

An approach to function identification in automated conceptual design of mechanism systems

Lin Tang

Received: 5 July 2006 / Revised: 13 August 2007 / Accepted: 19 November 2007 / Published online: 27 August 2008
© Springer-Verlag London Limited 2008

Abstract It has been a challenge in computer-aided conceptual design of mechanism systems as how to describe the kinematic requirements and the functions of a mechanism so that computers can easily recognize, and eventually make automatic selections and combinations among many different types of mechanisms (those combinations include mechanisms of single and multi-degree-of-freedom, and connected in series, or parallel, or overlap, or feedback). In this paper, a methodology is presented to describe the kinematic characteristics of mechanism systems and their associated requirements so that they can easily be recognized by computers. A practical approach is to define “a kinematic function” to represent the kinematic characterizations of a mechanism. It reflects the input and output motion of a given system. This kinematic function can regularly and accurately be described using function units and trees (any function tree consists of more than one function units and is regularly expressed). Because of its uniqueness, the process of mechanism synthesis can be easily recognized by a computer. The methodology demonstrated in the paper has universality and has been successfully incorporated into a software, which has been proven to be a useful tool for the conceptual design of mechanism systems.

Keywords Function identification · Function tree · Mechanism synthesis · Automation

1 Introduction

In computer-aided conceptual design of mechanism systems, the information processing of function is one of the important and also the challenges. The function information of mechanism systems should reflect their kinematic characteristics. In order to enable the computers to aid the conceptual design of mechanism systems, the references (Li et al. 1996; Chakrabarti and Bligh 1996; Chiou and Kota 1999; Tang 2000, 2002, 2006, 2008; Moon and Kota 2002; Roth 1981; Kusiak and Szczerbicki 1992; Kirschman and Fadel 1998; Stone and Wood 2000; Qi 1987; Tang et al. 2001) presented the different methodologies for expressing the function of mechanisms (Li et al. 1996; Chakrabarti and Bligh 1996; Chiou and Kota 1999; Tang 2000, 2002, 2006, 2008; Moon and Kota 2002; Roth 1981; Tang et al. 2001), machines (Kusiak and Szczerbicki 1992; Kirschman and Fadel 1998) and products (Roth 1981; Stone and Wood 2000). In them, the references (Li et al. 1996) and (Chiou and Kota 1999; Tang 2000, 2002, 2006, 2008; Moon and Kota 2002; Tang et al. 2001) discussed the description of kinematic function and the information processing of conceptual design in detail. Because the kinematic feature are generally the main manifestation of the mechanism function, the reference (Li et al. 1996) dealt with the function information by means of the rules for transforming function. Thus, it has to constitute a large number of the detail and specific rules. The references (Chiou and Kota 1999; Moon and Kota 2002) completed the synthesis of mechanism types with the aid of transformation matrixes (Chiou and Kota 1999) and dual-vector algebra (Moon and Kota 2002). The methodology demonstrated in Chiou and Kota (1999) can decompose a given task into sub-task automatically, but both Chiou and Kota (1999) and Moon and Kota (2002) only address to the

L. Tang (✉)
College of Mechanical Engineering, Donghua University,
2999 Ren Ming North-Road, 201620 Shanghai,
People's Republic of China
e-mail: Ltang@dhu.edu.cn; Ltang@126.com

