

УДК 621.313:681.3

В. И. МИЛЫХ**АВТОМАТИЗИРОВАННОЕ ФОРМИРОВАНИЕ РАСЧЕТНЫХ МОДЕЛЕЙ ТРЕХФАЗНЫХ АСИНХРОННЫХ ДВИГАТЕЛЕЙ ДЛЯ ПРОГРАММНОЙ СРЕДЫ FEMM**

Надано принципи автоматизованого формування геометричної і фізичної моделей трифазних асинхронних двигунів на алгоритмічній мові Lua для розрахунку їх магнітних полів і параметрів в програмному середовищі пакету FEMM. Представлені теоретичні основи, алгоритм і текст програми з докладними коментарями, файл вихідних даних, а також коротка інструкція щодо роботи зі скриптом Lua та приклади отримуваних результатів. Можливості складеного скрипту Lua демонструються на прикладі реального чотириполюсного асинхронного двигуна потужністю 15 кВт.

Ключові слова: метод скінчених елементів, магнітне поле, програма FEMM, трифазний асинхронний двигун, автоматизоване формування, графічна та фізична моделі, Lua скрипт.

Показаны принципы автоматизированного формирования геометрической и физической моделей трехфазных асинхронных двигателей на алгоритмическом языке Lua для расчета их магнитных полей и параметров в программной среде пакета FEMM. Представлены теоретические основы, алгоритм и текст программы с подробными комментариями, файл исходных данных, а также краткая инструкция для работы со скриптом Lua и примеры получаемых результатов. Возможности составленного скрипта Lua демонстрируются на примере реального четырехполюсного асинхронного двигателя мощностью 15 кВт.

Ключевые слова: метод конечных элементов, магнитное поле, программа FEMM, трехфазный асинхронный двигатель, автоматизированное формирование, графическая и физическая модели, Lua скрипт.

Attention to the popular FEMM (Finite Element Method Magnetics) program, effective in the numerical calculations of the magnetic fields in electrical machines is attended. The main problem of its using – high time costs on the formation of a graphical model, representing the construction and on the formation of a physical model, representing materials properties and machines winding currents, is solved. Principles of an automated formation of such models on the three-phase induction motor example are developed and presented for this purpose. A program written in an algorithmic language Lua which integrated into the package FEMM realize this task. The program is universal from the standpoint of varying the geometry and dimensions of the designed three-phase induction motor. It uses a minimum of input information in a digital form, representing the design of the whole three-phase induction motor and its fragments. A general structure of the Lua script is given, significant parts of its text, the graphic results of work's phases, as well as explanations of the program and instructions for its use are provided. Performance capabilities of the compiled Lua script on the example of the real 15 kW three-phase induction motor are presented.

Keywords: Finite Element Method Magnetics, program FEMM, three-phase induction motor, automated formation, graphical and physical models, Lua script.

Введение. Существуют программные продукты для численных расчетов магнитных полей (МП) различных электротехнических устройств. Программы позволяют учитывать достаточно точно реальную геометрию устройств и нелинейные магнитные свойства ферромагнитных сердечников. Среди таких продуктов широкое распространение получила программа FEMM [1], основанная на методе конечных элементов и ориентированная на расчет двухмерных МП. Этого достаточно для электрических машин (ЭМ) классической конструкции, имеющих практически плоскопараллельное МП в пределах аксиальной длины их активной части.

Программа FEMM для расчета МП уже не требует от пользователя разработки каких-либо математических моделей – формул и систем уравнений. Достаточно следовать прилагаемым инструкциям при построении *геометрических моделей*, отображающих конструкцию ЭМ, и *физических моделей*, отображающих физические свойства материалов и токовые параметры обмоток и постоянных магнитов. Надо следовать инструкциям и при извлечении результатов расчета МП и их использовании.

При эксплуатации программы FEMM обусловленные действия выполнялись и еще выполняются вручную. Наибольшее время расчетчики тратят на формирование геометрических и физических моделей и извлечение получаемых результатов расчетов, а также на преобразование их в форму, пригодную для

дальнейшего использования в процессе проектирования и исследования ЭМ.

Проблемы затрат времени расчетчиков становятся еще более серьезными при многовариантных расчетах МП и электромагнитных параметров ЭМ, в случаях варьирования их геометрий, размерами, токами, положением подвижных частей.

Обусловленные проблемы можно преодолеть автоматизацией построения геометрических и физических моделей ЭМ, управления расчетом и перебора параметров, а также извлечения, обработки и приведения в необходимую форму результатов. Такие возможности пользователям программы FEMM предоставляет интегрированный в нее алгоритмический язык Lua. И задача теперь заключается в написании соответствующего комплекса программ (*скриптов*). Эта задача является весьма сложной, так как требует владения знаниями в областях ЭМ, программирования и теории МП. Но эта задача вполне оправдана, так как программа пишется один раз, а используется множество раз и любым количеством пользователей.

Для обеспечения высокой эффективности расчетов МП и электромагнитных параметров в среде FEMM автором разработан комплекс скриптов Lua для турбогенераторов [2-4]. Эти программы позволяют многократно повысить производительность расчетчиков и при этом еще обеспечить возможность получения таких параметров и характеристик ТАД, которые практически невозможно получить в

«ручном» режиме работы с программой FEMM. На основе этих скриптов можно уже более легко разрабатывать аналогичные программные продукты для других ЭМ.

Целью данной работы является представление принципов составления скрипта Lua для автоматизированного формирования геометрической и физической моделей трехфазных асинхронных двигателей (ТАД) – наиболее распространенных и ответственных представителей семейства ЭМ [5].

Требованиями к такой программе является ее универсальность с точки зрения возможного варьирования геометрией и размерами проектируемых ТАД – при минимуме вводимой исходной информации – и только в числовой форме.

Расчетная модель ТАД.

Для иллюстраций к программе взят четырехполюсный ТАД с короткозамкнутой обмоткой ротора мощностью 15 кВт, проектные параметры которого есть в [6]. Он рассчитан на номинальные фазные напряжение $U_{sN}=220$ В и ток статора $I_{sN}=29$ А при частоте $f_s=50$ Гц. Пазы ротора выполняются закрытыми с перемычкой, без скоса; обмотка статора однослойная, диаметрально, распределенная, что характерно для ТАД определенного уровня мощности. Ряд параметров данного ТАД, необходимых для формирования его расчетной модели, приводится далее в файле исходных данных для скрипта Lua.

Для реализации в программе FEMM модель электромагнитной системы ТАД в его поперечном сечении выглядит как на рис. 1. Здесь различной штриховкой проводников в пазах статора показаны фазные зоны обмотки статора $A-A'$, $B-B'$ и $C-C'$, идентифицированные обозначениями мгновенных значений фазных токов i_{sA} , i_{sB} и i_{sC} .

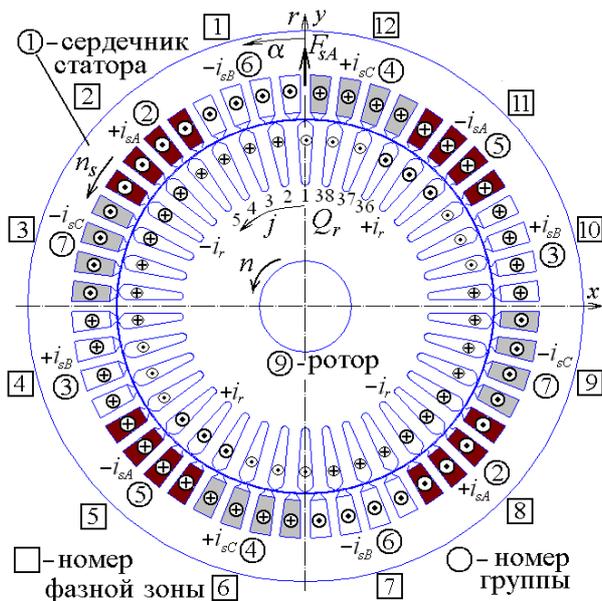


Рис. 1 – Расчетная модель электромагнитной системы ТАД в его поперечном сечении

При построениях моделей ТАД в скриптах Lua используется прямоугольная система координат x, y , а также полярные координаты r, α . Угловая координата

α отсчитывается от оси y и направлена в сторону вращения МП – по стрелке с символом n_s .

В обмотке статора подразумевается симметричная трехфазная система фазных токов:

$$\begin{aligned} i_{sA} &= I_{ms} \cos(\omega_s t); \\ i_{sB} &= I_{ms} \cos(\omega_s t - \frac{2}{3}\pi); \\ i_{sC} &= I_{ms} \cos(\omega_s t + \frac{2}{3}\pi), \end{aligned} \quad (1)$$

где t – время; I_{ms} – амплитуда, ω_s – угловая частота.

В начальный момент времени ток в фазной обмотке A равен амплитудному значению $i_{sA}=I_{ms}$, в B и C – $i_{sB}=i_{sC}=-0,5I_{ms}$. В пазах статора на рис.1 показаны соответствующие направления фазных токов, знаки $+$ и $-$ перед их символами должны быть еще добавлены к их мгновенным значениям.

Принято, что МДС F_{sA} фазной обмотки A на полюсном делении вверху ориентирована по оси y . Так фиксируются фазные обмотки и, соответственно, сердечник статора, поэтому его верхний зубец расположен по вертикальной оси, фазные зоны конструктивно симметричны относительно этой оси, а по направлениям токов – асимметричны. Все остальное занимает свои позиции автоматически в зависимости от значения числа пар полюсов p , которое может быть любым.

Угловая позиция ротора может быть произвольной, а его пронумерованные паза фиксируются заданием координаты первого паза α_{r1} , ближайшего к вертикальной оси, и сдвинуты друг от друга на угол $\alpha_{rn}=360^\circ/Q_r$, где Q_r – число пазов ротора.

