A Finite State Machine Model to Support the Visualization of Complex Dynamic Systems

Brian J. d'Auriol
Ohio Supercomputer Center
Columbus, Ohio, 43212, USA
Email: dauriol@acm.org

Abstract**—** *Linear and non-linear controllable systems are commonly found in many engineering problems and are examples of models that involve complex high-dimensional spatially-related data sets. Such systems range in size and complexity from small-scale systems of a few state variables to large-scale systems comprising many state variables that evolve in complex ways. Particular evolution trajectories and regionalized evolutionary behavior of such complex systems are often non-intuitive and may reflect bifurcations in the system. A state space discretization approach leading to a novel structure termed Orthogonal Organized Finite State Machines (OOFSM) is proposed as a modeling technique to support visualizations of the properties of these types of complex systems.*

Linear and non-linear controllable systems are commonly found in many engineering problems [1]. Such systems range in size and complexity from small-scale systems of a few state variables to large-scale systems comprising many state variables that evolve in complex ways. Particular evolution trajectories and regionalized evolutionary behavior of such complex systems are often non-intuitive and may reflect bifurcations in the system. For large-scale systems in particular, the state space exists in high dimensional spaces and many of the transitions are expected to be intra-dimensional. Understanding the operation and behavior of the system together with the details about specific evolution trajectories and how such relates with other trajectories or with the system as a whole may provide insight into the underlying dynamics of the system as it evolves towards desired or undesired behavior. In addition, this may also facilitate decision making by policy makers who are involved in control operations.

An *orthogonal organized finite state machine (OOFSM)* is proposed in this paper as a discrete state space abstraction of the evolutionary behavior of dynamic systems. A lattice partitioning applied to the state space discretizes the state space. A discrete vector field that abstracts the intersection of trajectories with the boundaries of the discretized state space provides for the abstraction of the evolutionary behavior. This paper formalizes the OOFSM as a finite state machine abstraction of the discretized state space with the discrete vector field abstraction.

The rest of this paper is organized as follows. The discretization approach and the OOFSM are described in Section I. Applications of this abstraction are presented in Section II. Conclusions are given in Section III.

I. Dynamic System Discretization

Finite state machines (FSMs) as models for dynamic systems have previously been considered in the literature. There is a lot of literature concerning FSM modeling of Discrete Event Systems (DES), see for example [2]; also, hybrid systems. In [3], an FSM models discrete event systems so as to take into account the issue of partial observability. The work presented here however is based on discretizing a continuous system. A more general treatment of finite state modeling of continuous systems is described in [4]. The approach taken in this paper is to derive a spatially organized finite state machine. In the literature, the term "Orthogonal finite-State Machine (OFSM)" [5] is introduced in 2002. The model presented here is similar, but derived specifically from an analysis of complex dynamic systems and therefore is intended to wholly represent and support the modeling of these systems, including, the visualization of properties of these systems. The term "organized finite state machine" also appears

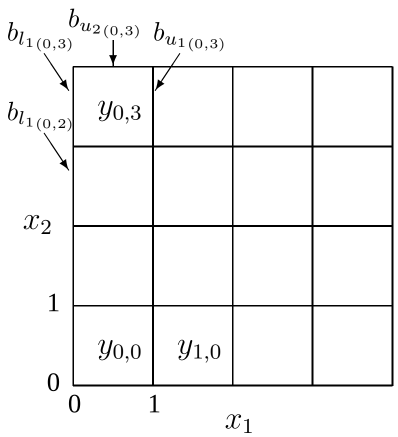

Fig. 1. Illustration of state space definitions, $n = 2$, $o = 16$ and uniform unit partition $\mathcal{L}$.

in the literature [6], however, this usage tends to refer to hierarchically organized structures.

