

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique
Département de Génie électrique
Option

ELECTROTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Projet de fin d'étude pour l'obtention du diplôme d'ingénieur d'état en
ELECTROTECHNIQUE



***Réalisation d'un logiciel de simulation des
convertisseurs d'électronique de puissance***

Etudié et réalisé par:
Mr. BENDEMMAGH Walid
Mr. MANSAR Zohir

Membres du jury:

E.M BERKOUK	Professeur	Rapporteur
B.HEMICI	Chargé de cours	Président du jury
R.TAHMI	Maître de conférence	Examineur

Promotion: juin 2007

ENP 10, Avenue Hassen-Badi, EL-HARRACH, ALGER.

Remerciement

Nous remercions dieu de nous avoir donné la force pour accomplir ce travail.

*Nous remercions notre encadreur
Pr: E.M. BERKOUK pour son aide, son encouragement et sa patience.*

Et nous n'oublions pas tous les enseignants de département génie électrique.

Dédicace

Je dédie ce travail,

*A mes très chers parents qui ont sacrifié leur vie pour
ma réussite, et pour leurs encouragements. Que dieu
les garde éternellement heureux.*

A toute la famille BENDEMMAGH,

*A tout mes amis et mes collègues, et à tous ceux qui
m'ont aidés pour réaliser ce mémoire.*

BENDEMMAGH Walid

Dédicace

Je dédie ce travail,

A mes très chers parents qui ont sacrifié leur vie pour ma réussite, et pour leurs encouragements. Que dieu les garde éternellement heureux.

A toute la famille MANSAR,

A tout mes amis et mes collègues, et à tous ceux qui m'ont aidés pour réaliser ce mémoire.

بسم الله الرحمن الرحيم

ملخص:

يهدف هذا العمل و المتمثل في برمجة جميع أنواع المحولات الساكنة و المتمثلة في:

- المقومات,
- المقطعات,
- المموجات,
- المدرجات.

يمكن لمستخدم هذا البرنامج معرفة شكل توتر و تيار دارة معينة, حيث يقوم باختيار نوع المحول, نوع التغذية (مستمرة, مربعة, أو جيبية), و كذلك نوع الحمولة (مقاومة, مقاومة+ذاتية, مقاومة+ذاتية+قوة محرك كهربائية, أو محرك ذو تيار مستمر).

بعد هذه الاختيارات ما يبقى على المستخدم سوى إدخال مميزات التغذية من ذروة, اهتزاز, و طور ابتدائي.

يقوم هذا البرنامج كذلك بحساب الإستطاعات و معامل الاستطاعة, و القيم المتوسطة و المنتجة للتوتر و التيار الكهربائي.

كلمات مفتاحية: دلفي, مقومات, مقطعات, مموجات, مدرجات, و التصنع.

Résumé:

Le but de ce mémoire est de programmer les quatre types de convertisseurs d'électronique de puissance.

- Les redresseurs,
- Les hacheurs,
- Les onduleurs,
- Les gradateurs.

L'utilisateur peut visualiser les différents signaux d'un circuit, il suffit de choisir le type de convertisseur, le type d'alimentation, et le type de charge, après il faut remplir les cases des paramètres des circuits tel que l'amplitude, le déphasage, et la fréquence du signal.

Le logiciel peut calculer les différents types de puissances, le facteur de puissance, et les valeurs moyennes et efficaces des tensions et du courant électrique.

Mots clés: Delphi, Redresseurs, hacheurs, onduleurs, gradateurs, et simulation.

Abstract:

The goal of this memory is to program the four types of converters of electronics of power.

- The rectifiers,
- The choppers ,
- The inverters ,
- The gradators,

The user can visualize the various signals of a circuit; it is enough to choose the type of converter, the type of food, and the type of load, after it is necessary to fill breakages of the parameters of the circuits such as the amplitude, dephasing, and the frequency of the signal.

The software can calculate the various types of powers, the power-factor, and the average and effective values of the tensions and the electrical current.

Keys words: Delphi, rectifiers, choppers, inverters, gradators, simulation.

Table des matières

CHAPITRE I: ENVIRONNEMENT DE TRAVAIL DE DELPHI	- 2 -
I.1. INTERFACE DE DELPHI:	- 3 -
I.1.1. LA PREMIERE FENETRE:	- 4 -
I.1.2. LA DEUXIEME FENETRE:	- 4 -
I.1.3. LA TROISIEME FENETRE:	- 5 -
I.1.4. LA QUATRIEME FENETRE:	- 5 -
I.2. LES COMPOSANTS UTILISEES:	- 6 -
I.3. LES INSTRUCTIONS:	- 10 -
I.3.1. L' INSTRUCTION "IF"	- 10 -
I.3.2. L' INSTRUCTION "WHILE"	- 10 -
I.3.3. L' INSTRUCTION "FOR"	- 10 -
I.3.4. L' INSTRUCTION "PROCEDURE"	- 10 -
I.3.5. L' INSTRUCTION "FUNCTION"	- 11 -
I.4. LA CONVERSION:	- 11 -
CHAPITRE II: MODELISATION DES CONVERTISSEURS STATIQUES	
II.1. LES REDRESSEURS:	- 13 -
II. 1.1. TENSION REDRESSEE:	- 14 -
II. 1.2. VALEUR MOYENNE DE LA TENSION REDRESSEE	-14-
II. 1.3. LE DEVELOPPEMENT EN SERIE DE FOURIER:	- 14 -
II. 1.4. TENSION INVERSE:	- 15 -
II. 1.5. COURANTS ET PUISSANCES:	- 15 -
II. 1.6. FACTEUR DE PUISSANCE	- 16 -
II. 1.7. LES MONTAGES USUELS:	- 16 -
II. 1.7.1. REDRESSEUR MONOPHASE SIMPLE:	- 16 -
II. 1.7.1.1. MONTAGE PRATIQUE	- 16 -
II. 1.7.1.2. TENSION REDRESSEE	- 16 -
II. 1.7.1.3. VALEUR MOYENNE DE LA TENSION REDRESSEE	- 17 -
II. 1.7.1.4. TENSION INVERSE AUX BORNES DE LA DIODE (THYRISTOR)	- 17 -
II. 1.7.1.5. RESEAU DE PETRI:	- 17 -
II. 1.7.2. REDRESSEMENT MONOPHASE EN PONT:	- 18 -
II.1.7.2.1. MONTAGE PRATIQUE	- 18 -
II. 1.7.2.2. TENSION REDRESSEE	- 18 -
II. 1.7.2.3. VALEUR MOYENNE DE LA TENSION REDRESSEE	- 18 -
II. 1.7.2.4. TENSION INVERSE AUX BORNES DE LA DIODE D_1 (THYRISTOR TH_1) ..	- 19 -
II. 1.7.2.5. RESEAU DE PETRI:	- 19 -
II. 1.7.3. REDRESSEMENT TRIPHASE PARALLELE SIMPLE:	- 20 -
II.1.7.3.1. MONTAGE PRATIQUE	- 20 -
II. 1.7.3.2. TENSION REDRESSEE	- 20 -
II. 1.7.3.3. VALEUR MOYENNE DE LA TENSION REDRESSEE	- 21 -

Table des matières

II. 1.7.3.4. TENSION INVERSE AUX BORNES DE LA DIODE D_1 (TH_1)	- 21 -
II. 1.7.3.5. RESEAU DE PETRI:	- 21 -
II.2. LES HACHEURS:	- 22 -
II. 2.1. HACHEUR SERIE (DEVOLTEUR):.....	- 23 -
II. 2.1.1. MONTAGE PRATIQUE.....	- 23 -
II. 2.1.2. LA VALEUR MOYENNE DE LA TENSION DE SORTIE:	- 24 -
II.2.1.3. LA VALEUR MOYENNE DU COURANT:.....	- 24 -
II. 2.1.4. ONDULATION DU COURANT:	- 24 -
II. 2.1.5. RESEAU DE PETRI :.....	- 24 -
II. 2.1.6. TENSION AUX BORNES D'UNE MACHINE A COURANT CONTINU:	- 25 -
II. 2.2. HACHEUR PARALLELE (SURVOLTEUR):.....	- 26 -
II.2.2.1. MONTAGE PRATIQUE	- 26 -
II. 2.2.2. LA VALEUR MOYENNE DE LA TENSION DE SORTIE:	- 26 -
II.3. LES ONDULEURS:	- 27 -
II.3.1. ONDULEUR EN DEMI- PONT:	- 28 -
II. 3.1.1. MONTAGE PRATIQUE.....	- 28 -
II. 3.1.2. TENSION AUX BORNES DE LA CHARGE	- 28 -
II. 3.2. ONDULEUR EN PONT:.....	- 29 -
II. 3.2.1. MONTAGE PRATIQUE.....	- 29 -
II. 3.2.2. TENSION AUX BORNES DE LA CHARGE	- 29 -
II. 3.3. LE DEVELOPPEMENT EN SERIE DE FOURIER:.....	- 29 -
II. 3.4. ONDULEUR EN TENSION TRIPHASE:	- 30 -
II. 3.4.1. MONTAGE PRATIQUE.....	- 30 -
II.3.4.2. TENSION AUX BORNES DE LA CHARGE	- 30 -
II.4. LES GRADATEURS:.....	- 31 -
II. 4.1. GRADATEUR MONOPHASE.....	- 32 -
II.4.1.1. MONTAGE PRATIQUE	- 32 -
II. 4.1.2. TENSION AUX BORNES DE LA CHARGE	- 32 -
CHAPITRE III: LE LOGICIEL CONÇU	
III.1. VUE UTILISATEUR:	- 34 -
III.1.1. ELTSIM 1.0:.....	- 35 -
III.1.2. ENVIRONNEMENT DE TRAVAIL	- 35 -
III.1.2.1. PREMIER CONTACT AVEC ELTSIM 1.0.....	- 35 -
III.1.2.2. PARCOURS RAPIDE DE L'ENVIRONNEMENT:	- 36 -
III.1.2.2.1. LA ZONE ALIMENTATION:	- 36 -
III.1.2.2.2. LA ZONE CHARGE:	- 37 -
III.1.2.2.3. LA ZONE CIRCUIT:	- 38 -
III.1.2.2.4. LA ZONE CONVERTISSEUR:	- 40 -
III.1.2.2.5. LA ZONE VISUALISATION:.....	- 40 -
III.1.2.2.6. LA ZONE AFFICHAGE:	- 42 -
III.1.2.2.7. LA ZONE GRAPHE:	- 42 -

Table des matières

III.1.2.2.8. LA ZONE CALCULS:	- 43 -
III.1.3. LES ETAPES DE SIMULATION JUSQU'A L'OBTENTION DES RESULTATS:	- 44 -
III.1.4. LES OPTIONS DE ELTSIM 1.0	- 47 -
III.2. VUE PROGRAMMATEUR:	- 34 -
III. 2.1. LES PROCEDURES ET LES FONCTIONS PUBLIQUES:	- 51 -
III. 2.1.1. LA PROCEDURE RK4:	- 51 -
III. 2.1.2. LA PROCEDURE RK4MCC:	- 51 -
III. 2.1.3. LA PROCEDURE INTEGRAL:	- 52 -
III. 2.1.4. LA PROCEDURE FOURIER5:.....	- 53 -
III. 2.1.5. LA PROCEDURE MAXN:	- 55 -
III. 2.1.6. LA FONCTION IMP:.....	- 55 -
III. 2.1.7. LA FONCTION V (T):	- 56 -
III. 2.1.8. LA PROCEDURE PLOT:	- 57 -
III. 2.1.9. LA PROCEDURE IZONE:.....	- 57 -
III. 2.1.10. LA PROCEDURE GRID:.....	- 58 -
III. 2.2. LA STRUCTURE GLOBALE DU PROGRAMME:.....	- 59 -
III. 2.2.1. COMMENTAIRES SUR L'ORGANIGRAMME:	- 61 -
CHAPITRE IV: VALIDATION DES RESULTATS
IV.1. REDRESSEMENT MONOPHASE SIMPLE A DIODE	- 64 -
IV.2. INTERPRITATION DES RESULTATS:	- 67 -
IV.3. REDRESSEMENT MONOPHASE SIMPLE A THYRISTOR.	- 68 -
IV.4. INTERPRITATION DES RESULTATS:	- 71 -
IV.5. REDRESSEMENT MONOPHASE EN PONT A DIODES	- 72 -
IV.6. INTERPRETATION DES RESULTATS:	- 75 -
IV.7. REDRESSEMENT MONOPHASE EN PONT A THYRISTORS.....	- 76 -
IV.8. INTERPRETATION DES RESULTATS:	- 79 -
IV.9. REDRESSEMENT TRIPHASE PARALLELE SIMPLE A DIODES	- 80 -
IV.10. INTERPRETATION DES RESULTATS:.....	- 83 -
IV.11. REDRESSEMENT TRIPHASE PARALLELE SIMPLE A THYRISTORS.....	- 84 -
IV.12. HACHEUR SERIE:.....	- 87 -
IV.13. ONDULEUR EN DEMI- PONT:	- 89 -
IV.14. ONDULEUR MONOPHASE EN PONT:	- 91 -
IV.15. ONDULEUR TRIPHASE:.....	- 93 -
IV.16. GRADATEUR MONOPHASE:.....	- 94 -
IV.17. MACHINE A COURANT CONTINU:	- 95 -
IV.18. INTERPRETATION DES RESULTATS:.....	- 96 -
IV.19. CALCULS DES PUISSANCES:.....	- 97 -
IV.20. INTERPRETATION DES RESULTATS DES CALCULS	- 100 -

Table des matières

V. ANNEXE:	
V.1. AUTRES COMPOSANTS DE DELPHI:	- 1 -
V.2. QUELQUES RELATIONS:.....	- 2 -
V.3. MOTEUR A COURANT CONTINU (EXCITATION SEPARÉE):	- 3 -
V. 3.1. DESCRIPTION, ROLE DES ELEMENTS:	- 3 -
V.3.2. SCHEMA EQUIVALANT	- 3 -
V.3.3. EXPRESSION DU COUPLE.....	- 4 -
V. 3.4. POINT DE FONCTIONNEMENT:.....	- 4 -
V. 4. DECOMPOSITION EN SERIE DE FOURIER:	- 5 -
V. 5. INTEGRATION NUMERIQUE:	- 5 -
V. 6. RESOLUTION NUMERIQUE D'EQUATIONS (METHODE DE RUNGE- KUTTA):	- 6 -
V. 7. THD:	- 7 -
V.8. VALIDATION DES RESULTATS POUR LES HARMONIQUES.....	-8-
BIBLIOGRAPHIE	

INTRODUCTION GENERALE

La programmation et la simulation ont une grande importance dans tout les domaines, en électronique de puissance lorsque il s'agit de connaître le comportement des tensions et des courants sans faire les montages et les essais pratiques, ainsi que le gain du temps. L'objet de ce mémoire est de développer un logiciel de simulation des convertisseurs d'électronique de puissance en utilisant le langage Delphi.

Dans le premier chapitre, nous présentons l'environnement du travail de Delphi. L'interface de Delphi, les fenêtres, les composants, et les instructions utilisées sont ainsi représentées.

Dans le deuxième chapitre, nous rappellerons le fonctionnement et le comportement des tensions et des courants des différents types de convertisseurs statiques pour mieux les programmer. Cette partie comporte les redresseurs (commandés et non commandés), les hacheurs, les onduleurs, et les gradateurs.

Dans le troisième chapitre, nous expliquerons le logiciel conçu, et par conséquent nous allons expliqués quelques parties et algorithmes de la partie programmation (Intégrale numérique, Résolution numérique des équations différentielles, procédure, et fonctions).

Une validation des résultats est considérée dans le quatrième chapitre, c'est l'exécution du logiciel sur les différents types de convertisseurs statiques, une comparaison entre les résultats obtenus avec ce *Eltsim* (logiciel conçu) et ceux obtenus avec *Simplorer* est faite.

Nous terminerons notre travail par une conclusion générale donnant une vue générale sur la précision des résultats obtenus par *Eltsim*, et des propositions pour l'amélioration de ce logiciel.

Chapitre I:

Environnement de travail de Delphi

Introduction:

Delphi est un environnement de programmation permettant de développer des applications pour Windows 95, Windows 98/2000 et Windows XP. Il présente en pratique la suite logique de la famille ***Turbo Pascal*** avec ses nombreuses versions.

Delphi est un outil moderne, qui fait appel à une conception visuelle des applications, à la programmation objet.

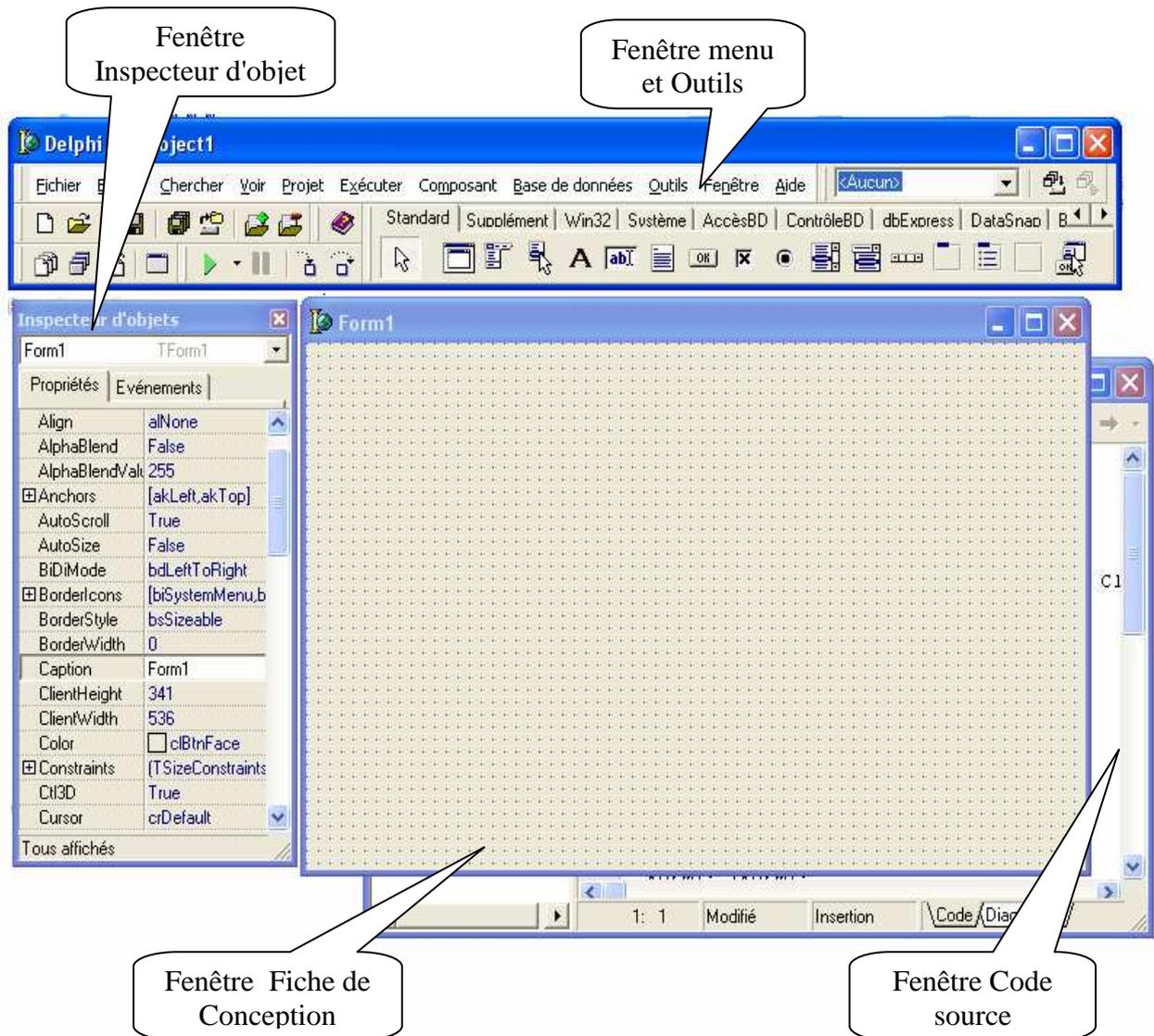
Il diffère des autres langages classiques tels que Pascal ou Fortran, Delphi fournit tous les outils qui vous sont nécessaire pour développer, tester et déployer des applications, incluant une importante bibliothèque de composants réutilisables. Delphi ressemble plus à un atelier où l'on dispose d'un ensemble d'outils de conception, des modèles d'applications et de fiches et des experts de programmation. On n'écrit pas une application mais on la fabrique.

Vous pouvez utiliser Delphi pour concevoir tout type d'application, que ce soit un utilitaire de portée générale ou un programme complexe de gestion des données ou des applications distribuées.

Les outils de base de données de Delphi et ses composants orientés données permettent de développer rapidement des applications de bases de données. [01]

I.1. Interface de Delphi:

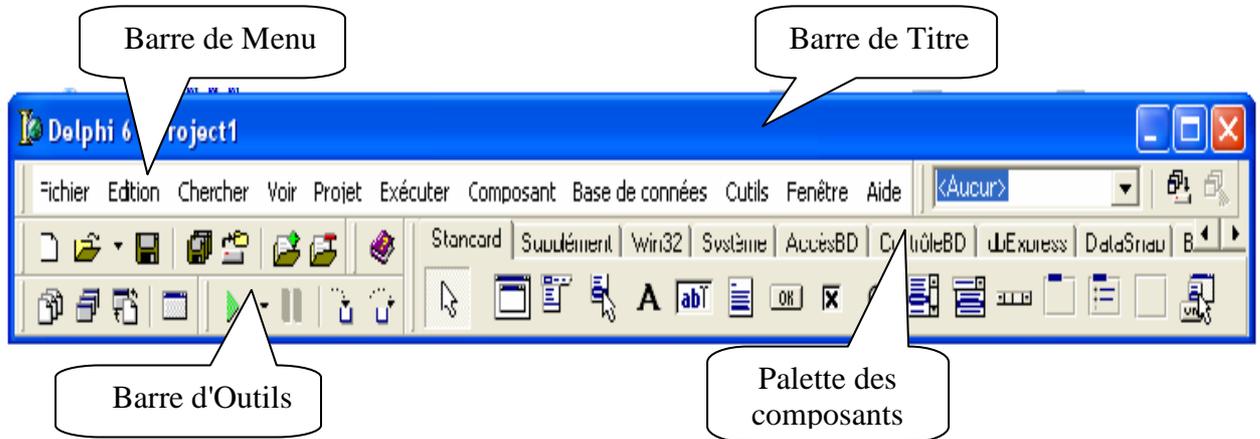
Après l'exécution de Delphi, vous pouvez constater que l'écran se compose de quatre fenêtres.[01]



I.1.1. La première fenêtre:

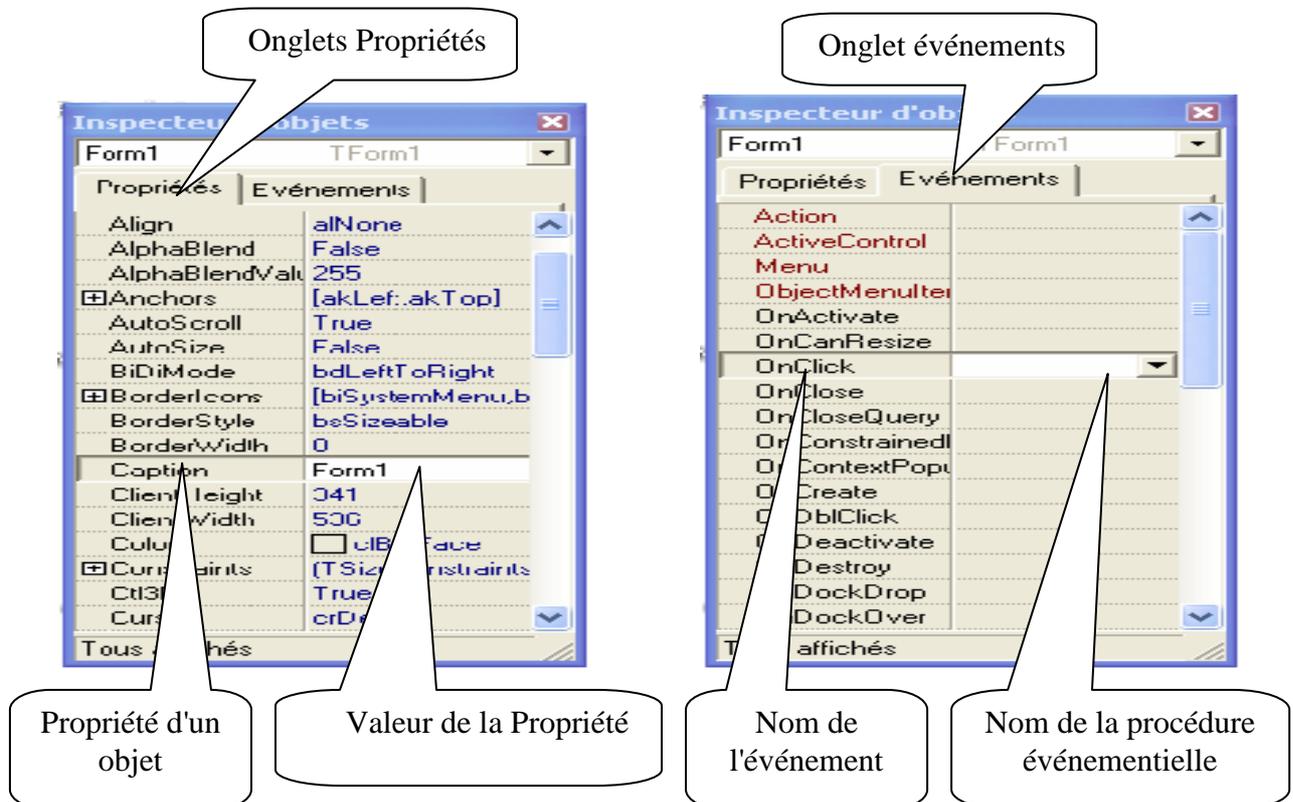
Occupe la partie de l'écran; elle contient:

- La barre de titre
- La barre de menu
- La barre d'outils
- La palette des composants. [01]



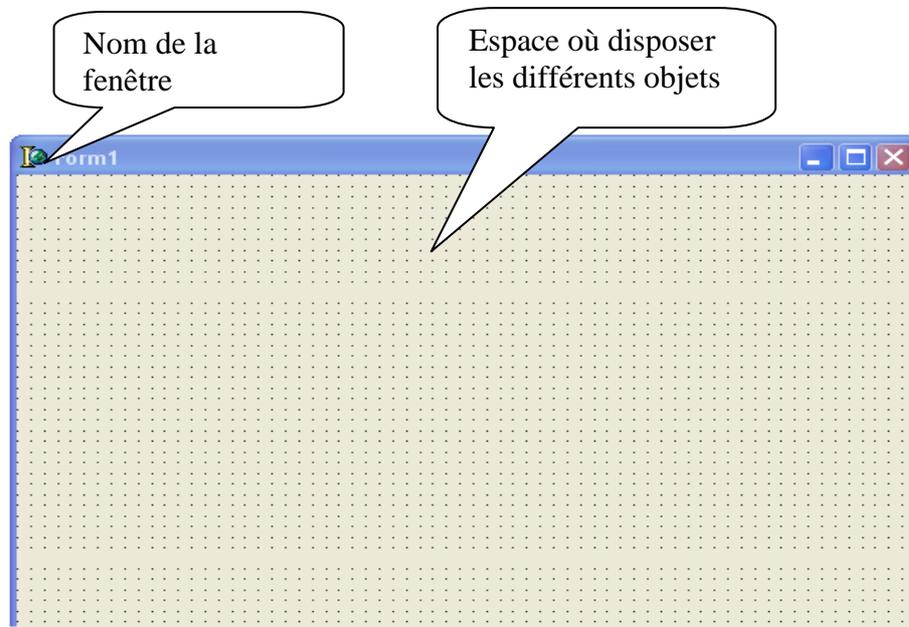
I.1.2. La deuxième fenêtre:

Elle se trouve par défaut à gauche de l'écran. Elle correspond à l'inspecteur d'objets. Elle est composée de deux volets (onglets), l'un correspond aux propriétés d'un objet, l'autre aux événements auxquels l'objet peut répondre. [01]



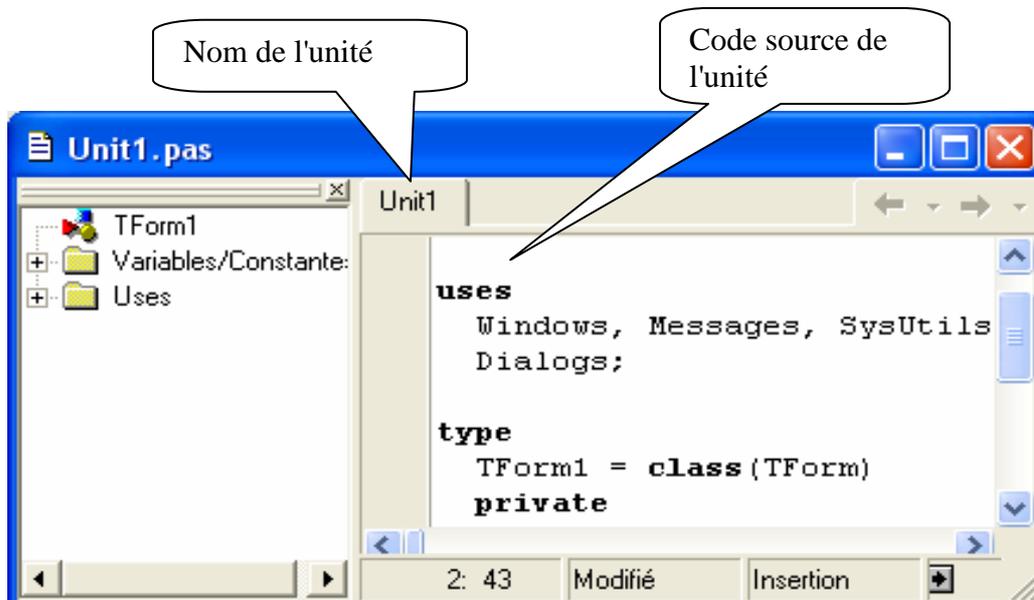
I.1.3. La troisième fenêtre:

Elle constitue la fiche principale de la future application. Il s'agit d'une fenêtre vide dans laquelle on placera les divers objets (composants). On y trouve beaucoup d'applications telles que Forme (Form), feuille ou Fenêtre. [01]



I.1.4. La quatrième fenêtre:

Elle est cachée sous la fenêtre Fiche, on appelle "L'Editeur de code", cette fenêtre contient de code source relatif à la fiche, elle représente une unité de l'application parmi d'autres. [01]



I.2. Les composants utilisés:

Dans ce qui suit, on va situer les composants importants de l'interface, c'est ceux que nous avons utilisés dans la conception de ce logiciel.



Utilisez un objet *TEdit* pour placer dans une fiche un contrôle de saisie standard. Les contrôles de saisie permettent à l'utilisateur de saisir du texte. Les contrôles permettent également d'afficher du texte.



Utilisez *TLabel* pour ajouter du texte ou un bitmap que l'utilisateur ne peut pas modifier dans une fiche. Ce texte ou un bitmap peut être utilisé pour libeller un autre contrôle et peut attribuer la focalisation à ce contrôle quand l'utilisateur saisit un raccourci clavier.



Utilisez *TMainMenu* pour placer un menu principal dans une fiche. Pour commencer la conception d'un menu, ajoutez un composant menu principal à votre fiche, puis double-cliquez sur le composant.