Fig. 1 Function tree

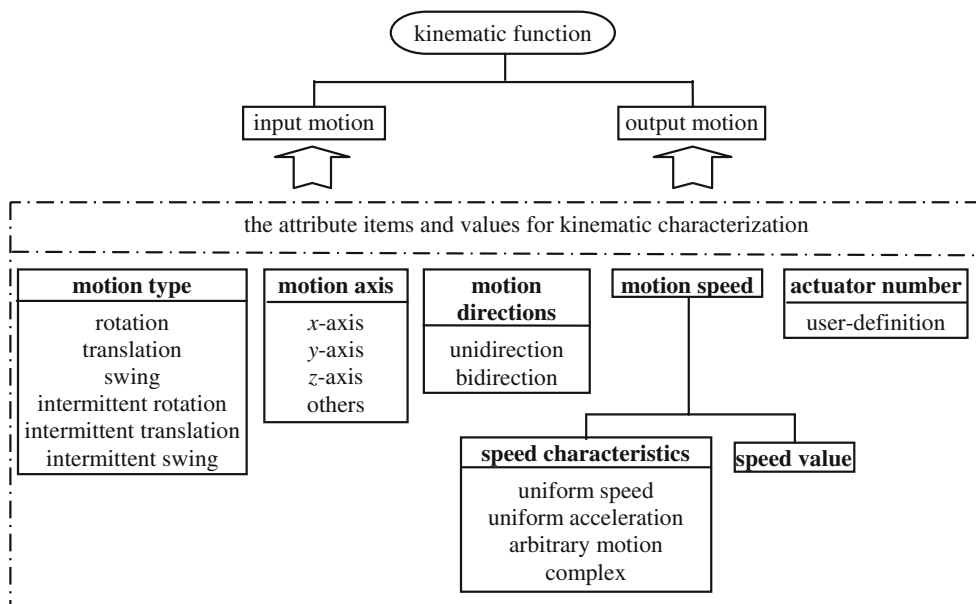
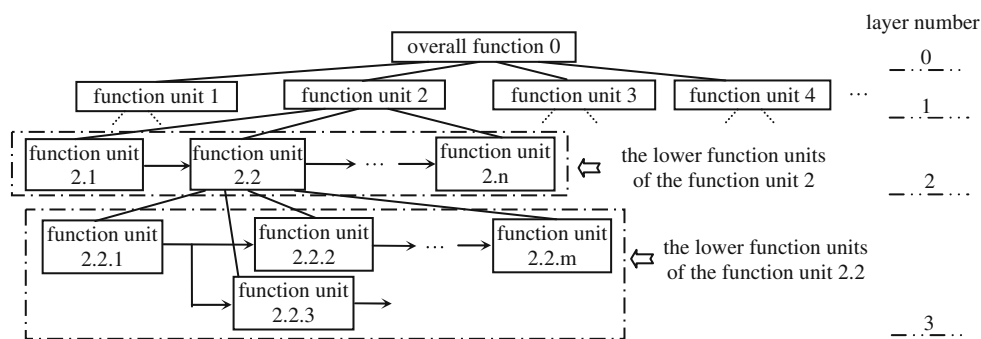


Fig. 2 The expression of kinematic function

synthesis of single degree-of-freedom (DOF) mechanisms and focus on the series combination of mechanisms. As the kinematic function of mechanism systems becomes complicated, such as a multi-DOF mechanism system or a system that is composed of difference combination types of some single DOF and multi-DOF mechanisms, it would become very complicated or even impossible to process the information by the methodology described in Chiou and Kota (1999) and Moon and Kota (2002). In order to tackle some difficulties systematically in the mechanism synthesis, this paper describes and defines the kinematic function of function units as shown in Fig. 2 and the function tree that is composed of the function units as shown in Fig. 1 to address the automation of both single DOF also multi-DOF mechanism selections and the automation of combinations for different types (in series, or in parallel, or in overlap, or in feedback). The author used the methodology described in the paper, combined with the established rules and algorithms presented in Tang (2000, 2002, 2006, 2008),

Tang et al. (2001), and developed a software that can synthesize both single DOF and multi-DOF mechanisms, such as the selection of mechanisms and the combination of mechanisms in different types under the given function trees. This paper presents this unique approach and demonstrates how it works in the software.

In the conceptual design of mechanism systems, if the overall function, i.e., the function of a system is complex so that no single mechanism can directly realize its motion requirements, it is usually decomposed into some sub-functions layer by layer, which (including the overall function) are defined as the function units in the paper, and their relations are represented with the function tree (Tang 2000; Qi 1987). The kinematic functions of the function units are simpler and simpler from the top to bottom of the function tree (shown in Fig. 1). Thus, the definition and expression of the kinematic function, the mark of the function units and the description of the function tree become the focuses for the computers to identify the

conceptual design process of mechanism systems, which are just the issue of the paper. In addition, the paper establishes the rules and the flowchart for a computer to aid the conceptual design of mechanism systems starting from the motion requirements of an overall function and ones of function units in a given function tree.

2 Expression of the kinematic function

The kinematic function reflects the kinematic characteristics of mechanism systems or the function units and is expressed by means of the attribute items, such as motion type, motion axis, motion direction, motion speed, actuator number of the inputs and outputs. By the attribute values of these items, we can concretely and exactly express the kinematic function (shown in Fig. 2), so to avoid the expression ambiguity and enable the computers to identify the kinematic function readily.

