal to w^* when the state is on the corresponding hypersurface, and is either $+\infty$ or $-\infty$, when it is on the hypersurface corresponding to the distinguished point of Inf. The qualitative state space consists of four states

$$\Theta = \{\theta_{-1}, \theta_0, \theta_1, \theta_\infty\}.$$

The qualitative abstraction function of the state space is given by

$$\chi_Q(h, V) = \begin{cases} \theta_0, & h/V = w^*, \\ \theta_\infty, & h/V = \text{Inf}, \\ \theta_{-1}, & h/V < w^* \wedge V > 0, \\ \theta_{-1}, & h/V > w^* \wedge V < 0, \\ \theta_1, & h/V > w^* \wedge V > 0, \\ \theta_1, & h/V < w^* \wedge V < 0. \end{cases}$$

The hypersurfaces

$$\pi_{TQX}(t, a, V_0, h_0) = C$$

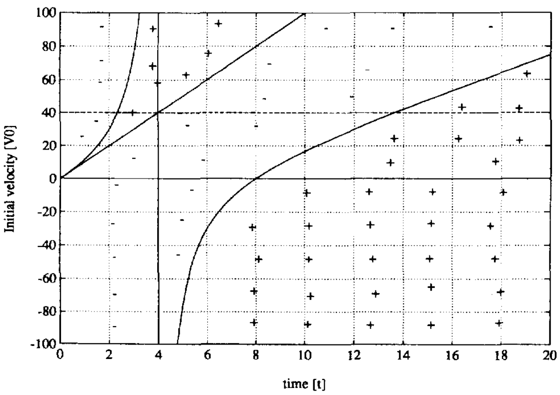

Fig. 5. The ball example: Critical hypersurfaces in the TQX space.

in TQX are obtained as the image of the critical hypersurfaces π_Q through f^{-1}

$$\begin{aligned}\{(t^*, a^*, V_0^*, h_0^*)\} &= f^{-1}(h^*, V^*)\\ &= \{(t, a, V_0, h_0) : f(t, a, V_0, h_0)\\ &= (h^*, V^*)\}.\end{aligned}$$

The formula describing the hypersurface corresponding to w^* can be obtained by substituting the formulae for V and h (the f function) into the expression describing π_Q

$$\frac{h_0 + V_0 t + 1/2 \cdot at^2}{V_0 + at} = w^*.$$

This formula can be simplified and transformed into a more explicit representation

$$V_0 = \frac{at(1/2 \cdot t - w^*) + h_0}{w^* - t}.$$

An example of such a hypersurface for $w^* = 4,\ h_0 = 0$, and $a = -10$ is presented in Fig. 5. As a function of time, it is zero at the origin, then it goes to infinity as t approaches 4, where it is discontinuous, reaches $-\infty$ to the right of $t = 4$, crosses $V_0 = 0$ at $t = 8$, and then increases ever after.

The hypersurface corresponding to Inf is described as

$$V_0 + at = 0,$$

which for this example, i.e., for $h_0 = 0$ and $a = -10$, represents a straight line, as shown in Fig. 5.

TQX is subdivided by the above hypersurfaces into four regions representing four qualitatively-different input objects. The qualitative input event space Λ consists of four qualitative values

$$\Lambda = \{\lambda_{-1}, \lambda_0, \lambda_1, \lambda_\infty\}.$$

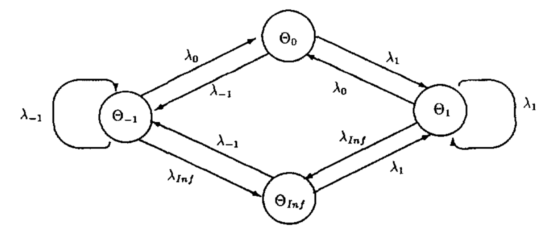

Fig. 6. The ball example: Qualitative state transitions.

The qualitative input event abstraction function χ_{TQX} is given by

$$\chi_{TQX}(t, a, V_0, h_0) = \begin{cases} \lambda_0, & V_0 = \frac{at(1/2 \cdot t - w^*) + h_0}{w^* - t} \\ \lambda_\infty, & V_0 = -at \\ \lambda_{-1}, & V_0 > \frac{at(1/2 \cdot t - w^*) + h_0}{w^* - t} \wedge t < w^* \\ \lambda_{-1}, & V_0 < -at \wedge t < w^* \\ \lambda_{-1}, & V_0 > \frac{at(1/2 \cdot t - w^*) + h_0}{w^* - t} \wedge V_0 < -at \wedge t > w^* \\ \lambda_1, & V_0 < \frac{at(1/2 \cdot t - w^*) + h_0}{w^* - t} \wedge V_0 > -at \wedge t < w^* \\ \lambda_1, & V_0 > -at \wedge t > w^* \\ \lambda_1, & V_0 < \frac{at(1/2 \cdot t - w^*) + h_0}{w^* - t} \wedge t > w^*. \end{cases}$$

