таточно низкий порядок, и независимо от шага интегрирования в любом случае будут решаться только две задачи Коши в противоположных направлениях к концам подынтервалов пристрелки.

ЛИТЕРАТУРА

1. Деккер К., Вервер Я. Устойчивость методов Рунге-Кутта для жестких нелинейных дифференциальных уравнений // Пер. с англ. – М., 1983. С. 200.

УДК 519.714

О.В. Герман, доцент; Д.В. Занько, ассистент

СИНТЕЗ УПРАВЛЕНИЯ СИСТЕМОЙ НА КОНЕЧНОМ ОРИЕНТИРОВАННОМ ГРАФЕ

The paper deals with the problem of synthesizing control algorithm for the finite discrete systems. A new method is described which uses normalization procedure in order to procedure the required control alongside with some regular labeling mechanism for the graph nodes.

Введение

Статья посвящена проблеме синтеза управления дискретной конечной системой. Различные точки зрения по этому вопросу сформулированы в [1–3]. Предлагается метод синтеза, использующий процедуру нормализации графа и некоторую регулярную процедуру разметки вершин-состояний.

1. Постановка задачи

Рассмотрим задачу синтеза управления системой, сформулированную на ориентированном конечном графе $G=(U,\overline{V})$, вершины которого определяют классы состояний, а дуги помечены символами операторов y_i, при реализации которых осуществляется переход из одного класса состояний в другой. Каждый класс будем описывать набором элементов из {0, 1, *}: 0, когда значение переменной состояния равно 0; 1, когда значение переменной состояния равно 1; "*", когда значение переменной состояния безразлично. Размерность вектора фиксирована и равна n – числу переменных состояния системы. Например, класс $S_i = <0*1*0>$ включает 4 состояния:

0 0 1 0 0
0 0 1 1 0
0 1 1 0 0
0 1 1 1 0,

которые получаем заполнением разрядов, отмеченных "*", 0 или 1. Допускаем также класс, содержащий лишь одно состояние.

Определение 1. Два класса называются ортогональными, если они отличаются хотя бы в одном разряде. Т.е. если в одном и том же разряде эти классы имеют значения переменных состояния системы, равные 0 и 1.

Определение 2. Графом нормального вида будем называть граф, который представлен множеством попарно ортогональных вершин.

В качестве иллюстрации мы будем использовать систему, состоящую из следующих классов:

1 0 1 * *
1 1 * * 1
* 1 * 1 0
0 * * 0 *
* 0 0 1 *
0 0 1 1 *

Пусть граф переходов этой системы имеет вид,представленный на рис.

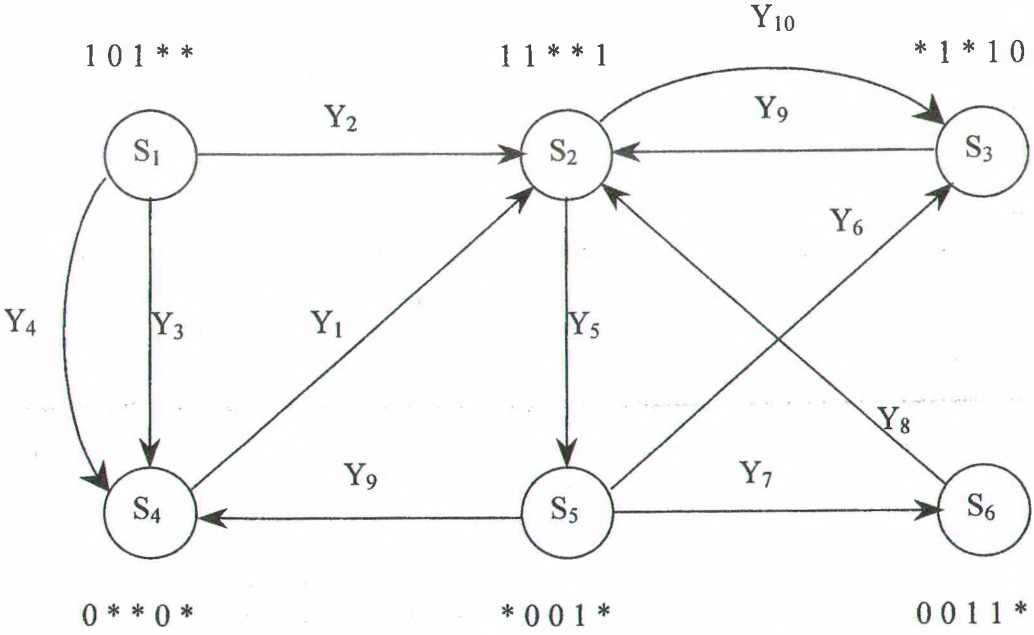

Рис.

