

**DEVELOPMENT OF A COMPUTER-AIDED LEARNING
TOOL TO OPTIMIZE STUDENTS' LEARNING OF
UNDERGRADUATE ELECTROMAGNETICS AT
OHIO NORTHERN UNIVERSITY**

Khalid S. Al-Olimat, Ph.D.

**Electrical and Computer Engineering Department
Ohio Northern University**

Email: k-al-olimat@onu.edu

Students have generally found electromagnetics a difficult subject to understand and learn. Despite the publication of many textbooks in this field, each one is intended to provide an improvement over previous textbooks. Students continue to remain perplexed as a result of the numerous conditions that must often be remembered and correlated in solving a problem. Various possible difficulties experienced by students such as mathematical difficulty and unclear relevance explanation of principles and concepts. Many publications focused on the second possibility as an approach to overcome the lack of students' interest in this subject. This paper is intended to address the first concern that is mathematical difficulty through the use of a computer-aided learning tool. The aim of this paper is to present the developed computer-aided learning tool that is designed by a team consists of three students at Ohio Northern University as a small project and to show how such tool helps students in the learning of undergraduate electromagnetics course.

Introduction

With the rapid change of technology and its attendant introduction of new material into the curriculum, most colleges and universities face an ever-increasing demand to optimize their curriculum and increase the content of courses. This places large demands on educators to determine more efficient ways of presenting material in order to ensure that students continue to have the same basic understanding of fundamental concepts and skills as in the past. The primary goal of this paper is to address that concern. The development of a computer-aided learning tool, with the use of MatLab, will create a significant software to aid the student's learning of introductory electromagnetism that include vector analysis and transmission lines. When the student opens the MatLab command window and accesses the software, a hyper-linked table of contents will provide guidance to the appropriate MatLab screen. Depending upon the topic, the screen may present symbolic or numerical computation. In all screens, the student can easily make changes to the problem specifications and quickly see the effects on the particular observations.

All screens contain the symbols that most electromagnetics textbooks use which give this software the ability to accompany any electromagnetic textbook. Topics for further exploration can also be found in the user's manual that has been written for this software. It was found that the software's ability to compute and visualize basic electromagnetic related quantities to be a powerful pedagogical aid. It increases students' understanding of fundamental concepts, allows them to appreciate the relevancy of the material, and facilitates a mastery of the mathematical concepts required in the study of electromagnetics.

Purpose of Software

The purpose of the developed tool is to be used in undergraduate electromagnetics course for vector analysis and transmission line analysis. The software design has the following characteristics:

- The design is simple.
- The design is ascetic.
- The design is capable of expansion.
- The program code is well documented and efficient.
- The program is in the form of a graphical user interface (GUI).

The content of the software focuses on two areas of Electromagnetics. These are vector analysis and transmission line analysis. The software is designed so that further sections can be added easily by future work. For the area of vector analysis the designed software performs the following operations:

- Gradient
- Divergence
- Curl
- Conversion between common coordinate systems
- Evaluation of input/output at a given point

In the area of transmission lines the designed software calculates the following:

- Transmission line parameters
- Propagation constant
- Phase Velocity
- Characteristic impedance
- Properties of standing wave on a lossless transmission line

These are the basic functions that the student needs for aid in electromagnetic study. In the future, other areas of study as well as additional functions pertaining to the current areas can be added due to the flexibility of the software.

One of the primary requirements for this design involves creating a Graphical User Interface (GUI). This is an important requirement because it allows the user the most flexibility when

using the software. Most users are accustomed to GUIs because of their current popularity in the Windows environment. This gives the software a standard format.

Design Alternatives

Two solutions were considered in developing this electromagnetic software. The first alternative was to use the programming language C++ and to use Visual Basic to create the GUI. These are two popular programming languages that are widely used by software manufacturers. The most significant advantage of using C++ and Visual Basic would be that the program could be designed as a Windows application; however the team lacked considerable experience in these languages. While a program could be developed as a windows application, the time required to develop the program would significantly increase. The second solution was to use Matlab. Matlab is a mathematical software package that is currently popular in industry and educational settings. It also contains its own high level language that is used for programming solutions. This programming language is similar to C++ but in some ways is at a slightly higher level. Some of the advantages of using Matlab were as follows:

- Advanced mathematical functions already contained in it.
- Students' familiarity with it.
- Symbolic variable library contained in it.
- Development of GUI is relatively simple.