It is assumed that a state space is readily available and given by a set of state variables: $X = \{x_1, x_2, \ldots, x_n\}, x_i \in \mathbb{R}$, hence, X is continuous in $\mathbb{R}^n$; n is also called the dimension of the system. A lattice partitioning $\mathcal{L}$ applied to the state space leads to a set of discretized states $\mathcal{L} : X \to Y$ where $Y = \{y_0, y_1, \ldots, y_{o-1}\}$ for some finite o such that y_j is bounded by a convex polytope (hypercube) in $\mathbb{R}^n$. At the most coarse partitioning, $\mathcal{L}$ defines a single discretized state $|Y| = 1$. In general, $\mathcal{L}$ defines a set of partition boundaries $P = \{p_{i_j} | 1 \leq i \leq n, 0 \leq j \leq o-1\}$ with $p_{i_j} \in \mathbb{R}^{n-1} = (b_l, b_u)_{i_j}, b_u > b_l$. Each p_{i_j} is aligned normal with the corresponding ith state variable; $\iota(b)$ denotes this value. These definitions are illustrated in Figure 1. In the figure, $X = \{x_1, x_2\}$, $n = 2$, $\mathcal{L}$ is uniform unit and, as shown, $o = 16$. Each discrete state in the figure has the two pairs of boundaries p_1 and p_2; more specifically, for state $y_{(0,3)}$, the pair $(b_{l_{1(0,3)}}, b_{u_{1(0,3)}})$ is normal to x_1 and the pair $(b_{l_{2(0,3)}}, b_{u_{2(0,3)}})$ is normal to x_2. And, $\iota(b_{l_{2(0,0)}}) = 0$. For convenience, elements of Y may be interchangeably expressed in terms of the dimension of the system.

The following simple procedure to accomplish the mapping $\Gamma : X \to Y$ is proposed. First, let us recall the property of convex polytopes that all points on the locus of a line segment defined by two internal points are also internal. For every $x_i \in X$ determine $y_{(\ldots,b_{l_i},\ldots)} \mid \iota(b_{l_i}) \leq x_i < \iota(b_{u_i})$. For example, $\Gamma(0.2, 3.7)$ in reference to Figure 1, then

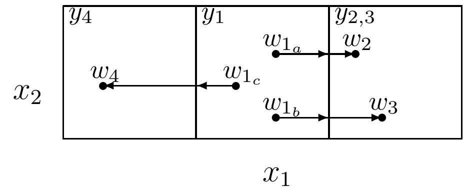

Fig. 2. Example: possible set of evolution paths, note the direction vectors at the boundaries.

for $i = 1,\ 0 \leq 0.2 < 1$ and for $i = 2,\ 3 \leq 3.7 < 4$, therefore, $y_{(0,3)}$ is identified. More importantly, all x_i bounded by p_i is associated with discrete state $y_{(b_{l_1}, b_{l_2}, \ldots, b_{l_n})}$, that is, this procedure identifies states based on the lower bound values of the partition boundaries of the state variables.

An evolution of the dynamic system traces a path through the continuous state space. For any two selected points on this path, say w_1 and w_2 such that w_2 occurs after w_1 in time, let the direction vector $v_1 = (v_{1_1}, v_{1_2}, \ldots, v_{1_n})$, $v_{1_i} \in \{-1, 0, 1\}$ denote the direction of the evolution from w_1 to w_2 tangent at w_1 in terms of the basis vectors. For example, let $w_1 = (0.2, 3.7)$ and $w_2 = (1.2, 2.7)$ in reference to Figure 1, then, $v_1 = (1, -1)$ and $v_2 = (-1, 1)$.

Given a boundary partition p_{i_j} and two points w_1, w_2 that define the path $\overrightarrow{w_1, w_2}$, then the *discrete direction vector* is $v_{w_1} = (\ldots, a_i, \ldots)$ where

$$a_i = \begin{cases} 0 & \text{if neither } b_{l_i} \text{ nor } b_{u_i} \text{ intersects } \overline{w_1, w_2}; \\ 1 & \text{if } b_{u_i} \text{ intersects } \overline{w_1, w_2}; \\ -1 & \text{if } b_{l_i} \text{ intersects } \overline{w_1, w_2}; \end{cases}$$

In general, multiple trajectories describe the evolution behavior of a system. Given a set of discrete direction vectors $\{v_{j_k} \mid k \geq 1\}$ in state y_j, then the *discrete direction vector field* $\overline{v}_j = (\ldots, a_i, \ldots)$ where $a_i = \bigcup_k v_{j_{k_i}}$.

