

---

УДК 621.039.56

**В.Д. Самойлов, С.Д. Винничук**, доктора техн. наук  
Ин-т проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины  
(Украина, 03164, Киев, ул. Генерала Наумова, 15,  
тел. (044) 4241063, e-mail: samoylov.vd@gmail.com),

**Р.П. Абрамович**  
АО «ЛьвовОРГРЭС»  
(Украина, 79011, Львов, ул. Тютюнников, 55,  
тел. (032) 2420215, e-mail: romanabramovych@gmail.com)

## **Метод подъема токов нагрузок к узлу ввода для расчета энергетических распределительных сетей**

Рассмотрена задача построения моделей коммутирующих структур электрических подстанций для тренажеров в случаях, когда необходимо определение напряжений в узлах и токов через коммутирующие элементы электроэнергетических сетей с древовидной структурой графа. Для расчета распределения перетоков предложен метод и описан алгоритм подъема токов нагрузок к узлу ввода с оценкой времени работы итерационного алгоритма на одной итерации  $T(n) = \Theta(n)$ .

Розглянуто задачу побудови моделей комутуючих структур електричних підстанцій для тренажерів у випадках, коли необхідним є визначення напруги у вузлах і струмів через комутуючі елементи електроенергетичних мереж з деревовидною структурою графа. Для розрахунку розподілення перетоків запропоновано метод і описано алгоритм підймання струмів навантажень до вузла вводу з оцінкою часу роботи ітераційного алгоритма на одній ітерації  $T(n) = \Theta(n)$ .

*Ключевые слова:* информационные технологии, коммутационная структура, конструирование тренажеров, конструирование моделей.

Расчет токов дуг и напряжений в узлах коммутирующей структуры (КС), характерной для энергетических распределительных сетей (ЭРС) (рис. 1, а), можно выполнять более эффективно, если учитывать их древовидность, которая должна сохраняться при нормальной работе сети. Для такой КС связи любого  $j$ -го узла имеют вид, представленный на рис. 1, б, где  $I_j$  — единственная входная дуга подвода тока к узлу  $j$ ,  $I_j = \sum_i I_{ji} + P_j / U_j$ ;  $I_{ji}$  ( $i \in 1 \div n$ ) — набор дуг, отводящих токи из этого узла;  $P_j$  — нагрузка узла;  $U_j$  — напряжение в узле.

При создании мультимедийных тренажеров для персонала энергоузла и подстанций ЭРС можно рассматривать как линейные электрические

© В.Д. Самойлов, С.Д. Винничук, Р.П. Абрамович, 2015

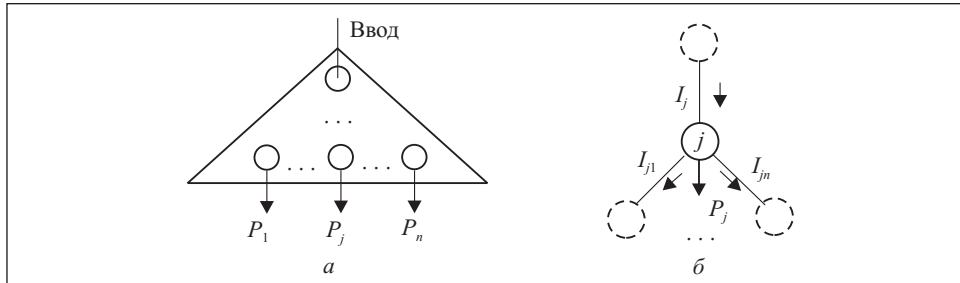


Рис. 1

цепи, если известны сопротивления (или проводимости) нагрузок и элементов цепи [1]. Тогда задача определения токов дуг и напряжений сводится к решению СЛАУ, формируемых согласно законам Кирхгофа или Максвелла. Сложность решения СЛАУ классическими прямыми методами составляет  $O(n^3)$ , где  $n$  — число узлов (для матриц Максвелла), либо число контурных токов. Для расчета потокораспределения в электрических цепях разработаны и детально описаны прямые [2], итерационные [3] методы и программные библиотеки, в которых учтены особенности структуры матриц СЛАУ. Время работы данных алгоритмов составляет  $O(n^{3/2})$  [4]. В работе [1], с учетом древовидности сети, предложен алгоритм, обладающий линейной временной сложностью. Общий алгоритм расчета токов и напряжений в ЭРС предполагает итерационное определение проводимостей нагрузок. Предлагаемый эвристический метод подъема токов не требует определения проводимостей нагрузок и характеризуется меньшей трудоемкостью на одной итерации.

**Алгоритм метода подъема токов.** Выделим в древовидной КС узел корня дерева (Исток) и узлы с нагрузками (Стоки). В узле Исток задано напряжение ввода, в узлах нагрузок заданы их мощности. С помощью алгоритма поиска в глубину (или в ширину) по заданному списку смежности для каждого из узлов можно определить узел-предок.

**А л г о р и т м.**

1. Определение напряжений в узлах нагрузок по известному начальному приближению для токов дуг.
2. Формирование токов нагрузок по известной мощности нагрузок с учетом итерационного значения напряжений в узлах нагрузок.
3. Определение токов дуг по токам нагрузок согласно первому закону Кирхгофа.
4. Определение напряжений в узлах нагрузок.
5. Расчет итерационного значения мощности в узлах нагрузок и максимальной относительной ошибки их определения.

6. Если максимальная относительная ошибка превышает допустимую величину, перейти к шагу 2, иначе — завершить расчет.

