

СИСТЕМА АВТОМАТИЗОВАНОГО ФОРМУВАННЯ РОЗРАХУНКОВИХ МОДЕЛЕЙ
ЕЛЕКТРИЧНИХ МАШИН ДЛЯ ПРОГРАМНОГО СЕРЕДОВИЩА FEMM

В.І. Мілих, докт.техн.наук

Національний технічний університет "Харківський політехнічний інститут",

вул. Кирпичова, 2, Харків, 61002, Україна.

E-mail: mvikemkpi@gmail.com

Показано принципи автоматизованого формування геометричної та фізичної моделей електричних машин алгоритмічною мовою Lua для розрахунку їхніх магнітних полів програмою FEMM. Надано організаційні основи алгоритму і програмування, фрагменти файлів текстів програми і вхідних даних. Можливості скриптів Lua продемонстровано на прикладі машин класичних типів. Подальший розвиток таких скриптів у поєднанні з керованою ними програмою FEMM є ефективним інструментом, який при розрахунках електромагнітних і силових параметрів та процесів машин за швидкодією і можливостями покликаний замінити існуючий аналітичний формульний ряд. Бібл. 4, рис. 5.

Ключові слова: електричні машини, магнітне поле, програма FEMM, геометрична і фізична моделі, формування, Lua скрипт.

Вступ. В Україні, зокрема, Харкові, традиційно виробляються електричні машини (ЕМ) класичних типів: синхронні генератори, асинхронні двигуни (АД) і двигуни постійного струму (ДПС). Для забезпечення попиту на них потрібне постійне їхнє удосконалення. В умовах жорсткої конкуренції на ринку ЕМ і відповідно ліміту часу актуальною є динамічна система проектування, яка дає змогу створювати нові зразки ЕМ швидко і практично без дослідного їх доведення.

Класичні методи проектування, засновані, серед іншого, на теорії магнітних кіл, не дають можливості повною мірою впоратися з обумовленою проблемою. Ефективним її рішенням може бути використання чисельних методів розрахунку магнітних полів (МП). Вони дозволяють враховувати досить точно реальну геометрію пристроїв і нелінійні магнітні властивості феромагнітних осердь. Для таких цілей поширення набула програма FEMM (Finite Element Method Magnetics) [1], заснована на методі скінченних елементів і орієнтована на розрахунок двомірних МП. Цього вистачає для ЕМ класичної конструкції [2] на їхній активній довжині.

При експлуатації програми FEMM найбільший час витрачається на формування *геометричних моделей*, що відображають конструкцію ЕМ, і фізичних моделей, що відображають магнітні властивості матеріалів і струмові параметри обмоток, а також на отримання результатів розрахунків і перетворення їх у форму, придатну для подальшого використання в процесі проектування і дослідження ЕМ.

Проблеми витрат часу розраховувачів стають ще серйознішими при багатоваріантних розрахунках ЕМ, у випадках варіювання їхньою геометрією, розмірами, струмами, положенням рухомих частин.

Обумовлені проблеми можна здолати автоматизацією побудови фізико-геометричних моделей ЕМ, розрахунків і перебору параметрів, а також отримання, обробки і переведення результатів у необхідну форму. Такі можливості користувачам програми FEMM надає інтегрована до неї алгоритмічна мова Lua. І завдання тепер полягає в написанні відповідного комплексу програм (скриптів). Це завдання є дуже складним, оскільки вимагає володіння знаннями в сферах ЕМ, програмування і теорії МП. Проте вирішення цієї задачі є виправданим, оскільки програма пишеться один раз, а використовується безліч разів і будь-якою кількістю користувачів.

Для забезпечення високої ефективності розрахунків МП і електромагнітних параметрів у середовищі FEMM автором розроблено комплекс скриптів Lua для різних типів ЕМ. Ці програми дають змогу багаторазово підвищити продуктивність розраховувачів і при цьому ще забезпечити можливість отримання широкого спектра параметрів і характеристик, у тому числі таких, які практично недоступні при класичних методах проектування, або навіть у "ручному" режимі роботи з програмою FEMM. Вимогами до таких скриптів є універсальність з точки зору можливого варіювання геометрією і розмірами проєктованих ЕМ – при мінімумі інформації, що вводиться, – і тільки в числовій формі.

Метою цієї роботи є представлення принципів складання скриптів Lua для автоматизованого формування фізико-геометричних моделей ЕМ на прикладі згаданих вище їхніх типів: турбогенератора (ТГ), АД і ДПС із загальновідомими класичними конструкціями [2].