Определение тока ротора в режиме номинальной нагрузки. Важным аспектом подготовительной работы к расчету МП в ТАД является определение значений токов обмотки ротора, а также их фазового сдвига α_{sr} по отношению к токам обмотки статора, которые уже заданы с обобщенной нулевой начальной фазой (1). Это возможно при учете проектных данных ТАД с использованием векторной диаграммы, что подробно представлено в [7]. На основе этого в краткой форме представим функцию многофазной системы мгновенных значений токов в стержнях короткозамкнутого ротора, а именно:

$$i_{rk} = I_{mr} \sin\{p[(k-1)\alpha_{rn} + \alpha_{sr} + \alpha_{r1}]\}, \quad (2)$$

где $k=1, 2, \dots, Q_r$ – номер паза соответственно рис.1; $I_{mr} = \sqrt{2} I_r$ – амплитуда фазного тока ротора.

Действующее значение фазного тока ротора I_r и фазовый сдвиг α_{sr} для (2) берутся непосредственно из проектных данных [6] или определяются на их основе по методике, изложенной в [7, 8]. Так, для рассматриваемого ТАД $I_r=440$ А и $\alpha_{sr}=-83^\circ$, причем отрицательное значение угла дает отставание «волны» токов ротора (2) от «волны» токов статора (1).

Общая характеристика скрипта Lua

Lua – это быстрый, легкий и удобный скриптовый язык программирования, который создавался для программирования баз данных. Простой синтаксис и легкость встраивания Lua в другие программы обеспечили Lua широкое распространение. Реализация Lua имеет отдельно компилятор из

исходного языка в выполняемый код и виртуальную машину для выполнения сгенерированного кода. Lua является собой язык с динамическим определением типов данных. Переменные языка могут содержать значение любого типа. Все значения в Lua могут храниться в переменных, использоваться в качестве аргументов при вызове функций и возвращаться в виде результата их выполнения. Lua является собой ядро языка, которое можно встраивать в прикладную программу, например, в FEMM.

Тексты файлов скриптов и файлов данных можно писать в редакторе *Блокнот*. В строках этих файлов все после двойного дефиса (--) программа воспринимает как комментарии и для работы не использует.

В скриптах базовой является команда задания параметров новой *Задачи*: ее тип (МП или иное), единицы измерения размеров, активная длина по аксиальной оси, тип координат, точность и т.д.

Построение геометрических моделей посредством Lua основано на командах расстановки *узлов* – координатных точек (x,y). На их основе своими командами строятся *прямые отрезки* и *дуги*. В образующихся замкнутых подобластях – *блоках* ставятся их метки, через которые им приписываются физические свойства. Они задаются отдельными командами, в которых указываются все необходимые параметры материалов, токов, треугольников конечно-элементной структуры и др. Есть также свои команды для выбора и расстановки *границных условий*.

Перечисленные выше объекты можно объединять в нумерованные *группы*, чтобы оперировать со всеми объектами группы сразу. На рис. 1 в кружочках даны принятые номера групп: 1 – сердечник статора, 2, 5, 3, 6, 4, 7 – токовые структуры в фазных зонах обмотки статора $+i_{sA}$, $-i_{sA}$, $+i_{sB}$, $-i_{sB}$, $+i_{sC}$, $-i_{sC}$, соответственно. Все элементы ротора объединены в общую группу с номером 9 (для поворотов ротора в других программах).

Не вошедшие в перечень объекты остаются в нулевой группе, а номер 8 является резервным.

Lua, как и другие языки программирования, позволяет создавать функции, оперировать с циклами, условными операторами, делать вычисления по формулам, стандартным функциям, например: *abs* – абсолютное значение; *sqrt* – корень квадратный; *sin*, *cos*, *tan*, *acos*, *atan* – синус, косинус, тангенс, арккосинус, арктангенс и др., *Pi* – число π .

Для тригонометрических функций углы надо задавать в радианах, а при построении геометрических объектов Lua оперирует углами в градусах.

Скрипты запускаются из рабочего окна в FEMM «кнопкой» *Open Lua Script* в меню *File*.

Приведем часто встречающиеся команды, в которых (x,y), ($x1,y1$) и ($x2,y2$) – координаты узлов:

1) добавление:

mi_addnode(x,y) – узел;

mi_addsegment($x1,y1,x2,y2$) – отрезок между узлами;

mi_addarc($x1,y1,x2,y2,angle,maxseg$) – дуга между узлами с опорным углом '*angle*' из угловых сегментов размером '*maxseg*';

mi_addblocklabel(x,y) – ярлык блока;

2) выбор ближайших к точке (x,y) объектов:

mi_selectnode(x,y) – узел;

mi_selectsegment(x,y) – отрезок;

mi_selectarcsegment(x,y) – дуга;

mi_selectlabel(x,y) – ярлык блока;

mi_selectgroup(n) – n -ая группа;

3) копирование:

mi_mirror($x1,y1,x2,y2,edit$) – выбранных объектов относительно линии, проходящей через точки ($x1,y1$) и ($x2,y2$), где *edit* – действие редактирования: 0 – для узлов, 1 – отрезков, 2 – дуг, 3 – меток, 4 – групп;

mi_copyrotate($bx, by, angle, copies, edit$), где *copies* – столько раз с поворотом на угол *angle* относительно базовой точки bx, by ; *edit* – см. выше;

4) очистка выделения всех ранее выбранных объектов: *mi_clearselected*();

5) установка свойства материала и токов конкретной подобласти – блока:

mi_setblockprop("blockname", *automesh, meshsize, incircuit, magdir, group, turns*), где "blockname" – имя ярлыка для магнитных свойств блока, *automesh* – выбор способа триангуляции конечных элементов, *meshsize* – ограничивающий размер элемента, "incircuit" – символ тока, *magdir* – направление намагничивания, *group* – номер группы, *turns* – число эффективных проводников с током.

Оператор чтения *read(f_d, "*"n, "*"1)* из очередной строки файла данных берет значения столько числовых или строковых данных, сколько раз в нем фигурирует опция "*"n", а опцией "*"1" будет сделан перевод на новую строку, т.е. оставшийся в строке любой текст игнорируется и используется расчётчиком как комментарий (здесь его принято отделять символом :).

Оператор *write* записывает данные в файл результатов, в нем опция "r\n" дает переход на новую строку, а символ %6.3f является форматом (format) числа, т.е. в данном примере под число выделяется 6 позиций, в том числе 3 – под дробную часть.

Более полный набор команд и процедур находится в приложении к программе FEMM [1].

Конструкторская расчетная модель ТАД показана на рис. 2.

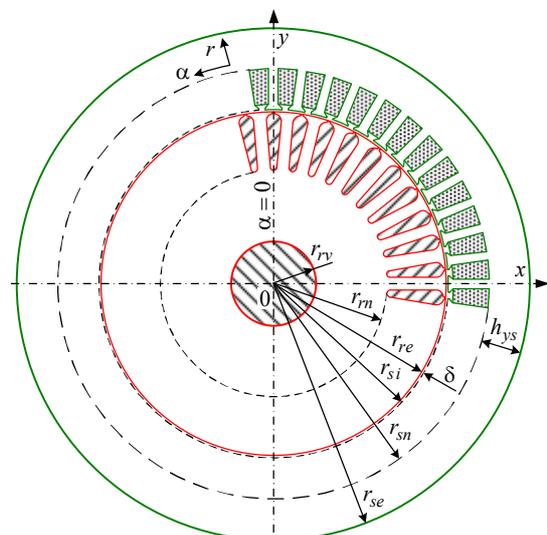


Рис. 2 – Конструкторская расчетная модель ТАД

Представим исходные параметры ТАД, задаваемые в файле данных для полного формирования геометрической и физической моделей ТАД.

К исходными схемно-конструктивным параметрам ТАД (рис. 2) относятся:

- p, m_s – числа пар полюсов и фаз;
- l_a – активная длина статора и ротора;
- k_{Fe} – коэффициент заполнения сердечников статором;

r_v, r_{re}, r_{si} – радиусы окружностей вала и по поверхностям сердечников ротора и расточки статора;

h_{ys} – высота спинки сердечника статора;

Q_r, Q_s, N_s, a_s – числа пазов ротора и статора, последовательных витков и параллельных ветвей фазной обмотки статора;

I_s, I_r, α_{sr} – действующие значения фазных токов статора и ротора, а также упоминавшийся в (2) пространственный угол сдвига «волн» этих токов.

Детализации требуют зубцово-пазовые структуры статора и ротора. Принятые формы их пазов с обозначением минимума размеров и опорных точек представлены на рис. 3 и рис. 4.

Для паза статора на рис. 3 обозначено:

- h_{s1}, b_{s1} – высота и ширина шлица;
- h_s, b_{s2}, b_{s3} – высота и две ширины паза.

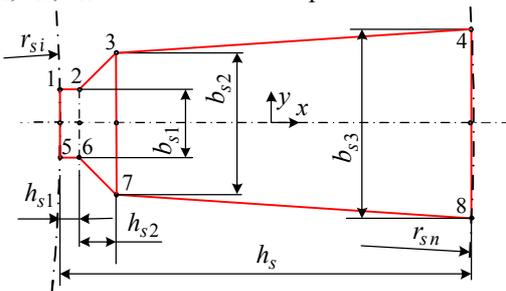


Рис. 3 – Конструкция паза статора

Для паза ротора на рис. 4 приняты обозначения:

- h_r, h_{r0} – высота паза и толщина перемычки (при $h_{r0}=0$ перемычка отсутствует и паз полуоткрытый);
- h_{r1}, b_{r1} – высота и ширина шлица;
- r_{r1}, r_{r2}, h_{r2} – радиусы образующих окружностей и расстояние между их центрами;
- α_{r1} – угол поворота 1-го паза ротора от оси y .

