Article **Journal-Mathematical and Quantitative Methods**
June, 2022 Vol.6 No.10 25-36

Development of a graphical interface programmed with Matlab App Designer for the analysis of the SCR thyristor using a sinusoidal signal source for its single-phase configuration

Desarrollo de una interfaz gráfica programada con Matlab App Designer para el análisis del tiristor SCR usando una fuente con señal sinusoidal para su configuración monofásica

GONZÁLEZ-GALINDO, Edgar Alfredo†*, VAZQUEZ-RIOS, Alan Ismael, FLORES-DELGADO Alejandro and GARCÍA-PÉREZ, Jorge

Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Aragón, Centro Tecnológico Aragón, Measurement and Instrumentation and Control Laboratory. Mexico.

ID 1 st Author: *Edgar Alfredo, González-Galindo* / **ORC ID:** 0000-0003-4654-9595, **Researcher ID Thomson:** G-79272018, **CVU CONACYT ID:** 351785

ID 1st Co-author: *Alan Ismael Vazquez Rios* / **ORC ID:** 0000-0001-5416-5001, **CVU CONACYT ID:** 1259380

ID 2nd Coauthor: *Alejandro Flores Delgado*/ **ORC ID:** 0000-0002-3600-034X, **Researcher ID Thomson:** GZG-6412-2022, **CVU CONACYT ID**: 1192963

ID 3rd Co-author: *Jorge, Pérez García* / **ORC ID:** 0000-0002-6085-6375, **Researcher ID Thomson:** ACR-3522-2022, **CVU CONACYT ID**: 10146

DOI: 10.35429/JMQM.2022.10.6.25.36 Received March 28, 2022; Accepted June 10, 2022

Abstract

A graphical interface programmed with MATLAB App Designer was developed for the analysis of the SCR thyristor using an alternating current source with a single-phase sinusoidal signal. As a first step, the mathematical analysis was made to design the graph of the firing angle in a circuit made up of an SCR. Following this, programming was carried out to calculate the intersection points of the zero crossing for the simulation in a more interactive way, where the user makes changes in the parameters in order to alter the oscillations and obtain the graphic behavior of the load and the thyristor visually. This program allows students to use it intuitively along with the view of the calculations, graphs, and to control the trigger signal on the interface without making use of virtual simulators and measurement equipment available on the market at a high cost. This facilitates the teaching-learning abilities in the subjects of computer fundamentals, applied programming and power electronics. This graphical interface allows the development of theoretical-practical activities in person and at a distance, to the design of electronic circuits.

SCR, Electronic, Power, Interface

Resumen

Se desarrolló una interfaz gráfica programada con MATLAB App Designer para el análisis del tiristor SCR usando una fuente con señal sinusoidal para su configuración monofásica. Se hizo el análisis matemático para realizar la gráfica del ángulo de disparo en un circuito conformado por un SCR, se llevó a cabo la programación para calcular los puntos de intersección del cruce por cero para la simulación de forma interactiva donde el usuario realiza los cambios de los parámetros para alterar las oscilaciones y obtener el comportamiento gráfico de la carga y del tiristor en una interfaz gráfica. Este programa permite que los estudiantes puedan usarlo de forma intuitiva y visualizar la interfaz de los cálculos, la gráfica y controlar la señal de disparo sin hacer uso de simuladores virtuales y equipos de medición que son costosos en el mercado, esto permite que los estudiantes complementen la enseñanza-aprendizaje en las materias de fundamentos de computación, programación aplicada y electrónica de potencia. Esta interfaz gráfica facilita la elaboración de las actividades teórico-prácticas en forma presencial y a distancia para el diseño de circuitos electrónicos.

SCR, Electrónica, Potencia, Interfaz

Citation: GONZÁLEZ-GALINDO, Edgar Alfredo, VAZQUEZ-RIOS, Alan Ismael, FLORES-DELGADO Alejandro and GARCÍA-PÉREZ, Jorge. Development of a graphical interface programmed with Matlab App Designer for the analysis of the SCR thyristor using a sinusoidal signal source for its single-phase configuration. Journal-Mathematical and Quantitative Methods. 2022. 6-10: 25-36

* Correspondence to Author (e-mail: unam_alf@comunidad.unam.mx)
† Researcher contributing first author

Introduction

It is of great relevance to apply previous knowledge of programming to develop graphical interfaces that allow them to be coupled to projects in the subjects of the electrical engineering degree in order to reinforce teaching-learning. In some subjects it is essential to complement the electronic subjects by using simulators, many of these tools are installed on the computers in the computer rooms within the faculty. Engineering students have disadvantages when carrying out assigned activities such as homework, because many simulators are very expensive on the market.

There are computer programmes that the university has made available so that students, teachers and researchers can make use of these tools, due to agreements with some companies that develop these programmes, without restrictions or trial periods so that they can be installed on the computers of each student and work remotely. Many simulators only show the behaviour of electronic circuits using some of the virtual instruments that the programme has, and do not indicate the physical-mathematical procedure to reach the result or how the programming is structured. Students are able to develop object-oriented programming in order to generate a graphical interface.

This will help the creativity of power electronics projects and generate didactic material to reinforce teaching in an interactive way. It is important that the development of a graphical interface is accompanied by a small manual that describes and allows the user to enter and obtain data on the physical variables, according to Marulanda (2022) a description manual is important in case you have to configure the communication of a hardware and/or data acquisition card to communicate in any port of the computer, in order to synchronise the transmission speed and depending on the need, it can be configured within the graphical interface.

LabVIEW graphical interface

In the development of graphical interfaces there are different tools, such as the LabVIEW programme distributed by the company National Instruments, from which an infinite number of applications have been created.

According to Rosales (2012), with the support of the metrology laboratory of the company Electricaribe, they developed a project consisting of the automation of a single-position calibration table for three-phase active and reactive energy meters, using the LabVIEW programme in its 2010 version. Natural disasters have motivated the generation of new technologies that help to react more quickly in order to be prepared and not run the risk of losing human lives. The Metropolitan Institute of Technology set out to make a graphical interface using LabVIEW, developed a low-cost seismograph, with portable devices deployed to form a network of sensors that are connected to the network and thus allow us to remotely detect, analyse and report seismic peaks that are outside the Earth's natural threshold. In this way, we can generate graphical reports and alert those in charge of monitoring the state of the earth in real time (Isaza, 2018).