Фізико-геометричні моделі ЕМ. Початковим кроком створення скрипту Lua є підготовка строго детермінованих таких моделей ЕМ. Ці моделі для АД, ТГ і ДПС показано на рис. 1 – 3. ТАД і ДПТ тут чотириполюсні, ТГ – двополюсний, але скрипти реалізують і інші варіанти кількості пар полюсів, також як і кількості пазів статора та ротора. При побудовах у скриптах Lua використовується прямокутна система координат x, y , а як допоміжні – полярні координати r, α . Відлік α і нумерація полюсів і пазів ведуться від верхньої півосі y , обертання ротора відповідає напрямку α .

Перед написанням програми для побудови моделей треба визначитися з базовими позиціями елементів

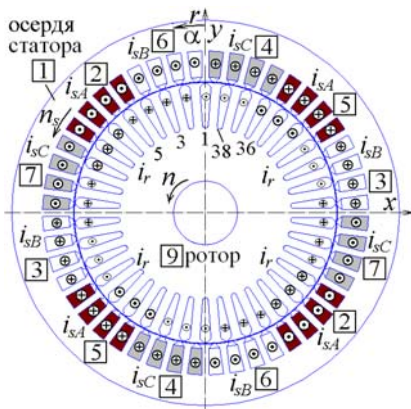


Рис. 1

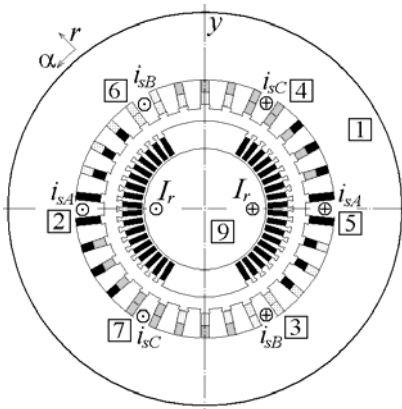


Рис. 2

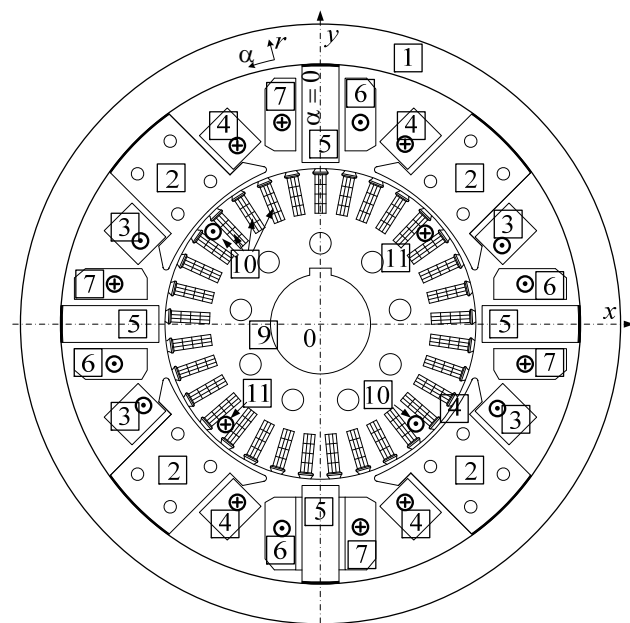


Рис. 3

конструкцій, які потім можуть модифікуватися залежно від розрахункового режиму МП. Так, для АД і ТГ фазні зони обмотки статора $A-A'$, відмічені фазними струмами i_{sA} з різними знаками, розташовуються так, що її МРС спрямована за віссю y . У ДПС за віссю y діє МРС верхнього додаткового полюса. Тоді інші елементи конструкції ЕМ займають зумовлені позиції так само, як і умовні напрями струмів в обмотках, які вказано на рис. 1 – 3.

Загальна характеристика скрипту Lua.

Lua – це швидка і зручна скриптова мова програмування, яка створювалася як мова для баз даних. Простий синтаксис і легкість вбудовування Lua в інші програми, приміром, FEMM, забезпечують Lua все більш широке застосування. Мова Lua підтримана компілятором з вихідних термінів у програмний код і віртуальною машиною для виконання генерованого коду. Lua має динамічне визначення типів даних. Змінні мови можуть містити значення будь-якого типу, використовуватися як аргументи при виклику функцій.

