

## ***Introduction***

Le mot ElEsim est composé de deux parties :

- ELE: abréviation de "**Electronique**".
- Sim: abréviation de "**Simulation**".

On définir quelques mots liés à celui-ci :

**L'Electronique** L'électronique est une science technique, ou science de l'ingénieur, constituant l'une des branches les plus importantes de la physique appliquée, qui étudie et conçoit les structures effectuant des traitements de signaux électriques, c'est-à-dire de courants ou de tensions électriques, porteurs d'informations.

**L'Electronique de puissance** est la branche de l'Electrotechnique qui a pour objet l'étude de la conversion statique de l'énergie électrique.

**Un convertisseur statique** est un dispositif qui transforme l'énergie électrique disponible en une forme appropriée à l'alimentation d'une charge.

Ce logiciel englobe les quatre types de conversion suivants:

- \_ Conversion: **Alternative** ➡ **Continue**.
- \_ Conversion: **Continue** ➡ **Continue**.
- \_ Conversion: **Continue** ➡ **Alternative**.
- \_ Conversion: **Alternative** ➡ **Alternative**.

### III.1. Vue Utilisateur:

# Vue Utilisateur

## *Introduction:*

Un utilisateur de ElEsim, voir le comme un logiciel de simulation sans intéresser à la conception de celle-ci, ou au programme qui gère le fonctionnement du logiciel, c'est pour ça dans cette section du chapitre III on faire une description externe de ElEsim pour expliquer comment comprendre leurs constitutions, leur fonctionnement, leurs options qui améliorent l'utilisation de celle-ci.

### III.1.1. ELESim 1.0:



*Figure (III.1): Le Logo de ELESim 1.0*

ELESim permet de mieux comprendre le fonctionnement des convertisseurs et de faire leurs dimensionnements.

Les circuits proposés dans ELESim sont disponibles à l'utilisation directe, alors l'utilisateur ne perd pas du temps pour grouper les composants de circuit avant chaque simulation comme dans d'autres logiciels.

Plusieurs combinaisons présentées dans ELESim pour couvrir presque tous les convertisseurs les plus essentiels dans le domaine de l'Electronique de puissance.

Pour chaque type de circuit à simuler, ELESim dispose pas mal des types d'alimentation du circuit et du type de charge a alimenté, ainsi que l'utilisateur preuve changer plusieurs paramètres de circuit à simuler et de faire le dimensionnement de celles-ci.

### III.1.2. Environnement de Travail

Dans cette section on faire une description générale de l'Environnement de ELESim,

Puis on expliquent les étapes a suivre pour faire une simulation dans celui-ci.

### III.1.2.1. Premier contact avec ELEsim1.0

Une fois le logiciel lancé, la capture d'écran suivante apparaît:  
Cette fenêtre comporte les informations sur le logiciel tel que la version, la date de création et l'email de contact.



*Figure (III.2): la capture d'écran qui apparaît  
À l'exécution d'ELEsim*

### III.1.2.2. Parcours rapide de l'environnement:

L'interface principale d'ELESIM est divisée en zones. Chaque zone englobe des catégories bien spécifiques et liées entre elles, alors on définit la constitution de chaque zone:

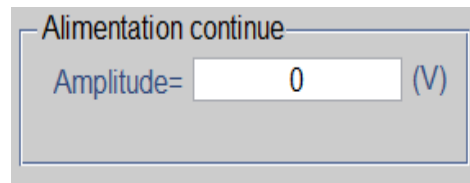
#### III.1.2.2.1. La zone alimentation:

La zone alimentation contient les informations liées à l'alimentation du circuit à simuler:

- Le type d'alimentation: affiché comme titre de la zone.
- Les paramètres d'alimentation: comme l'amplitude, la fréquence, la phase...etc.

Il y a trois types d'alimentation possibles:

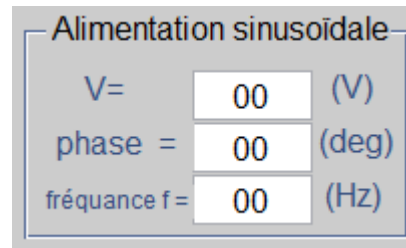
- **Alimentation continue:** (Fig.III.3) Il suffit de définir l'amplitude d'alimentation (en volts).



*Figure (III.3): alimentation continue.*

- **Alimentation sinusoïdale:** (Fig.III.4) On définit:

- \_ Leur amplitude (en volts),
- \_ La fréquence (en Hz),
- \_ La phase initiale (en degrés).



*Figure (III.4): alimentation sinusoïdale.*

#### III.1.2.2.2. La zone Charge:

La zone charge contient les informations liées à la charge de circuit à simuler, et le type de la charge affiché comme titre de la zone.

Il y a quatre types de charge:

- **Charge R:**

Il suffit de définir la résistance R (en Ohm).

- **Charge RL:**

Il faut définir:

- \_ La résistance R (en Ohm),
- \_ L'inductance L (en Henry).

- **Charge RLE(t):**

- \_ La résistance R (en Ohm),
- \_ L'inductance L (en Henry).
- \_ La fem E (en Volts).

- **Charge RLE(t):**
  - \_ La résistance R (en Ohm),
  - \_ L'inductance L (en Henry).
  - \_ La fem E alternative (en Volts),

Elle a trois paramètres:

  - \_ Leur amplitude (en volts),
  - \_ La fréquence (en Hz),
  - \_ La phase initiale (en degrés).