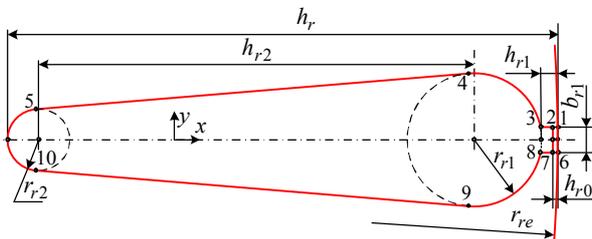


Рис. 4 – Конструкция паза ротора

На основе представленных исходных данных в программе вычисляется целый ряд дополнительных параметров геометрической модели ТАД:

- $r_{se}=r_{si}+h_s+h_{ys}$ – радиус статора внешний;
- $r_{sn}=\sqrt{x^2+y^2}$ – радиус по углам дна пазов статора, где $x=r_{si}+h_s, y=b_{s3}/2$;
- $h_{yr}=r_{re}-h_r-r_{rv}$ – высота спинки сердечника ротора;
- $\delta=r_{si}-r_{re}; r_d=r_{re}+\delta/2$ – воздушный зазор (потом будет δ) и его средний радиус;

$t_s=2\pi r_{si}/Q_s; t_r=2\pi r_{re}/Q_r$ – зубцово-пазовые шаги по расточке статора и по поверхности ротора, мм;

$t_{sa}=360/Q_s; t_{ra}=360/Q_r$ – эти же зубцово-пазовые шаги статора и ротора, но в градусах;

$t_p=Q_s/(2p)$ – полюсный шаг (деление);

$q_{sp}=t_p/m_s$ – число пазов на полюс и фазу;

$N_{cs}=N_s/(p \cdot q_{sp})$ – число эффективных проводников в пазу статора.

Паза статора и ротора строятся по координатам узловых точек (реперные точки), пронумерованных на рис. 3 и рис. 4. Для определения этих координат паза повторены на рис. 5 и рис. 6 с добавлением необходимых обозначений размеров и углов.

Координатные размеры элементов паза статора (рис. 5) вычисляются по формулам: $y_a=b_{s1}/2; y_b=b_{s2}/2;$

$y_c=b_{s3}/2; x_a=\sqrt{r_{si}^2-y_a^2}; x_b=x_a+h_{s1}; x_d=\sqrt{r_{sn}^2-y_c^2};$

$x_c=x_b+h_{s2};$ центральные угловые размеры $u_1=\arctg(y_a/x_a); u_5=t_{sa}/2-u_1$ и дополнительно

– угол наконечника зубца $u_{59}=t_{sa}-2 \cdot u_1$.

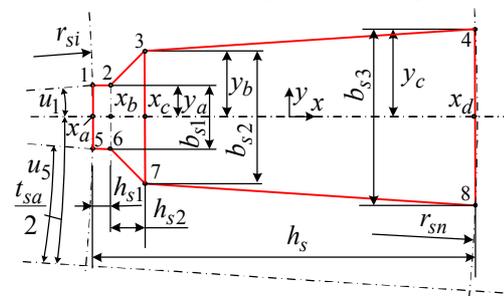


Рис. 5 – Паз статора с обозначением размеров, реперных точек и их координат

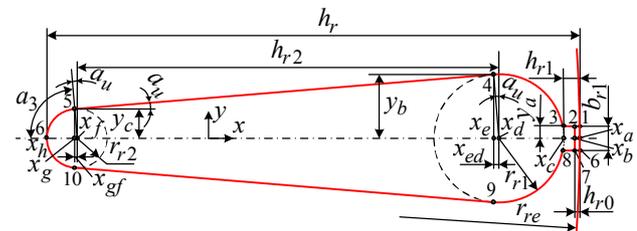


Рис. 6 – Паз ротора с обозначением размеров, реперных точек и их координат

Координатные и угловые размеры элементов паза ротора (рис. 6) вычисляются по формулам с учетом дополнительных обозначений на рис. 7: $y_a=b_{r1}/2;$

$$\alpha_u = \arctg \frac{r_{r1}-r_{r2}}{\sqrt{h_{r2}^2-(r_{r1}-r_{r2})^2}}; \quad \alpha_1 = \arctg \frac{y_a}{\sqrt{r_{r1}^2-y_a^2}};$$

$$x_a = \sqrt{r_{re}^2-y_a^2}; \quad \alpha_2 = 90-\alpha_1+\alpha_u; \quad \alpha_3 = 90-\alpha_u;$$

$$y_b = r_{r1} \cos \alpha_u; \quad x_{ed} = r_{r1} \sin \alpha_u; \quad y_c = r_{r2} \cos \alpha_u;$$

$$x_{gf} = r_{r2} \sin \alpha_u; \quad \alpha_0 = \arctg \frac{y_a}{x_a}; \quad \alpha_6 = \frac{t_{ra}}{2}-\alpha_0;$$

координаты точки 6 (рис. 7, б): $x_6 = r_{re} \sin \alpha_6;$

$$y_6 = r_{re} \cos \alpha_6.$$

Угол поворота из позиции, изображенной на рис. 7, а, в позицию на рис. 1: $a_{ir} = 90^\circ - 0,5t_{ra} + a_{r1}.$

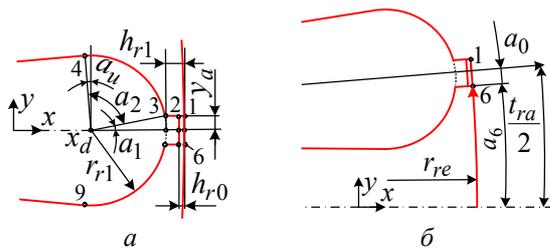


Рис. 7 – Верхня часть паза ротора

Далее в тексте используемые величины обозначены так, как это требуют правила идентификации в скрипте Lua. При этом шрифт становится прямым и индексы поднимаются до уровня основного символа.

Магнитные свойства сердечников ТАД представляются кривыми намагничивания – зависимостями магнитной индукции от напряженности магнитного поля: $B(H)$. Эти кривые вводятся в программу FEMM числовыми массивами (по сути – таблицами), а затем используются программой автоматически.

В ТАД используются две кривые намагничивания: для спинок сердечников и для их зубцов. Условные имена кривых предварительно вводятся из файла исходных данных, также как и имена соответствующих им файлов, откуда берутся массивы кривых намагничивания. Для имен кривых и файлов приняты (произвольно) следующие обозначения:

Sty – имя стали для спинок сердечников;
 bhy – имя файла с кривой намагничивания для Sty;
 Stz – имя стали для зубцов сердечников;
 bhz – имя файла с кривой намагничивания для Stz.

Для сформированной расчетной модели ТАД проводится тестовый расчет его МП в режиме нагрузки и определяется электромагнитный момент.

Управление конечно-элементной структурой расчетной модели ТАД.

Расчет МП методом конечных элементов проводится на конечно-элементной (сеточной) структуре, которая генерируется программой FEMM автоматически. Грубая (с крупными треугольниками) структура приводит к расчетным погрешностям, излишне детализированная структура сопряжена с затягиванием времени расчета МП.

Для варьирования этой структурой между этапами предварительных пробных расчетов и уточненных итоговых расчетов вводятся следующие максимально допустимые размеры сторон треугольников: fe1 – в зазоре, fe2 – в ярме статора, fe3 – в зубцах статора, fe4 – в пазах статора, fe5 – в пазах ротора, fe6 – в зубцах ротора, fe7 – в ярме ротора.

Структура программы построения геометрической и физической моделей ТАД.

Представляемый ниже алгоритм построения моделей ТАД по сути является последовательным. В нем выделены основные логические части программы, выполняемые друг за другом. Аналогичные названия действий можно найти затем и в программе в виде комментариев. Но в программе есть еще много промежуточных комментариев, которые поясняют

локальные действия внутри условно выделенных логических частей в представляемом перечне:

1. Задание имени файла исходных данных.
2. Открытие внешнего файла и чтение из него данных.
3. Организация файла результатов расчета.
4. Назначение имени геометрической модели.
5. Расчет дополнительных параметров на основе исходных данных.
6. Выдача исходных и дополнительных данных в файл результатов.
7. Создание файла для магнитной *Задачи* и задание ее параметров.
8. Организация файла .ftm для запоминания созданной модели ТАД.
9. Создание центральной точки и опорных точек для поверхности статора, создание этой поверхности.
10. Создание зубцово-пазовой структуры статора.
11. Создание зубцово-пазовой структуры ротора.
12. Создание окружности вала ротора.
13. Дорисовка вспомогательной окружности в роторе под его зубцами, объединяющей их в один блок.
14. Создание свойства материалов с магнитной проницаемостью воздуха.
15. Создание параметров сердечников – имен магнитных свойств.
16. Ввод кривых намагничивания сталей.
17. Подготовка фазных токов обмотки статора.
18. Подготовка параметров геометрической структуры обмотки статора.
19. Задание материала и токов в пазах фазных зон обмотки статора.
20. Задание материала и распределение токов в пазах ротора.
21. Задание магнитных свойств зазора.
22. Задание материала для ярма статора.
23. Задание материала для зубцов статора.
24. Задание материала для зубцов ротора.
25. Задание материала для ярма ротора.
26. Задание свойства материала вала ротора.
27. Создание и назначение граничных условий Дирихле на внешней поверхности сердечника статора.
28. Выполнение тестового расчета магнитного поля.
29. Переход к показу и извлечению результатов.
30. Тестовый расчет электромагнитного момента.

Файлы исходных данных для программы на скрипте Lua.

Файл исходных данных пишется, как и программа, в текстовом редакторе «Блокнот».

Этот файл начинается с двух строк комментариев, которые не участвуют в работе программы. Но эти строки при вводе информации главной программой должны быть «сняты» как текстовые строки.