The example shown in Figure 2 clarifies the discrete direction vector field. Let six points w_{1_a}, w_{1_b}, w_{1_c}, w_2, w_3 and w_4 define three paths $e_1 = \overline{w_{1_a}, w_2}$, $e_2 = \overline{w_{1_b}, w_3}$ and $e_3 = \overline{w_{1_c}, w_4}$ such that all w_1's lie very close to each other and are in y_1, w_2 and w_3 are in $y_{2,3}$ and w_4 is in y_4. Then, $v_{e_1} = v_{e_2} = (1, 0)$ but $v_{e_3} = (-1, 0)$ (meaning the direction vectors at the w_1's along the indicated path). Here, y_1 could evolve either to $y_{2,3}$ or y_4, and, $\overline{v}_{y_1} = (\{1, -1\}, \{0\})$ indicates that the state may evolve either in the positive or negative x_1 direction from y_1, but not evolve in the x_2 direction.

The discrete direction vector field is derived from an input set of multiple trajectories. It is assumed in this paper that a sufficiently large set of trajectories is available so as to generate as complete a direction vector field as possible.

Two other useful abstractions are the *region field* and the *uniform region field*. A region field is defined as the set of all component discrete direction vector fields in a defined region: $\overline{V}_R = \{\overline{v}_j \mid j \in R\}$ for region R; $\overline{V}_Y$ denotes some general region field. A uniform region field is a region field whose components are identical: $\overline{v}_i \doteq \overline{v}_j$ for $\overline{v}_i, \overline{v}_j \in \overline{V}_R$ and $\doteq$ denotes lexicographic equality.

The previous discussions establish a structured orthogonal grid where each grid cell has cell transition attributes. This structure is termed an *Orthogonal Organized Finite State Machine (OOFSM)* and is the primary abstraction proposed in this paper for the modeling of dynamic systems.

There are many finite state machine (FSM) models in the literature. The NFA as presented in [7] is selected: $N = (S, \Sigma, T, s_o, F)$ where S is the set of states, Σ is the set of input symbols, T is the transition function over state-symbol pairs, s_o is the starting state and F is the set of terminal states.

Definition 1: The *Orthogonal Organized Finite State Machine* is defined by $M = (Y, \mathcal{L}, \overline{V}_Y)$.

Lemma 1: The Orthogonal Organized Finite State Machine is an NFA.

For the proof of this lemma, let $S = Y$, determine Σ based on $\mathcal{L}$, show that $\overline{V}_Y$ is equivalent to T, determine the language accepted by M and lastly define the starting state and terminal states. Recall, by definitions of X and P, $\mathcal{L}$ defines the boundaries of each state in terms of specific values given by ι. Let these values define the inputs to the NFA, that is, the NFA is input-driven. Figure 3 illustrates the correspondence of S and Y using the typical circle and edge notation from the finite state machine literature superimposed on M.

Let $\Sigma = \{0, \pm i \mid 1 \leq i \leq n\}$ so that each of the possible $2n$ egress transitions given by $\mathcal{L}$ can be labeled.

Given $\overline{v} = (\ldots, a_i, \ldots)$, by definition, $a \in \{\{-1\}, \{0\}, \{1\}, \{-1,0\}, \{-1,1\}, \{0,1\}, \{-1,0,1\}\}$. Equivalently, drawn from Σ, let the string s label the transitions as follows.

$$s = \begin{cases} 0 & \text{if } a_i = \{0\} \\ i & \text{if } a_i = \{1\} \\ -i & \text{if } a_i = \{-1\} \\ 0 \vee i & \text{if } a_i = \{0,1\} \\ -i \vee i & \text{if } a_i = \{-1,1\} \\ -i \vee 0 & \text{if } a_i = \{-1,0\} \\ -i \vee 0 \vee i & \text{if } a_i = \{-1,0,1\} \end{cases}$$

Let a discrete path $p = (y_k, y_{k+1}, y_{k+2}, \ldots)$ with associated $\overline{v}_k, \overline{v}_{k+1}, \ldots$, then the string labeling corresponding to p is $s_p = s_k || s_{k+1} || \ldots$ where the $||$ denotes string catenation. For example, in reference to Figure 1, if the discrete path is $(y_{0,0}, y_{1,0}, y_{1,1}, y_{1,1}, y_{1,0})$ and the following discrete direction vector fields are defined $\overline{v}_{y_{0,0}} = (\{-1,1\}, \{-1,1\}), \overline{v}_{y_{1,0}} = (\{0\}, \{-1,1\}), \overline{v}_{y_{1,1}} = (\{-1,0,1\}, \{-1\})$ then $s = 120\bar{2}$ where the overbar notation as in $\bar{2}$ refers to a negated symbol, -2 in this case.

