

Н.Е. Сергеев, Е.Р. Мунтян, А.А. Целых, А.Н. Самойлов

**ОБОБЩЕНИЕ ГРАФОВ СИТУАЦИЙ НА ОСНОВЕ СПИСКОВОГО
АЛГОРИТМА СВЕРТКИ ДЛЯ ЗАДАЧ СИТУАЦИОННОГО УПРАВЛЕНИЯ***

В случае непрерывного изменения параметров текущих ситуаций при управлении объектами возможна дискретизация как самих информационных значений, так и управляющих воздействий. При этом необходимо избежать потери качественных характеристик управления. Предлагается для представления текущих, целевых и эталонных ситуаций использовать графовую парадигму. Для рассматриваемого спектра задач такие графы ситуаций могут иметь достаточно большие размеры, что представляет собой проблему в условиях ограниченных возможностей носителей мобильных подсистем управления. Управление группой объектов также требует алгоритмов представления решений по результатам частичной обработки доступной информации. Предлагается оценивать не последовательность ситуаций после одношаговых переходов, а в целом ситуацию, в которой система может оказаться после любого из переходов. При этом возможно учитывать критерии, которые позволили бы в некоторых направлениях обобщать ситуацию не на один, а сразу на несколько шагов. Выполнен анализ возможных способов представления графа ситуаций, отмечены их достоинства и недостатки для различных видов графов. Обоснована целесообразность представления графов в виде списков ребер. В качестве способа обработки графов в работе предлагается алгоритм свертки графа. Суть алгоритма заключается в разделении графа на кластеры разных уровней определенным способом, представлении каждого кластера графа в виде списков ребер и поглощении текущего кластера кластером следующего уровня. Особенности использования предложенного алгоритма проиллюстрированы на примере обобщения ситуационного графа. Предложенный алгоритм можно использовать при планировании и управлении как одиночными объектами, так и группой объектов. В случае наличия группы управляемых объектов задача построения графа ситуаций может быть распределена между ними с представлением поглощенных вершин в виде списков ребер.

Граф; ситуация; управление; список; алгоритм; свертка.

N.E. Sergeev, E.R. Muntyan, A.A. Tselykh, A.N. Samoylov

**SITUATION GRAPH GENERALIZATION FOR SITUATION AWARENESS
USING A LIST-BASED FOLDING ALGORITHM**

In this paper we make an assumption that the discretization of both the values of information and control actions is possible in case of a continuous change in the parameters of the current situations when controlling the objects. It is important to avoid the loss of qualitative characteristics of control. It is proposed to represent the current, target and reference situations using a graph paradigm. For the considered range of tasks such graphs of situations can be pretty large, that is the problem especially in conditions of limited possibilities of mobile media control subsystems. Control of a group of objects also requires the decision algorithms based on the results of partial processing of available information. This paper proposes to evaluate not the sequence of situations after step-by-step transitions, but the overall situation, in which the system may find itself after any transitions. It is possible to consider the criteria that allow in some directions to summarize the situation by several steps ahead. In this paper we performed an analysis of possible ways of representing the graph situations considering their advantages and disadvantages for different types of graphs. We explore the ways to represent graphs using lists of edges. In the paper we propose the algorithm of graph convolution or generalization. The essence of the algorithm

* Работа выполнена в рамках внутреннего гранта ЮФУ № ВнГр-07/2017-28.

is the partitioning of the graph into clusters of different levels in a certain way, the representation of each cluster of the graph as lists of edges and absorption of the current cluster by the cluster of the next level. Particular usage of the proposed algorithm is illustrated on the example of the situation graph generalization.

Graph; situation; control; list; algorithm; convolution.

Введение. Задачи ситуационного управления рассматриваются для самого широкого круга областей [1–6], в том числе для управления транспортом, производственными процессами и роботами. В процессе мониторинга функционирования систем часто удаётся выделить некоторые дискретные состояния [7]. Они могут быть обусловлены дискретностью некоторых параметров системы или дискретностью самого процесса мониторинга, либо процесса управления. Иногда «дискретизация» полезна и для систем с непрерывно меняющимися значениями параметров, в том числе и для искусственной фиксации как отдельных атрибутов, так и для ситуаций в целом.

Для мониторинга и управления такими дискретными или дискретизированными системами удобно использовать графы [8–10]. Граф ситуаций [11, 12] может быть построен заранее или строиться в процессе мониторинга. Он может быть конечным или условно бесконечным. Условность бесконечности здесь заключается в отсутствии возможности в настоящее время оценить его размеры. Часто возникает ситуация, когда количество возможных вариантов перехода в следующую ситуацию достаточно велико и оценить ситуацию, в которой находится объект управления, весьма трудно. В этом случае может оказаться полезным не исследование движения по графу ситуаций, начиная с ближайшего последующего по каждому направлению до возможного достижения целевой ситуации (ситуационное движение [13]), а последовательная оценка окружающих возможных ситуаций. Причем, очевидно правильно замечает Д.А. Пospelov [7, 14], что каждый сделанный выбор ограничивает свободу последующего. Особенно ситуацию обостряет неудачный выбор.