Одна итерация расчета древовидной КС выполняется в два этапа:

1. Подъем к Истоку токов, заданных узлами с нагрузками.

2. Последовательное определение напряжения во всех связанных с Истоком узлах, начиная от Истока с учетом полученных значений токов дуг и заданного напряжения истока  $E$ .

Определив на втором этапе итерации вектор  $U$  напряжений в нагруженных узлах, рассчитаем мощности в этих узлах,  $P_j^{i+1} = U_j I_j$ , на  $i+1$  итерации и определим максимальную ошибку относительно заданных мощностей:

$$\Delta P = \max_j (abs((P_j^3 - P_j^{i+1}) / P_j^3)). \quad (1)$$

На следующей итерации для определения токов нагрузок используется новый вектор напряжения  $U^{i+1}$ .

**Оценка временной сложности алгоритма на одной итерации.**

Пусть  $V$  — число вершин, а  $E$  — число дуг в графе. Для древовидных графов  $E = V - 1$ . Поэтому при оценке сложности будем использовать только число узлов сети. Итерационному процессу предшествует работа алгоритма поиска в глубину, сложность которого оценивается величиной  $O(E + V)$ , а для древовидных графов —  $O(V)$ .

На шаге 1 по известному начальному приближению для токов дуг (приняты равными нулю) определяются напряжения во всех узлах сети, включая узлы нагрузок. За время реализации алгоритма каждый узел анализируется один раз. Поэтому временная сложность шага 1 составляет  $O(V)$ .

На шаге 2 определение тока нагрузки занимает не более  $V$  делений, т.е. его временная сложность не выше  $O(V)$ .

На шагах 3 и 4 процедуры определения токов и напряжений организованы так, что каждый узел и каждая ветвь обрабатываются не более одного раза, поэтому их временная сложность составляет  $O(V)$ .

На шаге 5 выполняется расчет итерационных значений мощности и максимальной относительной ошибки для узлов нагрузок, число которых меньше числа всех узлов графа. При этом для каждого из узлов выполняются следующие операции: одно умножение, одно вычитание, одно деление и одно сравнение. Поэтому временная сложность шага 5 также составляет  $O(V)$ .

Следовательно, полная временная сложность алгоритма подъема токов оценивается величиной  $O(V)$ .

**Условия и скорость сходимости итерационного процесса.** Для электроэнергетических сетей (ЭЭС) характерно существенное преобладание изменения напряжения на нагрузке над его потерями в линиях

электрической сети и трансформаторах. Древоподобный граф ЭЭС можно дополнить ветвями, связывающими узел нагрузки с узлом «земля», а также ветвью между узлом «земля» и узлом ввода с ЭДС бесконечной мощности, обеспечивающей задаваемое значение напряжения в узле ввода. Полученный новый дополненный граф соответствует линейной электрической цепи.

Для определения токов в ветвях ее графа можно сформировать СЛАУ Кирхгофа относительно неизвестных токов в ветвях-хордах (контурные токи). В качестве хорд выберем все ветви, связывающие узлы нагрузки с узлом «земля». В каждом уравнении этой СЛАУ правая часть равняется напряжению в узле ввода, а матрица коэффициентов (сопротивления, отнесенные к ветви графа) характеризуется диагональным преобладанием, так как в ЭЭС изменение напряжения на нагрузке существенно преобладает над потерями напряжения в линиях электрической сети и трансформаторах. Степень такого преобладания определяется соотношением между потерями напряжения на нагрузке (на сформированной дополнительной ветви) и на остальной части ветвей контура от узла нагрузки до узла ввода.

Следовательно, данная СЛАУ может быть решена известными итерационными методами, например методом поконтурной увязки Андрияшева [5] или Кросса [6]. Тогда задачу расчета токов нагрузок можно свести к задаче, решаемой в работе [1], где при определении токов нагрузок на каждой итерации определяются значения сопротивления нагрузок.

Пусть в сети имеется  $k$  нагрузок и, следовательно, для дополненного графа определены  $k$  хорд. Тогда ток в каждой ветви сети можно представить в виде алгебраической суммы токов в ветвях-хордах:

$$i_n = \sum_{j=1}^k \alpha_{n,j} I_j, \quad (2)$$

где  $\alpha_{n,j}$  — коэффициент, равный либо единице, либо минус единице, либо нулю,  $n = 1 \div E$ ,  $j = 1 \div k$ .

Пусть в качестве хорд выбраны ветви дополненного графа, связывающие узлы нагрузки с узлом «земля». По системе хорд однозначно формируется система  $k$  фундаментальных циклов. Контурное уравнение согласно второму закону Кирхгофа, сформированное для произвольного контура  $s$  ( $s = 1 \div k$ ), с учетом (2) можно записать в виде

$$\sum_{n=1}^E \beta_{n,s} r_n i_n = \sum_{n=1}^E \beta_{n,s} r_n \sum_{j=1}^k \alpha_{n,j} I_j = \sum_{j=1}^k I_j \sum_{n=1}^E \beta_{n,s} \alpha_{n,j} r_n = \sum_{j=1}^k m_{j,s} I_j = U_{\text{вх}}, \quad (3)$$

