ISSN 2079-3316 ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ № 2(6), 2011, с. 61–66

УДК 517.977

А. В. Евсеев

Построение реализаций нелинейных динамических систем

Аннотация. Для нелинейных систем с управлением рассмотрен переход от описания с помощью уравнений отображения вход–выход к описанию с помощью переменных состояния. Приведены необходимые и достаточные условия существования такого перехода на языке 1-форм и векторных полей. Построен алгоритм поиска реализации. Алгоритм положен в основу программного комплекса в системе символьных вычислений Maple.

Ключевые слова и фразы: преобразование систем с управлением, дифференциально-геометрический подход, символьные вычисления.

Введение

Многие алгоритмы построения обратной связи предполагают запись уравнений системы в переменных состояния, причём правая часть не должна содержать производных управлений [1]. Однако для описания систем с управлением также используют уравнения отображения вход–выход. Данная работа посвящена переходу от уравнений вход–выход к уравнениям в переменных состояния с использованием систем символьных вычислений. Такой переход позволяет в дальнейшем использовать известные алгоритмы построения обратных связей.

1. Задача реализации отображения вход–выход

Существует два распространённых способа описания нелинейной системы с управлением: в виде уравнений отображения вход–выход и с помощью переменных состояния.

Работа выполнена при финансовой поддержке программы «Развитие научного потенциала высшей школы (2009–2010 г.)» (проект 2.1.1/227) и грантов РФФИ 10-07-00617 и 08-01-00203.

В первом случае выход ($y \in \mathbb{R}^p$) непосредственно связывается с входом ($u \in \mathbb{R}^m$) при помощи дифференциальных уравнений:

$$y_i^{(k_i)} = \varphi_i(t, y, \dot{y}, \ldots, y^{(k-1)}, u, \dot{u}, \ldots, u^{(s)}), \qquad i = \overline{1, p}. \tag{1}$$

В системе (1) функции φ_i зависят от выходов y_j и их производных до порядка не выше $k_j - 1$, $j = \overline{1, p}$, и от входов u_q, $q = \overline{1, m}$, и их производных до порядка не выше s.

Уравнения системы (1) называют *уравнениями отображения вход–выход*.

Второе описание кроме переменных входа и выхода использует дополнительные переменные $x = (x_1, \ldots, x_n)$, которые называют *переменными состояния*:

$$\dot{x} = f(t, x, u, \dot{u}, \ldots, u^{(r_0)}), \quad x \in \mathbb{R}^n, \ u \in \mathbb{R}^m, \tag{2}$$

$$y = h(t, x, u, \dot{u}, \ldots, u^{(r)}), \quad y \in \mathbb{R}^p. \tag{3}$$

Вход u изменяет состояние x системы в соответствии с системой (2), которую называют *уравнениями состояния*.

Выражения (3), связывающие выходы с переменными состояния, определяют замену переменных, позволяющую перейти от описания (2)–(3) к описанию (1). *Задача реализации* — это задача перехода от описания (1) к эквивалентному описанию (2)–(3), т.е. поиск системы (2)–(3) и таких функций

$$x = X(t, y, \dot{y}, \ldots, y^{(k_0-1)}, u, \dot{u}, \ldots, u^{(s_1)}), \tag{4}$$

что при подстановке выражений (3) в систему (1) с учётом (2) и при подстановке (4) в систему (2)–(3) при условии (1) получаются верные тождества.

От описания (1) к описанию (2)–(3) можно перейти с помощью следующей замены:

$$x_1 = y_1, \ldots, \ x_{k_1} = y_1^{(k_1-1)}, \ x_{k_1+1} = y_2, \ \ldots, \ x_{k_1+\ldots+k_p} = y_p^{(k_p-1)}.$$

Однако в этом случае уравнения (2) будут зависеть от производных управления u до порядка s включительно. В то же время желательно, чтобы максимальный порядок производной управления был как можно ниже [2]. Будем искать систему (2)–(3) так, чтобы правые части уравнений в (2) не зависели от производных управлений:

$$\dot{x} = f(t, x, u), \quad x \in \mathbb{R}^n, \ u \in \mathbb{R}^m. \tag{5}$$

2. Теоретическое обоснование алгоритма

Сформулируем полученные ранее результаты, устанавливающие возможность приведения системы (1) к виду (5)–(3), а также свойства функций h_j, $j = \overline{1,p}$, из (3).

