International Journal of Advanced Technology in Engineering and Science www.ijates.com
Volume No.02, Issue No. 09, September 2014 ISSN (online): 2348 – 7550

TRANSFORMER DESIGN GUI APPLICATION FOR HIGH FREQUENCY ELECTRICAL CIRCUITS

Hemant Pandey[1], Upendra Joshi[2], Protim Paul[3], Aditya Naithani[4]
[1, 4] *Department of CSE, ASET, Amity University, Noida, (India)*
[2, 3] *Department of EEE, ASET, Amity University, Noida, (India)*

ABSTRACT

The purpose of this research paper is to understand the concepts of Two-switch forward convertor, practically understand it's implementation and eventually linking it with the cyber world by create a GUI application for checking the transformer design. The manual process of performing this task is quite a tedious one. Even a slight error during the execution can lead to a lot of problems. Therefore this application is made to automate such an important task for its effective usage. The project allows us to check the transformer design by taking some of the required values from the user and then performing mathematical calculations as per certain formulae and hence give the desired output. The final project report also highlights the various software, tools and languages used in the development of the application.

Key Words–Transformer Design, GUI Application, High Frequency Transformers

I. INTRODUCTION

The research paper on Two-switch forward convertor aims to understand the concepts of Two-switch forward convertor, practically understand it's implementation and eventually linking it with the cyber world by create a GUI application for checking the transformer design. The manual process of performing this task is quite a tedious one. Even a slight error during the execution can lead to a lot of problems. Therefore this application is made to automate such an important task for its effective usage. The project allows us to check the transformer design by taking some of the required values from the user and then performing mathematical calculations as per certain formulae and hence give the desired output. The final project report also highlights the various software, tools and languages used in the development of the application. The process started by first gaining a thorough and sound knowledge about a transformer and then understanding the various fields required to perform the mathematical calculations which help in validating the design. The application is a GUI based application that is created using JAVA-programming language, including concepts of SWINGS, AWT. The application is created on Net Beans IDE which made our task easier because of its user friendly interface and coding options. Finally, the output of the application was recorded in MySQL database which was connected to the application using the JDBC-ODBC driver

