

УДК 629.78

В.Н. СПИРИДОНОВ*Днепропетровская государственная финансовая академия, Украина***МЕТОДИКА СОЗДАНИЯ МОДЕЛИ СЛОЖНОЙ СИСТЕМЫ
МЕТОДОМ ПЕРЕХОДНЫХ ФУНКЦИЙ**

Предложена методика создания модели сложной системы. Система отнесена к одному из классов сложных систем – дискретных динамических систем с постоянной структурой. Модель системы предназначена для решения классической задачи теории систем - задачи прогноза состояния. Метод моделирования базируется на описании функционирования системы в пространстве состояний переходными функциями состояния. Методика создания модели представляет собой определенную последовательность операций, обеспечивающую возможность создания моделей систем, пространство состояний которых может достигать астрономических величин.

Ключевые слова: моделирование сложных систем, метод переходных функций, методика создания модели.

Введение

При решении задачи прогноза состояния сложных систем [1,2] возникает проблема размерности модели. Одним из методов теории систем, позволяющих существенно снизить размерность модели, является применение переходной функции для описания связей в пространстве состояний системы. В работе [3] предложен метод моделирования одного из классов сложных систем - дискретных динамических систем с постоянной структурой. Математическая модель управления системы с постоянной структурой в пространстве состояний представлена таблицей переходных функций и набором переходных функций, описывающих переходы элементов системы в пространстве состояний.

В работе [4] рассмотрена задача моделирования одного из классов сложных технических объектов методом переходных функций. В качестве объекта моделирования рассмотрен класс объектов, предназначенных для автономного выполнения специальных заданий в течение определенного времени без связи с устройствами управления по заранее заданной временной программе управления. К данному классу объектов можно отнести роботы, гибкие производственные системы, космические аппараты, межпланетные станции и другие объекты. В соответствии с задачей исследования выбрано пространство состояний и формализованы переходные функции объекта моделирования.

В качестве задачи моделирования рассматривается задача прогноза состояния сложной системы. Модель объекта должна обеспечить операторов, управляющих системой, полной информацией о прогнозируемых состояниях системы на выданные команды программы управления. Однако в этих ра-

ботах проблема создания моделей методом переходных функций не рассматривалась.

Задача разработки методики создания модели является актуальной для каждого метода моделирования, поскольку в существенной мере определяет применимость метода и его эффективность. Особенно важно наличие эффективной методики при создании моделей большой размерности, поскольку в зависимости от квалификации специалистов, создающих модель, и их знаний о моделируемой системе могут быть получены различные по полноте и точности модели одной и той же системы. Это недопустимо при разработке моделей, особенно в тех случаях, когда модель используется в контуре управления системой.

Постановка задачи

Целью статьи является рассмотрение методики создания модели системы, имеющей особенности моделирования, изложенные в работе [3]. Там же представлена постановка задачи прогноза состояния дискретной динамической системы с постоянной структурой, которая выполнена в терминах теории множеств. Представленная в данной статье методика создания модели должна соответствовать терминам постановки задачи моделирования. Методика должна обеспечивать возможность автоматизации процесса создания модели, что является важным требованием для моделей большой размерности.

Результаты исследования

Методика построения модели по методу переходных функций заключается в последовательном выполнении следующих операций.

1. Выполнение декомпозиции системы на подсистемы. Система состоит из конечного множества подсистем:

$$C = \{c_1, c_2, \dots, c_m, \dots, c_M\}, m = \overline{1, M}, \quad (1)$$

$$c_1 \cap c_2 \cap \dots \cap c_m \cap \dots \cap c_M = \emptyset, c_m \neq \emptyset, \quad (2)$$

$$A = \bigcup_{m=1}^M c_m, m = \overline{1, M}, m \leq n, \quad (3)$$

где m – индекс подсистемы; M – количество подсистем в составе системы.

После выполнения декомпозиции название каждой подсистемы кодируется (или используется принятое сокращение). Если подсистема резервируется, в кодировке указывается номер комплекта.