Авторы предлагают оценивать не последовательность ситуаций после одношаговых переходов, а в целом ситуацию, в которой мы окажемся после любого из сделанных выборов. При этом можно учитывать критерии, которые позволили бы в некоторых направлениях обобщать ситуацию не на один, а сразу на несколько шагов. Например, если к следующим шагам ведёт только один путь и только из данной вершины.

В этом случае, может показаться, что мы теряем информацию о том, как и какой выбор может повлиять на последующую ситуацию. Для исключения возможных потерь предусматривается сделать ситуацию обобщения обратимой.

Поскольку все возможности перейти из текущей в другие ситуации являются свойствами этой ситуации, то рассмотрение их можно считать процессом обобщения. Для именования полученной обобщенной ситуации будем использовать термин лингвистики – гипероним (понятие, выражающее более общую сущность). В отношении некоторого множества объектов гиперонимом является понятие, отражающее надмножество к исходному. Частный результат обратной операции назовём гипонимом [15]. Процесс обобщения не мешает параллельно рассматривать и отдельно взятые вершины графа ситуаций.

Представление графа ситуаций. Граф ситуаций может быть представлен с помощью матриц смежности или инцидентности. В случае представления графов в виде матриц [8] возникают трудности в работе с графами больших размеров. Кроме того, очевидна избыточность требуемого ресурса памяти ЭВМ (в приведенных матрицах немалое количество ячеек чаще всего представлены нулями). В связи с этим в качестве способа представления ситуационных графов авторы использовали списки, которые представляют собой упорядоченные наборы данных [16, 17].

Предлагается использование списков и для решения задач обобщения. Списки встречаются двух видов: списки смежности и списки ребер. В первом случае для каждой вершины графа хранится список всех ее соседей. В списке ребер присутствует список пар вершин графа. Каждой паре вершин соответствует определенное ребро.

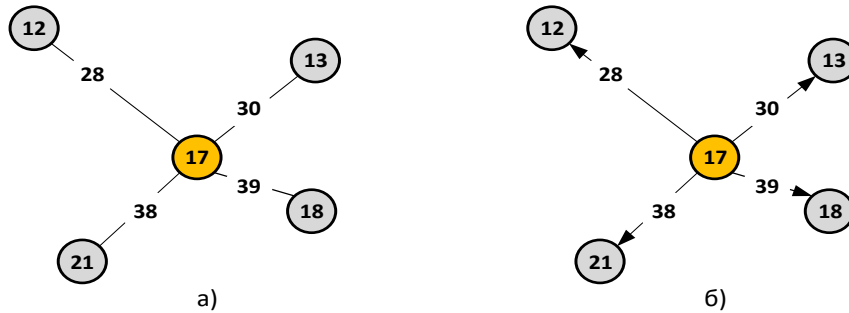


Рис. 1. Неориентированный граф (а) и ориентированный граф (б)

Рассмотрим порядок перехода от представления графа (рис. 1) в виде матриц (табл. 1 и 2) к спискам (рис. 2 и 3).

Таблица 1

Матрица смежности

	12	13	17	18	21
12	0	0	1	0	0
13	0	0	1	0	0
17	1	1	0	1	1
18	0	0	1	0	0
21	0	0	1	0	0

Таблица 2

Матрица инцидентности

	28	30	38	39
12	1	0	0	0
13	0	1	0	0
17	1	1	1	1
18	0	0	0	1
21	0	0	1	0

При этом представление графа, изображенного на рис. 1,а, в виде списков смежности X_c может выглядеть так: $X_c = (17, 12, 13, 18, 21), (12, 17), (13, 17), (18, 17), (21, 17)$ или как показано на рис. 2.

Здесь каждый элемент списка представляет собой две ячейки, левая из которых является указателем на текущую ячейку, например, 17, а правая – на соседнюю ячейку с номером 12, как показано на рис. 2.

При описании неориентированных графов происходит своего рода дублирование списков, например, для списка $(17, 12, 13, 18, 21)$ обратными будут списки $(12, 17), (13, 17), (18, 17), (21, 17)$. Списки ребер для графа, изображенного на рис. 1,а, будут иметь вид:

$$X_p = (17, 12), (17, 13), (17, 18), (17, 21), (12, 17), (13, 17), (18, 17), (21, 17).$$

В рассматриваемом примере граф содержит 4 ребра, поэтому список ребер включает 8 строк в случае неориентированного графа. Например, запись $(17, 13)$ означает ребро, которое соединяет вершины 17 и 13. Аналогично список ребер можно представить в виде изображения на рис. 3.