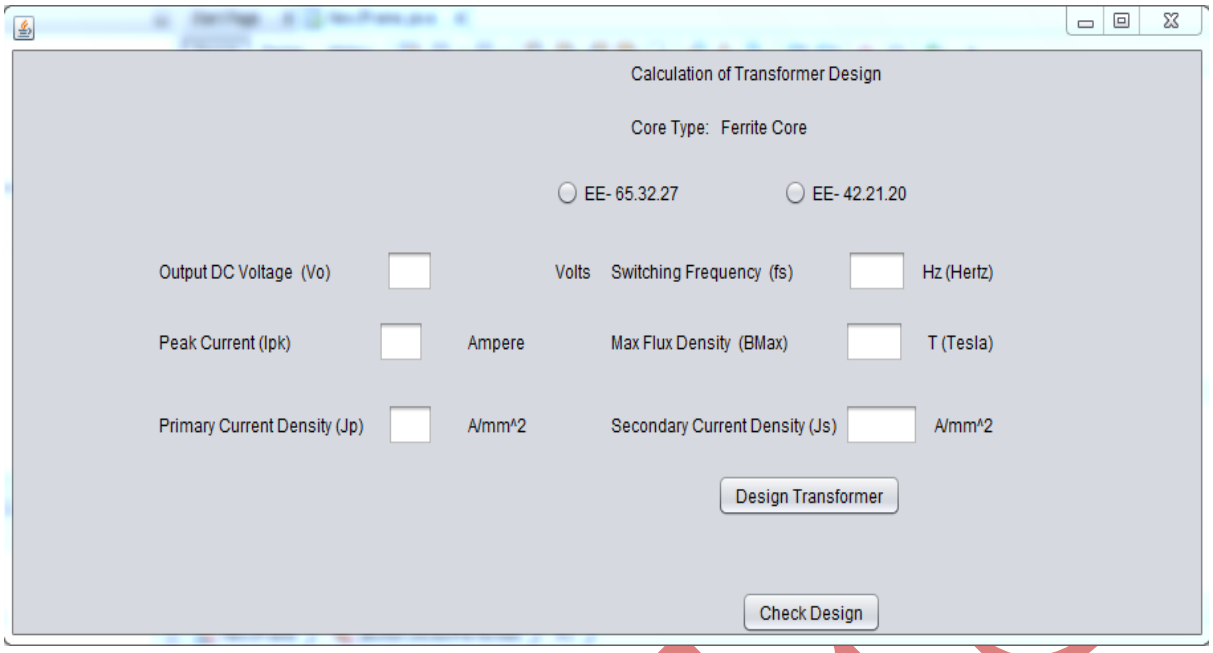

Fig 1: Front-Page Of GUI Application for Transformer Design

CODE WITH EXPLANATION:

The application is a GUI based application that is created using JAVA-programming language, including concepts of SWINGS, AWT. The application is created on NetBeans IDE which made our task easier because of its user friendly interface and coding options. Finally, the output of the application was recorded in MySQL database which was connected to the application using the JDBC-ODBC driver.importjava.sql.*;

importjava.math.*;

importjavax.swing.AbstractAction.*;

importjavax.swing.JDialog;

importjavax.swing.JOptionPane;

Through the above lines, we are importing the various JAVA packages from the library required to run the program efficiently.

public class NewJFrame extends javax.swing.JFrame {

in the above line, a class is defined for laying the body of the code.

public double ac, aw;

in the above line, variables are defined as public

publicNewJFrame() {

initComponents();

}

@SuppressWarnings("unchecked")

The above 4 lines, specify the jframe function/method and are used to initialize the components in the beginning. This method is called from within the constructor to initialize the form.

```
private void switchfreqActionPerformed(java.awt.event.ActionEventevt) {
}
```

The above line is the action method for the switching frequency text field

```
private void bmaxteslaActionPerformed(java.awt.event.ActionEventevt) {
}
```

The above line is the action method for the max flux density/bmax text field

```
private void jButton1ActionPerformed(java.awt.event.ActionEventevt) {
```

The above line is the action method for the DESIGN TRANSFORMER button. Everything written in it is responsible for the events taking place by clicking on it.

```
floatvo= Float.parseFloat(odcv.getText());
floatfs= Float.parseFloat(switchfreq.getText());
floatbmax= Float.parseFloat(bmaxtesla.getText());
floatipk= Float.parseFloat(ipkval.getText());
floatjp= Float.parseFloat(primjp.getText());
floatjs= Float.parseFloat(secjs.getText());
```

In the above lines, we are storing the values from the text field into variables, and eventually typecasting them.

```
if(jRadioButton1.isSelected()==true)
{
ac=537.24;aw=264;
}
else if(jRadioButton2.isSelected()==true)
{
aw=256.04;ac=244;
}
```

In the above lines we are making decisions by using the IF-ELSE cases of selecting the various radio buttons, and finally selecting the value of ac and aw

```
double n1= vo/((2*fs)*bmax*ac);
doubletr= 0.4*(504/(vo+1));
double n2= n1/tr;
doubleisec=ipk*(0.6324);
doubleiprim=isec/tr;
doubleacpri=iprim/jp;
doubleacsec=isec/js;
doubleawdg=(acpri*n1)+(acsec*n2);
```

the above lines, illustrate the various mathematical formulae used to validate the design of the transformer.

```
if (awdg<=(aw*0.4))
{
JOptionPane.showMessageDialog(null,"Design is Valid","Attention",JOptionPane.INFORMATION_MESSAGE);
}
else
{
JOptionPane.showMessageDialog(null,"Design is Invalid","Attention",JOptionPane.INFORMATION_MESSAGE);
}
```

In the above lines we are again making decisions by using the IF-ELSE cases, where if the mathematical condition is met, the corresponding message would be shown in another JOption Pane.

```
try
{

Class.forName("com.mysql.jdbc.Driver");
   Connection con= DriverManager.getConnection("jdbc:mysql://localhost/research?user="+"root"+"&password="+"root");
```