где  $r_n$ ,  $n=1 \div E$ , — коэффициент сопротивления ветви  $n$ ;  $\beta_{n,s}$ ,  $n=1 \div E$ ,  $s=1 \div k$ , — признак принадлежности ветви  $n$  контуру  $s$ : если ветвь  $n$  принадлежит контуру  $s$ , то  $\beta_{n,s} = 1$ , а иначе —  $\beta_{n,s} = 0$ . Согласно методу по-контурной увязки итерационный процесс решения СЛАУ (3) можно представить такой последовательностью операций:

$$\Delta I_s^{t+1} = \left( U_{\text{вх}} - \sum_{j=1}^k m_{j,s} I_j^t \right) / m_{s,s}, \quad t \geq 0, \quad s=1 \div k, \quad (4)$$

$$I_s^{t+1} = I_s^t + \Delta I_s^{t+1}, \quad t \geq 0, \quad s=1 \div k.$$

Сходимость итерационного процесса (4) доказана в работе [7] без дополнительного условия существенного преобладания изменения напряжения на нагрузке над его потерями в остальной части сети от узла ввода до узла нагрузки. Первое уравнение (4) представим в виде

$$m_{s,s} \Delta I_s^{t+1} = \left( U_{\text{вх}} - \sum_{j=1, j \neq s}^k m_{j,s} I_j^t - (m_{s,s} - r_s) I_s^t \right) - r_s I_s^t = U_s^t - r_s I_s^t, \quad s=1 \div k, \quad (5)$$

где  $U_s^t$ ,  $s=1 \div k$ ,  $t \geq 0$ , — напряжение на  $k$ -й нагрузке на шаге  $t$ . Если учесть дополнительное условие существенного преобладания изменения напряжения на нагрузке над его потерями в остальной части сети от узла ввода до узла нагрузки, то при замене коэффициентов  $m_{s,s}$  их составной частью  $r_s$  уравнение (5) можно представить в виде

$$r_s (\Delta I_s^{t+1} + I_s^t) = r_s I_s^{t+1} = U_s^t, \quad s=1 \div k. \quad (6)$$

Если  $r_s = (U_s^t)^2 / P_s$ , то  $(U_s^t)^2 / P_s I_s^{t+1} = U_s^t$ ,  $s=1 \div k$ , или

$$I_s^{t+1} = P_s / U_s^t, \quad s=1 \div k. \quad (7)$$

Формула (7) описывает способ определения нового итерационного значения токов нагрузок согласно алгоритму метода подъема токов, в котором не предполагается поиск точного решения линейной задачи определения токов нагрузок при известных сопротивлениях (или проводимостях) нагрузок. Кроме того, новые итерационные значения токов нагрузки, найденные согласно (7), близки к аналогичным значениям, определенным на одной итерации согласно (4), (5), поскольку близки по величине приращения токов нагрузок  $\Delta I_s^{t+1}$ , определяемые из (5), (6):

$$\Delta I_{s,5}^{t+1} = (U_s^t - r_s I_s^t) / m_{s,s}, \quad s=1 \div k, \quad (8)$$

$$\Delta I_{s,6}^{t+1} = (U_s^t - r_s I_s^t) / r_s, \quad s=1 \div k,$$

где  $m_{s,s} > r_s$ . Однако значения  $m_{s,s}$  и  $r_s$  близки, так как для ЭЭС характерно существенное преобладание изменения напряжения на нагрузке над его потерями в линиях электрической сети и трансформаторах.

Практические расчеты токов нагрузок для ряда примеров ЭЭС показали, что при  $m_s > r_s$  для приращения  $\Delta I_{s,6}^{t+1}$  итерационный процесс ускоряется. В случае, когда в качестве начального приближения токов нагрузок были выбраны их нулевые значения, значения токов нагрузок монотонно увеличивались до достижения требуемой точности. Реализованная простая итерация для большинства практических задач обеспечивает сходимость за два-три шага при  $\Delta P < 10^{-3}$ .

**Алгоритмы поиска токов и напряжений в древовидных КС.** Определение токов в дугах неориентированного графа КС реализуется на основе алгоритма поиска в глубину узлов, достижимых из узла Исток. Граф задан в виде списка смежности (*СписСмеж*). Спецификация алгоритмов выполнена в виде псевдокода [1, 8] с использованием метода модельного программирования для формульного логико-алгебраического представления кода [1]. Формульное представление оператора «Если *<условие>* То *<оператор или список операторов 1>* Иначе *<оператор или список операторов 2>*» может быть реализовано в виде «*<условие>* ? *<оператор или список операторов 1>* : *<оператор или список операторов 2>*» или в правой части формулы «*переменная = <условие>* ? *<выражение 1>* : *<выражение 2>*». Комментарий к строке псевдокода начинается после знака //.

**Метод поиска в глубину (ПоискГлуб(*u*))** узлов, достижимых из Истока, подробно описан в работе [8]. Узлы обрабатываются, начиная с заданного начального узла (Исток). Граф КС сохраняется в списке смежности (*СписСмеж*), где каждому узлу графа соответствует набор узлов, с которыми данный узел связан ветвями:

**ПоискГлуб(*u*, *СписСмеж*)**

1. *Цвет[u]* ← СЕРЫЙ
2. ЦиклДляВсех ( $v \in \text{СписСмеж}[u]$ ) // просмотр смежных с *u* узлов
3. (*Дуга[Индекс] = ЕСТЬ*) И (*Цвет[v]=БЕЛЫЙ*) И ПоискГлуб(*v*)
4. *Цвет[u]* ← ЧЕРНЫЙ // выход из текущей рекурсии

Узлы графа КС заданы неповторяющимися числами (номера узлов). В общем случае КС содержит включенные и отключенные ветви. При поиске просматриваются только включенные ветви.

