Interactions Between Computational Verbs

Tao Yang

***Abstract*— Different types of interactions between computational verbs are studied in a general context. The interactions can be categorized by their strengthes and directions as well as the configurations of interactions. Two interaction types; namely, coupling and parameter modulation are studied. The identical synchronization and generalized synchronization between computational verbs and their linguistic implications are investigated. The merge and split of computational verbs are constructed as two special forms of interactions between computational verbs.** *Copyright © 2008 Yang's Scientific Research Institute, LLC. All rights reserved.*

***Index Terms*— Computational verb, synchronization, general synchronization, dynamical system.**

I. Introduction

AN adverb can function as a mathematical operator acting upon the evolving function or the evolving system of a computational verb[48]. This is the mostly studied computations acting upon computational verbs. However, as studied long times ago by the author, there are many other kinds of computations between computational verbs[23]. For example, in [23] the author studied the computation between must and a computational verb V. In this paper, the author will revisit the computation between computational verbs in a more general context; namely, the computation between computational verbs will be studied as the interaction between computational verbs.

The recent motivation of studying the interaction between computational verbs is the reading of a few Chinese linguistic papers addressing the issues of the history and mechanism of the transforming process of some Chinese verbs into prepositions. There are some Chinese prepositions that had been historically solely used as verbs, however, as time went by, their roles as verbs become weaker and weaker until they become prepositions. Chinese linguists found the following mechanism of triggering the transform of verbs to prepositions. When two Chinese verbs were used in the same sentence and very much close to each other, some verbs were induced to undergo this transforming procedure. This observation immediately caught the author's attention because it is similar to a generalized synchronization between two verbs.

Recalling author's previous study of the computation between computational verbs, adverbs and computational verbs and the evolution of Chinese prepositions, the author realized that there is a unified computational platform for all these computations involving computational verbs. Since a computational verb is modeled by a dynamical system, any computation involving a computational verb is inevitably related to manipulate dynamical systems. From mathematically point of view, there are many ways to manipulate a dynamical system, some of them are listed as follows.

- Manipulate the phase space of a dynamical system. We can transform the phase space of a dynamical system by using different mathematical operators, which usually function as adverbs.
- Manipulate the time. Many temporal adverbs serve as the mathematical operators for this purpose.
- Manipulate the configurations, parameters and structures of the dynamical system. For example, we couple two verbs into one by modifying the configuration of equilibrium points.

The main focus here is to manipulate a verb using another verb; namely, to study the interaction between verbs.

The organization of this paper is as follows. In Section II, the brief history of computational verb theory will be given. In Section III, the mathematical representations of interactions between computational verbs will be presented. In Section IV, the identical synchronization between two computational verbs are studied and its linguistic implications will be given. In Section V, the reverse synchronization between two computational verbs are studied and its linguistic implication will be addressed. In Section VI, the linear generalized synchronization between two computational verbs will be studied. In Section VII, the process of merging two computational verbs into one computational verb will be presented. In Section VIII, the process of splitting a computational verb into two or more computational verbs will be addressed. In Section IX, some concluding remarks will be included.

II. A Brief History of Computational Verb Theory

As the first paradigm shift for solving engineering problems by using verbs, the computational verb theory[28] and physical linguistics[31], [48], [22] have undergone a rapid growth since the birth of computational verb in the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley in 1997[13], [14]. The paradigm of implementing verbs in machines was coined as *computational verb theory*[28]. The building blocks of computational theory are *computational verbs*[23], [17], [15], [24], [29]. The relation between verbs and adverbs was mathematically defined in [16]. The logic operations between verb statements were studied in [18]. The applications of verb logic to verb reasoning were addressed in [19] and further studied in [28]. A logic paradox was solved based on verb

Manuscript received October 10, 2008; revised October 22, 2008.

Tao Yang, Department of Electronic Engineering, Xiamen University, Xiamen 361005, P.R. China. Department of Cognitive Economics, Department of Electrical Engineering and Computer Sciences, Yang's Scientific Research Institute, 1303 East University Blvd., #20882, Tucson, Arizona 85719-0521, USA. Email: taoyang@xmu.edu.cn,taoyang@yangsky.com,taoyang@yangsky.us.

Publisher Item Identifier S 1542-5908(08)10312-8/$20.00

logic[25]. The mathematical concept of set was generalized into verb set in[21]. Similarly, for measurable attributes, the number systems can be generalized into verb numbers[26]. The applications of computational verbs to predictions were studied in [20]. In [30] fuzzy dynamic systems were used to model a special kind of computational verb that evolves in fuzzy spaces. The relation between computational verb theory and traditional linguistics was studied in [28], [31]. The theoretical basis of developing computational cognition from a unified theory of fuzzy and computational verb theory is the theory of the UNICOGSE that was studied in [31], [36]. The issues of simulating cognition using computational verbs were studied in [32]. In [57] the correlation between computational verbs was studied. A method of implementing feelings in machines was proposed based on grounded computational verbs and computational nouns in [38]. In [45] a theory of how to design stable computational verb controllers was given. In [39] the rule-wise linear computational verb systems and their applications to the design of stable computational verb controllers and chaos in computational verb systems were presented. In [43] the concept of computational verb entropy was used to construct computational verb decision tree for data-mining applications. In [42] the relation between computational verbs and fuzzy sets was studied by using computational verb collapses and computational verb extension principles. In [44] the distances and similarities of saturated computational verbs were defined as normalized measures of the distances and similarities between computational verbs. Based on saturated computational verbs, the verb distances and similarities are related to each other with a simple relation. The distances and similarities between verbs with different life spans can be defined based on saturated computational verbs as well. In [46] the methods of using computational verbs to cluster trajectories and curves were presented. To cluster a bank of trajectories into a few representative computational verbs is to discover knowledge from database of time series. We use cluster centers to represent complex waveforms at symbolic levels. In [11] computational verb controllers were used to control a chaotic circuit model known as Chua's circuit. Computational verb controllers were designed based on verb control rules for different dynamics of the region-wise linear model of the control plant. In [10] computational verb controllers were used to synchronize discrete-time chaotic systems known as Hénon maps. Different verb control rules are designed for synchronizing different kinds of dynamics. In [50], how can computational verb theory functions as the most essential building block of cognitive engineering and cognitive industries was addressed. Computational verb theory will play a critical important role in personalizing services in the next fifty years. In [47], [49] computational verb theory was used to design an accurate flame-detecting systems based on CCTV signal. In [53] the learning algorithms were presented for learning computational verb rules from training data. In [51] the structures and learning algorithms of computational verb neural networks were presented. In [55] the ambiguities of the states and dynamics of computational verbs were studied. In [52] the history and milestones in the first ten years of the studies of computational verb theory were given. In [3] a case study of modeling adverbs as modifiers of computational verbs was presented. In [12] computational verb rules were used to improve the training processes of neural networks.

The theory of computational verb has been taught in some university classrooms since 2005[1]. The latest active applications of computational verb theory are listed as follows.

1) Computational Verb Controllers. The applications of computational verbs to different kinds of control problems were studied on different occassions[27], [28]. For the advanced applications of computational verbs to control problems, a few papers reporting the latest advances had been published[34], [33], [45], [39], [58]. The design of computational verb controllers was also presented in a textbook in 2005[1].
2) Computational Verb Image Processing and Image Understanding. The recent results of image processing by using computational verbs can be found in[35]. The applications of computational verbs to image understanding can be found in [37]. The authors of [2] applied computational verb image processing to design the vision systems of RoboCup small-size robots.
3) Stock Market Modeling and Prediction based on computational verbs. The product of Cognitive Stock Charts[6] was based on the advanced modeling and computing reported in [40]. Computational verb theory was used to study the trends of stock markets known as Russell reconstruction patterns [41].

Computational verb theory has been successfully applied to many industrial and commercial products. Some of these products are listed as follows.

1) Visual Card Counters. The *YangSky-MAGIC* card counter[8], developed by Yang's Scientific Research Institute and Wuxi Xingcard Technology Co. Ltd., was the first visual card counter to use computational verb image processing technology to achieve high accuracy of card and paper board counting based on cheap webcams.
2) CCTV Automatic Driver Qualify Test System. The *DriveQfy* CCTV automatic driver qualify test system[9] was the first vehicle trajectory reconstruction and stop time measuring system using computational verb image processing technology.
3) Visual Flame Detecting System. The *FireEye* visual flame detecting system[4] was the first CCTV or webcam based flame detecting system, which works under color and black & white conditions, for surveillance and security monitoring system.
4) Smart Pornographic Image and Video Detection Systems. The *PornSeer*[7] pornographic image and video detection systems are the first cognitive feature based smart porno detection and removal software.