**La charge**

☐ Résistive    R= 00 (Ohm)

☐ Inductive    L= 00 (mH)

☐ LRE    E = 00 (V)

☐ LRE(t)    E(t) = .....

☐ MCC    V(t)ch = 00 (V)

            phase = 00 (Deg)

            fch = 00 (Hz)

**Figure (III.5): LA Charge.**

- **Charge MCC (machine a courant continu):**

Il faut définir:

  - \_ Les paramètres d'induit:
    - \_ La résistance Ri (en Ohm),
    - \_ L'inductance Li (en Henry).

Les paramètres de l'inducteur:

    - \_ La résistance Re (en Ohm),
    - \_ L'inductance Le (en Henry).

Autres paramètres:

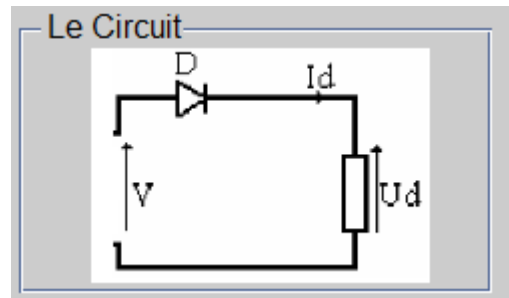
(Tension continue) Ue (en Volts).

    - \_ Le moment d'inertie de L'ensemble Rotor-charge Inertie (en Kg.m<sup>2</sup>).
    - \_ Le couple résistant Cr (en N.m).
    - \_ Le coefficient de frottement fr (en N.m.Rad/Sec).

Pour chaque charge l'utilisateur peut prendre leur propre choix pour une simulation donnée, mais il faut faire attention a des choix qui sont pas admissibles, soit a cause d'illogique valeurs affectés au paramètres d'une charge donnée (par exemple un mauvais choix pour les paramètres de la machine a courant continu), soit a cause des limites de la capacité du logiciel.

### III.1.2.2.3. La zone Circuit:

Dans cette zone, il apparaît le schéma du circuit a simuler (fig.III.07).



**Figure (III.07): Le Circuit.**

Il faut bien connaître que tous les symboles Utilisés dans tout partie de ce logiciel sont unifiés, alors pour par exemple, connaître la signification d'un, symbole d'affichage dans la zone de visualisation il suffit de chercher ce symbole dans la zone de circuit pour connaître leur signification.

On définit la signification de celle-ci un par un et selon le circuit à simuler:

- ☞ V: la tension de la source (Continue, sinusoïdale ou Carrée).
- ☞  $V_1, V_2, V_3$ : les trois tensions d'une source triphasée.
- ☞ I: le courant parcouru dans la source d'alimentation.
- ☞  $I_s$ : le courant parcouru dans la source d'alimentation (autre symbolisation).
- ☞  $U_d$ : la tension aux bornes de la charge.
- ☞  $I_d$ : le courant dans la charge.
- ☞ D: une diode.
- ☞ H: un hacheur.
- ☞  $T_H$ : un thyristor.
- ☞  $D_{RL}$ : une diode de roue libre.
- ☞ K: un interrupteur de l'onduleur.
- ☞  $V_D$ : la tension inverse aux bornes d'une diode D.
- ☞  $I_D$ : le courant dans une diode D.

- ☞  $V_{th}$ : la tension inverse aux bornes d'un thyristor TH.
- ☞  $I_{th}$ : le courant dans un thyristor TH.
- ☞  $V_h$ : la tension inverse aux bornes d'un hacheur H.
- ☞  $I_h$ : le courant dans un hacheur H.
- ☞  $V_{DRL}$ : la tension inverse aux bornes de la diode de roue libre DRL.
- ☞  $I_{DRL}$ : le courant dans la diode de roue libre DRL.
- ☞  $V_k$ : la tension inverse aux bornes d'un interrupteur de l'onduleur.
- ☞  $I_k$ : le courant dans un interrupteur de l'onduleur.
- ☞  $V_g$ : la tension inverse aux bornes d'un gradateur G.
- ☞  $V_L$ : la tension aux bornes de l'inductance L.
- ☞  $V_R$ : la tension aux bornes de la résistance R.
- ☞ E: la fem d'induit dans le cas de la machine à courant continu,

#### III.1.2.2.4. La zone Convertisseur:

La zone Convertisseur contient les paramètres du convertisseur statique spécifique au circuit à simuler.

On distingue:

\_ **Un Redresseur:** Un redresseur à diodes n'a pas besoin de paramètres à définir, par contre un redresseur à thyristors à besoin de définir le rapport cyclique (en %) des thyristors (fig.III.08).

\_ **Un Hacheur:** Pour un hacheur, on définit le rapport cyclique (en %), et la fréquence de hachage (en Hz) (fig. III.09)

\_ **Un Onduleur:** Pour un Onduleur, on définit la fréquence d'ondulation (en Hz) (commande pleine onde) (fig. III.10).

\_ **Un Gradateur:** Pour un Gradateur, on définit l'angle d'amorçage des thyristors montés en tête- bêche (fig. III.08).

**Figure (III.08): Thyristor**

**Figure (III.09): Thyristor**

**Figure (III.10): Onduleur**

#### III.1.2.2.5. La zone Visualisation:

La zone Visualisation (fig. III.11).comporte tous les signaux possibles à afficher dans la zone du graphe. Elle possède des cases à cocher par l'utilisateur pour faire le choix des signaux à afficher dans la zone du graphe.