3 Description of the function tree

The precise description of the function tree is the basis of the automated conceptual design of mechanism systems and is also the foundation for the computers to identify the design process. The function units in the function tree can be taken as the abstract manifestation of the kinematic functions. The function units with one input and one output motion are defined as “single DOF function unit”, and as “multi-DOF function unit” if otherwise. The function tree represents clearly the realization of the overall function and helps to carry concretely the abstract thinking of the early design stage of mechanism systems. In order to enable the computers to identify the conceptual design process of mechanism systems starting from a specified function tree, each function unit is marked with a specific label number (shown in Fig. 1), which reflects its position in the function tree. When the kinematic function of a function unit is complex, it can be further decomposed into some subfunction units, and its kinematic function is the synthesis of kinematic functions of these subfunction units. We call the decomposed function unit as the “upper function unit” of its subfunction units, and call the subfunction units as the “lower function units” subordinate to the “upper” one. The relationships of the function units between neighbor layers in the function tree are those of the upper function units and the lower ones. The upper one is the goal to attain after synthesizing the kinematic functions of its lower ones, while the lower ones are the means of realizing the kinematic function of their upper one. No matter how to decompose the overall function of mechanism systems, the following rules can be specified:

1. In the function tree, the label for the overall function is “0” and its layer number is also “0” (shown in Fig. 1).
2. The labels of the lower function units subordinate to the overall function are an integer beginning from number 1. For instance, function units 1, 2, 3 and 4 in Fig. 1 are the lower function units subordinate to the overall function 0.
3. For the further other lower function units, the rule for marking their labels is to append a dot “.” and then another integer in sequence after the dot.

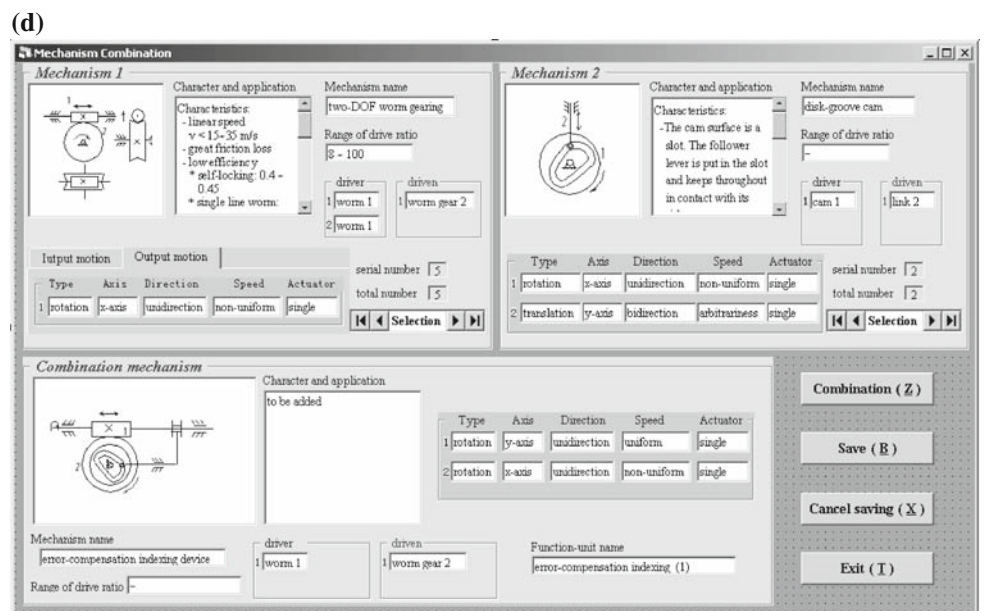
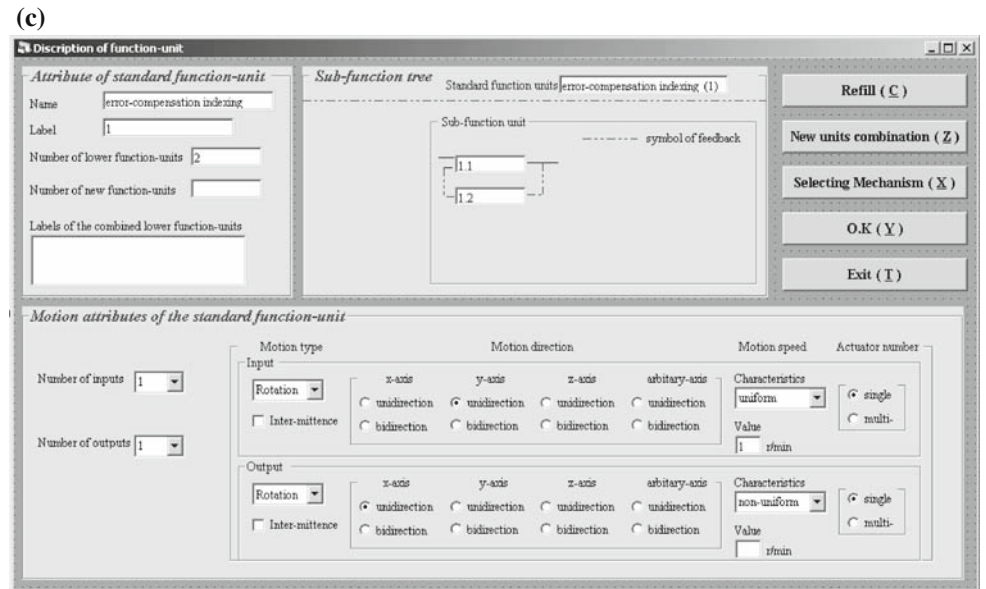
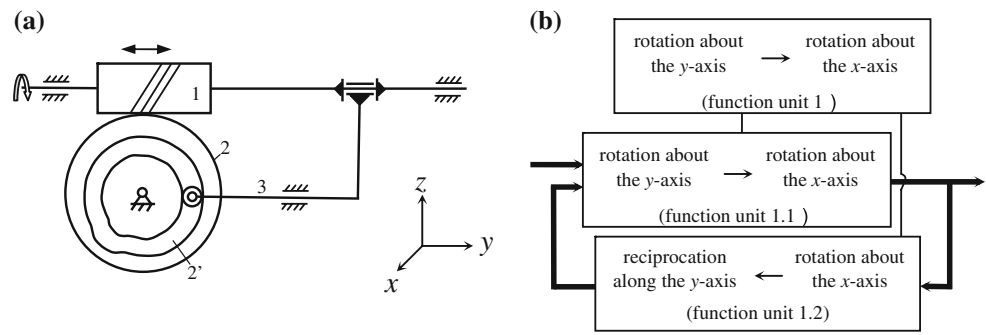
In the light of the above rules of marking the labels of function units, the function unit’s layer number will reflect its position in the function tree. For instance, the function unit 2.2.2 in Fig. 1 is the second lower function unit of its upper function unit 2.2, and is situated at third layer in the function tree. There are just three digits in the label. The standardized labels can indicate the status of function units, which reflect their subordinate relationships and also mark clearly their positions in the function tree. The arrowheads between the lower function units of the same upper function unit represent the link relationships of the inputs and outputs between them. These relationships provide the basis to establish rules and algorithms for any combination type of mechanism solutions of the lower function units. It is also the condition to enable the computer to identify and process the procedural information of the conceptual design of mechanism systems (Tang 2000, 2002, 2006, 2008; Tang et al. 2001). The principle and method of mechanism synthesize, for instance, the rules and algorithms to select and combine mechanisms, have been expatiated in the references (Tang 2000, 2002, 2006, 2008; Tang et al. 2001).