Utilisez *TPanel* pour placer un volet vide dans une fiche. Les volets disposent de propriétés permettant d'entourer le contrôle d'une bordure barrée, ainsi que des méthodes facilitant la gestion du positionnement des contrôles enfant incorporés dans le volet.



Utilisez *TButton* pour placer un bouton poussoir standard dans une fiche. *TButton* introduit plusieurs propriétés permettant de contrôler son comportement dans la définition d'une boîte de dialogue. Les utilisateurs choisissent des contrôles bouton pour effectuer une action.

RadioButton: 

Utilisez *TRadioButton* pour ajouter un bouton radio à une fiche. Les boutons radio proposent à l'utilisateur un ensemble d'options mutuellement exclusives, c'est-à-dire qu'un seul bouton radio d'un groupe peut être sélectionné à la fois à un moment donné. Quand l'utilisateur sélectionne un bouton radio, le bouton radio précédemment sélectionné devient désélectionné.

RadioGroup: 

Un objet *TRadioGroup* est une boîte groupe particulière ne pouvant contenir que des boutons radio. Les boutons radio contenus directement dans le même composant sont "groupés". Quand l'utilisateur active un bouton radio, tous les autres boutons radio de son groupe deviennent désactivés. Il ne peut donc y avoir deux boutons radio activés simultanément dans une fiche que s'ils sont placés dans des conteneurs distincts, comme des boîtes groupe.

CheckBox: 

Un composant *TCheckBox* propose une option à l'utilisateur. L'utilisateur peut activer la case à cocher pour sélectionner l'option, ou supprimer la coche pour la désélectionner.

GroupBox: 

Le composant *TGroupBox* représente une boîte groupe standard utilisée pour regrouper des contrôles associés d'une fiche. Quand un autre composant contrôle est placé à l'intérieur de la boîte groupe, la boîte groupe devient le parent de ce composant.

Bevel: 

Utilisez *TBevel* pour créer des boîtes, des cadres ou des lignes barrés. La barre peut apparaître en relief ou en creux.

Tree view: 

TTreeView représente une fenêtre affichant une liste hiérarchique d'éléments, les parties d'un document, les entrées d'un index ou les fichiers et répertoires d'un disque.



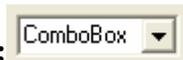
Image :

Utilisez *TImage* pour afficher une image graphique dans une fiche. Utilisez l'objet TPicture de la propriété Picture pour spécifier l'objet bitmap, dessin ou autre qui est affiché par TImage. Les propriétés et méthodes de TPicture permettent par exemple de charger une image à partir d'un fichier, d'effacer l'image de l'objet TImage et d'affecter une image à un autre contrôle.



Image list:

TImageList représente une collection d'images de tailles identiques, pouvant chacune être désignée par son indice. Les listes d'images sont utilisées pour gérer de manière efficace de grands nombres d'icônes ou de bitmaps. Toutes les images d'une liste d'images sont contenues dans un seul grand bitmap au format de périphérique écran.



ComboBox:

Un composant *TComboBox* est une boîte de saisie associée à une liste déroulante.

Les utilis

teurs peuvent sélectionner un élément de la liste ou entrer directement une valeur dans la boîte de saisie.



StatusBar:

Le composant *TStatusBar* est une ligne de volets, généralement placée en bas de la fiche, qui affiche des informations sur l'application en cours d'exécution. Chaque volet est représenté par un objet TStatusPanel énuméré dans la propriété Panels.



Scroll box:

TScrollBar représente une zone défilante (boîte de défilement) dans une fenêtre. Utilisez TScrollBar pour créer une boîte de défilement dans une fiche.



Page control:

Utilisez *TPageControl* est un ensemble de pages utilisées pour construire une boîte de dialogue multipage ou un classeur à onglets.

**String grid:**

TStringGrid est un contrôle grille conçu pour simplifier la gestion de chaînes et d'objets associés. Ajoutez-le à une fiche pour proposer des données textuelles dans un format tabulaire.

**UpDown :**

Utilisez *TUpDown* pour ajouter un contrôle flèches haut/bas à une fiche. Les contrôles flèches haut/bas consistent en une paire de boutons fléchés, comme dans un incrémenteur. Les contrôles flèches haut/bas permettent à l'utilisateur de modifier une valeur numérique en cliquant sur les boutons fléchés.

**Web Broser:**

TWebBrowser donne l'accès à la fonctionnalité de navigateur Web, pour vous permettre de créer une application navigateur Web personnalisée ou d'ajouter à vos applications Delphi des fonctions internet.

ToolBar :

TToolBar est un conteneur pour les boutons d'outils (TToolButton). Il offre un moyen simple d'organiser et de gérer des contrôles visuels. Tous les boutons d'outils d'une barre d'outils conservent une largeur et une hauteur uniformes.

Une barre d'outils peut contenir d'autres contrôles. Ces contrôles (maintenus en place par des boutons d'outils invisibles) conservent une hauteur uniforme.

Les contrôles peuvent passer automatiquement à la ligne suivante s'ils ne tiennent pas horizontalement sur la barre d'outils.

I.3. Les instructions:

I.3.1. L'instruction "if"

La syntaxe de l'instruction **if...then...else** est :

if expression **then** instruction1 **else** instruction2,

Où expression renvoie une valeur booléenne. Si expression vaut True, alors instruction1 est exécutée ; sinon instruction2 est exécutée.

I.3.2. L'instruction "While"

L'instruction **while** a la syntaxe suivante :

while expression **do** instruction

Où expression renvoie une valeur booléenne et instruction peut être une instruction composée.

L'instruction while exécute répétitivement son instruction, en testant expression avant chaque itération. Tant que expression renvoie True, l'exécution se poursuit.

I.3.3. L'instruction "for"

Une instruction **for**, à la différence de l'instruction `while`, nécessite la spécification explicite du nombre d'itérations que la boucle doit effectuer. L'instruction **for** a la syntaxe suivante :

for compteur := valeurInitiale **to** valeurFinale **do** instruction

Ou

for compteur := valeurInitiale **downto** valeurFinale **do** instruction

Où compteur est une variable locale (déclarée dans le bloc contenant l'instruction `for`) de type scalaire sans aucun qualificateur.

I.3.4. L'instruction "Procedure"

La déclaration d'une procédure a la forme :

procedure nomProcédure (listeParamètres); directives;

déclarationsLocales;

begin

instructions

end;

Où nomProcédure est un identificateur valide, instructions une série d'instructions qui s'exécute quand la procédure est appelée. Les éléments (listeParamètres), directives; et déclarationsLocales; sont facultatifs.

I.3.5. L'instruction "Function"

Une déclaration de fonction est similaire à la déclaration d'une procédure mais elle spécifie le type de la valeur renvoyé. Une déclaration de fonction a la forme suivante :

```
function nomFonction(listeParamètres): typeRenvoyé; directives;
```

```
déclarationsLocales;
```

```
begin
```

```
instructions
```

```
end;
```

Où nomFonction est un identificateur valide, typeRenvoyé est un nom de type, instructions la série des instructions exécutées quand la fonction est appelée. Les éléments (listeParamètres), directives; et déclarationsLocales; sont facultatifs.

I.4. La conversion:

Par fois il est nécessaire de faire une conversion d'un entier ou un réel vers un caractère ou l'inverse, pour cela on utilise les instructions suivantes:

Entier vers un caractère : inttostr

Réel vers un caractère: floattostr

Caractère vers un entier: strtoint

Caractère vers un réel: strtfloat

Chapitre II:

Modélisation des convertisseurs statiques

Introduction:

L'électronique de puissance est considérée parmi les grandes applications de l'électrotechnique, qui s'intéresse à la conversion statique de l'énergie électrique.

La conversion de l'énergie électrique à une forme appropriée après sa production nécessite des montages spéciaux pour faire cette conversion (ils sont appelés convertisseurs statiques).

Une étude des convertisseurs statiques, son fonctionnement, et le comportement des tensions et des courants est nécessaire, pour les programmer d'une façon juste.

Pour simplifier les calculs des tensions et des courants (partie programmation), on suppose que tous les semi-conducteurs sont parfaits.

II.1. Les redresseurs:

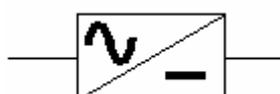
Les Redresseurs

Introduction:

Les montages redresseurs, sont les convertisseurs de l'électronique de puissance qui assurent directement la conversion alternatif- continu .Ils sont alimentés par une source de tension alternative monophasée ou polyphasée.

Souvent, les tensions monophasées et triphasées du réseau industriel de 50 Hz qui sont redressées. [07]

On représente un redresseur par le symbole suivant:



II. 1.1. Tension redressée:

La tension redressée U_d fournie par un montage redresseur à diode d'indice q est formée de q sommets de sinusoïdes par période T des tensions alternatives sinusoïdales d'alimentation.

Si ω est la pulsation de ses tensions, la tension U_d est donc une fonction périodique de période T/q dont le fondamental a pour pulsation $q \omega$. [04]

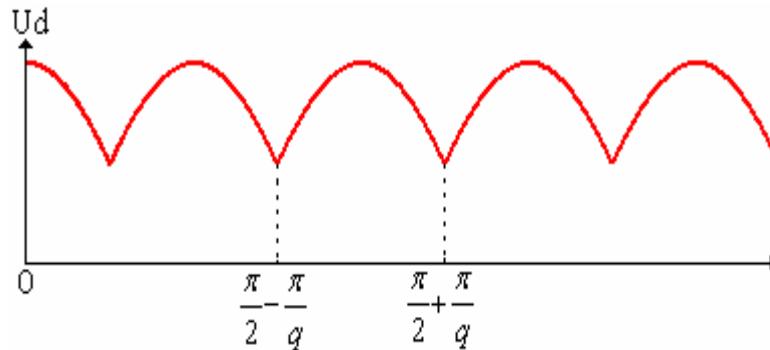


Figure (II.1): Tension redressée d'un redresseur à q phases

II. 1.2. Valeur moyenne de la tension redressée

$$U_{d0} = \frac{q}{2\pi} \int_{\frac{\pi}{2} - \frac{\pi}{q}}^{\frac{\pi}{2} + \frac{\pi}{q}} V_m \sin \omega t d\omega t = \frac{q}{\pi} V_m \sin \frac{\pi}{q} \quad (\text{II.1})$$

II. 1.3. Le développement en série de Fourier:

Le développement en série de la tension U_d comprend, en plus de U_{d0} , des termes sinusoïdaux de pulsation $q \omega$, $2 q \omega$, ..., $K q \omega$. [07]

$$U_d = U_{d0} + \sum_{K=1}^{\infty} U_K \sin Kq \omega t d\omega t \quad (\text{II.2})$$

$$U_K = \frac{2q}{2\pi} \int_0^{\frac{\pi}{q}} V_m \sin \omega t \sin Kq \omega t d\omega t = \frac{q}{\pi} \left[\frac{\sin(Kq+1)\frac{\pi}{q}}{Kq+1} + \frac{\sin(Kq-1)\frac{\pi}{q}}{Kq-1} \right] \quad (\text{II.3})$$

$$U_K = -\frac{2qV_m}{\pi(K^2q^2-1)} (-1)^K \sin \frac{\pi}{q}$$

$$\text{Donc la valeur de } U_d \text{ s'exprime comme suit: } U_d = U_{d0} \left[1 + \sum_{K=1}^{\infty} \frac{-2(-1)^K}{K^2q^2-1} \cos Kq \omega t \right] \quad (\text{II.4})$$

II. 1.4. Tension inverse:

La tension aux bornes d'une diode D_j par exemple, a pour expression $v_1 - v_j$, $j=1, q$.

La tension inverse maximale correspond au maximum de la plus grande de ces différences

Si q est pair, la tension la plus éloignée de v_1 est $v_{q/2+1}$, ce qui donne $v_{Dmax}=2v_m$.

Si q est impair les tensions les plus éloignées de v_1 sont obtenus pour $v_{(q+1)/2}$ et $v_{(q+3)/2}$, ce qui

$$\text{donne } v_{Dmax} = 2v_m \cos \frac{\pi}{2q} \quad .[04]$$

II. 1.5. Courants et puissances:

Si le montage débite un courant continu I_d constant, chaque diode assure le passage de I_d pendant l'intervalle de durée T/q où elle est conductrice.

D'où les valeurs maximales, moyennes et efficaces du courant dans chacune des diodes. [07]

$$i_{\max} = I_d, \quad (\text{II.5})$$

$$i_{\text{moy}} = \frac{I_d}{q}, \quad (\text{II.6})$$

$$I_{\text{eff}} = \sqrt{\frac{q}{2\pi} \int_0^q I_d^2 d\omega t} = \frac{I_d}{\sqrt{q}} \quad (\text{II.7})$$

- Si on néglige les chutes de tension, puisque le courant I_d est supposé constant, la puissance débitée par le secondaire du transformateur est :

$$P_d = U_{d0} I_d \quad (\text{II.8})$$

- Si on considère les chutes de tensions, et que i_d n'est pas constant, alors on obtient:

$$P = \frac{1}{2\pi} \int_0^{2\pi} I_d U_d d\omega t = U_0 I_0 + U_1 I_1 \cos \varphi_1 + U_2 I_2 \cos \varphi_2 + U_3 I_3 \cos \varphi_3, \dots \quad (\text{II.9})$$

Or la puissance apparente du secondaire formé de q enroulements, siège de tensions de valeur efficace $v = \frac{v_m}{\sqrt{2}}$ parcourus par des courants de valeur efficace I_{eff} est:

$$S = q \frac{V_m}{\sqrt{2}} I_{\text{eff}}, \quad (\text{II.10})$$

$$Q = q v_1 I_1 \sin \varphi_1, \quad (\text{II.11})$$

$$D = \sqrt{S^2 - P^2 - Q^2} \quad (\text{II.12})$$

II. 1.6. Facteur de puissance

Le facteur de puissance est défini comme le rapport entre la puissance active et la puissance apparente. [07]

$$F_s = \frac{P}{S} = \frac{I_d \frac{q}{\pi} V_m \sin \frac{\pi}{q} \cos \alpha}{q \frac{V_m}{\sqrt{2}} \frac{I_d}{\sqrt{q}}} = \frac{\sqrt{2q}}{\pi} \sin \frac{\pi}{q} \cos \alpha \quad (II.13)$$

La valeur de F_p calculée (pour des montages à diodes) pour quelques valeurs de q . [07]

q	2	3	4	6	12	18
F_p	0.636	0.675	0.636	0.55	0.40	0.332

II. 1.7. Les montages usuels:

II. 1.7.1. Redresseur monophasé simple:

II. 1.7.1.1. Montage pratique

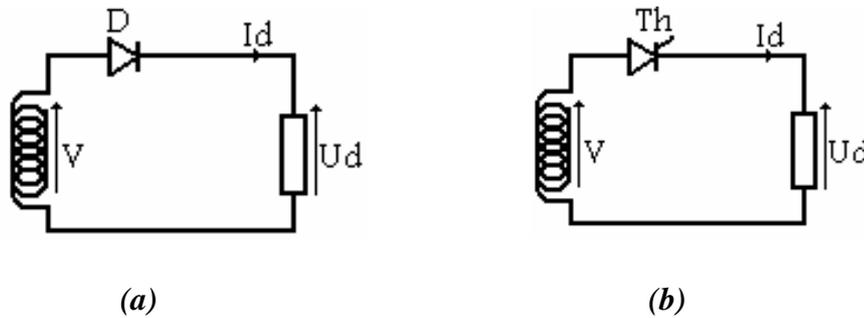


Figure (II.2): Montage pratique d'un redresseur simple (a) à diode , (b) à thyristor

II. 1.7.1.2. Tension redressée

$U_d = v$ si la diode (thyristor) est passante

$U_d = 0$ si la diode (thyristor) est bloquée

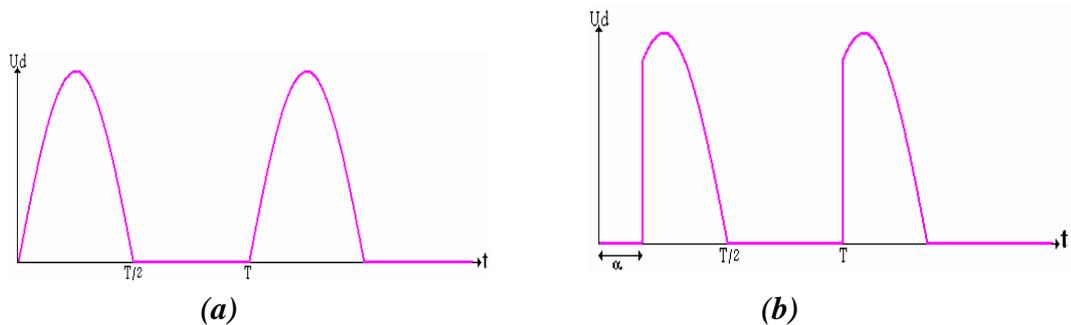


Figure (II.3): Tension redressée d'un redresseur simple sur charge R: (a) à diode, (b) à thyristor

II. 1.7.1.3. Valeur moyenne de la tension redressée

$$\overline{U_d} = \frac{1}{2\pi} \int_{\alpha}^{\pi+\alpha} v d\omega t = \frac{1}{2\pi} V_m (1 + \cos \alpha) \tag{II.14}$$

Pour la diode il suffit de poser $\alpha = 0$

II. 1.7.1.4. Tension inverse aux bornes de la diode (thyristor)

$v_D=0$ si la diode (thyristor) est passante

$v_D=v$ si la diode (thyristor) est bloquée

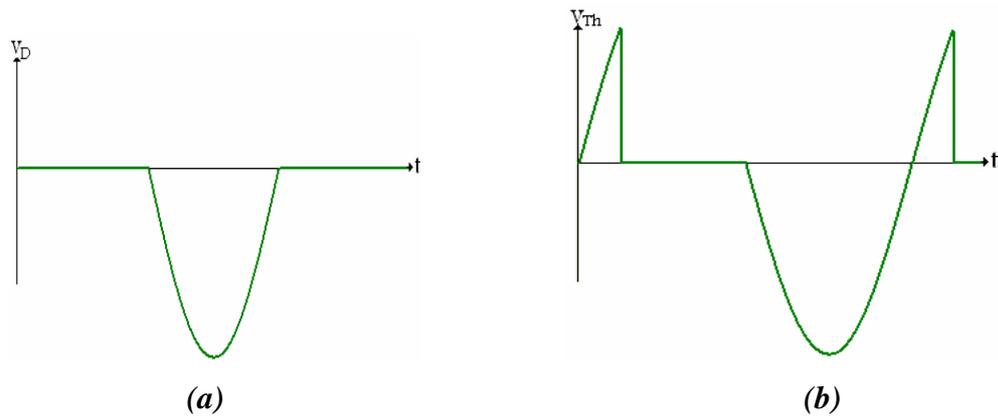


Figure (II.4): Tension inverse de la diode (thyristor) d'un redresseur simple sur charge R: (a) à diode, (b) à thyristor

II. 1.7.1.5. Réseau de Petri:

Pour que la diode (thyristor) soit bloquée, il faut que le courant s'annule, et pour qu'elle soit passante, il faut que la tension à ses bornes soit positive.

On pose les deux cas suivants:

E_1 : diode passante

E_2 : diode bloquée

On définit les deux transitions suivantes:

R_1 : $i_D = I_d = 0$

R_2 : $v_D = v - U_d \geq 0$

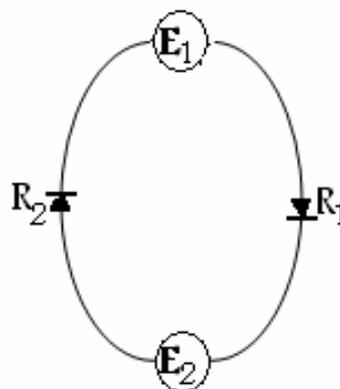


Figure (II.5): Réseau de Petri d'un redresseur simple à diode

II. 1.7.2. Redressement monophasé en pont:

Le redressement monophasé en pont est un redressement double voie, c'est-à-dire que deux éléments semi-conducteur conduisent au même temps, et les autres sont bloqués. [07]

II.1.7.2.1. Montage pratique

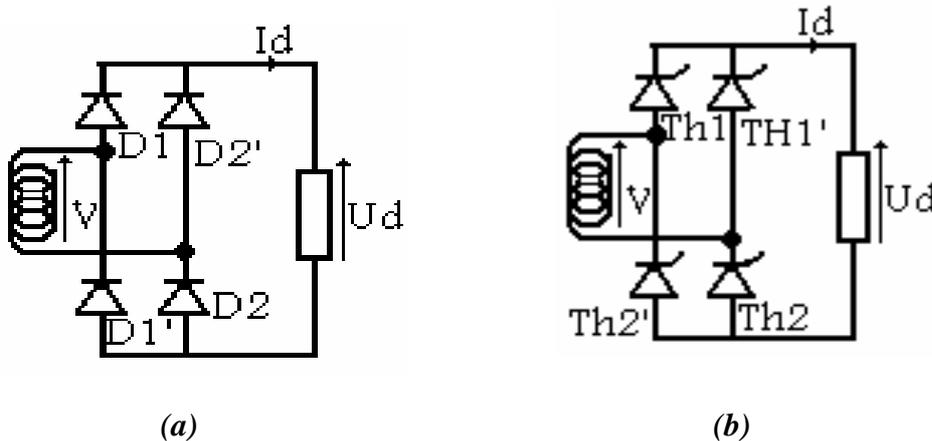


Figure (II.6): Montage pratique d'un redresseur monophasé en pont (a) à diodes, (b) à thyristors

II. 1.7.2.2. Tension redressée

$U_d = v$ si les diodes $D_1(Th_1)$ et $D_2(Th_2)$ sont passantes

$U_d = -v$ si les diodes $D_1'(Th_1')$ et $D_2'(Th_2')$ sont passantes

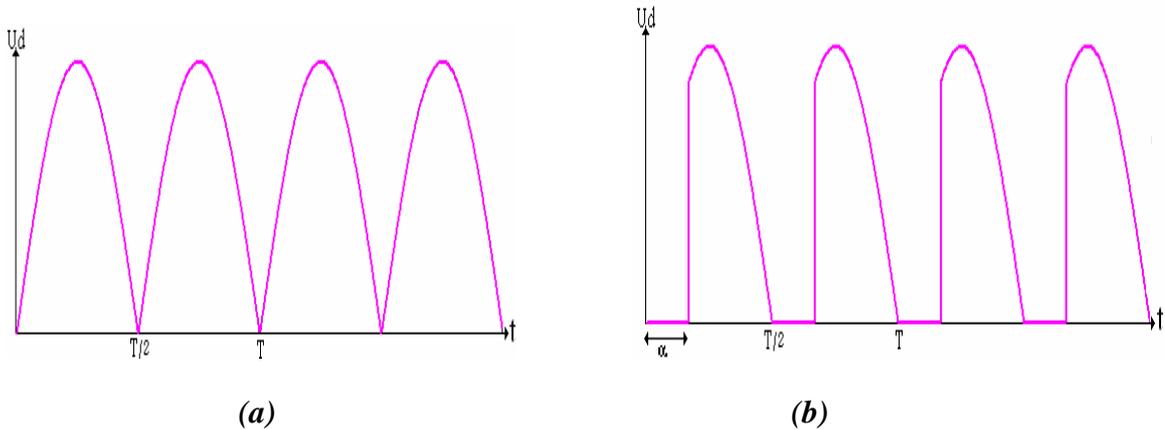


Figure (II.7): Tension redressée d'un redresseur monophasé en pont sur charge R: (a) à diodes, (b) à thyristors

II. 1.7.2.3. Valeur moyenne de la tension redressée

$$\overline{U_d} = \frac{1}{2\pi} \int_0^{2\pi} U_d \omega t = \frac{2}{2\pi} \int_{\alpha}^{\pi+\alpha} U_d \omega t = \frac{2}{\pi} V_m \cos \alpha = U_{d(diode)} \cos \alpha \tag{II.15}$$

II. 1.7.2.4. Tension inverse aux bornes de la diode D₁ (thyristor Th₁)

$v_{D1} = 0$ si les diodes D₁(Th₁) et D₂(Th₂) sont passantes

$v_{D1} = v$ si les diodes D₁'(Th₁') et D₂'(Th₂') sont passantes

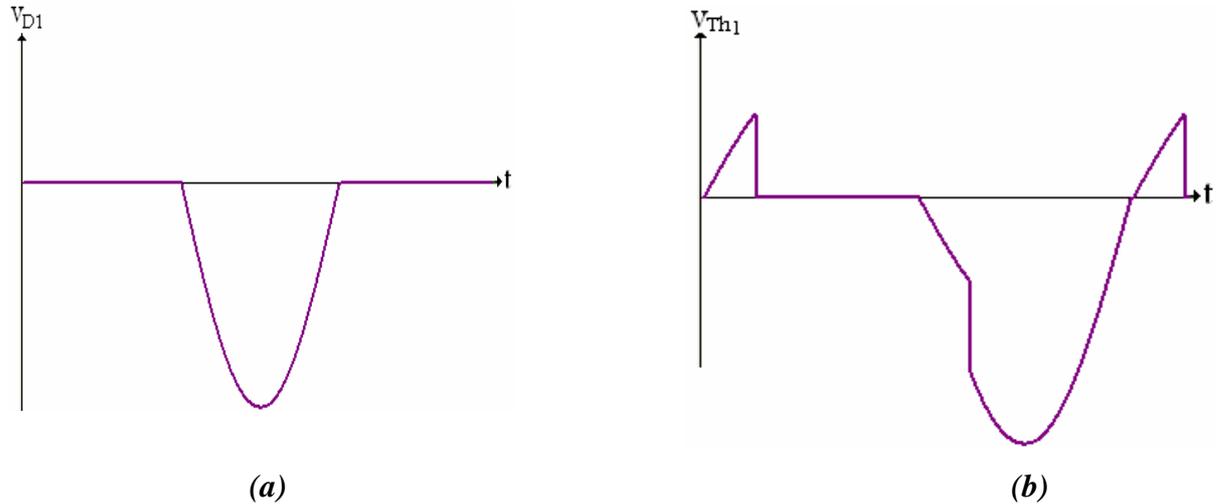


Figure (II.8): Tension inverse d'une diode (thyristor) d'un redresseur monophasé en pont sur charge R: (a) à diode, (b) à thyristor

II. 1.7.2.5. Réseau de Petri:

Pour que la diode (thyristor) soit bloquée, il faut que le courant s'annule, et pour qu'elle soit passante, il faut que la tension à ses bornes soit positive.

On pose les deux cas suivants:

E₁: diode passante

E₂: diode bloquée

On définit les deux transitions suivantes:

R₁: $i_{D1} = I_d = 0$

R₂: $v_{D1} = \frac{v - U_d}{2} \geq 0$

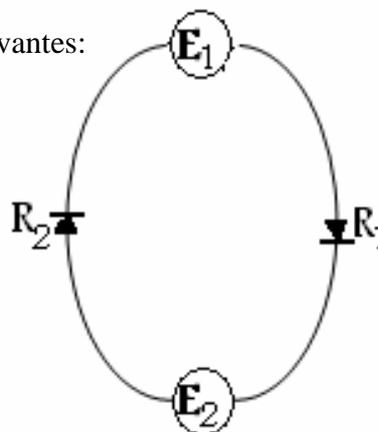


Figure (II.9): Réseau de Petri d'un redresseur monophasé en pont

II. 1.7.3. Redressement triphasé parallèle simple:

Le système triphasé est décalé dans le temps comme suit

$$\begin{cases} V_1(t) = \sqrt{2} \cdot V_m \sin(\omega t) \\ V_2(t) = \sqrt{2} \cdot V_m \sin(\omega t - \frac{2\pi}{3}) \\ V_3(t) = \sqrt{2} \cdot V_m \sin(\omega t - \frac{4\pi}{3}) \end{cases} \quad (II.16)$$

II.1.7.3.1. Montage pratique

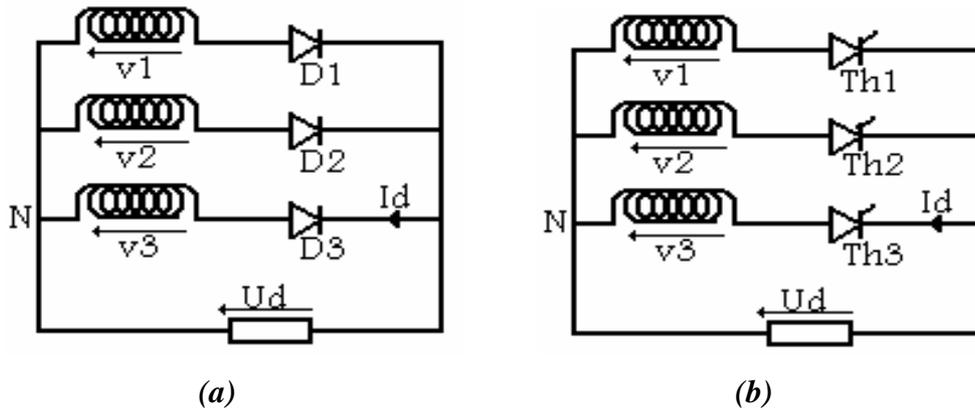


Figure (II.10): Montage pratique d'un redresseur triphasé parallèle simple (a) à diodes, (b) à thyristors

II. 1.7.3.2. Tension redressée

$$U_d = \sup \{ v_1, v_2, v_3 \}$$

$U_d = v_1$ si la diode D_1 (Th_1) est passante

$U_d = v_2$ si la diode D_2 (Th_2) est passante

$U_d = v_3$ si la diode D_3 (Th_3) est passante

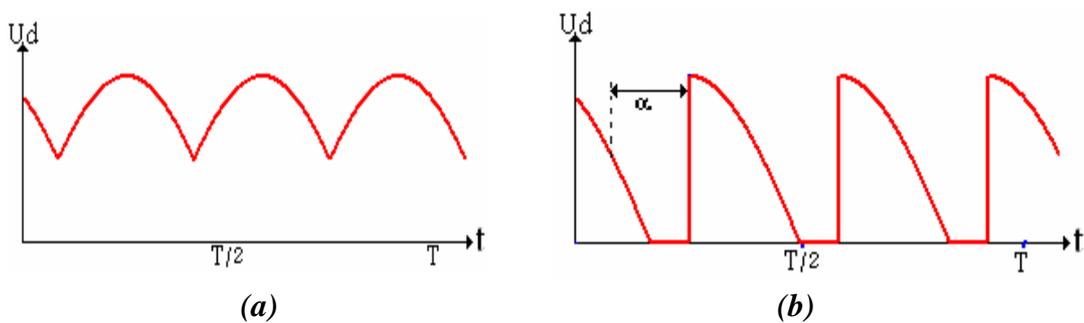


Figure (II.11): Tension redressée d'un redresseur triphasé parallèle simple sur charge R. (a) à diodes, (b) à thyristors

II. 1.7.3.3. Valeur moyenne de la tension redressée

$$\overline{U_d} = \frac{1}{2\pi} \int_0^{2\pi} v d\omega t = \frac{3}{2\pi} \int_{\frac{\pi}{6}+\alpha}^{\frac{5\pi}{6}+\alpha} v d\omega t = \frac{3\sqrt{3}}{2\pi} V_m \cos \alpha = U_{d(\text{diode})} \cos \alpha \quad (\text{II.17})$$

II. 1.7.3.4. Tension inverse aux bornes de la diode $D_1(\text{Th}_1)$

$v_{D1} = v_1 - v_j$ si la diode $D_j(\text{Th}_j)$ est passante ($j=1,3$)

$v_{D1} = v_1 - v_1 = 0$ si la diode $D_1(\text{Th}_1)$ est passante

$v_{D1} = v_1 - v_2$ si la diode $D_2(\text{Th}_2)$ est passante

$v_{D1} = v_1 - v_3$ si la diode $D_3(\text{Th}_3)$ est passante

II. 1.7.3.5. Réseau de Petri:

Pour que la diode (thyristor) soit bloquée, il faut que le courant s'annule, et pour qu'elle soit passante, il faut que la tension à ses bornes soit positive.