**Figure (III.11): La zone "Visualisation".**

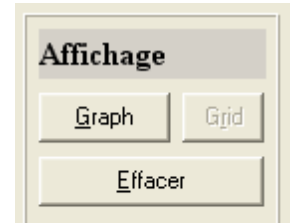
Chaque signal a deux paramètres:

- ✓ La case à cocher pour visualiser le signal spécifique dans la zone du graphe.
- ✓ Le symbole significatif du signal: c'est le même symbole utilisé dans toutes les parties du logiciel pour désigner le même signal, alors pour connaître la signification de ce symbole il suffit de voir le schéma du circuit dans la zone de circuit.

### III.1.2.2.6. La zone Affichage:

Cette zone avec la zone de visualisation présente le tableau de commande de la zone du graphe. Cette zone comporte alors (fig.III.17):

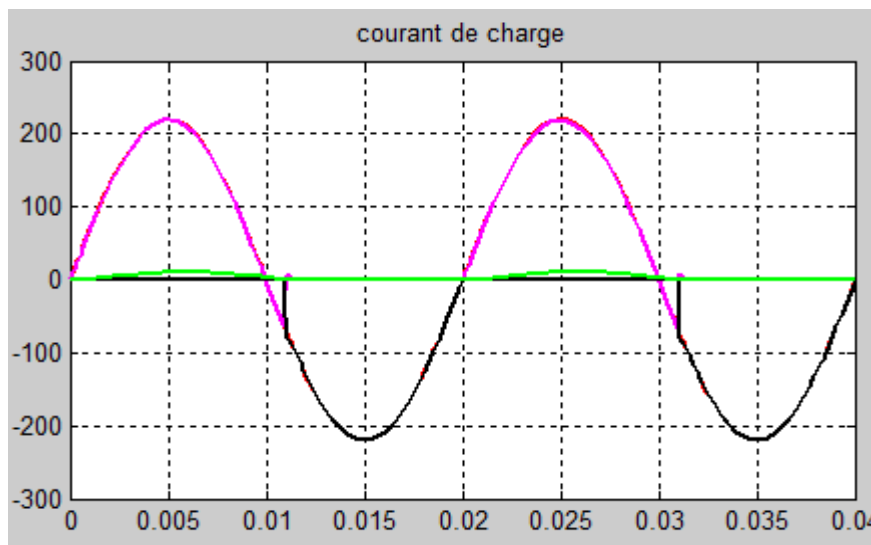
- ☞ Un bouton **Graphe**: pour lancer la Visualisation du graphe dans la zone appropriée.
- ☞ Un bouton **Grid off**: Balayez votre la grille dans la zone du graphe "grid en Anglais".
- ☞ Un bouton **Effacer**: pour effacer le dessin dans la zone du graphe



*Figure (III.12) : La zone "affichage".*

### III.1.2.2.7. La zone Graphe:

C'est la zone d'affichage des courbes à visualiser, on remarque que cette zone ne comporte pas une échelle au contraire de plus part des simulateurs, tout simplement a cause que chaque signal à son échelle spécifique présenté dans la zone de visualisation.



*Figure (III.13): La lecture de l'échelle.*

### III.1.2.2.8. La zone Calculs:

C'est la zone dans laquelle il y a l'affichage des grandeurs liées aux signaux affichés dans la zone du graphe (fig.III.14). Ces grandeurs sont:

The screenshot shows a window titled "calculs" with the following fields:

|         |   |     |     |   |       |
|---------|---|-----|-----|---|-------|
| Ud moy= | 0 | (V) | P=  | 0 | (W)   |
| I moy=  | 0 | (A) | Q=  | 0 | (VAR) |
| Udeff=  | 0 | (V) | S=  | 0 | (VA)  |
| Id eff= | 0 | (A) | Fp= | 0 |       |

A "valider" button is located to the right of the input fields.

**Figure (III.14): La zone de calcul à l'état d'affichage.**

$U_{dmoy}$ : la valeur moyenne de la tension aux bornes de la charge.

$I_{dmoy}$ : la valeur moyenne du courant dans la charge.

$U_{deff}$ : la valeur efficace de la tension aux bornes de la charge.

$I_{deff}$ : la valeur efficace du courant dans la charge.

$P$ : la puissance active dans la source.

$Q$ : la puissance réactive dans la source.

$S$ : la puissance apparente dans la source.

$F_p$ : le facteur de puissance de la source.

Les formules utilisées pour le calcul des puissances sont:

- La puissance active:

$$P = \frac{1}{2\pi} \int_0^{2\pi} I_s V d\omega t$$

- La puissance réactive:

$$Q = qV I \sin\phi_1$$

- La puissance apparente:

$$S = qv_{eff} I_{eff}$$

- La puissance déformante:

$$D = \sqrt{S^2 - P^2 - Q^2}$$

$V$ : la valeur max de la tension d'alimentation.

$I$ : la valeur max du courant parcourir dans la source.

$V_1$ : la valeur max du fondamental de la tension d'alimentation.

$I_1$ : la valeur max du fondamental du courant parcouru dans la source.