Essentially, $\overline{V}_Y$, which restricts the set of all possible strings in some region, defines the language accepted by the NFA: $L = \{s \leftarrow \overline{V}_Y\}$, that is, the language is the set of all strings over the alphabet Σ drawn from all sequences of the directional vector fields within some region R.

Finally, F is defined by any state that has a corresponding null $\overline{v}$ (all the entries are $\{0\}$) and s_0 is given by the initial conditions of the dynamic system as the set of boundary values.

The following completes the discussion about the equivalence of $\overline{V}_Y$ and T (for brevity the complete formulization is not given). Given some language over Σ, an initial state s_0 and any valid string s in the language, the ith element of the string may be one of the possible $2n+1$ symbols deriving a corresponding element of some v. The collection of the ith element from multiple strings derive $\overline{v}$ and the collection of all possible strings derives $\overline{V}_Y$. ■

The language accepted by M is determined without consideration of either history or prediction. It is akin to saying that any given state has a set of possible transitions out of that state. However, a trajectory provides a history of states that are transited. Given the construction of M above, the language accepted by M *given* a sequence of one

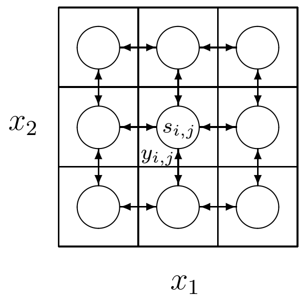

Fig. 3. Illustration of Lemma 1.

or more trajectories is a subset of the language accepted by M.

In general, either a theoretical control model or, if the system is observable, a set of observations about the state variables may determine the OOFSM. For example, as before, let $n = 2$ and uniform unit partition $\mathcal{L}$, let the observable initial conditions of some dynamic system be $[u_1, u_2]^T = [0.5, 1, 5]^T$ which defines $s_0 = y_{0,1}$ by formally, $\iota(b_{l_{1(0,1)}}) \leq u_1 \leq \iota(b_{u_{1(0,1)}})$ and $\iota(b_{l_{2(0,1)}}) \leq u_2 \leq \iota(b_{u_{2(0,1)}})$.

A list of some useful properties of M is extracted from the previous discussion and identified here to provide a succinct description of the OOFSM.

Property 1: Structure: There is a defined structure including: dimension, lattice partitioning, and extrinsic values mapped to each axis.

Property 2: Vector fields: The following partial and complete versions are defined: the discrete direction vector tangent at any point, the discrete direction vector field at any state, the region field at any region, the uniform region field at any region.

Property 3: Path: the labeled and unlabled discrete path of a system's evolution.

Property 4: Behavior: States and regions are indicative of the system's operational characteristics, including: start states, terminal states, and intermediate states in which the path (Property 3) may exhibit further operational characteristics such as oscillatory, resonance, bifurcation and inter-dimensional transition frequency (how often the path changes direction).

Property 5: Indexing: The dimension (Property 1) allows individual states, regions and subspaces (Property 6) to be identified and accessed.

Property 6: Subspace: The partial indexing (Property 5) defines a subspace that contains information pertinent to a subset of the system states.

Property 7: Observation: Observable events can bijectively be represented respectively inferred.

Property 8: Metrics: Various metrics as per the domain and user interests may be defined. Some possible metrics are: cyclic/acyclic discrete trajectory paths, discrete path lengths, number of intra-dimension transitions, number of inter-dimension transitions (before the next intra-dimension transition), and distance functions.

Property 9: Prediction: The behavior (Properties 4 and 2) together with a path or partially known path (Property 3) provide for the prediction or partial prediction of future/past operational characteristics (Property 4) and evolution (Property 3); when combined with observation (Property 7), prediction of future/past observable events is also possible.

II. Applications

This section considers the application of the OOFSM abstraction to dynamic systems. First, a simple pendulum is discussed. Next, a variation of the equation set is considered.

