International Journal of Intelligent Control and Systems
Vol. 2, No. 2 (1998) 211–224

OUTLINE OF A COMPUTATIONAL THEORY FOR LINGUISTIC DYNAMIC SYSTEMS: TOWARD COMPUTING WITH WORDS

FEI-YUE WANG
Systems and Industrial Engineering Department, the University of Arizona, Tucson, Arizona 85721, USA
E-mail: feiyue@sie.arizona.edu

This paper outlines a computational theory of linguistic dynamic systems for computing with words by fusing procedures and concepts from several different areas: Kosko's geometric interpretation of fuzzy sets, Hsu's cell-to-cell mappings in nonlinear analysis, equi-distribution lattices in number theory, and dynamic programming in optimal control theory. The proposed framework enable us to conduct a global dynamic analysis, system design and synthesis for linguistic dynamic systems that use words or linguistic terms in computation, based on concepts and methods well developed for conventional dynamic systems. This theory has significant potential for modeling and analysis of systems where model, goal, control and feedback are primarily specified in words or linguistic terms.

Keywords: Computing with words, fuzzy logic, linguistic dynamic systems, cell-to-cell mapping.

1. Introduction

For conventional dynamic systems modeled by differential/difference equations, various concepts and methods for both system analysis and synthesis have been well developed and widely used. For many large complex systems, however, due to the complexity involved and the intrinsic nature of information incompleteness, developing a conventional mathematical model to describe system behavior in a meaningful way are either infeasible or impracticable. Even in the case that a mathematical description has been established for a large complex system, in many situations it is still extremely difficult to covert collected data into forms appropriate to the model and then interpret obtained analysis and synthesis results in human understandable terms.

Fortunately, human being has accumulated working knowledge for many large complex systems from experience and practice. Generally, this kind of knowledge is expressed in natural languages, i.e., words or linguistic terms. Using this knowledge, we are able to describe, predict, control and evaluate behaviors of large complex systems. For example, operations of many large complex industrial processes cannot be modeled analytically, yet human operators are capable of monitoring the state of a given task process, and take appropriate control actions in response to the process state. And in the financial area, although mathematical market forecasting has been studied for many decades, experts usually assess market behaviors and specify actions to be taken in linguistic terms. In order to make the analysis and synthesis of this kind of systems systematic, consistent, and formal, we need to consider a special class of dynamic systems, i.e., *linguistic dynamic systems* (LDS), where problems (plants), situations (states), policies (controllers), observations (feedback), goal and evaluation (performance indices) are all or partially specified in linguistic terms. To deal with large complex systems effectively with linguistic human knowledge, especially when human and computers are integrated, a theory for modeling, analysis, and synthesis of linguistic dynamic systems must be developed.

Clearly, many efforts have already been taken toward this objective in the past.[1,2,10,11,15] For example, knowledge based systems, expert systems, linguistic structures, multi-valued logic, fuzzy logic, and so on. Although these methodologies have been successfully used to solve many problems in large complex systems, none of them has been able to establish a theoretical framework upon which concepts and methods for system analysis and synthesis parallel to those well known for conventional dynamic systems, such as stability analysis and control design, can be developed. Consider the problem of evaluating the stability of or designing a controller for a system described by a set of fuzzy IF-THEN rules, there has been no systematic approach to achieve it.

The objective of this paper is to propose a theory of linguistic dynamic systems that is computationally feasible and can utilize the concepts and methods developed for the analysis and synthesis of conventional dynamic systems, especially stability concepts, global analysis, and design methods for optimal controllers.[12] Figure 1 illustrates a block diagram representation of linguistic dynamic systems. Considering the process of making a policy or regulation to address certain problems. In this process, words are used to present the problems, describe the situation,

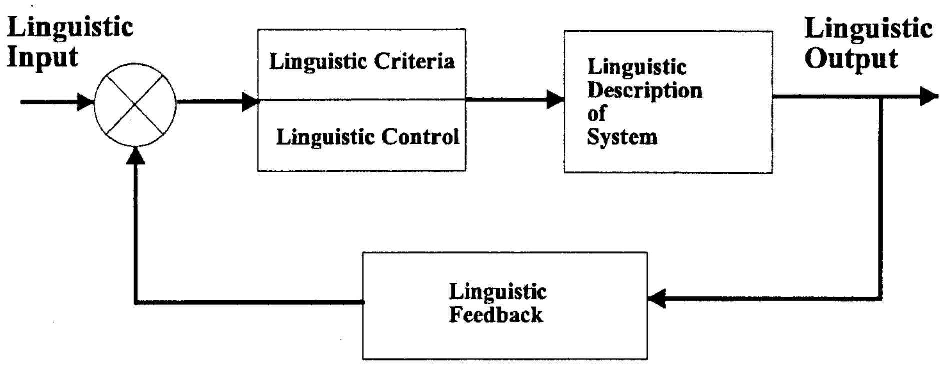

Fig. 1. Block diagram representation of linguistic dynamic systems.