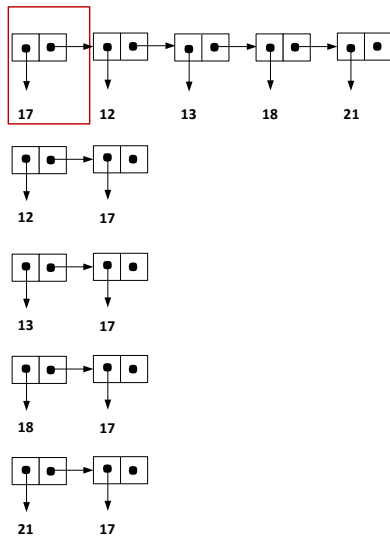


Рис. 2. Список смежности графа

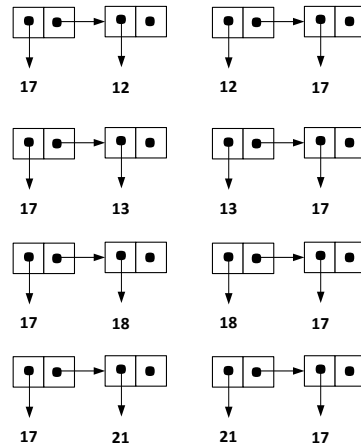


Рис. 3. Список ребер графа

Рассмотрим аналогичные формы представления ориентированного графа на рис. 1,б. Матрицы смежности и инцидентности для такого графа представлены табл. 3 и 4. При этом список смежности X_c ориентированного графа может выглядеть так: $X_c = (17, 12, 13, 18, 21)$ или как показано на рис. 4.

Таблица 3

Таблица 4

Матрица смежности

	12	13	17	18	21
12	0	0	1	0	0
13	0	0	1	0	0
17	0	0	0	0	0
18	0	0	1	0	0
21	0	0	1	0	0

Матрица инцидентности

	28	30	38	39
12	0	0	0	0
13	0	0	0	0
17	1	1	1	1
18	0	0	0	0
21	0	0	0	0

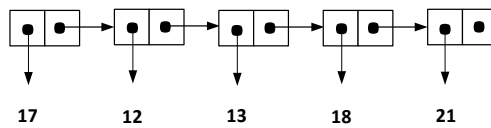


Рис. 4. Список смежности ориентированного графа

Списки ребер для графа, изображенного на рис. 1,б, имеют вид: $X_p = (17, 12), (17, 13), (17, 18), (17, 21)$. Графическое изображение X_p совпадает с частью списков ребер на рис. 3.

Рассмотрим операции со списками. В [16] описываются функции разделения и составления списка. Для функции разделения списка предлагается использовать операторы: $hd(x)$ и $tl(x)$, обозначающие начало или заголовок списка x и остаток (хвост) списка x .

Рассмотрим пример использования данных функций для представления списка $x = (17, 12, 13, 18, 21)$. Операторы примут значения: $hd(x) = (17)$ и $tl(x) = (12, 13, 18, 21)$. Если полученный остаток $tl(x)$ представить в качестве нового списка $x_1 = (12, 13, 18, 21)$, то получим, что $hd(x_1) = (12)$, а $tl(x_1) = (13, 18, 21)$. Аналогичным образом можно оформить списки x_2 и x_3 , чтобы получить список из одной вершины. Обобщив рассмотренный пример, отмечаем, что список x может быть представлен в виде:

$$hd(x), hd(tl(x)), hd(tl(tl(x))), hd(tl(tl(tl(x)))), \dots$$

Для составления или объединения списка применяется функция $cons$. Если принять, что a – элемент или список и b – список, то функция $cons(a, b)$ представит собой новую исходную ячейку списка.

Если в нашем случае $a = 17$ и $b = (12, 13, 18, 21)$, то функция примет вид: $cons(a, b) = (17, 12, 13, 18, 21)$.

Список $z = (z_1, z_2, z_3, z_4, \dots, z_n)$ можно построить по формуле

$$cons(z_1, cons(z_2, cons(z_3, cons(z_4, \dots, cons(z_n) \dots)))).$$

Можно сделать вывод об обратном отношении операторов $hd(x)$ и $tl(x)$ к $cons$: $hd(cons(a, b)) = a$, $tl(cons(a, b)) = b$. В то же время не всегда действуют обратные правила. Значением $cons(hd(x), tl(x))$ является новая ячейка памяти, которая содержит такие же указатели, как и исходная ячейка x . Т.е. результатом функции $cons(hd(x), tl(x))$ является копия ячейки x , а не сама ячейка x .

Как отмечалось выше, существует возможность использования двусторонне связанных списков, описанная в [17]. Элемент такого списка представляет собой запись, которая включает три поля, в отличие от предыдущего варианта, где использовалось два поля. Это поля: ключ key и два указателя $next$ (следующий) и $prev$ (предыдущий), как показано на рис. 5.