[1]Dr. G. Chen, EE 64152 - Introduction to Fuzzy Informatics and Intelligent Systems, Department of Electronic Engineering, City University of Hong Kong. Dr. Mahir Sabra, EELE 6306: Intelligent Control, Electrical and Computer Engineering Department, The Islamic University of Gaza. Dr. D. H. Guo, Artificial Intelligence, Department of Electronic Engineering, Xiamen University. Prof. T. Yang, Computational Methodologies in Intelligent Systems, Department of Electronic Engineering, Xiamen University.

5) Webcam Barcode Scanner. The *BarSeer*[5] webcam barcode scanner took advantage of the computational verb image processing to make the scan of barcode by using cheap webcam possible.
6) Cognitive Stock Charts. By applying computational verbs to the modeling of trends and cognitive behaviors of stock trading activities, cognitive stock charts can provide the traders with the "feelings" of stock markets by using simple and intuitive indexes.
7) TrafGo ITS SDK. Computational verbs were applied to model vehicle trajectories and dynamics of optical field and many other aspects of dynamics in complex environments for applications in intelligent transportation systems (ITS).

III. Interactions Between Two Computational Verbs

Assume that the evolving systems of two computational verbs V_1 and V_2, are given by the following ordinary differential equations(ODEs), respectively.

$$\begin{aligned} \mathcal{E}_1 &: \dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\boldsymbol{x}_1), \\ \mathcal{E}_2 &: \dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{x}_2) \end{aligned} \quad (1)$$

where $\boldsymbol{x}_1 \in \mathbb{R}^m$ and $\boldsymbol{x}_2 \in \mathbb{R}^n$ are states of the two computational verbs, respectively. In general, $\boldsymbol{f}_1(\cdot)$ and $\boldsymbol{f}_2(\cdot)$ are two nonlinear functions. Observe that both ODEs are independent because no interaction is set up between two verbs. The evolving functions of V_1 and V_2 are outputs of their evolving systems.

Since the interactions between two computational verbs are rooted from the input-output structure of computational verbs, it is more convenient to study the interactions between computational verbs based on their evolving systems rather than their evolving functions. In many cases, since the interactions may result in the changes of parameters or even structures of computational verbs, to study the interaction based on evolving systems is the only feasible way.

The interactions between two computational verbs might have the following types.

1) Weak interaction. The interaction is weak if and only if the qualitative properties of all coupled verbs keep unchanged; namely, the interaction doesn't change the qualitative behaviors of both verbs. Since qualitative behaviors of a computational verb are mathematically modeled in the continuum of the phase-space configurations of the computational verb, to keep the same qualitative behavior of the computational verb is to keep its phase-space configurations within a given range. Topological equivalence is one necessary condition of keeping qualitative behaviors the same. Furthermore, to keep qualitative behaviors the same is to make them "feel" the same or similar. The measurement of "feeling the same" is dependent on the mathematical model of computational verb feel, which is one of the most difficult verbs to model.
2) Strong interaction. The interaction is strong if and only if the qualitative properties of at least one verb is changed; namely, the interaction changes the following aspects of at least one verb.
   - The number of equilibrium points of the computational verb.
   - The types of equilibrium points of the computational verb. For example, to change a node into a saddle.
   - The topological equivalence.
   - The invariant sets. For example, to change a node into a limit cycle; to change a limit cycle into a strange attractor.
3) Mutual interaction. The interaction between two verbs are bi-directional. Since the interaction between two verbs can only influence both verbs from their input ports, which change either the states and/or the parameters of verbs, a mutual interaction doesn't distinguish the cause-effect of the interaction.
4) One-directional interaction. The interaction is directed from the *master verb* to the *slave verb* and the master verb is not affected by the slave verb. The states of the master verb can change the states and/or the parameters of the slave verb. In this configuration, the master verb is the cause and the slave verb is the effect of the interaction.

Since the above-mentioned types of interactions between computational verbs are categorized from coupling strengthes and coupling directions, there are many other categories based on the combinations of coupling strengthes and directions. For example, an interaction can be weak and mutual or strong and one-directional.

A. Couplings Between Two Computational Verbs

When the interaction between two computational verbs doesn't change the structure and parameters of evolving systems, we usually call this kind of interaction as *coupling*. In engineering applications, coupling is usually injected from the input ports of an engineering system.

Example 1 (Two Coupled Pendulums): As shown in Fig. 1, two pendulums mutually coupled through a beam. The stiffness of the beam and the pole determines the strength of the the coupling. The coupling from one pendulum to the other is though the motions of the pole and the beam. In this case, the interaction between two dynamical systems didn't change either the structures or parameters of two pendulums. The couplings are fed from input ports.

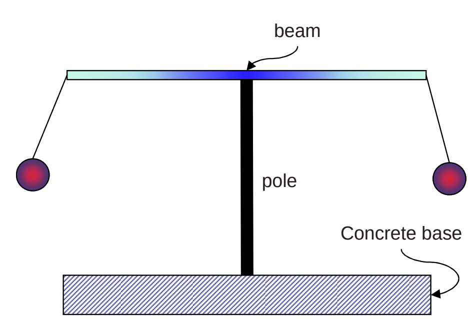

Fig. 1. Mutual coupling between two pendulums.

The interaction between the two computational verbs in Eq. (1) can be modeled as the following mutual coupled dynamical system

$$\begin{aligned}\mathcal{E}_1 &: \dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\boldsymbol{x}_1) + \boldsymbol{g}_1(\boldsymbol{x}_1, \boldsymbol{x}_2),\\ \mathcal{E}_2 &: \dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{x}_2) + \boldsymbol{g}_2(\boldsymbol{x}_1, \boldsymbol{x}_2)\end{aligned} \tag{2}$$

where $\boldsymbol{g}_1(\cdot,\cdot)$ and $\boldsymbol{g}_2(\cdot,\cdot)$ are two *coupling functions*. When two verbs coupled to each other, the coupling can have different effects on both verbs. Since in a one-directional coupling scheme, there is a master and a slave verb, we explicitly distinguish the difference between the master verb and the slave verb as follow.

$$\begin{aligned}\mathcal{E}_1 &: \dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\boldsymbol{x}_1),\\ \mathcal{E}_2 &: \dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{x}_2) + \boldsymbol{g}(\boldsymbol{x}_1, \boldsymbol{x}_2)\end{aligned} \tag{3}$$

where $\mathcal{E}_1$ and $\mathcal{E}_2$ are master and slave verbs, respectively. $\boldsymbol{g}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is the one-directional coupling function from the master verb to the slave verb. Observe that in this case, $\mathcal{E}_1$ is independent while $\mathcal{E}_2$ is driven by $\mathcal{E}_1$.

B. Coupling with Differences of States

The coupling functions in Eqs. (2) and (3) can be any functions. When $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are vectors of the same dimension, in many cases, we are more interested in a class of coupling functions, of which the variables are the difference between the states of computational verbs. It will be easier to study the differences between two computational verbs when we consider how near two computational verbs approach to each other after coupling to each other. In this case, the mutual coupling between two verbs can be explicitly represented as

$$\begin{aligned}\mathcal{E}_1 &: \dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\boldsymbol{x}_1) + \boldsymbol{g}_1(\boldsymbol{x}_1 - \boldsymbol{x}_2),\\ \mathcal{E}_2 &: \dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{x}_2) + \boldsymbol{g}_2(\boldsymbol{x}_1 - \boldsymbol{x}_2)\end{aligned} \tag{4}$$

and the one-directional case is given by

$$\begin{aligned}\mathcal{E}_1 &: \dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\boldsymbol{x}_1),\\ \mathcal{E}_2 &: \dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{x}_2) + \boldsymbol{g}(\boldsymbol{x}_1 - \boldsymbol{x}_2).\end{aligned} \tag{5}$$

C. Parameter Modulations Between Two Computational Verbs

When the interaction between two computational verbs changes the parameters of at least one computational verb, we call this kind of interaction as *parameter modulation*. Since the parameters must be adjustable in a parameter modulation, we need to explicitly represent the parameter vectors in the evolving systems of computational verbs as follows.

$$\begin{aligned}\mathcal{E}_1 \quad &: \quad \dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\boldsymbol{p}_1, \boldsymbol{x}_1),\\ &\quad\;\; \dot{\boldsymbol{p}}_1 = \boldsymbol{g}_1(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{h}_1(\boldsymbol{p}_1, \boldsymbol{p}_2),\\ \mathcal{E}_2 \quad &: \quad \dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{p}_2, \boldsymbol{x}_2),\\ &\quad\;\; \dot{\boldsymbol{p}}_2 = \boldsymbol{g}_2(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{h}_2(\boldsymbol{p}_1, \boldsymbol{p}_2)\end{aligned} \tag{6}$$

where $\boldsymbol{p}_1 \in \mathbb{R}^p$ and $\boldsymbol{p}_2 \in \mathbb{R}^q$ are two parameter vectors. In this case, the couplings are fed into the evolving systems by constantly changing the values of their parameters. Since the changes of their parameters can result in changes of structures of computational verbs, the parameter modulation can result in qualitative changes of dynamics. If the interaction is one-directional, then Eq. (6) becomes

$$\begin{aligned}\mathcal{E}_1 &: \dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\boldsymbol{p}_1, \boldsymbol{x}_1),\\ \mathcal{E}_2 &: \dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{p}_2, \boldsymbol{x}_2), \dot{\boldsymbol{p}}_2 = \boldsymbol{g}(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{h}(\boldsymbol{p}_1, \boldsymbol{p}_2).\end{aligned} \tag{7}$$