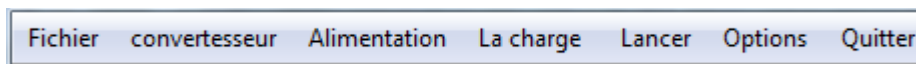
$V_{eff}$ : la valeur efficace de la tension d'alimentation.

$I_{eff}$ : la valeur efficace du courant parcouru dans la source.

$q$ : nombre de phases (1:monophasé, 3:triphase).

### III.1.3. Les Etapes de Simulation jusqu'à l'Obtention des résultats:

Pour réaliser une simulation donnée, il faut suivre les étapes suivantes:



**Figure (III.15): La barre des menus.**

Dans la barre des menus (Fig.III.15), l'utilisateur au début de la simulation doit faire son choix de convertisseur (1), puis d'alimentation (2), puis de la charge (3), puis lancer la préparation de l'interface correspondant a leur choix (4), le logiciel pour obliger l'utilisateur de suivre ce ordre désactive les menus suivants jusqu'à que l'utilisateur fasse son choix dans l'étape précédente.

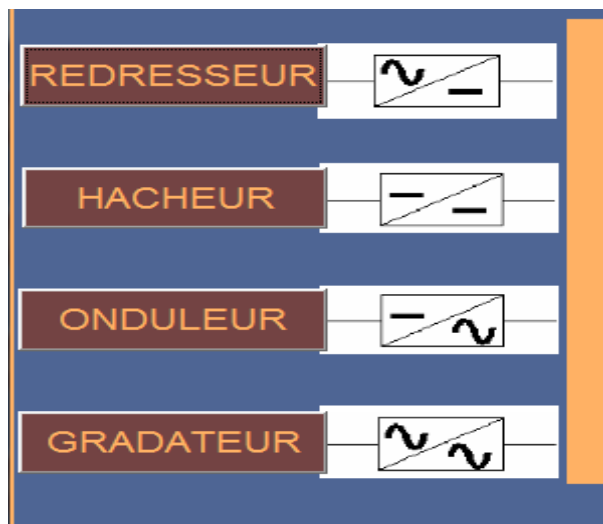
Cliquez sur le menu convertisseur, une fenêtre s'affiche (Fig.III.16), Alors faire le choix du convertisseur, il y a quatre types :

1-Redresseur.

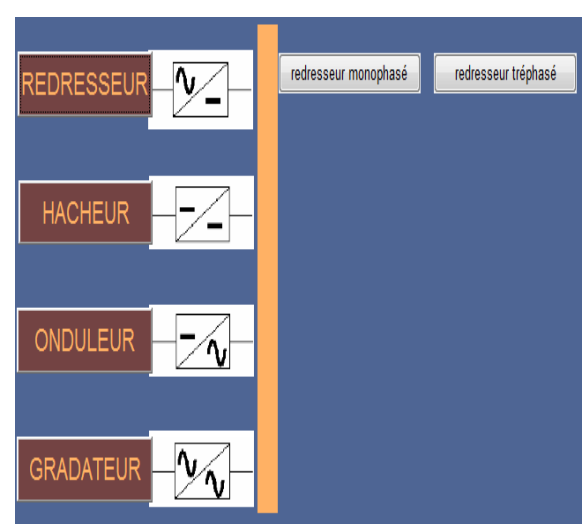
2-Hacheur.

3-Onduleur.

4-Gradateur.



**Figure (III.16): choix de type de convertisseur.**



**Figure (III.17): choix de sous type de convertisseur.**

Chaque convertisseur englobe sous-types affichés lors du choix de celui-ci (Fig.III.17).

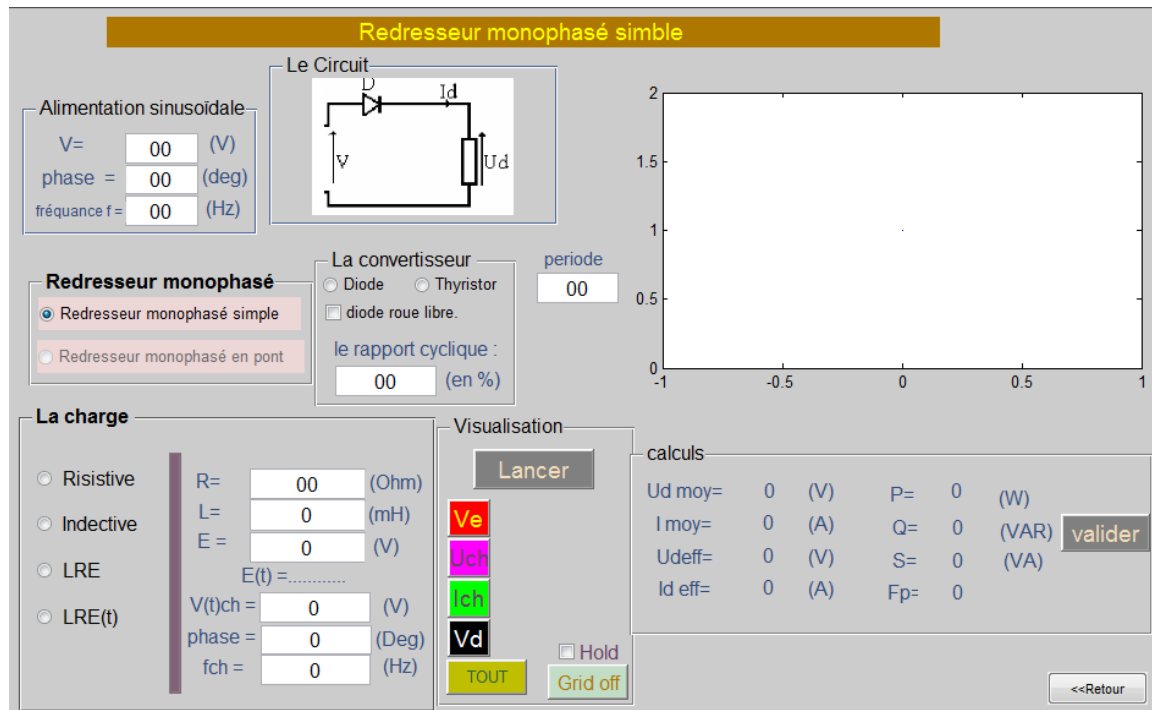
Un choix d'un des sous types provoque la fermeture automatique de cette fenêtre.