Пусть $\mathcal{F}$ — кольцо гладких функций, каждая из которых зависит от t, переменных y, u и некоторого конечного числа их производных. Рассмотрим следующий модуль 1-форм над кольцом $\mathcal{F}$:

$$\mathcal{H}_1 = \mathrm{span}_{\mathcal{F}}\{dt, dy_1, d\dot{y}_1, \dots, dy_1^{(k_1-1)}, \dots, dy_p^{(k_p-1)}, du_1, \dots, du_m^{(s)}\}.$$

Для 1-формы $\omega \in \mathcal{H}_1$ обозначим через $\dot{\omega}$ производную в силу системы (1). Определим по индукции

$$\mathcal{H}_{k+1} = \{\omega \in \mathcal{H}_k : \ \dot{\omega} \in \mathcal{H}_k\}, \qquad k \geq 1.$$

Можно показать [3], что $\mathcal{H}_k, k \geq 1$, есть модули над $\mathcal{F}$.

Для каждого выхода y_i обозначим через $\varkappa_i$ минимальный порядок производной в силу системы (1), для которого $y_i^{(\varkappa_i)}$ зависит от $u_q^{(s)}$ для некоторого $q = \overline{1,m}$. Если такого $\varkappa_i$ не существует, положим $\varkappa_i = \infty$.

Теорема 1 ([3]). *а) Реализация вида (5)–(3) локально существует для системы (1) тогда и только тогда, когда модуль имеет базис из точных 1-форм. При этом в качестве компонент векторной функции X из (4) можно выбрать функции, дифференциалы которых вместе с dt образуют базис $\mathcal{H}_{s+1}$.*

Если такая реализация существует, то
б) $n = k_1 + \ldots + k_p$;
в) функция $y_i = h_i$ в (3) зависит только от t, x, если $\varkappa_i > s$, и зависит от $t, x, u, \dot{u}, \dots, u^{(s-\varkappa_i)}$, если $\varkappa_i \leq s$.

Таким образом, для решения задачи реализации нужно найти базис модуля $\mathcal{H}_{s+1}$ и проверить интегрируемость соответствующего кораспределения. Даже при условии использования компьютерной системы аналитических вычислений это достаточно нетривиальная задача. Удобнее сформулировать условия существования описания вида (5)–(3) и функции X на языке векторных полей.

Обозначим через $\eta_1, \dots, \eta_N$ переменные

$$t, y_1, \dot{y}_1, \dots, y_1^{(k_1-1)}, y_2, \dot{y}_2, \dots, y_p^{(k_p-1)}, u_1, \dots, u_1^{(s-1)}, u_2, \dots, u_m^{(s-1)},$$

дифференциалы $d\eta_1, \dots, d\eta_N$ которых порождают $\mathcal{H}_1$.

Лемма 1 ([4]). *Для любого $j \geq 1$ базис модуля $\mathcal{H}_j$ можно выбрать из 1-форм вида $\sum_{l=1}^{N} f_l d\eta_l$, где функции $f_1, \ldots, f_N$ зависят только от*

$$t, y_1, \dot{y}_1, \ldots, y_1^{(k_1-1)}, \ldots, y_p^{(k_p-1)}, u_1, \ldots, u_1^{(j+s-2)}, \ldots, u_m^{(j+s-2)}. \tag{6}$$

Обозначим через $\mathcal{E}$ пространство с координатами (6) при $j = s+1$. По лемме 1 для любого $j = \overline{1, s+1}$ базис модуля $\mathcal{H}_j$ можно выбрать из 1–форм на $\mathcal{E}$. Рассмотрим на $\mathcal{E}$ векторное поле

$$D = \frac{\partial}{\partial t} + \sum_{\alpha=1}^{p} \sum_{l_\alpha=0}^{k_\alpha-2} y_\alpha^{(l_\alpha+1)} \frac{\partial}{\partial y_\alpha^{(l_\alpha)}} + \sum_{\alpha=1}^{p} \varphi_\alpha \frac{\partial}{\partial y_\alpha^{(k_\alpha-1)}} + \sum_{\beta=1}^{m} \sum_{k=0}^{2s-2} u_\beta^{(k+1)} \frac{\partial}{\partial u_\beta^{(k)}},$$

и распределения

$$\mathcal{D}_i = \operatorname{span}_{C^\infty(\mathcal{E})}\{B_1, \ldots, B_m, \operatorname{ad}_D B_1, \ldots, \operatorname{ad}_D B_m, \ldots, \operatorname{ad}_D^{i-1} B_m\},$$
$$i = 1, 2, \ldots,$$

где

$$B_\beta = \frac{\partial}{\partial u_\beta^{(s)}}, \qquad \beta = 1, \ldots, m.$$

Сопряженное дополнение к модулю $\mathcal{D}_i$ в $\mathcal{H}_0$ обозначим через $\mathcal{D}_i^T$:

$$\mathcal{D}_i^T = \{\omega \in \mathcal{H}_0 \colon \ \forall X \in \mathcal{D}_i \quad X \rfloor \omega \equiv 0\}.$$

Теорема 2 ([4]). *Для $j = \overline{2, s+1}$ имеем $\mathcal{H}_j = \mathcal{D}_{j-1}^T$.*

Из теорем 1 и 2 получаем

Следствие 1. *Реализация вида (5)–(3) локально существует для системы (1) тогда и только тогда, когда распределение $\mathcal{D}_s$ интегрируемо. При этом в качестве компонент векторной функции X из (4) можно взять набор первых интегралов распределения $\mathcal{D}_s$, который вместе с t образует функционально независимую систему.*

3. Алгоритм решения задачи реализации

Дано: система (1).