On pose les deux cas suivants:

E_1 : diode passante

E_2 : diode bloquée

On définit les deux transitions suivantes:

R_1 : $i_{D1} = I_d = 0$

R_2 : $v_1 - U_d \geq 0$

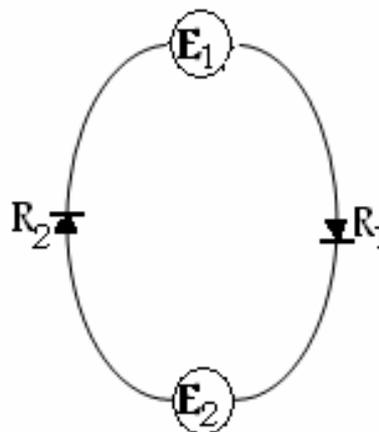


Figure (II.12): Réseau de Petri d'un redresseur triphasé parallèle simple

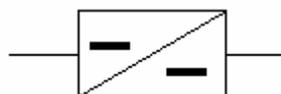
II.2. Les Hacheurs:

Les Hacheurs

Introduction:

Les hacheurs sont des convertisseurs d'énergie de type continu- continu, qui sont placés entre la source et la charge. Ils permettent de délivrer des tensions dont les sorties sont variables.

On représente un hacheur par le symbole suivant:



Selon le modèle du hacheur, un hacheur peut être abaisseur ou éleveur en tension (un quadrant), réversible en courant (deux quadrants), réversible en tension (deux quadrants), réversible en tension et en courant (quatre quadrants). [05]

II. 2.1. Hacheur série (dévolteur):

Le hacheur est mis en série avec la source d'entrée, et la charge.

II. 2.1.1. Montage pratique

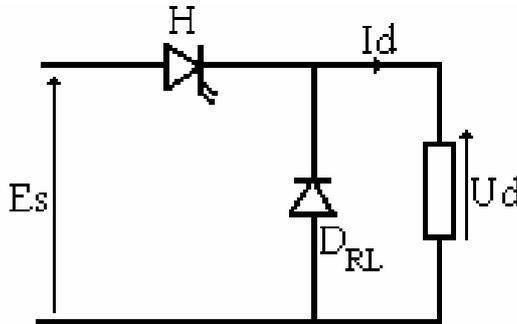


Figure (II.13): Montage pratique d'un hacheur série

Es: Source de tension continue

DRL: la diode de roue libre. Elle permet d'assurer la continuité du courant dans la charge et de protéger l'interrupteur contre les surtensions.

L'interrupteur H est commandé périodiquement sur une période T.

On définit le rapport cyclique $\alpha = \frac{t_{on}}{T}$.

On prend RL comme charge.

$0 < t < \alpha T$: H est fermé, DRL est bloquée

$$U_d = E_s = Ri + L \frac{di}{dt} \implies id = \frac{E_s}{R} + Ae^{-\frac{t}{\tau}}, \tau = \frac{L}{R}, \text{ et on prend en considération la condition}$$

$$\text{initiale: } i_d(0) = i_{dmin} \implies i_d = \frac{E_s}{R} + (i_{dmin} - E_s) e^{-\frac{t}{\tau}} \dots \dots \dots (a)$$

$\alpha T < t < T$: H est ouvert, DRL passante

$$0 = Ri_d + L \frac{di_d}{dt} \implies id = \beta e^{-\frac{t}{\tau}}, \text{ et on prend en considération la condition suivante}$$

$$i_d(\alpha T) = i_{dmax}, \implies i_d = i_{dmax} e^{-\frac{t-\alpha T}{\tau}} \dots \dots \dots (b)$$

$$\text{De (a) et (b), on trouve } \left\{ \begin{array}{l} i_{dmax} = \frac{E_s}{R} \frac{1 - e^{-\frac{\alpha T}{\tau}}}{1 - e^{-\frac{T}{\tau}}} \\ i_{dmin} = \frac{E_s}{R} \frac{1 - e^{-\frac{\alpha T}{\tau}}}{1 - e^{-\frac{T}{\tau}}} \end{array} \right. \quad (II.18)$$

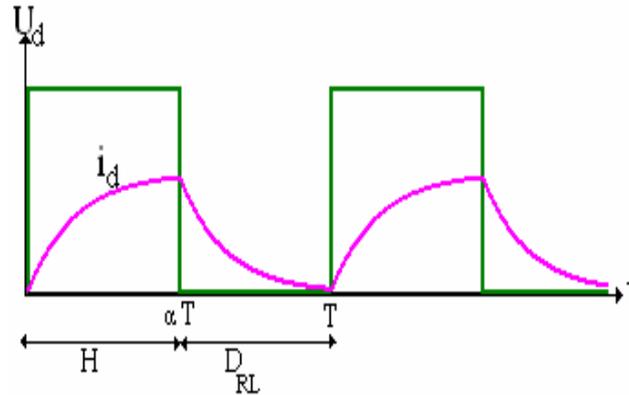


Figure (II.14): Tension, et courant délivrés par un hacheur série

II. 2.1.2. La valeur moyenne de la tension de sortie:

$$\overline{U_d} = \frac{1}{T} \int_0^T U_d dt = \frac{1}{T} \int_0^{\alpha T} E_s dt = \alpha E_s, \quad \overline{U_d} = \alpha E_s \quad (\text{II.19})$$

il est abaisseur de tension

II.2.1.3. La valeur moyenne du courant:

$$\overline{U_d} = \frac{1}{T} \int_0^T (Ri_d + L \frac{di_d}{dt}) dt = R \cdot \frac{1}{T} \int_0^T i_d dt = R \overline{i_d} \implies \overline{i_d} = \frac{\overline{U_d}}{R} \quad (\text{II.20})$$

II. 2.1.4. Ondulation du courant:

On définit le facteur d'ondulation comme suit:

$$i_{\%} = \frac{i_{d \max} - i_{d \min}}{i_{d \min}} = e^{\frac{(1-\alpha)T}{\tau}} - 1 \quad (\text{II.21})$$

Pour réduire l'ondulation du courant, on peut:

- Insérer une self de lissage en série avec la charge,
- Augmenter la fréquence de hachage.

II. 2.1.5. Réseau de Petri :

On définit les trois cas suivants:

E₁: H est passant

E₀: Blocage du hacheur ou de la diode

E₂: D passante

On définit les six transitions suivantes:

R₁: i_H=0

R₂: v_H>0 & impulsion d'amorçage de H

R₃: v_D>0

R₄: i_D=0

R₅: i_H ≠ 0 & impulsion de désamorçage de H

R₆: i_D ≠ 0 & impulsion d'amorçage de H

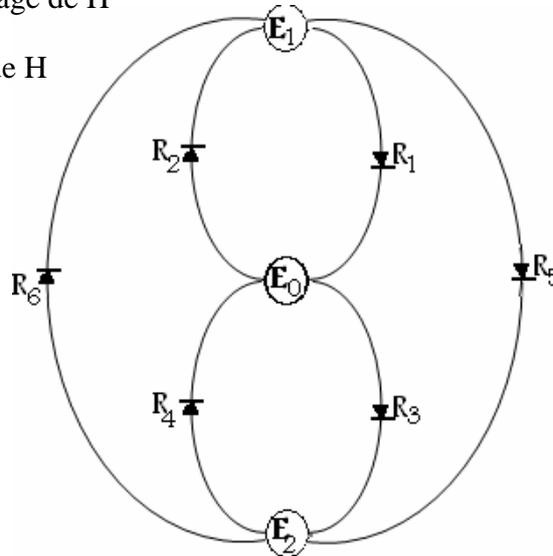


Figure (II.15): Réseau de Petri d'un hacheur série

II. 2.1.6. Tension aux bornes d'une machine à courant continu:

Dans la commande d'une machine à courant continu, il ya deux modes de conduction:

Conduction continue:

$$[0, \alpha T] \text{ H fermé, } D_{RL} \text{ bloquée } E_s = r i_d + \frac{L di_d}{dt} + e \dots\dots\dots(a)$$

$$[\alpha T, T] \text{ H ouvert, } D_{RL} \text{ passante } 0 = r i_d + \frac{L di_d}{dt} + e \dots\dots\dots(b)$$

D'après(a) et (b) en obtient: $\overline{U_d} = \alpha E_s$ (II.22)

Conduction discontinue:

$$[0, \alpha T] \text{ H fermé, } D_{RL} \text{ bloquée } E_s = r i_d + \frac{L di_d}{dt} + e \dots\dots\dots(a')$$

$$[\alpha T, \beta T] \text{ H ouvert, } D_{RL} \text{ passante } 0 = r i_d + \frac{L di_d}{dt} + e \dots\dots\dots(b')$$

$$[\beta T, T] \text{ H ouvert, } D_{RL} \text{ bloquée } U_d = e \dots\dots\dots(c')$$

D'après (a'), (b'), et (c') en obtient: $\overline{U_d} = \alpha E_s + (1 - \beta)e$ (II.23)

II. 2.2. Hacheur parallèle (survolteur):

Le hacheur est mis en parallèle avec la source d'entrée et la charge. Le générateur est de type inductif pour que le courant soit lissé, et le récepteur est de type capacitif pour que la tension aux bornes de la charge soit constante, et pour empêcher la décharge de la capacité dans le hacheur, on place une diode en série.

II.2.2.1. Montage pratique

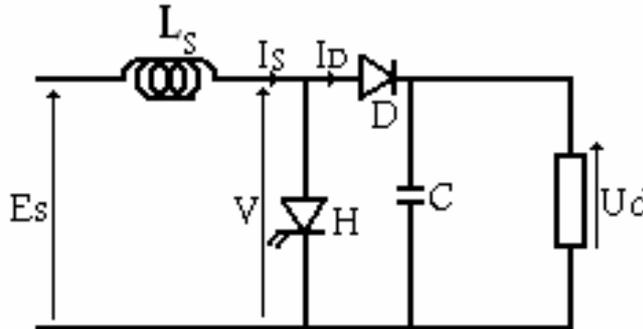


Figure (II.16): Montage pratique d'un hacheur parallèle

II. 2.2.2. La valeur moyenne de la tension de sortie:

$[0, \alpha T]$ H fermé, D bloquée

$$E_s = L_s \frac{di_s}{dt} \implies I_s = \frac{E_s}{L_s} t + I_{s\min}$$

$$I_{s\max} = \frac{E_s}{L_s} \alpha T + I_{s\min}, \implies \Delta i_s = \frac{E_s}{L_s} \alpha T \dots\dots\dots(a)$$

$[\alpha T, T]$ H ouvert, D passante $E_s - U_d = L_s \frac{di_s}{dt} \implies I_s = \frac{E_s - U_d}{L_s} (t - \alpha T) + I_{s\max}$

$$I_{s\min} = \frac{E_s - U_d}{L_s} (T - \alpha T) + I_{s\max}, \implies \Delta i_s = \frac{U_d - E_s}{L_s} (1 - \alpha) T \dots\dots\dots(b)$$

D'après(a) et (b) en obtient: $\overline{U_d} = \frac{E_s}{1 - \alpha}$ (II.24)

il est élévateur de tension

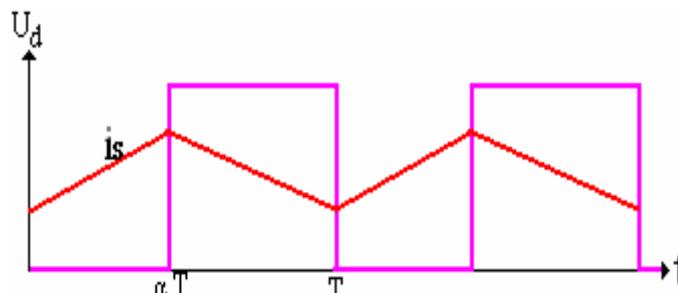


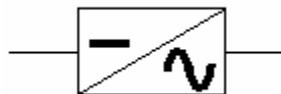
Figure (II.17): Tension délivrée par un hacheur parallèle

II.3. Les Onduleurs:

Les Onduleurs

Introduction:

Un onduleur est un convertisseur statique qui permet d'alimenter une charge en courant alternatif à partir d'une source continue, c'est un convertisseur continu- alternatif, il est symbolisé par :



L'onduleur est dit autonome quand il impose sa propre fréquence à la charge. Si la source continue est une source de tension, l'onduleur est appelé onduleur de tension. Il impose la forme d'onde de la tension aux bornes de la charge, la forme d'onde du courant dépend de la charge.

Si la source continue est une source de courant, l'onduleur est appelé commutateur de courant. Il impose la forme d'onde du courant, par contre la forme d'onde de la tension aux bornes de la charge dépend de la nature de la charge.

II.3.1. Onduleur en demi- pont:

II. 3.1.1. Montage pratique

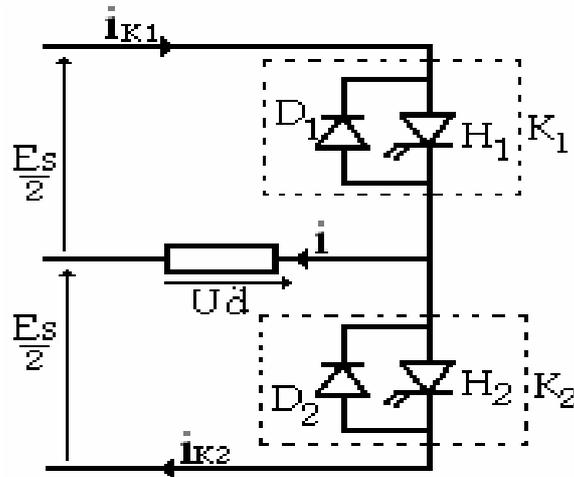


Figure (II.18): Montage pratique d'un onduleur en demi-pont

II. 3.1.2. Tension aux bornes de la charge

Les interrupteurs électroniques sont commandés alternativement et de façon complémentaire sur une période T à la fréquence f .

$[0, T/2]$: K_1 est fermé, (H_1 ou D_1 est passant) et K_2 est ouvert.

$[T/2, T]$: K_1 est ouvert, et K_2 fermé (H_2 ou D_2 est passant).

La diode branchée en antiparallèle sur chaque interrupteur permet au courant dans la charge de circuler dans les deux sens.

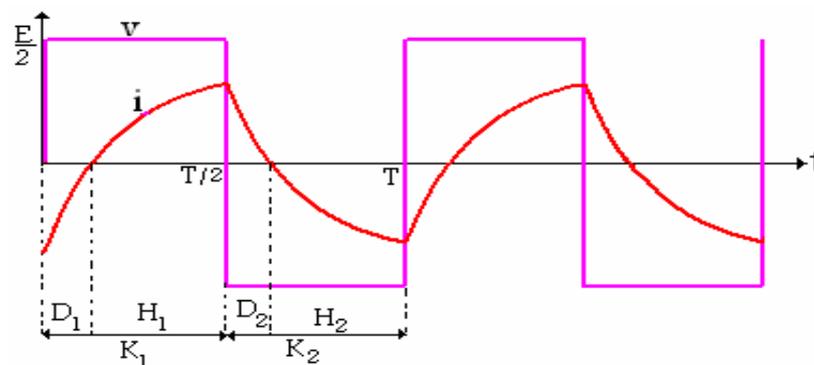


Figure (II.19): Tension délivrée par un onduleur en demi- pont

II. 3.2. Onduleur en pont:

II. 3.2.1. Montage pratique

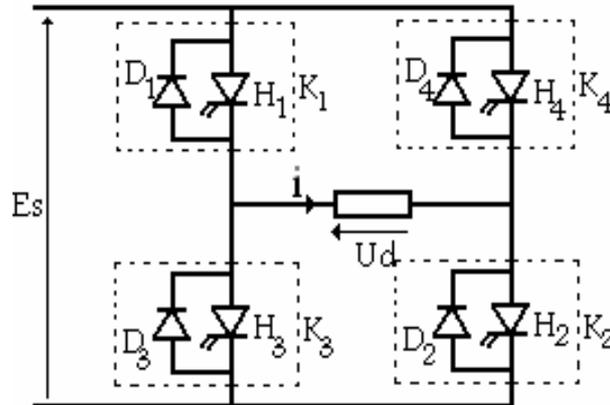


Figure (II.20): Montage pratique d'un onduleur en pont

II. 3.2.2. Tension aux bornes de la charge

Les interrupteurs électroniques sont commandés alternativement et de façon complémentaire sur une période T à la fréquence f.

[0, T/2]: K₁ et K₂ sont fermés, (H₁ et H₂, ou D₁ et D₂ passants), K₃ et K₄ sont ouverts

$$U_d = E_s \tag{II.25}$$

[T/2, T]: K₃ et K₄ sont fermés, (H₃ et H₄, ou D₃ et D₄ passants), K₁ et K₂ sont ouverts.

$$U_d = -E_s \tag{II.26}$$

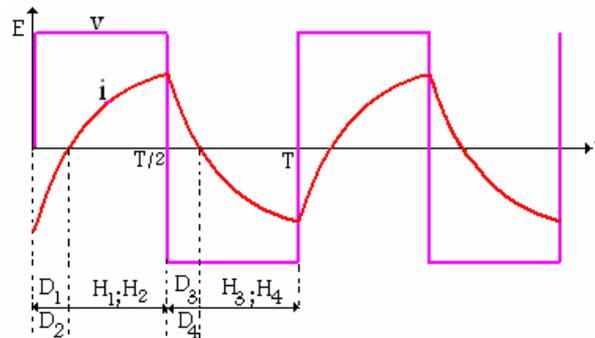


Figure (II.21): Tension délivrée par un onduleur en pont

II. 3.3. Le développement en série de Fourier:

La tension u(t) est une fonction créneau impaire de valeur moyenne nulle. Sa décomposition en série de Fourier ne contient pas des termes en cosinus et ne présente pas des harmoniques de rang paires. [07]

$$u(t) = \sum_{k=0}^{\infty} \frac{4E_s}{(2K+1)\pi} \sin(2K+1)\omega t \tag{II.27}$$

II. 3.4. Onduleur en tension triphasé:

L'onduleur triphasé en pont est constitué de trois cellules de commutation.

Trois des six interrupteurs K sont simultanément conducteurs. Les trois autres sont bloqués. Deux interrupteurs d'une même verticale sont commandés en complémentarité pour ne pas court-circuiter la source de tension, ce qui va donner des tensions simple triphasé qui sont obtenues de façon composée sur les trois bornes de sortie. [07]

II. 3.4.1. Montage pratique

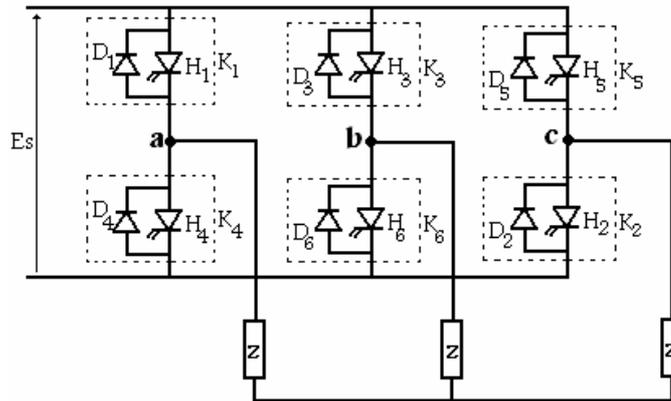


Figure (II.22): Montage pratique d'un onduleur triphasé

II.3.4.2. Tension aux bornes de la charge

Il ya six séquences par période. Chaque interrupteur électronique K est fermé pendant une demi-période (commande à 180°).

Les séquences de conduction (commande) sont:

(K_5, K_6, K_1) ; (K_6, K_1, K_2) ; (K_1, K_2, K_3) ; (K_2, K_3, K_1) ; (K_3, K_4, K_5) ; (K_4, K_5, K_6) . [07]

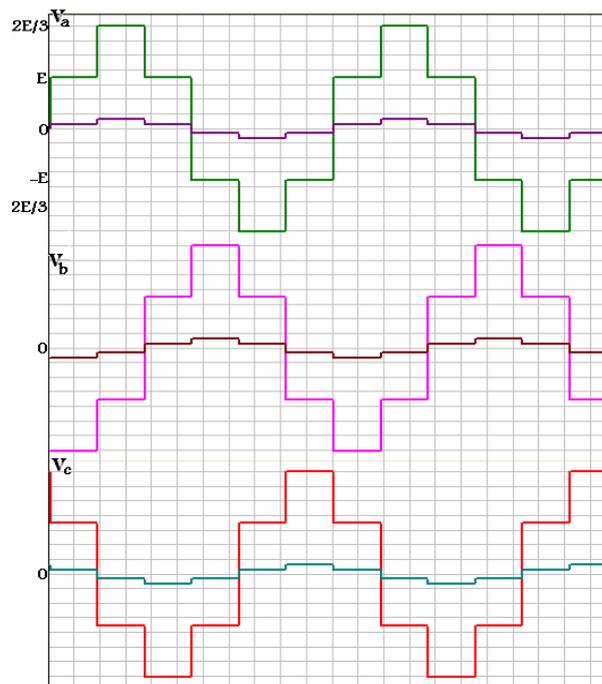


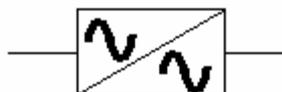
Figure (II.23): Courants et tensions délivrés par un onduleur triphasé débite sur des charges R.

II.4. Les Gradateurs:

Les Gradateurs

Introduction:

Un gradateur est un convertisseur statique qui permet d'alimenter une charge en courant alternatif à partir d'une source alternative, c'est un convertisseur alternatif- alternatif, il est symbolisé par :



L'élément de base est formé de deux thyristors montés en tête- bêche (Triac), placé entre la source et le récepteur. [04]

II. 4.1. Gradateur monophasé

Un gradateur monophasé permet le réglage continu de la valeur du courant qu'une source de tension débite dans un récepteur où le gradateur est placé entre eux.

II.4.1.1. Montage pratique

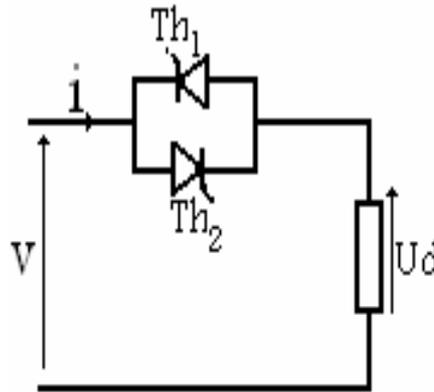


Figure (II.24): Montage pratique d'un gradateur monophasé

II. 4.1.2. Tension aux bornes de la charge

Si la charge est une résistance pure alors la tension et le courant de sortie du gradateur sont montrés dans la figure suivante.

$V_s = V$ si Th_1 , ou Th_2 passant

$V_s = 0$ si non

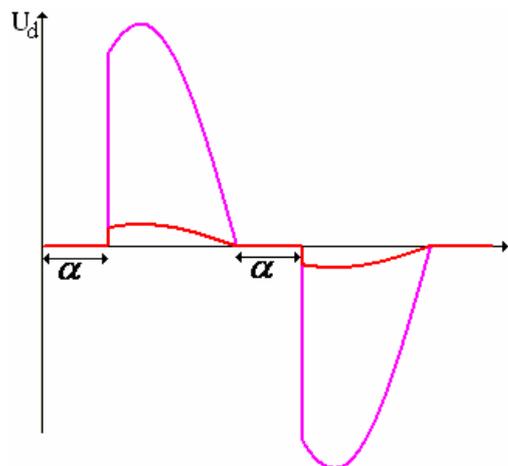


Figure (II.25): Tension, et courant délivrés par un gradateur monophasé sur une charge R.

Chapitre III:

Le logiciel conçu

Introduction

Le mot ELTSIM est composé de deux parties :

- ELT: abréviation de "**Electrotechnique**".
- SIM: abréviation de "**Simulation**".

On définir quelques mots liés a celui-ci :

L'Electrotechnique est l'étude des applications techniques de l'électricité, Son domaine d'intervention est la production, le transport, la distribution, le traitement, la transformation, la gestion et l'utilisation de l'énergie électrique.

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** \Leftrightarrow **Continue**.
- Conversion: **Continue** \Leftrightarrow **Continue**.
- Conversion: **Continue** \Leftrightarrow **Alternative**.
- Conversion: **Alternative** \Leftrightarrow **Alternative**.

III.1. Vue Utilisateur:

Vue Utilisateur

Introduction:

Un utilisateur de ELTSIM, voir le comme un logiciel de simulation sans intéresser a 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 ELTSIM pour expliquer comment comprendre leurs constitutions, leur fonctionnement, leurs options qui améliorent l'utilisation de celle-ci.

III.1.1. ELTSIM 1.0:

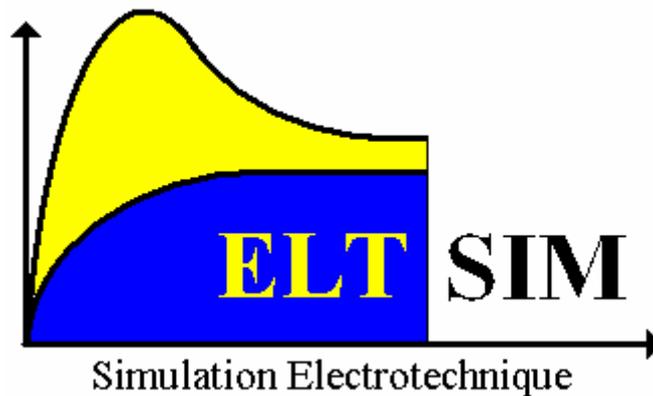


Figure (III.1): Le Logo de ELTSIM 1.0

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

Les circuits proposés dans ELTSIM 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 présentés dans ELTSIM pour couvrir presque tous les convertisseurs les plus essentiels dans le domaine de l'Electronique de puissance.

Pour chaque type de circuit à simuler, ELTSIM dispose pas mal des types d'alimentation du circuit et du type de charge à 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 ELTSIM, puis on expliquent les étapes à suivre pour faire une simulation dans celui-ci.

III.1.2.1. Premier contact avec ELTSIM 1.0

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

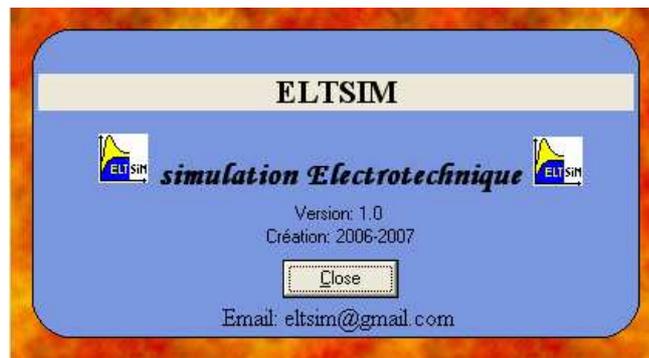


Figure (III.2): la capture d'écran qui apparaît À l'exécution de ELTSIM

Remarque: Click sur l'email affiché dans la capture d'écran, vous aurez l'ouverture de Outlook express avec l'email de ELTSIM et l'objet dans la fenêtre qui s'affiche, alors écrire votre email et le texte de message puis envoyer le message au fabricant de ELTSIM pour recevoir la réponse, a vos questions ou remarques. Une fois la capture d'écran s'affiche, click sur le bouton "Close" pour la fermeture de celle-ci et l'apparition de l'interface principale de ELTSIM.

III.1.2.2. Parcours rapide de l'environnement:

L'interface principale de ELTSIM 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).

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

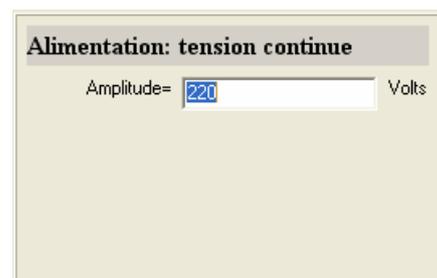


Figure (III.3): alimentation continue.

- ✓ La leur amplitude (en volts),
- ✓ La fréquence (en Hz),
- ✓ La phase initiale (en degrés).

- **Alimentation carrée:** (Fig.III.5) On

définit:

- ✓ Leur amplitude positive V_{m+} (en volts),
- ✓ Leur amplitude négative V_{m-} (en volts),
- ✓ La fréquence (en Hz),
- ✓ La phase initiale en degrés, et le rapport cyclique en %.

Figure (III.4): alimentation sinusoïdale.

Figure (III.5): alimentation carrée.

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 Cinq types de charge:

- Charge R: (Fig.III.6)

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

- Charge RL: (Fig.III.7)

Il faut définir:

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

- Charge RLE: (Fig.III.8)

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

- Charge RLE(t): (Fig.III.9)

- ✓ La résistance R (en Ohm),

Figure (III.6): Charge R.

Figure (III.7): Charge RL.

- ☞ 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).

- Charge MCC (machine a courant continu):

(Fig.III.10) 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:

- ☞ La tension d'alimentation de l'inducteur (tension continue) Ue (en Volts).
- ☞ Le moment d'inertie de L'ensemble Rotor-charge Jinertie (en Kg.m2).
- ☞ 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

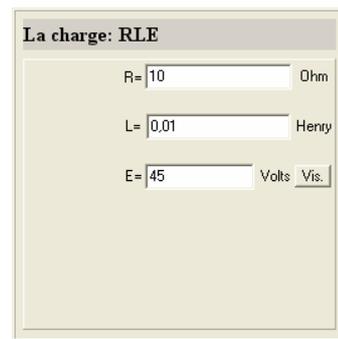


Figure (III.8): Charge RLE.

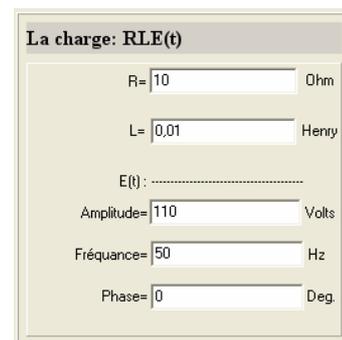


Figure (III.9): Charge RLE (t).

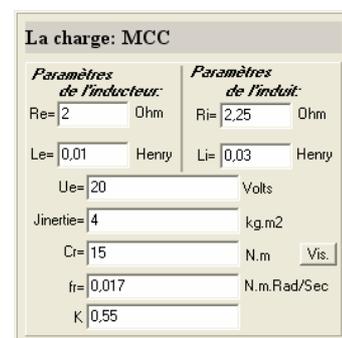


Figure (III.10): Charge MCC.

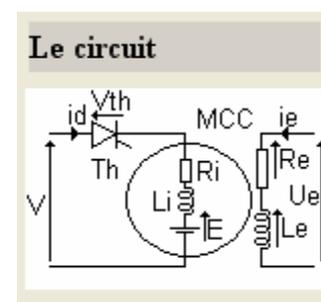


Figure (III.11): Le Circuit.

a simuler (fig.III.11).

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, sinon c'est la fem d'une charge RLE.

↳ I_e : le courant dans l'inducteur 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 l'angle d'amorçage des thyristors (fig.III.12).

- **Un Hacheur:** Pour un hacheur, on définit le rapport cyclique (en %), et la fréquence de hachage (en Hz) (fig. III.13)

- **Un Onduleur:** Pour un Onduleur, on définit la fréquence d'ondulation (en Hz) (commande pleine onde) (fig. III.14).

- **Un Gradateur:** Pour un Gradateur, on définit l'angle d'amorçage des thyristors montés en tête- bêche (fig. III.12).