A. Pendulum

The simple pendulum is commonly described in many textbooks. The presentation here is based on [8], [9]. In particular, Figure 4 shows the phase space representation of theta vs. omega for the small angle approximation with simplified physical properties and initial conditions. Following from [8], this figure graphically presents the solution $\theta = a_i \cos t$ and $\omega = a_i \sin t$ for a_i representing closed fixed energy trajectories.

A simulation based on the earlier developed model with $0.05 \leq a < 1.5$ in increments of 0.05 and with $0.1 \leq t < 30$ in increments of 0.1 provides the four uniform region fields $\overline{V}_1 = \{(\{-1\}, \{1\})\}$, $\overline{V}_2 = \{(\{-1\}, \{-1\})\}$, $\overline{V}_3 = \{(\{1\}, \{-1\})\}$ and $\overline{V}_4 = \{(\{1\}, \{1\})\}$ corresponding to the four quadrants x_1, x_2, $-x_1, x_2$, $-x_1, -x_2$, and $x_1, -x_2$ respectively; this is graphically shown in Figure 5.

The trajectory $p_{a=1.25}$ corresponding with $a = 1.25$ is selected since its discrete trajectory, shown

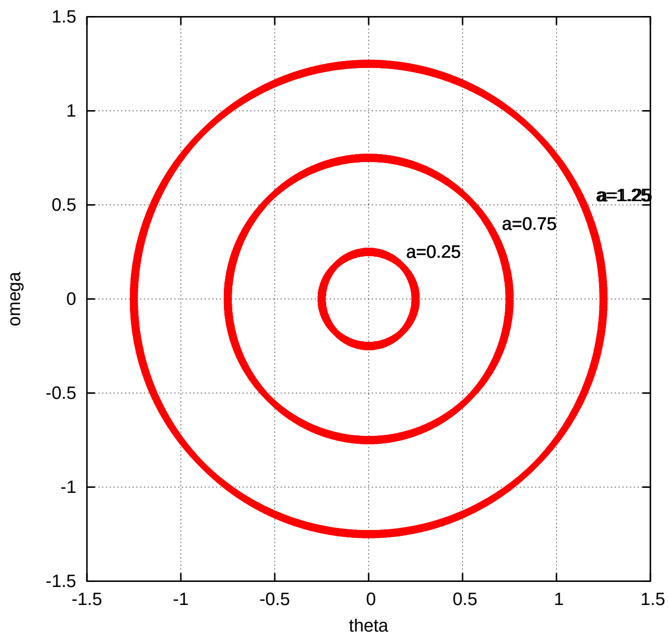

Fig. 4. A simple pendulum's omega vs. theta state space for simplified conditions.

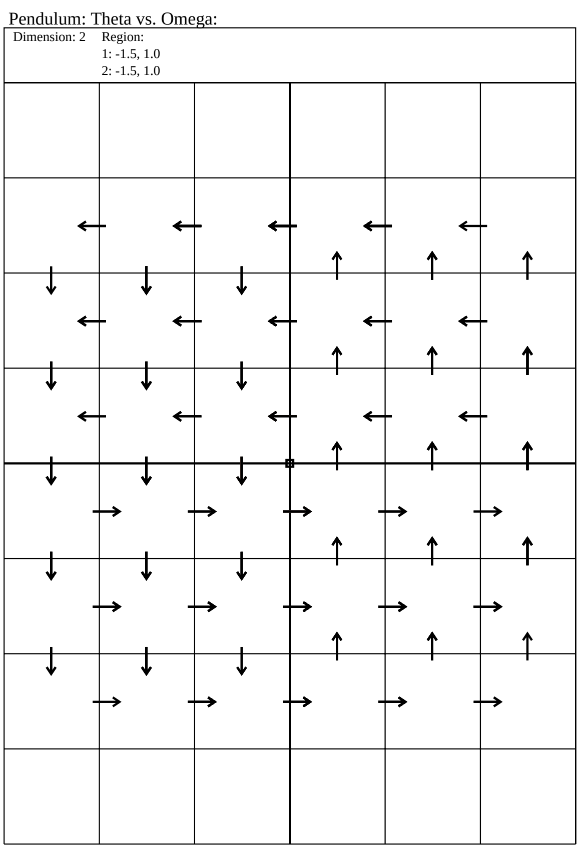

Fig. 5. Uniform region fields corresponding to the simple simplified pendulum.