state the goal of the policy, specify the policy, and finally define and the evaluation procedures. Correspondingly, a linguistic dynamic model for this process can be constructed as following:

- *Problems and Situation described in Words*
  Linguistic Plant Description
- *Goal Statement in Words*:
  Linguistic Objective Function
- *Policy Specification in Words*:
  Linguistic Control Design
- *Evaluation Procedures in Words*:
  Linguistic Feedback

This model will enable us to conduct dynamic simulation for different policies at a high level (word level).

Our basic idea here is to represent a LDS as a fuzzy dynamic system. Then, using Kosko's interpretation,[9] we can consider LDS as mappings on fuzzy hypercubes. By introducing cellular structures on hypercubes using equi-distribution lattices developed in number theory,[8] these mappings can be approximated by cell-to-cell mappings in a cell space,[5,6] in which each cell presents a linguistic term (a word) defined by a family of membership functions of fuzzy sets. In this way, LDS can be studied in the cell state space, and thus, methods and concepts of analysis and synthesis developed for conventional nonlinear systems, such as stability analysis and design synthesis, can be modified and applied for LDS. Especially, linguistic controllers for LDS can now be obtained using the dynamic programming method.

2. Fuzzy Sets as Points

Zadeh[16] first used a membership function to describe a fuzzy relationship between an entity and a set. Since then fuzzy sets has provided us an ideal tool to express the vagueness in human languages. But further research on methods to analyze and design fuzzy systems has difficulties based on this sets-as-functions representation. Kosko proposed an interpretation of fuzzy sets from a geometric point of view: *fuzzy sets as points*, which has been overlooked by most fuzzy theorists.[9] As we will see, this interpretation enable us to carry out the stability analysis and system synthesis for linguistic dynamical systems.

Let $V_d = \{v_1, v_2, \dots, v_n\}$ denote the discretized universe of discourse. Mapping $\mu_A : V_d \to [0,1]$ is a membership function for fuzzy set A. Traditionally, μ_A is only visualized as a two dimensional graph with domain V_d being on a one-dimensional axis. The new interpretation of a fuzzy set is based on the fuzzy power set $F(2^{V_d})$, which consists of all fuzzy subsets on V_d and is a unit N-dimensional *fuzzy hypercube* $I^N = [0,1]^N$. Using $F(2^{V_d})$, a fuzzy set is considered as a point or vector $X = (x_1, \dots, x_N)^T \in F(2^{V_d})$, where $x_i = \mu_A(v_i)$. According to this interpretation, the vertices of the cube I^N define the Boolean ordinary power set 2^{V_d} which consists of all 2^N nonfuzzy sets of V_d. So fuzzy sets fill in the lattice 2^{V_d}

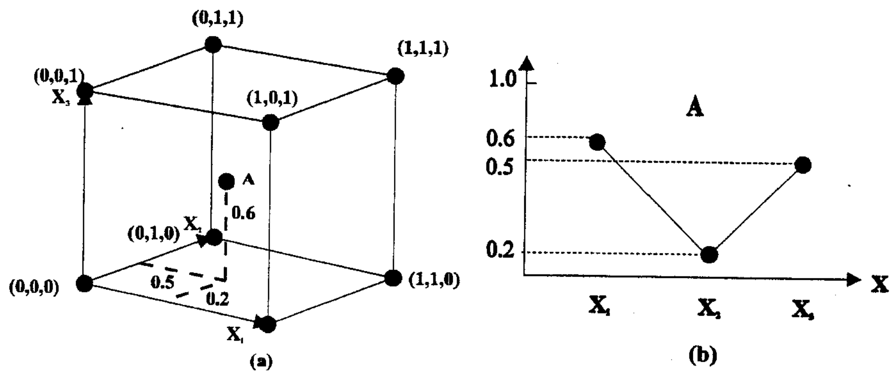

Fig. 2. Fuzzy hypercube and membership function.

to produce a solid fuzzy hypercube $F(2^{V_d}) = I^N$. Figure 2 presents an example of fuzzy hypercube when $N = 3$, where a point A (Fig. 2(a)) in the hypercube actually represents a membership function (Fig. 2(b)).