The meaning of the qualitative inputs (events) is such that λ_{-1} causes the qualitative state switch to (stay in) θ_{-1}, λ_0 switches state to θ_0, λ_1 to θ_1, and λ_∞ to θ_∞. This fact is represented in Fig. 6. Since we are interested in local transition functions, the graph represents only local transitions.

Fig. 6 is a pictorial representation of the qualitative state transition function Φ. The qualitative output function γ is a simple assignment

$$\gamma(\theta) = \begin{cases} \omega_{-1}, & \theta = \theta_{-1} \\ \omega_1, & \theta = \theta_1 \\ \omega_0, & \theta = \theta_0 \\ \omega_\infty, & \theta = \theta_\infty. \end{cases}$$

We close this section with two theorems about consistency of the above representation and about its maximality. Although we do not present a full proof of these theorems, it seems obvious how to develop such a proof. The process of constructing the hypersurfaces presented in the previous section gives enough information for the construction of such a proof.

Theorem 3: Let $\mathcal{B}$ be a general dynamic system as described above

$$\mathcal{B} = (T, X, W, Q, P, f, g, \leq),$$

where T is time, X and W describe the input set and the output set, respectively, $Q = \{q = (h, V)\}$—the inner states of the system, P—input processes $(p(t) = a = const)$, f—the (quantitative) state transition function, and g—the output function. Let $\Sigma = (\Lambda, \Omega, \Theta, \phi, \gamma)$ represent a qualitative dynamic system, and $\chi = (\chi_{TQX}, \chi_Q, \chi_W)$ represent an abstraction function as defined in this section.

The pair (Σ, χ) is a consistent representational structure of the dynamic system $\mathcal{B}$, i.e., the following consistency postulates are fulfilled $\forall t, a, h, V$

$$\gamma(\chi_Q(h,V)) = \chi_W(g(h,V))$$

$$\phi(\chi_Q(h_0,V_0), \chi_{TQX}(t,h_0,V_0,a)) = \chi_Q(f(t,h_0,V_0,a)).$$

Theorem 4: Let $W_\pi = W_{-1}; W_0; W_\infty; W_1$, where $W_{-1} = (-\infty, w^*)$, $W_0 = [w^*]$, $W_\infty = [Inf]$, $W_1 = (w^*, \infty)$, be a finite partition of the output space W, given by the inverse of the abstraction function χ_W^{-1}, as described in this section. Let Q_π describe a partition of $Q = h \times V$ defined by the inverse images of W_0 and W_∞ through g (critical hypersurfaces π_Q described in this section+), and TQX_π describe a partition of $h \times V \times G \times T$ determined by the critical hypersurfaces π_{TQX} described in this section.

Q_π is a maximal admissible partition of $Q = h \times V$, and TQX_π is a maximal admissible partition of $h \times V \times G \times T$.

X. Example: Reasoning

Although reasoning with the proposed representational structure is not the main topic of this paper, we will present a simple example of a reasoning procedure which utilizes both the qualitative and quantitative parts. We have implemented a simulation of a reasoning algorithm which uses the Q^2 representation, both qualitative and quantitative parts. We used Matlab software for this purpose. We tested the algorithm on the ball example described in the previous section. The reasoning goal was to infer qualitative state transitions and exact times at which the transitions take place, given the initial quantitative state of the dynamic system and the (quantitative) input process. The space of possible input processes was limited to step changes in acceleration a.

The simulation algorithm is presented in Table I. The main part of this algorithm is the "while" loop. This loop performs the three main functions: determining current quantitative and qualitative states using the quantitative state transition function f and qualitative state abstraction function χ_Q, determining the exact time of the next qualitative transition using the qualitative input abstraction function χ_{TQX}, and shifting the initial state of the dynamic system to the new state. Note that although this algorithm uses the quantitative model, it is different from a typical quantitative simulation algorithm, since it does not simulate the system in equal time increments, but instead, it "jumps" from one qualitative region to another.