Of the two options, the team was most familiar with Matlab. Matlab is also a standard tool in the engineering college of Ohio Northern University. Electrical and computer engineering students currently use Matlab in various courses such as energy conversion, power systems, control systems, digital signal processing and filter design. An application could be developed in Matlab for the area of electromagnetics that would also be available to the students. This means the loss of program independence but in this case it would not have any significant disadvantages since Matlab is commonly available. Matlab was the chosen method to implement the software design.

Function Coding

A considerable portion of the work involved with this design was spent in writing program codes. Functions were written that would perform the desired electromagnetic calculations. In addition, code was also needed for the GUIs. This included a Matlab generated m-file responsible for the GUI display and function calls that executes for GUI controls and callbacks.

During the design process, it was found that a switchyard style of programming was the most efficient. One function call is capable of producing more than one type of results. The function call performs the action depends on the argument of the function and a switch statement. At first, this type of programming style was not employed. But as the team became more experienced programmers, it proved a valuable tool in reducing the amount of m-files the

program needed. An example of switchyard programming can be found in the function divergence as illustrated by the example below. This function inputs four arguments. The first three are the vector component while the fourth argument is the type of that vector. By inputting the vector type only one function call was needed to perform the divergence operation for three different coordinate spaces.

Example: Switchyard Programming

```
function div = divergence(A,B,C,type)
switch type
case 'cartesian'
    %The Cartesian coordinate system divergence is performed here
    %Computes the first derivative of each unit vector in terms of
    that variable

    A = diff(A,sym('x'));
    B = diff(B,sym('y'));
    C = diff(C,sym('z'));

case 'cylindrical'
    %The Cylindrical coordinate system divergence is performed
    here
    %Computes the first derivative of each unit vector in terms
    of that variable

    r = sym('r'); %Sets up the sym 'r'
    A = diff(r*A,sym('r'));
    B = diff(B,sym('phi'));
    C = diff(C,sym('z'));

case 'spherical'
    %The divergence for the spherical coordinate system is
    performed here
    %Computes the first derivative of each unit vector in terms
    of that variable
    R = sym('R') %Sets up sym 'R'
    theta = sym('theta'); %sets up the sum 'theta'
    A = diff(R*R*A,sym('R'));
    B = diff(sin(theta)*B,sym('theta'));
    C = diff(C,sym('phi'));
end

%The output is then computed from the variables A,B,C
div = A + B + C;
if div ~= 0
    div = simplify(div);
end
```

Another advantage of switchyard programming is function accessibility without using the GUI. The electromagnetic functions were not designed specifically for the GUI. Instead the GUI was designed to provide easy access to the electromagnetic functions. It is also possible that the function calls would be needed in a later expansion of the project. If this happens, then they can be reused without difficulty.

Graphical User Interface Design (GUI)

A graphical user interface is an important requirement for this design. It gives the user an accessible method to use the Matlab functions and provides a professional atmosphere for the software. In order for the software to be compatible with the standards outlined earlier, the user interface needs to be easily comprehended. The user should be able to use the software with minimum help from a manual. In order to accomplish this, two areas of GUI design are carefully considered.

GUI controls are an important feature. These allow the user to enter and receive data. They also allow the user to access functions that perform operations on that data. In order for the GUI to be efficient, it is possible that some controls can be used for more than one purpose. For instance, when creating the user interface for the vector operations, it was desired that the user be able to evaluate the input or output vector at some particular point. Due to space limitations, another GUI was added to perform this evaluation. Instead of adding additional controls in the new GUI and requiring the user to reenter the vector, the new GUI is able to find the vector entered in the previous GUI. It then evaluates that vector at the desired points.

This is one example of design efficiency; others can be cited. It is important that controls be used sparingly in the software. Too many controls in one GUI add to a confusing amount of clutter. With too much clutter, the user loses the desire to use the software and the time required learning the software increase. For these reasons, the controls for the GUIs were used sparingly and when needed.

The GUI layout is also an important part of the design process. Not only must the amount of controls be kept to a minimum, but also the layout of those controls must be carefully considered. If the controls are not arranged in an orderly and neat fashion the GUI loses accessibility and simplicity. Without these two features, the GUI's value is diminished and it is easier for the user to call the functions without the use of the GUI.