Clearly, a fuzzy mapping can now be considered as a conventional mapping on fuzzy hypercubes. This provides us a basis for studying linguistic dynamic systems on fuzzy hypercubes.

Based on fuzzy hypercubes, concepts such as the distance between two fuzzy sets and a neighborhood of a fuzzy set can be introduced easily. For example, The distance between two fuzzy sets, represented as points X_1 and X_2 in a fuzzy hypercube, can be defined as,

$$d_W(X_1, X_2) = \|X_1 - X_2\|_W, \quad \|X\|_W = \sqrt{X^T W X} \tag{1}$$

where W is a metric, i.e., positive and definitive matrix. W can be used to reflect the subjectiveness of a person's view of membership functions. Correspondingly, a quantitative measure of vicinity of a fuzzy set X_0 can be defined as,

$$N(X_0, \delta) = \{X | d_W(X, X_0) \le \delta,\ X \in I^N\}, \quad \delta > 0 \tag{2}$$

where $N(X_0, \delta)$ denotes the neighborhood of X_0 with radius δ.

3. Formulation of Linguistic Dynamic Systems

Based on the discussion in the previous section, we can consider a conventional numerical dynamical system as a time dependent mapping from R^n to R^n, and a linguistical dynamical system as a time dependent mapping from I^N to I^N, where N is much larger than n (see Fig. 3). Furthermore, using fuzzy sets as linguistic terms, we can model a linguistic dynamic system as a fuzzy dynamic system on fuzzy hypercubes. Generally, a LDS can be described as,

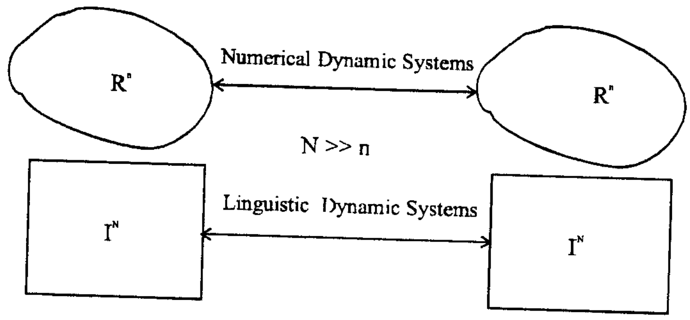

Fig. 3. Numerical vs linguistic dynamic systems.

State Equation:

$$X_{k+1} = F(X_k, U_k, k)\,, \quad \text{where} \quad F : I^n \times I^m \times Z^+ \to I^n \tag{3}$$

Output Equation:

$$Y_k = H(X_k, k)\,, \quad \text{where} \quad H : I^n \times Z^+ \to I^p \tag{4}$$

Feedback Control:

$$U_k = R(Y_k, V_k, k)\,, \quad \text{where} \quad R : I^p \times I^q \times Z^+ \to I^m \tag{5}$$

Note that $Z^+ = \{0, 1, \ldots\}$, $X_k \in I^n$ is a vector representing a term for the state of the system, $Y_k \in I^p$ the output term, $V_k \in I^q$ the input term, $U_k \in I^m$ the control term, k a discrete time instance, and F, H, R are fuzzy logic operators which define the system, output, and control mappings of the LDS, respectively.

The universe of discourse of each variable of the above system is defined as,

$$D_X = \{x_1, \ldots, x_n\}\,,$$
$$D_Y = \{y_1, \ldots, y_p\}\,,$$
$$D_U = \{u_1, \ldots, u_m\}\,,$$
$$D_V = \{v_1, \ldots, v_q\}\,.$$

The corresponding terms or fuzzy sets are defined as

$$X = \sum_{x_i \in D_X} \mu_X(x_i)/x_i\,,$$

$$Y = \sum_{y_i \in D_Y} \mu_Y(y_i)/y_i\,,$$

$$U = \sum_{u_i \in D_U} \mu_U(u_i)/u_i\,,$$

$$V = \sum_{v_i \in D_V} \mu_V(v_i)/v_i\,.$$

Clearly, various fuzzy rule-based systems are special cases of our formulation.

Incorporating the output equation and feedback control into the state equation, we can write a LDS in the form of,

$$X_{k+1} = F(X_k, V_k, k)\,, \quad k = 0, 1, 2, \ldots\,. \tag{6}$$

When there is no input, i.e., $V_k = 0$, the LDS is called an *autonomous LDS*.

4. Stability Definitions of Linguistic Dynamic Systems

Using fuzzy hypercubes, we can directly generalize almost all the concepts developed for conventional dynamic systems to a LDS. The following are some examples for a time-invariant autonomous LDS $X_{k+1} = F(X_k)$.