2. Определяются неуправляемые командами подсистемы в составе системы, и принимается решение о необходимости их представление в модели, как подсистем, функционирующих постоянно.

3. Выполняется декомпозиция каждой подсистемы на элементы. Подсистема состоит из конечного множества элементов:

$$c_m = \{a_1^m, a_2^m, \dots, a_g^m, \dots, a_G^m\}, g = \overline{1, G}, G \leq n, \quad (4)$$

где G – максимальное количество элементов в подсистеме c_m .

После выполнения декомпозиции название каждого элемента кодируется (или используется принятое сокращение). Если элемент резервируется, в кодировке указывается номер комплекта.

4. Определяется подсистема, имеющая максимальное количество элементов, и количество элементов в этой подсистеме G . Значение G определяет величину списка элементов в списочной таблице переходных функций (табл. 1).

5. Определяются неуправляемые командами элементы каждой подсистемы, и принимается решение о необходимости их представление в модели, как элементов, функционирующих постоянно.

6. Составляется общий список элементов системы, которые необходимо представить в модели:

$$A = \{a_1, a_2, \dots, a_i, \dots, a_n\}, i = \overline{1, n}, \quad (5)$$

где i – порядковый номер элемента; n – количество элементов в системе.

Значение n определяет размер списка элементов, который будет представлен в полной таблице переходных функций (табл. 2).

7. Выбирается и кодируется пространство состояний элементов. Пространство состояний элемента является конечным:

$$X_i = \{x_{i1}, x_{i2}, \dots, x_{iq}, \dots, x_{iQ}\}, q = \overline{1, Q}, \quad (6)$$

где x_{iq} – одно из возможных состояний элемента a_i ;

Q – размерность пространства состояний элемента.

Таблица 1

Перечень элементов подсистемы

U	d_f	a_1^m	a_2^m	...	a_g^m	...	a_G^m
u_1	d_2	a_7	a_8	...	0	...	0
u_2	d_1	a_1	a_2	...	0	...	0
...
u_j	d_f^m	a_1^m	a_2^m	...	a_g^m	...	a_G^m
...
u_j	d_f^M	a_1^M	a_2^M	...	a_g^M	...	a_G^M

Таблица 2

Перечень элементов системы

	a_1	a_2	...	a_i	...	a_n
u_1	d_2	d_3	...	0	...	0
u_2	d_3	d_2	...	d_2	...	0
...
u_j	0	0	...	d_1	...	0
...
u_j	d_1	d_2	...	d_r	...	d_R

Пространство состояний системы представляет декартово произведение пространства состояний её элементов:

$$X = \{X_1 \times X_2 \times \dots \times X_i \times \dots \times X_n\}, i = \overline{1, n}, \quad (7)$$

a состояние системы в любой момент времени представляет собой кортеж состояний ее элементов:

$$x(t) = \{x_1(t), x_2(t), \dots, x_i(t), \dots, x_n(t)\}, i = \overline{1, n}, \quad (8)$$

где $x(t)$ – состояние системы в момент времени t ;

$x_i(t)$ – состояние элемента a_i в момент времени t .

8. Система управляется временными программами управления:

$$E = \{e_1, e_2, \dots, e_v, \dots, e_V\}, v = \overline{1, V}, \quad (9)$$

где v – порядковый номер программы управления;

V – максимальное количество программ управления, используемых при управлении системой.

9. Временная программа управления представляет собой упорядоченную во времени последовательность команд управления с временами их выдачи:

$$e_v = (u_1^v(t_1^v), u_2^v(t_2^v), \dots, u_h^v(t_h^v), \dots, u_H^v(t_H^v)), h = \overline{1, H}, \quad (10)$$

где u_h^v – имя (номер) команды из множества команд управления системой U , выдаваемой в v -й программе управления; h – порядковый номер команды в v -ой программе управления; t_h^v – время выдачи h -й команды; N – максимальное количество команд в v -й программе управления.