In the above lines, we registered the driver to connect with MySQL.

```
PreparedStatementstmt = con.prepareStatement("INSERT INTO research.transformer (vo,fs,bmax,n1,ipk,n2,tr,iprim,isec,acpri,acsec,awdg) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ? )");
stmt.setFloat(1, vo);
stmt.setFloat(2, fs);
stmt.setFloat(3, bmax);
stmt.setDouble(4, n1);
stmt.setDouble(5, ipk);
```

```
stmt.setDouble(6, n2);
stmt.setDouble(7, tr);
stmt.setDouble(8, iprim);
stmt.setDouble(9, isec);
stmt.setDouble(10, acpri);
stmt.setDouble(11, acsec);
stmt.setDouble(12, awdg);
stmt.executeUpdate();
stmt.close();

 }
```

In the above lines, we added the values of the variables into the various fields in the database.

```
catch(Exception e)
 {
System.out.print(e);
 }
```

The above line allows us to through an exception if any error occurs.

```
}
private void jRadioButton1ItemStateChanged(java.awt.event.ItemEventevt) {
}
```

The above line is the action method for the first radio button.

```
private void jRadioButton2ItemStateChanged(java.awt.event.ItemEventevt) {
}
```

The above line is the action method for the second radio button.

```
private void jButton2ActionPerformed(java.awt.event.ActionEventevt) {

JOptionPane.showMessageDialog(null,"Design parameters \n \n
","Attention",JOptionPane.INFORMATION_MESSAGE);
   }

public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
newNewJFrame().setVisible(true);
        }
      });
```

In the above lines, we allow the visibility of the JOption Pane earlier used to validate the design of the transformer.

}

privatejavax.swing.JTextFieldbmaxtesla;
privatejavax.swing.ButtonGroup buttonGroup1;
privatejavax.swing.JTextFieldipkval;
privatejavax.swing.JButton jButton1;
privatejavax.swing.JButton jButton2;
privatejavax.swing.JLabel jLabel1;
privatejavax.swing.JLabel jLabel10;
privatejavax.swing.JLabel jLabel11;
privatejavax.swing.JLabel jLabel12;
privatejavax.swing.JLabel jLabel13;
privatejavax.swing.JLabel jLabel14;
privatejavax.swing.JLabel jLabel15;
privatejavax.swing.JLabel jLabel16;
privatejavax.swing.JLabel jLabel19;
privatejavax.swing.JLabel jLabel2;
privatejavax.swing.JLabel jLabel3;
privatejavax.swing.JLabel jLabel37;
privatejavax.swing.JLabel jLabel4;
privatejavax.swing.JLabel jLabel5;
privatejavax.swing.JLabel jLabel6;
privatejavax.swing.JLabel jLabel7;
privatejavax.swing.JLabel jLabel8;
privatejavax.swing.JLabel jLabel9;
privatejavax.swing.JRadioButton jRadioButton1;
privatejavax.swing.JRadioButton jRadioButton2;
privatejavax.swing.JTextFieldodcv;
privatejavax.swing.JTextFieldprimjp;
privatejavax.swing.JTextFieldsecjs;
privatejavax.swing.JTextFieldswitchfreq;

The above mentioned line mentions all the swing functionalities that we have used so far in designing the GUI of our application.