If the parameter modulation is based on the differences between the states of computational verbs, then for mutual interaction we have

$$\begin{aligned}\mathcal{E}_1 &: \dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\boldsymbol{p}_1, \boldsymbol{x}_1), \dot{\boldsymbol{p}}_1 = \boldsymbol{g}_1(\boldsymbol{x}_1 - \boldsymbol{x}_2) + \boldsymbol{h}_1(\boldsymbol{p}_1 - \boldsymbol{p}_2),\\ \mathcal{E}_2 &: \dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{p}_2, \boldsymbol{x}_2), \dot{\boldsymbol{p}}_2 = \boldsymbol{g}_2(\boldsymbol{x}_1 - \boldsymbol{x}_2) + \boldsymbol{h}_2(\boldsymbol{p}_1 - \boldsymbol{p}_2),\end{aligned} \tag{8}$$

and for one-directional interaction we have

$$\begin{aligned}\mathcal{E}_1 &: \dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\boldsymbol{p}_1, \boldsymbol{x}_1),\\ \mathcal{E}_2 &: \dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{p}_2, \boldsymbol{x}_2), \dot{\boldsymbol{p}}_2 = \boldsymbol{g}(\boldsymbol{x}_1 - \boldsymbol{x}_2) + \boldsymbol{h}(\boldsymbol{p}_1 - \boldsymbol{p}_2).\end{aligned} \tag{9}$$

The effects of interactions on the behaviors of computational verbs have a very wide spectrum, the author will focus on the most commonly known behaviors; namely, synchronization. Synchronization between two verbs is easier to study in the phase spaces of verbs. It is a process, in which two initially unrelated verbs become related to each other. When the uncorrelated initial states of verbs die out, we say the synchronization is achieved, and both verbs become closely correlated. Different correlation between verbs result in different types of synchronization.

IV. Identical Synchronization: Synonyms

If two verbs are synonyms, then their evolving systems are either identical or with the same structure. Two verb synonyms are at least topologically equivalent or topologically similar. Here we only study the cases when both of them are topologically equivalent. If $\mathcal{E}_1$ and $\mathcal{E}_2$ are identical, then in the case of coupling, Eqs. (4) and (5) become

$$\begin{aligned}\mathcal{E}_1 &: \dot{\boldsymbol{x}}_1 = \boldsymbol{f}(\boldsymbol{x}_1) + \boldsymbol{g}_1(\boldsymbol{x}_1 - \boldsymbol{x}_2),\\ \mathcal{E}_2 &: \dot{\boldsymbol{x}}_2 = \boldsymbol{f}(\boldsymbol{x}_2) + \boldsymbol{g}_2(\boldsymbol{x}_1 - \boldsymbol{x}_2)\end{aligned} \tag{10}$$

and

$$\begin{aligned}\mathcal{E}_1 &: \dot{\boldsymbol{x}}_1 = \boldsymbol{f}(\boldsymbol{x}_1),\\ \mathcal{E}_2 &: \dot{\boldsymbol{x}}_2 = \boldsymbol{f}(\boldsymbol{x}_2) + \boldsymbol{g}(\boldsymbol{x}_1 - \boldsymbol{x}_2),\end{aligned} \tag{11}$$

respectively. In the case of parameter modulation, Eqs. (8) and (9) become

$$\begin{aligned}\mathcal{E}_1 &: \dot{\boldsymbol{x}}_1 = \boldsymbol{f}(\boldsymbol{p}_1, \boldsymbol{x}_1), \dot{\boldsymbol{p}}_1 = \boldsymbol{g}_1(\boldsymbol{x}_1 - \boldsymbol{x}_2) + \boldsymbol{h}_1(\boldsymbol{p}_1),\\ \mathcal{E}_2 &: \dot{\boldsymbol{x}}_2 = \boldsymbol{f}(\boldsymbol{p}_2, \boldsymbol{x}_2), \dot{\boldsymbol{p}}_2 = \boldsymbol{g}_2(\boldsymbol{x}_1 - \boldsymbol{x}_2) + \boldsymbol{h}_2(\boldsymbol{p}_2)\end{aligned} \tag{12}$$

and

$$\begin{aligned}\mathcal{E}_1 &: \dot{\boldsymbol{x}}_1 = \boldsymbol{f}(\boldsymbol{p}_1, \boldsymbol{x}_1),\\ \mathcal{E}_2 &: \dot{\boldsymbol{x}}_2 = \boldsymbol{f}(\boldsymbol{p}_2, \boldsymbol{x}_2), \dot{\boldsymbol{p}}_2 = \boldsymbol{g}(\boldsymbol{x}_1 - \boldsymbol{x}_2) + \boldsymbol{h}(\boldsymbol{p}_1 - \boldsymbol{p}_2),\end{aligned} \tag{13}$$

respectively.

A. Analysis

Identical synchronization between two computational verbs can be achieved for different types of interactions. When an identical synchronization is achieved, both computational verbs act as one computational verb. Since two identically synchronized computational verbs have the same states, it is easier to study the dynamics of synchronization by using the difference in the states of both computational verbs. Let us define the *error vector* as $\boldsymbol{e}(t) = \boldsymbol{x}_1(t) - \boldsymbol{x}_2(t)$, when the identical synchronization is achieved, we have $\boldsymbol{e}(t) = \mathbf{0}$. Therefore, the stability of the identical synchronization between two computational verb is equivalent to the stability of the origin of the error system.

1) Coupling: For mutual coupling in Eq. (10), the error system is given by

$$\dot{\boldsymbol{e}} = \boldsymbol{f}(\boldsymbol{x}_1) - \boldsymbol{f}(\boldsymbol{x}_2) + \boldsymbol{g}_1(\boldsymbol{e}) - \boldsymbol{g}_2(\boldsymbol{e}). \tag{14}$$

For one-direction coupling in Eq. (11), the error system is given by

$$\dot{\boldsymbol{e}} = \boldsymbol{f}(\boldsymbol{x}_1) - \boldsymbol{f}(\boldsymbol{x}_2) - \boldsymbol{g}(\boldsymbol{e}). \tag{15}$$

Two coupled computational verbs can be identically synchronized to each other if and only if the origins of error systems (14) and (15) are asymptotically stable.

Example 2: The Chinese verb lai(来) has a meaning of come, if we set the origin of a reference frame as the destination of lai(来), then we can assume its evolving system as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \tag{16}$$

from which we have

$$\boldsymbol{f}(\boldsymbol{x}_1) - \boldsymbol{f}(\boldsymbol{x}_2) = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \boldsymbol{e}. \tag{17}$$

In Eqs. (14) and (15), let us assume

$$\boldsymbol{g}_1(\boldsymbol{e}) - \boldsymbol{g}_2(\boldsymbol{e}) = -\boldsymbol{g}(\boldsymbol{e}) = 0, \tag{18}$$

then the error systems for mutual coupling and one-directional coupling are the same and is given by

$$\dot{\boldsymbol{e}} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \boldsymbol{e}, \tag{19}$$

of which the origin is asymptotically stable. When two lai(来) couples to each other, they become identically synchronized no matter what initial difference between their states was. Because of this identical synchronization, in Chinese by repeating lai(来), we can get a new verb lailai(来来), which has the same trajectories as that of lai(来) when the trajectories are near enough to the destination of lai(来). However, since there are two dynamical systems in verb lailai(来来), it can have more than two sets of initial conditions; namely, it can represent the meaning of "coming from different locations at the same time" while lai(来) can only represent the meaning of "coming from one location". Therefore, lailai(来来) can represent a group of people lai(来) at the same time.