in Figure 6, includes a number of interesting details useful in describing aspects related to intended visualization applications. The OOFSM structure, vector fields and trajectory behaviors are shown in the figures. The corresponding discrete path starts at $s_0 = y_{2,0}$ and is $s_p = 2\bar{1}2\bar{1}\bar{1}\bar{2}\bar{1}\bar{2}\bar{2}\bar{2}1\bar{2}1112122$. This path crosses four region boundaries and parts of the path exists in the four behavior regions. Let us assume that the only observable state variable is the angular displacement; observability is therefore reflected in the x_1 axis only. The components of s_p that are observable are $s_p = \phi\bar{1}\phi\bar{1}\bar{1}\bar{1}\phi\bar{1}\phi\phi\phi1\phi111\phi1\phi\phi$ where ϕ refers to non-observability. It is evident from this representation that the system exhibits an even oscillatory cycle consisting of five translations first in the negative direction and then in the position direction. The OOFSM by its structure provides for indexing. The finite state oscillation starts at $s_0 = y_{2,0}$ and transitions through $y_{i,j}$ where i is ordered (i.e. s_p) and constrained by $-3 \leq i \leq 2$ and j ranges as constrained by the system. Also evident from this representation is the fact that the trajectory crosses $x_1 = 0$ exactly twice, once in each direction, per oscillation. Combining this with the region behavioral properties, these transitions occur once from either $\overline{V}_1, \overline{V}_4$ to either $\overline{V}_2, \overline{V}_3$ and once vice-versa. Also from the established regions of behavior, there would be one transition from $\overline{V}_4$ to $\overline{V}_1$ and one from $\overline{V}_2$ to $\overline{V}_3$.

Following from Lemma 1, The language accepted by this finite state machine is $L = (\bar{1}+2)^*, = (\bar{1}+\bar{2})^*, = (1+\bar{2})^*, = (1+2)^*$, depending upon which uniform region field, respectively, the current state resides in.

B. *Variation*

A variation of the previous case is considered here. This case still considers a two dimensional system and following from the derivation previously, the governing equations are $a\cos(t) + \sin(0.5t) + 0.5$ and $a\sin(t)$. The discrete simulation here varied $0.4 \leq a \leq 0.7$ in increments of 0.1 and t as before.

Figure 7 shows the region fields determined by this simulation. This figure includes a textual markup to more conveniently identify the uniform region fields. Here, the spatial distribution of the

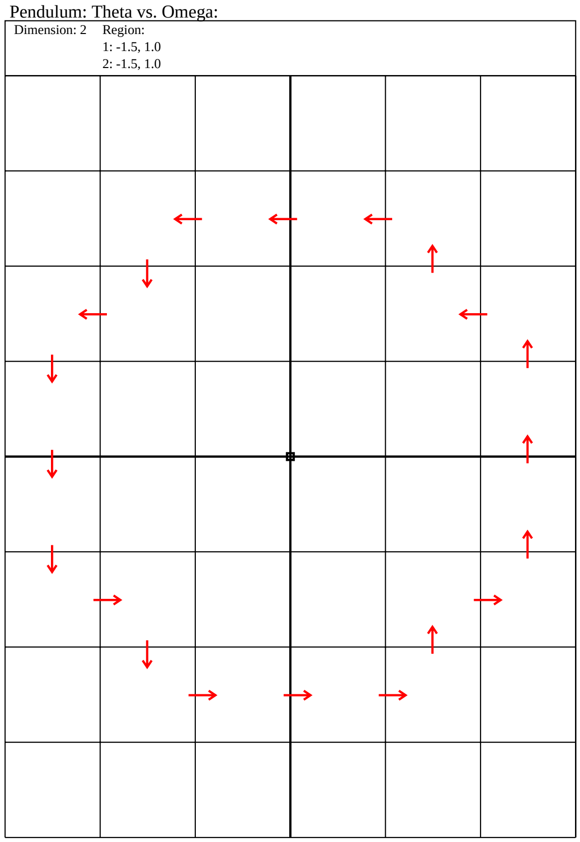

Fig. 6. Discrete direction vector field corresponding to $a = 1.25$ of the simple simplified pendulum.