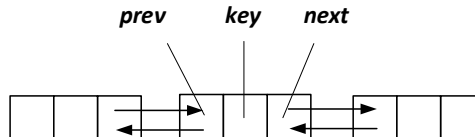


Рис. 5. Обозначение элемента двусторонне связанного списка

Рассмотрим список $x = (x_1, x_2, x_3, x_4, \dots, x_n)$. Если x_i – элемент списка, то $next(x_i)$ указывает на следующий элемент списка, а $prev(x_i)$ представляет указатель предшествующего элемента. Если $prev(x_i) = NIL$, то у x_i нет предшествующего элемента и он является головой списка $hd(x)$. В случае, если $next(x_i) = NIL$, то x_i – последний элемент списка и он является концом списка $tl(x)$.

В рассмотренных ранее списках смежности присутствуют списки из двух и пяти элементов. Список из пяти элементов $x = (17, 12, 13, 18, 21)$ подразумевает, что мы можем попасть из вершины 17 в вершины 12, 13, 18 и 21. При этом нет возможности перейти из вершины 12 в 13, кроме как через вершину 17. Соответственно, такой двусторонне связанный список примет вид (рис. 6). Т.е. фактически мы перешли к спискам ребер, рассмотренным выше.

Отсюда можно сделать вывод о целесообразности представления графов в виде списков ребер.

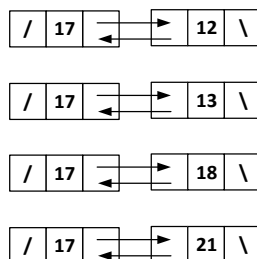


Рис. 6. Представление двусторонне связанного списка x

Описание алгоритма свертки графа. Авторами предлагается алгоритм свертки или поглощения графа, который заключается в выполнении последовательности действий (1–8).

Введем следующие обозначения:

- ◆ $Nodes = \{1..(n-1)\}$ – множество всех доступных вершин графа (ситуаций);
- ◆ $Nodes_j$ – одна вершина (j – номер вершины графа ситуаций);
- ◆ $cNode$ – исходная вершина графа;
- ◆ $Cluster_i$ – кластер i -го уровня;
- ◆ $Neighbors$ – множество соседних вершин для $cNode$;
- ◆ C – количество соседних вершин.

Алгоритм свертки графа ситуаций:

- 1) задаем уровень кластера ($i=0$);
- 2) определяем исходную вершину графа для свертки ($cNode = Nodes_j$);
- 3) выполняем функцию поиска соседних вершин $Neighbors$ для вершины $cNode$ ($C = get\ Neighbors(cNode)$);
- 4) если $C \neq 0$, то $i = i + 1$;
- 5) определяем кластер i -го уровня как совокупность исходной вершины $cNode$ и ее соседей $Neighbors(cNode)$, при этом кластер является списком $Cluster_i = cons(cNode, Neighbors(cNode))$, где $hd(Cluster_i) = cNode$, $tl(Cluster_i) = Neighbors(cNode)$;
- 6) объединяем вершины кластера i -го уровня в вершину $cNode$ ($cNode = Cluster_i$), получая тем самым новую исходную вершину $cNode$;
- 7) возвращаемся в п.4;
- 8) если $C=0$ (после п.4), то завершение работы алгоритма.

Пример использования алгоритма. Поясним особенности использования описанного алгоритма применительно к графу, изображенному на рис. 7. Исходный граф содержит n вершин $Nodes_j$, начальная из которых имеет номер 1, а конечная – номер 42. Для соединения этих вершин используются ребра 1-73. Заметим, что матрица инцидентности для такого графа будет иметь размерность 42×73 , а матрица смежности – 42×42 , причем многие из ячеек будут содержать нули. Обработка таких матриц, как говорилось ранее, весьма трудоемкая задача.

За исходную вершину графа принята вершина 17 ($cNode$), с которой связаны $Neighbors$ вершины 12, 13, 18 и 21.

Представим вершины графа в виде кластеров. В соответствии с алгоритмом свертки кластер i -го уровня $Cluster_i$ ($i=1$) включает исходную вершину $cNode = 17$ и соседние с ней вершины $Neighbors = 12, 13, 18, 21$. Перечисленные выше вершины могут быть представлены в виде списков ребер $Cluster_{1,4} = (17, 12), (17, 13), (17, 18), (17, 21)$. В соответствии с приведенным ранее алгоритмом выполним свертку вершин кластера 1-го уровня в вершину с номером больше конечной вершины исходного графа ($cNode = 43$).