Trajectory of a LDS: Let F^k denote mapping F applied k times with F^0 as the identity mapping. Starting with an initial term X_0, LDS $X_{k+1} = F(X_k)$ generates a sequence of terms $X_k = F^k(X_0)$, $k = 0, 1, 2, \ldots$. This sequence of terms is called a *term trajectory* or simply *trajectory* of the LDS with initial term X_0.

Limit Term of a Term: Consider a trajectory $F^k(X)$, $k = 0, 1, 2, \ldots$. A term X^* is said to be a *limit term* of X under F if there exists a sequence of non-negative integers n_k such that $n_k \to \infty$ and $F^{n_k}(X) \to X^*$ as $k \to \infty$. The set of all limit terms of X is denoted by $\Omega(X)$.

Positively (Negatively) Invariant Term Set: For a set of terms, $\Theta \subset I^N$, $F(\Theta)$ denotes the one-step image terms of Θ under mapping F. A term set Ψ is said to be *positively* (*negatively*) *invariant* under mapping F if $F(\Psi) \subset \Psi$ ($\Psi \subset F(\Psi)$). Ψ is called *invariant* under F if $F(\Psi) = \Psi$.

Equilibrium Term: A term X is said to be an *equilibrium term* if $X = F(X)$.

P-K Trajectory: A periodic trajectory of the LDS with period k is a sequence of k distinct terms X_k, $i = 1, 2, \ldots, k$, such that

$$X_{i+1} = F^i(X_1)\,, \quad i = 1, \ldots, k-1\,, \quad X_1 = F^k(X_1)$$

Any of the terms X_i, $i = 1, 2, \ldots, k$, is called a periodic term of period k. We call term sequence $(X_1, \ldots, X_k)$ a *P-K trajectory* and any of its elements a *P-K term*.

Stability: An equilibrium term $X^* \in I^N$ is *stable* if there is an $\epsilon_0 > 0$ with the following property: for all ϵ_1, $0 < \epsilon_1 < \epsilon_0$, there is an $\epsilon > 0$ such that whenever $\|X - X^*\|_W < \epsilon$ and $X \in I^N$, then $\|F^k(X) - X^*\|_W < \epsilon_1$ for all $k \geq 0$.

Asymptotical Stability: An equilibrium term $X^* \in I^N$ is *asymptotically stable* if it is stable and there is an $\epsilon_0 > 0$ with the following property: for all ϵ_1, $0 < \epsilon_1 < \epsilon_0$, if $\|X - X^*\|_W < \epsilon_1$ and $X \in I^N$, then $F^k(X) \to X^*$ as $k \to \infty$.

Global Asymptotical Stability: An equilibrium term $X^* \in I^N$ is *global asymptotical stable* if it is stable, and with arbitrary initial term $X_0 \in I^N$, $F^k(X_0) \to X^*$ as $k \to \infty$.

One may argue that our discussion so far works only for discrete universes of discourse while many fuzzy sets use continuous ones that would lead to hypercubes of infinite dimensions. In practice, this is not a problem since membership functions are subjective: we do not need to specify them with high accuracy; instead we only need to know their values at some typical sample points in a universe of discourse. Of cause, fine discretization of an universe of discourse will result in an accurate approximation of membership functions and a hypercube of higher dimension.

In theory, the above definitions will enable us to conduct modeling and analysis of LDS by applying various methods established for conventional numerical dynamic systems. However, a direct extension from conventional dynamic systems to LDS would not work for most of practical applications.

The reason is simple. Two adjacent points in a fuzzy hypercube will be treated as two different points in the above definitions. Therefore, a LDS that oscillates between two adjacent points will be considered as unstable. However, in terms of the two words represented by the two points, these two points might be considered as same in a practical sense and the system should be considered as stable. For example, Fig. 4 shows a point A and its adjacent point B in a fuzzy hypercube. Although A and B are treated as two different points (Fig. 3(a)), their corresponding membership functions are very close (Fig. 3(b)), and should be normally treated as

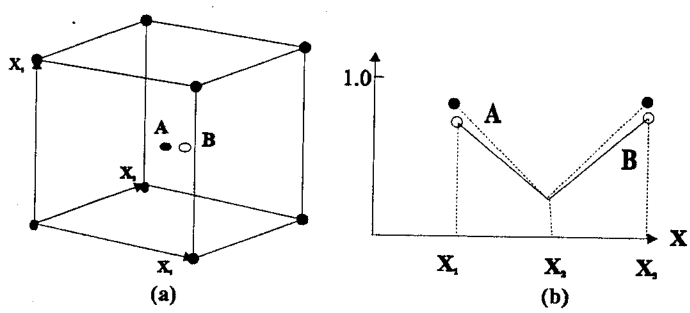

Fig. 4. A point and its adjacent in fuzzy hypercube.