Тексти файлів скриптів і даних пишуться в редакторі *Блокнот*. У рядках цих файлів усе після подвійного дефіса (--) сприймається як коментарі. Скрипти "запускаються" з робочого вікна FEMM "кнопкою" *Open Lua Script* в меню *File*.

У скриптах базовою є команда задання параметрів нової *Задачі*: її тип (МП або інше), одиниці виміру розмірів, аксіальна довжина ЕМ, тип координат, точність і т.п.

Побудова графічних моделей у скриптах Lua заснована на командах розставляння *вузлів* – координатних точок (x, y) . На них своїми командами будуються прямі *відрізки* – *сегменти* і *дуги* кіл. У замкнених зонах – *блоках* ставляться *мітки*, через які їм при-

писуються конкретні фізичні властивості. Вони задаються окремими функціями, в яких вказуються усі необхідні параметри матеріалів, струмів, скінчених елементів та ін. Є також функції для вибору і розставляння *граничних умов*, приміром, умови *Діріхле* для векторного магнітного потенціалу на зовнішніх поверхнях області розрахунку.

Перераховані вище об'єкти можна об'єднувати в нумеровані *групи* і оперувати з усіма об'єктами групи відразу. На рис. 1 – 3 у квадратиках показані прийняті для цих моделей ЕМ номери груп. Lua, як і інші мови програмування, дозволяє створювати власні функції, оперувати з циклами, умовними операторами, робити числення за формулами, стандартними функціями: *sqrt* – корінь квадратний; *sin*, *cos*, *tan*, *atan* – синус, косинус, тангенс, арктангенс та ін. Для тригонометричних функцій куту задаються в радіанах, при побудові геометричних об'єктів Lua оперує градусами. Вбудовані в FEMM функції скрипту Lua починаються з префікса *mi_*.

Наведемо необхідний мінімум команд, яке часто зустрічаються при формуванні фізико-геометричних моделей ЕМ, де позначення (x, y) , $(x1, y1)$ і $(x2, y2)$ – це координати вузлів:

- 1) додавання: *mi_addnode*(x, y) – вузол; *mi_addsegment*($x1, y1, x2, y2$) – відрізок між двома вузлами; *mi_addarc*($x1, y1, x2, y2, angle, maxseg$) – дуга між двома вузлами з опорним кутом '*angle*', що складається з відрізків куткових сегментів розміром '*maxseg*'; *mi_addblocklabel*(x, y) – мітка блока;
- 2) вибір найближчих до точки (x, y) об'єктів: *mi_selectnode*(x, y) – вузол; *mi_selectsegment*(x, y) – відрізок; *mi_selectarcsegment*(x, y) – дуга; *mi_selectlabel*(x, y) – мітка блоку; *mi_selectgroup*(n) – n -на група;
- 3) копіювання: *mi_mirror*($x1, y1, x2, y2, edit$) – вибраних об'єктів відносно лінії, що проходить через вузли $(x1, y1)$ і $(x2, y2)$, де *edit* – дія з: 0 – вузлами, 1 – відрізками, 2 – дугами, 3 – мітками, 4 – групами; *mi_copyrotate*($bx, by, angle, copies, edit$), де *copies* – стільки разів з поворотом на кут *angle* відносно базової точки з координатами (bx, by) ; *edit* – див. вище;

- 4) очищення виділення усіх раніше вибраних об'єктів: `mi_clearselected()`;
- 5) `mi_addcircprop("circuitname", i, circuittype)` – додає нову схемну властивість з ім'ям "circuitname" та заданим струмом `i`: `circuittype` – тип з'єднання провідників у даному блоці: 0 – паралельне, 1 – послідовне;
- 6) установка властивості матеріалу конкретної підобласті – блока:
`mi_setblockprop("blockname", automesh, meshsize, "incircuit", magdir, group, turns)`, де "blockname" – ім'я мітки для магнітних властивостей; `automesh` – спосіб триангуляції скінченних елементів; `meshsize` – максимальний розмір, що обмежує їх, "incircuit" – символ струму, `magdir` – напрям намагнічування, `group` – номер групи, `turns` – кількість ефективних провідників із струмом;
- 7) оператор `read(f_d, "*n", "*l")` зчитує з чергового рядка файла вхідних даних з ім'ям `f_d` значення одного числового або строкового даного, а решта тексту, що залишився в рядку, – це вільний коментар;
- 8) оператор `write` служить для запису строкових і форматуваних числових даних в архівний файл, наприклад `write(format(" Iг= %6.1f", Iг), format(" Qг= %3.0f", Qг), "\r\n")`, де "\r\n" робить перехід на новий рядок.