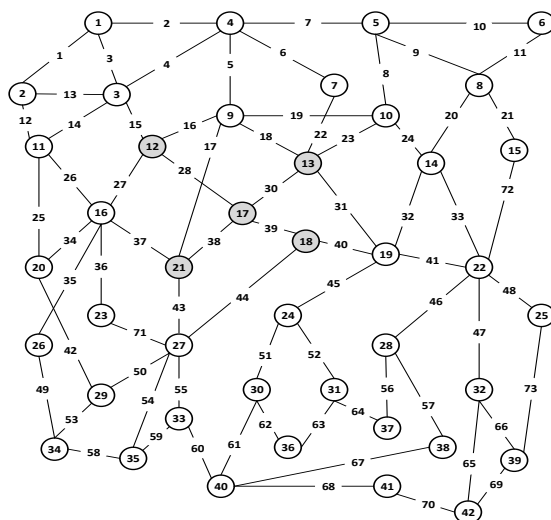


Рис. 7. Исходный граф с выделенным кластером 1-го уровня

Теперь для исходной вершины $sNode = 43$ кластера i -го уровня ($i=2$) соседними являются вершины с номерами: 3, 9, 7, 10, 19, 27, 16. В случае поглощения указанных вершин появляется список кластера 2-го уровня $Cluster_{2-8} = (43, 3), (43, 7), (43, 9), (43, 10), (43, 19), (43, 16), (43, 27), (9, 10)$.

Следует обратить внимание на тот факт, что некоторые из поглощаемых вершин связаны между собой несколькими ребрами, например, в списке $(43, 9)$ вершины связаны тремя ребрами 16, 17 и 18. Также следует обратить внимание на ситуацию, когда две или несколько вершин, которые будут поглощаться, имеют связь не только с исходной вершиной, но и между собой. Для корректного отображения поглощенных вершин следует учитывать подобные ситуации в списках, например, список $(9, 10)$.

Далее в качестве исходной выступает вершина $sNode = 44$, которую окружают вершины 1, 2, 4, 5, 11, 14, 20, 22, 23, 24, 26, 29, 33 и 35.

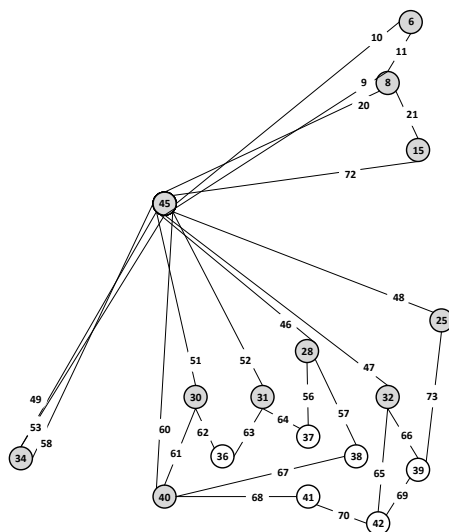


Рис. 8. Граф после свертки с выделенным кластером 4-го уровня

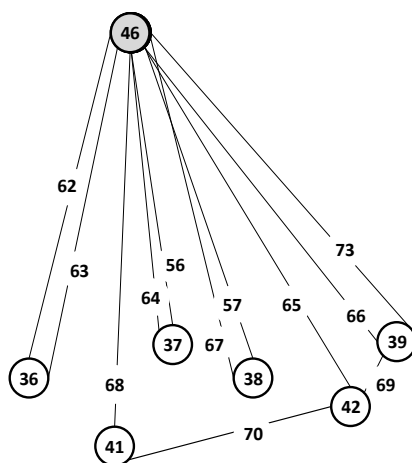


Рис. 9. Граф после свертки с выделенным кластером 5-го уровня

Списки, включающие поглощенные вершины кластера 3-го уровня, учтены аналогично предыдущим $Cluster_{3-22} = (44, 1), (44, 2), (44, 4), (44, 5), (44, 11), (44, 14), (44, 20), (44, 22), (44, 23), (44, 24), (44, 26), (44, 29), (44, 33), (44, 35), (2, 11), (2, 1), (1,4), (4,5), (14, 22), (11, 20), (20, 29), (33, 35)$.

На рис. 8 приведен пример поглощения перечисленных вершин кластера 4-го уровня в вершину с номером $sNode = 45$. В качестве соседних вершин здесь выступают вершины 6, 8, 15, 25, 32, 28, 31, 30, 40, 34 для исходной вершины 45.

Списки кластера 4-го уровня, поглощенные в вершину $sNode = 46$: $Cluster_{4-13} = (45, 6), (45, 8), (45, 15), (45, 25), (45, 28), (45, 30), (45, 31), (45, 32), (45, 34), (45, 40), (8, 6), (8, 15), (40, 30)$. Итоговым графом является вершина $sNode = 47$, которая включает в себя исходную вершину 46 и соседние с ней 36, 41, 37, 38, 42, 39 (рис. 9). Списки для вершин кластера 5-го уровня: $Cluster_{5-8} = (46, 36), (46, 37), (46, 39), (46, 38), (46, 41), (46, 42), (42, 41), (42, 39)$.