In order to keep the layout simple, the software had a title page. This page would allow the user to access the main covered categories of electromagnetics. In the case of this design, this only includes transmission lines and vector analysis (the design was made to be expandable for further additions). This accessibility keeps the layout simple.

One other important component of the GUI is the design of the callbacks. A callback is the action that a control undertakes when that control is activated (i.e. by clicking the mouse pointer on a button). Callback can either be Matlab code, a function call, or an m-file. Often, callbacks

are longer than a few lines of code and thus separate m-files are the best means of initiating a callback.

Troubleshooting

The first problem encountered was the need for symbolic manipulation of variables. Matlab has a symbolic library that allows for such manipulation. This was found to be a considerable advantage for choosing Matlab instead of C++. A search was made for a C++ symbolic library but such a library was unable to be located. Using Matlab was a simple solution to the problem.

Since GUIs were unfamiliar to the design team, a considerable amount of learning was needed. Matlab has a feature called GUIDE. This feature makes designing the GUI easier because it generates the code for GUI layout. However, GUIDE has its limitations and there were many times that certain information had to be accessed from the GUI, and text boxes had to be changed. Much time had to be spent learning how these functions are working.

It was found that when information was entered into the program, it was treated as a STRING variable. This meant that all the information had to be converted to the desired data type. For instance, numerical data had to be converted from a string to a double while symbolic data had to be converted from a string to symbolic variable. Likewise when displaying the output, the data had to be converted back to a string.

Matlab has a function that allows for simplification of symbolic expressions. However, it was found that this function had limitations. The function would not always simplify the expression to its simplest form. This occurred mostly significantly when using the vector conversion functions. It was found that when converting the vector $(x \mathbf{i} + y \mathbf{j} + z \mathbf{k})$ from Cartesian coordinates to spherical coordinates the result would be $(R \mathbf{R} + 0 \theta + 0 \phi)$. This was the correct result but upon going back from spherical to Cartesian a large complicated expression resulted. Upon close examination, it was seen that the answer was correct but was not in its simplest form. In order to compensate, some of the expressions used in the conversion process were pre-simplified before they were programmed. It must be noted that the program does not always simplify expressions to their simplest form. This is especially true when converting from spherical to Cartesian coordinates or from cylindrical to Cartesian coordinates. A significant portion of the problem was, however, removed.

When a GUI window was resized another problem arose. The controls belonging to the figure were not resized with the figure. This caused distortion. There were two possible solutions to this problem. One was to write a function that would resize the figure controls as the figure itself was resized. The other was to write a function that would not allow the figure to be resized.

Originally, it was desired to allow the user to control the size of the GUI window. Such a function was written for the GUI pertaining to vectors. Since resizing was a new concept the function grew as more problems with resizing were found. Once the function was completed, it was found that while the controls became the appropriate size, they did not assume a new

position in the window according to their size. This meant that another function had to be written that would alter the position of each of the controls. Writing the resize and positioning functions required more time than was originally thought. For this reason the rest of the figures used the second method to solve the problem.

The second method was to write a resize function that would not allow the user to change the size. While this sets constrictions upon the user, time limitations within the team required this method to be used. This method proved more simplistic and did not require much time. It was also an effective solution to the problem.

Testing went on throughout the software development. Once a section is finished, it is tested for problems that might occur. As a result, fewer problems are encountered when the sections are brought together for the finished software.

Verification Examples

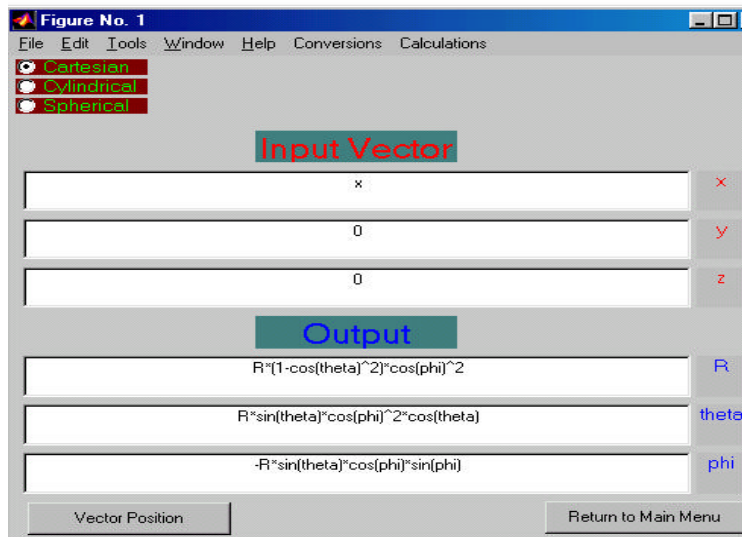
The following examples are presented to show that the software performs effectively according to the design specifications. Firstly, the analytical solution is shown and then it is verified using the software:

Example 1: Convert vector $\mathbf{A} = x\hat{x}$ from Cartesian Coordinate to Spherical Coordinate System.
Solution:

The analytical solution of vector \mathbf{A} in Spherical coordinates is

$$\begin{aligned} \mathbf{A} &= R \sin\theta \cos\phi \left(\sin\theta \cos\phi \hat{R} + \cos\theta \cos\phi \hat{\theta} - \sin\phi \hat{\phi} \right) \\ &= R \sin^2\theta \cos^2\phi \hat{R} + R \sin\theta \cos\theta \cos^2\phi \hat{\theta} - R \sin\theta \sin\phi \cos\phi \hat{\phi} \\ &= R(1 - \cos^2\theta) \cos^2\phi \hat{R} + R \sin\theta \cos^2\phi \cos\theta - R \sin\theta \cos\phi \sin\phi \hat{\phi} \end{aligned}$$

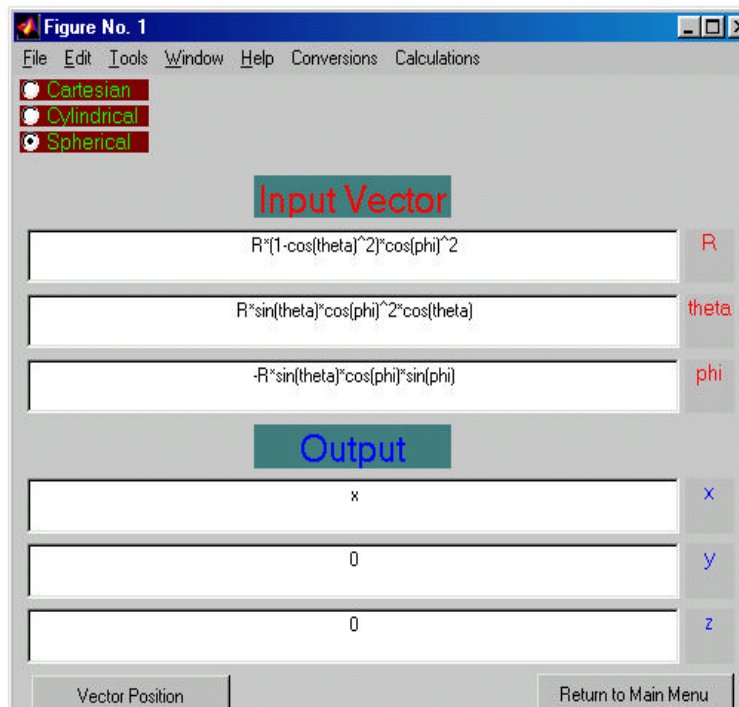
As verification, students will check their solution using the software. They will access the software, choose the conversion menu, choose coordinate system of the original vector as input, and choose the desired coordinate system of the vector as output. The window will show the input vector and the output vector as follow:



Example 2: Convert vector $\mathbf{B} = R(1 - \cos^2\theta)\cos^2\phi\hat{R} + R\sin\theta\cos^2\phi\cos\theta - R\sin\theta\cos\phi\sin\phi\hat{\phi}$ from Spherical Coordinate to Cartesian Coordinate systems.

Solution:

The vector above is the output of the previous example, the output vector will be converted back to the original vector as shown in the window below. The analytical solution is not presented because it is a very long solution.



The following three examples show the software capability of performing the curl, divergence, and gradient operations.