A trace of the simulation is presented in Table II. The input for this simulation was: maximum simulation time $t_{\text{sim}} = 20$, output landmark $w^* = 4$, initial state $q_0 = (0, 40)$, i.e., initial height $h_0 = 0$ and initial velocity $V_0 = 40$, and input process $X = (0, -10)$, i.e., the simulation starts at time $t = 0$ and the input's value (acceleration) is $g = -10$.

As can be seen from the table, the simulation gives "precise" times at which qualitative transitions take place.[7] In accordance with the claim that the proposed Q^2 representation is consistent, the system does not generate any spurious transitions. However, as we mentioned earlier in the paper, this reasoning is relatively simplistic. The sole purpose behind implementing this procedure was to gain some initial understanding of the difficulty of reasoning using the proposed Q^2 representation.

[7] The "precision" is limited by computer arithmetic errors.

TABLE I
Example of a Q^2 Simulation Algorithm

```
Denotations:
  t_step     :time of next step-change in input
  t_0        :initial time
  t_cur      :current time
  t_trans    :time of next qualitative transition
  t_sim      :maximum simulation time
  time       :time in the input process
  input      :input value in the input process
  q, q_0     :current and initial state
  w*         :output landmark
  p          :input process.
begin{program}
input(t_sim, q_0, w*, p)
  i <- 1                                  :first step input
  t_cur <- p(i, time)                     :current time
  t_0 <- t_cur                            :initial time
while t_cur < t_sim do
  q <- f(t_cur - t_0, p(i, time), q_0)    :current quant. state
  θ <- χ_Q(q)                             :current qualit. state
  t_trans <- min_t(χ_TQX(t, p(i, input), q_0)
  ≠ χ_TQX(t_0, p(i, input), q_0))         :next transition
  t_0 <- t_cur                            :push forward
  if t_cur + t_trans < p(i + 1, time)
    t_cur <- t_cur + t_trans
  else
    t_cur <- p(i + 1, time)               :next change in input
    i <- i + 1
  end{if}
  report transition
  q_0 <- q
end{while}
end{program}.
```

TABLE II
Example: A Simulation Trace

Quantitative state		Qualitative state	Time of transition
h	V		
0.000	40.000	θ_{-1}	2.343
66.274	16.569	θ_0	2.343
66.439	16.469	θ_1	4.000
80.000	0.000	θ_∞	4.000
79.999	-0.100	θ_{-1}	13.657
-386.274	-96.569	θ_0	13.657
-387.240	-96.669	θ_1	

XI. Closing Remarks

The ultimate goal of the research presented in this paper is to develop methods useful in the design and implementation of intelligent hierarchical systems [17, 18, 37, 38] and intelligent hybrid control systems [26–29]. We investigated the concept of an integrated (Q^2) representational structure in which (1) the qualitative (symbolic) structure is separated from the quantitative structure, and (2) consistent qualitative/quantitative reasoning can be carried out. To achieve consistency we proposed to interpret qualitative variables as regions in particular system spaces. We proposed to bound the regions by "critical hypersurfaces." Especially important was the idea of interpreting qualitative input events as regions in the Cartesian product of input, initial state and time. This approach is in contrast with a more traditional qualitative reasoning based upon intervals and hypercubes, which does not guarantee consistency.

The main point is that qualitative reasoning about deterministic systems does not have to be "naive," "incorrect," "non-sound," "inconsistent" or "fuzzy" and that, as was shown in this paper, one can prove theorems about the consistency. On the other hand, if uncertainty is inherent in the problem, the critical hypersurfaces proposed in this paper can serve as contours of either constant uncertainty or constant set membership. The understanding of the "consistent" qualitative reasoning model will help in making conscious and rational tradeoffs between imprecision and computational complexity.

The purely qualitative simulation approach has been criticized from both the systems science point of view [39] and AI [40]. Sacks and Doyle argue in [40] that experts in their reasoning do not consider impossible (spurious) solutions; they eliminate such solutions through the use of mathematics. In this paper we follow a similar line of reasoning, as presented in [39] and [40]. Whenever experts cannot find an answer through qualitative reasoning, they resort to the quantitative model, and whenever they arrive at a conclusion in one of the models, qualitative or quantitative, they are able to translate the conclusion into the other model. Purely qualitative reasoning uses qualitative models that do not have explicit links to their quantitative counterparts, and thus it cannot mimic an expert's integration of qualitative and quantitative reasoning.

The Q^2 representational structure proposed in this paper differs also from the control-based approaches. As was mentioned earlier, the DEVS representation of Zeigler [19] is not a purely qualitative structure since it contains continuous time. In the hybrid systems approach (cf. [29]), the issue of precise predictability of the system behavior, given specific inputs, is not directly addressed, since the imprecision of this prediction is supposed to be compensated by the feedback control loop.