Python graphical interface

Algorithms are used to solve problems, in an orderly manner by means of instructions that you want to perform. In the work carried out by the National Technological University of Argentina, a graphical interface was developed using the Python programming language, for the basis of the GUI prototype built in Matlab, and subsequently be used in the electronic instrument that is configured with an analogue autoculminator.

The software that was programmed is intended to establish a connection between the operator and the instrument. It gives the facility to position itself automatically without the need for the user to do it manually (Bergues, 2021). The University of Valladolid created a graphical interface developed in Python, which simulates the processes of a sugar plant, so that any operator is able to control it without the need for programming. It is worth mentioning that the software is designed with Wonderware, a tool that facilitates the simulation of industrial environments and these communicate with the OPC UA protocol that allows machine-to-machine communication. If at any time the application or server stops receiving data, a message will be sent to the client to go and monitor the problem (Hernández, 2019).

In parallel, at the Polytechnic University of Valencia, a graphical interface was developed in the Python programming language, which segments brain images using a set of machine learning algorithms or better known as "Deep learning". The aim is to speed up the process so that it is semi-automatic. This interface generates masks for the objects that need to be segmented. To obtain the medical images, pyCharm or ITK-Snap development environments were used (Estevan, 2019). In control and monitoring systems, HMI (Human Machine Interface) communication can be programmed using the Python programming language to monitor physical variables so that they can be measured in real time; PLCs are generally used, If Modbus communication is used, according to Romero (2022) using S71200 PLCs, a SIMATICS V20, a SENTRON PAC 3100 electrical parameter meter, programmed through Python makes systems that require licenses cheaper to use, which would be adapted to small and medium-sized industries.

The development of graphical interfaces can be found in different areas of engineering, which allow users to interact easily through HMI for manipulation and understanding. According to Lorenzana (2022) generating a graphical interface can help to monitor information and get an immediate response through serial communication with the OpenCm9.04 of ROBOTIS that controls the gesticulation of the face, as well as Matlab that uses the GUI and the App Designer, Python uses its GUI development library, and Kivy. The developed application allows to identify the facial gestures in an aesthetic and simple way to identify the imitation and the emotion to be sent by serial communication. Some companies dedicated to developing software commercialise development cards to be able to apply them to control systems by providing their libraries, according to Rodriguez (2022) used a control system manufactured by the company 'SlushEngine Model D' applied to the movement of the joints, using the RBX1 Robot which was provided with the Liberia for programming in Python called Slush, These libraries allow the routine movement of the actuators and speed control of rotation in the motors, the direction of rotation plus angular displacement, the release of current in the motors and the function that determines whether the motor is in motion, and its graphical interface was developed with Python GUI.

Graphical interface in C++

There are other programs that help develop graphical interfaces such as the C++ programming language. These have a command line interface, the disadvantage of using such applications is that they are very robust and general purpose. The advantage of using a graphical interface is the convenience and ease of use it provides to the user. The article presents a technique for developing a graphical interface in Subtyping and OStream. The GUI was developed with the help of the C++ programming language where standard library containers, algorithms and iterators are used to implement much of its functionality.

The aim was to integrate a graphical interface into the legacy code and a program that displays the average based on a list of student grades (Malloy, 2001). Another example is the work carried out at the Polytechnic University of Valencia. A portable device was developed for the measurement of salt concentration in saline solutions by applying electrical impedance spectroscopy, using a series of programs that interacted jointly. This was possible due to the versatility of interaction that they present with different programming environments, in this particular work, an ESP32 microcontroller was used, the programming was carried out in C++ for signal processing. Subsequently, a graphical interface was developed using the Python programming language, which communicates with the ESP32 microcontroller. The sensor used in the development of the work was the EIS, as it allows the measurement of salinity (Peter, 2021).

Another advantage of using graphical interfaces is the ease of manipulation and the opportunity to make a computational laboratory (simulators), to perform controlled and replicable experiments. In this case, the Trade Network Game Lab (TNG Lab) software was used, which is based on C++ and this same tool allows to see the evolution of the trade networks in a real-time animation. More precisely, the configuration screen facilitates the user to set market parameter values, as they are key for each bilateral market simulation run with the same number of buyers and sellers. The aim of this work was to demonstrate, constructively, how CLs can be used to explore complex socio-economic processes that are difficult to model using standard analytical and statistical tools.

This demonstrates that graphical interfaces are useful not only in engineering, but also in other fields such as the social sciences (McFadzean, 2001).

In the course of Electrical and Electronic Engineering, use is made of programs with graphical interfaces that allow to see the behaviour when a signal acts with an electronic component, usually composed of function generators and oscilloscopes that are graphical interfaces that are integrated within them. Some of these programs are Proteus, Multisim, Pspice (Orcad) and there are others online. In this work a graphical interface was developed that allows to enter data, to develop mathematical calculations and to observe the graphical expression of the behaviour of the SCR using a source with sinusoidal signal for its single-phase configuration, developed in Matlab with the App Designer tool to complement the theoretical-practical knowledge in the teaching-learning, of the subjects fundamentals of computing, applied programming and power electronics.

Objectives

A graphical interface programmed with MATLAB App Designer will be developed for the analysis of the SCR thyristor using a sinusoidal signal source for its single-phase configuration as a half-wave rectifier, where the data will be introduced, the calculations will be carried out and the behaviour will be shown through the graphical representation.