Example 3: The Chinese verb wang(往) has a meaning of go away, if we set the origin of a reference frame as the starting point of wang(往), then we can assume its evolving system as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \tag{20}$$

from which we have

$$\boldsymbol{f}(\boldsymbol{x}_1) - \boldsymbol{f}(\boldsymbol{x}_2) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{e}. \tag{21}$$

In Eqs. (14) and (15), let us assume

$$\boldsymbol{g}_1(\boldsymbol{e}) - \boldsymbol{g}_2(\boldsymbol{e}) = -\boldsymbol{g}(\boldsymbol{e}) = \begin{pmatrix} -2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{pmatrix} \boldsymbol{e}, \tag{22}$$

then the error systems for mutual coupling and one-directional coupling are the same and is given by

$$\dot{\boldsymbol{e}} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \boldsymbol{e}, \tag{23}$$

of which the origin is asymptotically stable. When two wang(往) couples to each other, they become identically synchronized no matter the distance between their initial locations. Because of this identical synchronization, in Chinese by repeating wang(往), we construct a new verb wangwang(往往), which has the same trajectories as that of wang(往) when the trajectories leave the origin far enough and the initial difference between them dies out. However, since there are two dynamical systems in verb wangwang(往往), it can have more than two sets of initial conditions; namely, it can represent the meaning of "go from different locations at the same time" while wang(往) can only represent the meaning of "go from one location". Therefore, wangwang(往往) can represent a group of people wang(往) at the same time.

2) Parameter Modulation: To achieve identical synchronization, the parameter vectors of both computational verbs approach to each other while the synchronization errors approach zero. It follows from Eqs. (12) that the error systems for states and parameters are given by

$$\begin{aligned} \dot{\boldsymbol{e}} &= \boldsymbol{f}(\boldsymbol{p}_1, \boldsymbol{x}_1) - \boldsymbol{f}(\boldsymbol{p}_2, \boldsymbol{x}_2), \\ \dot{\boldsymbol{e}}_p &= \boldsymbol{g}_1(\boldsymbol{e}) - \boldsymbol{g}_2(\boldsymbol{e}) + \boldsymbol{h}_1(\boldsymbol{p}_1) - \boldsymbol{h}_2(\boldsymbol{p}_2) \end{aligned} \tag{24}$$

where $\boldsymbol{e}_p = \boldsymbol{p}_1 - \boldsymbol{p}_2$ is the error vector of parameters.

It follows from Eqs. (13) that the error systems for states and parameters are given by

$$\begin{aligned} \dot{\boldsymbol{e}} &= \boldsymbol{f}(\boldsymbol{p}_1, \boldsymbol{x}_1) - \boldsymbol{f}(\boldsymbol{p}_2, \boldsymbol{x}_2), \\ \dot{\boldsymbol{e}}_p &= -\boldsymbol{g}(\boldsymbol{e}) - \boldsymbol{h}(\boldsymbol{e}_p). \end{aligned} \tag{25}$$

Both computational verbs are identically synchronized if and only if error systems Eqs. (24) and (25) are asymptotically stable at the origin of $\widetilde{\boldsymbol{e}} \triangleq (\boldsymbol{e}^\top \ \boldsymbol{e}_p^\top)^\top$.

Example 4: We add a parameter vector into Eq. (16) as follow

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} - \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix}, \quad (26)$$

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x}_1) - \boldsymbol{f}(\boldsymbol{x}_2) &= \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \boldsymbol{e} \\ &+ \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \boldsymbol{e}_p. \end{aligned} \quad (27)$$

In Eqs. (24) and (25), let us assume

$$\begin{aligned} &\boldsymbol{g}_1(\boldsymbol{e}) - \boldsymbol{g}_2(\boldsymbol{e}) = -\boldsymbol{g}(\boldsymbol{e}) = 0, \\ &\boldsymbol{h}_1(\boldsymbol{p}_1) - \boldsymbol{h}_2(\boldsymbol{p}_2) = -\boldsymbol{h}(\boldsymbol{e}_p) = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \boldsymbol{e}_p, \end{aligned} \quad (28)$$

then the error systems for mutual coupling and one-directional coupling are the same and is given by

$$\dot{\widetilde{\boldsymbol{e}}} = \begin{pmatrix} -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \widetilde{\boldsymbol{e}}, \quad (29)$$

of which the origin is asymptotically stable.

B. Linguistic Implications

When two computational verbs achieve identical synchronization, both computational verbs can be viewed to have the same meaning while the difference in their initial conditions is the only attribute to distinguish both computational verbs. Two identically synchronized computational verbs form a new computational verb in Chinese by enclosing two identical dynamical systems into one. The identically synchronized computational verbs provide us with a pair of identical trajectories to make the dynamics more collective and to make the dynamics to have more variations at the same time. By doing so, we can represent the collection of actions, which are the same type and happen at the same time period, by repeating the verb. For example, when we say lai(来), we mean only one person come here and when we repeat it as lailai(来来), we mean many people come here.

Since the identical synchronization between two computational verbs makes two computational verbs to function as one, to repeat two verbs can not intensify the dynamics. Therefore, the purpose of repeating two verbs in the identical synchronization is to show the diversity of the initial conditions of verbs; namely, to show the collective actions of the same verb. From this point of view, we can conjugate that to repeat two synchronized verbs in identical synchronization is to show that

- many people take the same action simultaneously;
- the same action takes place many times;
- different agents go thought the same process at the same time;
- the same process is repeated many times.

V. Reverse Synchronization: Antonyms

When a verb synchronized with its antonym, the process is called *reverse synchronization* because the identical synchronization can only be achieved when the time in the antonym reversed. Let's assume that the verb and its antonym as

$$\begin{aligned} \mathcal{E}_1 &: \dot{\boldsymbol{x}}_1(t) = \boldsymbol{f}_1(\boldsymbol{x}_1(t)), \\ \mathcal{E}_2 &: \dot{\boldsymbol{x}}_2(t) = \boldsymbol{f}_2(\boldsymbol{x}_2(t)). \end{aligned} \quad (30)$$

Let us reverse the time in $\mathcal{E}_2$ and form a new verb $\overline{\mathcal{E}}_2$. Since $\mathcal{E}_2$ is an antonym of $\mathcal{E}_1$, $\overline{\mathcal{E}}_2$ is an synonym of $\mathcal{E}_1$. $\overline{\mathcal{E}}_2$ is given by

$$\overline{\mathcal{E}}_2 : \dot{\boldsymbol{x}}_2(-t) = -\boldsymbol{f}_2(\boldsymbol{x}_2(-t)). \quad (31)$$

Therefore, it follows from the results in Sec. IV that $\boldsymbol{f}_1(\cdot) = -\boldsymbol{f}_2(\cdot)$ can guarantee the identical synchronization between $\mathcal{E}_1$ and $\overline{\mathcal{E}}_2$ by applying different interactions between them.

Example 5: Let the evolving system of lai(来) be the same as that in Example 2, and the evolving system of wang(往) be the same as that in Example 3, then the identical synchronization between lai(来) and reverse of wang(往) form a new verb laiwang(来往).

The linguistic implications of reverse synchronization between computational verbs are listed as follows.

- To display two reversible actions at the same time. In Chinese, the verb laiwang(来往) has the sense of embedding the actions of "come and go" simultaneously. However, since laiwang(来往) contains time of two reversed directions, the only possible way to make them synchronized is to eliminate the difference between directions of time. Or, to apply an absolute value to time in both verbs. This might result in reducing the importance of the role that time plays in both verbs when they reversely synchronized into a new verb. In other words, the dynamics of the individual verbs might be sacrificed when they are reversely synchronized.
- To make simple process complex. By reversely synchronize two verbs, the boundary between both verbs is blurred and the trajectories of both verbs mix to each other and to make it more complex, the time directions of these trajectories become blurred as well. This is because it is impossible to distinguish two synchronized verbs. Therefore, even if the process represented by each individual verb is clear and simple, the process represented by the reversely synchronized verb pair can be much more complex than individual verbs.
- To make representations of dynamics less consistent. Since a reversely synchronized verb pair can start from the same initial condition and go to two different directions along time, this introduces inconsistency in the new verb.
- To make representations of dynamics uncertain. Since each trajectory in the reversely synchronized verb pair can go backward and forward at the same time, extra

uncertainties are brought in even if the individual verbs are simple and crisp.

Therefore, when a verb and its antonym are reversely synchronized, different kinds of linguistic phenomena can arise from the reverse synchronization and therefore can result in a rich pool of linguistic phenomena. The examples of this kind of richness in Chinese will be reported in other publications.

VI. Linear Generalized Synchronization

When two verbs interact to each other in such a way that the states of one verb related to those of the other with a linear transformation, then both verbs are in linear generalized synchronization(GS). Linear GS is the simplest form of GS between two dynamical systems. The simplest linear generalized synchronization is to scale down an attractor by a scalar factor and keep both verb synchronized. Let us use an example to show how this can be achieved.