Для стандартного поиска в глубину необходим массив цветов узлов графа (*Цвет*). Перед началом поиска массив *Цвет* пуст. Элементы массива *Цвет* обеспечивают раскраску узлов графа в БЕЛЫЙ, СЕРЫЙ и ЧЕРНЫЙ цвета. Каждому достигнутому в процессе поиска узлу *u* соответствует

процедура обработки этого узла. В начале этой процедуры  $Цвет[u] \leftarrow$  СЕРЫЙ (строка 1).

Процедура обеспечивает просмотр всех узлов  $v_i$ , связанных с узлом  $u$  дугами ветвей, представленных в *СписСмеж* парой  $(u, v)$  (строка 2). Если цвет  $v$  БЕЛЫЙ, то выполняется переход на этот узел вследствие рекурсивного запуска процедуры **ПоискГлуб**( $v$ ) (строка 4). Если просмотр узла  $u$  закончен и нет переходов на белые узлы, то  $Цвет[u] \leftarrow$  ЧЕРНЫЙ и завершение процедуры обработки узла  $u$ , а именно происходит возврат к процедуре, из которой рекурсивно была вызвана данная процедура, а именно обработки предка данного узла.

Таким образом, любой переход из данного узла  $u$  на белый цвет  $v$  приводит к возврату из  $v$  в  $u$ . В процессе поиска в глубину происходит построение дерева поиска, так как на основании древовидности КС каждому доступному из Истока узлу соответствует единственный предок.

А л г о р и т м определения токов в дугах и напряжений в узлах КС содержит две последовательно выполняемые процедуры поиска в глубину — токов, **ПоискГлубТоки(Исток)**, и напряжений, **ПоискГлубНапр(Исток)**:

**ТокиНапр** (*СписСмеж*)

1. *Итерация*  $\leftarrow$  0
2. *НапрУз*  $\leftarrow$  [ ]
3. ЦиклПока (*ОшибкаМоц* > *ДопОшибкаМоц*)
4. *ТокДуг*  $\leftarrow$  [ ]
5. *Цвет*  $\leftarrow$  [ ]
6. *Предок*  $\leftarrow$  [ ]
7. **ПоискГлубТоки(Исток)**
8. *Цвет*  $\leftarrow$  [ ]
9. *Предок*  $\leftarrow$  [ ]
10. **ПоискГлубНапр(Исток)**
11. *Итерация*  $\leftarrow$  *Итерация* + 1

А л г о р и т м подъема токов к Истоку (этап 1). Псевдокод алгоритма поиска в глубину токов в дугах дерева имеет следующий вид:

**ПоискГлубТоки**( $u$ , *СписСмеж*)

1.  $Цвет[u] \leftarrow$  СЕРЫЙ
2. *ИндексПр*  $\leftarrow$  (*Предок* [ $u$ ],  $u$ ) // индекс дуги от *Предка*[ $u$ ] к  $u$
3. (*НагрУз*[ $u$ ] = ЕСТЬ) И **НагрУзлов** ( $u$ , *ИндексПр*) // токи нагруженных узлов
4. ЦиклДляВсех ( $v \in$  *СписСмеж* [ $u$ ])
5. *Индекс*  $\leftarrow$  ( $u, v$ )

6. (Дуга[Индекс] = ЕСТЬ) И (Цвет[v] = БЕЛЫЙ) И (Предок [v] ← u, ПоискГлубТоки(v))
  7. (ОтмУзТок[v] = ЕСТЬ) И (Предок [v] = u) И ПодъемТока(u, v, Индекс, ИндексПр)
  8. Цвет[u] ← ЧЕРНЫЙ
- НагрУзлов(u, ИндексПр)**
1. ОтмУзТок[u] ЕСТЬ // отметка узла u как токового
  2. (Итерация = 0) И (ТокДуги[ИндексПр] ← 0) // итерация с нулевыми токами
  3. (Итерация > 0) И (ТокДуги[ИндексПр] ← МощУз[u]/НапрУз[u]) // токи нагруженных узлов по НапрУз[u] из (Итерация = 0)
- ПодъемТока(u, v, Индекс, ИндексПредка)**
1. ОтмУзТок[u] ← ЕСТЬ //отметка узла u как токового
  2. ТокДуги[ИндексПр] ← ТокДуги[ИндексПр] + ТокДуги[Индекс] // подъем тока из дуги (u, v) в дугу-предок

Любое переключение в КС приводит к древовидной структуре графа, узлам которого соответствуют значения напряжений (массив *НапрУз*), а ветвям (дугам) — значения токов (массив *ТокДуги*). Токи в ветвях между узлами образуются вследствие нагрузок в некоторых узлах. Заданная мощность нагрузки  $P_i$  в узле  $u_i$  обеспечивает ток  $I_i = P_i/U_i$  в случае, если узел  $u_i$  доступен из узла Исток. В эквивалентной графу электрической цепи узел цепи  $u_i$  связан проводимостью  $g_i = I_i/U_i$  с общим узлом цепи («землей»). Если между узлом Исток и «землей» приложено напряжение  $E$ , а между узлом с нагрузкой  $v_i$  и узлом Исток есть связь, то от  $v_i$  к «земле» будет протекать ток.

Узел  $u_i$  с нагрузкой, достигнутый в процессе поиска, отметим как токовый в массиве *ОтмУзТок*. Такая отметка означает, что из этого узла протекают токи к другим узлам, т.е. он является предком токовых узлов, и к нему протекает входной ток от единственного предка. Ток от предка к  $u_i$  для древовидных структур равен сумме токов, направленных от узла, а узел предка  $u_i$  должен быть отмечен как токовый. Таким образом, все узлы цепочки по предкам от нагруженного узла к Истоку будут отмечены как токовые.