Більш повний перелік команд і функцій Lua знаходиться в розділі *Help* програми FEMM.

Відображення конструкції ЕМ. Для фізико-геометричної моделі конкретної ЕМ (рис. 1 – 3) спочатку необхідно підготувати її формалізовану конструкцію з літерною ідентифікацією усіх розмірів машини як у цілому (приміром, радіуси осердь), так і її елементів (поліусів, пазів, зубців, обмоток та ін.).

Як приклад, розглядатимемо формування зубцево-пазової структури ротора ТГ у вигляді "вирізок" з підготовлених повної конструкції, файла вхідних даних та скрипту Lua.

На рис. 4 для зубцево-пазової структури ротора прийняті позначення – ідентифікатори відповідно до правил імен в Lua: `hr`, `br`, `hsr`, `hkr`, `bkr`, `ar1`, `ar2`. За цим має йти файл вхідних даних для конкретного варіанта ЕМ, де кожному ідентифікатору повинне відповідати конкретне числове значення розмірів та інших величин (струми, кількості пар полюсів і паралельних віток обмоток, кількості послідовних витків, пазів, аксіальні довжини осердь і коефіцієнт заповнення їх сталлю тощо). Має бути повний перелік інформації, що зумовлює магнітне поле даної машини в конкретному режимі збудження, у тому числі і криві намагнічування сталей.

Фрагмент реального файла вхідних даних з деяким довільним ім'ям "TG_340" у вигляді однієї колонки скомпоновано в такі дві колонки (розміри в мм):

10 --Nг-кількість провідників в пазу ротора	12.8	-- hsr - висота шліца паза ротора
2244 -- Iг - значення струму збудження, А	36.7	-- hkr - висота клина паза ротора
36 -- Qг - кількість пазів ротора обмотаних	46	-- bkr - ширина клина паза ротора
52 -- Qгу - умовна кількість пазів ротора	45	-- ar1 - кут скосу клина паза ротора, градус
560 -- rге - радіус ротора зовнішній	20	-- ar2 - кут скосу клина паза ротора, градус
160 -- hr - висота паза ротора	"St3"	-- St_rot - ім'я стали для осердя ротора
33.9 -- br - ширина паза ротора	8	-- fe5 -- макс. розмір трикутників у пазах і зубцях ротора

Зі значеннями величин тут наведено назви згаданих позначень з рис. 4 разом з усіма іншими необхідними величинами для формування фізико-геометричної моделі паза.

Конструкція ЕМ та образ її фізико-геометричної моделі з системою позначень дають змогу продумати алгоритм формування власне фізико-геометричної моделі, і вже на цій основі створювати скрипт Lua, що реалізує цей алгоритм. Повні скрипти, що забезпечують конкретний тип ЕМ, мають "солідний" лістинг у сотні рядків. Але робота цілком себе виправдовує, зважаючи на наступне множинне використання скриптів.

Структура програми побудови геометричної і фізичної моделей ЕМ. Для ЕМ, що розглядаються (рис. 1 – 3), та інших їхніх типів структура програми – скрипту Lua в укрупненому вигляді є досить універсальною і містить наступні логічні частини:

- завдання імені зовнішнього файла вхідних даних, його відкриття і читання з нього значень даних;
- організація файла результатів розрахунку і призначення імені фізико-геометричної моделі;
- розрахунок додаткових параметрів на основі вхідних даних;
- створення файла для магнітної *Задачі* та завдання її параметрів;
- створення центральної точки моделі та поверхонь внутрішнього ротора і зовнішнього статора;
- створення геометрії зубцево-пазової або полюсної структури статора і ротора;
- створення матеріальних властивостей осердь і немагнітних ділянок та відповідних їхніх імен;