В строках исходных данных для примера взяты числовые данные для обусловленного ТАД. И для главной программы вводятся и представляют интерес именно числа, либо текстовые обозначения в двойных кавычках (" ") соответствующих строковых данных. Все остальное, как отмечалось, программа воспринимает как комментарии и для работы не использует.

Единица измерения геометрических размеров должна быть такой, которая указана в задаваемой ин-

струкции для главного процессора программы FEMM – обычно принимаются миллиметры.

В следующем тексте файла данных содержатся все исходные параметры уже представленные выше, а потом – кривые намагничивания сталеи).

Содержание файла данных для построения физико-геометрической модели ТАД

Файл данных для построения расчетной модели ТАД программой LuaTAD_ "TAD_15_2" : name_TAD - имя модели ТАД "08.03.2017" : data_dan - дата данных "RezTAD_15_2" : name_rez - файл результатов
 29 : Is - фазный ток статора, А
 440 : Ir - фазный ток ротора, А
 82.55 : asr - угол сдвига токов, град
 50 : fs - частота тока статора
 2 : p - число пар полюсов
 3 : ms - число фаз обмотки статора
 130 : la - активная длина статора
 1 : as - число парал. ветвей обм. статора
 48 : Qs - число пазов статора
 112 : Ns - число витков фазы статора
 92.5 : rsi - радиус расточки статора
 21.7 : hys - высота спинки статора
 21.8 м hs - высота паза статора
 1 : hs1 - высота шлица паза статора
 2 : hs2 - высота скошенной части паза
 3.7 : bs1 - ширина шлица паза статора
 7.7 : bs2 - ширина узкой части паза
 10.2 : bs3 - ширина широкой части паза
 38 : Qr - число пазов ротора
 92 : rre - радиус ротора внешний
 22.5 : rrv - радиус вала
 32 : hr - высота паза ротора
 0.7 : hr1 - высота шлица паза ротора
 0.3 : hr0 - высота переемычки паза рот.
 25.3 : hr2 - высота между центрами паза
 1.5 : br1 - ширина шлица паза ротора
 3.9 : rr1 - 1-ый радиус паза ротора
 1.9 : rr2 - 2-ой радиус паза ротора
 0 : ar1 - угол 1-го паза рот. от оси у
 0.97 : kFe - коэфф. заполнен. сердечников
 "Sty" : Sty - имя стали спинки сердеч.
 "2013y" : bhy - имя файла с кривой Sty
 "Stz" : Stz - имя стали зубцов сердеч.
 "2013z" : bhz - имя файла с кривой Stz
 0.1 : fe1 - макс. разм. треугольн зазора
 5 : fe2 - в ярме статора
 1.5 : fe3 - в зубцах статора
 1.5 : fe4 - в пазах статора
 1.5 : fe5 - в пазах ротора
 1.5 : fe6 - в зубцах ротора
 5 : fe7 - в ярме ротора

Файлы данных с кривыми намагничивания

Каждая из упомянутых выше кривых намагничивания вводится числовым массивом парами чисел: магнитная индукция – напряженность магнитного поля. Числовые данные предваряются заголовком-комментарием, который при вводе информации главной программой должен быть «снят».

Далее приводятся числовые данные файлов с представленными выше именами: "bhy" – для ярма и "bhз" – для зубцов сердечников, выполненных из электротехнической стали марки 2013.

Сталь 2013 ярма: В-Н (Тл, А/м) - bhy
 0.40 60 0.80 124 0.85 138 0.90 152

0.95 168 1.00 185 1.05 203 1.10 221
 1.15 241 1.20 262 1.25 289 1.30 320
 1.35 357 1.40 400 1.45 450 1.50 520
 1.55 630 1.60 750 1.65 940 1.70 1150
 1.75 1500 1.80 2000 1.85 2810 1.90 4070
 1.95 6300 2.00 9190 2.05 14390
 2.10 23000 2.15 45000 2.20 80000
 2.25 117000 2.30 155000 0 0

Сталь 2013 зубцов: В-Н (Тл, А/м) - bhз
 0.20 60 0.40 124 0.45 138 0.50 154
 0.55 171 0.60 188 0.65 205 0.70 223
 0.75 240 0.80 256 0.85 271 0.90 286
 0.95 304 1.00 324 1.05 346 1.10 370
 1.15 396 1.20 424 1.25 455 1.30 486
 1.35 533 1.40 586 1.45 646 1.50 709
 1.55 777 1.60 850 1.65 990 1.70 1150
 1.75 1330 1.80 1520 1.85 1770 1.90 2200
 1.95 2890 2.00 3400 2.05 4400
 2.10 5500 2.15 7100 2.20 9100 2.25 12000
 2.30 15000 2.35 23000 2.40 45000
 2.45 80000 2.50 117000 2.55 155000 0 0

Нулевую начальную точку кривых намагничивания программа FEMM добавляет сама, а нули в конце файлов воспринимаются программой как конец вводимого массива кривой намагничивания.

Программная реализация расчетной модели ТАД на скрипте Lua

Далее приводится полный текст программы, которая после ее вызова в среде FEMM формирует геометрическую и физическую модели ТАД и запускает программу FEMM на пробный тестовый расчет МП.

Строки программы, начинающиеся с двойного дефиса --, как уже отмечалось, являются свободными комментариями и никак не влияют на работу программы. Эти комментарии объясняют суть выполняемых действий в остальных – командных и вычислительных строках.

По ходу представляемого далее листинга программы вставлены рисунки, показывающие результат выполнения тех или иных команд и процедур на данный момент ее работы. В программе используется нумерация точек пазов статора и ротора, представленная на рис. 3 – рис. 7.

```

pause() --пауза перед началом работы
name_prog=" LuaTAD MC" --имя программы
data_prog=" 02.03.2017" --дата версии
--Задание файла исходных данных
f_d=prompt("Введите имя файла с данными
=> DanTAD_PN_p")
--Открытие файла данных
f_d = openfile(f_d .. ".txt","r")
--Считывание двух строк заголовка в нем
for i=1,2 do str = read(f_d,"*1") end
gr=Pi/180 --коэф.из градусов в радианы
--Ввод данных из файла f_d
name_TAD=read(f_d,"*", "*1") --имя ТАД
data_dan=read(f_d,"*", "*1") --дата
name_rez=read(f_d,"*", "*1") --файл
рез.
Is=read(f_d,"*", "*1") --ток статора
Ir=read(f_d,"*", "*1") --ток ротора
asr=read(f_d,"*", "*1") --угол сдвига
fs=read(f_d,"*", "*1") --частота
p=read(f_d,"*", "*1") --числ.пар полюс.
ms=read(f_d,"*", "*1") --числ.фаз стат.
    
```

```

la=read(f_d,"*n","*1") --активная длина
--Параметры и размеры сердечн. статора
as=read(f_d,"*n","*1") --чис.парал.вет.
Qs=read(f_d,"*n","*1") --чис.паз.стат.
Ns=read(f_d,"*n","*1") --чис.вит.стат.
rsi= read(f_d,"*n","*1") --радиус раст.
hys=read(f_d,"*n","*1") --высота спинки
hs =read(f_d,"*n","*1") --высота паза
hs1=read(f_d,"*n","*1") --высота шлица
hs2 = read(f_d,"*n","*1") --высота
--скошенной части паза статора
bs1=read(f_d,"*n","*1") --ширина шлица
--Ширина узкой и широкой частей паза
bs2=read(f_d,"*n","*1")
bs3=read(f_d,"*n","*1")
--Параметры и размеры сердечн. ротора
Qr=read(f_d,"*n","*1") --числ.паз.рот.
rre=read(f_d,"*n","*1") --радиус внеш.
rrv=read(f_d,"*n","*1") --радиус вала
hr=read(f_d,"*n","*1") --высота паза
hrl=read(f_d,"*n","*1") --высота шлица
hr0=read(f_d,"*n","*1") --выс.перемычки
hr2=read(f_d,"*n","*1") --меж. центров
brl=read(f_d,"*n","*1") --ширина шлица
rr1= read(f_d,"*n","*1") --1-ый радиус
rr2= read(f_d,"*n","*1") --2-ой радиус
arl=read(f_d,"*n","*1") --угол 1-го па-
за
kFe=read(f_d,"*n","*1") --коэф.заполн.
Sty=read(f_d,"*n","*1") --сталь спинок
bhy=read(f_d,"*n","*1") --файл для Sty
Stz=read(f_d,"*n","*1") --сталь зубцов
bhZ=read(f_d,"*n","*1") --файл для Stz
--Максимальные размеры треугольников
fe1=read(f_d,"*n","*1") -- в зазоре
fe2=read(f_d,"*n","*1") --в ярме стат.
fe3=read(f_d,"*n","*1") --зубц.статора
fe4=read(f_d,"*n","*1") --пазы статора
fe5=read(f_d,"*n","*1") --пазы ротора
fe6=read(f_d,"*n","*1") --зубцы ротора
fe7=read(f_d,"*n","*1") --ярмо ротора
closefile(f_d) --закрытие файла данных
-- Досчет геометрических параметров
rse=rsi+hs+hys --радиус стат. внешний
--Радиус по углам дна пазов статора
x=rsi+hs y=bs3/2 rsn=sqrt(x*x+y*y)
hyr=rre-hr-rrv --высота спин.серд.рот.
delta=rsi-rre --воздушный зазор
rd=rre+delta/2 --его средний радиус
--Пазовые шаги статора и ротора,мм
ts =2*Pi*rsi/Qs tr=2*Pi*rre/Qr
--Пазовые шаги статора и ротора,град.
tsa =360/Qs tra=360/Qr
tp=Qs/(2*p) --полюсный шаг
qsp=tp/ms --число пазов на полюс и фазу
Ncs=Ns/(p*qsp) --число эффективных
--проводников в пазу статора
--Организация файла результатов
writeto(name_rez) --name_rez - имя
--Имя геометрич. модели ТАД - тип fem
geom_mod="Fe Md_"..name_TAD.."fem"
--Выдача данных в файл name_rez
write("ПОСТРОЕНИЕ МОДЕЛИ ТАД"
..name_TAD.." от ".data_dan,"\r\n")
write(" Программа ",name_prog," от ",
data_prog,"\r\n")
write(format(" Is=%5.1f",Is),
format(" Ir=%5.1f",Ir),
format(" asr=%6.2f",asr),"\r\n")
write(format(" p=%1.0f",p),
format(" Qs=%3.0f",Qs),
format(" Qr=%3.0f",Qr),
format(" ms=%1.0f",ms),
format(" qsp=%1.0f",qsp),
format(" tp=%2.0f",tp),"\r\n")
write(format(" Ns=%2.0f",Ns),
format(" Ncs=%2.0f",Ncs),
format(" as=%2.0f",as),"\r\n")
write(format(" la=%3.0f",la),
format(" delta=%4.2f",delta),
format(" rd=%6.2f",rd),"\r\n")
write(format(" rrv=%6.2f",rrv),
format(" rre=%6.2f",rre),
format(" rsi=%6.2f",rsi),
format(" rsn=%6.2f",rsn),
format(" rse=%6.2f",rse),
format(" kFe=%6.3f",kFe),"\r\n")
write(format(" hys=%5.1f",hys),
format(" hs=%6.2f",hs),
format(" hs1=%4.1f",hs1),
format(" hs2=%4.2f",hs2),
format(" bs1=%4.1f",bs1),
format(" bs2=%4.2f",bs2),
format(" bs3=%4.2f",bs3),"\r\n")
write(format(" hr=%6.2f",hr),
format(" hrl=%4.1f",hrl),
format(" hr0=%4.1f",hr0),
format(" hyr=%5.1f",hyr),
format(" hr2=%5.2f",hr2),"\r\n")
write(format(" brl= %4.1f",brl),
format(" rr1= %4.1f",rr1),
format(" rr2= %4.1f",rr2),"\r\n")
write(format(" arl=%5.2f",arl),
format(" tr=%5.2f",tr),
format(" ts=%5.2f",ts),
format(" tra=%5.2f",tra),
format(" tsa=%5.2f",tsa),"\r\n")
write(" Сталь спинок сердечников ",Sty,
" по кривой намагнич.",bhy,"\r\n")
write(" Сталь зубцов сердечников ",Stz,
" по кривой намагнич.",bhZ,"\r\n")
-- НАЧИНАЕТСЯ РАБОТА ПРОГРАММЫ FEMM
--Создание документа магнитной задачи
Start_date=date() create(0)
--Задание параметров задачи
mi_probdef(0, "millimeters", "planar",
1E-8, la)
--Организация файла FEMM с именем
--geom mod для сохранения модели ТАД
mi_saveas(geom_mod)
-----
--ФОРМИРОВАНИЕ ГЕОМЕТРИЧЕСКОЙ МОДЕЛИ ТАД
--Внешняя поверхность сердечн. статора
--Точки центра, верха и низа окружности
mi_addnode(0,0)
mi_addnode(0,rse) mi_addnode(0,-rse)
-- Дуги поверхности статора
m_seg=5 --размер сегмента дуг, градусы
--Левая и правая полуокружности
mi_addarc(0,rse,0,-rse,180,m_seg)
mi_addarc(0,-rse,0,rse,180,m_seg)
--Создание граничных условий Дирихле
mi_addboundprop("A=0",0,0,0,0,0,0,0)
--Выделение дуг и задание условия
--Дирихле на поверхности сердечн. стат.
mi_selectarcsegment(rse,0)
mi_selectarcsegment(-rse,0)
mi_setarcsegmentprop(m_seg,"A=0",0,0)