Example 6: This example was first reported in [54]. The evolving systems of two computational verbs are modeled as the following two Lorenz systems.

$$\mathcal{E}_1 \quad : \quad \begin{cases} \dot{x} = -\sigma x + \sigma y, \\ \dot{y} = rx - y - xz, \\ \dot{z} = xy - bz, \end{cases}$$
$$\mathcal{E}_2 \quad : \quad \begin{cases} \dot{\tilde{x}} = -\sigma\tilde{x} + \sigma\tilde{y}, \\ \dot{\tilde{y}} = \lambda(r-\mu)x + \mu\tilde{x} - \tilde{y} - \lambda xz, \\ \dot{\tilde{z}} = \lambda xy - b\tilde{z}, \end{cases} \tag{32}$$

where σ, r, b are constant parameters. Observe that there is a one-directional coupling from $\mathcal{E}_1$ to $\mathcal{E}_2$. Based on the conditions provided in [54], a scaled generalized synchronization can be achieved between $\mathcal{E}_1$ and $\mathcal{E}_2$

The simulation results with $\lambda = 0.5$ are shown in Fig. 2. Figure 2(a) shows the attractor of $\mathcal{E}_1$. Figure 2(b) shows the attractor of $\mathcal{E}_2$. Observe that the attractor of $\mathcal{E}_2$ is a down-scaled version of that of $\mathcal{E}_1$ by a scaling factor $\lambda = 0.5$. Figures 2(c), (d) and (e) show the relations of x versus $\tilde{x}$, y versus $\tilde{y}$ and z versus $\tilde{z}$, respectively.

When $\mathcal{E}_1$ controls $\mathcal{E}_2$ into a down-scaled replication of itself, the interaction between $\mathcal{E}_1$ and $\mathcal{E}_2$ results in the loss of dynamical range of $\mathcal{E}_2$; namely, $\mathcal{E}_1$ changes $\mathcal{E}_2$ from an "active" verb into a "less active" verb. If we decrease the scale factor λ to a value such that the range of the attractor of $\mathcal{E}_2$ becomes so small that is comparable to the lowest resolution of human perceptions, then $\mathcal{E}_2$ loses its property of verb. In Chinese, this interaction can change $\mathcal{E}_2$ into a preposition.

A more general linear GS between two verbs results in linear distortions in the attractors of the verbs shown in the following example[56], [54].

Example 7: In this example, the evolving systems of two computational verbs are modeled as the following two Lorenz systems.

$$\mathcal{E}_1 \quad : \quad \begin{cases} \dot{x} = -\sigma x + \sigma y, \\ \dot{y} = rx - y - xz, \\ \dot{z} = xy - bz, \end{cases}$$
$$\mathcal{E}_2 \quad : \quad \begin{cases} \dot{\tilde{x}} = -\sigma\tilde{x} + \sigma\tilde{y} + \sigma x(r-\mu-z), \\ \dot{\tilde{y}} = \mu\tilde{x} - \tilde{y} - x(r-\mu-z), \\ \dot{\tilde{z}} = -b\tilde{z} - bxy. \end{cases} \tag{33}$$

If the linear GS between $\mathcal{E}_1$ and $\mathcal{E}_2$ is achieved, the following relations should be satisfied

$$\begin{aligned} \tilde{x} &= -\sigma x + \sigma y \triangleq f_1(x,y), \\ \tilde{y} &= \mu x - y \triangleq f_2(x,y), \\ \tilde{z} &= -bz. \end{aligned} \tag{34}$$

The simulation results are shown in Fig. 3. Figure 3(a) shows the attractor of $\mathcal{E}_2$. Observe that this attractor is completely different to the famous "butterfly" attractor as shown in Fig. 2(a) though the former is only a linearly transformed version of the latter. Figures 3(b) and (c) show the plots of $\tilde{x}$ versus $f_1(x,y)$ and $\tilde{y}$ versus $f_2(x,y)$, respectively. Observe that the linear verb similarity transformation is true. Figure 3(d) shows the plot of $\tilde{z}$ versus z. We show the $\tilde{x}$ versus x plot and $\tilde{y}$ versus y plot in Figs. 3(e) and 3(f), respectively.

Remark 1: When a master verb drives a slave verb into a generalized synchronization, the following scenarios can happen.

- The master verb suppresses the ranges of the slave verb into a smaller region comparing that for a uncoupled slave verb. In the cases of different scaling factors, which change the ranges of the slave verb, the slave verb might lose its dynamical activities to different degree. In many cases, the slave verb might function less and less like a verb and finally result in a preposition. This phenomena were widely found in Chinese.
- The master verb can also dramatically enhance the dynamical activities of the slave verb. In this case, the master verb functions more like an adverb rather than a verb. The result can be most likely to change the master verb into an adverb. Or, on the other hand, if both master verb and the slave verb are synonyms before the GS happens, it is also very possible that the GS will turn the range of the slave verb too big such that the master verb become less significant, in this sense, the master verb can be less "active" and might even become a preposition.
- Both master verb and slave verb can keep their ranges of attractor comparable even after the GS is achieved. In this case, both verbs are most likely to keep their own properties and the synchronized verbs become a compound verb as a whole to have a new level of meaning.

VII. Merge More Than One Computational Verb into One

The interaction between two or more computational verbs can merge these computational verbs into one. To merge two computational verbs into one is to mix up the phase spaces of two dynamical systems, there are many ways to do so. Some scenarios, which constitute only a portion of all possible cases, are listed as follows.

- Two computational verbs have the same kind of phase space, the merging result keeps all equilibrium points of both computational verbs.
- Two computational verbs have the same kind of phase space, the merging result eliminates some equilibrium points of one or both computational verbs.

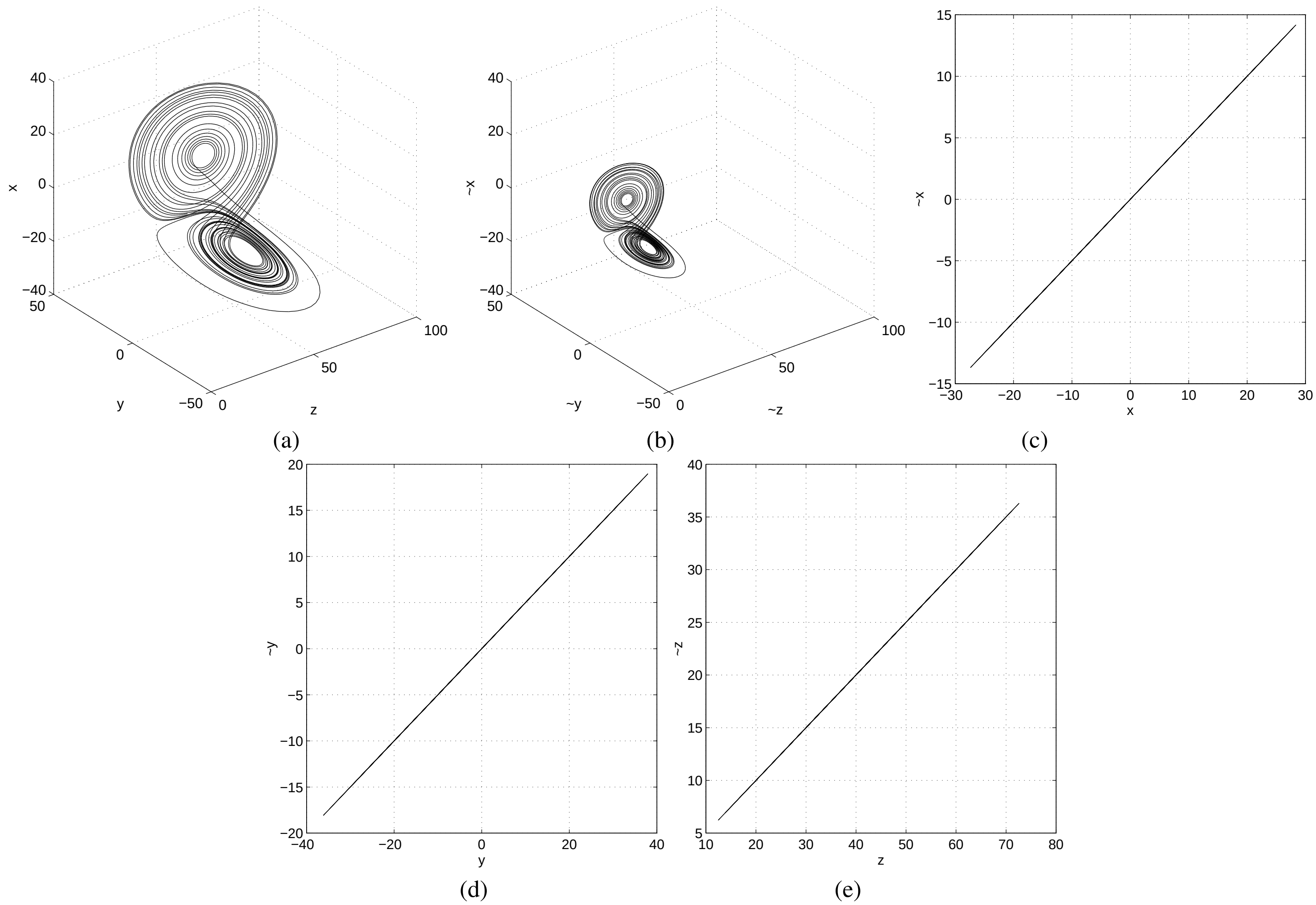

Fig. 2. Scaled GS of two Lorenz systems for modeling verb similarity. (a) The attractor of $\mathcal{E}_1$. (b) The attractor of $\mathcal{E}_2$. (c) x versus $\tilde{x}$ plot. (d) y versus $\tilde{y}$ plot. (e) z versus $\tilde{z}$ plot.