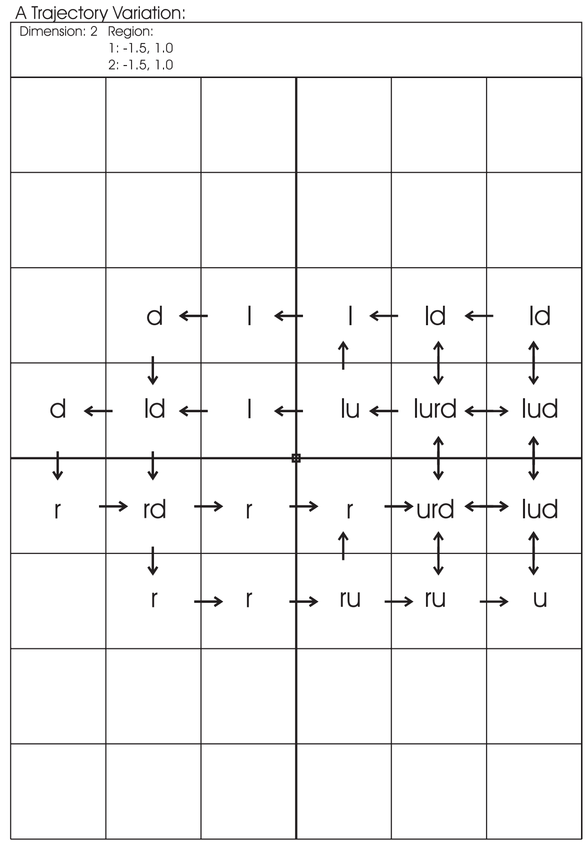

Fig. 7. Region fields for $a\cos(t) + \sin(0.5t) + 0.5, a\sin(t)$ for $0.4 \leq a \leq 0.7$, region fields are also identified by l–left, r–right, u–up and d–down

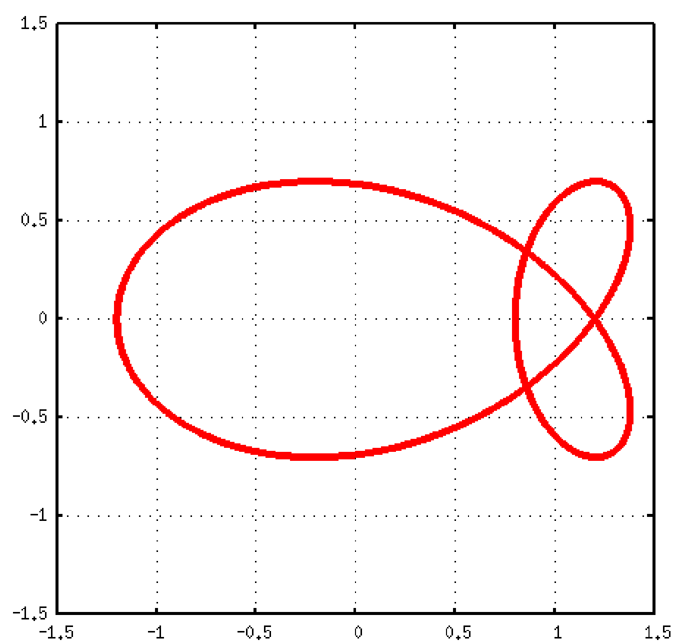

Fig. 8. Trajectory of $a\cos(t)+\sin(0.5t)+0.5, a\sin(t)$ with $a = 0.7$.

uniform region fields is more complex than in the previous case.

The trajectory shown in Figure 8 is generated with $a = 0.7$. Figure 9 shows the discrete direction vector fields generated by the simulation for this trajectory. The corresponding finite state representation is $s_0 = y_{1,0}$ and $s_p = \bar{1}2\bar{1}\bar{1}\bar{1}\bar{2}\bar{1}\bar{2}1\bar{2}1112122\bar{1}\bar{2}\bar{2}\bar{2}122$.