Cliquez sur le menu "La charge", une fenêtre s'affiche (Fig.III.18), Alors faire le choix de la charge du circuit à simuler,

Il y a Cinq types de charge:

- \_ Charge R (résistance).
- \_ Charge RL (Résistance- inductance).
- \_ Charge RLE (Résistance- inductance- force électromotrice
- \_ Charge RLE (t) (Résistance- inductance- fem sinusoïdale).
- \_ Charge MCC (Machine à courant continu).

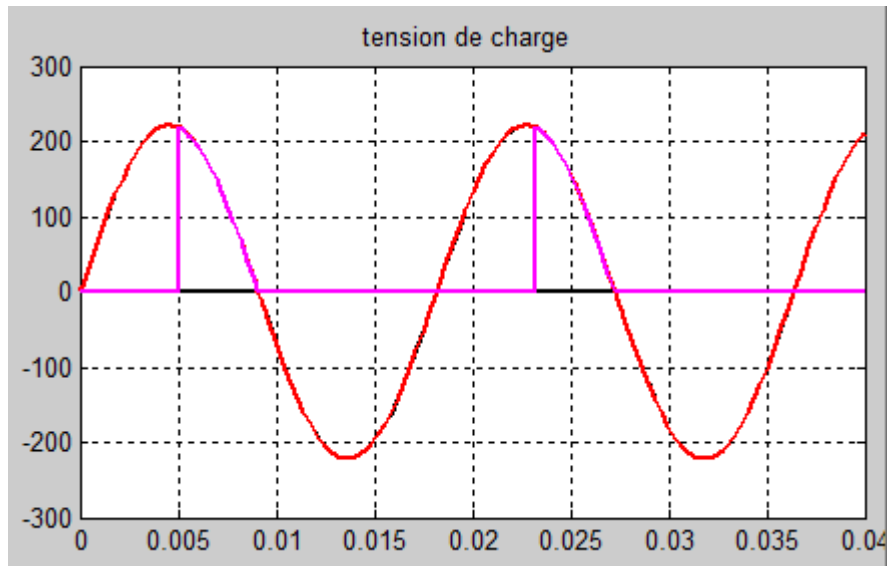
Cliquez sur le menu "Lancer", pour lancer la présentation des choix principaux de simulation définis par l'utilisateur, alors l'interface comme apparaît dans (fig.III.18) devient comme dans la figure:



**Figure (III.19): L'interface après avoir cliqué sur le bouton "Lancer".**

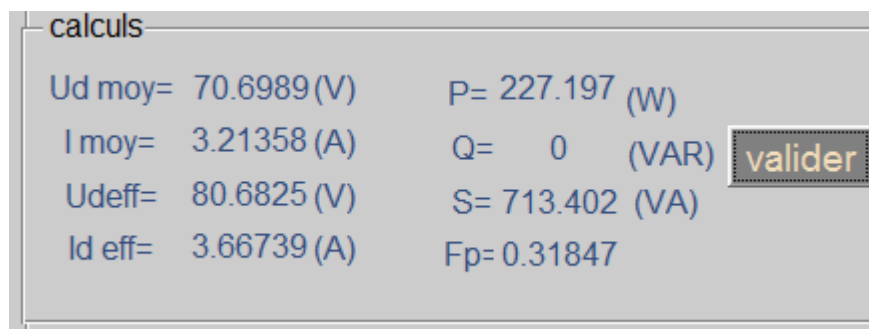
Au premier lieu les choix par défaut vous permettent de simuler un type de circuit donné, alors il suffit de cocher la case à gauche de signal désiré dans la zone

"Visualisation", puis cliquez sur le bouton Graphe dans la zone "affichage" pour afficher les signaux choisis dans la zone du graphe (figure.III.20).



**Figure (III.20): Affichage des signaux dans la zone du graphe.**

Cliquez sur le bouton "Valider" dans la zone "Calculs" pour voir les différents grandeurs liés a la source (P, Q, D, S, Fp) et la charge (Udmoy, Idmoy, Udeff, Ideff)



**Figure (III.21): Affichage des résultats de calcul des grandeurs.**

### ***III.2. Vue programmeur:***

#### ***Introduction:***

Dans cette partie, on va présenter la structure globale du logiciel d'une vue "Simulation et programmation".

Le programme principal constitué d'un ensemble de procédures et de simulation en simulink l'une appelle à l'autre. Chaque procédure a des variables externes pour l'échange avec l'extérieur et autres internes pour des traitements à l'intérieur de celle-ci.