```

```
mi_clearselected() --снимаем выделение
-----
--Создание зубцово-пазового слоя
--статора и объединение его в группу 1
--Координатные размеры элементов паза
--статора (смотри рис. 3 и 5)
ya=bs1/2 yb=bs2/2 yc=bs3/2
xa=sqrt(rsi^2-ya^2) xb=xa+hs1 xc=xb+hs2
xd=xa+hs
--Ставятся точки от 1 до 4
mi_addnode(xa, ya) mi_addnode(xb, ya)
mi_addnode(xc, yb) mi_addnode(xd, yc)
--Проводятся линии между этими точками
mi_addsegment(xa, ya, xb, ya)
mi_addsegment(xb, ya, xc, yb)
mi_addsegment(xc, yb, xd, yc)
--Размечаются эти точки и линии
mi_selectnode(xa, ya)
mi_selectnode(xb, ya)
mi_selectnode(xc, yb)
mi_selectnode(xd, yc)
x=(xa+xb)/2 mi_selectsegment(x, ya)
x=(xb+xc)/2 y=(ya+yb)/2
mi_selectsegment(x, y) x=(xc+xd)/2
y=(yb+yc)/2 mi_selectsegment(x, y)
--Назначение точкам и линиям группы 1
mi_setnodeprop(" ", 1) --точки
mi_setsegmentprop(" ", " ", 1, 0, 1) --
линия
mi_clearselected() --Рис. 8, а
```

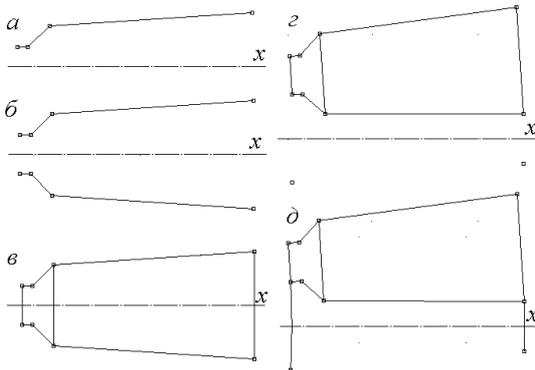


Рис. 8 – Шаги построения паза статора

```
--Выделение полученной стенки паза
--и ее отображение симметрично оси x
mi_selectgroup(1)
mi_mirror(0, 0, rse, 0, 4)
mi_clearselected() --Рис. 8, б
--Проведение и разметка линий 1-5, 3-7
-- и 4-8 и назначение им группы 1
mi_addsegment(xa, ya, xa, -ya)
mi_addsegment(xc, yb, xc, -yb)
mi_addsegment(xd, yc, xd, -yc)
mi_selectsegment(xa, 0)
mi_selectsegment(xc, 0)
mi_selectsegment(xd, 0)
mi_setsegmentprop(" ", " ", 1, 0, 1)
mi_clearselected() --Рис. 8, в
--Выделение и поворот паза на
--ползубц. шага вверх над осью x
mi_selectgroup(1)
mi_moverotate(0, 0, tsa/2, 4)
--Определение координат тчк 5, 8 и
--копия их под ось x -это тчк. 9, 10
u5=0.5*tsa*gr-atan(ya/xa)
x5=rsi*cos(u5) y5=rsi*sin(u5)
```

```
u8=0.5*tsa*gr-atan(yc/xd)
x8=rsn*cos(u8) y8=rsn*sin(u8)
mi_selectnode(x5, y5)
mi_selectnode(x8, y8)
mi_mirror(0, 0, rse, 0, 0)
--Рис. 8, г
--Проводим дуги между тчк 5-9 и 8-10
m_seg=1 ua=2*u5/gr ub=2*u8/gr
mi_addarc(x5, -y5, x5, y5, ua, m_seg)
mi_addarc(x8, -y8, x8, y8, ub, m_seg)
--Выделение дуг и назначение группы 1
mi_selectarcsegment(rsi, 0)
mi_selectarcsegment(rsn, 0)
mi_setarcsegmentprop(1, " ", 0, 1)
mi_clearselected()
--Рис. 8, д
--Поворот зубцового деления по оси y
mi_selectgroup(1)
mi_moverotate(0, 0, 90, 4)
--Копирование зубцов и получение полной
--зубцово-пазовой структуры статора
mi_selectgroup(1)
mi_copyrotate(0, 0, tsa, (Qs-1), 4)
-----
--Создание зубцово-пазового слоя
--ротора и объединение его в группу 9
--Координатные размеры паза ротора
--и его элементов (см. рис.4, 6, 7)
y=rr1-rr2 x=sqrt(hr2^2-y^2)
au=atan(y/x) ya=br1/2
x=sqrt(rr1^2-ya^2)
a1=atan(ya/x) a2=90+(au-a1)/gr
a3=90-au/gr yb=rr1*cos(au)
yc=rr2*cos(au)
xa=sqrt(rre^2-ya^2) xb=xa-hr0 xc=xa-hr1
xed=rr1*sin(au) xgf=rr2*sin(au)
xh=xa-hr xf=xh+rr2 xg=xf-xgf xd=xf+hr2
xe=xd-xed
--Ставятся точки от 1 до 6-ой
mi_addnode(xa, ya)
if hr0>1e-2 then mi_addnode(xb, ya) end
mi_addnode(xc, ya) mi_addnode(xe, yb)
mi_addnode(xg, yc) mi_addnode(xh, 0)
--Проводятся линии между тчк 1-2-3, 4-5
if hr0>1e-2 then --если есть перемычка
mi_addsegment(xa, ya, xb, ya)
mi_addsegment(xb, ya, xc, ya)
else mi_addsegment(xa, ya, xc, ya) end
mi_addsegment(xe, yb, xg, yc)
--Проводятся дуги между тчк 3-4 и 5-6
m_seg=10 --размер сегмента дуг, градус
mi_addarc(xc, ya, xe, yb, a2, m_seg)
mi_addarc(xg, yc, xh, 0, a3, m_seg)
--Размечаются точки от 1 до 6-ой
mi_selectnode(xa, ya)
if hr0>1e-2 then
mi_selectnode(xb, ya) end
mi_selectnode(xc, ya)
mi_selectnode(xe, yb)
mi_selectnode(xg, yc)
mi_selectnode(xh, 0)
--Разметка линий между тчк 1-2-3, 4-5
if hr0>1e-2 then x=(xa+xb)/2
mi_selectsegment(x, ya) end
x=(xb+xc)/2 mi_selectsegment(x, ya)
x=(xe+xg)/2 y=(yb+yc)/2
mi_selectsegment(x, y)
--Разметка дуг между точками 3-4, 5-6
y=rr1*sin(Pi/4) x=xd+y
```

```

mi_selectarcsegment(x,y)
y=rr2*sin(Pi/4) x=xf-y
mi_selectarcsegment(x,y)
--Назнач. тчк, линиям и дугам группы 9
mi_setnodeprop(" ",9)
mi_setsegmentprop(" "," ",1,0,9)
mi_setarcsegmentprop(m_seg," ",0,9)
mi_clearselected() --Рис. 9, а
--Выделение полученной стенки паза
--и отображение ее симметрично оси x
mi_selectgroup(9)
mi_mirror(0,0,rre,0,4)
mi_clearselected() --Рис. 9, б
--Проводятся и размечаются линия 2-7 и
--дуга 1-6 и им назначается группа 9
a0=2*atan(ya/xa) m_seg=0.5