The mechanism solutions of the decomposed overall functions can be obtained by combining the mechanism solutions of the lower function units in the function tree according to their link relationships. Based on the function tree so described, the computers can completely process the procedural information of the conceptual design of mechanism systems. Figures 3b and 4b are the applied examples of function tree described upon the above rules.

4 Identification of the function units

The function units are the basic objects to be solved in the automated conceptual design of mechanism systems; however, how to specify the information about the function units is the crux for a computer to aid the conceptual design of mechanism systems. In the software for computer-aided conceptual design, the above approach presented had been used to develop the platform for specifying the attribute items of the function units. Figures 3c and 4c are the two platform interfaces in this software. Both consist of three modules: one specifies the basic attributes of the function

Fig. 3 **a** The exemplary indexing mechanism (Sh 1999). **b** The function tree corresponding to **a** (Tang 2006). **c** The attribute values of the function unit 1 in **b** (Tang 2006) (The interface for specify the attribute values of single DOF function units). **d** One of the obtained mechanism solutions of the standard function unit 1 in **c** (Tang 2006)



units to be solved; another describes the sub-function tree which reflects the relationships of the lower function units, whose mechanism solutions will be combined; and the third one reflects the kinematic function generated after

synthesizing the lower function units in sub-function tree. In what follows, it will, respectively, explain the concrete contents to be marked in the modules and the method for

4.1 Module for specifying the basic attributes of function unit

In the paper, the function units whose mechanism solutions are being solved are defined as “standard function unit”. This module is used to specify the basic attributes of the standard function unit and involves its name that is the label in the function tree and the number of its lower function units. They can be directly filled according to the information of the standard function unit in the function tree.