10. Определяется и кодируется множество команд управления системой. Множество команд управления системой является конечным:

$$U = \{u_1, u_2, \dots, u_j, \dots, u_J\}, j = \overline{1, J}, \quad (11)$$

где j – порядковый номер команды; J – максимальное количество команд, используемых при управлении системой; u_j – имя (номер) команды управления. Составляется список команд управления.

11. Последовательно анализируется воздействие каждой команды управления на элементы и подсистемы системы. При этом возможны три варианта воздействия.

Первый (12) описывает переход системы из одного состояния в другое при выдаче команды управления, действующей на один из элементов системы:

$$x(t+1) = \left(d_1 \{x_i(t), u_j(t+1)\}, x_{A \setminus i}(t) \right), \quad (12)$$

где $x_i(t)$ – состояние элемента a_i в момент времени t ; $x_{A \setminus i}(t)$ – состояния остальных элементов системы, которые по команде $u_j(t+1)$ в момент времени $(t+1)$ не изменились. d_1 – переходная функция элемента.

Второй (13) описывает переход системы из одного состояния в другое при выдаче команды управления u_j , действующей на одну из подсистем системы c_m :

$$x(t+1) = \left(d_f^m \{x_{c_m}(t), u_j(t+1)\}, x_{A \setminus c_m}(t) \right), \quad (13)$$

где $x_{c_m}(t)$ – состояния подсистемы в момент времени t ; $x_{A \setminus c_m}(t)$ – состояния остальных элементов системы, которые по команде $u_j(t+1)$ в момент времени $(t+1)$ не изменились; d_f^m – переходная функция подсистемы c_m .

Третий вариант предусматривает случай, когда команда управления не воздействует ни на один элемент или подсистему системы, например, технологическая команда. Для такой команды делается отметка в списке команд управления о не включении её в модель системы.

12. В первом варианте, описываемом соотношением (12), рассматривается правило перехода элемента в его пространстве состояний по данной команде управления. Каждому уникальному правилу перехода (переходной функции элемента) присваивается уникальное имя. Если такая переходная функция встречалась ранее при описании переходов других элементов системы по другим командам управления, то используется ранее присвоенное имя переходной функции.

13. Для второго варианта анализируется характер воздействия команды управления на элементы подсистемы. Соотношение (14) описывает переход системы из одного состояния в другое для общего случая, когда команда управления u_j действует на одну из подсистем системы c_m , при этом характер воздействия команды на элементы подсистемы не одинаков:

$$x(t+1) = \left(d_1^m \{x_{c_1}(t), u_j(t+1)\}, \dots, d_R^m \{x_{a_G^m}(t), u_j(t+1)\}, x_{A \setminus c_m}(t) \right). \quad (14)$$

В этом случае переходная функция подсистемы представляет собой упорядоченное множество переходных функций элементов подсистемы c_m :

$$d_f^m = \left(d_1^m, d_2^m, \dots, d_r^m, \dots, d_R^m \right), r = \overline{1, R}, \quad (15)$$

где R – количество переходных функций элементов в подсистеме.

Составляется список переходных функций подсистемы.

14. В частном случае, когда характер воздействия команды u_j на все элементы подсистемы одинаков, переход системы из одного состояния в другое при выдаче команды управления u_j , действующей на одну из подсистем системы c_m , представляется в виде соотношения:

$$x(t+1) = \left(d_1 \{x_{a_1^m}(t), x_{a_2^m}(t), \dots, x_{a_G^m}(t), u_j(t+1)\}, x_{A \setminus c_m}(t) \right) \quad (16)$$

где d_1 – переходная функция элемента. Определяется переходная функция подсистемы.

15. Составляется общий перечень переходных функций системы, имеющих уникальное имя. Система описывается конечным множеством переходных функций:

$$D = \{d_1, d_2, \dots, d_l, \dots, d_L\}, l = \overline{1, L}, \quad (17)$$

где l – индекс переходной функции; L – максимальное количество переходных функций, исполь-

зують при описанні системи. Величина L визначає кількість правил переходів, використовуваних при описанні системи, і в результаті, кількість модулів в бібліотеці переходних функцій системи.