- Two computational verbs have the same kind of phase space, the merging result gives birth to new equilibrium points that are not previously existing in both computational verbs.
- Two computational verbs have different types of phase spaces, the merging result constructs a new phase space, which contains the phase spaces of both verbs as its subspaces. The number of equilibrium points of the merging result can be any cases.
- Two computational verbs have different types of phase spaces, the merging result constructs a new phase space, which contains partially the phase space of one and the entire phase space of the other verb as its subspaces, or partially the phase spaces of both verbs as its subspaces.

A. Keep All Equilibrium Points

Assume that the two computational verbs in Eq. (1) have the same phase space and are coupled in such a way that all their equilibrium points are kept in the new compound verb V, of which the evolving function is given by

$$\mathcal{E} : \dot{\boldsymbol{x}} = \boldsymbol{g}_1(\boldsymbol{f}_1(\boldsymbol{x}))\boldsymbol{g}_2(\boldsymbol{f}_2(\boldsymbol{x})) \tag{35}$$

where $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are smooth in the vicinity of equilibrium points and satisfy

$$\boldsymbol{g}_1(\boldsymbol{0}) = \boldsymbol{0}, \boldsymbol{g}_2(\boldsymbol{0}) = \boldsymbol{0}. \tag{36}$$

It is easy to see that all solution of $\boldsymbol{f}_1(\boldsymbol{x}) = \boldsymbol{0}$ and $\boldsymbol{f}_2(\boldsymbol{x}) = \boldsymbol{0}$ are the equilibrium points of evolving system (35). However, the stability of each equilibrium point may change in the compound computational verb. Observe that functions $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ may introduce extra nonlinearity into the compound computational verb and result in more complex behaviors that can't observe from the individual computational verbs. The linguistic implication of this phenomenon is that by merging two simple verbs we can construct a more "complex verb". For example, we can merging two simple verbs into a chaotic verb.

Example 8: Let the two computational verbs be

$$\begin{aligned} \mathcal{E}_1 &: \dot{\boldsymbol{x}} = \begin{pmatrix} x_1 - 1 & 0 & 0 \\ 0 & x_2 - 1 & 0 \\ 0 & 0 & x_3 - 1 \end{pmatrix}, \\ \mathcal{E}_2 &: \dot{\boldsymbol{x}} = \begin{pmatrix} x_1 + 1 & 0 & 0 \\ 0 & x_2 + 1 & 0 \\ 0 & 0 & x_3 + 1 \end{pmatrix}, \end{aligned} \tag{37}$$

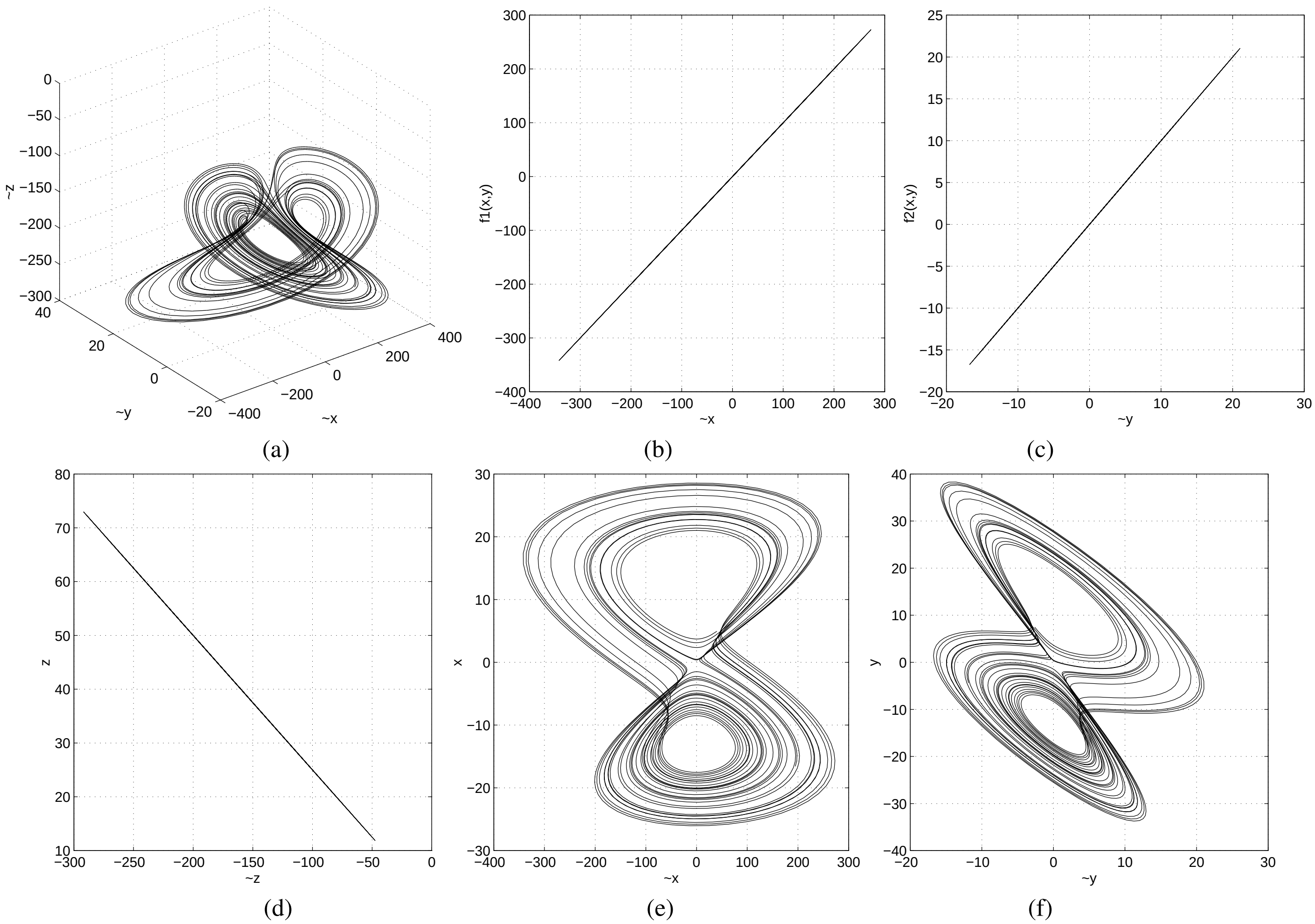

Fig. 3. Linear generalized synchronization of two Lorenz systems for modeling verb similarity with linear transformation. (a) The attractor of $\mathcal{E}_2$. (b) $\tilde{x}$ versus $f_1(x, y)$ plot. (c) $\tilde{y}$ versus $f_2(x, y)$ plot. (d) $\tilde{z}$ versus z plot. (e) $\tilde{x}$ versus x plot. (f) $\tilde{y}$ versus y plot.

then the following merged computational verbs keep the all equilibrium points of $\mathcal{E}_1$ and $\mathcal{E}_2$

$$\begin{aligned}\mathcal{E} \quad &: \quad \dot{\boldsymbol{x}} = \begin{pmatrix} x_1 - 1 & 0 & 0 \\ 0 & x_2 - 1 & 0 \\ 0 & 0 & x_3 - 1 \end{pmatrix} \\ &\qquad \begin{pmatrix} x_1 + 1 & 0 & 0 \\ 0 & x_2 + 1 & 0 \\ 0 & 0 & x_3 + 1 \end{pmatrix}. \\ \mathcal{E} \quad &: \quad \dot{\boldsymbol{x}} = \begin{pmatrix} (x_1 - 1)^2 & 0 & 0 \\ 0 & \tan^{-1}(x_2 - 1) & 0 \\ 0 & 0 & \tan(x_3 - 1) \end{pmatrix} \\ &\qquad \begin{pmatrix} x_1 + 1 & 0 & 0 \\ 0 & (x_2 + 1)^3 & 0 \\ 0 & 0 & \sin(x_3 + 1) \end{pmatrix}. \end{aligned} \tag{38}$$

B. Eliminate Some Equilibrium Points

In this case, the merge of two computational verbs partially keeps the equilibrium points of both verbs. One way of implementing such scenario is to choose $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ in Eq. (35) such that at least one of them doesn't satisfy the condition Eq. (36).