4.2 Module for specifying sub-function tree

If a standard function unit has more than one lower function units, in order to simplify the mechanisms combination, the software system should combine the mechanism solutions of only two lower function units subordinate to the standard function unit every time (in references Tang 2006; Tang et al. 2001, the content is narrated in detail). This module is only used to specify the sub-function tree that is composed of two lower function units to be synthesized. According to the combination form of the two lower ones in the function tree, the sub-function tree will be automatically generated by way of man–machine interaction. When a standard function unit has no lower function unit, then there will be no sub-function tree and also no content in the module.

4.3 Module for specifying the kinematic function

This module is used to describe the kinematic function composed of the external input and output motions of both the function units in the sub-function tree. If standard function unit has no lower function unit, i.e., there is no sub-function tree; then the kinematic function in the module is the one of the standard function unit. As shown in Fig. 2, the kinematic function reflects the motion characterizations of the input and output of the function units. And the motion characterizations are composed of some attribute items that represent the various aspects of the motion. In this module, the designer can easily and precisely specify the kinematic function of the function units of the sub-function tree by choosing or writing the values of the attribute items. When the speed characterization is uniform motion, its attribute value (e.g., speed value) should be filled according to the actual requirement. Figure 3c is the interface for specifying the kinematic function of single DOF function units. For multi-DOF function units, because they have several input and/or output motions, the attribute values of the items are, respectively, chosen or written in two separate TabStrips, the one is used for input motions, and the other for output motions (shown in Fig. 4c).

The kinematic function of function units or the overall function is very important information for the conceptual design of mechanism systems. And how to enable a computer to identify this information is another crux of the automated conceptual design of mechanism systems.

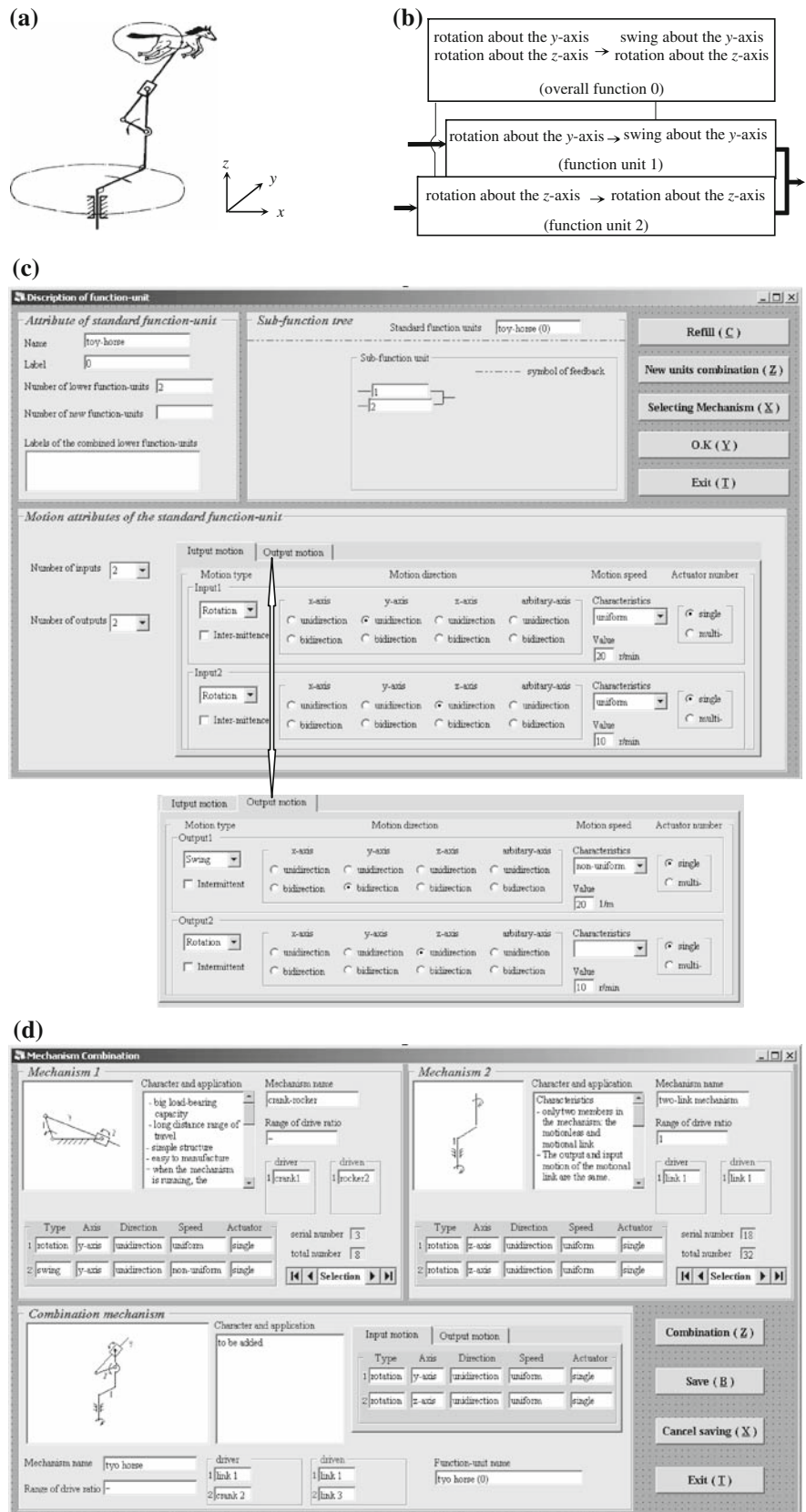
The afore-mentioned regular method for describing the kinematic function is not only very simple, convenient, quick and precise that designers can easily complete the work of describing kinematic function of function units, but can also ensure the expression of kinematic function uniquely and, so that the computers can accurately identify it. Thus, it can avoid the designers to misunderstand and a computer being unable to identify the kinematic function expressed in human natural language, which is usually fuzzy and hard to unify. Simultaneously, it lays a good foundation for the realization of the automated conceptual design of mechanism systems and also solves the challenge for a computer to identify the requirements of the kinematic function of mechanism systems.