To efficiently implement and utilize the framework presented in this paper, several issues need to be investigated. First of all, we need to investigate to which classes of dynamic systems (as presented in the systems science literature, e.g., [4]) such an approach is applicable. Second, we need to analyze various reasoning scenarios and develop procedures for qualitative/quantitative reasoning that can be carried out using this representation.

In the example presented in this paper, we used the known mathematical model (a differential equation) to derive such a representation. Unfortunately, we do not always have such a complete knowledge of the process. In our future research we will investigate methods that can be used for (semi-) automatic generation of such representational structures through machine learning.

One of our goals is to implement a framework in which various dynamic systems can be represented and reasoning procedures can be tested. The goal of these investigations is to provide experimental evidence of the appropriateness of the proposed approach and its computational efficiency. Finally, the approach presented in this paper needs to be tested on real physical systems.

ACKNOWLEDGMENT

The author is very grateful for the comments and suggestions received from Leonard Monk, Jerzy Tomasik, Kyriakos Zavoleas, Spyros Reveliotis, Zbigniew Korona and Stephen Linder. Thanks also to the three anonymous reviewers for their insightful comments, and for suggesting many editorial changes.

REFERENCES

[1] C. C. Chang and H. J. Keisler, *Model Theory* Amsterdam: North Holland, 1992.
[2] M. R. Genesereth and N. J. Nilsson, *Logical Foundations of Artificial Intelligence*. Los Altos, CA: Morgan Kaufman, 1987.
[3] P. Struss, "Problems of interval-based qualitative reasoning," in *Readings on Qualitative Reasoning about Physical Systems*, D. S. Weld and J. de Kleer, Eds. Los Altos, CA: Morgan Kaufman, pp 288–305, 1990.
[4] M. D. Mesarovic and Y. Takahara, *Abstract Systems Theory*. Berlin: Springer Verlag, 1989.
[5] G. Klir, *Architecture of Systems Problem Solving*. New York: Plenum Press, 1985.
[6] L. Padulo and M. A. Arbib, *Systems Theory: A Unified State Space Approach to Continuous and Discrete Systems*. Philadelphia, PA: W. B. Saunders, 1974.
[7] W. M. Gorman, "More scope for qualitative economics," *Review of Economic Studies*, vol. 31, pp. 65–68, 1964.
[8] K. Lancaster, "The scope of qualitative economics," *Review of Economic Studies*, vol. 29, pp. 99–123, 1962.
[9] J. Maybee and J. Quirk, "Qualitative problems in matrix theory," *SIAM Review*, vol. 11, pp. 30–51, 1969.
[10] D. S. Weld and J. de Kleer, *Readings on Qualitative Reasoning About Physical Systems*. San Mateo, CA: Morgan Kaufman, 1990.
[11] J. de Kleer and J. S. Brown, "Qualitative physics based on confluences," *Artificial Intelligence*, vol. 24, pp. 7–83, 1984.
[12] B. J. Kuipers, "Qualitative simulation," *Artificial Intelligence*, vol. 29, pp. 289–338, 1986.
[13] K. Yip, "Extracting qualitative dynamics from numerical experiments," in *Proc. AAAI-87*, pp. 665–670.
[14] E. P. Sacks, "Automatic analysis of one-parameter planar ordinary equations by intelligent numeric simulation," *Artificial Intelligence*, vol. 48, pp. 27–56, 1991.
[15] F. Zhao, "Extracting and representing qualitative behaviors of complex dynamic systems in phase spaces," *Tech. Rep. AIM-1274*, MIT, Cambridge, MA, 1991.
[16] D. Berleant and B. Kuipers, "Bridging the gap from qualitative to numerical simulation," *Tech. Rep. AI 91-158*, University of Texas, Austin, 1991.
[17] G. N. Saridis, "Intelligent robotic control," *IEEE Trans. Automat. Contr.*, vol. AC-28, no. 5, pp. 547–557, 1983.
[18] P. J. Antsaklis, K. M. Passino and S. J. Wang, "Towards intelligent autonomous control systems: Architecture and fundamental issues," *Intelligent and Robotic Systems*, vol. 1, no. 4, pp. 315–342, 1989.
[19] B. P. Zeigler, *Multi-Facetted Modelling and Discrete Event Simulation*. San Diego, CA: Academic Press, 1984.
[20] F. Cellier, "Time windows: Automated abstraction of continuous-time models into discrete-event models in high autonomy systems," *Int. J. General Systems*, vol. 19, pp. 241–262, 1991.
[21] P. A. Fishwick, "The role of process abstraction in simulation," *IEEE Trans. Syst. Man Cyber.*, vol. 18, no. 1, pp. 18–39, 1988.
[22] P. A. Fishwick, "An integrated approach to system modeling using a synthesis of artificial intelligence, software engineering and simulation

methodologies," *ACM Trans. Modeling and Simulation*, vol. 2, no. 4, pp. 307–329, 1992.