```

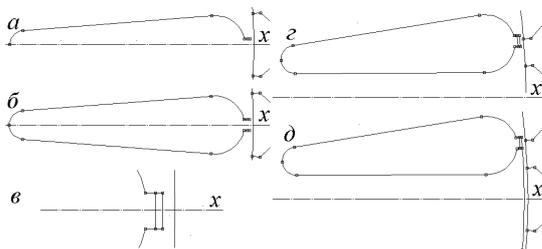


Рис. 9 – Шаги построения паза ротора

```

if hr0>1e-2 then --если есть перемычка
mi_addsegment(xb,ya,xb,-ya) end
mi_addarc(xa,-ya,xa,ya,a0,m_seg)
if hr0>1e-2 then
mi_selectsegment(xb,0) end
mi_selectarcsegment(rre,0)
if hr0>1e-2 then
mi_setsegmentprop(" "," ",1,0,9) end
mi_setarcsegmentprop(m_seg," ",0,9)
mi_clearselected() --Рис. 9, в
--Выделение и поворот паза на
--ползубцового шага вверх над осью x
mi_selectgroup(9)
mi_moverotate(0,0,tra/2,4)
--Рис. 9, г
--Определение координат тчк б и копия
--ее под ось x - это тчк. 11
u6=tra*gr/2-atan(ya/xa) x6=rre*cos(u6)
y6=rre*sin(u6) mi_addnode(x6,-y6)
--Проводим дугу между точками 6-11
m_seg=1 u=2*u6/gr
mi_addarc(x6,-y6,x6,y6,u,m_seg)
--Выделяем тчк. 11 и дугу окружности
-- и назначаем им группу 9
--Рис. 9, д
mi_selectnode(x6,-y6)
mi_selectarcsegment(rre,0)
mi_setnodeprop(" ",9)
mi_setarcsegmentprop(m_seg," ",0,9)
mi_clearselected()
--Зубцовое деление ротора ставится в
--плановую позицию первого паза
mi_selectgroup(9) u=90-tra/2+ar1
mi_moverotate(0,0,u,4)
--Это деление копируется и получается
-- полная структура пазов ротора

```

```

mi_selectgroup(9)
mi_copyrotate(0,0,tra,(Qr-1),4)
-----
--Рисуем окружность вала ротора
--Точки верха и низа окружности
mi_addnode(0,rrv) mi_addnode(0,-rrv)
m_seg=10
--Левая и правая полуокружности
mi_addarc(0,rrv,0,-rrv,180,m_seg)
mi_addarc(0,-rrv,0,rrv,180,m_seg)
--Рисуем окружность под пазами ротора
rrn=xh-rr2/3 --радиус окружности
--Точки верха и низа круга
mi_addnode(0,rrn) mi_addnode(0,-rrn)
--Левая и правая полуокружности
mi_addarc(0,rrn,0,-rrn,180,m_seg)
mi_addarc(0,-rrn,0,rrn,180,m_seg)
--Размечаются точки центра ТАД и дуг
-- и дуги вала под пазами ротора
mi_selectnode(0,0) mi_selectnode(0,rrv)
mi_selectnode(0,-rrv)
mi_selectarcsegment(rrv,0)
mi_selectarcsegment(-rrv,0)
mi_selectnode(0,rrn)
mi_selectnode(0,-rrn)
mi_selectarcsegment(rrn,0)
mi_selectarcsegment(-rrn,0)
--Выделенным точкам и дугам - группа 9
mi_setnodeprop(" ",9)
mi_setarcsegmentprop(m_seg," ",0,9)
mi_clearselected() --снятие выделения
-----
--Рисуем окружность посередине зазора
m_seg=1 --допустимый размер сегмента
--Точки справа и слева на окружности
mi_addnode(rd,0) mi_addnode(-rd,0)
--Верхняя и нижняя полуокружности
mi_addarc(rd,0,-rd,0,180,m_seg)
mi_addarc(-rd,0,rd,0,180,m_seg)
pause() --геометрическая модель готова
--Рис. 10
--*****