In computer-aided the conceptual design of mechanism systems, the information relative to the function units to be solved is collected by means of the interface for specifying the function units. The informations of the sub-function tree and the kinematic function are the basis of finding mechanism solutions of the standard function unit, while the basic attributes of the standard function unit are the foundation for the computers to process the conceptual design of mechanism systems.

5 Examples specifying the kinematic function of mechanism systems

In Figure 3a shown an indexing mechanism (Shen 1999), which consists of a worm (denoted as 1) and a worm gear (denoted as 2) and is a two-DOF mechanism. The worm 1 can rotate about the y -axis and translate along the y -axis. The worm gear 2 has a disk-groove cam 2' with it. The profile of the groove cam 2' is designed based on the measured indexing error to compensate it. Namely, the cam drives the worm 1 to translate along the y -axis. This translation brings worm gear 2 a subjoined rotation to compensate the indexing errors. Figure 3b is the function tree corresponding to Fig. 3a. The “function unit 1” is the standard function unit that has one external input: the rotation about y -axis, and one external output: the rotation about x -axis. Therefore, it belongs to the single DOF function unit. Figure 3c is the interface for specifying the attribute values of the single DOF function unit 1. Figure 3d is one of the obtained mechanism solutions of the standard function unit 1 in Fig. 3c with the help of the software for computer-aided conceptual design of mechanism systems.

Fig. 4 **a** The exemplary toy-horse mechanism (Sun and Chen 2001). **b** The function tree corresponding to **a** (Tang 2006). **c** The attribute values of the “overall function 0” in **b** (Tang 2006) (The interface for marking the attributive values of multi-DOF function units). **d** The one of the mechanism solutions of the “overall function 0” in **c** (Tang 2006)



the morphological matrix for finding mechanisms of upper function unit	the lower function units subordinate to the upper function unit		the mechanism solutions of the lower function units					
			1	2	...	<i>i</i>	...	<i>m</i>
	1	F_1	L_{11}	L_{12}		L_{1i}		L_{1m}
	2	F_2	L_{21}	L_{22}		L_{2i}		L_{2m}
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
<i>n</i>	F_n	L_{n1}	L_{n2}		L_{ni}		L_{nm}	
the mechanism solutions of the upper function unit			mechanism 1			mechanism 2		

Fig. 5 The morphological matrix for finding the mechanisms of upper function unit

Figure 4a is an example of toy-horse mechanism (Sun and Chen 2001). Figure 4b is the function tree corresponding to Fig. 4a. The “overall function 0” has two external inputs: the rotation about the y-axis as the input of “function unit 1”, and the rotation about the z-axis as the input of “function unit 2”. Its external output is the superposition of the output motions of both function units “1” and “2”, i.e., the swing about the y-axis (the output of function unit 1) and the rotation about the z-axis (the output of function unit 2). The kinematic characteristic of the overall function is the synthesis of the kinematic characteristics of “function unit 1” and “function unit 2”. In Fig. 4a, the crank-swing rock is the solution of “function unit 1”, and the two-bar linkage is the solution of “function unit 2”. Because the “overall function 0” has two external input motions, it belongs to the multi-DOF function unit. Figure 4c is the interface for describing the attribute items of the multi-DOF function units and has been specified with the attribute values of the “overall function 0”. Figure 4d is one of the obtained mechanism solutions of the overall function 0 in Fig. 4c.