Figure (III.12): Thyristor

Figure (III.13): Thyristor

Figure (III.14): Onduleur

III.1.2.2.5. La zone Visualisation:

La zone Visualisation (fig. III.15).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.

Signal.	Echelle:(Unité/div.)	val au balayeur.	Le temps:(Sec/div.)
<input checked="" type="checkbox"/> V	113,793	215,125	0,005000 Sec/div.
<input type="checkbox"/> i	aucun	aucun	
<input checked="" type="checkbox"/> Ud	113,793	215,125	
<input checked="" type="checkbox"/> Id	113,793	21,513	
<input type="checkbox"/> Vth1	aucun	aucun	
<input type="checkbox"/> Ith1	aucun	aucun	

Le temps au balayeur:(Sec) 0,0044 Vis.

Figure (III.15): La zone "Visualisation".

Chaque signal a cinq 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 toute 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.
- ☞ Un petit rectangle coloré à côté de la case à cocher pour désigner la couleur de la courbe du signal dessiné dans la zone du graphe.
- ☞ Echelle (Unité/div.): chaque signal dessiné dans la zone "graphe" a leur propre échelle affiché à coter de La case à cocher réservé pour ce signal, la lecture d'échelle s'effectué après l'affichage de la grille dans la zone "graphe" (fig.III.16).

Le mot "Unité" ne peut être que:

- **Volts:** si le signal est une tension.
- **Ampère:** si le signal est un courant.
- **Newton*mètre:** si le signal est un Couple.
- **Rad/Sec:** si le signal est une Vitesse.

- ☞ Un bouton à droite de l'échelle pour pouvoir changer l'échelle du signal donc le résultat de ce changement va répondre automatiquement et simultanément dans la zone du graphe.

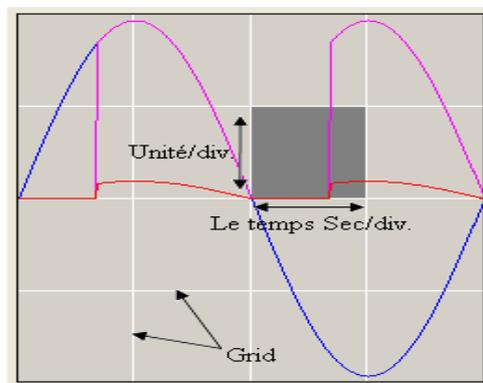


Figure (III.16): La lecture de l'échelle.

- ☞ Val au balayeur: c'est une méthode intelligente qui permet de mesurer l'amplitude du signal dans la zone du graphe au point où le balayeur est présenté et l'unité de la valeur affichée dans cette case est la même que celle définie au précédemment.

- ☞ Temps (Sec/div.): affiche l'échelle du temps dans la zone du graphe et cette échelle ne peut être utilisée que dans le cas où la grille est présentée dans la zone du graphe (fig.III.16).

- ☞ Le temps au balayeur (Sec): affiche le temps au point où le balayeur est présenté dans la zone du graphe.

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 **Graph**: pour lancer la Visualisation du graphe dans la zone appropriée.

☞ Un bouton **Grid**: pour présenter la grille dans la zone du graphe "grid en Anglais".

☞ Un bouton **Effacer**: pour effacer le dessin dans la zone du graphe

☞ **Intervalle du temps**: comporte deux cases:

- case **ti** (Sec): pour définir la borne inférieure de l'Intervalle du temps.

- case **tf** (Sec): pour définir la borne supérieure de l'intervalle du temps.

☞ Une liste contient des valeurs de l'Intervalle du temps en fonction de la période du signal périodique ("T" soit: la période de la tension d'alimentation dans le cas où elle est périodique, ou la période de hachage dans le cas de hacheur, ou la période d'ondulation dans le cas d'onduleur).

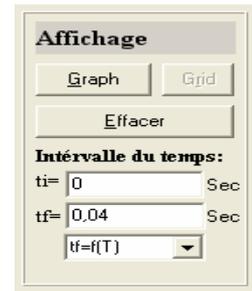


Figure (III.17): La zone "affichage".

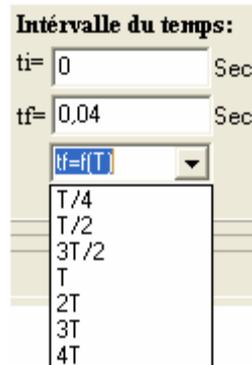


Figure (III.18): liste déroulante du temps.

III.1.2.2.7. La zone Graphe:

C'est la zone d'affichage des courbes à visualiser, on remarquons que cette zone ne comporte pas un 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.

Et pour plus de précision de mesure des grandeurs dans cette zone il existe:

■ Le balayeur: constitué d'un segment vertical pouvant glisser dans la zone du graphe pour donner au point où elle est située, les coordonnées de tout les signaux affichés a ce point.

L'utilisateur peut faire glisser le balayeur par l'intermédiaire des deux boutons gauche (pour glisser le balayeur vers la gauche) et droite (pour glisser le balayeur vers la droite).

■ La case du temps au pointeur de la souris: avec le déplacement de la souris sur la zone de graphe, on voit simultanément avec ce déplacement la valeur du temps qui correspond au point du pointeur de la souris affiché dans la barre de titre dans cette zone (fig.III.19).

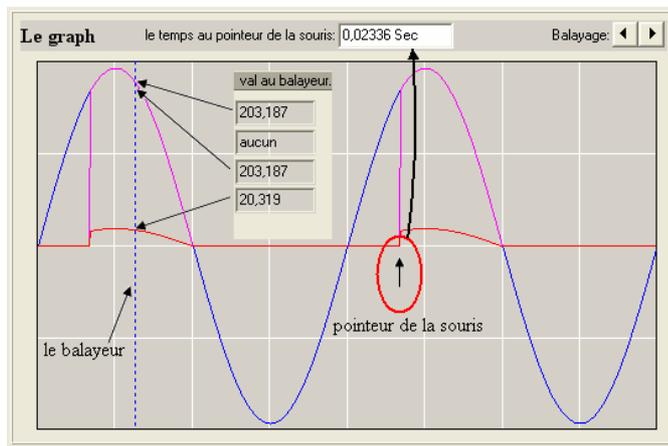


Figure (III.19): La lecture de l'échelle.

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.20). Ces grandeurs sont:

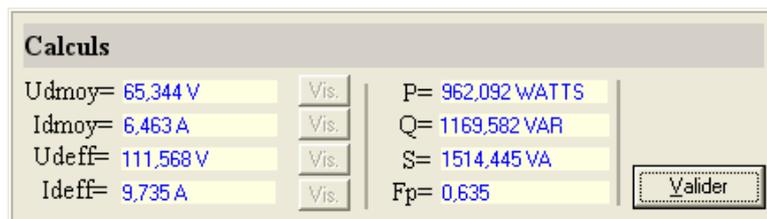


Figure (III.20): La zone de calcul à l'état d'affichage.

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:
$$Q = qV_1 I_1 \sin \phi_1$$

➤ La puissance apparente:
$$S = qV_{eff} I_{eff}$$

➤ La puissance déformante:
$$D = \sqrt{S^2 - P^2 - Q^2}$$

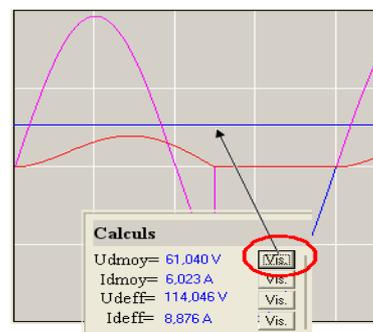


Figure (III.21): La visualisation des grandeurs affichés dans la zone "Calculs" dans la zone "Graphe".

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, 2:biphasé, 3:triphasé).

On peut visualiser les valeurs moyennes et efficaces de U_d et I_d dans la zone du graphe pour les comparer avec les formes instantanées des mêmes signaux par simple click sur le bouton "Vis" (signifier: visualiser) à côté de chaque grandeur.

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.22): La barre des menus.

Dans la barre des menus (Fig.III.22), 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 oblige 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.23), Alors faire le choix du convertisseur, il y a quatre types :

- Redresseur.
- Hacheur.
- Onduleur.
- Gradateur.



Figure (III.23): choix de type de convertisseur.

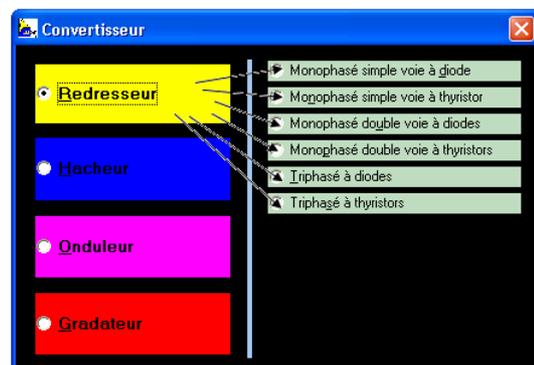


Figure (III.24): choix de sous type de convertisseur.

Chaque convertisseur englobe sous-types affichés lors du choix de celui-ci(Fig.III.24). Un choix d'un des sous types provoque la fermeture automatique de cette fenêtre.

Cliquez sur le menu alimentation, une fenêtre s'affiche (Fig.III.25) ou (Fig.III.26) ou (Fig.III.27) (selon le type de convertisseur), Alors faire le choix d'alimentation.

Il y a trois types d'alimentation: Tension sinusoïdale, Tension Carrée, Tension Continue



**Figure (III.25):
Alimantation forme1.**



**Figure (III.26):
Alimantation forme2.**



**Figure (III.27):
Alimantation forme3.**

Cliquez sur le menu "La charge", une fenêtre s'affiche (Fig.III.28), 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).



**Figure (III.28):
Choix de la charge.**

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.29) devient comme dans la figure (fig.III.30) :

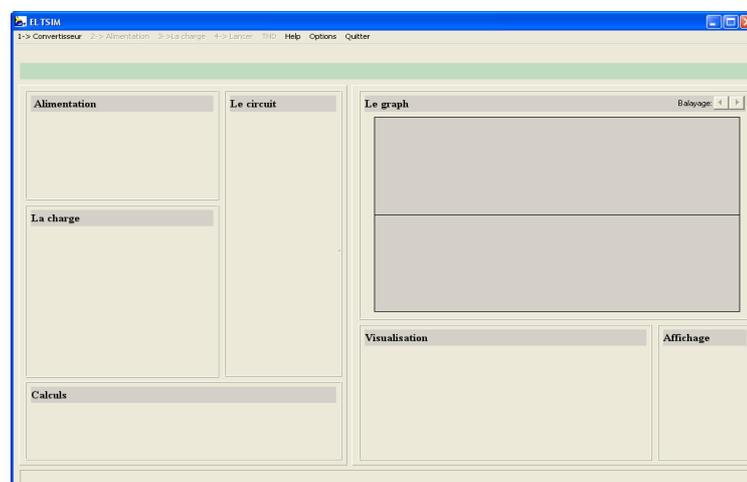


Figure (III.29): L'interface avant cliquer sur le bouton "Lancer".

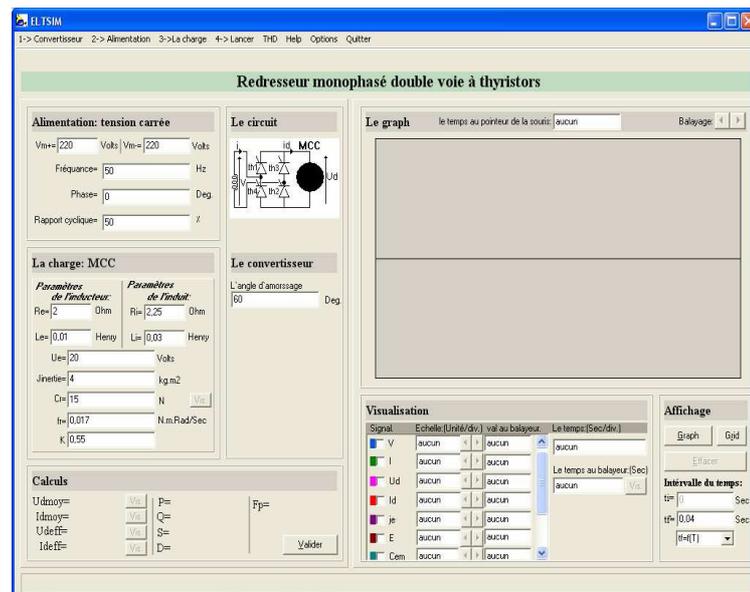


Figure (III.30): L'interface après avoir cliqué sur le bouton "Lancer".

Au premier lieu les choix par défaut vous permet 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.31).

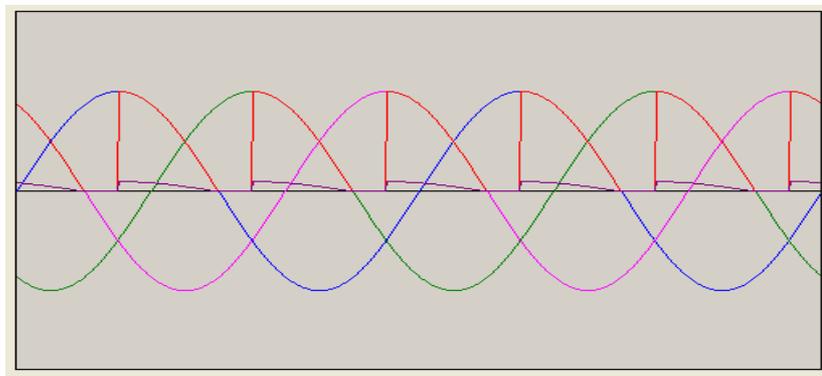


Figure (III.31): Affichage des signaux dans la zone du graphe.

L'utilisateur peut faire changer les paramètres de circuit dans les différents zones puis clique autre fois sur le bouton "Graphe", les résultats s'affichent dans la zone de graphe avec les nouveaux paramètres.

Cliquez sur le bouton "Valider" dans la zone "Calculs" pour voir les différents grandeurs liés à la source (P, Q, D, S, Fp) et la charge (Udmoy, Idmoy, Udeff, Ideff) (fig.III.32):

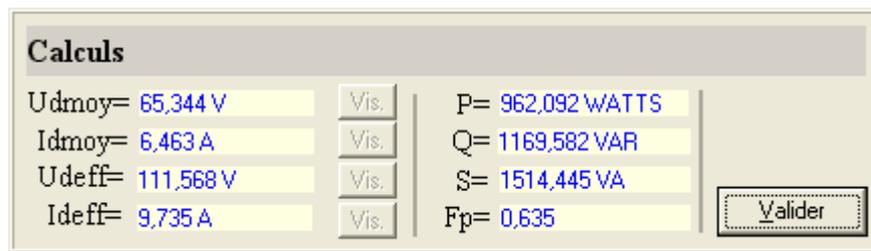


Figure (III.32): Affichage des résultats de calcul des grandeurs.

Chaque changements effectué sur les paramètres de circuits va provoquer un changement simultané sur les résultats dans la zone "Calculs".

III.1.4. Les Options de ELTSIM 1.0

Plusieurs options sont présentées dans ELTSIM 1.0, on cite:

- ✎ La possibilité de mettre des annotations manuscrites dans la zone du graphe (fig.III.33).

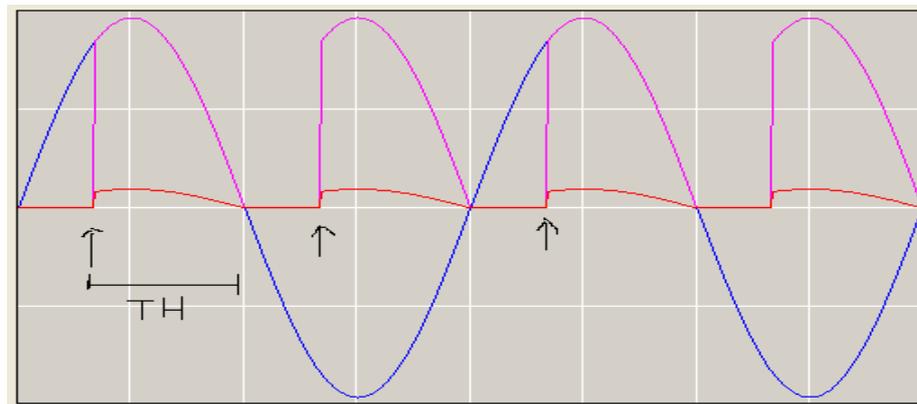


Figure (III.33): Affichage des résultats de calcul des grandeurs.

- ✎ La possibilité de changer la couleur de l'arrière plan de la zone du graphe, il y a quatre couleurs possible pour changer l'aperçu des signaux affichés dans cette zone (fig.III.34).

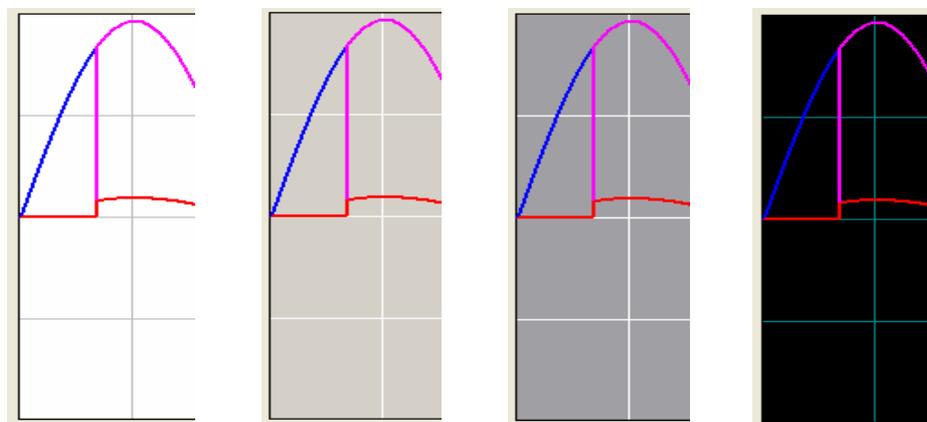


Figure (III.34): Les couleurs possibles pour l'arrière plan du graphe

Pour changer la couleur de l'arrière plan de la zone du graphe, dans le menu "Option" choisir "Options d'Affichage de graphe" puis "couleur d'arrière plan", une fenêtre s'affiche (fig.III.35), alors choisir une couleur parmi quatre couleurs disponibles (blanc- gris clair – gris foncé – noir), ces couleurs sont choisies de tel façon à avoir un bon aperçu des signaux dans la zone du graphe.

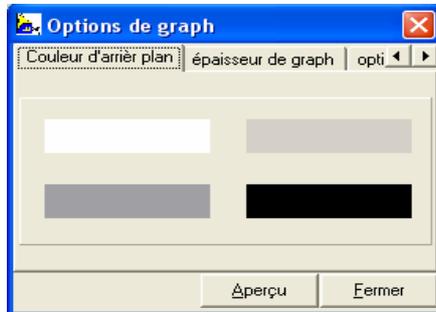


Figure (III.35): Le choix de l'arrière plan de la zone du graphe.

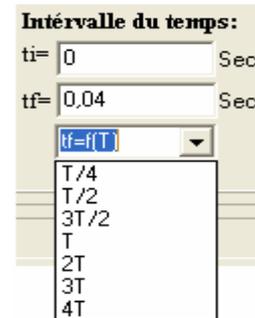


Figure (III.36): Simplicité de choix du temps de simulation.

✎ Pour chaque changement des paramètres d'une simulation donnée et dans n'importe quelle zone qui comporte des possibilités de changement par l'utilisateur, il suffit de cliquer sur le bouton entrée du clavier pour valider ces changements.

✎ On peut voir les signaux dans la zone du graphe avec une durée en fonction de nombre de périodes sans faire le calcul de la durée, tout simplement dans la zone "Affichage" faire un choix du nombre des périodes dans la liste "tf=f(T)" (fig.III.36).

✎ On peut changer les paramètres de la grille dans la zone du graphe, dans le menu "Option" choisir "Options d'Affichage de graphe" puis "Options de grid", une fenêtre s'affiche (fig.III.37), alors choisir le nombre des lignes horizontaux et verticaux de la grille ainsi le style des lignes.

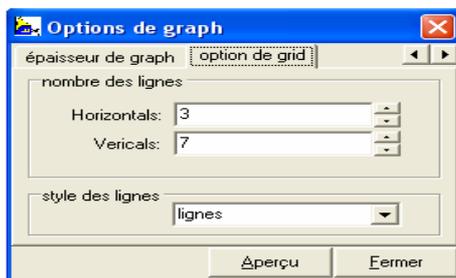


Figure (III.37): Le choix des paramètres de la grille.

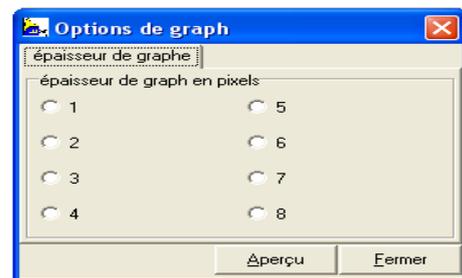


Figure (III.38): Le choix des paramètres de la grille.

✎ La possibilité de changer l'épaisseur de tracé des courbes, dans le menu "Option" choisir "Options d'Affichage du graphe" puis "épaisseur du graphe", une fenêtre s'affiche (fig.III.38) alors choisir l'épaisseur du graphe en "pixels".

On peut voir les harmoniques de la tension et du courant redressé, On choisit dans la barre des menus le menu "THD", une fenêtre s'affiche (fig.III.39) alors les harmoniques sont représentés dans cette fenêtre ce forme des barres en pourcentage, dans cette fenêtre on peut choisir le nombre des harmoniques à afficher.

Pour plus de détaille on peut voir une liste contient les informations sur les harmoniques, on avoir cette liste par simple clique sur le bouton "liste" dans la fenêtre précédent (fig.III.40):

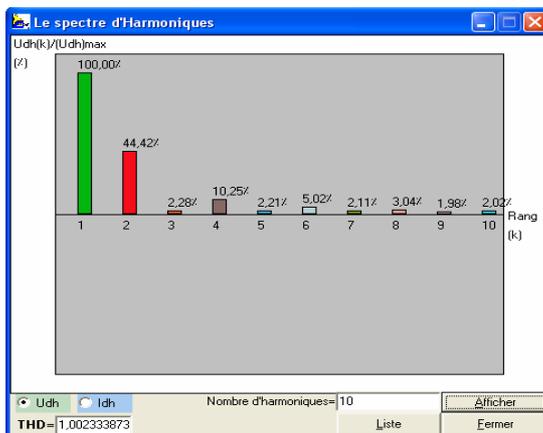


Figure (III.39): Les harmoniques.

N°harmonique	fréq(Hz)	module 'max'	module 'eff'	phase 'Rad'	phase 'Deg'	%/max
1	50,000000	110,485329	78,124925	0,022800	1,306316	100,000000
2	100,000000	49,076354	34,702223	1,552585	88,956547	44,418888
3	150,000000	2,520420	1,782206	1,032214	59,141532	2,281226
4	200,000000	11,326638	8,009143	1,425746	81,689224	10,251712
5	250,000000	2,443004	1,727465	0,671191	38,456433	2,211157
6	300,000000	5,542007	3,918790	1,180211	67,621108	5,016057
7	350,000000	2,330445	1,647873	0,307040	17,592101	2,109280
8	400,000000	3,359275	2,375366	0,838847	48,062403	3,040472
9	450,000000	2,186939	1,546399	-0,061867	-3,544709	1,979393
10	500,000000	2,227112	1,574806	0,412330	23,624771	2,015754

Figure (III.40): Liste des harmoniques.

Des nombreuses opérations possèdent des équivalents clavier, on l'utilises par clique simultané sur la touche "Alt" et le Lettre spécifique de l'opération désirée (lettre soulignée) (fig.III.41).



Figure (III.41): Les équivalents clavier.

On peut sauvegarder des résultats obtenus par une simulation dans un fichier pour permettre de récupérer ces résultats ultérieurement.

III.2. Vue programmeur:

Vue programmeur

Introduction:

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

Le programme principal constitué d'un ensemble de procédures et de fonctions 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 rk4:

La procédure "rk4" résolve l'équation différentielle: $\frac{dI_d}{dt} = \frac{1}{L}(U_d - E - RI_d)$

(La signification des symboles est présentée dans la section 1 de ce chapitre).

La résolution se fait par la méthode de Runge- Kutta d'ordre 4. Cette procédure est utilisée dans le cas des charges RLE et RL (avec E=0).

Le code source de la procédure est le suivant:

1	procedure tform1.rk4(var eq,i:integer;ti,idf0,v0,h:real;var idf:real);
2	var r,l,e,em,fe,phie,f0,f1,f2,f3:real;
3	begin
4	r:=strtofloat(edit5.text);
5	l:=strtofloat(edit6.text);
6	e:=0;
7	if form1.label8.caption='La charge: RLE' then e:=strtofloat(edit7.text);
8	if form1.label8.caption='La charge: RLE(t)' then begin
9	r:=strtofloat(edit73.Text);
10	l:=strtofloat(edit74.Text);
11	em:=strtofloat(edit75.Text);
12	fe:=strtofloat(edit76.Text);
13	phie:=strtofloat(edit77.Text);
14	e:=em*sin(2*pi*fe*ti+phie);
15	end;
16	if eq=0 then begin //v0<--V,idf<---id.
17	f0:=(1/l)*v0-(1/l)*e-(r/l)*idf0;
18	f1:=(1/l)*v0-(1/l)*e-(r/l)*(idf0+h/2*f0);
19	f2:=(1/l)*v0-(1/l)*e-(r/l)*(idf0+h/2*f1);
20	f3:=(1/l)*v0-(1/l)*e-(r/l)*(idf0+h*f2);
21	idf:=idf0+h/6*(f0+2*f1+2*f2+f3);
22	end;end;

III. 2.1.2. La procédure rk4mcc:

La procédure "rk4mcc" résolve le système d'équations différentielles associé au modèle d'une machine à courant continu:

$$\left. \begin{aligned} \frac{dJ_e}{dt} &= \frac{1}{L_e}(U_e - R_e J_e) \\ \frac{dI_d}{dt} &= \frac{1}{L_i}(U_d - E - R_i I_d) \\ \frac{d\Omega}{dt} &= \frac{1}{J}(C_{em} - \text{signe}(\Omega)C_r - f_r \Omega) \end{aligned} \right\}$$

(La signification des symboles est présentée dans la section 1 de ce chapitre).

La résolution se fait par la méthode de Runge- Kutta d'ordre 4. Cette procédure est utilisée dans le cas de la charge MCC (machine à courant continu).

Le code source de la procédure est le suivant:

```

1  procedure tform1.rk4mcc(var eq:integer;h,v0,id0,e0,cem0:real;var
2  id:real);
3  var re,le,ri,li,ue,jin,cr,fr,f0,f1,f2,f3,signv:real;
4  begin
5  re:=strtofloat(edit29.text);
6  le:=strtofloat(edit30.text);
7  ri:=strtofloat(edit31.text);
8  li:=strtofloat(edit32.text);
9  ue:=strtofloat(edit24.text);
10 jin:=strtofloat(edit25.text);
11 cr:=strtofloat(edit26.text);
12 fr:=strtofloat(edit27.text);
13  //-----
14  if eq=1 then begin //u<--0,id<---je.
15  f0:=(1/le)*ue-(re/le)*id0;
16  f1:=(1/le)*ue-(re/le)*(id0+h/2*f0);
17  f2:=(1/le)*ue-(re/le)*(id0+h/2*f1);
18  f3:=(1/le)*ue-(re/le)*(id0+h*f2);
19  id:=id0+h/6*(f0+2*f1+2*f2+f3);
20  end;//-----
21  if eq=2 then begin //u<--ud,id<---id.
22  f0:=(1/li)*(v0-e0-ri*id0);
23  f1:=(1/li)*(v0-e0-ri*(id0+h/2*f0));
24  f2:=(1/li)*(v0-e0-ri*(id0+h/2*f1));
25  f3:=(1/li)*(v0-e0-ri*(id0+h*f2));
26  id:=id0+h/6*(f0+2*f1+2*f2+f3);
27  end;//-----
28  if eq=3 then begin //u<--ud,id<---vitesse(omiga).
29  if id0<0 then signv:=-1
30  else signv:=1;
31  f0:=(1/jin)*(cem0-signv*cr-fr*id0);
32  f1:=(1/jin)*(cem0-signv*cr-fr*(id0+h/2*f0));
33  f2:=(1/jin)*(cem0-signv*cr-fr*(id0+h/2*f1));
34  f3:=(1/jin)*(cem0-signv*cr-fr*(id0+h*f2));
35  id:=id0+h/6*(f0+2*f1+2*f2+f3);
36  end;end;
```

III. 2.1.3. La procédure de calcul d'une intégrale (integral):

La procédure " **integral**" calcule les intégrales numériques par la méthode de SIMPSON.

Le code source de la procédure est le suivant:

```

1  procedure tform1.integral(var h:real;var int:real);
2  var i,m:integer;
3  s:real;
4  begin
5  s:=0;
6  i:=1;
7  m:=100;
8  while i<=2*m do begin
9  s:=s+fi[i]+4*fi[i+1]+fi[i+2];
10 inc(i,2);
11 end;
12 int:=h/6*s;
13 end;
    
```

III. 2.1.4. La procédure fourier5:

La procédure " **fourier5**" calcule les paramètres de série de FOURIER (A_n , B_n) de I_d et U_d ainsi que l'amplitude et la phase de chaque harmonique de celui-ci (lignes:8 à 40 du la procédure en bas) puis elle va afficher les paramètres dans un tableau (fig.III.42) (lignes:41-59) et dessiner les amplitudes des harmoniques jusqu'à l'harmonique de rang k de U_d ou I_d selon le choix de l'utilisateur comme indiqué à la figure(fig.III.43) (lignes:60-83).

N°harmonique	fréq(Hz)	module "max	module "eff	phase "Rad"	phase "Deg"	%/max
1	50,000000	110,485329	78,124925	0,022800	1,306316	100,000000
2	100,000000	49,076354	34,702223	1,552585	88,956547	44,418888
3	150,000000	2,520420	1,782206	1,032214	59,141532	2,281226
4	200,000000	11,326638	8,009143	1,425746	81,688224	10,251712
5	250,000000	2,443004	1,727465	0,671191	38,456433	2,211157

Figure (III.42): Liste des harmoniques.

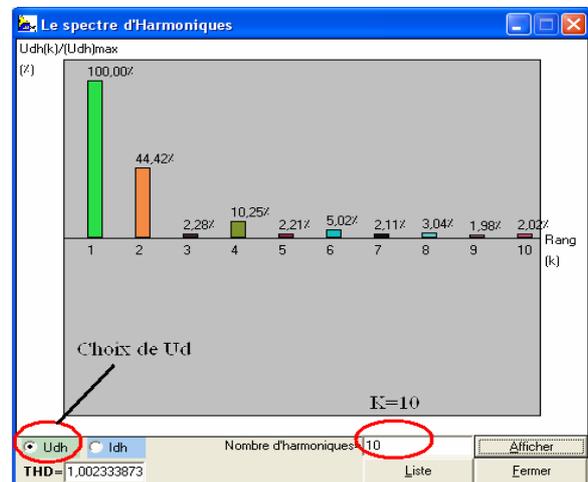


Figure (III.43): Les harmoniques.