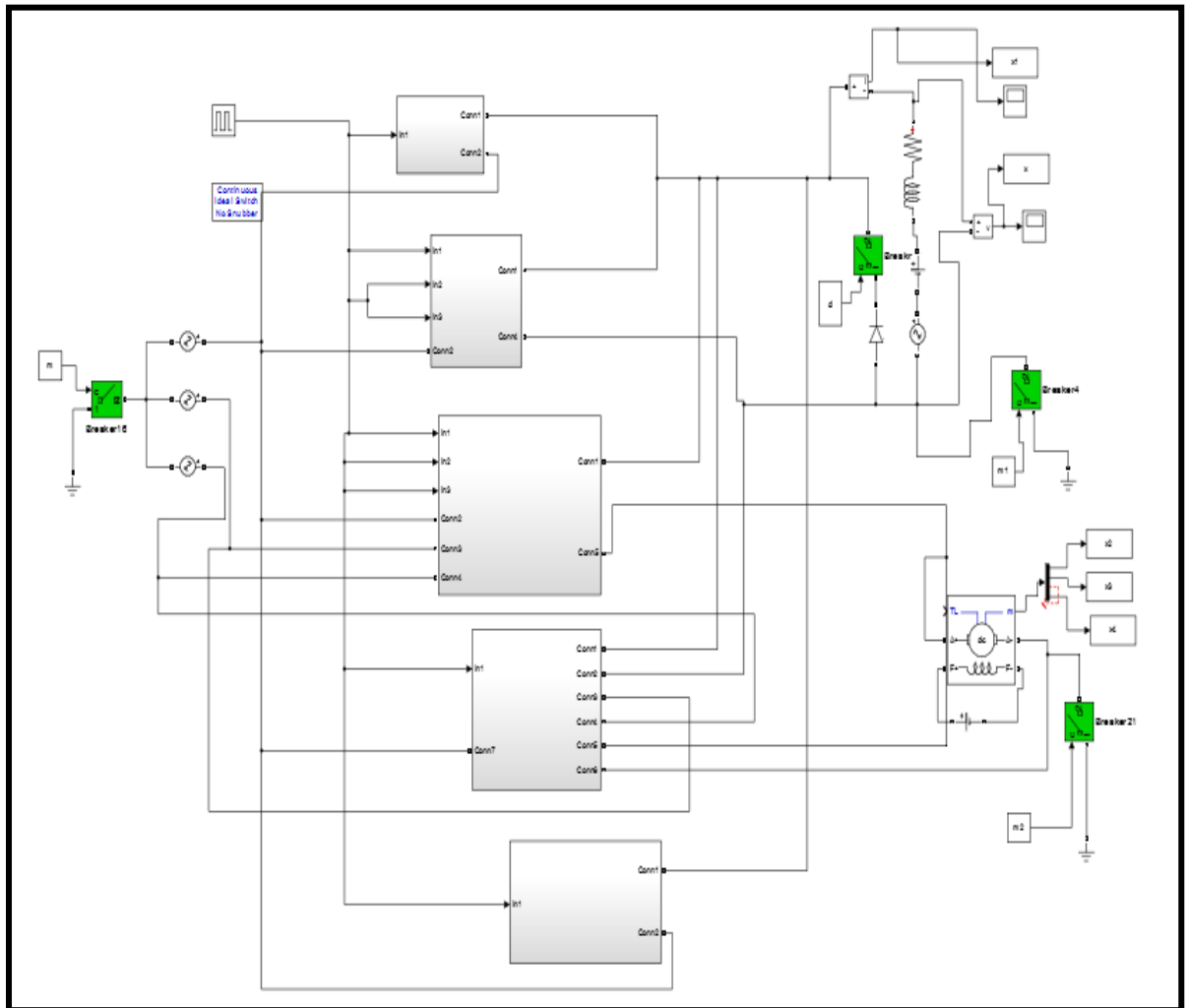
Ces procédures sont portées dans des unités. Chaque unité a une forme (interface) pour présenter les différents résultats demandés par l'utilisateur lors de l'exécution.

On va d'abord présenter les procédures et les fonctions publiques, puis on va présenter la grille on liant entre les différentes procédures dans la section "***La structure globale du programme***".

#### **III. 2.1. Les procédures et les fonctions publiques:**

Les procédures et les fonctions publiques (écrites par le programmeur) sont des procédures utilisables par n'importe quel programme dans l'unité qui contient la procédure ou dans les autres unités utilisant cette unité. On va citer les plus importantes.

### III. 2.1.1. La procédure block les convertisseurs statiques en simulink:



*Figure (III.22): block les convertisseurs statiques en simulink.*

### III. 2.1.2. Création du fichier m correspondant :

Une fois les différents composants de l'interface graphique mis en place, il faut les faire interagir, c'est-à-dire, programmer l'interface graphique. GUIDE permet la génération automatique d'un fichier : m correspondant a un fichier : fig. Celui-ci contient notamment une fonction s'exécutant au lancement du programme et les appels aux différent composants.