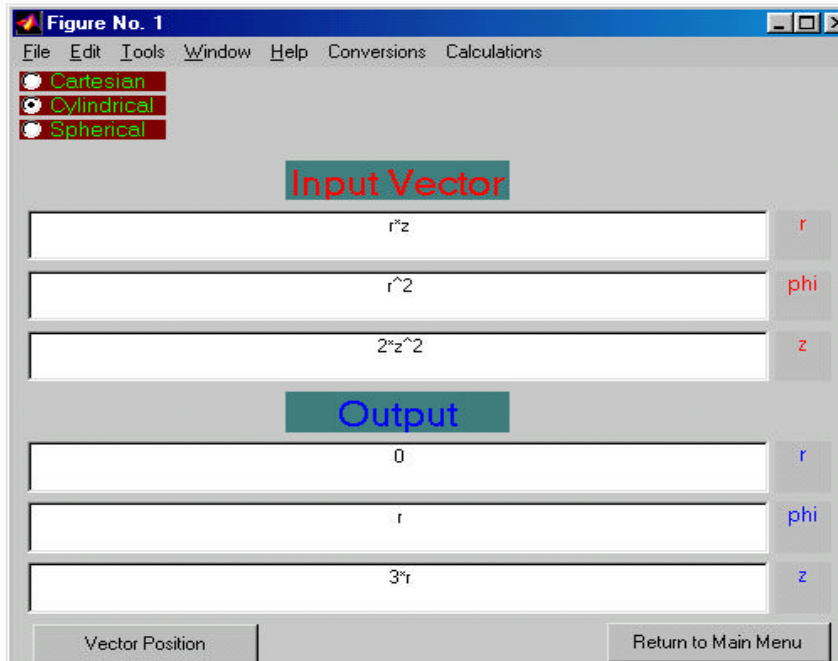
Example 3: Calculate the curl of vector $\mathbf{B} = rz\hat{r} + r^2\hat{\phi} + 2z^2\hat{z}$.

Solution:

The vector is given in Cylindrical coordinate system, the analytical solution of the curl is:

$$\begin{aligned}\nabla \times \mathbf{B} &= \left[\frac{1}{r} \frac{\partial}{\partial \phi} (2z^2) - \frac{\partial}{\partial z} (r^2) \right] \hat{r} + \left[\frac{\partial}{\partial z} (rz) - \frac{\partial}{\partial r} (2z^2) \right] \hat{\phi} + \frac{1}{r} \left[\frac{\partial}{\partial r} (r^3) - \frac{\partial}{\partial \phi} (rz) \right] \hat{z} \\ &= r\hat{\phi} + 3r\hat{z}\end{aligned}$$

The software will show the following window after the user choose the appropriate menus and functions, which confirms the analytical answer:



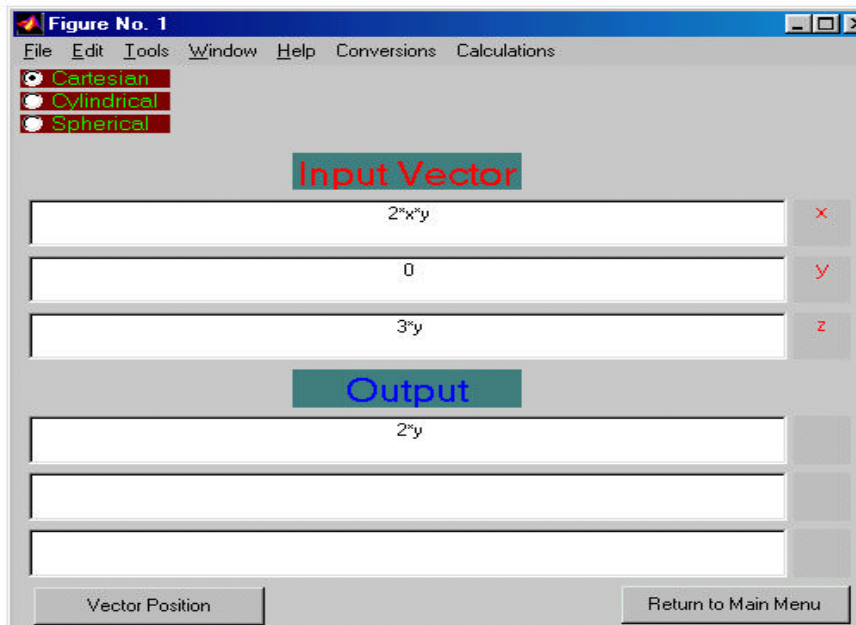
Example 4: Calculate the divergence of the vector field $\mathbf{F} = 2xy\hat{x} + 3y\hat{z}$.