16. Визначається застосовуваний тип таблиці переходних функцій (повна або списочна). Аналізуються результати виконання п.п. 13 і 14. В загальному випадку, якщо всі переходи підсистем в просторі станів відповідають вираженню (14), вибирається повна таблиця переходних функцій. В певному випадку, якщо всі переходи підсистем в просторі станів відповідають вираженню (16), вибирається списочна таблиця переходних функцій.

17. Заповнюється обрана таблиця переходних функцій. Для кожної команди, представленої в списку команд управління, заповнюється відповідна строка таблиці. В клітинках повної таблиці (представлено весь список елементів системи в строці таблиці) для кожної команди управління записуються відповідні імена переходних функцій, якщо дана команда впливає на елемент, і 0 – в протилежному випадку (табл. 2). Розмірність моделі в цьому випадку визначається розмірністю таблиці: добуток кількості команд управління на кількість елементів системи $N=J \times (n+1)$.

18. Для певного випадку, якщо всі переходи системи в просторі станів відповідають вираженню (16), то можна використовувати списочну таблицю переходних функцій (табл. 1). В цій таблиці імена функцій виведено в окремий стовпець, а в клітинках строки таблиці поміщається список імен елементів підсистеми, на які впливає дана команда. Так як підсистеми мають різну кількість елементів, то список елементів підсистем, які мають менше кількість елементів, доповнюється нулями. Розмірність моделі в цьому випадку дорівнює $N=J \times (G+2)$, де G – кількість елементів підсистеми, яка має максимальну кількість елементів. Цим досягається зменшення розмірності моделі в $\frac{n+1}{G}$ раз.

19. В тому випадку, якщо переходи системи в просторі станів описані вираженнями (14) і (16), то необхідно використовувати повну таблицю переходних функцій, так як списочна таблиця переходних функцій є певним випадком повної таблиці. Однак, якщо при описанні системи кількість команд, представлених в обох таблицях порівнянимо, можна використовувати обидва типи таблиць, що дозволяє знизити розмірність моделі до мінімально можливої. В цьому випадку моделюючий алгоритм повинен передбачувати воз-

можливість роботи з обома видами таблиць, забезпечивши більшу універсальність алгоритму за рахунок певного ускладнення. Цей варіант реалізації моделі є більш переважним в тому випадку, якщо модель створюється в автоматизованому режимі, що дозволяє користувачам використовувати обидва типи таблиць.

Заповнені таблиці переходних функцій системи і описання переходних функцій представляють собою математичну модель системи, створену методом переходних функцій. Реалізація математичної моделі і моделюючого алгоритму в обчислювальному середовищі не викликає труднощів.

Приведена послідовність операцій справедлива для побудови моделі будь-якої складної системи розглянутого класу. Однак, в залежності від призначення моделі системи, наявності технічної документації на систему, наявності раніше розроблених моделей систем цього класу і інших факторів, можливі певні особливості їх виконання.

Модель системи може використовуватися на різних етапах життєвого циклу системи, наприклад, на етапі проектування або на етапі експлуатації системи. На етапі проектування системи модель може бути використана для отримання проектних характеристик, проведення експериментів з моделлю з метою оптимізації характеристик системи. В цьому варіанті цілорозумно, щоб модель була підсистемою автоматизованої системи проектування розроблюваної системи. Модель повинна створюватися по мірі проектування системи і відображати її стан на різних етапах проектування. Для цього варіанта актуально забезпечення можливості автоматизованої системою проектування створення моделей окремих підсистем. Створення моделі системи в цьому випадку може бути забезпечено шляхом злиття таблиць переходних функцій підсистем, розроблених різними розробниками. При цьому в автоматичному режимі може бути проведено ряд перевірок правильності моделі.