II. SOME SCREEN SHOT OF THIS APPLICATION ARE GIVEN BELOW:

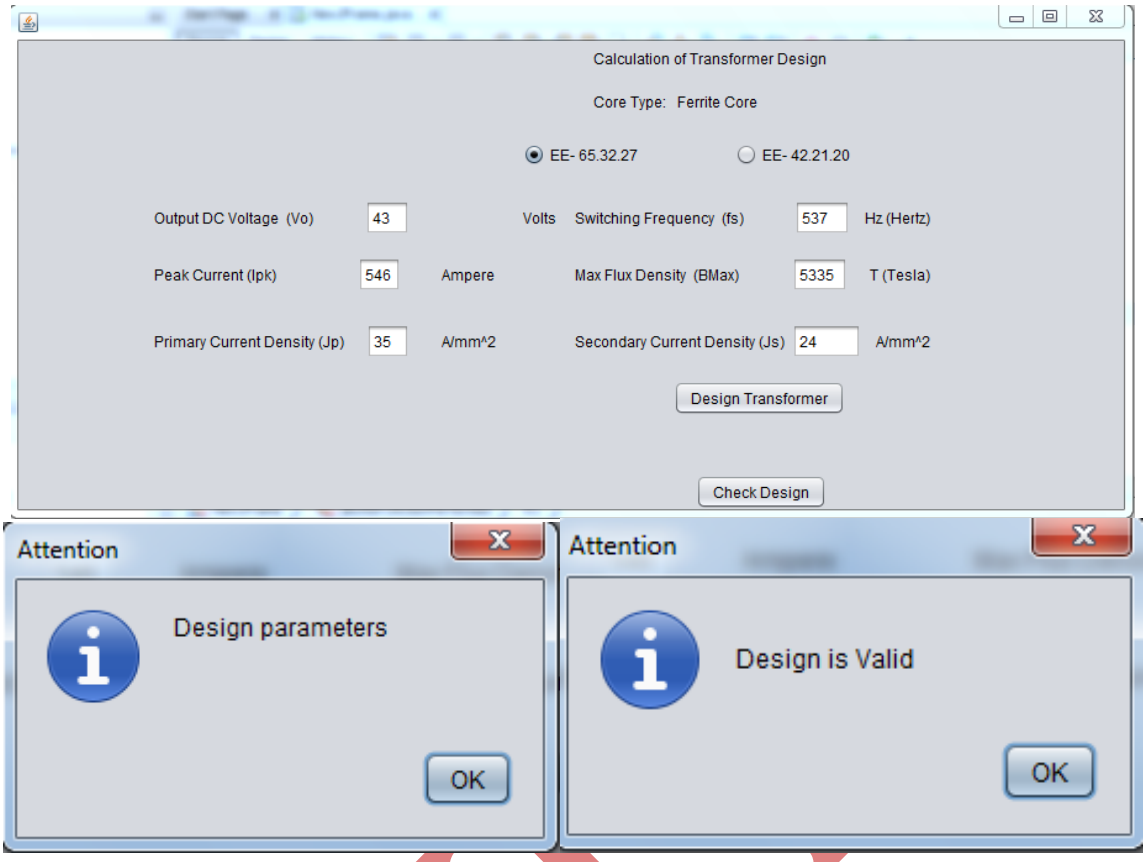

Fig2: Screen-Shot of GUI Application for Transformer Design

III. CONCLUSION

Throughout the course of the development phase of the application, we made sure that the application would certainly play an impactful role in this research. It automates the task of validating a transformer design. For the future, we will also add option for validating the inductor design and other relevant features associated with a two-switch forward convertor.

REFERENCES

[1] Muhammad H. Rashid, "Power Electronics, Circuits, Devices, and Applications", Third Edition, Pearson Education, Inc., 2004.

[2] Carl Nelson & Jim Williams, "Linear Technology, LT1070 Design Manual", 1986.

[3] Marty Brown, "Practical Switching Power Supply Design", New York: Academic Press, Inc., 1990.

[4] Irving M. Gottlieb, "Power Supplies, Switching Regulators, Inverters, & Converters", New York: McGraw-Hill, 1993.

[5] D. M. Mitchell, "DC-DC Switching Regulator Analysis", New York: McGraw-Hill, 1988.

[6] Anita Soni, "DC-DC Switching Boost Converter", ILLINOIS College of Engineering, 1999.

[7] A. Harry, E. Robert and M. Dragan "DC-DC Converter Design for Battery-Operated Systems," 26th Annual IEEE

[8] Power Electronics Specialists Conference, CO 80309-0425 USA, 1998.