the same linguistic term due to the nature of the subjectiveness of membership functions. This consideration leads us to consider the possibility of treating points in a small neighborhood as a single point, i.e., taking the entire neighborhood as a single "linguistic term". Another argument for this consideration is the subjectiveness of fuzzy logic operators. It is well known that different operators can be used for the same logic operations, thus a fuzzy mapping can map a point on the domain hypercube to different points on the image hypercube under different logic operators. Since these image points should be close and within a small neighborhood, the whole neighborhood could be considered as a single image term. These observations have inspired us to introduce cellular structures to fuzzy hypercubes.

5. Cell Spaces on Fuzzy Hypercubes

The cellular structure of a cell space can be introduced in various ways. Figure 5 shows a general cellular structure. The most simple one is the uniform cell division of a fuzzy hypercube. The basic idea behind the fuzzy cell space is to consider a fuzzy hypercube space not as a continuum but as a collection of cells, with each cell being taken as an entity.

Let x_i, $i = 1, 2, \ldots, N$, be N components of a point in the fuzzy hypercube I^N, and let the coordinate axis of variable x_i be divided into a large number of intervals of uniform interval size h_i. Interval z_i along the x_i-axis is defined to be the one that contains all x_i satisfying

$$z_i - h_i/2 \leq x_i < z_i + h_i/2\,, \quad 0 \leq x_i \leq 1$$

Such an N-tuple $(z_1, z_2, \ldots, z_N)$ is then called a *fuzzy cell vector* (or *fuzzy cell, cell*) and is denoted by Z. The *fuzzy cell state space* S_C is the collection of all such cell vectors. The cell center is a representative of the cell in the sense that all points in the cell have similar membership functions and are considered as a single linguistic term.

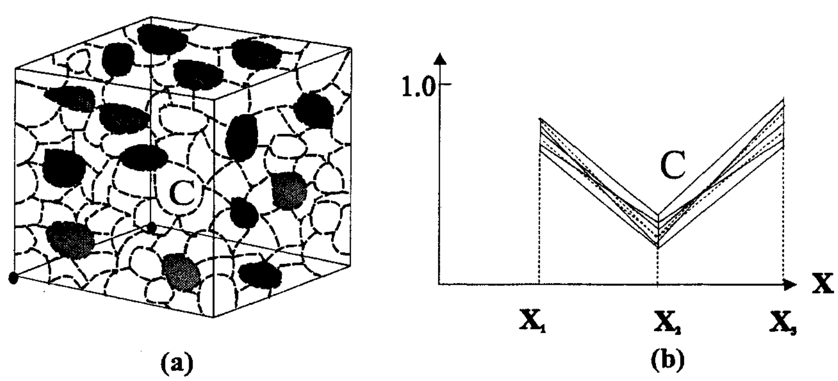

Fig. 5. Cellular structure over fuzzy hypercube.

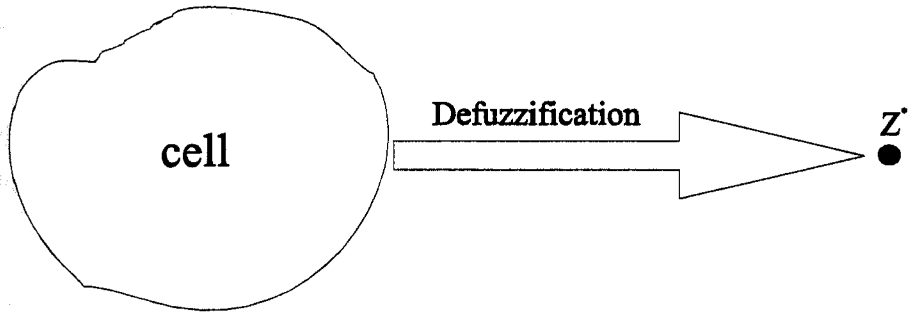

Fig. 6. Defuzzification of a fuzzy cell.