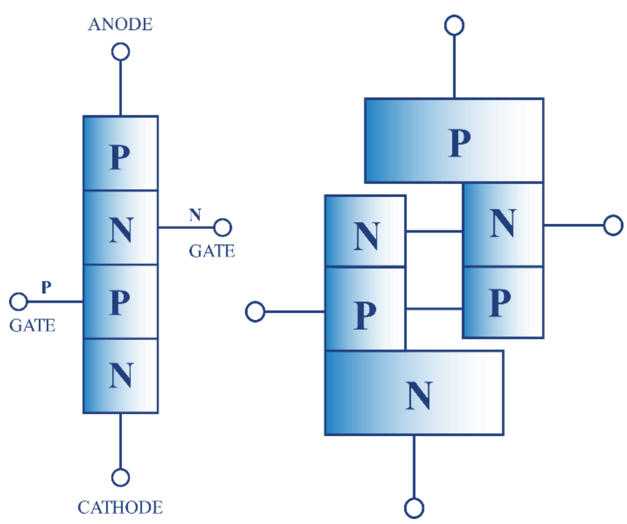
Hypothesis

The return to normality after the SARS CoV2 health contingency. The goals and challenges of the university academics of the Electrical and Electronic Engineering course at the Faculty of Higher Education Aragon at the National Autonomous University of Mexico, to transmit the knowledge of teaching-learning and to be an incentive that encourages young students of the engineering course to solve problems using the tools that are available to the university community. If a graphic interface programmed with MATLAB App Designer is developed for the analysis of the SCR thyristor using a source with a sinusoidal signal for its single-phase configuration as a half-wave rectifier.

Then it will be possible to develop projects that will put into practice what has been learned in the subjects of fundamentals of computing, applied programming and power electronics, omitting the electronics simulators because they only present the behaviour of the signal and do not show the mathematical analysis.

Development and Methodology

The SCR is very similar to a diode, except that, unlike the diode, the SCR consists of three terminals: anode (negative), cathode (positive) and the gate (the terminal control). The thyristor allows current to flow in one direction only, however, this action is subject to the firing angle (α), which is a pulse of current entering the gate. The behaviour of a thyristor is based on the PNPN semiconductor structure as each doping layer must preserve the ratio between them. When the gate has a pulse with the voltage and current values, to produce the trigger (Novas, 2019). The simple PNPN structure, like that of a conventional SCR, can be visualised as two transistors, one PNP and one NPN connected as shown in Figure 1.



Figuea 1 Diagram of the simple structure of an SCR

From the schematic circuit it is evident that the collector of the NPN transistor (together with a negative bias by the gate) distributes the base bias across the PNP transistor as shown in Figure 2.

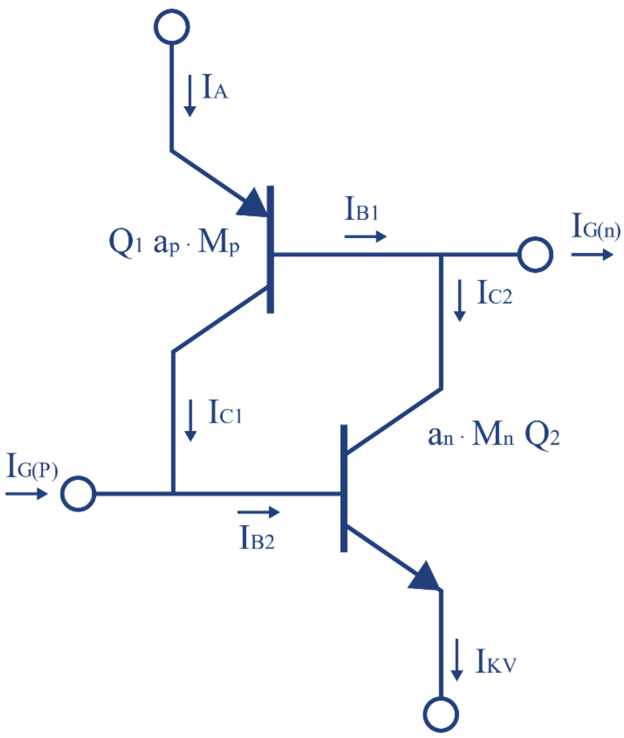


Figure 2 Equivalent circuit of an SCR with a PNP and NPN transistor array

Thyristors can be considered as triggers that can be controlled automatically by means of controlled pulses. The term thyristor refers to a semiconductor device designed to perform switching functions, in which its operating state depends on a reactive function between the p-n-p-n junctions. Thyristors can have two, three or four terminals, they can be unidirectional and bidirectional (Manuale, 1982),

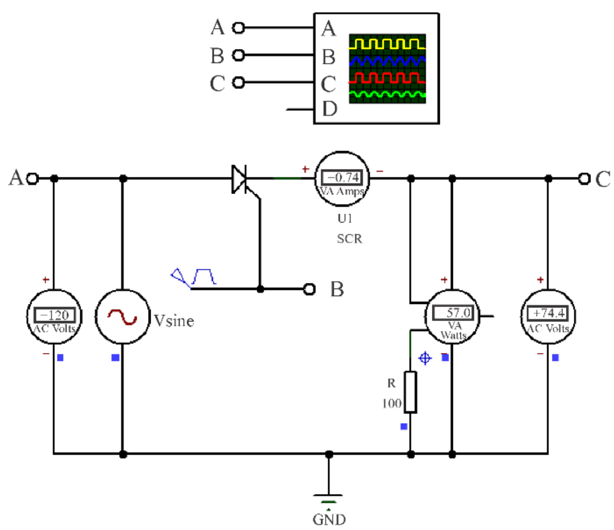
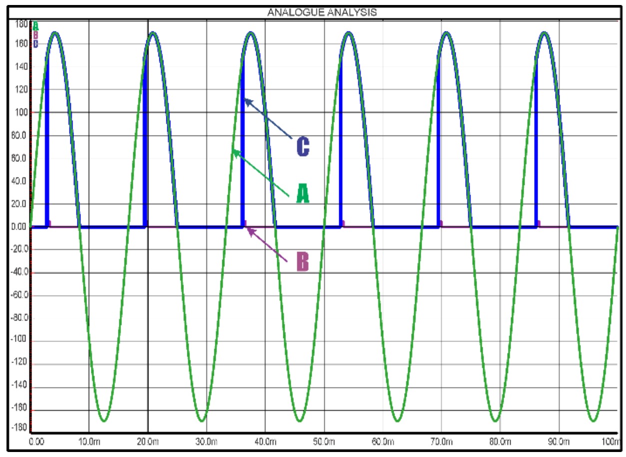


Figure 3 Circuit analysed to develop the graphic interface

Some of the most widely used simulators in the Electrical and Electronic Engineering course are Multisim and Proteus, both of which have virtual instruments that allow the behaviour of the signals to be observed, as shown in Figure 3, a four-channel oscilloscope.

Both programs only show the signals as they are: the input, the pulse or firing angle and the output or load signal as shown in Figure 1.