Le code source de la procédure est le suivant:

```

1  procedure TForm1.fourier5(var c,m:integer;h0,f:real);
2  var i,j, xm0,xm1, ym0,ym1:integer;
3  t0,w,int0,max0,pourcent,moy,eff,phideg,thd:real;
4  ac,bc:array[1..500,1..100] of real;
5  ft,xn:array[1..500] of real;
6  begin
    
```

```

7      w:=2*pi*f;
8      //calcul des paramètres de série de FOURIER :
9      for i:=1 to 2*m+1 do ft[i]:=fi[i];
10     //calcul de a0:
11     integral(h0,int0);
12     moy:=int0*f;
13     //calcul de la valeur efficace:
14     for i:=1 to 2*m+1 do fi[i]:=sqr(ft[i]);
15     integral(h0,int0);
16     eff:=sqrt(abs(int0*f));
17     //calcul de an et bn:
18     for j:=1 to c do begin
19     for i:=1 to 2*m+1 do begin
20     t0:=(i-1)*h0;
21     ac[i,j]:=ft[i]*sin(j*w*t0);
22     bc[i,j]:=ft[i]*cos(j*w*t0);
23     end;
24     for i:=1 to 2*m+1 do fi[i]:=ac[i,j];
25     integral(h0,int0);
26     an[j]:=2*f*int0;
27     for i:=1 to 2*m+1 do fi[i]:=bc[i,j];
28     integral(h0,int0);
29     bn[j]:=2*f*int0;
30     xn[j]:=sqrt(sqr(an[j])+sqr(bn[j]));
31     xheff[j]:=xn[j]/sqrt(2);
32     if an[j]<>0 then phih[j]:=arctan(bn[j]/an[j])
33     else begin
34     if bn[j]>0 then phih[j]:=pi/2
35     else phih[j]:=-pi/2;
36     end;end;
37     thd:=xn[1]/eff;
38     if calfour=1 then begin
39     //THD:
40     form5.edit3.Text:=floattostr(thd);
41     //liste des harmoniques:
42     with form6.stringgrid1 do begin
43     Cells[0,0]:='N°harmonique';
44     Cells[1,0]:='fréq(Hz)';
45     Cells[2,0]:='module "max"';
46     Cells[3,0]:='module "eff"';
47     Cells[4,0]:='phase "Rad"';
48     Cells[5,0]:='phase "Deg"';
49     Cells[6,0]:='%/max';
50     rowcount:=c+1;
51     for i:=1 to c do begin
52     phideg:=(180/pi)*phih[i];
53     Cells[0,i]:=inttostr(i);
54     Cells[1,i]:=formatfloat('0.000000',f*i);
55     Cells[2,i]:=formatfloat('0.000000',xn[i]);
56     Cells[3,i]:=formatfloat('0.000000',xheff[i]);
57     Cells[4,i]:=formatfloat('0.000000',phih[i]);

```

58	Cells[5,i]:=formatfloat('0.000000',phideg);
59	end;end;
60	//affichage de spectre:
61	for i:=1 to c do x0[i]:=xn[i]/eff;
62	maxn(c,max0);
63	for i:=1 to c do begin
64	xm0:=40+trunc(400*(i-0.5)/c);
65	//ym0:=40+trunc(150-(150*xn[i]/max0));
66	ym0:=40+trunc(150-(150*(xn[i]/eff)/max0));
67	xm1:=trunc(xm0+(400/(3*c)));
68	ym1:=40+150;
69	with form5.image2.Canvas do begin
70	//les graphiques:
71	pen.width:=sizepen;
72	pen.Color:=clBlack;
73	brush.Color:=rgb(random(255),random(255),random(255));
74	brush.Style:=bsSolid;
75	rectangle(xm0,ym0,xm1,ym1);
76	//les textes:
77	pourcent:=((xn[i]/eff)/max0)*100;
78	form6.stringgrid1.Cells[6,i]:=formatfloat('0.000000',pourcent);
79	ym0:=ym0-15;
80	brush.Style:=bsclear;
81	if c<=10 then textout(xm0,ym0,formatfloat('0.00',pourcent)+'%');
82	ym0:=190+5;
83	if c<=30 then textout(xm0,ym0,inttostr(i));
84	end;end;end;end;

III. 2.1.5. La procédure maxn:

La procédure " **maxn** " calcule le maximum d'un vecteur de dimension n et la variable max1 reçoit ce maximum (ligne 7).

Le code source de la procédure est le suivant:

1	procedure tform1.maxn(var n:integer;var max1:real);
2	var i,j:integer;
3	begin
4	for i:=1 to n do max1:=x0[i];
5	for j:=1 to n do begin
6	if max1<x0[j] then begin
7	max1:=x0[j];
8	end;end;
9	for i:=1 to n do x0[i]:=0;
10	end;

III. 2.1.6. La fonction imp:

La fonction " **imp** " envoie la valeur 0 ou 1 (lignes 13 et 14) selon l'état du temps avant ou après l'angle d'amorçage définie par l'utilisateur dans le cas où l'interrupteur est un thyristor.

f: la fréquence de la tension d'alimentation (ligne 4 voir la procédure en bas).

phi: la phase initiale de la tension d'alimentation (ligne 5).

alpha: l'angle d'amorçage (ligne 8).

tp: la position du temps dans une période (avant ou après l'angle d'amorçage (ligne 12))

Le code source de la fonction est le suivant:

1	function tform1.imp(t:real):integer;
2	var f,alpha,tp,phi:real;
3	begin
4	f:=strtofloat(form1.edit2.text);
5	phi:=strtofloat(form1.edit3.text);//phi en deg.
6	phi:=(pi/180)*phi; //phi en rad.
7	phi:=(1/(2*pi*f))*phi; //phi en sec.
8	alpha:=strtofloat(edit22.text); //alpha en deg.
9	alpha:=(pi/180)*alpha; //alpha en rad.
10	alpha:=(1/(2*pi*f))*alpha; //alpha en sec.
11	t:=t+phi; //rotard de phi.
12	tp:=(t*f-trunc(t*f))/f;
13	if tp<alpha then result:=0
14	else result:=1;
15	end;

III. 2.1.7. La fonction v (t):

La fonction " **v (t)** " nous donne la valeur de la tension d'alimentation à l'instant "t" quelque soit le type définie par l'utilisateur pour celle-ci. Dans les lignes 13,17 et 26 il y a le test de type de la tension d'alimentation pour choisir la formule de calcul de celle-ci à l'instant "t".

Le code source de la fonction est le suivant:

1	function tform1.v(t:real):real;
2	var vm,vmp,vmm,phi,f,rc,w,tp:real;
3	begin
4	vm:=strtofloat(form1.edit1.text);
5	vmp:=strtofloat(form1.edit28.text);
6	vmm:=strtofloat(form1.edit33.text);
7	phi:=strtofloat(form1.edit3.text);//phi en deg.
8	f:=strtofloat(form1.edit2.text);
9	rc:=strtofloat(form1.edit4.text);
10	w:=2*pi*f;
11	phi:=(pi/180)*phi; //phi en rad.
12	//la tension d'alimentation V=x1:-----
13	if form1.label2.caption='Alimentation: tension sinusoïdale' then begin
14	result:=vm*sin(w*t+phi);

```

15   end;
16   //-----
17   if form1.label2.caption='Alimentation: tension carrée' then begin
18   phi:=(1/(2*pi*f))*phi; //phi en sec.
19   rc:=rc/(100*f);
20   t:=t+phi;
21   tp:=(t*f-trunc(t*f))/f;
22   if tp<=rc then result:=vmp
23   else result:=-vmm;
24   end;
25   //-----
26   if form1.label2.caption='Alimentation: tension continue' then begin
27   result:=vm;
28   end;end;

```

III. 2.1.8. La procédure plot:

La procédure " **plot** " permet de dessiner un segment dans la zone de graphe (appelée canvas) entre les coordonnées (x11,y11) (ligne 16) et (x22,y22) (ligne 17), alors l'appel successive de cette procédure nous permet d'avoir la forme de la courbe de signal a affiché dans la zone du graphe.

Le code source de la procédure est le suivant:

```

1   procedure tform1.plot(var ampli0:real);
2   var x11,x22,yi,y11,yj,y22,i,n:integer;
3   begin
4   n:=499;
5   image2.Canvas.pen.Width:=strtoint(form7.Edit1.Text);
6   for i:=1 to n do begin
7   x0[i]:=ampli0*x0[i];
8   end;
9   for i:=1 to n-1 do begin
10  x11:=i;
11  yi:=trunc(x0[i]);
12  y11:=150-yi;
13  x22:=i+1;
14  yj:=trunc(x0[i+1]);
15  y22:=150-yj;
16  image2.Canvas.moveto(x11,y11);
17  image2.Canvas.lineto(x22,y22);
18  end;
19  image2.Canvas.pen.Width:=1;
20  end;

```

III. 2.1.9. La procédure izeone:

La procédure " **izone** " permet d'effacer le contenu de la zone du graphe. La couleur utilisée ici pour l'arrière plan de la zone du graphe est celle définie par l'utilisateur dans le menu "option" (voir la section précédente) (lignes 6,7,8 et 9).

Le code source de la procédure est le suivant:

1	procedure tform1.izone;
2	begin
3	with image2.Canvas do begin
4	pen.width:=sizepen;
5	pen.Color:=clBlack;
6	if form7.Edit2.Text=inttostr(1) then brush.Color:=form7.Label3.Color;
7	if form7.Edit2.Text=inttostr(2) then brush.Color:=form7.Label4.Color;
8	if form7.Edit2.Text=inttostr(3) then brush.Color:=form7.Label5.Color;
9	if form7.Edit2.Text=inttostr(4) then brush.Color:=form7.Label6.Color;
10	//brush.Color:=clBlack;
11	brush.Style:=bsSolid;
12	rectangle(0,0,500,300);
13	moveto(0,150);
14	lineto(500,150);
15	end;end;

III. 2.1.10. La procédure grid:

La procédure " **grid** " dessine la grille dans la zone de graphe. Le nombre des lignes verticales et horizontales est défini par l'utilisateur dans le menu "option" (voir la section précédente) (lignes 6, 7,8 et 9).

n1: le nombre de lignes horizontales (ligne 37).

n2: le nombre de lignes verticales (ligne 38).

Le code source de la procédure est le suivant:

1	procedure tform1.grid;
2	var i,j,i1,j1,n1,n2:integer;
3	m1,m2:real;
4	sty:string;
5	begin
6	izone;
7	sty:=form7.ComboBox1.Text;
8	With image2.Canvas do begin
9	if form7.Edit2.Text=inttostr(1) then pen.Color:=form7.Label3.Color;
10	if form7.Edit2.Text=inttostr(2) then pen.Color:=form7.Label4.Color;
11	if form7.Edit2.Text=inttostr(3) then pen.Color:=form7.Label5.Color;
12	if form7.Edit2.Text=inttostr(4) then pen.Color:=form7.Label6.Color;
13	pen.style:=pssolid;
14	moveto(0,150);
15	lineto(500,150);
16	if form7.Edit2.Text=inttostr(1) then pen.Color:=clsilver;
17	if form7.Edit2.Text=inttostr(2) then pen.Color:=clWhite;

18	if form7.Edit2.Text=inttostr(3) then pen.Color:=clWhite;
19	if form7.Edit2.Text=inttostr(4) then pen.Color:=clteal;
20	if sty='Uni' then begin
21	pen.style:=pssolid;
22	end;
23	if sty='Tiret' then begin
24	pen.style:=psDash;
25	end;
26	if sty='Point' then begin
27	pen.style:=psDot;
28	end;
29	if sty='Tiret-point' then begin
30	pen.style:=psDashDot;
31	end;
32	if sty='Tiret-point-point' then begin
33	pen.style:=psDashDotDot;
34	end;
35	brush.Color:=clnone;
36	brush.Style:=bsclear;
37	n1:=strtoint(form7.edit3.text);
38	n2:=strtoint(form7.edit4.text);
39	m1:=300/(n1+1);
40	m2:=500/(n2+1);
41	if n2<>0 then begin
42	for i:=1 to n2 do begin
43	i1:=trunc(i*m2);
44	moveto(i1,0);
45	lineto(i1,300);
46	end;end;
47	if n1<>0 then begin
48	for j:=1 to n1 do begin
49	j1:=trunc(j*m1);
50	moveto(0,j1);
51	lineto(500,j1);
52	end;end;
53	if n1<>1 then begin
54	pen.style:=psDot;
55	moveto(0,150);
56	lineto(500,150);
57	end;
58	pen.style:=pssolid;
59	pen.Color:=clBlack;
60	rectangle(0,0,500,300);
61	pen.Color:=clBlack;
62	end;end;

III. 2.2. La structure globale du programme:

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

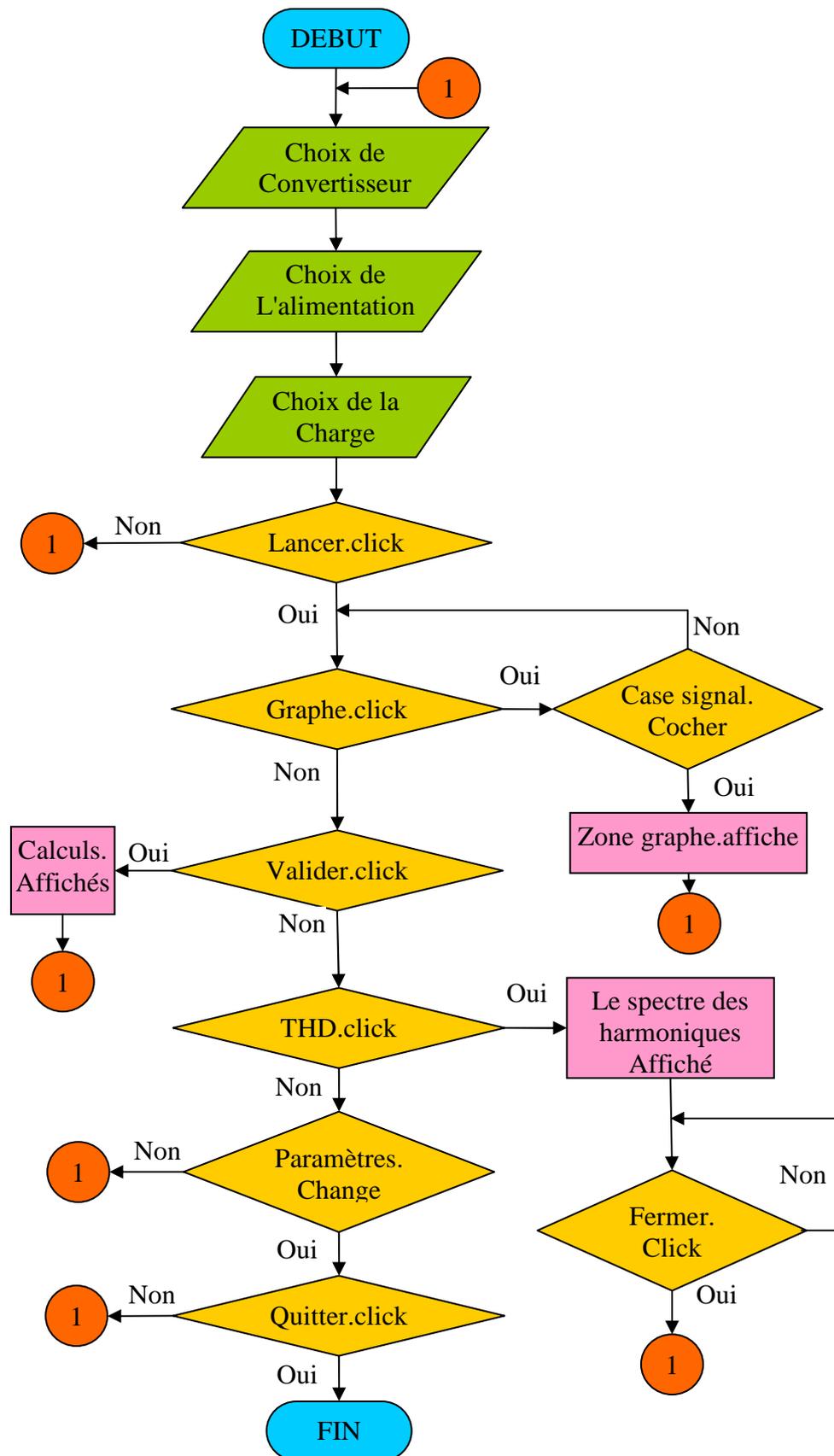


Figure (III.44): L'organigramme du fonctionnement du logiciel.

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: choix du convertisseur, choix de l'alimentation, et choix de la charge. Puis, on lance la validation du choix par le bouton menu "Lancer" (Lancer.click: c'est l'événement qui simule un click 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 click 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, l'événement "THD.click" simule un click sur le bouton menu THD pour ouvrir une nouvelle fenêtre réservé pour tout qui concerne les calculs des harmoniques. Le programme reste a l'état d'attente à ce point jusqu'à un click sur le bouton "Fermer" 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 chaque changement au début pour refaire l'exécution avec les nouveaux paramètres, dans ce cas l'utilisateur peut voir les résultats des changements par un click sur la touche "Entrée" du clavier, a ce point une procédure est réservé pour refaire l'exécution des autres procédure concernait par cette changement:

1	procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
2	begin
3	if key=#13 then
4	begin
5	if igraphe=1 then button2.Click;
6	if ical=1 then button1.Click;
7	end;
8	If key = '.' then key:= DecimalSeparator;
9	if not (Key in['.',',','0'..'9',Chr(8)]) then Key:= #0;
10	end;

Cette procédure liée à l'événement on change de chaque composant permettre d'effectuer un changement sur les paramètres de simulation, elle a trois rôles:

1- effectuer un test de click sur la touche "Entrée" du clavier (ligne 3) si le résultat est positif alors on va effectuer un deuxième test, c'est la détection d'un graphe afficher dans la zone du graphe (ligne 5) si le résultat logique est "true" donc va simuler un click sur le bouton 2 (c'est le nom du bouton "graphe" dans le code source de logiciel) pour lancer les procédures concernées par le changement effectué, un troisième test effectuer de la même façon (ligne 6) mais dans ce cas va faire une détection d'un affichage dans la zone de calculs, et "bouton1" c'est le nom du bouton "Valider" dans le code source de logiciel.

2- effectuer un test en parallèle avec l'utilisateur qui va saisir un nombre réel dans un "edit" (zone de saisir) et ne permet d'écrire que des chiffres de 0 à 9 et de virgule, car n'est pas logique de taper un caractère dans un "edit" (ligne 9).

3- effectuer un test sur la nature de séparateur saisi par l'utilisateur pour séparer la partie décimale de la partie entier d'un nombre réel (point ou virgule) (ligne 8) après le test le séparateur toujours remettre à le symbole de virgule.

Chapitre VI:

Validation des résultats

Introduction:

Au but de savoir la précision d'*Eltsim* sur la simulation des différents circuits, et les allures des tensions et des courants qui lui donne, nous allons fait dans ce chapitre une validation des résultats sur les quatre types des convertisseurs, alimentant plusieurs types de charges.

Une comparaison entre les allures des tensions et des courants obtenus par *Eltsim* et celui obtenus par *Simplorer* et faite, pour voir la proche de *Eltsim* a *Simplorer* de côté de résolution des équations par la méthode de Runge- Kutta.

Une deuxième comparaison entre les calculs des puissances qui donne *Eltsim*, et celui obtenus par *Simplorer* est aussi faite, pour voir la proche de *Eltsim* de *Simplorer* de côté de intégration numérique par la méthode de Simpson.

Dans la simulation des quatre types des convertisseurs, on prend un temps de simulation des signaux égale à 40ms c'est-à-dire deux périodes (1.8ms/div pour *Eltsim* pour les charges R, RL, RLE, et RLE (t)), sachant que *Simplorer* utilise les unités [Volts] pour les tensions, [Ampère] pour les courants, et [s] pour le temps.

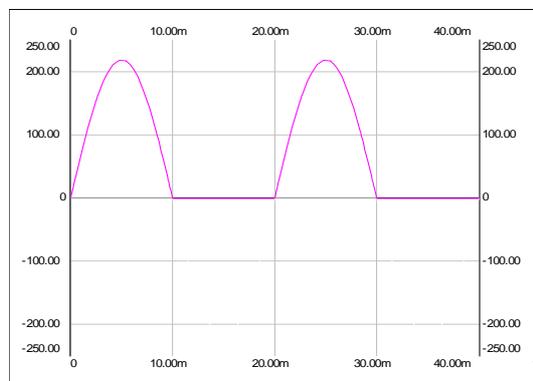
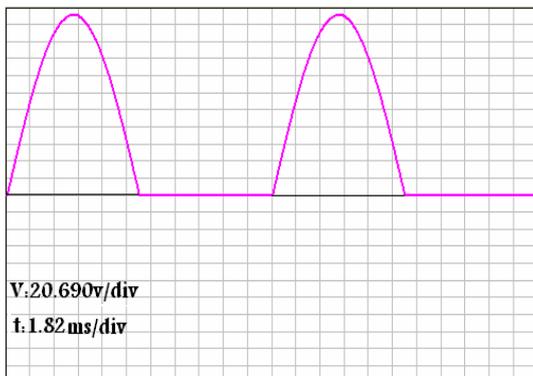
IV.1. Redressement monophasé simple à diode.

Charge R:

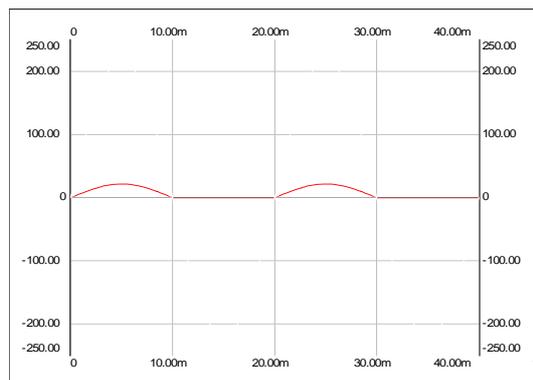
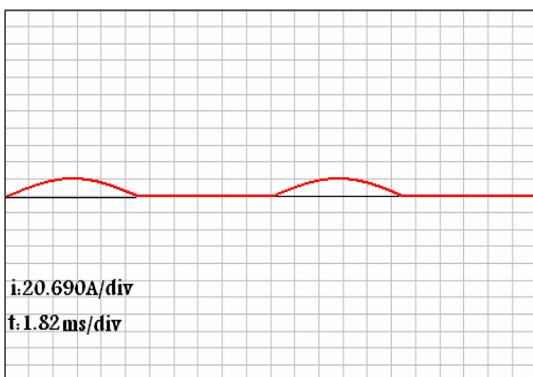
ELTSIM

SIMPLORER

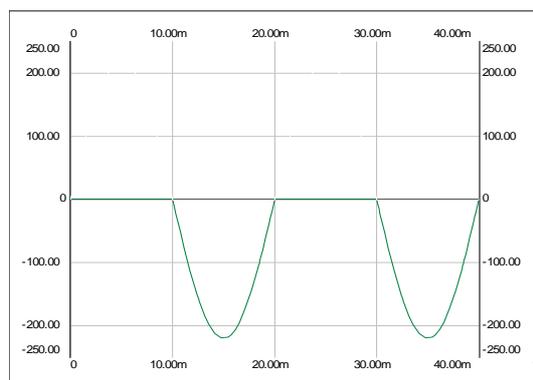
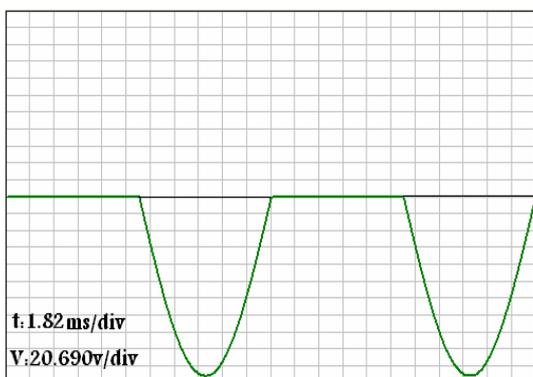
Tension redressée



Courant redressé



Tension inverse aux bornes de la diode



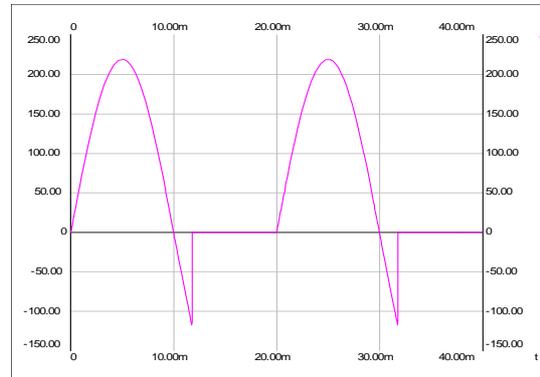
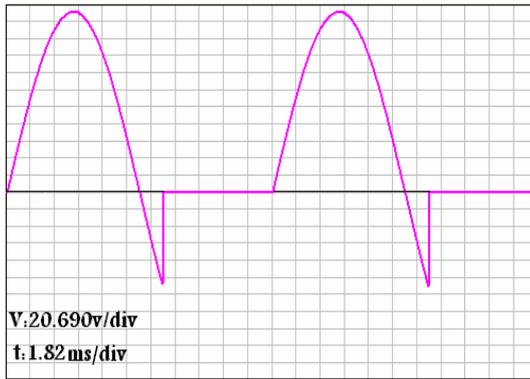
Figure(IV.1):Tension redressée, courant et tension inverse aux bornes de D
 $V_m=220V$, $R=10\Omega$, $f=50Hz$, et $t=40ms$

Charge RL:

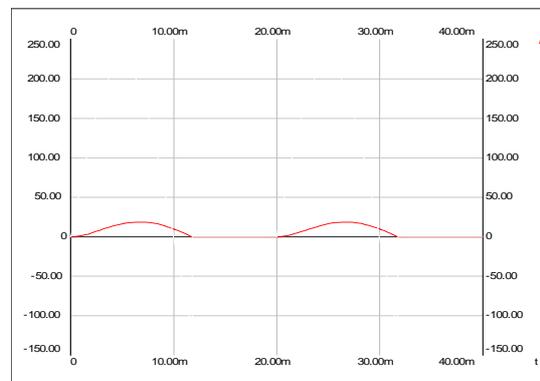
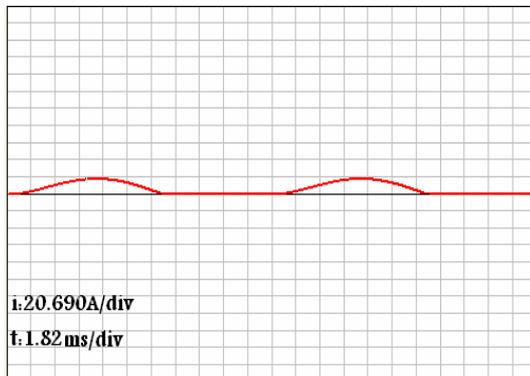
ELTSIM

SIMPLORER

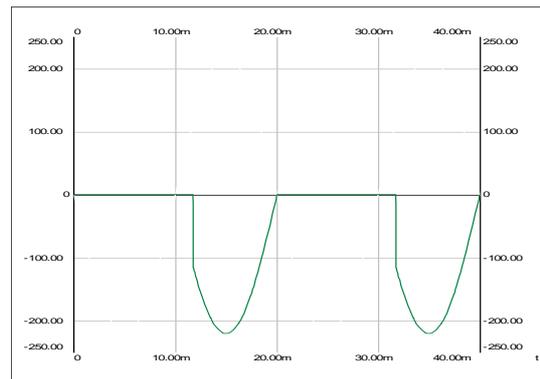
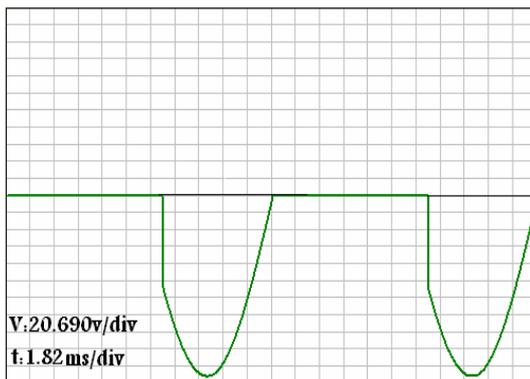
Tension redressée



Courant redressé



Tension inverse aux bornes de la diode

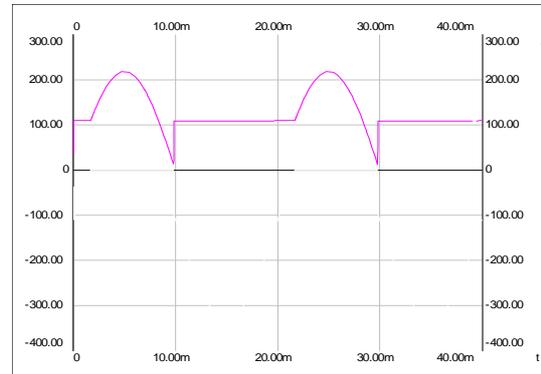
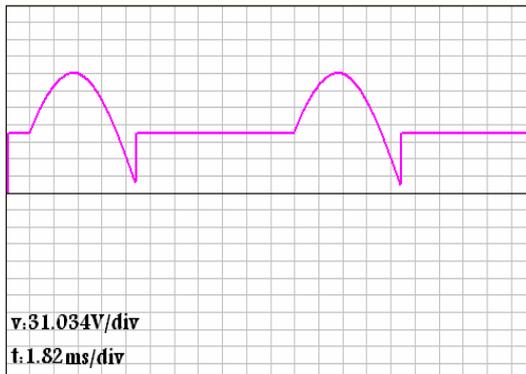
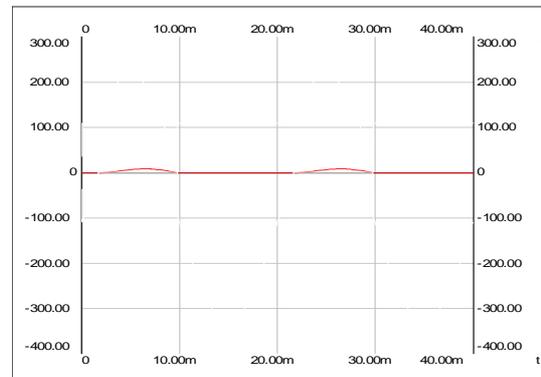
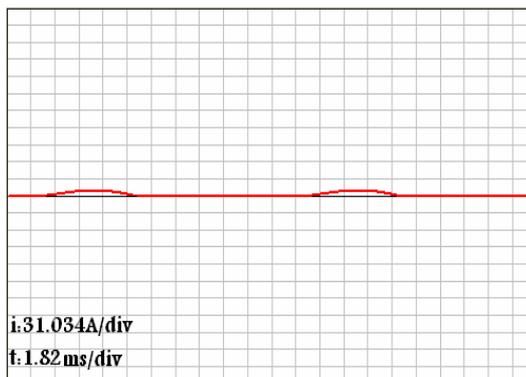
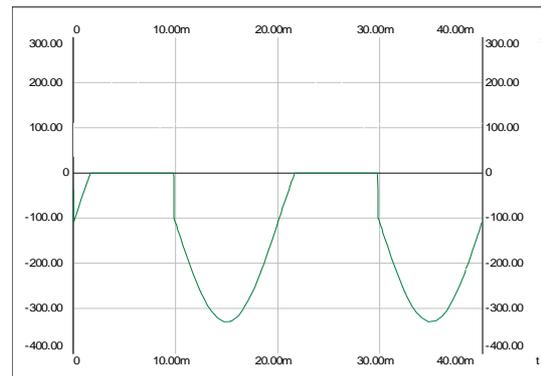
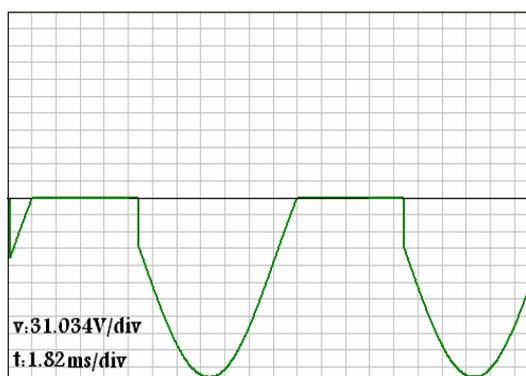


Figure(IV.2): Courant, tension redressée et tension inverse aux bornes de D
 $V_m=220V$, $R=10\Omega$, $L=20mH$, $f=50Hz$, et $t=40ms$.