Якщо модель системи використовується тільки на етапі експлуатації системи, наприклад, в контурі автоматизованого управління системою, модель розроблюється на етапі, коли технічна документація на моделювану систему готова і частина операцій виконав розробник системи. Наприклад, при розробці моделі складного технічного об'єкта типу бортової апаратури космічного апарату [4], модель розроблюється на основі інструкції по управлінню, в якій декомпозиція об'єкта на елементи і підсистеми, їх кодування виконано розробником. Дається також перелік команд управління, описання їх функцій по управ-

лению объектом. В этом случае модель может быть разработана как в ручном, так и в автоматизированном режиме. При автоматизированном построении модели эта операция может быть выполнена путем вывода на экран монитора форм соответствующих таблиц и заполнения их необходимой информацией.

Выводы

Разработана методика создания модели сложной системы, моделируемой методом переходных функций. Система относится к классу дискретных динамических систем с постоянной структурой. Методика создания модели представляет собой определенную последовательность операций, обеспечивающую возможность создания моделей систем, пространство состояний которых может достигать астрономических величин. Методика обеспечивает возможность автоматизации процесса создания модели и может быть реализована в виде программного

комплекса, являющегося подсистемой системы автоматизированного проектирования сложной системы или объекта.

Литература

1. Бусленко Н.П. *Моделирование сложных систем* / Н.П. Бусленко. – М.: Наука, 1978. – 400 с.
2. Бусленко Н.П. *Лекции по теории сложных систем* / Н.П. Бусленко, И.Н. Коваленко, В.В. Калашников. – М.: Сов. радио, 1973. – 440 с.
3. Спиридонов В.Н. *Построение математической модели дискретных динамических систем с постоянной структурой* / В.Н. Спиридонов // *Вестник Академии таможенной службы Украины. Сер. Технические науки.* – 2009. – № 2 (42). – С. 73-79.
4. Спиридонов В.Н. *Моделирование сложных технических объектов: состояния и переходные функции* / В.Н. Спиридонов // *Вестник Академии таможенной службы Украины. Сер. Технические науки.* – 2010. – № 1 (43).

Поступила в редакцию 2.03.2010

Рецензент: д-р тех. наук, проф., зав. каф. высшей математики и компьютерных технологий А.А. Рядно, Днепропетровская государственная финансовая академия, Днепропетровск.

МЕТОДИКА СТВОРЕННЯ МОДЕЛІ СКЛАДНОЇ СИСТЕМИ МЕТОДОМ ПЕРЕХІДНИХ ФУНКЦІЙ

В.М. Спиридонов

Запропоновано методику створення моделі складної системи. Система віднесена до одного із класів складних систем - дискретних динамічних систем з постійною структурою. Модель системи призначена для рішення класичної задачі теорії систем - задачі прогнозу стану. Метод моделювання базується на описі функціонування об'єкта в просторі станів перехідними функціями стану. Методика створення моделі являє собою певну послідовність операцій, що забезпечує можливість створення моделей систем, простір станів яких може досягати астрономічних величин.

Ключові слова: моделювання складних систем, метод перехідних функцій, методика створення моделі.

TECHNIQUE OF CREATION OF COMPLEX SYSTEM MODEL BY TRANSITIVE FUNCTIONS METHOD

V.M. Spiridonov

In article the technique of creation of complex system model is offered. The system is carried to one of classes of complex systems - discrete dynamic systems with constant structure. The system model is intended for the decision of a classical problem of the theory of systems - problem of the forecast of a state. The modeling method is based on the description of functioning of system in space of states by transitive functions of states. The technique of model creation represents the certain sequence of operations providing possibility of creation of system models which space of states can reach astronomical sizes.

Key words: modelling of complex systems, method of transitive functions, technique of modal creation.

Спиридонов Владимир Николаевич – кандидат технических наук, доцент кафедры высшей математики и компьютерных технологий, Днепропетровская государственная финансовая академия, Днепропетровск, Украина, e-mail: spiridonov-vladimir@rambler.ru.