Graph 1 The signals from the simulation of the circuit in Figure 3 are shown, where A corresponds to the input, B is the firing angle and C is the load

Analysing the signal of the load, it is integrated from α to π, if the period of 2π is taken, knowing that from π to 2π, the integral is omitted because the SCR behaves as a diode letting pass from the firing angle and part of the positive half-cycle, and does not allow the negative half-cycle to pass, as shown in Equation 1.

$$V_{dc} = V_o = \frac{1}{2\pi}\int_{\alpha}^{\pi} V_m \sin(\omega t) d(\omega t) \qquad (1)$$

The value of the signal amplitude V_m was taken as a constant because it is not a function of ωt at the time of integration, therefore V_m= √2×120=169.71 V, by evaluating we have that cosα- 〖(cos〗 〖(π))〗 =cosα-(-1)=1+cosα, as seen in Equation 2.

$$V_{dc} = V_o = \frac{V_m}{2\pi}(1 + \cos\alpha) \qquad (2)$$

To obtain the current I_{dc}, Ohm's law is applied and V_{dc} is substituted from Equation 2.

$$I_{dc} = \frac{V_{dc}}{R} = \frac{V_m}{2\pi R}(1 + \cos\alpha) \qquad (3)$$

Obtaining the RMS voltage or root mean square value, it is applied to the periodic waveform of the input voltaje

$$V_{rms} = \sqrt{\frac{1}{2\pi}\int_{0}^{2\pi} v_0^2\,(\omega t) d(\omega t)} \qquad (4)$$

ISSN 2531-2979

Substituting Equation 1 into Equation 2 gives the following expression to obtain the rms voltage

$$V_{rms} = \sqrt{\frac{1}{2\pi}\int_{\alpha}^{\pi}[V_m \sin(\omega t)]^2 \, d(\omega t)} \qquad (5)$$

As the 〖Vm〗 [2] is a constant and is not a function of ωt it can be taken from the integral by applying the square root, giving us the value of the signal amplitude V_m

$$V_{rms} = \frac{V_m}{2}\sqrt{1 - \frac{\alpha}{\pi} + \frac{\sin(2\alpha)}{2\pi}} \qquad (6)$$

To obtain the rms current I_{rms}, Ohm's law is applied and V_{rms} is substituted from Equation 6, resulting in the following expression

$$I_{rms} = \frac{V_{rms}}{2} = \frac{V_m}{2R}\sqrt{1 - \frac{\alpha}{\pi} + \frac{\sin(2\alpha)}{2\pi}} \qquad (7)$$

To calculate the power factor, it is obtained from the ratio of the root mean square voltage squared inversely to the load value and divided by the product directly proportional to the root mean square current and inversely the product of the RMS source voltage times the root mean square current as shown in the equation below.

$$pf = \frac{Vrms^2}{R(Vs_{rms})(I_{rms})} \qquad (8)$$

If the firing angle $\alpha = \frac{\pi}{2}$ is substituted into Equation 2 gives V_{dc} which is expressed as follows, in case you want to calculate the $V_{dc} = 27\,V$

$$V_{dc} = \frac{V_m}{2\pi}\left(1 + cos\left[\frac{\pi}{2}\right]\right) \qquad (9)$$

$$V_{dc} = 0.1592(V_m)$$

Calculating the current I_{dc} in Equation 3 gives the following expression

$$I_{dc} = \frac{V_m}{2\pi}\left(1 + cos\left[\frac{\pi}{2}\right]\right) \qquad (10)$$

$$I_{dc} = 0.1592\left(\frac{V_m}{R}\right)$$

To obtain the r.m.s. voltage, the firing angle is substituted $\alpha = \frac{\pi}{2}$ in Equation 6

$$V_{rms} = \frac{V_m}{2R}\sqrt{\frac{1}{\pi}\left(\pi - \frac{\pi}{2} + \frac{\sin\left(2\left[\frac{\pi}{2}\right]\right)}{2}\right)} \qquad (11)$$

$$V_{rms} = (0.3536)V_m$$

To obtain the rms current I_{rms}, Ohm's law is applied and V_{rms} is substituted from Eq. 7

$$I_{rms} = \frac{V_m}{2R}\sqrt{\frac{1}{\pi}\left(\pi - \frac{\pi}{2} + \frac{\sin\left(2\left[\frac{\pi}{2}\right]\right)}{2}\right)} \qquad (12)$$

$$I_{rms} = 0.3536\frac{V_m}{R}$$

Equation 12 is used to calculate the AC output power.

$$P_R = (Vrms)(Irms) \qquad (13)$$

$$P_R = \frac{([0.3536]V_m)^2}{R}$$

The power of the direct current output is obtained as shown in Equation 14.

$$P_R = V_{dc}I_{dc} = \left(\frac{0.2534\times 10^{-4}[V_m{}^2]}{R}\right) \qquad (14)$$

The percentage efficiency is inversely proportional to the voltage V_{dc} with respect to the rms voltage V_{rms} , when $\alpha = \frac{\pi}{2}$ you have

$$\eta = \left(\frac{V_{dc}}{V_{rms}}\right)^2 \times 100\% = 20.27\% \qquad (15)$$

The form factor is the ratio between the rms value and the mean value, if the form factor is large, it indicates that the wave has little direct and much alternating component, the form factor is inversely proportional to the rms value V_{rms} with respect to the voltage V_{dc}, to calculate it, Equation 16 is used.

$$FF = \frac{V_{rms}}{V_{dc}} \times 100\% = 222.1\% \qquad (16)$$

Equation 17 is used to calculate the percentage wave factor.

$$RF = \left(\sqrt{FF^2 - 1}\right) \times 100\% = 198.31\% \qquad (17)$$

To calculate the ratio between the maximum system demand and the nominal capacity, the following ratio is used

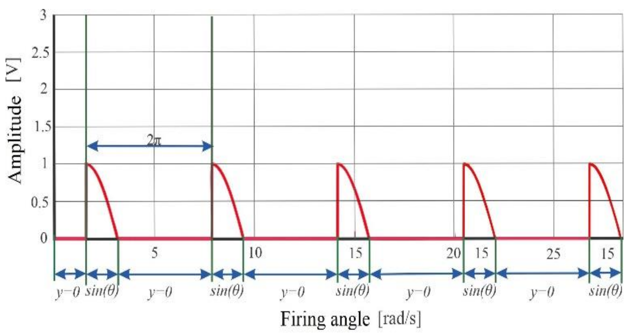
$$TUF = \frac{P_{dc}}{VI} = \frac{(V_{dc})(I_{dc})}{V_{rms}I_{rms}} = \frac{(0.0253(V_m)^2)R}{0.25(V_m^2)R} = 1.014 \qquad (18)$$

To calculate the firing angle we use Equation 2, by clearing the angle alpha from the cosine argument we have the expression as shown in the following equation.