```

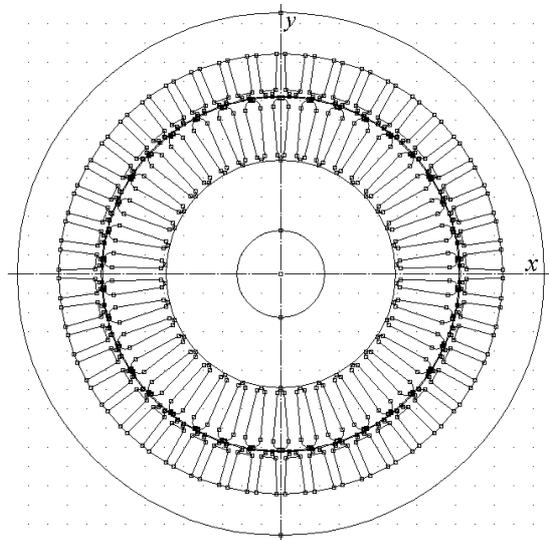


Рис. 10 – Геометрическая модель ТАД

```
--СОЗДАНИЕ ФИЗИЧЕСКОЙ МОДЕЛИ ТАД
--Свойства материалов и токи обмоток
-----
--Материал с немагнитными свойствами
mi_addmaterial("muo",1,1) --muo - имя
--Задание материала в частях зазора
c=delta/4 x1=rd-c x2=-rd-c
--Расстановка меток свойств
mi_addblocklabel(x1,0)
mi_addblocklabel(x2,0)
mi_selectlabel(x1,0)
mi_selectlabel(x2,0)
--Присвоения меткам немагнитных св-ств
mi_setblockprop("muo",0,fe1," ",0,0,0)
mi_clearselected()
-----
--Ввод кривых намагничивания стали-B(H)
--Функция ввода B(H) и св-в сердечников
function B_H(bh_name,St,dsl,kFe)
--Материал участка магнитопровода
mi_addmaterial(St,0,0,0,0,dsl,0,kFe,0,0)
--Открытие файла с именем bh_name
kriv=openfile(bh_name .. ".txt","r")
str=read(kriv,"*l") --снятие заголовка
repeat --оператор возврата цикла чтения
--Чтение магн. индукции и напряженности
b=read(kriv,"*n") h=read(kriv,"*n")
--Приписывание значений B и H имени St
mi_addbpoint(St,b,h)
--Проверка достижения конца файла
until (b==0) and (h==0)
--Закрытие файла и конец function B_H
closefile(kriv) end
--Подготовка свойств участков
--магнитопровода через вызов функции
--Кривая намагнич. спинок сердечников
mvi=get_B_H(bhy,Sty,0.5,kFe)
--Кривая намагнич. зубцов сердечников
mvi=get_B_H(bhz,Stz,0.5,kFe)
-----
--Материал и токи в пазах статора
--Амплитуда и значения фазных токов
Ism=Is*2^0.5/as c=2*Pi/3
IA=Ism*cos(0) IB=Ism*cos(-c)
IC=Ism*cos(c)
--Функция для задания токов и свойств
--в пазах фазных зон обмотки статора
function tok_faz(n,is_name,is,N_grp)
-- n - номер фазной зоны (см.рис.1),
--is_name -имя тока, is-значение тока,
--N_grp - номер группы
r=rsi+hs/2 --средний радиус пазов
--Связка имени тока со значением тока
mi_addcircprop(is_name,is,1)
for z=1,p do --перебор пар полюсов
m=n+(z-1)*2*ms --номер фазной зоны
--Цикл перебора пазов фазной зоны
for q=1,qsp do i=(m-1)*qsp+q
--Угловая позиция и координаты паза
ai=(i-0.5)*tsa*gr
x=-r*sin(ai) y=r*cos(ai)
--Добавление и выделение метки свойств
```

```
mi_addblocklabel(x,y)
mi_selectlabel(x,y)
--Задание свойств и тока в пазу
mi_setblockprop("muo",0,fe4,is_name,0,N_grp,Ncs)
mi_clearselected() end --конец цикла q
end --конец перебора пар полюсов по z
end --конец function tok_faz
--Распределение токов по шести фазным
--зонам обмотки стат. на всех полюсах
zap=tok_faz(2,"IA+",IA,2)
zam=tok_faz(5,"IA-",IA,5)
zap=tok_faz(4,"IB+",IB,3)
zam=tok_faz(1,"IB-",IB,6)
zap=tok_faz(6,"IC+",IC,4)
zam=tok_faz(3,"IC-",IC,7)
-----
--Материал и токи в пазах ротора
--Амплитуда и имя фазного тока ротора
Imr=Ir*2^0.5 str="Ir" Ncr=1
r=xf+hr2/2 --средний радиус стержней
--Перебор пазов ротора и их координата
for z=1,Qr do ai=((z-1)*tra+ar1)*gr
--Фазы и токи стержней обмотки ротора
aci=p*((z-1)*tra+asr+ar1)*gr
irq=Irm*sin(aci)
--Координаты стержней в пазу
x=-r*sin(ai) y=r*cos(ai)
cir_name=str..z --имя очередного тока
--Связь имени тока с его значением
mi_addcircprop(cir_name,irq,1)
--Добавление и выделение метки свойств
mi_addblocklabel(x,y)
mi_selectlabel(x,y)
--Задание свойств тока в стержнях
mi_setblockprop("muo", 0, fe5,
cir_name, 0, 9, Ncr)
mi_clearselected()
end --конец шагов по пазам ротора
-----
--Задание материалов и свойств
--участкам статора и ротора
-----
--Материал в зоне клиньев статора
--Перебор пазов статора и их угол
for z=1,Qs do ai=((z-0.5)*tsa)*gr
--Координаты зон клиньев в пазу
r=rsi+hs1 x=r*sin(ai) y=r*cos(ai)
mi_addblocklabel(x,y)
--Выделение зоны и задание свойств
mi_selectlabel(x,y)
mi_setblockprop("muo",0,fe4," ",0,1,0)
mi_clearselected() end --пазов статора
--Материал для спинки сердечн. статора
y=rse-hys/2 mi_addblocklabel(0,y)
mi_selectlabel(0,y)
mi_setblockprop(Sty,0,fe2," ",0,1,0)
mi_clearselected()
--Материал для зубцов сердечн.статора
for z=1,Qs do --цикл зубцов и их углы
--Полярные и прямоуг. координ. зубцов
ai=((z-1)*tsa)*gr ris=rsi+hs/2
```

```

x=-ris*sin(ai) y=ris*cos(ai)
--Добавка и выделение меток свойств
mi_addblocklabel(x,y)
mi_selectlabel(x,y)
--Задание свойства метки
mi_setblockprop(Stz,0,fe3," ",0,1,0)
mi_clearselected() end --зубцов статора
--Материал для ярма ротора
--Метка свойств и ее выделение
y=rrv+hvr/2 mi_addblocklabel(0,y)
mi_selectlabel(0,y)
--Задание свойств метки
mi_setblockprop(Sty,0,fe7," ",0,9,0)
mi_clearselected()
--Материал для зубцов ротора, которые
--соединены через зону под ними
--Координаты для метки и ее добавление
u=(ar1+tra/2)*gr r=rre-hr/2 y=r*cos(u)
x=-r*sin(u) mi_addblocklabel(x,y)
--Выделение метки и задание ее свойств
mi_selectlabel(x,y)
mi_setblockprop(Stz,0,fe6," ",0,9,0)
mi_clearselected() --снять выделение
--Материал для перемычки ротора
if hr0>1e-2 then --если она есть
--Перебор пазов ротора и их координаты
for z=1,Qr do
ai=((z-1)*tra+ar1)*gr rir=rre-hr0/2
x=-rir*sin(ai) y=rir*cos(ai)
--Метка свойств и ее выделение
mi_addblocklabel(x,y)
mi_selectlabel(x,y)
--Задание свойств метки
mi_setblockprop(Stz,0,fe1," ",0,9,0)
mi_clearselected()
end end --перебора перемычек
--Материал вала ротора
--Метка для материала и ее выделение
y=rrv/2 mi_addblocklabel(0,y)
mi_selectlabel(0,y)
--Задание свойства метки
mi_setblockprop("muo",0,5," ",0,9,0)
mi_clearselected() --снять выделение
-----
--ВЫПОЛНЕНИЕ РАСЧЕТА МАГНИТНОГО ПОЛЯ
mi_analyze(1)
--Перейти к показу результатов расчета
mi_loadsolution()
-----
--Тестовый расчет электромагн. момента
write("Электромагнитные момент и",
" мощность","\r\n")
--Выделение блока ротора
mo_groupselectblock(9)
--Считывание электромагнитного момента
Mem=mo_blockintegral(22) Mem=abs(Mem)
mo_clearblock() --снято выделение
--Электромагнитная мощность ТАД
Pem=Mems*2*Pi*fs/p
write(format("Mem=%6.2f",Mem)," Н;",
format(" Pem=%6.2f",Pem*1e-3)," кВт",
"\r\n")

```

```

write("Завершение программы","\r\n")
writeto() --Закрытие файла результатов
mi_close() --Закрыт документ
--препроцессора магнетизма
--и уничтожено его окно

```

Результаты работы программы

Построенная физико-геометрическая модель ТАД представлена на рис. 11 – она скопирована с экрана компьютера в программной среде FEMM.

На рис. 12 даны наглядные результаты расчета МП. Здесь показаны тонированное распределение плотности тока в пазах статора и ротора, картина силовых линий МП, векторы магнитной индукции.

Кроме этого, естественно, получается весьма большой объем числовой информации, которая представлена в [7, 8] и других работах. А в принципе, объем этой информации не ограничен и все зависит от необходимости оперирования с нею.

Из всего листинга файла результатов для примера показываем только его начальную и конечную части, а именно:

```

ПОСТРОЕНИЕ МОДЕЛИ ТАД
TAD_15_2_n от 08.03.2017
Программа LuaTAD_MC от 02.03.2017
Is= 29.0 Ir=442.0 asr= 82.55
p=2 Qs= 48 Qr= 38 ms=3 qsp=4 tp=12
Ns=112 Ncs=14 as= 1
. . . . .
Электромагнитные момент и мощность
Mems=103.19 Н; Pem= 16.21 кВт

```

Успешное завершение всей программы расчетов

Инструкция по использованию скрипта.

Для использования скрипта построения физико-геометрической модели ТАД, необходимо следующее:

1. Файл скрипта с расширением *.lua, а также файлы с исходными данными для построения геометрии и файлы с данными для построения кривых намагничивания должны находиться в одной папке.

2. Запускаем программу FEMM и в меню File выбираем опцию **Open Lua script**.

3. Далее в появившемся окне надо выбрать имеющуюся папку со скриптом Lua, выбрать или ввести необходимое имя файла с программой построения физико-геометрической модели ТАД и клавишей Enter или двойным кликом клавиши «мыши» запустить программу, которая активизирует этот файл.

4. Дальнейшие построения модели ТАД программа Lua выполняет автоматически. При желании в программу могут быть вставлены команды Pause(), которые обеспечат останов программы в каждом конкретном месте. После просмотра текущего изображения на экране компьютера клавишей Enter можно продолжить выполнение программы.

5. Вставленные в желательных местах программы команды mi_saveas(name) позволяют сохранить очередной файл с текущим – промежуточным результатом построения модели ТАД под произвольным именем name, например, name="geom_mod_01.fem".