An essential question for cell space construction is that, for a specified number N_t of terms to be used to describe a system, how can we create a cellular structure so that these N_t terms or cells can be "optimally" distributed in a fuzzy hypercube? This question is especially important for dealing with high dimensional hypercubes, since we can only assigned a limited number of cells. One way to define "optimality" is to distribute these terms as uniform as possible. This can be achieved by using the equi-distribution lattices developed in number theory for numerical integration in high dimensional spaces.[4] For example,

$$Z_i = \left(\left\{\frac{i}{F_n}\right\}, \left\{\frac{iF_n(2)}{F_n}\right\}, \ldots, \left\{\frac{iF_n(N)}{F_n}\right\}\right),$$

where $1 \le i \le N_t = F_n$, $\{w\}$ is the decimal remainder of w, $F_0 = F_1 = \cdots = F_{N-2} = 0$, $F_{N-1} = 1$, $F_k = F_{k-1} + \cdots + F_{k-N+1} + F_{k-N}$, $k \ge N$, and $F_k(j) = F_{k+j-1} - F_{k+j-2} - \cdots - F_k$, $2 \le j \le N$. Cell Z_i includes all points X such that $d_W(X, Z_i) = \min_{1<j<N_t} d_W(X, Z_j)$. Many other equi-distribution lattices can be found in Ref. 8.

Based on the defuzzification of fuzzy sets, we can introduce the defuzzification of a fuzzy cell. For example, the *center of area* method can be generalized to defuzzify a cell Z as,

$$Z^* = \sum_{i=1}^{N} \alpha_i v_i\,, \quad \alpha_i = \int_V \frac{x_i}{\sum_{j=1}^{N} x_j} dV/|V| \tag{7}$$

where V is the domain occupied by Z (see Fig. 6). Obviously, Z^* can be used to calculate the fuzzy cell vector for a fuzzy cell.

6. Analysis of LDS using Cell-to-Cell Mappings

Once cell spaces have been established on fuzzy hypercubes, we can use the methods developed by Hsu[5,6] for nonlinear differential dynamic systems to convert a LDS defined by Eqs. (3–5) to cell-to-cell mappings,

$$Z_{k+1} = C_F(Z_k, Q_k, t_k),$$
$$P_k = C_H(Z_k, t_k), \quad Q_k = C_R(P_k, W_k, t_k) \tag{8}$$

where t_k is the new time interval, $Z_k \in S_C^X$, $Q_k \in S_C^U$, $P_k \in S_C^Y$, $W_k \in S_C^V$ are cells defined in the fuzzy state, control, output, and input hypercubes, and C_F, C_H, C_R are cell-to-cell mappings constructed from F, H, and R. Combining the above equations into one, we get,

$$Z_{k+1} = C(Z_k, W_k, t_k), \tag{9}$$

Z_k and Z_{k+1} are called the *domain cell* and *image cell*, respectively.

All concepts introduced for a LDS in fuzzy hypercubes can directly be used in cell spaces. In cell spaces, however, the system analysis becomes computational feasible. Once the cell mapping C is known, we can systematically study the global dynamic properties of a LDS, i.e., equilibrium cells, P-K trajectories, and stabilities. For the purpose of numerical analysis, we introduce the following two additional definitions:

R-Step Removable Cell: A cell Z is said to be *r-step removable* from a P-K trajectory if r is the minimum integer such that $C^r(Z) = Z_j$, where Z_j is any one of the P-K cells in a P-K trajectory.

R-Step Domain of Attraction: For a P-K trajectory, the set of cells that are removed from the trajectory in r or less steps is called its *r-step domain of attraction.* When $r \to \infty$, we get the *total domain of attraction* or simply the *domain of attraction* for a P-K trajectory.

Since the number of cells is finite, a cell is a member of either a P-K trajectory or a r-step domain of attraction of a P-K trajectory. A global search algorithm for analysis of autonomous dynamic systems using cell mappings has been developed in Ref. 7. This search algorithm will find the total number of different P-K trajectories of a dynamic system, denoted as M, and link each cell with one of P-K trajectories for an autonomous LDS $Z_{k+1} = C(Z_k)$. Assume we have enumerated all its P-K trajectories by $g = 1, \ldots, M$. Let $S(Z)$ be the number of steps for cell Z to reach a P-K trajectory g, $G(Z) = g$ the group number of the P-K trajectory reached, and $P(Z) = k$ the periodic number of the P-K trajectory. To start the search algorithm, we first set $M = 0$, and classify cells in the searching process into three types, i.e., *unprocessed cells*, *cells under processing*, and *processed cells.* The group number of unprocessed cells are set to be zero at the beginning of the algorithm. When an unprocessed cell is called upon to be processed, we change its group number to -1. After the process, a cell is assigned a group number and becomes a processed cell.