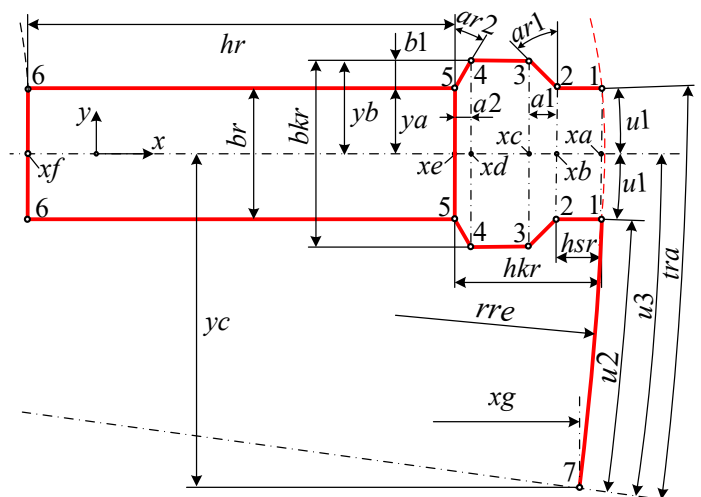


Рис. 4

- введення кривих намагнічування сталей у бібліотеку властивостей матеріалів програми FEMM;
- задання властивостей немагнітних ділянок і феромагнітних елементів осердь ротора і статора;
- задання магнітних властивостей матеріалу і струмів у перерізах боків обмоток;
- створення і призначення граничних умов Діріхле на зовнішній поверхні осердя статора;
- виконання тестового розрахунку МП і перехід до показу та витягання результатів розрахунку.

Приклад складання скрипту Lua. Для зручності програмування в кожному елементі конструкції задається система опорних точок – вузлів. На рис. 4 таких точок сім із відповідними цифровими номерами (базові вгорі і дзеркальні відносно осі x точки мають однакові номери). Тоді від розмірних позначень слід перейти до координатних позначень $x_a - x_g, y_a - y_c$, що забезпечують ідентифікацію усіх вузлів. Їхній зв'язок з розмірами конструкції встановлюється на основі звичайних тригонометричних і алгебраїчних виразів та представляється безпосередньо далі у фрагменті скрипту.