Charge RLE:

ELTSIM

SIMPLORER

Tension redressée*Courant redressé**Tension inverse aux bornes de la diode*

Figure(IV.3): Courant, tension redressée et tension inverse aux bornes de D
 $V_m=220V$, $R=10\Omega$, $L=20mH$, $E=110v$, $f=50Hz$, et $t=40ms$.

IV.2. Intèrpritation des résultats:

Charge R:

La diode conduit un courant I_d dans l'intervalle $[0, \pi]$, lorsque la tension a ses bornes ($v_m \sin \omega t$) est positive, donc la tension inverse aux bornes de la diode est nulle.

La diode cesse de conduire dans l'intervalle $[\pi, 2\pi]$ lorsque la tension a ses bornes devient négative, c'est-à-dire que le courant et la tension redressée sont nuls.

Charge RL:

La diode conduit un courant I_d dans l'intervalle $[0, \theta_{ex}]$, où la tension a ses bornes ($v_m \sin \omega t$) est positive. Dans l'intervalle $[\pi, \theta_{ex}]$ la bobine décharge sa puissance électromagnétique emmagasinée dans la résistance jusqu'à où le courant soit nul, donc la tension inverse aux bornes de la diode devient négative.

Généralement l'angle de d'extinction θ_{ex} est compris entre π et $\frac{3}{2}\pi$, et le courant devient lissé par rapport a la charge R.

Charge RLE:

La diode conduit un courant I_d dans l'intervalle $[0, \theta_{ex}]$, où la tension a ses bornes ($v_m \sin \omega t - E$) est positive. Dans l'intervalle $[\pi, \theta_{ex}]$ la bobine décharge sa puissance électromagnétique emmagasinée dans la résistance jusqu'à où le courant soit nul et la tension redressée égale a E, donc la tension inverse aux bornes de la diode devient négative.

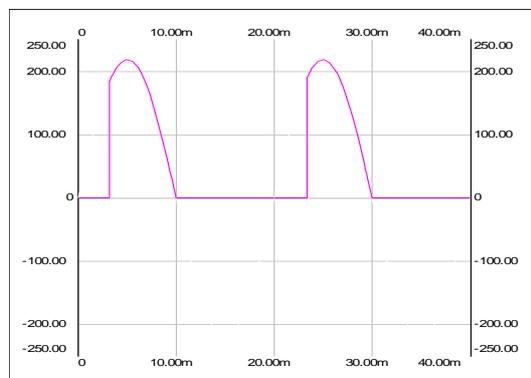
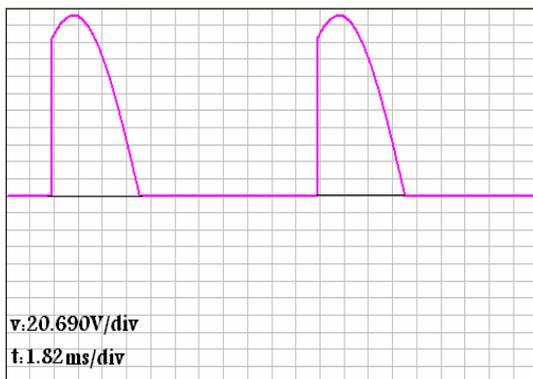
IV.3. Redressement monophasé simple à thyristor.

Charge R:

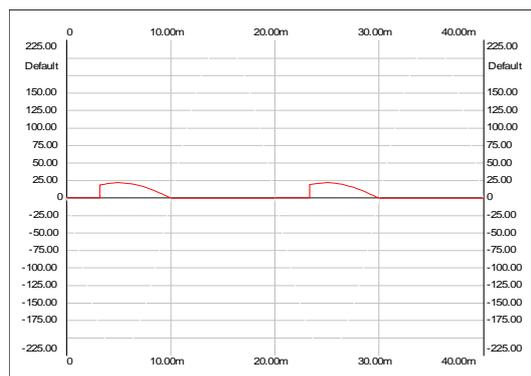
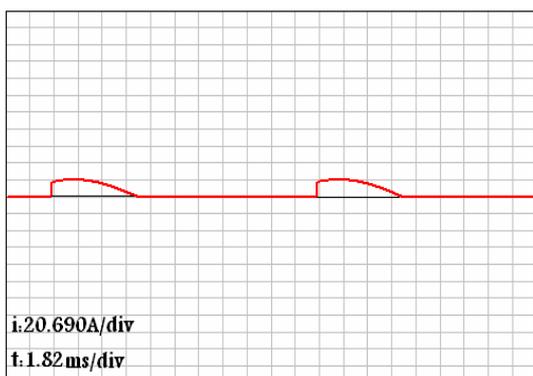
ELTSIM

SIMPLORER

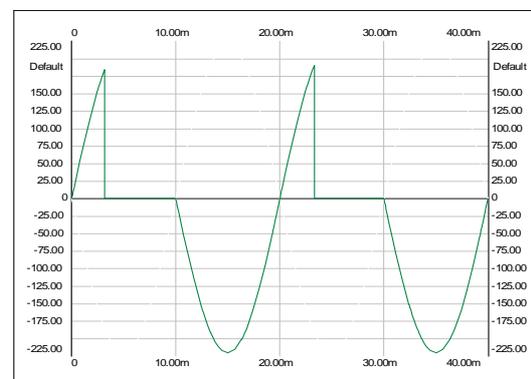
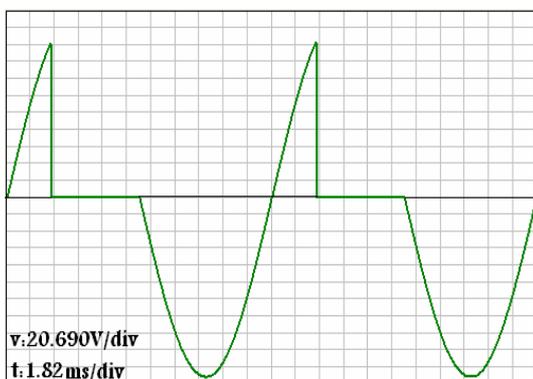
Tension redressée



Courant redressé



Tension inverse aux bornes du thyristor Th



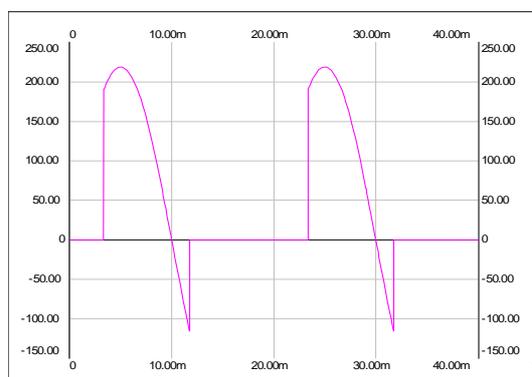
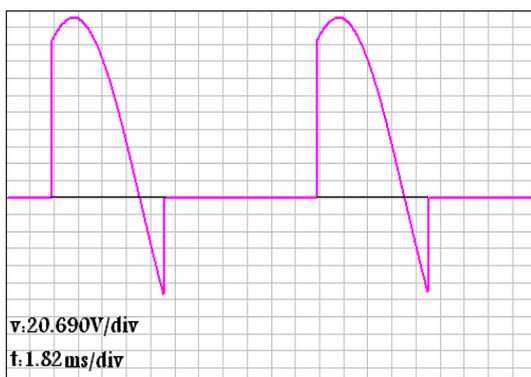
Figure(IV.4): Courant, tension redressée et tension inverse aux bornes de Th
 $V_m=220V, R=10\Omega, f=50Hz, t=40ms, \alpha=60^\circ$.

Charge RL:

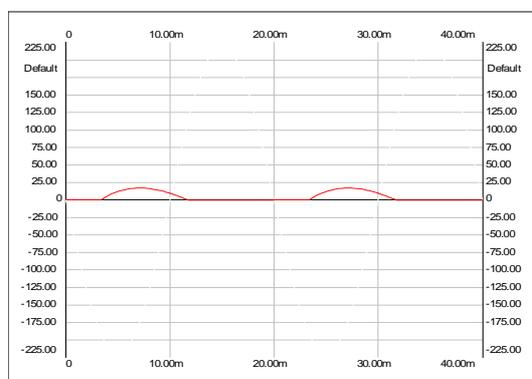
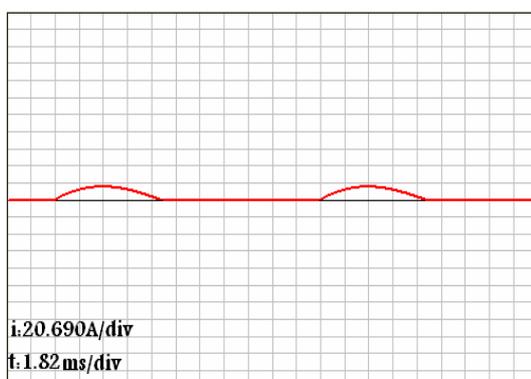
ELTSIM

SIMPLORER

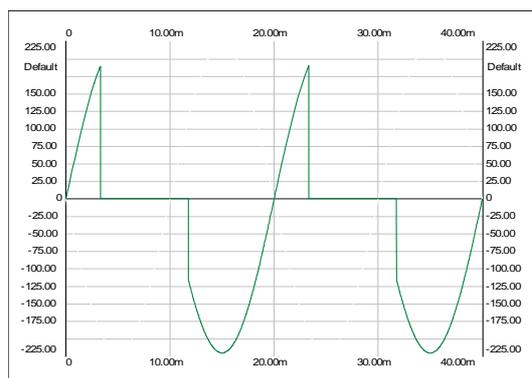
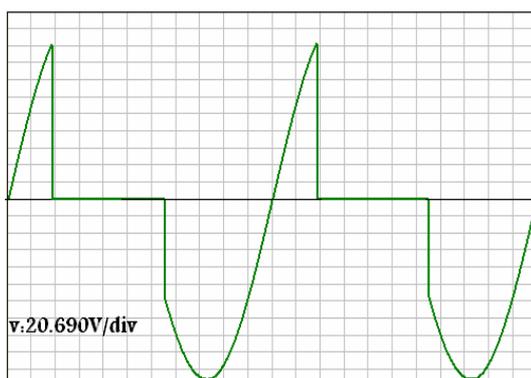
Tension redressée



Courant redressé



Tension inverse aux bornes du thyristor Th



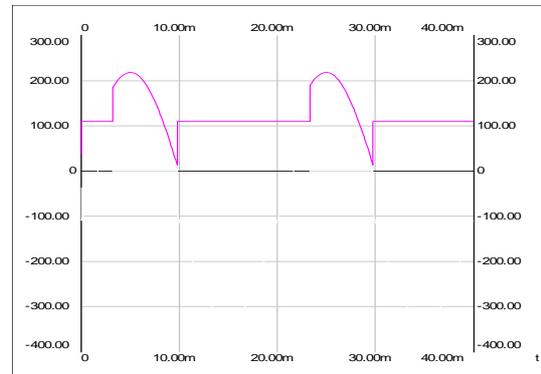
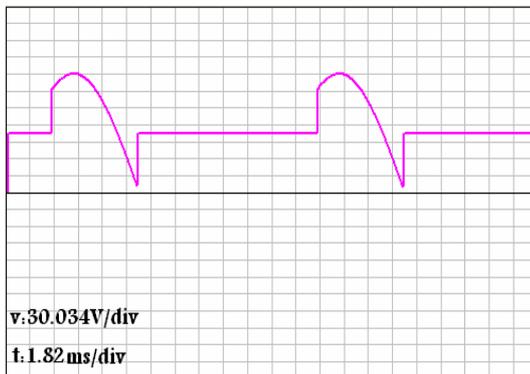
Figure(IV.5): Courant, tension redressée et tension inverse aux bornes de Th
 $V_m=220V, R=10\Omega, L=20mH, f=50Hz, t=40ms, \alpha=60^\circ$.

Charge RLE:

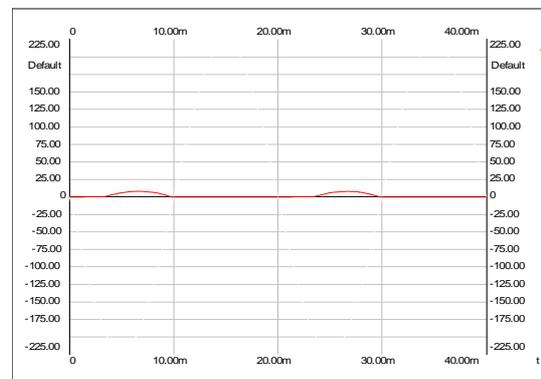
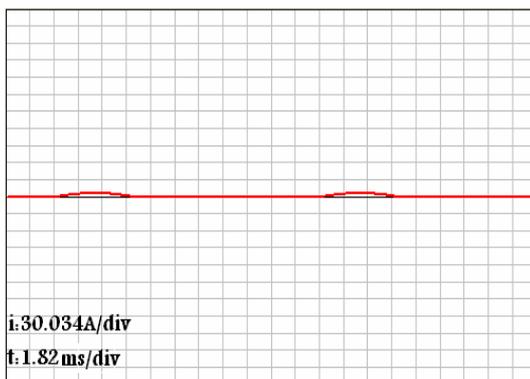
ELTSIM

SIMPLORER

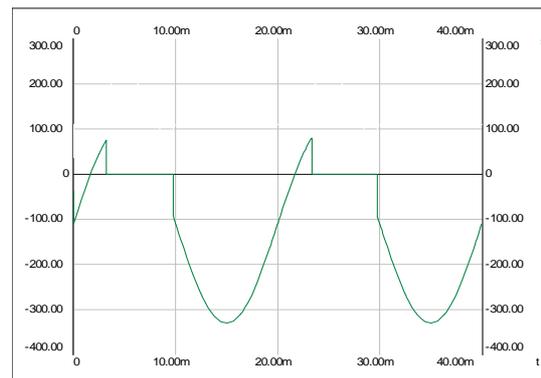
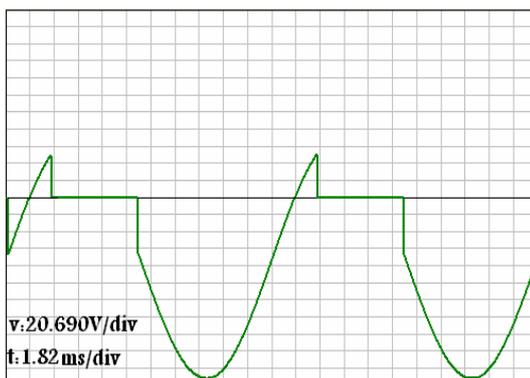
Tension redressée



Courant redressé



Tension inverse aux bornes du thyristor Th



Figure(IV.6): Courant, tension redressée et tension inverse aux bornes de Th
 $V_m=220V, R=10\Omega, L=20mH, E=110v, f=50Hz, t=40ms, \alpha=60^\circ$.

IV.4. Intèrpritation des résultats:**Charge R:**

Le thyristor conduit un courant I_d dans l'intervalle $[\alpha, \pi]$, lorsque il reçoit une impulsion où la tension a ses bornes ($v_m \sin \omega t$) est positive.

Le thyristor cesse de conduire dans l'intervalle $[\pi, 2\pi]$, lorsque la tension a ses bornes devient négative, c'est-à-dire que le courant et la tension redressée sont nuls.

Charge RL:

Le thyristor conduit un courant I_d dans l'intervalle $[\alpha, \theta_{ex}]$, lorsque il reçoit une impulsion où la tension a ses bornes ($v_m \sin \omega t$) est positive. Dans l'intervalle $[\pi, \theta_{ex}]$ la bobine décharge sa puissance dans la résistance jusqu'à où le courant soit nul.

Charge RLE:

Le thyristor conduit un courant I_d dans l'intervalle $[\alpha, \theta_{ex}]$, lorsque il reçoit une impulsion où la tension a ses bornes ($v_m \sin \omega t - E$) est positive.

Dans l'intervalle $[\pi, \theta_{ex}]$ la bobine décharge sa puissance jusqu'à où le courant soit nul, et la tension redressée devient égale à E , donc la tension inverse aux bornes de la diode devient négative.

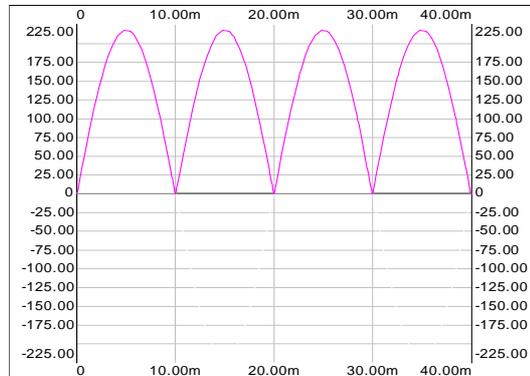
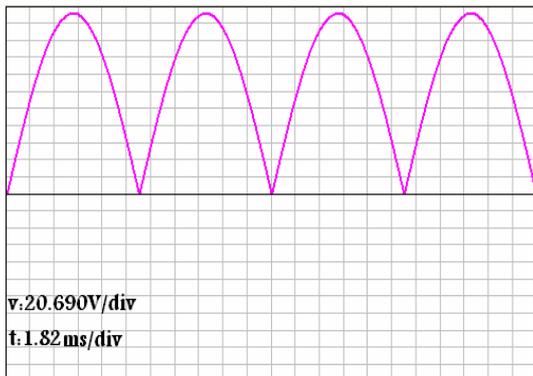
IV.5. Redressement monophasé en pont à diodes.

Charge R:

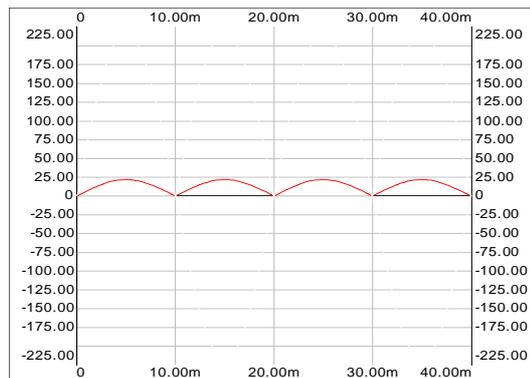
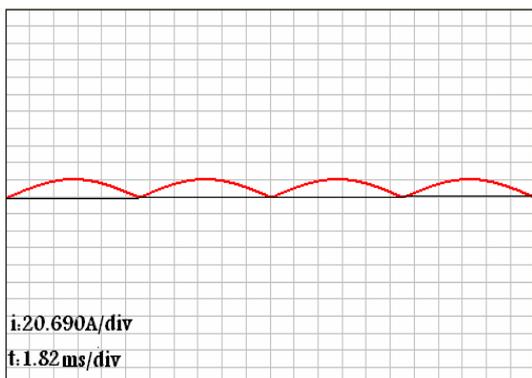
ELTSIM

SIMPLORER

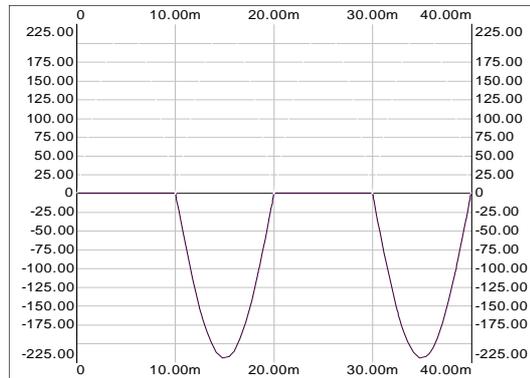
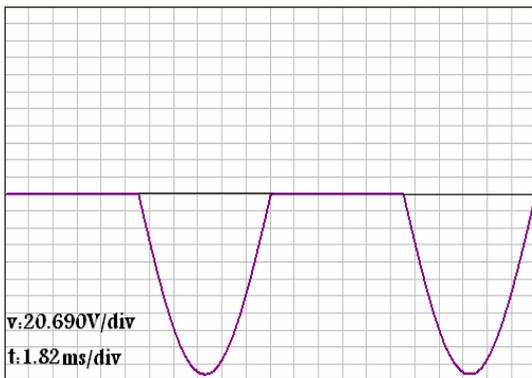
Tension redressée



Courant redressé



Tension inverse aux bornes de la diode D1



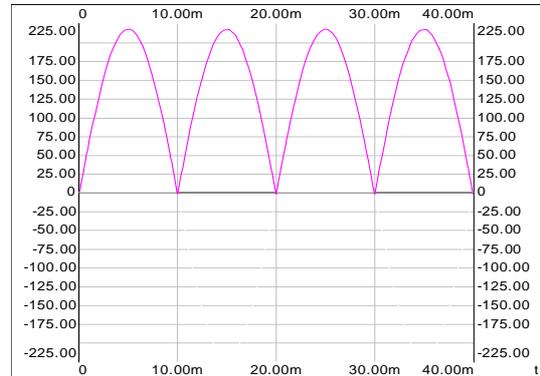
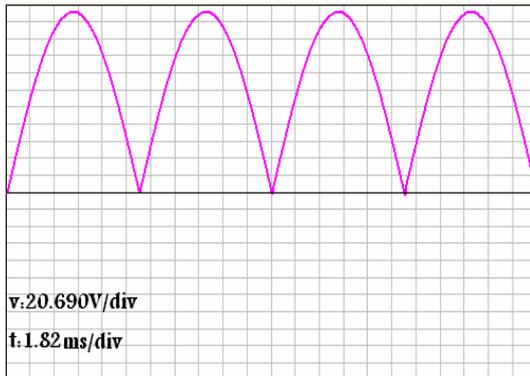
Figure(IV.7): Courant, tension redressée et tension inverse aux bornes de D1
 $V_m=220V, R=10\Omega, f=50Hz, t=40ms.$

Charge RL:

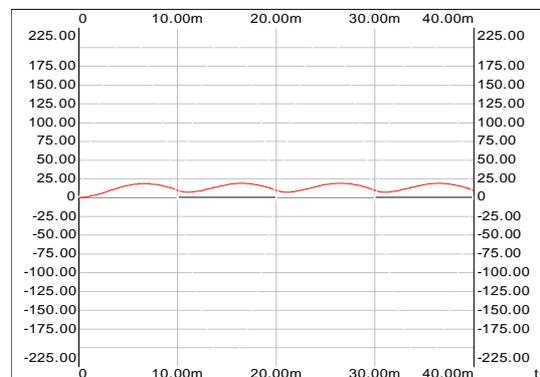
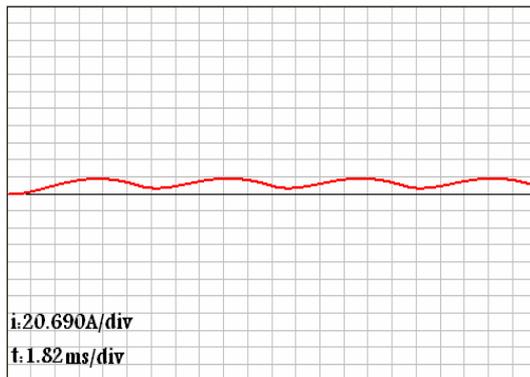
ELTSIM

SIMPLORER

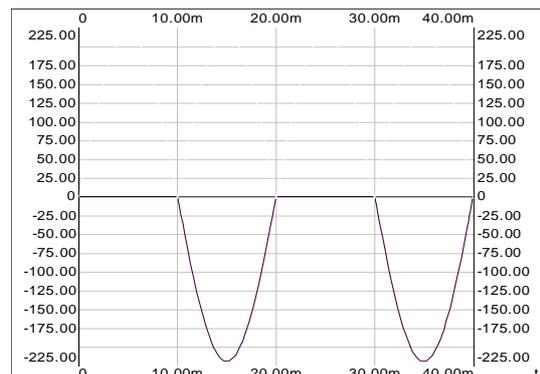
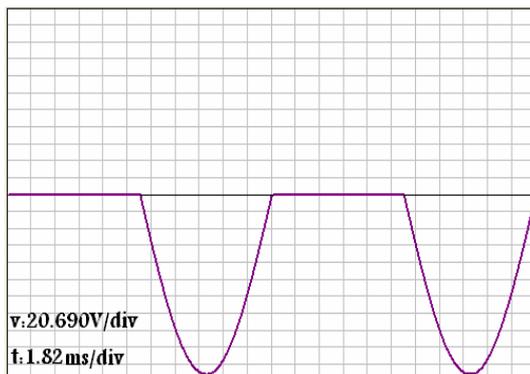
Tension redressée



Courant redressé



Tension inverse aux bornes de la diode D1



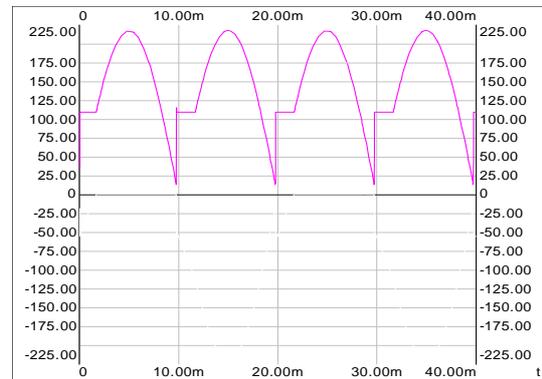
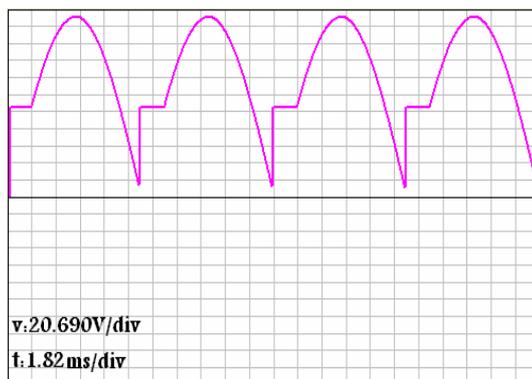
Figure(IV.8): Courant, tension redressée et tension inverse aux bornes de D1
 $V_m=220V, R=10\Omega, L=20mH, f=50Hz, t=40ms.$

Charge RLE:

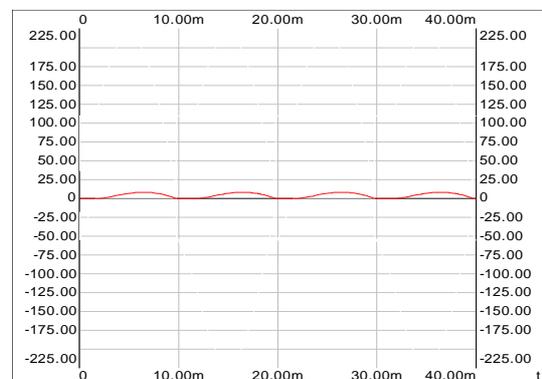
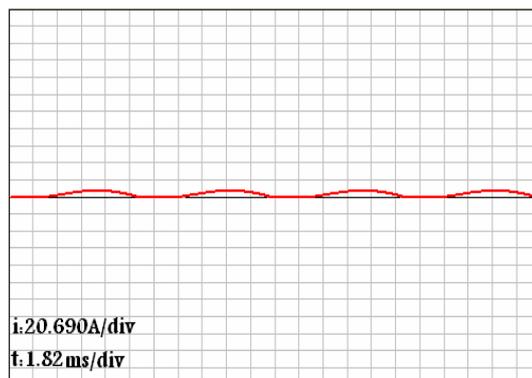
ELTSIM

SIMPLORER

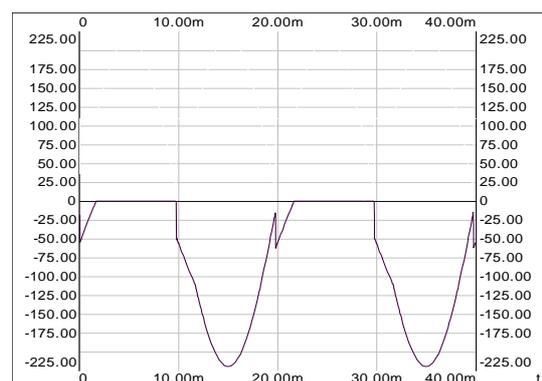
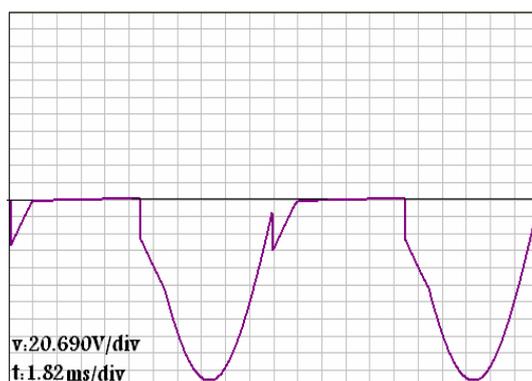
Tension redressée



Courant redressé



Tension inverse aux bornes de la diode D1



Figure(IV.9): Courant, tension redressée et tension inverse aux bornes de D1
 $V_m=220V$, $R=10\Omega$, $L=20mH$, $E=110v$, $f=50Hz$, $t=40ms$.

IV.6. Interprétation des résultats:

Charge R:

La conduction du courant I_d dans l'intervalle $[0, \pi]$ se fait simultanément à l'aide des deux diodes D_1 et D_2 , lorsque la tension aux bornes de chaque diode $(v_m \sin \omega t)/2$ est positive. Les deux diodes cessent de conduire dans l'intervalle $[\pi, 2\pi]$, qui entre les deux autres diodes D_1' et D_2' en conduction (conduction naturelle), où la tension aux bornes de chacun d'eux $(v_m \sin \omega t)/2$ est positive, c'est-à-dire que le courant est nul dans les diodes D_1 et D_2 .

Le courant ne s'annule que dans les moments de changements des fonctionnements des diodes (points de commutation).

Charge RL:

La conduction du courant I_d dans l'intervalle $[0, \pi]$ se fait simultanément à l'aide des deux diodes D_1 et D_2 , lorsque la tension aux bornes de chaque diode $(v_m \sin \omega t)/2$ est positive. Les deux diodes cessent de conduire dans l'intervalle $[\pi, 2\pi]$, qui entre les deux autres diodes D_1' et D_2' en conduction, où la tension aux bornes de chacun d'eux $(v_m \sin \omega t)/2$ est positive, se qui assure la continuité du courant dans la charge car la bobine se décharge pendant le changement des deux diodes, c'est-à-dire que le courant est nul dans les diodes D_1 et D_2 .

Charge RLE:

La conduction du courant I_d dans l'intervalle $[0, \pi]$ se fait simultanément à l'aide des deux diodes D_1 et D_2 .

Les deux diodes cessent de conduire dans l'intervalle $[\pi, 2\pi]$, qui entre les deux autres diodes D_1' et D_2' en conduction, dans l'intervalle où le courant devient nul à cause de la présence de E la tension aux bornes de la charge devient égal à E.