A function tree contains kinematic feature of each function unit and the relationship between function units. It is the foundation of mechanism synthesis. For a given function tree, if designers choose different mechanisms from the mechanism base to match kinematic features of the function units, the synthesized results will be different.

6 Identification of the conceptual design process of mechanism systems

The mechanism solutions of each function unit in the function tree can be obtained by selecting or combining mechanisms, or by combining the mechanism solutions of

its lower function units. Based on the function tree as shown in Fig. 1 and the above described concepts, the conceptual design process of mechanism systems can then be systematized. In the course of automated conceptual design, the computer will find the mechanism solutions of each function unit layer by layer from the bottom to the top of the function tree with the aid of the specified attribute information of the standard function units. The methodology for computer to identify the conceptual design process of mechanism systems may be summarized into the following five steps:

1. The mechanism solutions of the function units in the bottom layer of the function tree (which have no lower ones), can first be obtained from the sub-model “selection of mechanism types” of the software system (Tang 2008). These mechanisms will form the mechanism solution catalogs of these function units.
2. If an upper function unit has lower function units, then its mechanism solutions will be the combination mechanism obtained by combining the mechanism solutions of its lower function units. The morphological matrix for obtaining its mechanism solutions can then be established (Tang 2002, 2006; Qi 1987). In the matrix, the elements in every row are just the mechanism solution catalog of the lower function unit subordinate to the standard function unit (shown in Fig. 5).
3. According to the synthetic way of the lower function units in the function tree and the combination compatibility of the mechanism solutions of different rows in the matrix, we can obtain various mechanism solutions of the standard function unit, which consist of its solution catalog.

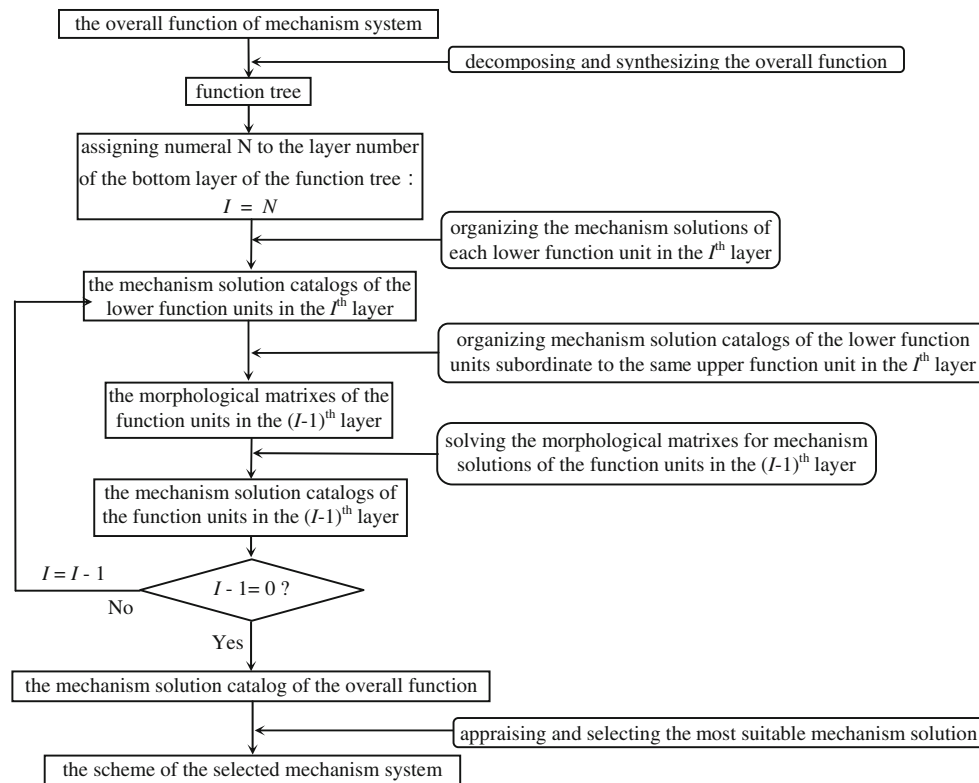


Fig. 6 The flowchart for automated conceptual design of mechanism systems

4. Repeating the steps 2. and 3. from the bottom upward to the top of the function tree, we can obtain the mechanism solutions of all the function units in the function tree and establish their solution catalogs until obtaining finally the ones of the overall function of the mechanism system.
5. By appraising the listed mechanisms in the solution catalog of the overall function of the system, we can then select the most suitable one as the mechanism solution of the system for the design task.