Заключение. Предложенный алгоритм обобщений или свертки графа ситуаций можно использовать самостоятельно при планировании и управлении как одиночными объектами, так и группой объектов. Причём каждый объект управляемой группы может получать доступ к какой-то из частей графа ситуаций, используя алгоритм обобщений. Этот алгоритм может также использоваться в комбинации с другими действиями [18] с вершинами графа ситуаций. В случае наличия группы управляемых объектов задача построения графа ситуаций может быть распределена между ними с представлением поглощенных вершин в виде списков ребер.

Представленный алгоритм свертки списков графа ситуаций показал за счет снижения избыточности вычислений экономию памяти до 30 % и оказался быстрее на 20–30 % в зависимости от плотности [19, 20] графа, в отличие от алгоритмов, использующих матрицы смежности или инцидентности и последовательный перебор вершин. Для обработки условно бесконечных графов такой алгоритм оказался единственно возможным. Снижение требований к ресурсам вычислителей позволяет использовать алгоритм на мобильных устройствах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гюльмамедов Р.В. Метод построения стратегии в системах ситуационного управления // Научный журнал «Информационно-управляющие системы». – СПб.: ГУАП, 2011. – № 6. – С. 36-39. – Режим доступа: <http://cyberleninka.ru/article/n/metod-postroeniya-strategii-v-sistemah-situatsionnogo-upravleniya>.

2. *Розенберг И.Н.* Ситуационное управление в сфере транспорта // Образовательные ресурсы и технологии. – М.: МУ им. С.Ю. Витте, 2015. – № 2 (10). – С. 42-48. – Режим доступа: <http://cyberleninka.ru/article/n/situatsionnoe-upravlenie-v-sfere-transporta>.
3. *Терехин Д.Э., Тузовский А.Ф.* Системы ситуационного управления на основе технологий Semantic Web // Всероссийская конференция с международным участием "Знания – Онтологии – Теории" (ЗОНТ-2015), Новосибирск, 6-8 октября 2015 г. – С. 151-155. – Режим доступа: <http://elib.ict.nsc.ru/jspui/handle/ICT/1394>.
4. *Рассоха В.И., Бондаренко Е.В.* Ситуационное управление городским пассажирским транспортом // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). – 2010. – № 2. – С. 86-91.
5. *Колесников А.А., Саитгареева Р.Ш.* Построение системы ситуационного управления нестационарными производственными процессами на основе структурной перестройки координационной схемы производственных процессов // Вестник УГАТУ. – 2012. – Т. 16, № 3 (48). – С. 38-41. – Режим доступа: <http://journal.ugatu.ac.ru/index.php/vestnik/article/view/107>.
6. *Мелихов А.Н., Берштейн Л.С., Коровин С.Я.* Ситуационные советующие системы с нечеткой логикой. – М.: Наука, 1990. – 272 с.
7. *Поспелов Д.А.* Ситуационное управление: теория и практика – М.: Наука, 1986. – 288 с.
8. *Оре О.* Теория графов. – М.: Наука, 1968. – 352 с.
9. *Зыков А.А.* Основы теории графов. – М.: Наука, 1986. – 382 с.
10. *Кристофидес Н.* Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с.
11. *Бородянский Ю.М., Сафонов И.В., Цикунов И.К.* Автоматизированная система анализа ситуаций. Математическое обеспечение для моделирования сложных систем / под ред. В.М. Глушкова. – Киев: АН УССР, 1973.
12. *Баринин С.В., Курейчик В.М., Гладков Л.А.* Компоновка МЭС на основе итерационной кластеризации с учетом временных задержек // Известия ТРТУ. – 2006. – № 8 (63). – С. 120-126.
13. *Солодухо Н.М.* Манифест ситуационного движения // Фундаментальные исследования. – 2005. – № 8. – С. 87-88.
14. *Поспелов Д.А.* Моделирование рассуждений. Опыт анализа мыслительных актов. – М.: Радио и связь, 1989. – 184 с.
15. *Tang J., Leung H.-F., Luo Q., Chen D., Gong J.* Towards ontology learning from folksonomies // IJCAI International Joint Conference on Artificial Intelligence. – 2009. – P. 2089-2094.
16. *Фостер Дж.* Обработка списков. – М.: Мир, 1974. – 72 с.
17. *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы. Построение и анализ. – М.: МЦНМО, 2000. – 960 с.
18. *Сергеев Н.Е., Целых А.А.* Нечеткие теоретико-графовые подходы к моделированию и анализу социосемантических сетей знаний для задач принятия решений в научной и научно-технической экспертизе // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. – 2016. – № 09 (123). – С. 1-21. – Режим доступа: <http://ej.kubagro.ru/2016/09/pdf/27.pdf>.
19. *Wasserman S., Faust, K.* Social Network Analysis: Methods and Applications. – Cambridge University Press, 1994. – 857 p.
20. *Целых Ю.А.* Теоретико-графовые методы анализа нечетких социальных сетей // Программные продукты и системы. – 2008. – № 2. – С. 48-50.