### III. 2.1.3. La procédure Menus :

Un menu est un titre complété par une liste de sous-menu. Les actions (callbacks) sont généralement lancés à partir des sous-menus. L'ajout de menus spécifique se fait par :

```
menu1 = uimenu( fig1 , 'label' , ' Statist.' );
```

Un sous-menu est un élément du menu principal, donc de l'entité père. Il est donc déclaré car menu du menu principal.

```
smenu1 = uimenu( menu1 , 'label' , 'Normale' , 'callback' , 'methode_normale' )  
smenu2 = uimenu( menu1 , 'label' , 'Binomiale' , 'callback' , 'methode_binomiale');
```

### III. 2.1.4. La procédure Ascenseur ou slider :

L'ascenseur a pour objectif de fixer la valeur d'un paramètre entre deux bornes fixées. La valeur prise par la variable est représentée par la position du curseur .

```
slid1=uicontrol(fig1,'style','slider','position',[100,50,150,20] , 'Min' , -1 , 'Max' , 1 , ...  
'callback' , 'val_variable = get(slid1 , "value" ) ' );
```

### III. 2.1.5. La procédure Texte Editable

Permet à l'utilisateur de saisir une valeur. C'est une fonction importante.

```
Text1 = uicontrol ( fig1 , 'style' , ' edit' , 'position', [100,50,150,20] , 'Max' , 1 , 'string' , '0' );
```

Généralement, Il faut associer un texte fixe pour préciser le rôle de la fenêtre de saisie à l'utilisateur. Exemple : le texte est placé à gauche de la fenêtre de saisie

```
uicontrol ( fig1 , 'style' , ' texte' , 'position', [10,50,90,20] , 'string' , 'Valeur A' );
```

### III. 2.1.6. La procédure Liste de choix

La liste de choix ou pop-up menu permet de sélectionner une valeur parmi une liste.

Généralement, cette valeur est un texte. La valeur retournée lors du choix (paramètre 'Value') est le numéro de ligne du choix.

```
choix1 = uicontrol ( gcf , 'Style' , 'popup' , 'String' , 'Pierre|Paul|Jacques' , 'Position'  
    , [10 10 100 80] );
```

La liste du texte est modifiable après la création de la fenêtre de choix, en modifiant la propriété 'String'.

### III. 2.1.7. La procédure Bouton Radio

Le bouton Radio permet de fixer un paramètre binaire (0 ou 1), représentant souvent un choix ou une option dans une application.

```
fig1 = figure ;  
radio1 = uicontrol( fig1 , 'style' , 'Radio' , 'Position' , [ 30 20 130 25 ] , 'String' , ' OptionA ' );  
radio2 = uicontrol( fig1 , 'style' , 'Radio' , 'Position' , [ 30 45 130 25 ] , 'String' , ' OptionB ' );  
uicontrol( fig1 , 'style' , 'Text' , 'Position' , [ 30 70 130 30 ] , 'String' , ...  
    ' Choix des Options ' , 'FontWeight' , 'bold' );
```

### III. 2.1.8. La procédure Cadre

Le cadre permet de dessiner un rectangle de présentation (par exemple regroupement de diverses entités graphiques dans un rectangle).

Le cadre se déclare par :

```
cadre1 = uicontrol ( fig1 , 'style' , 'frame' , 'position' , [ posX,posY,tailleX,tailleY])
```

### III. 2.1.9. La procédure Graphiques

Les graphiques se dessinent dans une partie de la fenêtre définie par la fonction 'subplot', dont les paramètres sont différents de l'emploi classique (division de la fenêtre en sous-fenêtres de taille égale)

```
subplot( 'Position' , [ Xpos Ypos Xtaille Ytaille])
```

### III. 2.1.10. La procédure plot:

Ecrire le code suivant dans la fonction Plot Callback:

```
axes(handles.axes1)  
plot(t,f(t))
```

### III. 2.1.11. La procédure grid off:

La procédure " **grid off** " dessine la grille dans la zone de graphe. Le nombre des lignes verticales et horizontales :

```
Grid on ;
```

### III. 2.1.13. La procédure hold on:

```
j=get(hObject,'Value');
if j==1
    hold on;
else
    hold off;
end
```

### III.2.1.14. La procédure Appel à simulink et Commencez le processus de simulation:

```
find_system('name','R_mo_sim_d_r');
open_system('R_mo_sim_d_r');
set_param(gcs,'simulationcommand','start');
```

### III. 2.1.15. La procédure Insérer les variables à partir de interface à simulink

```
function pushbutton23_Callback(hObject, eventdata, handles)
global ve fe p r l vc vc1 T f1 p1 ta at m V Ud Id Vd o j n
set(handles.pushbutton10,{ 'enable'}, { 'on'});
ve=str2num(get(handles.edit1,'string'));
fe=str2num(get(handles.edit5,'string'));
p=-1*str2num(get(handles.edit3,'string'));
r=str2num(get(handles.edit7,'string'));
l=str2num(get(handles.edit8,'string'))*0.001;
vc=str2num(get(handles.edit9,'string'));
vc1=str2num(get(handles.edit10,'string'));
f1=str2num(get(handles.edit14,'string'));
p1=str2num(get(handles.edit13,'string'));
T=str2num(get(handles.edit15,'string'));
ta=1/(fe*2);
m=str2num(get(handles.edit6,'string'));
at=(m*((1/(2*fe))))/100
find_system('name','red_pont');
open_system('red_pont');
set_param(gcs,'simulationcommand','start')
```

### III. 2.2. La structure globale du programme:

L'organigramme de la (fig.III.44), nous donne une idée sur La structure du programme.