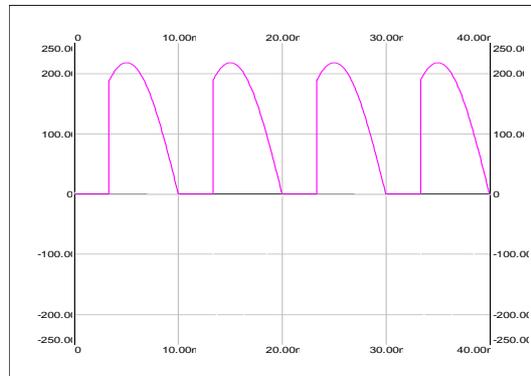
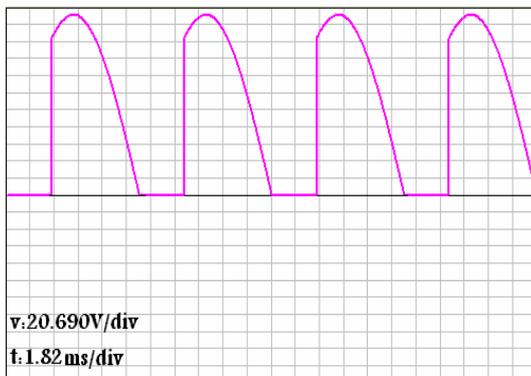
IV.7. Redressement monophasé en pont à thyristors.

Charge R:

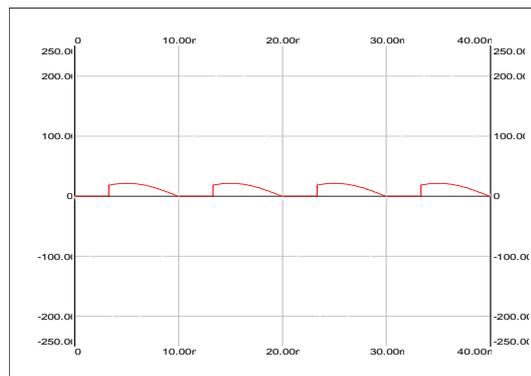
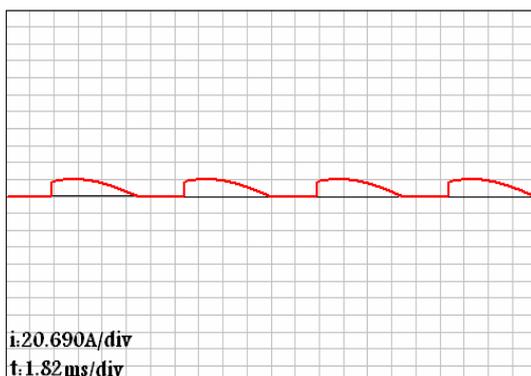
ELTSIM

SIMPLORER

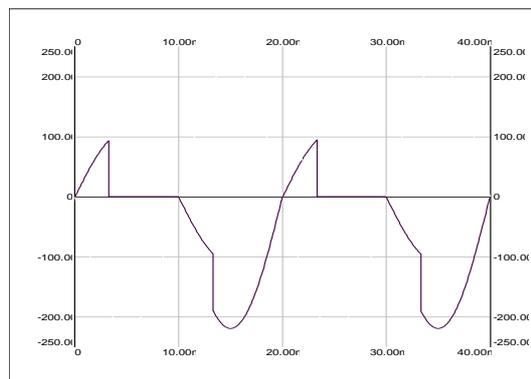
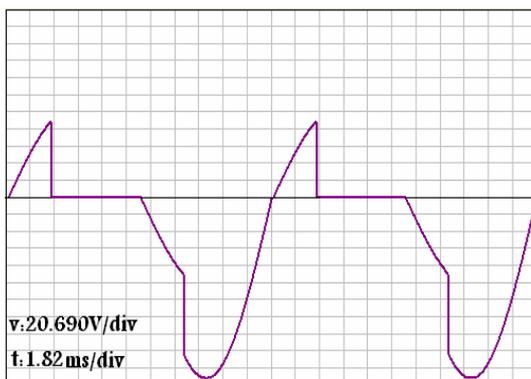
Tension redressée



Courant redressé



Tension inverse aux bornes du thyristor Th1

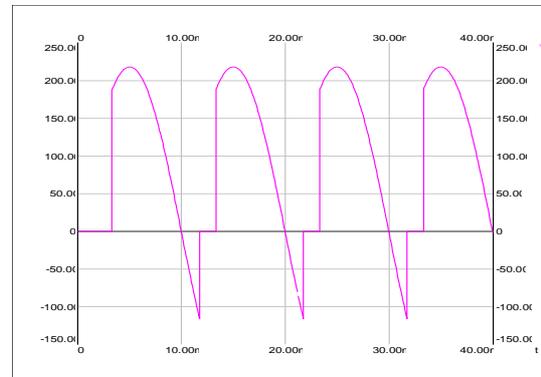
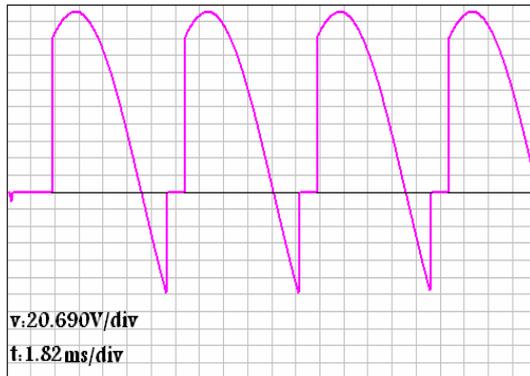
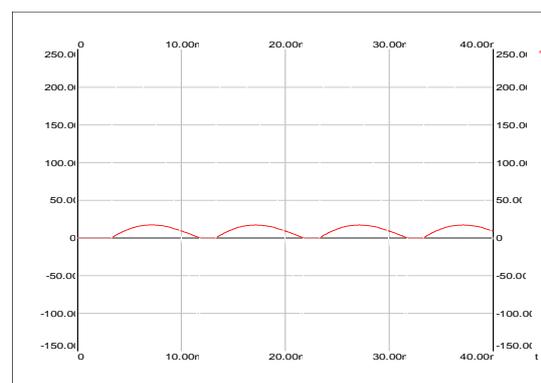
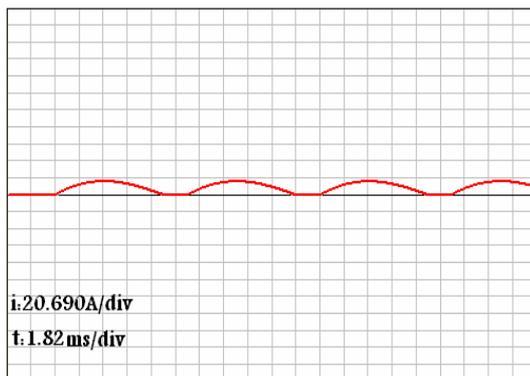
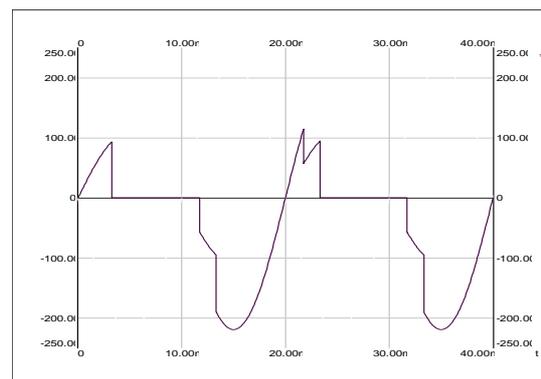


Figure(IV.10): Courant, tension redressée et tension inverse aux bornes de Th1
 $V_m=220V, R=10\Omega, f=50Hz, t=40ms, \alpha=60^\circ$.

Charge RL:

ELTSIM

SIMPLORER

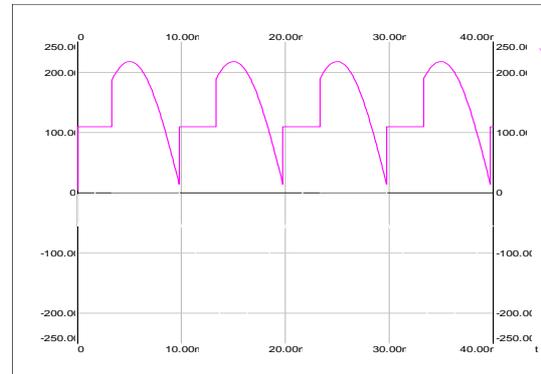
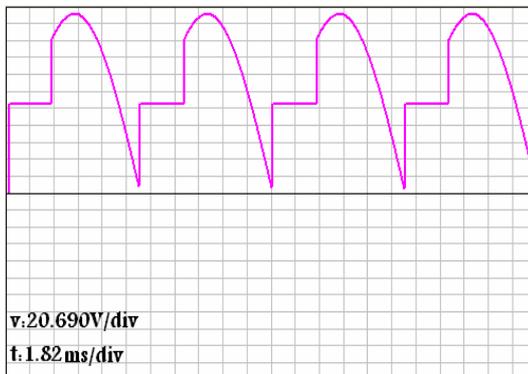
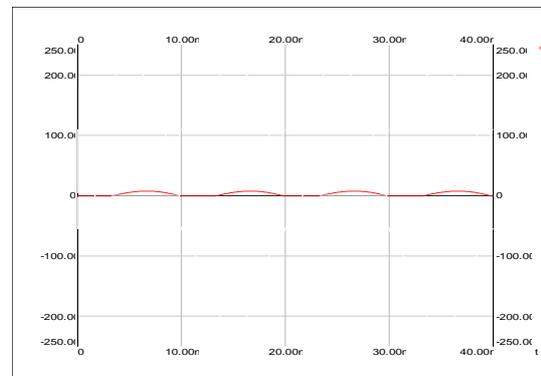
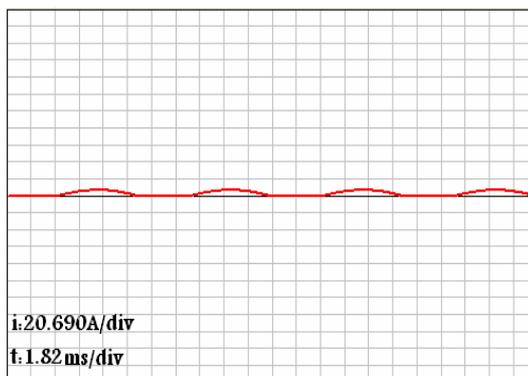
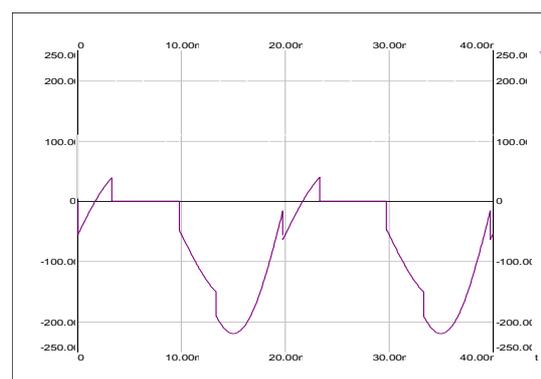
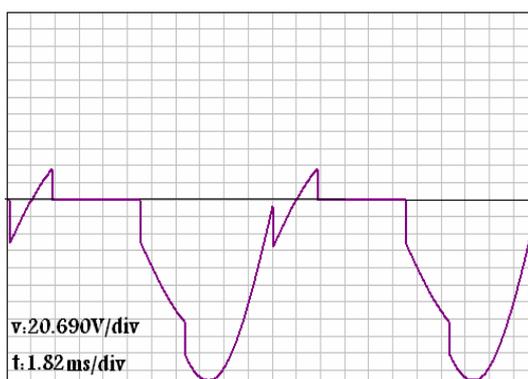
Tension redressée*Courant redressé**Tension inverse aux bornes du thyristor Th1*

Figure(IV.11): Courant, tension redressée et tension inverse aux bornes de Th1
 $V_m=220V$, $R=10\Omega$, $L=20mH$, $f=50Hz$, $t=40ms$, $\alpha=60^\circ$.

Charge RLE:

ELTSIM

SIMPLORER

Tension redressée*Courant redressé**Tension inverse aux bornes du thyristor Th1*

Figure(IV.12): Courant, tension redressée et tension inverse aux bornes de Th1
 $V_m=220V$, $R=10\Omega$, $L=20mH$, $E=110v$, $f=50Hz$, $t=40ms$, $\alpha=60^\circ$.

IV.8. Interprétation des résultats:

Charge R:

La conduction du courant I_d dans l'intervalle $[\alpha, \pi]$ se fait simultanément à l'aide des deux thyristors Th_1 et Th_2 , lorsque ils reçoivent des impulsions où la tension aux bornes de chaque thyristor $(v_m \sin \omega t)/2$ est positive.

Les deux thyristors cessent de conduire dans l'intervalle $[\pi, 2\pi + \alpha]$, qui entre les deux autres thyristors Th_1' et Th_2' en conduction, lorsque ils reçoivent à leurs tour des impulsions où la tension aux bornes de chacun d'eux $(v_m \sin \omega t)/2$ est positive, c'est-à-dire que le courant est nul dans les diodes D_1 et D_2 .

Charge RL:

La conduction du courant I_d dans l'intervalle $[\alpha, \theta_{ex}]$ se fait simultanément à l'aide des deux thyristors Th_1 et Th_2 , lorsque ils reçoivent des impulsions où la tension aux bornes de chaque thyristor $(v_m \sin \omega t)/2$ est positive. Les deux thyristors cessent de conduire dans l'intervalle $[\theta_{ex}, 2\pi + \alpha]$, qui entre les deux autres thyristors Th_1' et Th_2' en conduction, lorsque ils reçoivent des impulsions où la tension aux bornes de chacun d'eux $(v_m \sin \omega t)/2$ est positive.

Charge RLE:

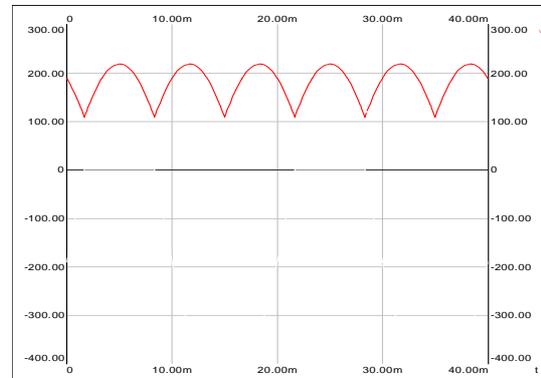
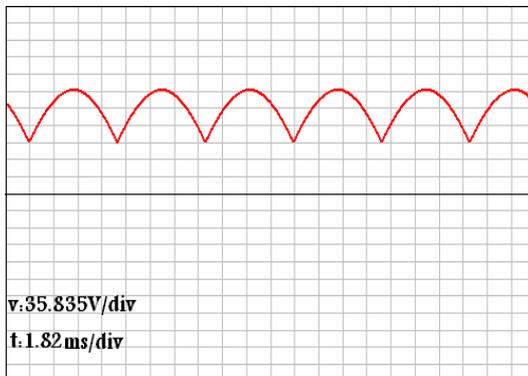
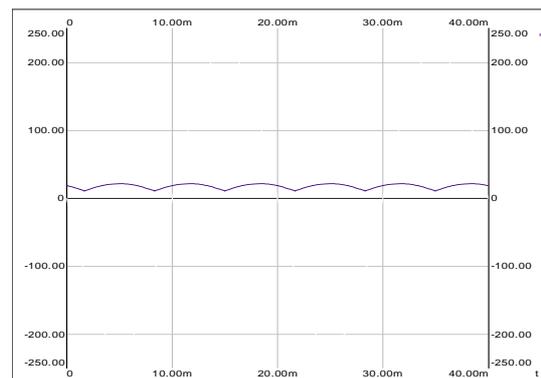
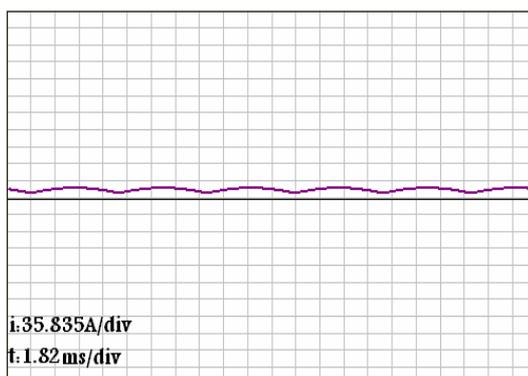
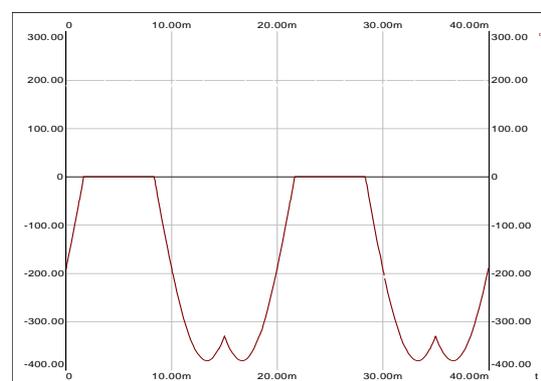
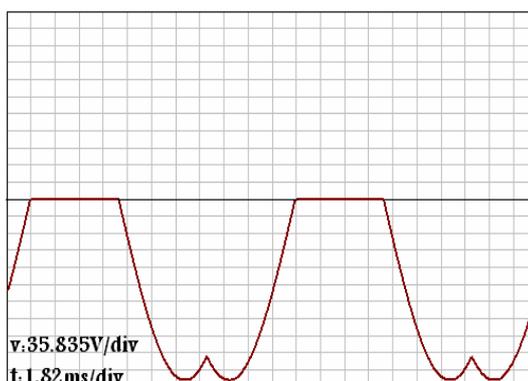
La conduction du courant I_d dans l'intervalle $[\alpha, \theta_{ex}]$ se fait simultanément à l'aide des deux thyristors Th_1 et Th_2 . Les deux diodes cessent de conduire dans l'intervalle $[\theta_{ex}, 2\pi + \theta_{ex}]$, qui entre les deux autres thyristors D_1' et D_2' en conduction, dans l'intervalle où le courant devient nul à cause de la présence de E la tension aux bornes de la charge devient égal à E.

IV.9. Redressement triphasé parallèle simple à diodes.

Charge R:

ELTSIM

SIMPLORER

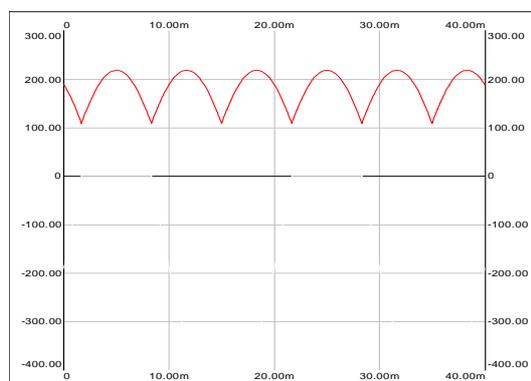
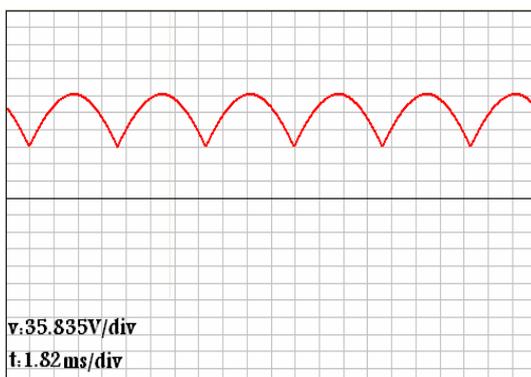
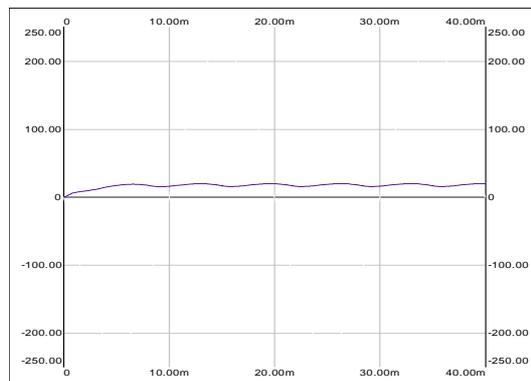
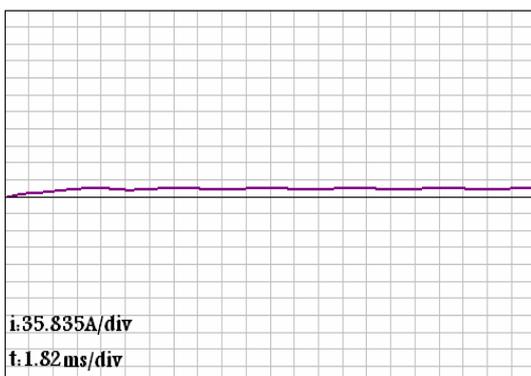
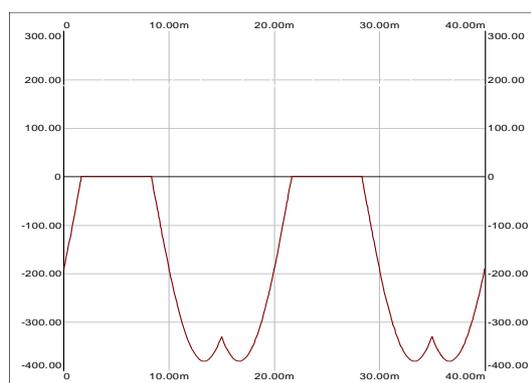
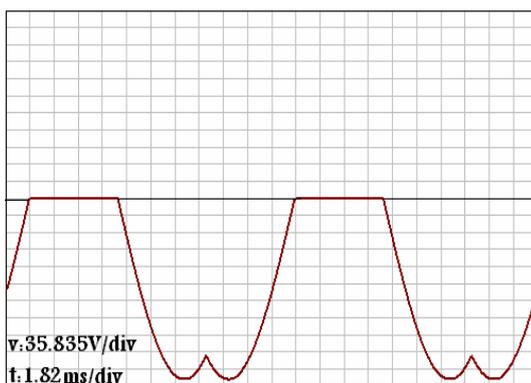
Tension redressée*Courant redressé**Tension inverse aux bornes de la diode D1*

Figure(IV.13): Courant, tension redressée et tension inverse aux bornes de Th1
 $V_m=220V, R=10\Omega, f=50Hz, t=40ms.$

Charge RL:

ELTSIM

SIMPLORER

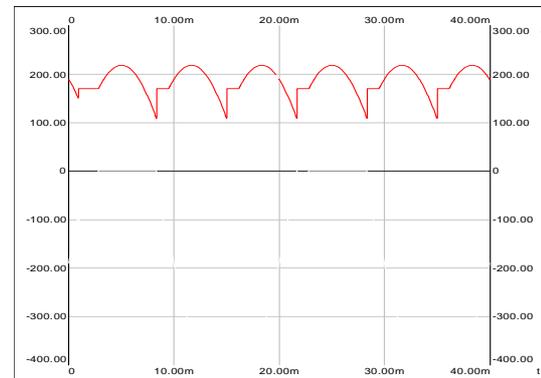
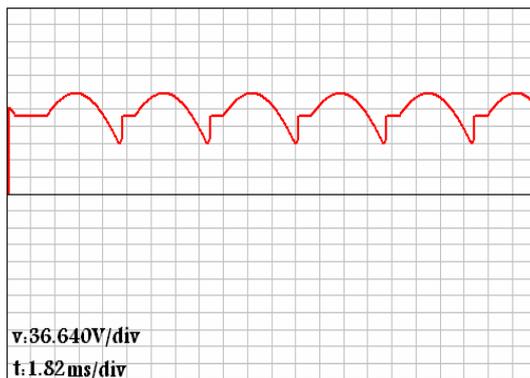
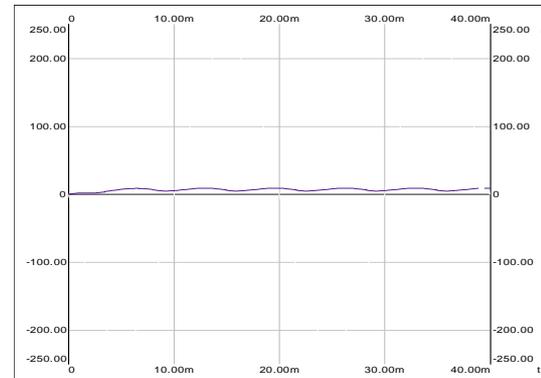
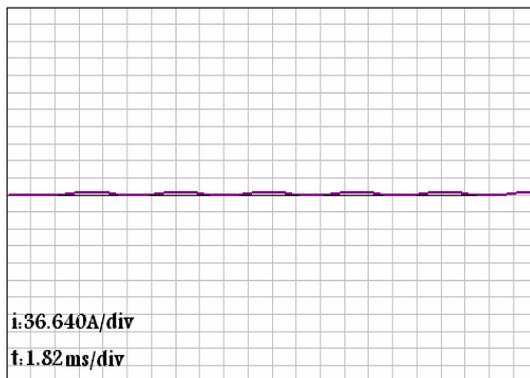
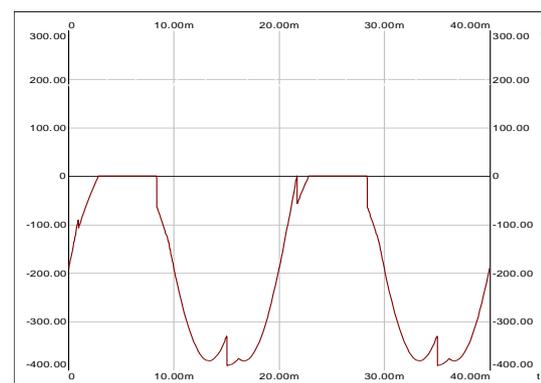
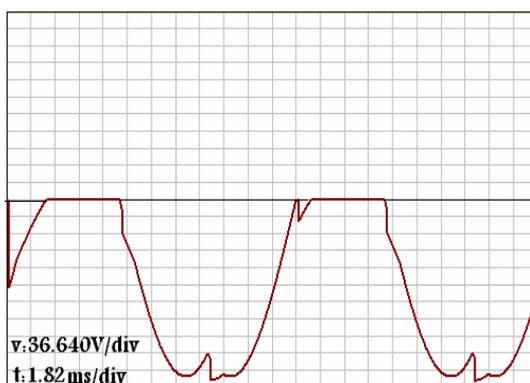
Tension redressée*Courant redressé**Tension inverse aux bornes de la diode D1*

Figure(IV.14): Courant, tension redressée et tension inverse aux bornes de Th1
 $V_m=220V, R=10\Omega, L=20mH, f=50Hz, t=40ms.$

Charge RLE:

ELTSIM

SIMPLORER

Tension redressée*Courant redressé**Tension inverse aux bornes de la diode D1*

Figure(IV.15): Courant, tension redressée et tension inverse aux bornes de Th1
 $V_m=220V$, $R=10\Omega$, $L=20mH$, $E=170v$, $f=50Hz$, $t=40ms$.

IV.10. Interprétation des résultats:**Charge R:**

Les trois diodes travaillent en alternance assurant un courant de charge I_d durant toute la période T, la diode passante est celle qui présente une tension plus élevée a ses bornes, les deux autres diodes sont bloquées (commutation naturelle), le courant ne s'annule pas et à des ondulations, car au point de commutation des diodes il existe toujours un courant lie a l'ancienne conduction.

La tension inverse aux bornes de la diode D_1 est nulle lorsqu'elle conduit, est égale à v_1-v_j lorsqu'elle ne conduit pas.

Charge RL:

Les trois diodes travaillent en alternance assurant un courant de charge I_d durant toute la période T, cette fois les trois diodes circule un courant I_d lissé (diminution des ondulations) a cause de la décharge de la bobine dans la charge.,

Charge RLE:

Le courant de charge I_d est déminé en amplitude a cause de la présence de la force électromotrice E, et la bobine ce décharge dans la résistance lorsque la tension de source égale a E, lorsque le décharge se termine la tension redressée devient égale à E.

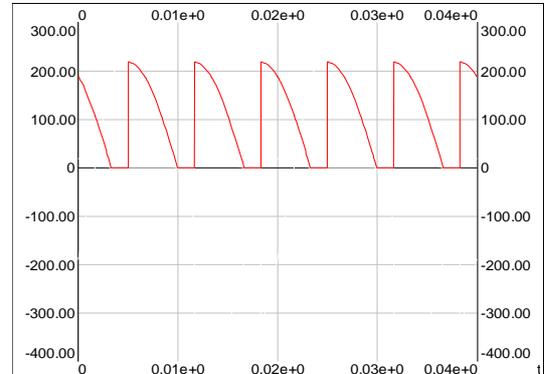
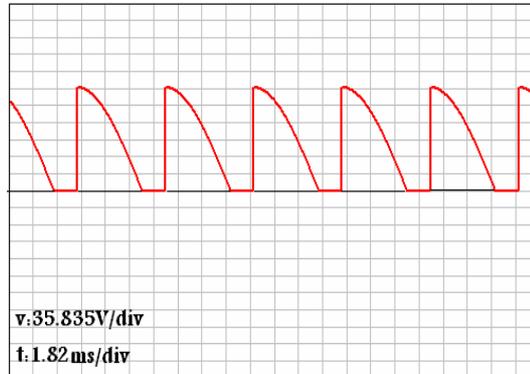
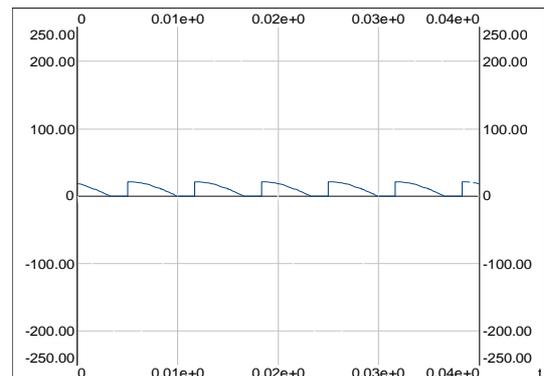
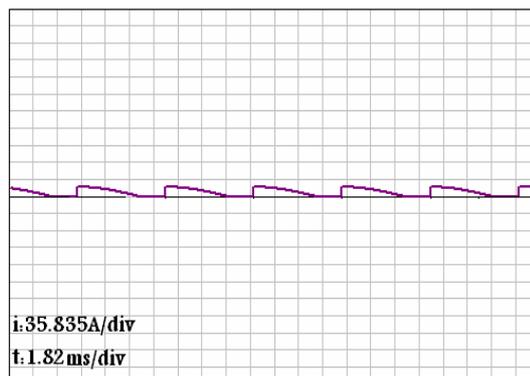
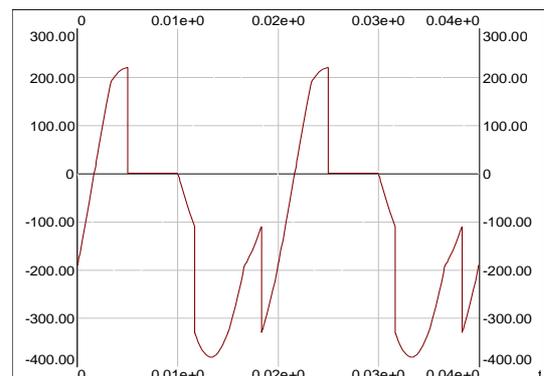
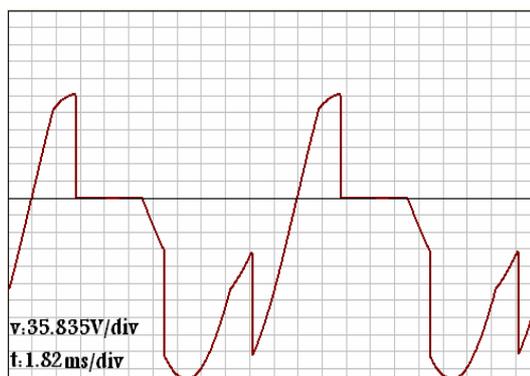
La tension inverse aux bornes de la diode D_1 est nulle lorsqu'elle conduit, est égale à v_1-v_j lorsqu'elle ne conduit pas.