REFERENCES

1. *Gyul'mamedov R.V.* Metod postroeniya strategii v sistemakh situatsionnogo upravleniya [A method of building strategies in systems of situational management], *Nauchnyy zhurnal «Informatsionno-upravlyayushchie sistemy»* [Scientific journal "Information and control system"]. Saint Petersburg: GUAP, 2011, No. 6, pp. 36-39. Available at: <http://cyberleninka.ru/article/n/metod-postroeniya-strategii-v-sistemah-situatsionnogo-upravleniya>.
2. *Rozenberg I.N.* Situatsionnoe upravlenie v sfere transporta [Situational management in the field of transport], *Obrazovatel'nye resursy i tekhnologii* [Educational resources and technology]. Moscow: MU im. S.Yu. Vitte, 2015, No. 2 (10), pp. 42-48. Available at: <http://cyberleninka.ru/article/n/situatsionnoe-upravlenie-v-sfere-transporta>.

3. *Terekhin D.E., Tuzovskiy A.F.* Sistemy situatsionnogo upravleniya na osnove tekhnologii Semantic Web [System of situational management based on Semantic Web technologies], *Vserossiyskaya konferentsiya s mezhdunarodnym uchastiem "Znaniya – Ontologii – Teorii" (ZONT-2015), Novosibirsk, 6-8 oktyabrya 2015 g.* [Proceedings of Russian conference with international participation "Knowledge – Ontology – Theory" (an UMBRELLA-2015), Novosibirsk, Russia, October 6-8, 2015, pp. 151-155. Available at: <http://elib.ict.nsc.ru/jspui/handle/ICT/1394>.
4. *Rassokha V.I., Bondarenko E.V.* Situatsionnoe upravlenie gorodskim passazhirskim transportom [Situational management of urban passenger transport], *Vestnik Moskovskogo avtomobil'no-dorozhnogo gosudarstvennogo tekhnicheskogo universiteta (MADI)* [Bulletin of Moscow state automobile and road technical University], 2010, No. 2, pp. 86-91.
5. *Kolesnikov A.A., Saitgareeva R.Sh.* Postroenie sistemy situatsionnogo upravleniya nestatsionarnymi proizvodstvennymi protsessami na osnove strukturnoy perestroyki koordinatsionnoy skhemy proizvodstvennykh protsessov [Construction of system of situational management by non-stationary production processes based on structural adjustment coordinating the design of business processes], *Vestnik UGATU* [Bulletin of the Ufa state aviation technical University], 2012, Vol. 16, No. 3 (48), pp. 38-41. Available at: <http://journal.ugatu.ac.ru/index.php/vestnik/article/view/107>.
6. *Melikhov A.N., Bershteyn L.S., Korovin S.Ya.* Situatsionnye sovetuyushchie sistemy s nechetkoy logikoy [Situational advising systems with fuzzy logic]. Moscow: Nauka, 1990, 272 p.
7. *Pospelov D.A.* Situatsionnoe upravlenie: teoriya i praktika [Situational management: theory and practice]. Moscow: Nauka, 1986, 288 p.
8. *Ore O.* Teoriya grafov [Graph theory]. Moscow: Nauka, 1968, 352 p.
9. *Zykov A.A.* Osnovy teorii grafov [The basics of graph theory]. Moscow: Nauka, 1986, 382 p.
10. *Kristofides N.* Teoriya grafov. Algoritmicheskii podkhod [Graph theory. Algorithmic approach]. Moscow: Mir, 1978, 432 p.
11. *Borodyanskiy Yu.M., Safonov I.V., Tsikunov I.K.* Avtomatizirovannaya sistema analiza situatsiy. Matematicheskoe obespechenie dlya modelirovaniya slozhnykh sistem [Automated system for analysis of situations. Mathematical support for modeling complex systems], ed. by V.M. Glushkova. Kiev: AN USSR, 1973.
12. *Barinov S.V., Kureychik V.M., Gladkov L.A.* Komponovka MES na osnove iteratsionnoy klasterizatsii s uchetom vremennykh zaderzhek [The layout of the MES based on iterative clustering, taking into account time delays], *Izvestiya TRTU* [Izvestiya TSURE], 2006, No. 8 (63), pp. 120-126.
13. *Solodukho N.M.* Manifest situatsionnogo dvizheniya [Manifesto of situational movement], *Fundamental'nye issledovaniya* [Fundamental research], 2005, No. 8, pp. 87-88.
14. *Pospelov D.A.* Modelirovanie rassuzhdeniy. Opyt analiza myslitel'nykh aktov [Modeling of reasoning. Experience in the analysis of mental acts]. Moscow: Radio i svyaz', 1989, 184 p.
15. *Tang J., Leung H.-F., Luo Q., Chen D., Gong J.* Towards ontology learning from folksonomies, *IJCAI International Joint Conference on Artificial Intelligence*, 2009, pp. 2089-2094.
16. *Foster DzH.* Obrabotka spiskov [Processing lists]. Moscow: Mir, 1974, 72 p.
17. *Kormen T., Leyzerson Ch., Rivest R.* Algoritmy. Postroenie i analiz [Algorithms. The construction and analysis]. Moscow: MTsNMO, 2000, 960 p.
18. *Sergeev N.E., Tselykh A.A.* Nechetkie teoretiko-grafovye podkhody k modelirovaniyu i analizu sotsiosemanticheskikh setey znaniy dlya zadach prinyatiya resheniy v nauchnoy i nauchno-tekhnicheskoy ekspertize [Fuzzy-theoretic graph approaches to modeling and analysis socioemotional networks of knowledge for decision-making problems in the scientific and technical expertise], *Politematicheskii setevoy elektronnyy nauchnyy zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta* [Polythematic network electronic scientific journal of the Kuban state agrarian University], 2016, No. 09 (123), pp. 1-21. Available at: <http://ej.kubagro.ru/2016/09/pdf/27.pdf>.
19. *Wasserman S., Faust, K.* Social Network Analysis: Methods and Applications. Cambridge University Press, 1994, 857 p.
20. *Tselykh Yu.A.* Teoretiko-grafovye metody analiza nechetkikh sotsial'nykh setey [Theoretical-graph methods of analysis of fuzzy social networks], *Programmnye produkty i sistemy* [Software products and systems], 2008, No. 2, pp. 48-50.