Договоримся об интерпретации управлений Y_i, реализуемых в состояниях S_k. Пусть текущее состояние $S^t = < x_1^t\ x_2^t \dots x_n^t >$. Состояние S^t допустимо, если оно принадлежит одному и только одному классу S_i ($S^t \in S_i$). В силу того, что S нормальна, произвольное S^t может принадлежать не более чем одному классу (но может не принадлежать ни одному из классов). Например, $S^t = <00000> \in S_4 = <0**0*>$ и, следовательно, S^t допустимо в этом случае. Если управление Y_i реализуется в классе S_j, реализуя при этом переход в S_k, то оно не изменяет значения разрядов, отличных от "*" в S_k. Например, если Y_1 применяется к $S^t = <00000>$, то переходим в состояние $<11001> \in S_2$, не изменяя 3-й и 4-й разряды.

Пусть даны два допустимых состояния S^0 и S^f из разных классов. Считаем S^0 – начальным состоянием, а S^f – конечным состоянием. Спрашивается, существует ли конечная последовательность C_{0f} реализуемых управлений такая, что под ее воздействием система перейдет из состояния S^0 в S^f.

Для решения этой задачи следует заметить, что условие существования пути в графе из S^0 в S^f необходимо, но не достаточно. Например, пусть $S^0 = <11111>$ и $S^f = <01110>$. Имеется прямой переход из $S_2 \in S^0$ в $S_3 \in S^f$, однако управление Y_{10}, примененное к S^0, порождает состояние $<11110> \neq S^f$.

Сначала мы опишем процедуру решения задачи, а затем докажем ее корректность, под которой понимаем способность процедуры решить (в заданном выше смысле) поставленную задачу синтеза для произвольной системы S нормального вида и допустимых состояний S^0 и S^f.

2. Описание процедуры синтеза управления

Процедура синтеза строит цепочку управлений S, стартуя из конечной вершины S^f в направлении начальной вершины S^0. «Движение» происходит в направлении, противоположном ориентации дуг, соединяющих просматриваемые вершины. При этом на каждом такте выделяется одна текущая вершина – $\widetilde{S}^t$, на базе которой выполняется попытка помечать каждую вершину S_i, соединенную с $\widetilde{S}^t$ дугой, выходящей из S_i. Если попытка неудачна, то вершина S_i исключается из списка просмотра L, в противном случае она включается в L вместе с меткой. Если $L = \varnothing$, то алгоритм заканчивается ответом: «нельзя построить требуемый путь». В противном случае S_i включается в L со своей меткой, и при условии, что S_i уже не содержится в L с той же меткой или более общей (метка L' более общая, чем L'', если в каждом разряде, где L' содержит 0 или 1, L'' также содержит 0 или 1). Затем список L просматривается заново. Из него выбирается вершина, которой приписана метка с наибольшим числом вхождений символа "*". Если метка вершины S_z целиком состоит из "*", то решается отдельно задача о наличии маршрута из S^0 в S_z. Если такой маршрут найден, то исходная задача получает требуемое решение. В противном случае S_z удаляется из L. Если метка вершины S_z не состоит из одних только символов "*", то процедура повторяется до тех пор, пока L не опустеет либо не будет достигнута исходная вершина – состояние S^0. Прежде чем дать формальное описание алгоритма, изложим его действие на примере графа на рис. 1.

Пусть $S^0 = <00000>$, $S^f = <11110>$.

1. $L = \{[S_3, 11110]\}$.

Стартуем из S_3. Из S_3 можно пометить S_2 меткой <* * 1 * *>, пометить S_5 невозможно. Поясним этот шаг. Управление $Y10$ реализуемо из класса S_2 из состояния <1 1 1 0 1> либо из <1 1 1 1 1>. При образовании метки для $\widetilde{S}_2$ заменим звездочкой те разряды, которые не равны "*" в классе S_2, т.е. разряды 1-й, 2-й и 5-й, а также те разряды, которые не равны в S_3, т.е. 2-й, 4-й и 5-й.

Таким образом, метка <* * 1 * *>, приписанная S_2, означает, что какова бы ни была цепочка С, при попадании в класс S_2 третий разряд должен быть установлен в "1", значение остальных разрядов считается «безразличным».

Из класса S_5 нельзя достичь $S^f = <11110>$, т.к. управление Y_6 не изменяет 3-й разряд.

Итак, преобразуем L к виду

$L = \{[S_2, **1**]\}$.