Consider an unprocessed cell Z, when it is called upon for processing and generates a sequence of image cells,

$$Z \to C(Z) \to C^2(Z) \to \cdots \to C^j(Z)$$

then the search algorithm can be described as,

1. The newly generated image cell $C^j(Z)$ is an unprocessed cell. In this case, we change its group number to -1, which indicates it has become a cell under processing, and proceed to the next image cell $C^{j+1}(Z)$.
2. The newly generated cell $C^j(Z)$ is a processed cell with a positive group number. In this case, the current processing sequence is mapped into a cell with known properties, i.e., its group number, step number and periodic number have been found. Obviously, all cells in the sequence will have the same group and periodicity numbers with those of $C^j(Z)$. The step numbers of the other cells in the sequence can be calculated as,
$$S(C^i(Z)) = S(C^j(z)) + j - i\,, \quad 0 \le i \le j-1 \tag{10}$$
All these cells are then marked as processed and the search process continues with another unprocessed cell.
3. The newly generated cell $C^j(Z)$ is a cell under processing. In this case, a new periodic motion is discovered, therefore $M = M+1$. All the cells in the sequence are assigned a new group number $g = M$. Assume that the first reappearing cell to be the $(i-1)$th cell in the sequence, i.e., $C^l(Z)$=$C^i(Z)$ for the minimum $l > i$. Then the periodicity number is $(l-i)$ for this new periodic trajectory and it is assigned to every cell in the sequence. The step numbers of the cells be determined by
$$\begin{aligned} &S(C^k(Z)) = i - k\,, \quad k = 0, 1, \ldots, i-1\,, \\ &S(C^k(Z)) = 0\,, \quad k \ge i\,, \end{aligned} \tag{11}$$
When each of these three possible situations has been processed, the algorithm will call another unprocessed cell and begin a new search.

This process will be conducted repetitively until every cell in the cell space has been processed and the global dynamic behavior of the system is then completely determined. As a result, the cell space is divided into different P-K trajectories and their domains of attractions. Every cell in the cell space belongs to a certain group. The step number of a cell indicates the distance of the cell to its converging periodic trajectory.

7. Design of Optimal Control in Cell Spaces

The problem of optimal control design for a LDS is to find a linguistic feedback law that optimizes a linguistic performance index,

$$J(U; X_0) = \sum_{i=0}^{T} \Phi(X_k, U_k, V_k, k)\,, \quad X_T \in \Gamma \subset I^n$$

where Γ is the linguistic target subset representing the terms to be reached in T steps, $\Phi : I^n \times I^m \times I^q \times Z^+ \to I^l$ is the logic operator of the one-step linguistic cost function, I^l is the fuzzy hypercube of the one-step cost universe $D_\Phi = \{\phi_1, \ldots, \phi_l\}$. To make the fuzzy addition meaningful, we can assume that Φ always generates a fuzzy number.[3] However, as we can see, this is not important in our formulation. For the sake of simplicity, we assume $Y = X$, i.e., output terms are the same as state terms.

In cell spaces, one-step cost function can be approximated by a cell-to-cell mapping,

$$\Phi_k = C_\Phi(Z_k, Q_k, W_k, t_k)\,, \quad Z_T \in S_\Gamma \subset S_C^X$$

where S_Γ is the cell subspace of the target subspace, and Φ_k is a cell in I^l. Using the defuzzification method introduced in Sec. 5, we can defuzzify Φ_k so that a numerical value $\gamma_k = DF(\Phi_k)$ is obtained for cell Φ_k, i.e.,

$$\gamma_k = C_\Phi(Z_k, Q_k, W_k, t_k)\,, \quad Z_T \in S_\Gamma \subset S_C^X \tag{12}$$

Clearly, with cell-to-cell mappings (5) and (9), the design of optimal controller for a LDS becomes a standard dynamic programming problem with discrete state and control variables. Therefore, all the methods developed for dynamic programming can be used here.

A general search algorithm for optimal linguistic control has been proposed in Ref. 13. In this method, the one-step cost is quantified into finite levels, $\beta_1\Delta < \beta_2\Delta < \cdots < \beta_L\Delta$, where β_i are integers and Δ is a quantifying unit. As a result, the accumulated cost can be expressed as,

$$J = \Delta \sum_{i=1}^{L} k_i \beta_i$$

where k_i are integers. In this way, the accumulated cost is also quantified into finite levels.

The search procedure can be outlined simply as,

Step 1: Set $S_0 = S_\Gamma$ and $i = 1$. Mark all cells in S_Γ as processed, and all other cells as unprocessed.

Step 2: Search through all unprocessed cells to find the cells that can be mapped into $S_{i-1} \cup \cdots \cup S_0$ with the ith accumulated cost level. Let S_i be the collection of all newly founded cells. Mark all the cells in S_i as processed.

Step 3: If all interested cells have been marked as processed, stop. Otherwise, set $i = i + 1$, and go back to Step 2.