Фрагмент скрипту, що забезпечує побудову зубцево-пазової структури ротора.

```
gr=Pi/180 -- множник переведення кутових градусів у радіани
f_d = openfile("TG_340" .. ".txt","r") -- відкриття файлу даних з ім'ям "TG_340" на їх читання
-- Читання даних з файлу f_d з ім'ям "TG_340" у тому ж порядку, як вони розташовані в цьому файлі
Ncr=read(f_d, "*n", "*1") Ir=read(f_d, "*n", "*1") Qr=read(f_d, "*n", "*1") Qru=read(f_d, "*n", "*1")
rre=read(f_d, "*n", "*1") hr=read(f_d, "*n", "*1") br =read(f_d, "*n", "*1") hsr=read(f_d, "*n", "*1")
hkr=read(f_d, "*n", "*1") bkr=read(f_d, "*n", "*1") ar1=read(f_d, "*n", "*1") ar2=read(f_d, "*n", "*1")
ar1=ar1*gr ar2=ar2*gr St_rot=read(f_d, "*n", "*1") fe5=read(f_d, "*n", "*1")
-- Обчислення додаткових даних за введеними даними
tra=360/Qru -- пазовий крок ротора в градусах
-- Координати для системи точок-вузлів, що зображені на рис. 4
ya=br/2 yb=bkr/2 xa=sqrt(rre^2-ya^2) xb=xa-hsr xe=xa-hkr xf=xa-hr bl=yb-ya a1=b1*tan(ar1) xc=xb-a1
a2=b1*tan(ar2) xd=xe+a2 u1=atan(ya/xa) u2=tra*gr-2*u1 u3=u1+u2 xg=rre*cos(u3) yc=rre*sin(u3)
-- Розставляння вузлів від 1 до 7-го
mi_addnode(xa,ya) mi_addnode(xb,ya) mi_addnode(xc,yb) mi_addnode(xd,yb)
mi_addnode(xe,ya) mi_addnode(xf,ya) mi_addnode(xg,-yc)
-- Проведення (створення) відрізків ліній (сегментів) між вузлами 1-2-3-4-5-6
mi_addsegment(xa,ya,xb,ya) mi_addsegment(xb,ya,xc,yb) mi_addsegment(xc,yb,xd,yb)
mi_addsegment(xd,yb,xe,ya) mi_addsegment(xe,ya,xf,ya)
-- Виділення (розмітка) вузлів від 1 до 6-го
mi_selectnode(xa,ya) mi_selectnode(xb,ya) mi_selectnode(xc,yb)
mi_selectnode(xd,yb) mi_selectnode(xe,ya) mi_selectnode(xf,ya)
-- Виділення (розмітка) сегментів між вузлами 1-2-3-4-5-6
x=(xa+xb)/2 mi_selectsegment(x,ya) x=(xb+xc)/2 y=(ya+yb)/2 mi_selectsegment(x,y)
x=(xc+xd)/2 mi_selectsegment(x,yb) x=(xd+xe)/2 mi_selectsegment(x,y) x=(xe+xf)/2 mi_selectsegment(x,ya)
-- Призначення виділеним вузлам і сегментам групи 9 і зняття виділення
mi_setnodeprop(" ",9) mi_setsegmentprop(" ", " ", 1,0,9) mi_clearselected()
-- Виділення верхньої стінки паза 1-2-3-4-5-6 і відображення її вниз симетрично відносно осі x
mi_selectgroup(9) mi_mirror(0,0,rre,0,4) mi_clearselected()
-- Проведення і розмітка сегментів 5-5 і 6-6 і призначення ним групи 9
mi_addsegment(xe,ya,xe,-ya) mi_addsegment(xf,ya,xf,-ya)
mi_selectsegment(xe,0) mi_selectsegment(xf,0) mi_setsegmentprop(" ", " ", 1, 0, 9) mi_clearselected()
-- Виділення (розмітка) вузла 7 і призначення йому групи, зняття виділення
mi_selectnode(xg,-yc) mi_setnodeprop(" ",9) mi_clearselected()
-- Проведення дуги коронки зубця між вузлами 1 (нижній) і 7, виділення її і призначення групи 9
maxseg=1 mi_addarc(xg,-yc,xa,-ya,u2/gr,maxseg) --maxseg – опорний кут сегментів дуги в 1 градус
x=(xa+xg)/2 y=-(ya+yc)/2 mi_selectarcsegment(x,y) mi_setarcsegmentprop(1," ",0,9) mi_clearselected()
mi_addmaterial("mu0",1,1) -- створення властивості матеріалів немагнітних блоків
--Зв'язок імені струму "Ir-" у правій частині ротора зі значенням Ir, додавання і виділення мітки властивостей
mi_addcircprop("Ir-",-Ir, 1) x=rre-0.5*hr y=0 mi_addblocklabel(x,y) mi_selectlabel(x,y)
-- Завдання властивостей матеріалу і струму стрижня обмотки збудження в пазу правої частини ротора
mi_setblockprop("mu0", 0, fe5, "Ir-", 0, 9, Ncr) -- мітка властивостей відразу віднесена до групи 9
-- Поворот паза в праву нижню позицію, його копіювання і отримання структури правої половини ротора
mi_selectgroup(9) mi_moverotate(0,0,-(Qr/4-0.5)*tra,4) mi_selectgroup(9) mi_copyrotate(0,0,tra,(Qr/2-1),4)
-- Проставляння мітки властивостей і завдання матеріалу для зубців осердя ротора, група 9
x=rre-hr/2 y=0 mi_addblocklabel(x, y) mi_selectlabel(x,y) mi_setblockprop(St_rot,0,fe5," ",0,9,0)
```

Фрагмент побудованої зубцево-пазової структури ротора і статора, показано на рис. 5.

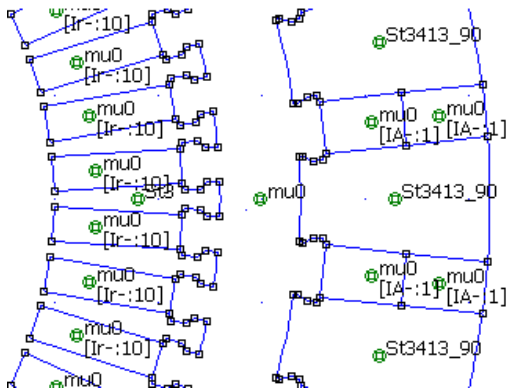


Рис. 5

Так само на основі представленого скрипту Lua можна програмним шляхом будувати й інші елементи конструкцій ЕМ практично будь-якого рівня складності, про що свідчать, зокрема, зображені на рис. 1 – 3 моделі їхніх класичних типів. Вони побудовані за допомогою авторських скриптів Lua. А продовженням були дослідження їхніх МП, електромагнітних параметрів і процесів, як у статиці, так і в динаміці, представлені, наприклад, у [3, 4] і інших роботах. При цьому сама організація розрахунків, а також отримання різноманітних результатів також автоматизовані за допомогою скриптів Lua.