Статью рекомендовал к опубликованию д.т.н., профессор Г.Е. Веселов.

Сергеев Николай Евгеньевич – Южный федеральный университет; e-mail: nesergeev@sfedu.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371550; кафедра вычислительной техники; д.т.н.; профессор.

Мунтян Евгения Ростиславна – e-mail: ermuntyan@sfedu.ru; кафедра вычислительной техники; старший преподаватель.

Самойлов Алексей Николаевич – e-mail: asamoylov@sfedu.ru; кафедра вычислительной техники; к.т.н.; доцент.

Целых Алексей Александрович – e-mail: tselykh@sfedu.ru; тел.: +7863431743; кафедра информационно-аналитических систем безопасности; к.т.н.; доцент.

Sergeev Nikolai Evgenievich – Southern Federal University; e-mail: nesergeev@sfedu.ru; 44, Nekrasovsky, Taganrog, 347928, Russia; phone: +78634371550; the department of computer engineering; dr. of eng. sc.; professor.

Muntyan Evgenia Rostislavna – e-mail: ermuntyan@sfedu.ru; the department of computer engineering; senior lecturer.

Samoylov Alexey Nikolaevich – e-mail: asamoylov@sfedu.ru; the department of computer engineering; cand. of eng. sc.; associate professor.

Tselykh Alexey Alexandrovich – e-mail: tselykh@sfedu.ru; phone: +7863431743; the department of information security systems; cand. of eng. sc.; associate professor.

УДК. 681.51

DOI 10.23683/2311-3103-2017-3-121-130

А.А. Скляр, Т.Е. Похилина

СИНТЕЗ НЕЛИНЕЙНОГО ЗАКОНА УПРАВЛЕНИЯ МОБИЛЬНОЙ РОБОТОТЕХНИЧЕСКОЙ ПЛАТФОРМОЙ НА КОЛЕСАХ ИЛОНА*

Демонстрируется новый подход к нелинейному управлению мобильной робототехнической платформой на колесах Илона. В качестве объекта управления выбран мобильный робот на колесах Илона, так как среди прочих наземных транспортных средств он является наиболее маневренным и предназначенным для работы в условиях ограниченного пространства, рассчитан на выполнение задач в складских помещениях. Платформа имеет форму треугольника, так как колеса расположены под углом 120 градусов, оси которых проходят через центр транспортного средства, что позволяет данной конструкции перемещать грузы в ограниченном пространстве. Для учета нелинейных характеристик объекта управления в статье рассмотрены вопросы анализа математической модели мобильной робототехнической платформой на колесах Илона. Также, приводится обзор современных методов и подходов к управлению робототехнической платформой на колесах Илона. Выделяются проблемы управления Платформой, в частности, применение методов и подходов основанных на применении методов линеаризации системы, что делает робототехническую систему с данными законами управления ограниченной определенными локальными алгоритмами управления. Поэтому в работе приводится обоснование использования новых нелинейных подходов к управлению мобильными роботами, в частности синергетической теории управления. Основным методом, в рамках данной теории, является метод аналитического конструирования агрегированных регуляторов (АКАР), который позволяет синтезировать законы управления для сложных нелинейных систем большой размерности без применения процедур линеаризации или других упрощений, поэтому для синтеза синергетического закона управления мобильного робота применяется именно этот метод. Полученный закон пространственного управления учитывает нелинейные свойства модели мобильной робототехнической платформы на колесах Илона, поэтому

* Работа выполнена при финансовой поддержке РФФИ (грант №16-08-00875А).