Example 9: Let the two computational verbs be

$$\begin{aligned}\mathcal{E}_1 \quad &: \quad \dot{\boldsymbol{x}} = \begin{pmatrix} (x_1 - 1)^3 & 0 & 0 \\ 0 & x_2 & 0 \\ 0 & 0 & x_3 \end{pmatrix}, \\ \mathcal{E}_2 \quad &: \quad \dot{\boldsymbol{x}} = \begin{pmatrix} x_1 + 1 & 0 & 0 \\ 0 & x_2 + 1 & 0 \\ 0 & 0 & x_3 + 1 \end{pmatrix}. \end{aligned} \tag{39}$$

Choose $\boldsymbol{g}_1$ such that

$$\begin{aligned}\mathcal{E} \quad &: \quad \dot{\boldsymbol{x}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} (x_1 - 1)^3 & 0 & 0 \\ 0 & x_2 & 0 \\ 0 & 0 & x_3 \end{pmatrix} \\ &\qquad \begin{pmatrix} x_1 + 1 & 0 & 0 \\ 0 & x_2 + 1 & 0 \\ 0 & 0 & x_3 + 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & x_2 & 0 \\ 0 & 0 & x_3 \end{pmatrix} \begin{pmatrix} x_1 + 1 & 0 & 0 \\ 0 & x_2 + 1 & 0 \\ 0 & 0 & x_3 + 1 \end{pmatrix}. \end{aligned} \tag{40}$$

Observe that the equilibrium points at $(1\ 0\ 0)^\top$ for the first verb were eliminated.

The other scenarios can be constructed based on the similar ways. More studies of merging verbs in Chinese will be presented in further papers.

VIII. Split a Computational Verb into Two or More

The interaction between two computational verbs can also result in the split of one computational verb into two or more computational verbs in many ways, of which a partial list is given as follows.

- Under the influence of the interaction between two verbs, the equilibrium points of one verb are regrouped into two or more groups, of which each becomes a new verb.
- Under the influence of the interaction between two verbs, the equilibrium points of two verbs are regrouped into three or more groups, of which each becomes a new verb.

Also, as a reverse process of merging two verbs, the split of a verb into two or more can also happen without explicit interaction between verbs. However, since in the evolution of any natural language, it is impossible to change the property of a verb in a very short time, we expect to see the interaction between the two sub-verbs. And this kind of interaction between sub-verbs are more likely to be repelling than attractive such that the sub-verbs evolving into independent verbs along the evolution of natural languages.

One example of splitting verb is to reconstruct the following two verbs from the verb in Eq. (35)

$$\begin{aligned} \mathcal{E}_1 &: \dot{\boldsymbol{x}} = \boldsymbol{f}_1(\boldsymbol{x}), \\ \mathcal{E}_2 &: \dot{\boldsymbol{x}} = \boldsymbol{f}_2(\boldsymbol{x}). \end{aligned} \tag{41}$$

Therefore, the problem of splitting a verb into two is to split a dynamical system into two. Since a dynamical system can be split into different combinations of sub-dynamical systems, one might wonder what kind of way of splitting a verb is the right way. To find the right way, we must consider the meaning of the sub-verbs. For example, if one verb was split into a node verb and a focus verb, then we must regroup the equilibrium points of the verb into a set of nodes and a set of focuses. Since each scenario is context-dependent, the study of split of verb will be most likely case by case.

IX. Concluding Remarks

The interaction between computational verbs reveals a very wide spectrum of phenomena, amount which the synchronization between computational verbs were studied in details in this paper. The purpose of this paper is to provide a platform, which is as universal as possible, for modeling different types of interactions between computational verbs. The future work will take advantage of the predicting ability provided by the mathematical models proposed in this paper to inspect different linguistic materials in a much more crisp and mathematical way. For example, the evolution of Chinese propositions will be studied based on the results presented here.

References

[1] Guanrong Chen and Trung Tat Pham. *Introduction to Fuzzy Systems*. Chapman & Hall/CRC, November 2005. ISBN:1-58488-531-9.

[2] Wanmi Chen, Yanqin Wei, Minrui Fei, and Huosheng Hu. Applications of computational verbs to image processing of RoboCup small-size robots. In *Intelligent Control and Automation*, volume 344/2006 of *Lecture Notes in Control and Information Sciences*, pages 494–499. Springer, Berlin / Heidelberg, 2006.

[3] Yi Guo. A study of adverbs as modifiers of computational verbs. *International Journal of Computational Cognition*, 6(1):31–35, March 2008 [available online at $http://www.YangSky.com/ijcc/ijcc61.htm$, $http://www.YangSky.us/ijcc/ijcc61.htm$].

[4] Yang's Scientific Research Institute LLC. **FireEye** *Visual Flame Detecting Systems*. http://www.yangsky.us/products/flamesky/index.htm, http://www.yangsky.com/products/flamesky/index.htm, 2005.

[5] Yang's Scientific Research Institute LLC. **BarSeer** *Webcam Barcode Scanner*. http://www.yangsky.us/demos/barseer/barseer.htm, http://www.yangsky.com/demos/barseer/barseer.htm, 2006.

[6] Yang's Scientific Research Institute LLC. *Cognitive Stock Charts*. http://www.yangsky.us/products/stock/, http://www.yangsky.com/products/stock/, 2006.

[7] Yang's Scientific Research Institute LLC. **PornSeer** *Pornographic Image and Video Detection Systems*. http://www.yangsky.us/products/dshowseer/porndetection/PornSeePro.htm, http://www.yangsky.com/products/dshowseer/porndetection/PornSeePro.htm, 2006.

[8] Yang's Scientific Research Institute LLC. and Wuxi Xingcard Technology Ltd. **YangSky-MAGIC** *Visual Card Counters*. http://www.yangsky.us/products/cardsky/cardsky.htm, http://www.yangsky.com/products/cardsky/cardsky.htm, 2004.

[9] Yang's Scientific Research Institute LLC. and Chinese Traffic Management Research Institute of the Ministry of Public Security(TMRI-China). **DriveQfy** *Automatic CCTV Driver Qualify Testing Systems*. http://www.yangsky.us/products/driveqfy/driveqfy.htm, http://www.yangsky.com/products/driveqfy/driveqfy.htm, 2005.

[10] R. Tonelli and T. Yang. Synchronizing Hénon maps using computational verb controllers. *Phys. Rev. E.*, 2007. submitted.

[11] R. Tonelli and T. Yang. Controlling Chua's circuits using computational verb controllers. *International Journal of Robust and Nonlinear Control*, 18(17):1622–1636, Nov. 25 2008.

[12] H.-B. Wang and T. Yang. Training neural networks using computational verb rules. *International Journal of Computational Cognition*, 6(2):17–32, June 2008 [available online at $http://www.YangSky.us/ijcc/ijcc62.htm$, $http://www.YangSky.com/ijcc/ijcc62.htm$].

[13] T. Yang. Verbal paradigms—Part I: Modeling with verbs. Technical Report Memorandum No. UCB/ERL M97/64, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA 94720, 9 Sept. 1997. page 1-15.

[14] T. Yang. Verbal paradigms—Part II: Computing with verbs. Technical Report Memorandum No. UCB/ERL M97/66, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA 94720, 18 Sept. 1997. page 1-27.

[15] T. Yang. Computational verb systems: Computing with verbs and applications. *International Journal of General Systems*, 28(1):1–36, 1999.

[16] T. Yang. Computational verb systems: Adverbs and adverbials as modifiers of verbs. *Information Sciences*, 121(1-2):39–60, Dec. 1999.

[17] T. Yang. Computational verb systems: Modeling with verbs and applications. *Information Sciences*, 117(3-4):147–175, Aug. 1999.

[18] T. Yang. Computational verb systems: Verb logic. *International Journal of Intelligent Systems*, 14(11):1071–1087, Nov. 1999.

[19] T. Yang. Computational verb systems: A new paradigm for artificial intelligence. *Information Sciences—An International Journal*, 124(1-4):103–123, 2000.

[20] T. Yang. Computational verb systems: Verb predictions and their applications. *International Journal of Intelligent Systems*, 15(11):1087–1102, Nov. 2000.

[21] T. Yang. Computational verb systems: Verb sets. *International Journal of General Systems*, 20(6):941–964, 2000.

[22] T. Yang. Computational verb systems: Towards a unified paradigm for artificial languages. In *Proceedings of the Fifth Joint Conference on Information Sciences(JCIS 2000)*, pages 29–32, Atlantic City, NJ, Feb. 27-Mar. 03 2000.

[23] T. Yang. *Advances in Computational Verb Systems*. Nova Science Publishers, Inc., Huntington, NY, May 2001. ISBN 1-56072-971-6.