After the search algorithm is complete, an optimal control table is established that gives the optimal control action at each cell to enter the specified target S_Γ. Note that the result of this search process is independent of the initial state. Discussions on parallel search algorithms have been given in Ref. 14.

8. Conclusion

A computational theory for linguistic dynamic systems (LDS) has been outlined in this paper. The proposed theory is a fusion of procedures and concepts from several different areas: geometric theory of fuzzy sets, cell-to-cell mappings in nonlinear analysis, equi-distribution lattices in number theory, and dynamic programming in optimal control theory. This theory enables us to numerically conduct the global dynamic analysis, system design and synthesis for linguistic dynamic systems based on concepts and methods developed for conventional dynamic systems.

References

1. M. Braae and D. A. Rutherford, Theoretical and linguistic aspects of the fuzzy logic controller, *IFAC J. Automatica* **15**, 5 (1979) 553–577.
2. B. Cleveland and A. Meystel, Dynamic predictive planning + fuzzy logic control = Intelligent control, *Proc. of IEEE Symp. Intell. Control*, Philadelphia, PA, 1990.
3. A. Bardossy and L. Duckstein, *Fuzzy Rule-Based Modeling in Geophysical, Economic, Biological, and Engineering Systems*, Boca Raton, FL: CRC Press, 1995.
4. J. H. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, *Numerical Mathematics* **2** (1960) 84–90.
5. C. S. Hsu, *Cell-to-Cell Mapping*, New York, NY: Springer-Verlag, 1987.
6. C. S. Hsu, A generalized theory of cell-to-cell mapping for nonlinear dynamical systems, *ASME J. Applied Mechanics* **48** (1981) 634–842.
7. C. S. Hsu and R. S. Guttalu, An unravelling algorithm for global analysis of dynamic systems: An application of cell-to-cell mapping, *ASME J. Appl. Mechanics* **47** (1985) 940–948.
8. L. K. Hua and Y. Wang, *Applications of Number Theory to Numerical Analysis*, New York, NY: Springer-Verlag, 1981.
9. B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Englewood Cliffs, NJ: Prentice Hall, 1992.
10. E. H. Mamdani and S. Assilian, Applications if fuzzy logic to approximate reasoning using linguistic synthesis, *IEEE Trans. Computer* **C-26**, 12 (1977) 1182–1191.
11. T. J. Procyk and E. H. Mamdani, A linguistic self-organizing process controller, *IFAC J. Automatica* **14** (1978) 15–30.
12. F.-Y. Wang, Modeling, analysis and synthesis of linguistic dynamic systems: A computational theory, *Proc. of IEEE Int'l Workshop on Architecture for Semiotic Modeling and Situation Control in Large Complex Systems*, Monterey, CA, August 1995.
13. James Bing Pu, *A Cell-to-Cell Mapping Based Analysis and Design of Fuzzy Dynamic Systems and Its Applications*, Ph.D. Dissertation, SIE Department, U. of Arizona, 1995.
14. F.-Y. Wang and X. Y. Fan, Time-optimal trajectory generation for coordinated robot arms handling a common object using the cell-to-cell mapping method, *J. Optimization Theory and Applications* **86**, 2 (1995).
15. P. P. Wang and C-Y. Tyan, Fuzzy dynamic system and fuzzy linguistic controller classification, *IFAC J. Automatica* **30**, 12 (1994).
16. L. A. Zadeh, Fuzzy sets, *Information and Control.* **2** (1965) 338–353.

Fei-Yue Wang received the BE degree in Chemical Engineering from Shandong Institute of Chemical Engineering, Qingdao, China, the MS degree in Mechanics from Zhejiang University, Hangzhou, China, and the PhD degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, New York, in 1981, 1984, and 1990, respectively. Since 1990, he has been with the Systems

and Industrial Engineering Department at the University of Arizona. His areas of technical expertise include intelligent control, real-time embedded systems, mechatronics, robotics and automation, computer vision, computer-integrated manufacturing, operation control and scheduling. Dr. Wang has published more than 50 refereed journal articles and book chapters, and more than 100 conference papers and technical reports in the areas of applied mathematics, mechanics, mechatronics, artificial intelligence, control theory, communication, computer vision, robotics and automation, computer-integrated manufacturing, fuzzy logic, neural networks, and intelligent machines. Prof. Wang has been a principal investigator or co-principal investigator on several major research projects, funded by NSF, DOE, DOT, AT&T, BHP, and Caterpillar. He is the editor-in-chief of *International Journal of Intelligent Control and Systems* and the program chair of 1998 IEEE International Symposium on Intelligent Control.