2. Из S_2 можно пометить вершины S_1, S_3, S_4, S_6.

Список L примет вид

$$L = \begin{Bmatrix} [S_1, & *****] \\ [S_3, & **1**] \\ [S_4, & **1**] \\ [S_6, & *****] \end{Bmatrix}.$$

Метка S_1 строится из того, что 3-й разряд установлен в S_1 в "1", поэтому его устанавливать не требуется, чего не скажешь про S_3 и S_6.

Итак, в силу предварительных замечаний нужно просто проверить, что S_4 связано с S_1 либо S_6. Если S_4 и S_6 не связаны путем с S_1, то элемент $[S_1, *****]$ и $[S_6, *****]$ удаляются из L.

В нашем случае находим требуемое решение:

$$S^0 \in S_4 \xrightarrow{Y_4} S_1 \xrightarrow{Y_2} S_2 \xrightarrow{Y_{10}} S_3 \in S^f$$

и

$$S^0 \in S_4 \xrightarrow{Y_1} S_2 \xrightarrow{Y_5} S_5 \xrightarrow{Y_7} S_6 \xrightarrow{Y_8} S_2 \xrightarrow{Y_{10}} S_3 \in S^f.$$

3. Формальное описание алгоритма синтеза

Пусть $L_t = \left\{ < S_i, F_{iz} > \middle| \, i = 1..k^t \right\}$ – список помеченных вершин S_i графа переходов и их меток F_{iz} на шаге t = 0, 1, 2 … T. Для t = 0 L_0 содержит единственную вершину $S_r \supset S^f$; S^f – целевое состояние системы. Процедура синтеза реализуется систематическим применением следующих шагов.

Шаг 1. Если $L_t = \varnothing$ или содержит только использованные метки, то конец – синтезировать требуемую цепочку нельзя.

Если $L_t \neq \varnothing$, то выбираем, например, член $< S_i, F_{iz} >$, в котором F_{iz} содержит наибольшее число "*" (при неоднозначности выбираем любой).

Пусть $< S_i, F_{iz} >$ таков, что $F_{iz} = < ** ... * >$. Построим путь из S^0 в S_i. Если это возможно, то задача решена и искомая цепочка получена. Если нет, то удаляем член $< S_i, F_{iz} >$ из L_t. После этого повторяем шаг 1 сначала.

Если $F_{iz} \neq < ** ... * >$, то переход на шаг 2.

Шаг 2. Из $< S_i, F_{iz} >$ помечаем все вершины S_j, связанные с S_i дугой, заканчивающейся в S_i. Пусть $< S_j, F_{jz} >$ – один из новых членов. Добавляем его в L_t, тогда и только тогда.

(i) F_{jz} не является общей меткой для S^0;

(ii) в L_t нет члена $< S_j, F'_{jz} >$ с более общей меткой F'_{jz}.

В случае, если (i) ложно, алгоритм завершается построением нужной цепочки, т.к. достигли начальную вершину-состояние.

Пусть (i) и (ii) – истинны. Удаляем из L_t члены менее общие, чем $< S_j, F_{jz} >$.

Переходим на шаг 3.

Шаг 3. Помечаем $< S_i, F_{iz} >$ как использованный член, полагаем $t = t + 1$ и переходим на шаг 1.

4. Оценка сложности алгоритма синтеза

Пусть m – число вершин графа; n – число переменных состояний. Если заметить, что очередная метка попадает в L, если и только если она содержит на один символ "*" больше, чем существующие метки для этой вершины, то для каждой вершины может быть не более n попаданий соответствующих членов в список L. Пусть на каждой итерации добавляется только один новый член в L. Ясно, что через $m*n$, в худшем случае, итераций либо будет построено решение, либо одна из вершин графа будет рассмотрена. Отсюда верхняя оценка числа итераций в худшем случае составляет полностью $O(m^2 n)$.

Остается только заметить, что одна и та же вершина не может быть одновременно представлена в списке L двумя именами, т.к. «старая» метка не может быть более общей, чем новая, ибо при распространении меток «вдоль пути» число символов "*" может только возрастать. Поэтому при возврате к ранее помеченной вершине такой член либо удалит «старый», либо не будет введен в L.

ЛИТЕРАТУРА

1. Ефимов Е.И. Решатели интеллектуальных задач. – М.: Наука, 1982.
2. Попов Э. В., Фирдман Г.Р. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта. – М.: Наука, 1976.
3. Герман О.В., Семерюк Д.В. Одна полиномиально разрешимая задача синтеза поведения интеллектуального робота // Автоматика и телемеханика. – 2001. № 2. С. 15–24.