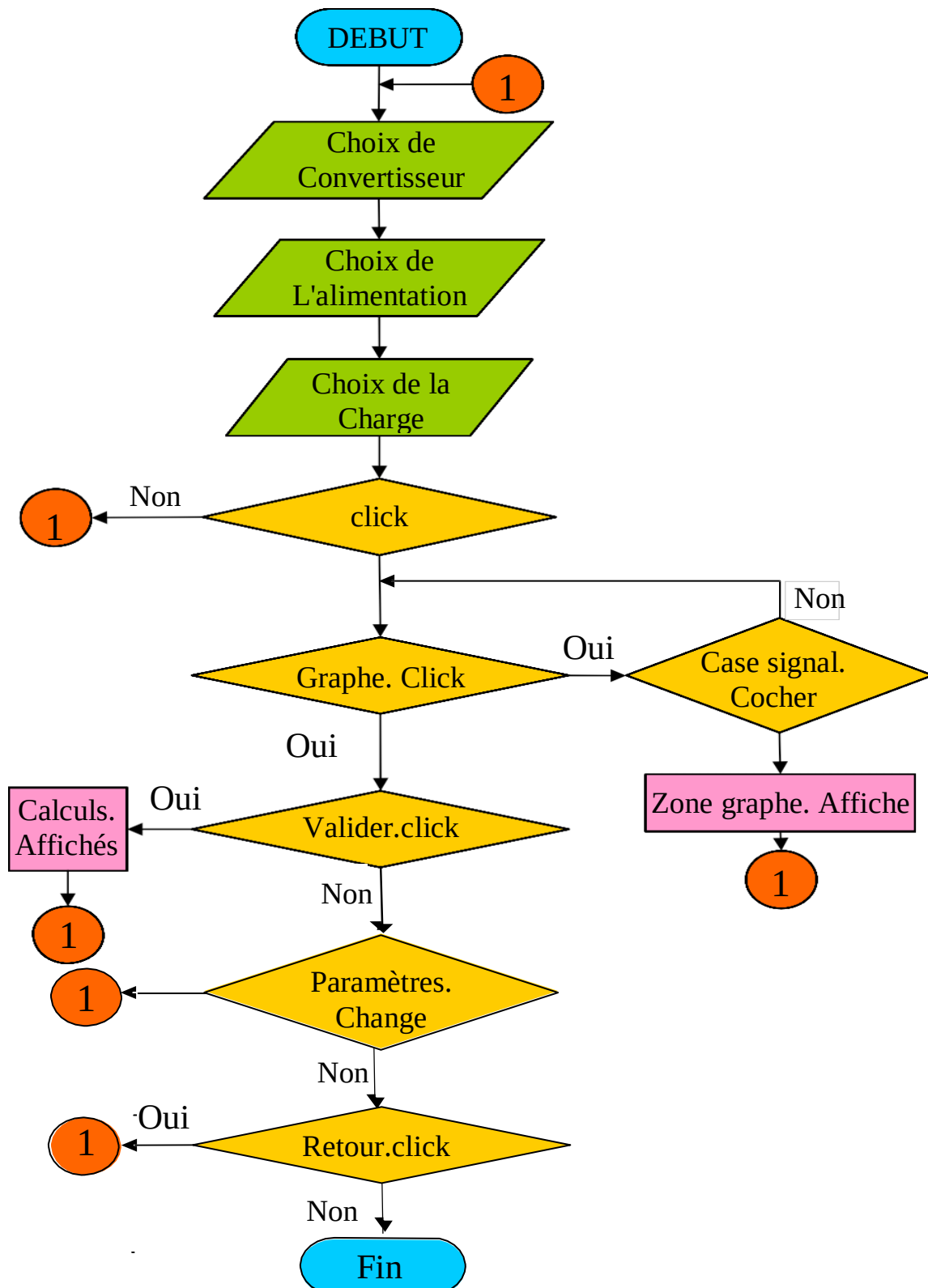


Figure (III.44): L'organigramme du fonctionnement du logiciel.

### III. 2.2.1. Commentaires sur l'organigramme:

Le début et la fin sont respectivement, l'exécution et la fermeture du logiciel. A l'exécution du logiciel, on va toujours faire le choix de trois choses dans l'ordre décrit on lance la validation du choix par le bouton menu "Lancer" (Lancer click: c'est l'événement qui simule un clic sur le bouton Lancer). Si on ne click pas sur ce bouton, on peut revenir au début pour changer les choix précédents (1: c'est un lien vers le début du programme). Graphe. Click: simule un clic sur le bouton Graphe qui lance une procédure de calcul pour calculer les signaux sélectionnés pour l'affichage de sont allures dans la zone de graphe, mais l'affichage dans cette zone n'aura lieu que dans le cas où une ou plusieurs cases de sélection des signaux est coché sinon le programme revient à l'état d'attente avant l'événement "Graphe. Click", l'événement "Valider click" simule un click sur le bouton Valider dans la zone de calcul pour lancer la procédure de calcul de  $U_{dmoy}$ ,  $I_{dmoy}$ ,  $U_{deff}$ ,...etc, puis affiche les résultats dans cette zone, et le programme revient a l'état d'attente de nouveau, un click sur le bouton "Retour" dans la même fenêtre dans ce cas on revient de nouveau à la fenêtre principale du logiciel, à ce point le programme attend n'importe quelle changement effectué par l'utilisateur et revient à a chaque changement au début pour refaire l'exécution avec les nouveaux paramètres.