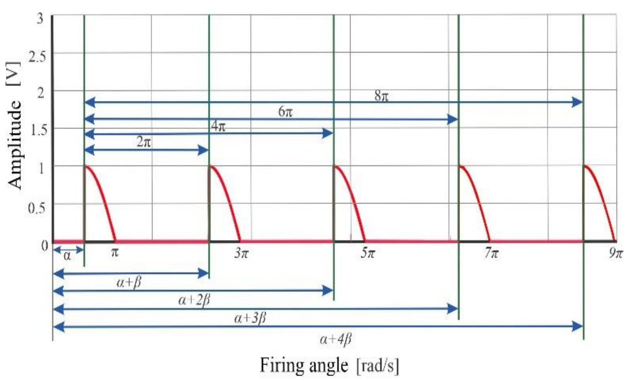
$$V_{dc} = \frac{V_m}{2\pi}(1 + \cos\alpha)$$

$$\alpha = cos^{-1}\left[V_0\left(\frac{2\pi}{V_m}\right) - 1\right] \qquad (19)$$

The following graph shows the behaviour of the SCR when it allows the positive half-cycle signal to pass when the trigger angle is sychronised and does not allow the negative half-cycle signal to pass, so we have a rectified half-wave signal and it shows the signal in a sychronised form.



Graph 2 SCR behaviour graph with synchronised firing angle for programming development in Matlab App Designer



Graph 3 The periodic behaviour of the SCR when the firing angle is activated synchronously α+nβ where n=1,2,3,…

Results

The relevance is that the student can develop the graphical interface to generate projects that facilitate creativity and make use of reasoning for the analysis of the SCR and perform the models for mathematical calculations.

In this particular case the use of MATLAB App Designer can generate the *.exe file which is the executable and 99% of the students see more dynamic and interactive in a way that helps the students of the Electrical and Electronic Engineering Career to stimulate and continue developing applications, since the way of programming is very simple and easy.

The programming code introduced in the MATLAB App Designer to place it in object-oriented programming, is very simple since we only select the button that will execute the operation by clicking the right mouse button, and select Sing Source callbacks→ Go to SingSourceButtonPushed callback, for the development of the calculations (MATLAB, 2022).

```
% Button pushed function: CalculatedoperationButton
function CalculatedoperationButtonPushed(app, event)
Vs = app.Vs.Value
f  = app.Frecuencia.Value
Vo = app.Vo.Value
R  = app.R.Value
osc = app.Oscilaciones.Value
b     =  acos(Vo*((2*pi)/(sqrt(2)*Vs))-1);
angulo=acos(Vo*((2*pi)/(sqrt(2)*Vs))-1);
    angrados=angulo*(180/pi);
    frecuancia_angular=2*pi*f;
    Vrms = ((sqrt(2)*Vs)/2)*sqrt(1-(angulo/pi)+(sin(2*angulo)/(2*pi)))
    Prms = Vrms^2/R;
    Irms = Vrms/R;
    Vca  = Vs*sqrt(2);
    Vdc  = Vo;
    Idc  = Vdc/R;
    n    = ((Vdc/Vrms)^2)*100
    FF   = (Vrms/Vdc)*100
    fp   = ((Vrms^2) / (R *120*Irms))
    RF   = sqrt((Vrms/Vdc)^2-1)*100;
    TUF  = (Vdc*Idc)/(Vrms*Irms);
    app.Wc.Text = [num2str(frecuancia_angular)]
    app.ang.Text = [num2str(angulo)]
    app.anggrados.Text = [num2str(angrados)]
    app.Vrms.Text = [num2str(Vrms)]
    app.Prms.Text = [num2str(Prms)]
    app.Irms.Text = [num2str(Irms)]
    app.Vca.Text = [num2str(Vca)]
    app.n.Text = [num2str(n)]
    app.FF.Text = [num2str(FF)]
    app.fp.Text = [num2str(fp)]
```

```
        app.RF.Text = [num2str(RF)]
        app.TUF.Text = [num2str(TUF)]
end
```

To consider the SCR as a diode where only the positive half-cycle is allowed to pass and to see the behaviour of the signal, select the Half-wave rectified signal button by right-clicking the mouse, callbacks→ Go to SingSourceButtonPushed callback, to make the graph.

```
function HalfwaverectifiedsignalButtonPushed(app, event)
cla(app.g1)
osc = app.Oscilaciones.Value
Vo = app.Vo.Value
Vs = app.Vs.Value
f  = app.Frecuencia.Value
b      =  acos(Vo*((2*pi)/(sqrt(2)*Vs))-1)
a=pi; c=2*pi; d=0; e=0; m=0; n=0;
for i=1:1:((f*1/f)*(osc)+1)
if rem(i,2)==0
m(1,1)=a
m(2,1)=a+(c)
end
if rem(i,2)>=0;
m(i+1,1)=a+(c*i);
end
end
for i=1:1:((f*1/f)*(osc)+1)
if(i<=1)
n(1,1)=b;
n(2,1)=b+(c*i);
end
if(i>=2)
n(i+1,1)=b+(c*i);
end
end
[p,q]=size(m)
m(p)
disp('      n         m')
disp([n m])
osc = app.Oscilaciones.Value
Vs = app.Vs.Value
f  = app.Frecuencia.Value
w =1 %2*pi*f
t= 0:0.1:n(i,1);
y=Vs*sin(w*t); %  xx= 0:0.01: (b);
y(y<0)=0
plot(app.g1,t,y,'.b') %plot(xx,0,'.r')
end
```