The aforementioned method makes the design process systemic and regular. It is practically feasible both to describe the function tree and to find the mechanism solutions of the function units in the tree. It is also beneficial to construct conceptual design modules of mechanism systems that can be identified by computers. Figure 6 is the flowchart for the automated conceptual design of mechanism systems starting from a given function tree.

7 Conclusion

This paper demonstrates the methodology as how to construct function trees and how to describe function units to represent kinematic function so a computer can identify them, and explains in detail the process of the automated

conceptual design of mechanism systems from three aspects, i.e., (1) the normative expression of the kinematic function, (2) the description of the function tree, (3) the standardized methods to find mechanism solutions of the function units. The kinematic function reflects the kinematic characteristics of function units and is expressed by the values of attribute items of input and output motions, which can ensure the accurate expression of the kinematic function. Further, the regular descriptions of both the function units and the function tree lay a good foundation for the computers to identify the conceptual design process of mechanism systems. The rules of finding mechanism solutions satisfying the kinematic functions of function units and the flowchart shown as in Fig. 6 are appropriate to programming and enable a computer to identify the conceptual design process. The methodology described in the paper has been successfully incorporated into a software, in which automated selection and combination of both single DOF also multi-DOF mechanisms (Tang 2002, 2008) has been realized under the given function trees. It can be applied to different combination types, including series, parallel, overlap and feedback of mechanisms (Tang 2002, 2006; Tang et al. 2001).

Acknowledgments The author thank ICCDM (Internet-Based Collaborative Research Center on Modern Design and Manufacturing of the Ministry of Education of China) for providing the software

resources to carry out this research, and also expresses her gratitude to Prof. Ren-Ze Chen from Donghua University and Dr. Jun Liu from Jet Propulsion Laboratory of California Institute of Technology for assistance in the language correction.

References

- Chakrabarti A, Bligh TP (1996) An approach to functional synthesis of mechanical design concepts: theory, applications, and emerging research issues. *Artif Intell Eng Des Anal Manuf* 10:313–331
- Chiou SJ, Kota S (1999) Automated conceptual design of mechanisms. *Mech Mach Theory* 34:467–495
- Kirschman CF, Fadel GM (1998) Classifying functions for mechanical design. *ASME J Mech Des* 120:475–482
- Kusiak A, Szczerbicki E (1992) A formal approach to specifications in conceptual design. *ASME J Mech Des* 114:659–665
- Li CL, Tan ST, Chan KW (1996) A Qualitative and heuristic approach to the conceptual design of mechanisms. *Eng Appl Artif Intell* 9:17–31
- Moon Y-M, Kota S (2002) Automated synthesis of mechanisms using dual-vector algebra. *Mech Mach Theory* 37:143–166
- Qi Ch-Z (1987) *Modern design methodology for machine*. China Building Industry Press, Beijing
- Roth KH (1981) Foundation of methodical procedures in design. *Des Stud* 2:107–115
- Shen Y-Sh (1999) *Theory of machines and mechanisms*. Tsinghua University Press, Beijing
- Stone RB, Wood KL (2000) Development of a functional basis for design. *ASME J Mech Des* 122:359–370
- Sun H, Chen ZM (2001) *Principle of Machinery*. Higher Education Press, Beijing
- Tang L (2000) Automated synthesis theory and realization for mechanism types. Ph.D. Dissertation (in Chinese), Shanghai Jiaotong University, Shanghai, People's Republic of China
- Tang L (2002) Research on automated system for selection and combination of mechanisms based on set theory. Report of Postdoctoral Research Work (in Chinese), Donghua University, Shanghai, People's Republic of China
- Tang L (2006) Many-degree of freedoms mechanism combination in automated conceptual design of mechanism systems. *J Mech Eng Sci* 220(5):727–740
- Tang L (2008) An approach to the selection of mechanism types in computer aided conceptual design. *J Mech Eng Sci* 222(3): 423–433
- Tang L, He W, Zou HJ (2001) Das Prinzip automatisierter Kombination von Mechanismen und dessen Realisierung (The principle for automated connection of mechanisms and its realization). *Mech Mach Theory* 36:997–1008