[23] P. A. Fishwick and B. P. Zeigler, "A multimodel methodology for qualitative model engineering," *ACM Trans. Modeling and Simulation*, vol. 2, no. 1, pp. 53–81, 1992.

[24] H. Praehofer, "System theoretic formalisms for combined discrete-continuous system simulation," *Int. J. General Systems*, vol. 19, no. 3, pp. 219–240, 1991.

[25] C.-J. Luh and B. P. Zeigler, "Abstracting event-based control models for high autonomy systems," *IEEE Trans. Syst. Man Cyber.*, vol. 23, no. 1, pp. 42–54, 1993.

[26] A. Benveniste and P. Le Guernic, "Hybrid dynamical systems and the signal language," *IEEE Trans. Automat. Contr.*, vol. 35, no. 5, pp. 535–546, 1990.

[27] A. Gollu and P. Varaiya, "Hybrid dynamical systems," in *Proc. 28th Conference on Decision and Control*, 1989, pp. 2708–2712.

[28] W. Kohn and A. Nerode, "Multiple agent autonomous hybrid control systems," in *Proc. of the 31st Conference on Decision and Control*, 1992, pp. 2956–2966.

[29] M. D. Lemmon, J. A. Stiver and P. J. Antsaklis, "Learning to coordinate control policies of hybrid systems," in *Proceedings of the American Control Conference*, 1993, pp. 31–35.

[30] M. M. Kokar, "Critical hypersurfaces and the quality space," in *Proc. AAAI-87, Sixth National Conference on Artificial Intelligence*, AAAI, 1987, pp. 616–620.

[31] C. Chiu, "Abstracting qualitative behaviors from quantitative simulation models," *Tech. Rep.*, University of Texas, Austin, 1987.

[32] H. T. Dangelmaier, "A qualitative representation for manipulator kinematics and other vector and scalar fields," *Master's Thesis*, The University of British Columbia, 1989.

[33] F. Pichler, "Dynamic systems: A survey," in *Systems and Control Encyclopedia: Theory, Technology and Applications*, M. Singh, Ed. New York: Pergamon Press, pp. 1282–1286, 1987.

[34] F. Giunchiglia and T. Walsh, "A theory of abstraction," *Artificial Intelligence*, vol. 57, 1992.

[35] L. S. Levy, *Discrete Structures of Computer Science*. New York: John Wiley, 1980.

[36] J. Dugunji, *Topology*. Boston, MA: Allyn and Bacon, 1966.

[37] J. S. Albus, "Outline for a theory of intelligence," *IEEE Trans. Syst. Man Cyber.*, vol. 21, pp. 473–509, 1991.

[38] A. Meystel, "Nested hierarchical control," in *Introduction to Intelligent and Autonomous Control*, K. M. Passino and P. J. Antsaklis, Eds. Norwell, MA: Kluwer, 1992.

[39] P. A. Fishwick and B. P. Zeigler, "Qualitative physics: Towards the automation of systems problem solving," in *AI, Simulation and Planning in High Autonomy Systems*, pp. 118–134, 1990.

[40] E. P. Sacks and J. Doyle, "Prolegomena to any future qualitative physics," *Computational Intelligence*, 1991.

Mieczyslaw M. Kokar (M'87) received the M.S. and Ph.D. degrees in computer systems engineering from the Technical University of Wroclaw, Poland, in 1969 and 1973, respectively.

He then worked at the Technical University of Wroclaw from 1969 to 1981, and Millersville University, Millersville, PA, from 1982 to 1984. Since 1984, he has been with the Department of Industrial Engineering and Information Systems, Northeastern University, Boston, MA. His research interests include machine learning of physical laws, learning control, qualitative dynamics and hybrid control systems, sensor/data fusion, and software engineering.

Dr. Kokar served as a committee member of the IEEE International Symposium on Intelligent Control from 1989 to 1994, and was the Publications Chair of that conference in 1991. He was also on the committee of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, and Intelligent Robotic Systems (IRS). He is a member of the AAAI and ACM.