In order to plot the load signal when the firing angle is activated, the following code is entered, Loading sign SCR is selected by right-clicking the mouse, callbacks→ Go to SingSourceButtonPushed callback.

```
function LoadingsignSCRButtonPushed(app, event)
cla(app.g2)
osc = app.Oscilaciones.Value
Vo = app.Vo.Value
Vs = app.Vs.Value
f  = app.Frecuencia.Value
b      =  acos(Vo*((2*pi)/(sqrt(2)*Vs))-1)
a=pi; c=2*pi; d=0; e=0; m=0; n=0;
for i=1:1:((f*1/f)*(osc)+1)
if rem(i,2)==0
m(1,1)=a
m(2,1)=a+(c)
end
if rem(i,2)>=0;
m(i+1,1)=a+(c*i);
end
end
for i=1:1:((f*1/f)*(osc)+1)
if(i<=1)
n(1,1)=b;
n(2,1)=b+(c*i);
end
if(i>=2)
n(i+1,1)=b+(c*i);
end
end
disp('      n         m')
disp([n m])
tic
for i=1:1:((f*1/f)*(osc))
W=1
hold(app.g2, 'all');
% -----------Primero----------
xx= 0:0.1: (b);
plot(app.g2,xx,0,'.r')  %r
%------------Segundo---------
yyy= 0:5:(Vs*sin(b));
plot(app.g2,n(i,1),yyy,'.k')  %r
% -----------Tercero-----------
x= n(i,1):0.05: m(i,1);
y=Vs*sin(W*x);
plot(app.g2,x,y,'.b')
%-------------cuarto-----------
xxxx= m(i,1):0.1: n(i+1,1);
plot(app.g2,xxxx,0,'.g')
end
toc
grid
end
```

The graph of the complementary signal that does not allow the SCR to pass is generated when Complementary signal is selected by right clicking on the mouse, callbacks→ Go to SingSourceButtonPushed callback, enter the following code.

```
function
ComplementarysignalButtonPushed(app, event)
cla(app.g3)
osc = app.Oscilaciones.Value
Vo = app.Vo.Value
Vs = app.Vs.Value
f  = app.Frecuencia.Value
b      =  acos(Vo*((2*pi)/(sqrt(2)*Vs))-1)
a=pi; c=2*pi; d=0; e=0; m=0; n=0;
for i=1:1:((f*1/f)*(osc)+1)
if rem(i,2)==0
m(1,1)=a
m(2,1)=a+(c)
end
if rem(i,2)>=0;
m(i+1,1)=a+(c*i);
end
end
for i=1:1:((f*1/f)*(osc)+1)
if(i<=1)
n(1,1)=b;
n(2,1)=b+(c*i);
end
if(i>=2)
n(i+1,1)=b+(c*i);
end
end
disp('      n         m')
disp([n m])
tic
for i=1:1:((f*1/f)*(osc))
W=1
hold(app.g3, 'all');
% -----------Primero-----------
xx= 0:0.01:(b);
y=Vs*sin(W*xx);
plot(app.g3,xx,y,'.g')  %r
%------------Segundo---------
yyy= 0:10:(Vs*sin(b));
plot(app.g3,n(i,1),yyy,'.k')  %r
% ---------Tercero-------------
x= n(i,1):0.1: m(i,1);
plot(app.g3,x,0,'.b')
% -------------cuarto-----------
xxxx= m(i,1):0.1:n(i+1,1);
y=Vs*sin(W*xxxx);
plot(app.g3,xxxx,y,'.r')
end
toc
grid
end
end
```

Using the above equations and taking the values from the schematic circuit in Figure 3, where there is a load R=100Ω, the amplitude of the source signal V_s =120[V], while V_m=√2×120=169.71[V], with a frequency of 60 Hz, to obtain the angular frequency $\omega_C = 2\pi f = 376.9911 \left[\frac{rad}{s}\right]$ y $V_o = V_{dc} = 40\ [V]$ por this particular case, substituting we can calculate the firing angle α=61.2° and its value in radians is α=1.07[rad] , from Equation 2, the quadratic voltage or RMS is V_rms=75.6 [V] and the average current of the load is I_{rms}=0. 756 [A], to calculate the power dissipated in the load P_R=57.1 [W], while the average power is S=90.72VA, if we have both powers, we can calculate the power factor being pf=0. 629, the percentage calculation η=27.98%, the form factor FF=189.0613%, calculating the wave factor RF=160.39%, calculating the relationship between the maximum demand of the system and the nominal capacity TUF=0.2799 the calculations are visualised in Figure 3. All the previous calculations can now be observed in a simpler and more complete way in Figure 4 and no longer depend on a simulator as these do not show the information of the mathematical calculations made.

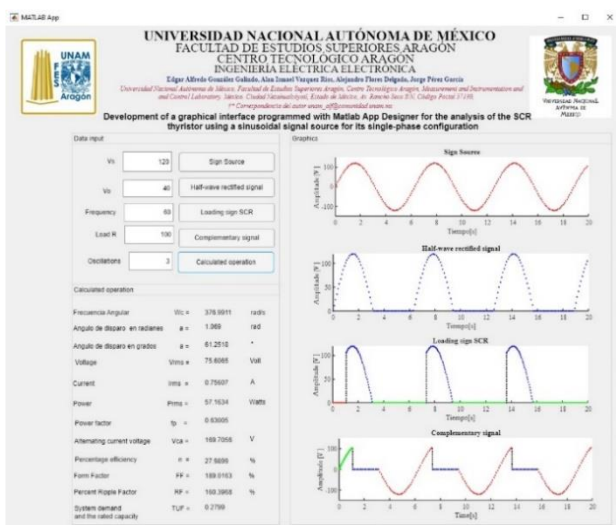


Figure 4 Development of the graphical interface where the mathematical calculations of Figure 3 of the SCR schematic circuit can be developed

Figure 5 shows section 1 where the data that will allow the calculations to be developed are entered. Section 2 will show each of the graphs and section 3 will show the values of the programmed calculations.