Используем метод поиска токов, описанный в [9], который расширяет стандартную процедуру поиска в глубину посредством применения двух расчетных схем:

- 1) определение токов от нагруженных узлов — **НагрУзлов(u, Индекс Предка)**;
- 2) подъем суммы токов, передаваемых из узла  $u_i$  в дугу-предок — **Подъем Тока(u, v, Индекс, ИндексПредка)**.



Схема 1 в процедуре **ПоискГлубТоки** выполняется при достижении узла с нагрузкой (строка 3). На начальной итерации (*Итерация* = 0) выполняется расчет КС при нулевых токах нагрузок (строка 2). На последующих итерациях используются токи нагрузок, определяемые в каждом узле как отношение мощности нагрузки к напряжению, найденное на предыдущей итерации. Ток нагрузки сохраняется в массиве *ТокУзНагр* (строка 4).

Схема 2 выполняется, если для дуги ( $u, v$ )  $v$  является токовым узлом, но не предком  $u$  (строка 8). При этом узел отмечается как токовый и формируется ток в дуге-предке. Токи, передаваемые от дуг ( $u, v$ ) в дугу-предок, корректируются с учетом коэффициента трансформации, если узел  $u$  является входным узлом трансформатора (строка 2). В строке 3 ток дуги ( $u, v$ ) добавляется к току дуги-предка.

А л г о р и т м определения напряжений в узлах КС (этап 2). Псевдокод алгоритма определения напряжений основан на процедуре поиска в глубину и имеет следующий вид:

**ПоискГлубНапр**( $u$ , *СписСмеж*)

1. *Цвет*[ $u$ ] ← СЕРЫЙ
2. ЦиклДляВсех(  $v \in$  *СписСмеж* [ $u$ ])
3.     *Индекс* ← ( $u, v$ )
4.     (*Дуга*[*Индекс*] = ЕСТЬ) И (*Цвет*[ $v$ ] = БЕЛЫЙ) И
5.         (**НапрУзла**( $u, v, \text{Индекс}$ ), **ПоискГлубНапр**( $v$ ))
6.         //расчет напряжения в узле  $v$  и рекурсивный переход в  $v$
7.     (*НагрУз*[ $v$ ] = ЕСТЬ) И **ОшибкаМощн**[ $v$ ] // если ЕСТЬ нагрузка в узле  $v$ , // то расчет ошибки полученной мощности нагрузки относительно заданной
8.     *Цвет*[ $u$ ] ← ЧЕРНЫЙ

**НапрУзла**( $u, v, \text{Индекс}$ )

1. *НапрУз*[ $v$ ] ← *НапрУз*[ $u$ ] – *ТокДуг*[*Индекс*]/*Пров* [*Индекс*]

Здесь стандартная процедура поиска в глубину дополнена расчетом напряжений в узлах  $v$  дуг ( $u, v$ ) (процедура **НапрУзла**( $u, v, \text{Индекс}$ )) по известному напряжению в  $u$  и проводимости дуги (*Пров* [*Индекс*]). Если *ТокДуг*[*Индекс*] = 0, то *НапрУз*[ $v$ ] = *НапрУз*[ $u$ ]. Поиск в глубину обеспечивает требуемую последовательность расчета, начиная с Истока, в котором задано напряжение  $E$ .

Для расчета критерия (1) остановки итерационного процесса используем процедуру **ОшибкаМощн**[ $v$ ]. В ней для каждого нагруженного узла определяется ошибка полученной на итерации мощности относительно заданной величины. Затем определяется и запоминается в переменной *ОшибкаМощ* максимальная из относительных ошибок для всех нагру-

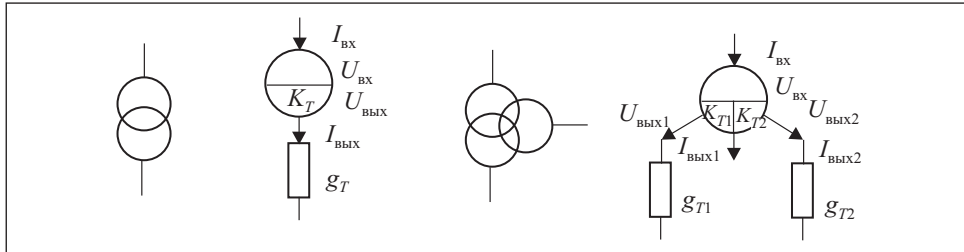


Рис. 2

женных узлов. Выполняется итерационное уточнение мощностей узлов, пока величина этой ошибки остается больше заданной.

**Учет наличия трансформаторов.** В КС ЭРС содержатся двух- и трехобмоточные силовые трансформаторы (рис. 2). В графе КС им соответствуют узлы специального вида и ветви с заданными проводимостями ( $g_T$ ,  $g_{T1}$ ,  $g_{T2}$ ), которые определяются по потерям на трансформаторе при максимальной рабочей нагрузке [9]. При наличии трансформаторов в КС они в качестве компонентов появляются в ветвях *СписСмеж*. Таким образом, ветви *СписСмеж* КС могут содержать два типа компонентов: 1) выключатели и разъединители; 2) трансформаторы.

Первый тип ветвей назовем симметричным, так как при любом порядке их просмотра проводимость одинакова:  $g_{uv} = g_{vu} = g$ .