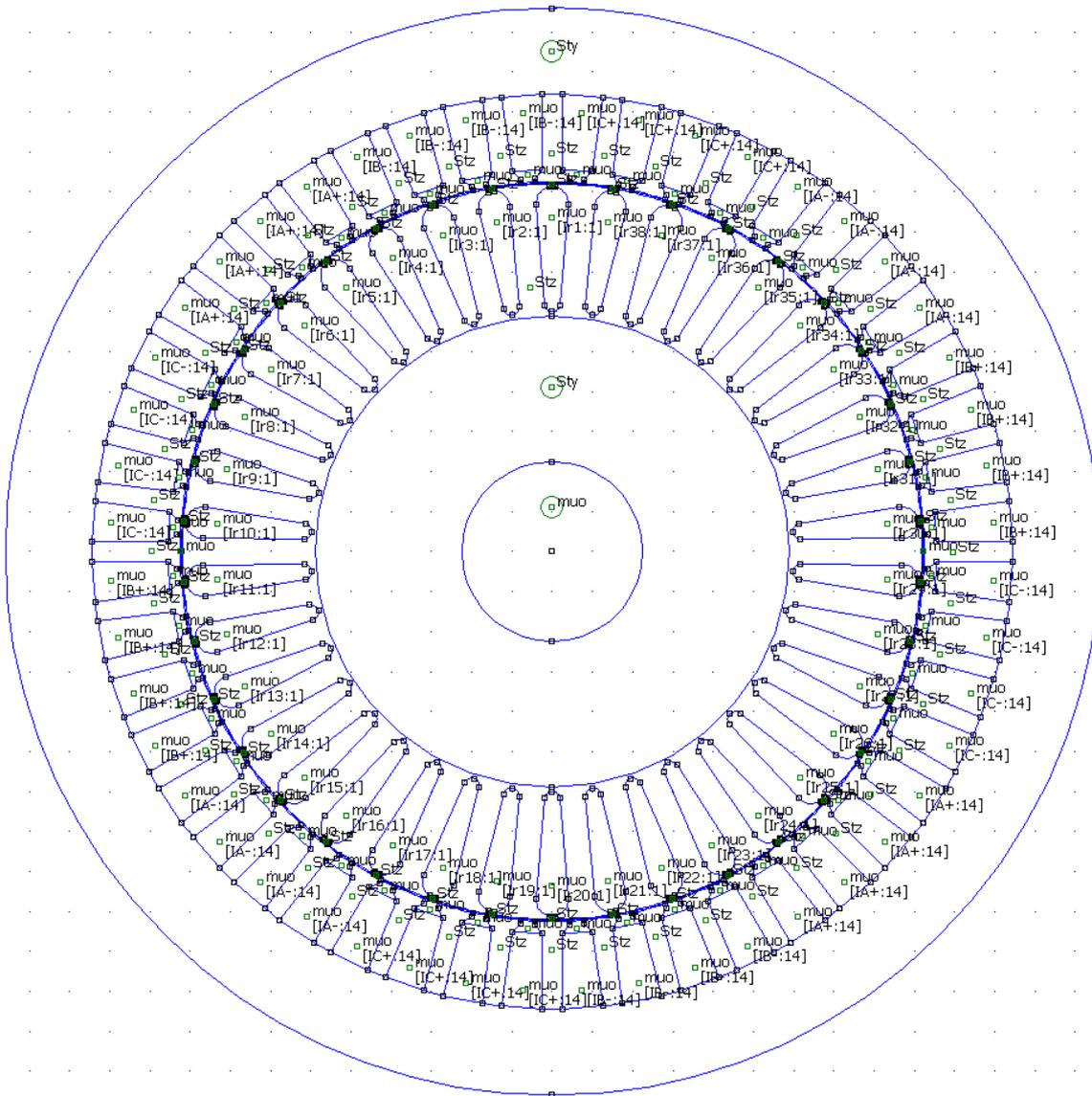


Рис. 11 – Графическое представление созданной физико-геометрической модели ТАД

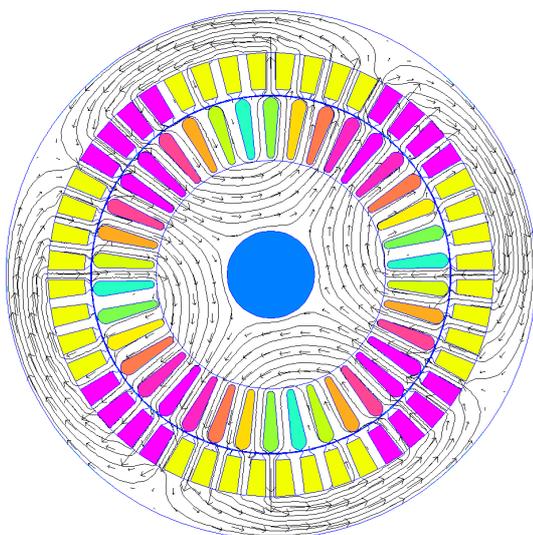


Рис. 12 – Графическое представление результатов работы программы FEMM

Именно таким способом сохранены иллюстрации, представленные на рис. 8 – 10 и сопровождающие приведенный текст программы – скрипта Lua.

Вывод.

Представленный скрипт Lua дает широкие возможности пользователям программы FEMM в создании расчетных моделей для обеспечения расчетов магнитных полей и электромагнитных параметров ТАД. Этот скрипт универсален с точки зрения геометрических размеров и структуры электромагнитной системы ТАД в рамках их распространенной типовой конструкции. На базе рассмотренных принципов формирования расчетных моделей могут быть разработаны аналогичные скрипты для других модификаций конструкции ТАД и других типов ЭМ.

Список литературы

- 1 Finite Element Method Magnetics : OldVersions. FEMM 4.2 11Oct2010 Self-Installing Executable. – Режим доступа: <http://www.femm.info/wiki/OldVersions>.
- 2 Милых В. И. Автоматизированное формирование расчетных моделей турбогенераторов для программной среды FEMM / В. И. Милых, Н. В. Полякова // Электротехника і електромеханіка.

- 2015. – №4. – С. 7-14. doi: 10.20998/2074-272X.2015.4.02.
3. Милых В.И. Автоматизированные расчеты в программной среде FEMM динамики электромагнитных процессов турбогенераторов / В. И. Милых, Н. В. Полякова // Электротехника і електромеханіка. – 2015. – № 6. – С.14-20. doi: 10.20998/2074-272X.2015.6.04.
 4. Милых В.И., Полякова Н.В. Определение электромагнитных параметров и фазовых соотношений в турбогенераторах автоматизированным расчетом магнитного поля в программной среде FEMM // Электротехника і електромеханіка. – 2016. – № 1. – С.20-26. doi: 10.20998/2074-272X.2016.1.05.
 5. Вольдек А. И. Электрические машины. Машины переменного тока / А. И. Вольдек, В. В. Попов. – СПб.: Питер, 2010. – 356 с.
 6. Копылов И. П. Проектирование электрических машин / И. П. Копылов, Ф. А. Горяинов, Б. К. Клоков и др. Под ред. И. П. Копылова. – М.: Издательство Юрайт, 2011. – 767 с.
 7. Милых В. И. Численно-полевой поверочный анализ проектных параметров трехфазных асинхронных двигателей / В. И. Милых, Л. В. Шилкова // Электромеханічні і енергозберігаючі системи. Випуск 1/2016 (33). – С. 58-65.
 8. Милых В.И. Численно-полевой анализ магнитного поля трехфазного асинхронного двигателя в статике и динамике / В. И. Милых, Л. В. Шилкова // Вісник НТУ «ХПІ». Серія: «Електричні машини та електромеханічне перетворення енергії». – Х. : НТУ «ХПІ», 2016. – № 11 (1183). – С. 80–87.
- References (transliterated)**
1. "Finite Element Method Magnetics : OldVersions. FEMM 4.2 11Oct2010 Self-Installing Executable", available at: <http://www.femm.info/wiki/OldVersions> (In English).
 2. Milykh V.I., Polyakova N.V. Avtomatizirovannoe formirovanie raschetnykh modeley turbogeneratorov dlya programmnoy sredy FEMM [Automated formation of calculation models of turbogenerator for software environment FEMM]. *Electrotekhnika i Electromekhanika – Electrical engineering & electromechanics*. 2015, no. 4, pp. 7-14. doi: 10.20998/2074-272X.2015.4.02.
 3. Milykh V.I., Polyakova N.V. Avtomatizirovannyye rascheti v programmnoy srede FEMM dinamiki elektromagnitnykh protsessov turbogeneratorov [Automated calculations of the dynamics of a turbogenerator electromagnetic processes in the software environment FEMM]. *Electrotekhnika i Electromekhanika – Electrical engineering & electromechanics*. 2015, no. 6, pp. 16-20. doi: 10.20998/2074-272X.2015.6.04.
 4. Milykh V. I., Polyakova N. V. Opredelenie elektromagnitnykh parametrov i fazovykh sootnosheniy v turbogeneratorah avtomatizirovannyim raschetom magnitnogo polya v programmnoy srede FEMM [Determination of electromagnetic parameters and phase relations in turbo-generators by the automated calculation of the magnetic field in the software environment FEMM]. *Electrotekhnika i Electromekhanika – Electrical engineering & electromechanics*. 2016, no. 1, pp. 20–26. doi: 10.20998/2074-272X.2016.1.05.
 5. Voldek A. I., Popov V. V. *Elektricheskie mashiny. Mashiny peremennogo toka* [Electrical Machines. Machines of alternating current], SPb, Piter Publ, 2010, 356 p.
 6. Kopylov I. P., Goryainov F. A., Klokov B. K. *Proektirovanie elektricheskikh mashin* [Design of electrical machines]. Moscow, Yurait Publ, 2011, 767 p.
 7. Milykh V. I., Shilkova L. V. Chislennno-polevoy poverochnyy analiz proektnykh parametrov trehfaznykh asinhronnykh dvigateley [Numerically-field confirmatory analysis of a three-phase induction motor design parameter]. *Elektromekhanichni i energozberigayuchi sistemi – Electromechanical and energy saving systems*. Release 1/2016 (33), pp. 58-65.
 8. Milykh V.I., Shilkova L.V. Chislennno-polevoy analiz magnitnogo polya trehfaznogo asinhronnogo dvigatelya v statike i dinamike [Numerical-field analysis of the magnetic field of a three-phase induction motor in statics and dynamics]. *Bulletin of NTU KhPI. Series: Electrical Machines and Electromechanical Energy Conversion*, 2016, no. 11 (1183), pp. 80–87.

Поступила (received) 08.03.2017

Бібліографічні описи / Библиографические описания / Bibliographic descriptions

Автоматизоване формування розрахункових моделей трифазних асинхронних двигунів для програмного середовища FEMM / В. І. Мілих // Вісник НТУ «ХПІ». Серія: "Електричні машини та електромеханічне перетворення енергії". – Х. : НТУ «ХПІ», 2017. – № 1 (1223). – С. 03–15. – Бібліогр.: 8 назв. – ISSN 2409-9295.

Автоматизированное формирование расчетных моделей трехфазных асинхронных двигателей для программной среды FEMM / В. И. Милых // Вісник НТУ «ХПІ». Серія: "Електричні машини та електромеханічне перетворення енергії". – Х. : НТУ «ХПІ», 2017. – № 1 (1223). – С. 03–15. – Бібліогр.: 8 назв. – ISSN 2409-9295.

Automated formation of calculation models of a three-phase induction motors for software environment FEMM / V. I. Milykh // Bulletin of NTU "KhPI". Series: "Electric machines and electromechanical energy conversion." – Kharkiv : NTU "KhPI", 2017. – No. 1 (1223). – P. 03–15. – Bibliogr.: 8. – ISSN 2409-9295.

Відомості про авторів / Сведения об авторах / About the Authors

Мілих Володимир Іванович, доктор технічних наук, професор, Національний технічний університет «Харківський політехнічний інститут», завідувач кафедри електричних машин, тел. (057)707-65-14; e-mail: mvikemkpi@gmail.com.

Милых Владимир Иванович, доктор технических наук, профессор, Национальный технический университет «Харьковский политехнический институт», заведующий кафедрой электрических машин, тел. (057)707-65-14; e-mail: mvikemkpi@gmail.com.

Milykh Vladimir Ivanovich, Doctor of Technical Sciences, Full Professor, National Technical University «Kharkiv Polytechnic Institute», Chair of the Department of Electrical machines, tel. +38 (057) 707-65-14; e-mail: mvikemkpi@gmail.com.