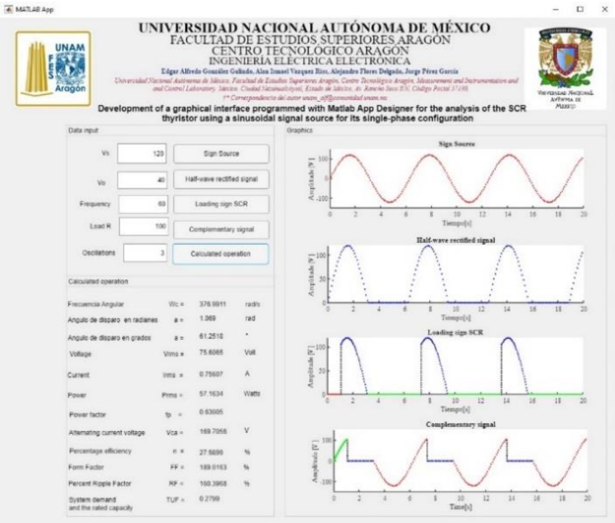


Figure 4 Development of the graphical interface where the mathematical calculations of Figure 3 of the SCR schematic circuit can be developed.

Figure 5 shows section 1 where the data that will allow the calculations to be developed are entered. Section 2 will show each of the graphs and section 3 will show the values of the programmed calculations.

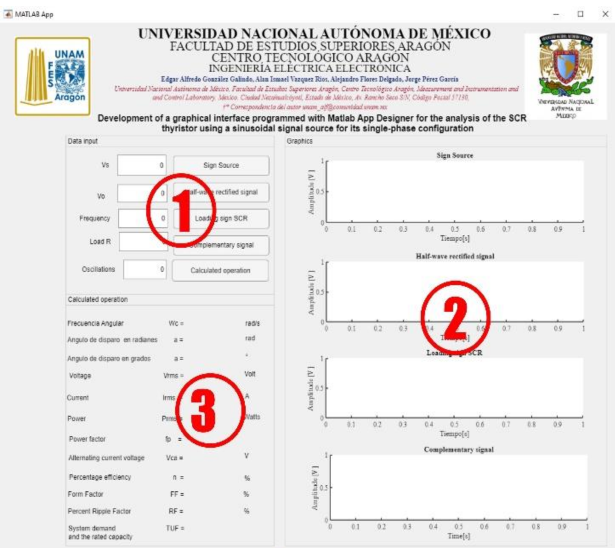


Figure 5 Sections of the graphical interface for entering the data, performing the calculations and displaying the graphical interpretation.

Figure 6 will show the calculation data when we present a firing angle $\alpha = \frac{\pi}{4}$ or as the case may be $\alpha = 45°$

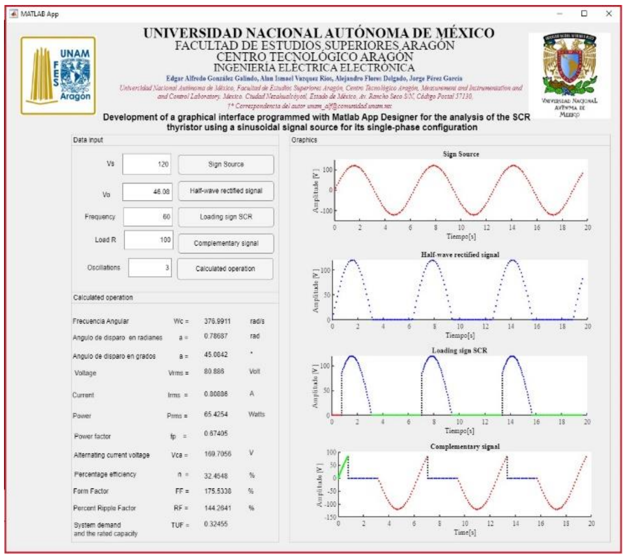


Figure 6 By introducing $\alpha = \frac{\pi}{4}$ se you can see the graphical expression, as well as the calculations by observing the shift that alpha generates when $\alpha = 45°$

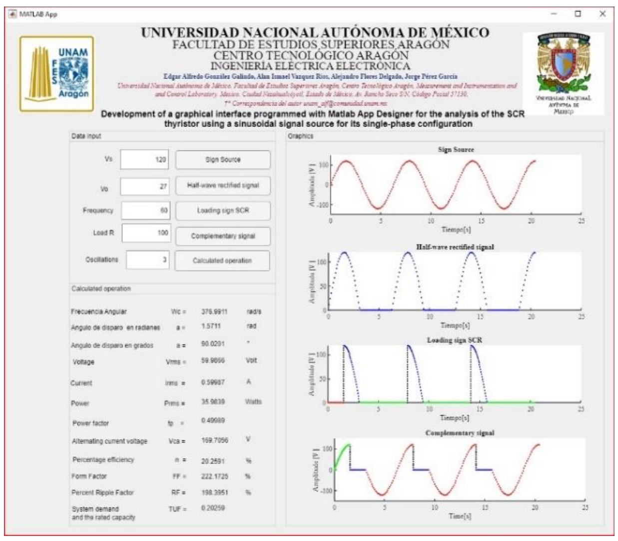


Figure 7 By introducing $\alpha = \frac{\pi}{2}$ the graphical expression can be observed, as well as the calculations by observing the shift that alpha generates when $\alpha = 90°$

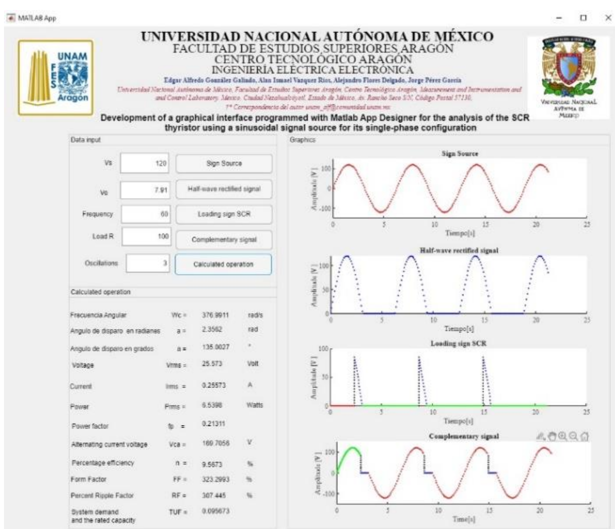


Figura 8. By introducing $\alpha = \frac{3\pi}{4}$ the graphical expression can be observed, as well as the calculations by observing the shift that alpha generates when $\alpha = 135°$

ISSN 2531-2979

Conclusions

Programming was achieved through MATLAB App Designer, where a graphical interface was generated to see the behaviour of the SCR when it is fed with a single-phase signal. The interface allows the operations to be carried out more quickly and to generate a self-assessment for the student of Electrical and Electronic Engineering, in the subject of Power Electronics for teaching-learning. The graphical interface was programmed and designed for engineering students to apply the knowledge of the subjects of fundamentals of computing and applied programming to omit virtual simulators that are expensive in the market and that students do not understand how they were internally programmed, this graphical interface allowed 99% of students to solve the activities related to the Power Electronics projects.