Второй тип ветвей назовем несимметричным. В трансформаторной ветви параметры дуг для вычисления токов и напряжений в узлах не одинаковы: для ребра  $(u, v)$  коэффициент трансформации  $K_T$ , а для ребра  $(v, u)$  —  $1/K_T$ .

Для несимметричных ветвей существенное значение имеет порядок просмотра при подъеме токов и вычислении напряжения поиском в глубину достижимых узлов. Логика обработки узла  $u$ , в котором находятся входы трансформаторов, должна учитывать несимметричность передачи ветвей по току и напряжению. Такой узел назовем трансформаторным узлом (ТУ). Трансформаторные узлы не нарушают древовидности КС, в которую они включены. В каждом ТУ заданы коэффициенты трансформации  $K_T$ , с помощью которых выполняется преобразование входных напряжений в выходные и по выходным токам определяются входные.

Для двухобмоточного трансформатора запишем:  $I_{BX} = I_{ВЫХ}/K_T$ ,  $U_{ВЫХ} = U_{BX}/K_T$ ; для трехобмоточного —  $I_{BX} = I_{ВЫХ1}/K_{T1} + I_{ВЫХ2}/K_{T2}$ ,  $U_{ВЫХ1} = U_{BX}/K_{T1}$ ,  $U_{ВЫХ2} = U_{BX}/K_{T2}$ .

Рассмотрим логику обработки ТУ при подъеме токов к Истоку и вычисление напряжений по известным токам дуг. Одним из параметров

трансформатора является *ИндексТр* — индекс основной дуги ( $u, v$ ) размещения трансформатора, где  $u$  и  $v$  — узлы входа и выхода трансформатора.

Токи нагруженных узлов. Процедура обработки нагруженного узла  $u$  с учетом наличия ТУ в дуге-предке представлена в следующем псевдокоде **НагрУзлов**.

**НагрУзлов**( $u$ , *ИндексПр*)

1. *ОтмУзТок*[ $u$ ] ← ЕСТЬ //отметка узла  $u$  как токового
2. (*Итерация* = 0) И (*ТокДуги*[*ИндексПр*] ← 0) //итерация с нулевыми токами
3. *ЕстьТрУПри* ← ( $u \neq \text{Исток}$ ) && (*ИндТр*[*ИндексПр*] = ЕСТЬ) && (*ИндексПр* ≠ *ИндексТр* [*ИндексПр*]) //есть ТУ трансформатора дуги-предка (*Предок*[ $u$ ],  $u$ ) в  $u$
4. *КоэфТока* ← *ЕстьТрУПри* ? *КоэфТр*[*ИндексПр*] : 1;
5. (*Итерация* > 0) && (*ТокДуг*[*ИндексПр*] ← *МощУз*[ $u$ ]/*НапрУз*[ $u$ ]\*  
*КоэфТока*) //токи нагруженных узлов по *НапрУз*[ $u$ ] из (*Итерация* = 0)

Строки 1 и 2 аналогичны псевдокоду процедуры **ПодъемТока**. Наличие трансформатора в дуге-предке (строка 3) обуславливает *КоэфТока* = *КоэфТр*, и в строке 5 с помощью этого коэффициента корректируется ток, создаваемый нагрузкой узла (*МощУз*[ $u$ ]).

Подъем токов дуг.

Псевдокод **ПодъемТока**( $u, v$ , *Индекс*, *ИндексПредка*)

1. *ОтмУзТок*[ $u$ ] ← ЕСТЬ //отметка узла  $u$  как токового
2. *ЕстьТрУив* ← (*ИндексТр* [*Индекс*] = ЕСТЬ) && (*Индекс* = *ИндексТр* [*Индекс*]) //есть ТУ трансформатора дуги ( $u, v$ ) в  $u$
3. *ЕстьТрУПри* ← ( $u \neq \text{Исток}$ )&&(*ИндексТр* [*ИндексПр*] = ЕСТЬ) && (*ИндексПр* ≠ *ИндексТр* [*ИндексПр*]) // есть ТУ трансформатора дуги-предка (*Пр*[ $u$ ],  $u$ ) в  $u$
4. *КоэфТока* = (*ЕстьТрУПри* ? *КоэфТр*[*ИндексПр*] : 1)/(*ЕстьТрУив* ? *КоэфТр*[*Индекс*] : 1)
5. *ТокДуги*[*ИндексПр*] ← *ТокДуги*[*ИндексПр*] + *ТокДуги*[*Индекс*] \* *КоэфТока* //подъем тока из дуги ( $u, v$ ) в дугу-предок

Для узла  $u$  дуги ( $u, v$ ) (которая задана переменной *Индекс*) возможны следующие варианты размещения ТУ в  $u$ , влияющие на коэффициент передачи тока (*КоэфТока*) дуги ( $u, v$ ) в дугу-предок (*Предок*[ $u$ ],  $u$ ) (которая задана переменной *Индекс Пр*):

если в узле  $u$  нет ТУ, то *КоэфТока* = 1;

если в узле  $u$  находится ТУ трансформатора дуги ( $u, v$ ) (условие *ЕстьТрУив* в строке 2), то *КоэфТока* = 1/ *КоэфТр* [*Индекс*];

если в узле  $u$  находится ТУ трансформатора дуги ( $Предок[u], u$ ) (условие  $ЕстьТрУПри$  в строке 3), то  $КоэфТока = КоэфТр [ИндексПр]$ ;

если в узле  $u$  находятся два ТУ — дуги  $(u, v)$  и дуги ( $Предок[u], u$ ) (условия строк 2 и 3), то  $КоэфТока = КоэфТр [ИндексПр] / КоэфТр [Индекс]$ .