Solution:

The vector is given in Cartesian coordinate system, the analytical solution of the divergence is:

$$\begin{aligned}\nabla \cdot \mathbf{F} &= \frac{\partial}{\partial x}(2xy) + \frac{\partial}{\partial y}(0) + \frac{\partial}{\partial z}(3y) \\ &= 2y\end{aligned}$$

From the above result, it can be seen that the final answer, which is the divergence of a vector field, is a scalar. The software will show a window confirms the obtained result. It is shown also the result is a scalar.



Example 5: Calculate the transmission line parameters at 1 MHz for a rigid coaxial air-line with an inner conductor diameter of 0.6 cm and an outer conductor diameter of 1.2 c where c is the speed of light. The conductors are made of copper.

After accessing the software and choosing the proper menus and functions while inserting zero for the unknown values or the values to be calculated, the output will be as shown in the following window:

Transmission Line Selection

Coaxial

Accept Selection

σ_c

Copper

Value of Sigma-c
If "User" selected, enter value.
Press enter while in input box

58000000

Accept Sigma

a-outer radius of inner conductor: 0.003 meters

b-inner radius of outer conductor: 0.006 meters

Rs-Intrinsic Resistance: 0 Ohms

f-frequency: 1000000 Hertz

Relative Permittivity Epsilon-r: 1

Relative Permeability Mu-r: 1

Conductivity of Insulator Sigma: 0 Semens / meter

Only one of these parameters must be known. The value 0 (zero) must be entered into the appropriate unknown characteristic cell.

Coaxial Line

Get Answers

Transmission Line Parameters

R	0.020761	Ohms / meter	L	1.3863e-007	Henrys / meter
G	0	Semens / meter	C	8.0259e-011	Farads / meter

Back to Main Menu

Example 6: A 1.05-GHz generator circuit is connected to a load $Z_{load} = (100 + j50) \Omega$, through a 50Ω , 67-cm-long lossless transmission line. The phase velocity is 0.7 times the speed of light. Find Γ and the standing wave ratio. The software will show the following window:

Gamma: 0

omega: 6597344572.53 rad/sec

Zo: 50 Ohms

Phase Velocity: 210000000 m/sec

Zload: 100+50i Ohms

Length: 0.67 m

lambda: 0 m

Vo: 0 Volts

Circuit Type: Loaded Circuit

Directions:

Enter Appropriate Data in cells, Enter 0 (zero) when the variable is unknown or if the variable is to be solved for.

Clear

Answer Section

Gamma: 0.4+0.2i

Vminima: 0 Volts

Zin: 0+36.3271i Ohms

Standing Wave Ratio: 2.618

Vmaxima: 0 Volts

Zload: 100+50i Ohms

Get Answers

Conclusions

The goal of this design was to develop an electromagnetic software package. This software's primary purpose would be to help students of electromagnetics with their studies. It would provide a supplement for the students and gives them a way to check homework solutions. The initial software performs vector conversions, vector calculations, and transmission line analysis. This software would also be made versatile so that future additions to the subject matter could be added.

Problems did arise during the development of the software. These problems included unfamiliarity with GUIs, symbolic simplification, resizing, and general programming problems. The best and most economical solution was used to solve these problems. The program was also tested. This testing insured that the function coding was correct and would provide the right answers to the students.

A graphical user interface (GUI) is used to provide an atmosphere of professionalism and accessibility. All design goals were met. The software has shown its effectiveness.

Acknowledgment

The work done by Artemas Herzog, Matt Orient, and Greg Davis is acknowledged.

Bibliography

1. B. S. Guru and H. R. Hiziroglu, Electromagnetic Field Theory Fundamentals, PWS, 1998.
2. C. R Paul, K. W. Whites and S. A. Nasar, Introduction to Electromagnetic Fields, 3rd Edition, McGraw-Hill, 1998.
3. K. Demarest, Engineering Electromagnetics, Prentice Hall, 1998.
4. F. Ulaby, Fundamentals of Applied Electromagnetics, Prentice Hall, 2001.
5. D. Hanselman, B. Littlefield, Mastering MatLab, Prentice Hall, 2001.

Biography

KHALID S. AL-OLIMAT was born in Jordan in 1967. He received the B.S. degree in electrical engineering from Far Eastern University in 1990, the M.S. degree in manufacturing engineering from Bradley University in 1994, and the Ph.D. degree in electrical engineering from The University of Toledo in 1999. His research and teaching interests are in the areas of power systems, electric machines, control systems, and fuzzy logic.