Требуется найти замену переменных (4), приводящую систему (1) к виду (5), (3) и найти реализацию (5), (3).

Алгоритм:

$i := 1$.

Вычисление распределения $\mathcal{D}_i$ и проверка его интегрируемости.

Пока (распределение $\mathcal{D}_i$ интегрируемо) и ($i < s$), выполнить:

$i := i + 1$.

Вычисление распределения $\mathcal{D}_i$ и проверка его интегрируемости.

Конец цикла.

Если (распределение $\mathcal{D}_i$ интегрируемо) и ($i = s$), то

Выбор функционально независимых первых интегралов $\mathcal{D}_i$, формирование замены (4).

Обращение замены (4), вычисление выражений (3).

Дифференцирование замены переменных (4) в силу системы (1) с последующим исключением выходов y с помощью выражений (3).

Запись системы в виде (5).

Иначе

Невозможно найти реализацию вида (5), (3) средствами компьютерной алгебры, требуется дополнительное исследование.

Конец если.

4. Программная реализация

Приведённый выше алгоритм реализован в среде символьных вычислений MAPLE [5] в виде процедуры **realization_varchange**, осуществляющей поиск замены переменных (4), и программного кода, выполняющего построение реализации (5), (3).

Процедура **realization_varchange** включает в себя процедуру вычисления коммутатора векторных полей.

На вход процедуры **realization_varchange** подаётся система, записанная в переменную типа *list*, и параметры системы в переменной того же типа.

Далее производится формирование списка координат и преобразование производных y и u в имена (изначально производные считаются функциями). Затем формируются векторные поля D и $B_\beta = \frac{\partial}{\partial u_\beta^{(s)}}, \quad \beta = \overline{1, m}$.

Далее происходит выполнение алгоритма решения задачи реализации.

Для проверки интегрируемости распределения, порождённого векторными полями, вычисляются первые интегралы, общие для этих векторных полей (решается система в частных производных с помощью функции **pdsolve**), то есть интегрируемость проверяется непосредственно, а не по условиям Фробениуса. Это позволяет избежать возможной некорректной работы процедуры определения ранга **LinearAlgebra[Rank]** в случае функциональных матриц. Если проинтегрировать систему при помощи функции **pdsolve** не удаётся, то в соответствии с алгоритмом требуется вручную проверить условия Фробениуса. Распределение может являться интегрируемым, однако его первые интегралы при этом могут не выражаться аналитически.

Для поиска обратной к (4) замены переменных использовалась функция **solve**. Если обратная замена не может быть найдена, следует попробовать найти её вручную. Далее, после дифференцирования замены (4), подстановки выражений из системы (1) и выражений из обратной к (4) замены, получаем систему вида (5), (3), являющуюся искомой реализацией системы с управлением.

Список литературы

[1] Isidori A. Nonlinear control systems. London : Springer-Verlag, 2001.— 549 p. ↑[]

[2] Delaleau E., Respondek W. *Lowering the Orders of Derivatives of Controls in Generalized State Space Systems* // Journal of Mathematical Systems, Estimation and Control, 1995. **5**, no. 3, p. 1–27. ↑1

[3] Крищенко А. П., Четвериков В. Н. *Преобразования описаний нелинейных систем* // Дифференциальные уравнения, 2009. **45**, № 5, с. 706–715. ↑2, 1

[4] Крищенко А. П., Четвериков В. Н. *Минимальные реализации нелинейных систем* // Дифференциальные уравнения, 2010. **46**. ↑1, 2

[5] Аладьев В. В. Основы программирования в Maple. Таллин : Международная академия ноосферы, 2006.— 301 с. ↑4

A. V. Evseev. *Realization of non-linear dynamic systems.*

Abstract. Realizations of input-output differential equations for non-linear control systems are considered. Necessary and sufficient conditions for realization possibility are given. Conditions are formulated either in terms of vector fields and in terms of differential forms. Realization algorithm is built. Algorithm-based software package is created in symbolic computations system Maple.

Key Words and Phrases: transform of control systems, differential geometry approach, symbolic computations.

Образец ссылки на статью:

А. В. Евсеев. *Построение реализаций нелинейных динамических систем* // Программные системы: теория и приложения : электрон. научн. журн. 2011. № 2(6), с. 61–66. URL: http://psta.psiras.ru/read/psta2011_2_61-66.pdf