Досвід роботи автора показує, що на підготовку файла вхідних даних (за його шаблоном) для конкретної ЕМ потрібні лічені хвилини. А власне процес побудови фізико-геометричної моделі на комп'ютері середнього рівня обчислюється у секундах.

Висновок. Представлені інформація про мову Lua і принципи використання існуючих функцій та написання власних скриптів є первинною основою для створення програм автоматичного формування фізико-геометричних моделей ЕМ. Це дає широкі можливості користувачам програми FEMM щодо створення розрахункових моделей для забезпечення розрахунків МП різноманітних типів ЕМ. Створювані скрипти мають бути універсальними з точки зору геометричних розмірів і структури електромагнітної системи конкретного типу ЕМ у рамках їх базової конструкції. Подальший розвиток цих скриптів у поєднанні з керованою ними програмою FEMM дає ефективний інструмент, який при розрахунках і аналізі параметрів і процесів ЕМ за швидкодією та можливостями покликаний замінити існуючий аналітичний формульний ряд, що створювався і використовувався в попередні часи.

1. Voldek A.I., Popov V.V. Electrical Machines. Machines of alternating current. SPb: Piter Publ, 2010. 356 p. (Rus.).
2. Milykh V.I., Polyakova N.V. Automated calculations of the dynamics of a turbo-generator electromagnetic processes in the software environment FEMM. *Electrical Engineering & Electromechanics*. 2015, No 6. P. 16-20. DOI: <https://doi.org/10.20998/2074-272X.2015.6.04> (Rus.).
3. Milykh V.I. Numerically-field analysis of the adequacy of the design data of three-phase induction motors and the method of their refinement on this basis. *Tekhnichna Elektrodynamika*. 2018. No 1. P. 47-55. (Rus.) DOI: <https://doi.org/10.15407/techned2018.01.047> (Rus.).
4. Finite Element Method Magnetics: OldVersions. FEMM 4.2 11Oct2010 Self-Installing Executable, available at: <http://www.femm.info/wiki/OldVersions> .

УДК 621.313:681.3

СИСТЕМА АВТОМАТИЗИРОВАННОГО ФОРМИРОВАНИЯ РАСЧЕТНЫХ МОДЕЛЕЙ ЭЛЕКТРИЧЕСКИХ МАШИН ДЛЯ ПРОГРАММНОЙ СРЕДЫ FEMM

В.И. Милых, докт. техн. наук

**Национальный технический университет "Харьковский политехнический институт",
ул. Кирпичева, 2, Харьков, 61002, Украина. e-mail: mvikemkpi@gmail.com**

Показаны принципы автоматизированного формирования физико-геометрических моделей электрических машин (ЭМ) на алгоритмическом языке Lua для расчета их магнитных полей программой FEMM. Даны основы организации алгоритма и скрипта Lua, фрагменты текстовых файлов программы и исходных данных. Возможности скриптов Lua демонстрируются на примере ЭМ трех типов. Развитие таких скриптов в сочетании с управляемой ими программой FEMM дает эффективный инструмент, который при расчетах электромагнитных и силовых параметров и процессов ЭМ по быстродействию и возможностям призван заменить существующий аналитический формульный ряд. Библи. 4, рис. 5.

Ключевые слова: электрические машины, магнитное поле, программа FEMM, физико-геометрические модели, формирование, Lua скрипт.

THE SYSTEM OF AUTOMATED FORMATION OF ELECTRICAL MACHINES COMPUTATIONAL MODELS FOR THE FEMM SOFTWARE ENVIRONMENT

V.I. Milykh

National technical university "Kharkov polytechnic institute",

2, Kyrpychova str., Kharkov, 61002, Ukraine.

e-mail: mvikemkpi@gmail.com

The principles of automated formation of geometric and physical models of electric machines (EM) are shown for calculating their magnetic fields by the FEMM program in the algorithmic Lua language. The basis for the organization of the algorithm and script Lua, fragments of the program text files and the source data are given. The capabilities of Lua scripts are demonstrated on the example of EM of three types. The development of such scripts in combination with the FEMM program controlled by them provides an effective tool that in the calculation of electromagnetic and power parameters and EM processes in terms of speed and capabilities is intended to replace the existing analytical formula series. References 4, figures 5.

Key words: electric machines, magnetic field, FEMM program, graphical and physical models, formation, Lua script.

Надійшла 05.03.2018

Остаточний варіант 21.03.2018