Выполнение всех вариантов обеспечивает формула строки 5.

**О п р е д е л е н и е н а п р я ж е н и й в у з л а х.** Напряжение в узле можно определить, если оно известно в одном из смежных узлов. Если узел с известным напряжением является ТУ, то необходимо учитывать изменение входного напряжения с использованием коэффициента трансформации.

**НапрУзла**( $u, v, Индекс$ )

1.  $ЕстьТр \leftarrow (ИндексТр [Индекс] = ЕСТЬ) //$  есть трансформатор в дуге  $(u, v)$
2.  $ЕстьТрУи \leftarrow ЕСТЬТр \ \&\& \ (Индекс = ИндексТр[Индекс]) //$  есть ТУ трансформатора дуги  $(u, v)$  в  $u$
3.  $ЕстьТрУv \leftarrow ЕСТЬТр \ \&\& \ (Индекс \neq ИндексТр[Индекс]) //$  есть ТУ трансформатора дуги  $(u, v)$  в  $v$
4.  $(ТокДуг[Индекс] = ПУСТО) \ \&\& \ (ТокДуг[Индекс] \leftarrow 0)$
5.  $НапрУз[v] \leftarrow НапрУз[u] / (ЕстьТрУи ? КоэфТр[Индекс] : 1) - ТокДуг[Индекс] / Пров [Индекс] * (ЕстьТрУv ? КоэфТр[Индекс] : 1)$

Место размещения ТУ определяется с помощью параметра  $ИндексТ$ . В узле  $u$  ТУ размещен в случае истинности выражения  $(Индекс = ИндТр [Индекс])$ , а в узле  $v$  — в случае  $(Индекс \neq ИндТр [Индекс])$ . Для определения напряжения в узле  $v$  при известном напряжении в узле  $u$  с учетом возможного наличия в дуге  $(u, v)$  трансформатора требуется проанализировать следующие три варианта:

- трансформатор в дуге  $(u, v)$  отсутствует;
- есть трансформатор в дуге  $(u, v)$  и его ТУ находится в  $u$ ;
- есть трансформатор в дуге  $(u, v)$  и его ТУ находится в  $v$ .

Скорректированный псевдокод **НапрУзла** следующий. В строке 1 истинность переменной  $ЕстьТр$  свидетельствует о наличии трансформатора в дуге  $(u, v)$ . В строке 2 формируется условие  $(ЕстьТрУи)$ , т.е. размещение ТУ в узле  $u$ , а в строке 3 — в узле  $v$  ( $ЕстьТрУv$ ).

Формула в строке 5 обеспечивает определение напряжения в узле  $v$  для любого из трех вариантов вида дуги  $(u, v)$ .

**Модель КС с несколькими вводами.** Каждый ввод КС соответствует отдельному дереву подвода напряжений к нагрузкам. Несколько вводов создают лес деревьев. Моделирование леса деревьев может быть выполнено последовательным расчетом каждого дерева с сохранением параметров узлов просмотренных деревьев.

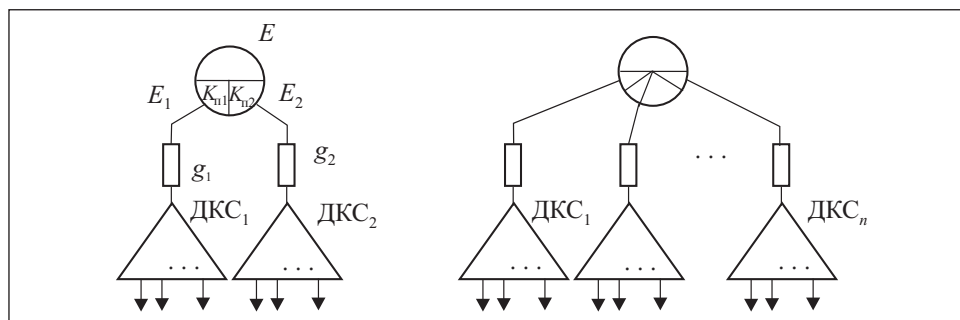


Рис. 3

Представляется более эффективным метод моделирования КС с несколькими вводами посредством создания единого дерева из леса деревьев. Это можно выполнить с помощью добавления в КС специального узла вводов (УВ). Он является истоком единого дерева, в котором расчет токов и напряжений может быть выполнен с помощью схемы УВ, представленной на рис. 3

Структура УВ аналогична модели многообмоточного трансформатора, число обмоток (выходов) которого равно числу деревьев в лесу КС. В схеме замещения УВ для каждого выхода определяются два параметра:  $K_{pi}$  — коэффициент передачи для формирования заданного входного напряжения  $E_i$   $i$ -го ввода и  $g_i$  — проводимость, связывающая узел ввода с истоком  $i$ -го дерева КС (ДКС $_i$ ). Проводимость  $g_i$  не должна влиять на распределение токов и напряжений в ДКС $_i$ . В идеальном случае  $g_i = \infty$ . При практической реализации модели значение  $g_i$  выбирается значительно большим, чем максимальное значение проводимости ветвей КС. Поскольку эквивалентная схема УВ аналогична схеме замещения многообмоточного трансформатора, нет необходимости в специальном алгоритме для обработки УВ. Программа воспринимает УВ как трансформатор с коэффициентом  $K_{pi} = E/E_i$  и  $g_i \leftarrow \infty$ . Проводимость  $g_i$  можно использовать для задания проводимости линий к вводам.