III. Conclusion

The main contribution of this paper is the proposed Orthogonal Organized Finite State Machine (OOFSM) together with its derivation to model the state space transitions induced by complex dynamic systems, linear and non-linear controllable systems. These types of systems are commonly found in many engineering problems. Although state space representations and discrete state space representations are commonly described, the OOFSM representation presented here appears to be novel. The OOFSM provides unique properties including indexing and language description that may enable

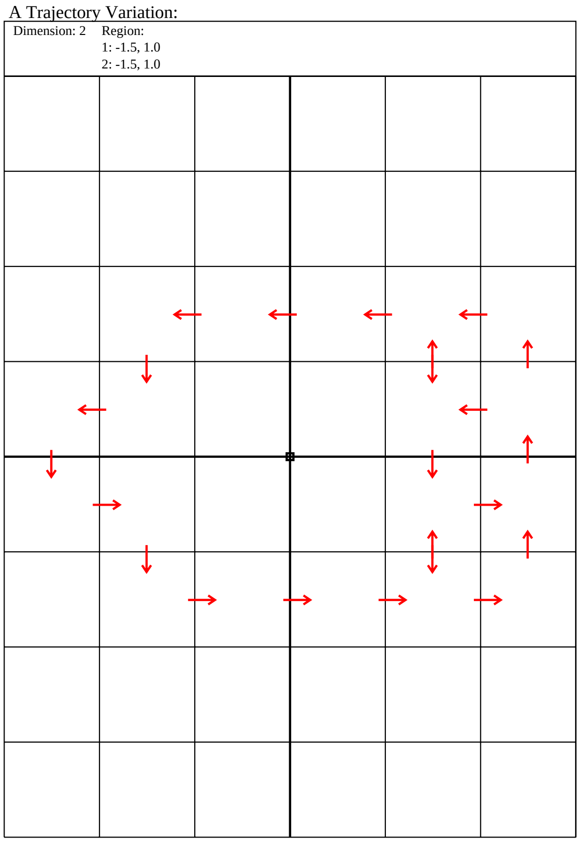

Fig. 9. Discrete direction vector field with $a = 0.7$ corresponding Figure 8.

further analysis of these types of systems. Its generality allows it to be widely applicable to small-scale and large-scale complex systems incorporating hundreds of state variables. Two simplified examples as applications of the OOFSM are also presented. These serve to illustrate the applicability of the OOFSM model and some of its properties.

Although the OOFSM model is derived from considering dynamic systems, its application is not limited to these types of systems. The model may be applied to other domains as well. The intent of this work is to define a broadly useful data abstraction that will support the visualization of diverse application domains, in particular, the visualization of complex dynamic systems.

ACKNOWLEDGEMENT

The support of The Ohio Supercomputer Center is acknowledged. The simulations described in this paper were conducted on an Itanium 2 cluster funded by the Hewlett-Packard Company under the Hewlett-Packard Company Advanced Technology Platforms - Itanium 2 2003 Academic Grant Initiative, P.I. Brian J. d'Auriol. The assistance of Dr. Judith Gardiner is appreciated.

REFERENCES

[1] R. C. Dorf, *Modern Control Systems, Second Edition*. Addison Wesley, 1974.

[2] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.

[3] H. Marchand, O. Boivineau, and S. Lafortune, "Optimal control of discrete event systems under partial observation, Tech. Rep. CGR-00-10, September 2000.

[4] S. Blouin, "Finite-state machine abstractions of continuous systems," Ph.D. dissertation, Chemical Engineering Department, Queens University, Kingston, Canada, October 2003.

[5] S. Jodogne, "Orthogonal finite-state representations," in *Sixth Meeting of the ADVANCE Project*, Edinburgh, March 2002. [Online]. Available: http://www.liafa.jussieu.fr/~haberm/ADVANCE

[6] L. wei He, M. F. Cohen, and D. H. Salesin, "The virtual cinematographer: A paradigm for automatic real-time camera control and directing," in *Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques*, New Orleans, Louisiana, USA, 1996, pp. 217–224.

[7] A. V. Aho, R. Sethi, and J. D. Ullman, *Compilers, Principles, Techniques, and Tools*. Addison Wesley, 1988.

[8] G. L. Baker and J. P. Gollub, *Chaotic Dynamics an Introduction*. Cambridge University Press, 1990.

[9] G. L. Baker and J. A. Blackburn, *The Pendulum A Case Study in Physics*. Oxford University Press, 2005.