Acknowledgements

The authors would like to thank the Coordination of the Technology Centre of the Facultad de Estudios Superiores Aragón and the Electrical and Electronic Engineering Department of the Universidad Nacional Autónoma de México, for the facilities in the use of tools provided through agreements with different software companies. They would also like to thank Alberto González Ledesma for his technical support in the area of computer architecture of the Computer Engineering Department.

References

B.A. Malloy, J. D. McGregor, S.R. Hughes, (2015),“Exploiting iostreams to integrate a gui into a command driven application”, p 2, 7 URL https://doi-org.pbidi.unam.mx:2443/10.1080/1206212X.2001.11441646

Bergues, G. J., Schürrer, C., & Brambilla, N. (2021). Algoritmo de detección e Interfaz gráfica de usuario diseñados en Python para autocolimador digital. *Revista Tecnología y Ciencia*, *41*, 75–90. p,75,88 https://doi.org/10.33414/rtyc.41.75-90.2021

C. M. Isaza Monsalve, J. E. Muñoz Saldarriaga, L. M. Agudelo Mesa, (2018) “Sistema inalámbrico detector de vibraciones sísmicas, con interfaz gráfica en el entorno industrial LabVIEW”, [Informe final de trabajo de grado, Instituto Tecnológico Metropolitano], p. 2,59,60, URL: https://repositorio.itm.edu.co/bitstream/handle/20.500.12622/431/IsazaMonsalveCatherineMaria2018.pdf?sequence=1&isAllowed=y

D. McFadzean, D. Stewart and L. Tesfatsion, (2001), "A computational laboratory for evolutionary trade networks," in IEEE Transactions on Evolutionary Computation, vol. 5, no. 5, pp. 546-560, Oct. 2001, doi: 10.1109/4235.956717. https://ieeexplore.ieee.org/document/956717

H. Rosales Vuelvas y N. Namora Díaz, (2012), “Elaboración de una interfaz gráfica en LabVIEW para la automatización de un sistema de calibración de medidores trifásicos de energía eléctrica activa” [Tesis Licenciatura, Universidad Tecnológica de Bolívar], Repositorio Universidad Tecnológica de Bolívar URL: https://repositorio.utb.edu.co/bitstream/handle/20.500.12585/957/0063154.pdf?sequence=1&isAllowed=y

J. A. Lorenzana, (2022), “Implementación de un Sistema de Visión por Computadora para el Reconocimiento Facial y de Emociones para el Rostro Animatrónico de la Universidad del Valle de Guatemala” Trabajo de graduación para optar al grado académico de Licenciado en Ingeniería Mecatrónica p. 43, 47, URL: https://repositorio.uvg.edu.gt/handle/123456789/4273

J. D. Rodríguez, S. González, (2022), “Plataforma didáctica RBX1 desarrollada en ROS”, Universidad de La Salle Ciencia Unisalle, Bogotá D. C., Colombia, Trabajo de Grado Para Optar al Título de Ingeniero en Automatización p. 18, 40, 48 URL: https://ciencia.lasalle.edu.co/ing_automatizacion/815/

J. Peter Gessler, M. Alcañiz Fillol, (2021), "Sensor para análisis de alimentos aplicación de impedancia espectroscopia y redes neuronales artificiales", Trabajo Fin de Máster presentado en el Departamento de Ingeniería Electrónica de la Universidad Politécnica de València para la obtención del Título de Máster Universitario en Ingeniería de Sistemas Electrónicos, 2021, P3, 60 URL: http://hdl.handle.net/10251/174498

Manuale Degli SCR, (1982), Cap 1, Pag 1-2. Editorial Jackson URL: http://www.energiazero.org/libri_elettronica_80/Manuale_degli_SCR_Vol_1.pdf

MATLAB, (2022), "Creación de apps con interfaces gráficas de usuario en MATLAB" p. 1, Recuperado el 15 de noviembre de 2022, https://es.mathworks.com/discovery/matlab-gui.html

M. Estevan Moreno,(2019)" Diseño e implementación de una interfaz gráfica de usuario en lenguaje Python para el entrenamiento interactivo de redes profundas de segmentación de imagen médica", [Trabajo de grado, Instituto Tecnológico Metropolitano], Grado en Ingeniería Informática, p 3-20-21- 52, URL: https://riunet.upv.es/bitstream/handle/10251/125188/Estevan%20-%20Dise%c3%b1o%20e%20implementaci%c3%b3n%20de%20una%20interfaz%20grafica%20de%20usuario%20en%20lenguaje%20python%20para%20el%20e....pdf?sequence=1&isAllowed=y

Marulanda Ángel, J. P. (2022). Interfaz gráfica de usuario para el manejo de la carga electrónica Serie Bk 8500 usando la herramienta de Matlab. p.22 URL:http://repositorio.itm.edu.co/bitstream/handle/20.500.12622/5735/JuanPablo_MarulandaAngel_2022.pdf?sequence=6&isAllowed=y

Novas Castellano, N., Soler Ortiz, M. J., García Salvador, R. M., Fernández Ros, M. & Gàzquez Parra, J. A. (2019). Electrónica de Potencia. Google Books. Recuperado 2 de diciembre de 2022, de https://books.google.com.mx/books?id=6jl8DwAAQBAJ

Romero Herrera, F. D. (2022). Diseño e implementación de un sistema de control y monitoreo de parámetros eléctricos dentro de procesos industriales mediante software libre y comunicación Modbus. , p.20, 49, 87, URL: http://dspace.espoch.edu.ec/handle/123456789/17704

S. Hernández Sánchez, (2019)" Interfaz gráfica de aplicaciones de control desarrolladas con Python", p. 2,88 URL: https://uvadoc.uva.es/bitstream/handle/10324/37821/TFG-I-1311.pdf?sequence=1&isAllowed=y