На практике для реализации тренажеров оперативных переключений используется гибридный подход с проверкой на древовидность структуры КС. Если она обеспечена, расчет выполняют с помощью предложенного «быстрого» алгоритма, если нет, то используют другие методы, например решение СЛАУ.

## Выводы

При создании мультимедийных тренажеров для персонала энергоузла и подстанций необходимы модели КС, формирующие напряжения в узлах и токи через коммутационный элемент КС.

Можно значительно сократить время расчета модели КС, если учитывать древовидность структуры распределительных сетей. При решении нелинейной задачи расчета токов нагрузок для большинства практических задач требуется две-три итерации.

## СПИСОК ЛИТЕРАТУРЫ

1. Винничук С.Д., Самойлов В.Д. Определение токов в коммутационных структурах электроэнергетических сетей с древовидной структурой графа// Электрон. моделирование. — 2015. — 37, № 5. — С. 89—104.
2. Davis T.A. Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms). — Society for Industrial and Applied Mathematics, 2006. — 218 p.
3. Saad Y. Iterative Methods for Sparse Linear Systems. Second Edition.— Society for Industrial and Applied Mathematics, 2003. — 527 p.
4. Bangerth W., Rannacher R. Adaptive Finite Element Methods for Differential Equations.— Springer Science & Business Media, 2003. — 207 p.
5. Андрияшев М.М. Гидравлические расчеты водоводов и водопроводных сетей. — М. : Стройиздат, 1964. — 107 с.
6. Cross H. Analysis of flow in networks of conduits or conductors. — Urbana, Illinois, 1936, November. — Bull. No 286. — 29 p.
7. Пшеничный Б.Н. Расчет энергетических сетей на ЭВМ. — Журн. вычислительной математики и математической физики. — 1962. — № 5. — С. 942—947.
8. Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л. и др. Алгоритмы: построение и анализ. 2-е изд. Пер. с англ. — М. : Изд. дом «Вильямс», 2007. — 1296 с.
9. Самойлов В.Д., Абрамович Р.П. Поиск токов в коммутационных структурах решением СЛАУ// Электрон. моделирование. — 2013. — 35, № 1. — С. 95 — 107.

*V.D. Samoylov, S.D. Vynnychuk, R.P. Abramovych*

## THE METHOD OF LIFTING THE LOAD CURRENTS TO INPUT NODE TO CALCULATE THE ENERGY DISTRIBUTION NETWORKS

The problems of constructing models of commutating structures of electrical substations for training simulators are considered in the paper. The model should determine the node voltage and currents through the switching elements of such structures. Algorithmic iterative method of calculation the tree-like interconnect structure in two passes using a depth-first search algorithm is proposed. This algorithm requires less time to calculate and gives the same results as algorithm that uses LSE to calculate.

*Keywords : informational technologies, switching structure, designing simulators, designing models.*

REFERENCES

1. Vinnichuk, S.D. and Samoilo, V.D. (2015), "Determination of currents in commutative structure of the electricity networks with a tree structure graph", *Elektronnoe modelirovanie*, Vol. 37, no. 5, pp.89-104.
2. Davis, T.A. (2006), *Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms)*, Society for Industrial and Applied Mathematics Review.
3. Saad, Y. (2003), *Iterative Methods for Sparse Linear Systems*, 2nd edition, Society for Industrial and Applied Mathematics Review.
4. Bangerth, W. and Rannacher, R. (2003), *Adaptive Finite Element Methods for Differential Equations*, Springer Science & Business Media, New York, USA.
5. Andriyashov, M.M. (1964), *Gidravlicheskie raschoty vodovodov i vodoprovodnykh sistem* [Hydraulic calculations of water pipelines and water supply systems], Stroyizdat, Moscow, Russia.
6. Cross, H. (1936), *Analysis of flow in networks of conduits or conductors*, Bull. no. 286, University of Illinois, Urbana, USA.
7. Pshenichny, B.N. (1962), "Calculation of power networks by computers", *Zhurnal vychislitel'noy matematiki i matematicheskoy fiziki*, no. 5, pp. 942-947.
8. Cormen, T.H., Leiserson, Ch.I., Rivest, R.L. and Stein, C. (2009), *Algoritmy: postroyeniye i analiz* [Introduction to Algorithms, 3rd ed.], Translated from English, Izdatelskiy dom "Vilyams", Moscow, Russia.
9. Samoilo, V.D. and Abramovych, R.P. (2013), "Search of currents in commutative structures of electrical substations for training simulators of operational switches", *Elektronnoe modelirovanie*, Vol. 35, no. 1, pp. 95-108.

Поступила 19.10.15

*САМОЙЛОВ Виктор Дмитриевич, д-р техн. наук, профессор, гл. науч. сотр. Ин-та проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины. В 1960 г. окончил Украинскую академию сельскохозяйственных наук. Область научных исследований — компьютерные технологии моделирования, тренажеры, профессиональная диагностика в энергетике.*

*ВИННИЧУК Степан Дмитриевич, д-р техн. наук, и.о. зав. отделом Ин-та проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины. В 1977 г. окончил Черновицкий госуниверситет. Область научных исследований — модели, методы и программные средства для анализа систем сжимаемой и несжимаемой жидкостей, режимы электроэнергетических систем, частотная противоаварийная автоматика, теория алгоритмов.*

*АБРАМОВИЧ Роман Петрович, руководитель отделения АО «ЛьвовОРГРЭС». В 2007 г. окончил Национальный университет «Львовская политехника». Область научных исследований — технологии моделирования при построении тренажеров, подготовка персонала в энергетике.*