[24] T. Yang. Computational verb systems: Computing with perceptions of dynamics. *Information Sciences*, 134(1-4):167–248, Jun. 2001.

[25] T. Yang. Computational verb systems: The paradox of the liar. *International Journal of Intelligent Systems*, 16(9):1053–1067, Sept. 2001.

[26] T. Yang. Computational verb systems: Verb numbers. *International Journal of Intelligent Systems*, 16(5):655–678, May 2001.

[27] T. Yang. *Impulsive Control Theory*, volume 272 of *Lecture Notes in Control and Information Sciences*. Spinger-Verlag, Berlin, Aug. 2001. ISBN 354042296X.

[28] T. Yang. *Computational Verb Theory: From Engineering, Dynamic Systems to Physical Linguistics*, volume 2 of *YangSky.com Monographs in Information Sciences*. Yang's Scientific Research Institute, Tucson, AZ, Oct. 2002. ISBN:0-9721212-1-8.

[29] T. Yang. Computational verb systems: Verbs and dynamic systems. *International Journal of Computational Cognition*, 1(3):1–50, Sept. 2003.

[30] T. Yang. *Fuzzy Dynamic Systems and Computational Verbs Represented by Fuzzy Mathematics*, volume 3 of *YangSky.com Monographs in Information Sciences*. Yang's Scientific Press, Tucson, AZ, Sept. 2003. ISBN:0-9721212-2-6.

[31] T. Yang. *Physical Linguistics: Measurable Linguistics and Duality Between Universe and Cognition*, volume 5 of *YangSky.com Monographs in Information Sciences*. Yang's Scientific Press, Tucson, AZ, Dec. 2004.

[32] T. Yang. Simulating human cognition using computational verb theory. *Journal of Shanghai University(Natural Sciences)*, 10(s):133–142, Oct. 2004.

[33] T. Yang. Architectures of computational verb controllers: Towards a new paradigm of intelligent control. *International Journal of Computational Cognition*, 3(2):74–101, June 2005 [available online at http://www.YangSky.com/ijcc/ijcc32.htm, http://www.YangSky.us/ijcc/ijcc32.htm].

[34] T. Yang. Applications of computational verbs to the design of P-controllers. *International Journal of Computational Cognition*, 3(2):52–60, June 2005 [available online at http://www.YangSky.us/ijcc/ijcc32.htm, http://www.YangSky.com/ijcc/ijcc32.htm].

[35] T. Yang. Applications of computational verbs to digital image processing. *International Journal of Computational Cognition*, 3(3):31–40, September 2005 [available online at http://www.YangSky.us/ijcc/ijcc33.htm, http://www.YangSky.com/ijcc/ijcc33.htm].

[36] T. Yang. Bridging the Universe and the Cognition. *International Journal of Computational Cognition*, 3(4):1–15, December 2005 [available online at http://www.YangSky.us/ijcc/ijcc34.htm, http://www.YangSky.com/ijcc/ijcc34.htm].

[37] T. Yang. Applications of computational verbs to effective and realtime image understanding. *International Journal of Computational Cognition*, 4(1):49–67, March 2006 [available online at http://www.YangSky.com/ijcc/ijcc41.htm, http://www.YangSky.us/ijcc/ijcc41.htm].

[38] T. Yang. Applications of computational verbs to feeling retrieval from texts. *International Journal of Computational Cognition*, 4(3):28–45, September 2006 [available online at http://www.YangSky.com/ijcc/ijcc43.htm, http://www.YangSky.us/ijcc/ijcc43.htm].

[39] T. Yang. Rule-wise linear computational verb systems: Dynamics and control. *International Journal of Computational Cognition*, 4(4):18–33, December 2006 [available online at http://www.YangSky.com/ijcc/ijcc44.htm, http://www.YangSky.us/ijcc/ijcc44.htm].

[40] T. Yang. Applications of computational verbs to cognitive models of stock markets. *International Journal of Computational Cognition*, 4(2):1–13, June 2006 [available online at http://www.YangSky.us/ijcc/ijcc42.htm, http://www.YangSky.com/ijcc/ijcc42.htm].

[41] T. Yang. Applications of computational verbs to the study of the effects of Russell's annual index reconstitution on stock markets. *International Journal of Computational Cognition*, 4(3):1–8, September 2006 [available online at http://www.YangSky.us/ijcc/ijcc43.htm, http://www.YangSky.com/ijcc/ijcc43.htm].

[42] T. Yang. Bridging computational verbs and fuzzy membership functions using computational verb collapses. *International Journal of Computational Cognition*, 4(4):47–61, December 2006 [available online at http://www.YangSky.us/ijcc/ijcc44.htm, http://www.YangSky.com/ijcc/ijcc44.htm].

[43] T. Yang. Computational verb decision trees. *International Journal of Computational Cognition*, 4(4):34–46, December 2006 [available online at http://www.YangSky.us/ijcc/ijcc44.htm, http://www.YangSky.com/ijcc/ijcc44.htm].

[44] T. Yang. Distances and similarities of saturated computational verbs. *International Journal of Computational Cognition*, 4(4):62–77, December 2006 [available online at http://www.YangSky.us/ijcc/ijcc44.htm, http://www.YangSky.com/ijcc/ijcc44.htm].

[45] T. Yang. Stable computational verb controllers. *International Journal of Computational Cognition*, 4(4):9–17, December 2006 [available online at http://www.YangSky.us/ijcc/ijcc44.htm, http://www.YangSky.com/ijcc/ijcc44.htm].

[46] T. Yang. Using computational verbs to cluster trajectories and curves. *International Journal of Computational Cognition*, 4(4):78–87, December 2006 [available online at http://www.YangSky.us/ijcc/ijcc44.htm, http://www.YangSky.com/ijcc/ijcc44.htm].

[47] T. Yang. Accurate video flame-detecting system based on computational verb theory. AS *Installer*, (42):154–157, August 2007. (in Chinese).

[48] T. Yang. *The Mathematical Principles of Natural Languages: The First Course in Physical Linguistics*, volume 6 of *YangSky.com Monographs in Information Sciences*. Yang's Scientific Press, Tucson, AZ, Dec. 2007. ISBN:0-9721212-4-2.

[49] T. Yang. Applications of computational verb theory to the design of accurate video flame-detecting systems. *International Journal of Computational Cognition*, 5(3):25–42, September 2007 [available online at http://www.YangSky.us/ijcc/ijcc53.htm, http://www.YangSky.com/ijcc/ijcc53.htm].

[50] T. Yang. Cognitive engineering and cognitive industry. *International Journal of Computational Cognition*, 5(3):1–24, September 2007 [available online at http://www.YangSky.us/ijcc/ijcc53.htm, http://www.YangSky.com/ijcc/ijcc53.htm].

[51] T. Yang. Computational verb neural networks. *International Journal of Computational Cognition*, 5(3):57–62, September 2007 [available online at http://www.YangSky.us/ijcc/ijcc53.htm, http://www.YangSky.com/ijcc/ijcc53.htm].

[52] T. Yang. Computational verb theory: Ten years later. *International Journal of Computational Cognition*, 5(3):63–86, September 2007 [available online at http://www.YangSky.us/ijcc/ijcc53.htm, http://www.YangSky.com/ijcc/ijcc53.htm].

[53] T. Yang. Learning computational verb rules. *International Journal of Computational Cognition*, 5(3):43–56, September 2007 [available online at http://www.YangSky.us/ijcc/ijcc53.htm, http://www.YangSky.com/ijcc/ijcc53.htm].

[54] T. Yang and L.O. Chua. Channel-independent chaotic secure communication. *International Journal of Bifurcation and Chaos*, 6(12B):2653–2660, Dec. 1996.

[55] T. Yang and Y. Guo. Measures of ambiguity of computational verbs based on computational verb collapses. *International Journal of Computational Cognition*, 5(4):1–12, December 2007 [available online at http://www.YangSky.us/ijcc/ijcc54.htm, http://www.YangSky.com/ijcc/ijcc54.htm].

[56] T. Yang and S.M. Shahruz. Channel-independent chaotic secure communication system using general chaotic synchronization. *Telecommunications Review*, 7(2):240–254, Mar.-Apr. 1997.

[57] Jian Zhang and Minrui Fei. Determination of verb similarity in computational verb theory. *International Journal of Computational Cognition*, 3(3):74–77, September 2005 [available online at http://www.YangSky.us/ijcc/ijcc33.htm, http://www.YangSky.com/ijcc/ijcc33.htm].

[58] Sheng Zhu, Zhong-Jie Wang, Yong Liu, and Bao-Liang Xia. An improvement of the design of computational verb PID-controllers. *System Simulation Technology*, 2(1):25–30, Jan. 2006. (in Chinese).