IV.11. Redressement triphasé parallèle simple à thyristors.

Charge R:

ELTSIM

SIMPLORER

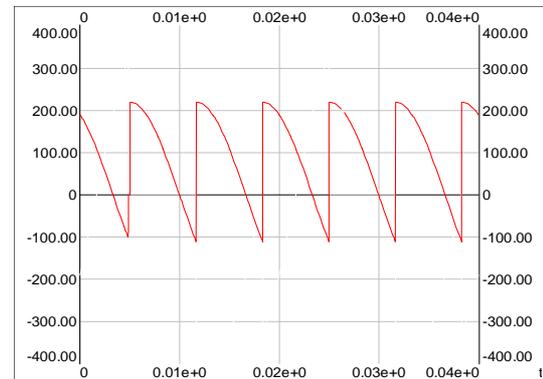
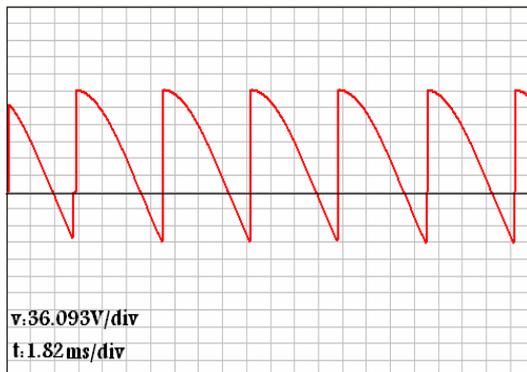
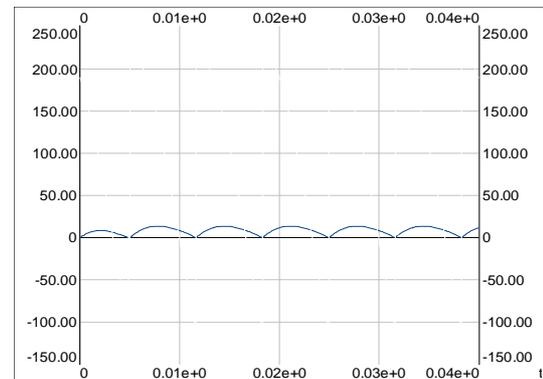
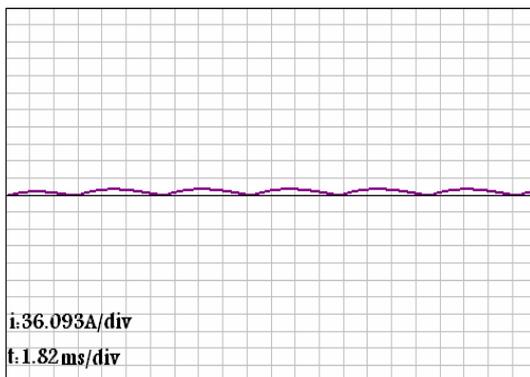
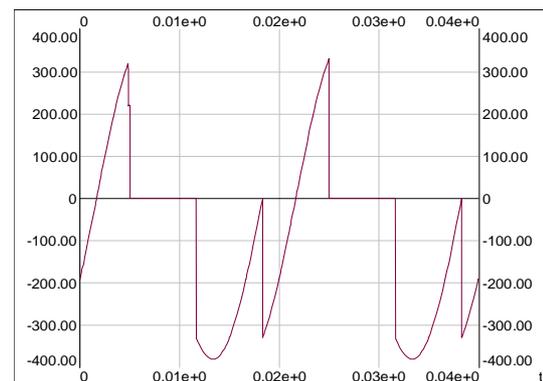
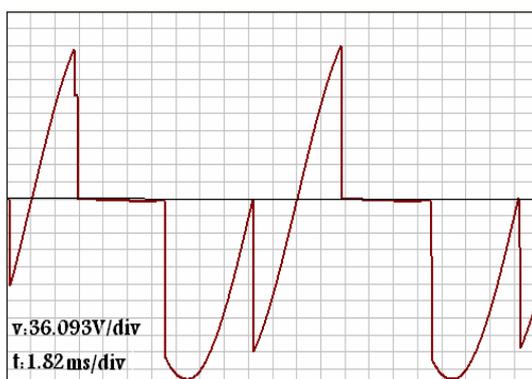
Tension redressée*Courant redressé**Tension inverse aux bornes du thyristor Th1*

Figure(IV.16): Courant, tension redressée et tension inverse aux bornes de Th1
 $V_m=220V, R=10\Omega, f=50Hz, t=40ms, \alpha=60^\circ$.

Charge RL:

ELTSIM

SIMPLORER

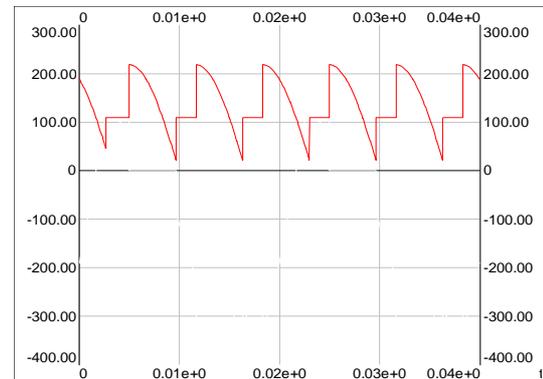
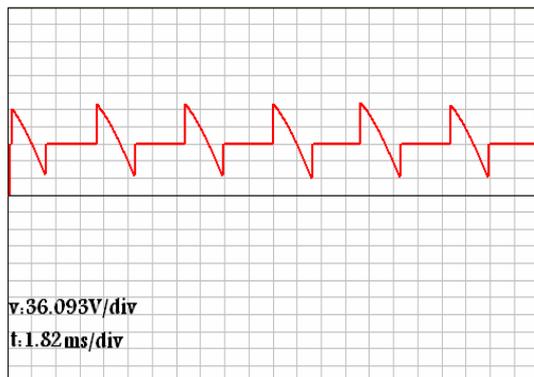
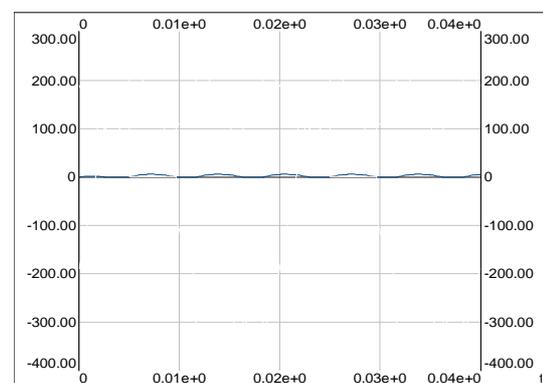
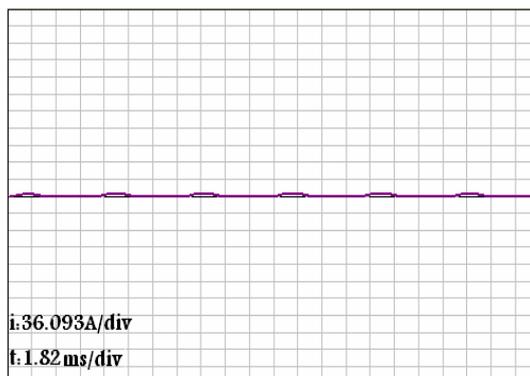
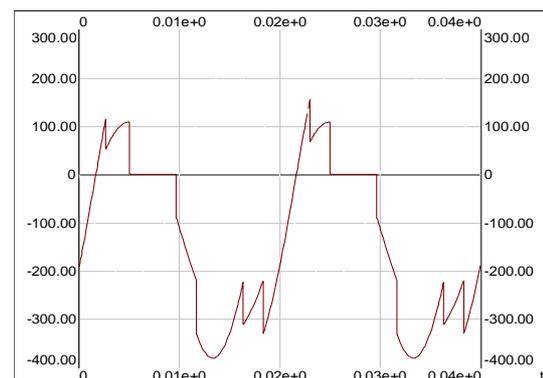
Tension redressée*Courant redressé**Tension inverse aux bornes du thyristor Th1*

Figure(IV.17): Courant, tension redressée et tension inverse aux bornes de Th1
 $V_m=220V, R=10\Omega, L=20mH, f=50Hz, t=40ms, \alpha=60^\circ$.

Charge RLE:

ELTSIM

SIMPLORER

Tension redressée*Courant redressé**Tension inverse aux bornes du thyristor Th1*

Figure(IV.18): Courant, tension redressée et tension inverse aux bornes de Th1
 $V_m=220V$, $R=10\Omega$, $L=20mH$, $E=170v$, $f=50Hz$, $t=40ms$, $\alpha=60^\circ$.

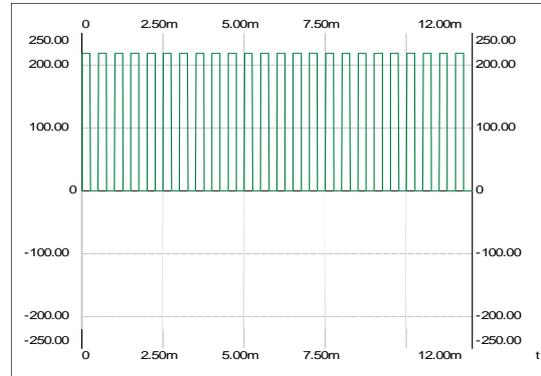
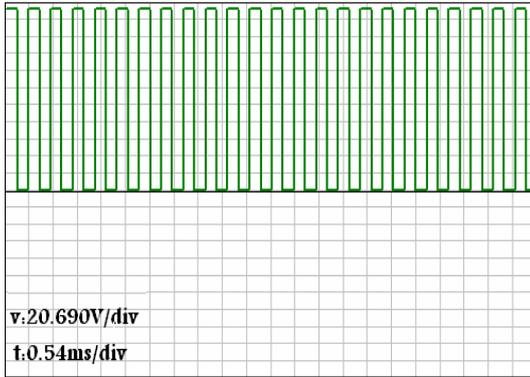
IV.12. Hacheur série:

Charge R:

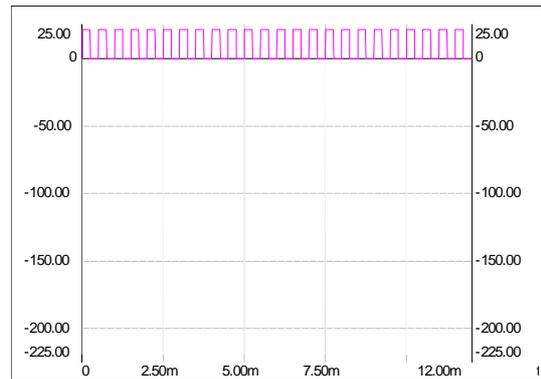
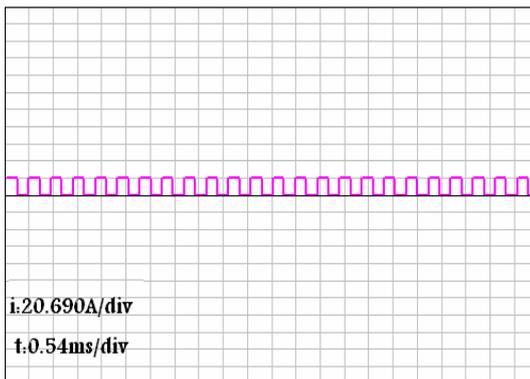
ELTSIM

SIMPLORER

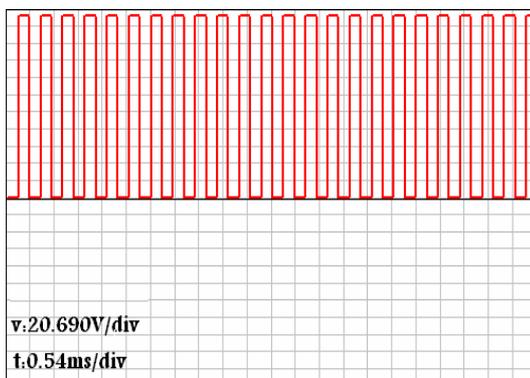
Tension



Courant



Tension inverse aux bornes du hacheur H

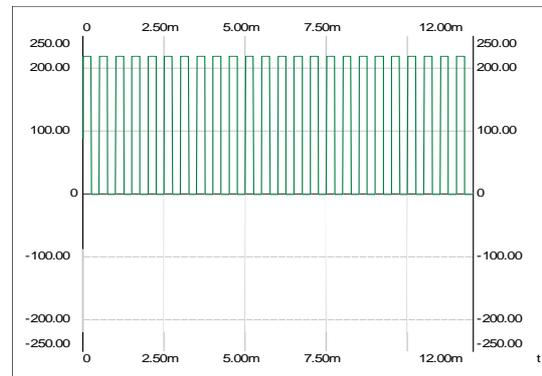
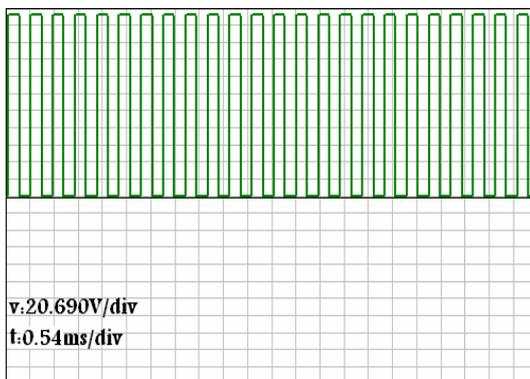
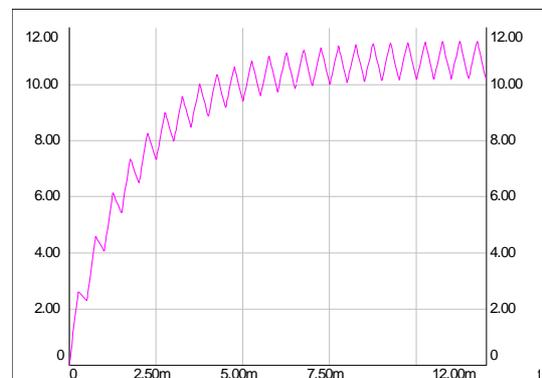
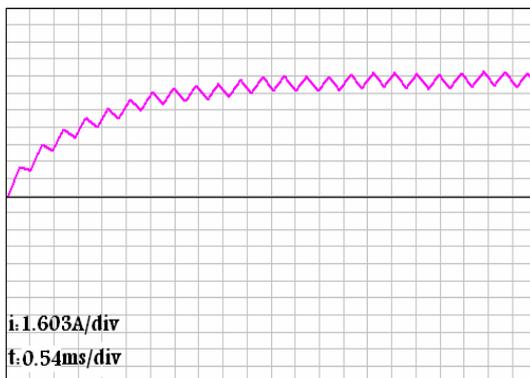
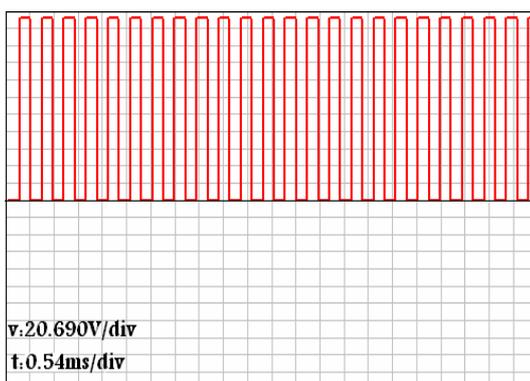


**Figure(IV.19): Courant, et tension délivrée par un hacheur série.
 $E_S=220V$, $R=10\Omega$, $f=2000Hz$, et $t=8ms$.**

Charge RL:

ELTSIM

SIMPLORER

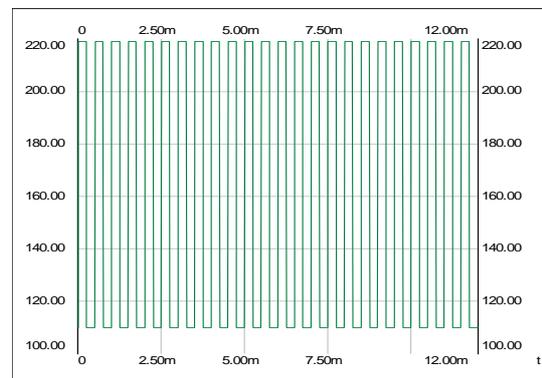
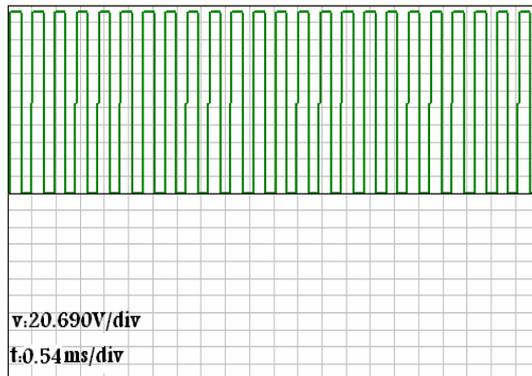
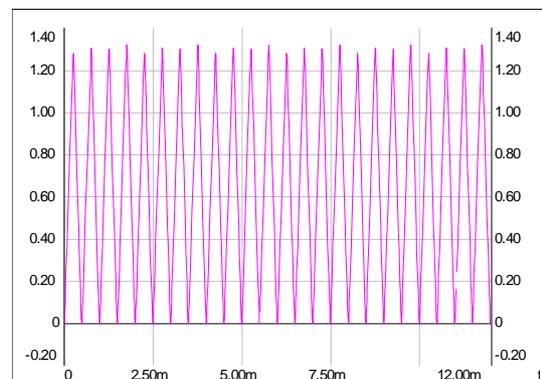
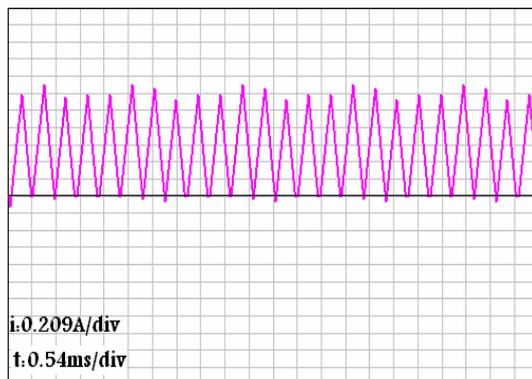
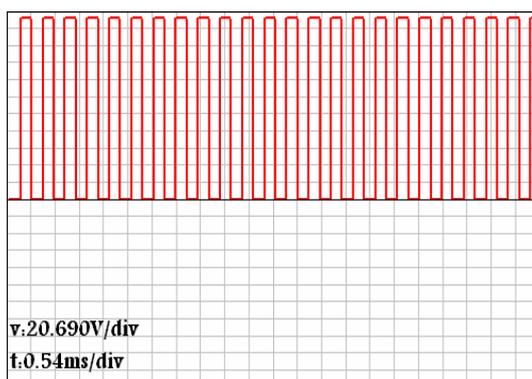
Tension*Courant**Tension inverse aux bornes du hacheur H*

**Figure(IV.20): Courant, et tension délivrée par un hacheur série.
 $E_S=220V$, $R=10\Omega$, $L=20mH$, $E=110v$, $f=2000Hz$, et $t=12ms$.**

Charge RLE:

ELTSIM

SIMPLORER

Tension*Courant**Tension inverse aux bornes du hacheur H*

**Figure(IV.21): Courant, et tension délivrée par un hacheur série.
 $E_S=220V$, $R=10\Omega$, $L=20mH$, $E=110v$, $f=2000Hz$, et $t=12ms$.**

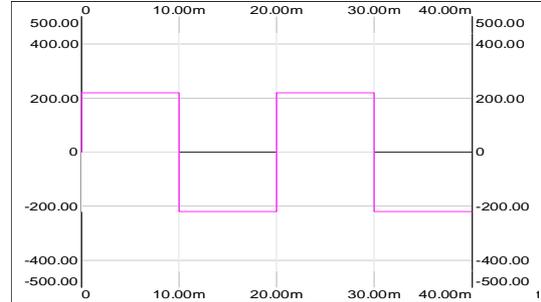
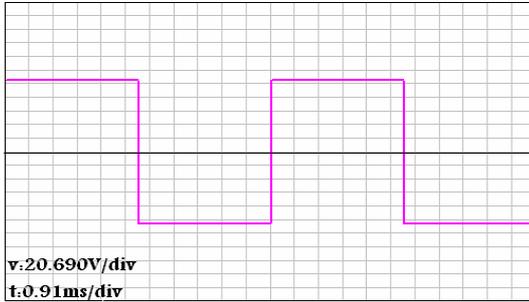
IV.13. Onduleur en demi- pont:

Charge R:

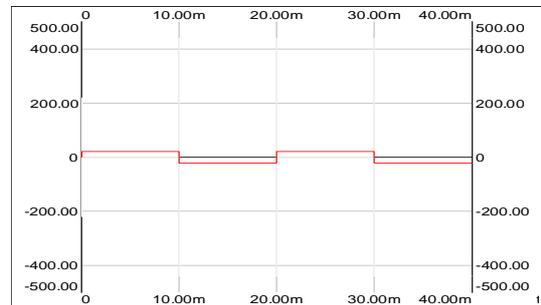
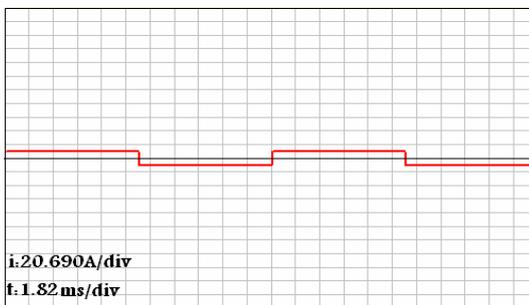
ELTSIM

SIMPLORER

Tension ondulée



Courant ondulé



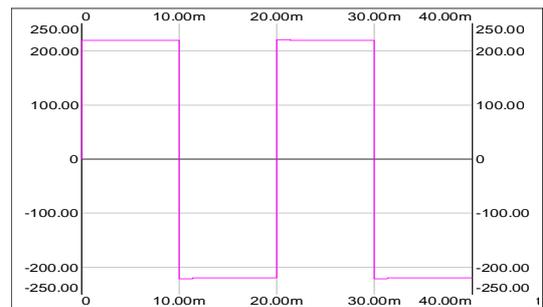
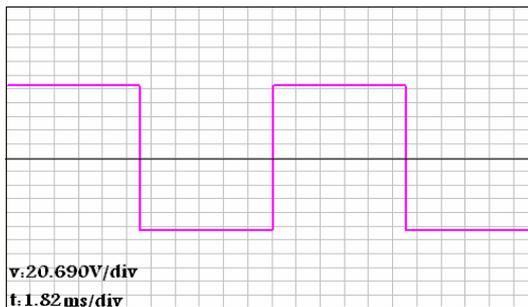
Figure(IV.22): Courant, et tension ondulés, $E_S=220V$, $R=10\Omega$, et $f=50Hz$, et $t=40ms$.

Charge RL:

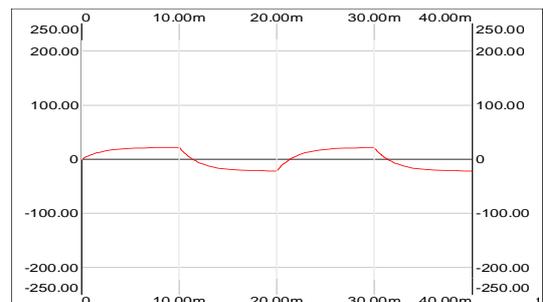
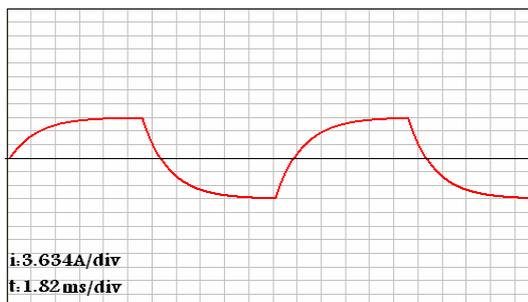
ELTSIM

SIMPLORER

Tension ondulée



Courant ondulé



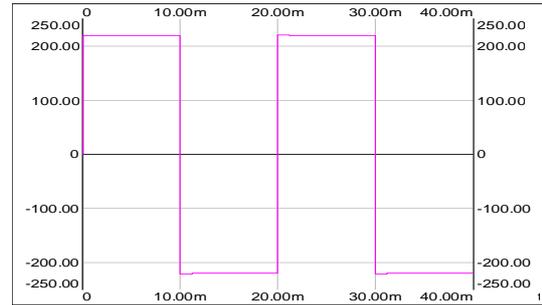
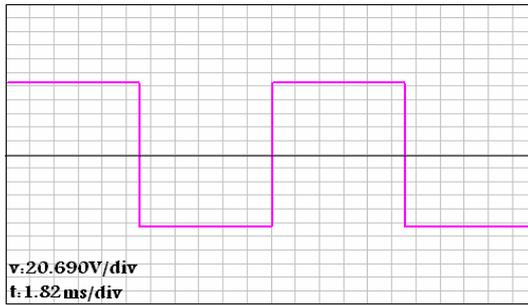
Figure(IV.23): Courant, et tension ondulés. $E_S=110V$, $R=10\Omega$, $L=20mH$, $f=50Hz$, et $t=40ms$.

Charge RLE(t):

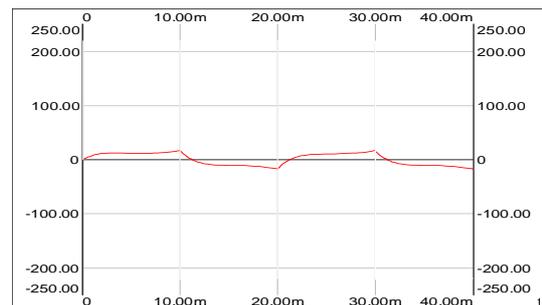
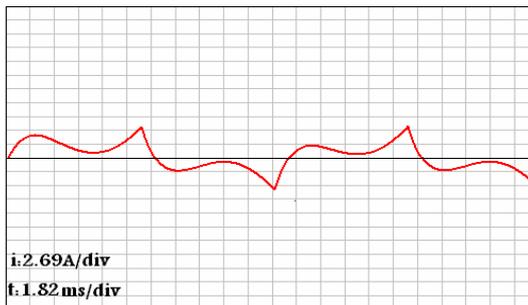
ELTSIM

SIMPLORER

Tension ondulée



Courant ondulé



Figure(IV.24):Courant, et tension ondulés.
 $E_S=220V$, $R=10\Omega$, $L=20mH$, $E=110\sin(100\pi t)$, $t=40ms$. et $f=50Hz$.

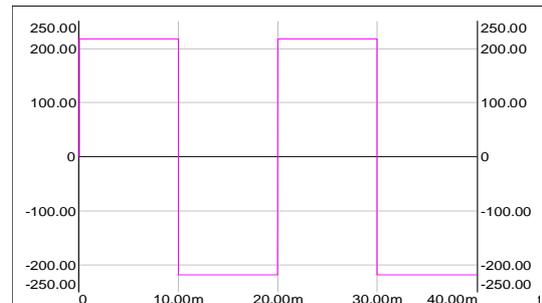
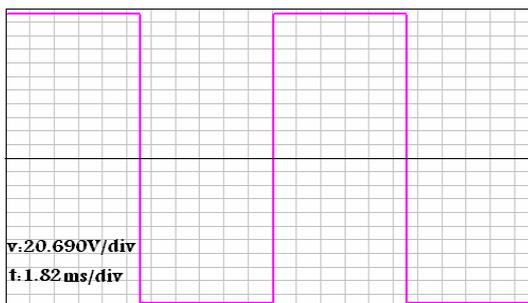
IV.14. Onduleur monophasé en pont:

Charge R:

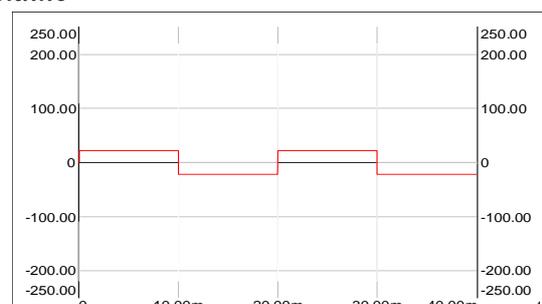
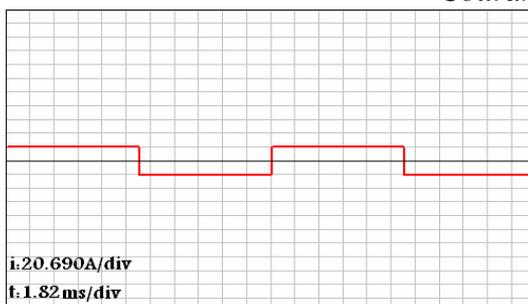
ELTSIM

SIMPLORER

Tension ondulée



Courant ondulé



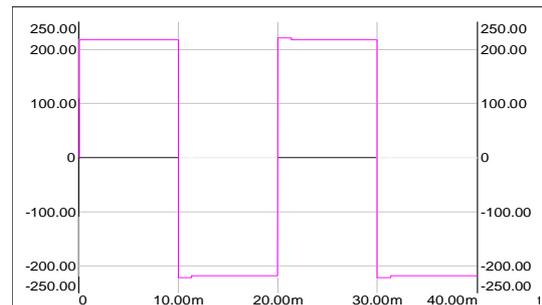
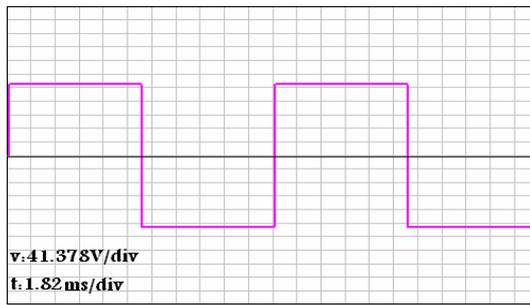
Figure(IV.25):Courant, et tension ondulés.
 $E_S=220V$, $R=10\Omega$, $f=50Hz$, et $t=40ms$.

Charge RL:

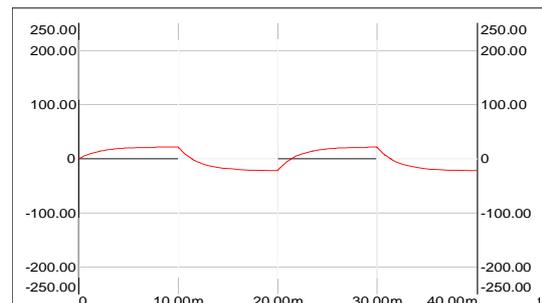
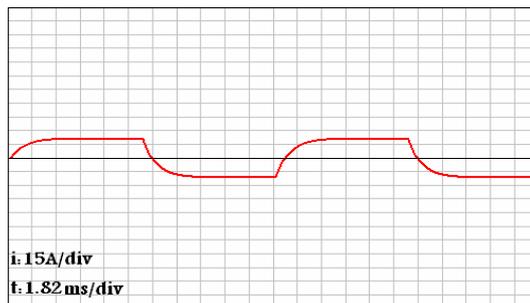
ELTSIM

SIMPLORER

Tension ondulée



Courant ondulé



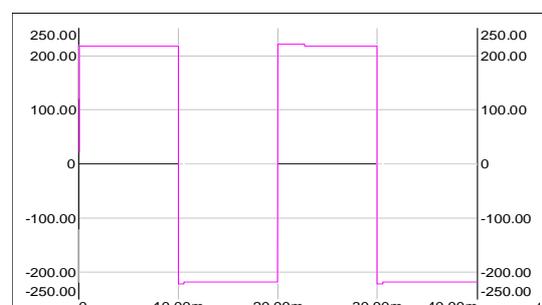
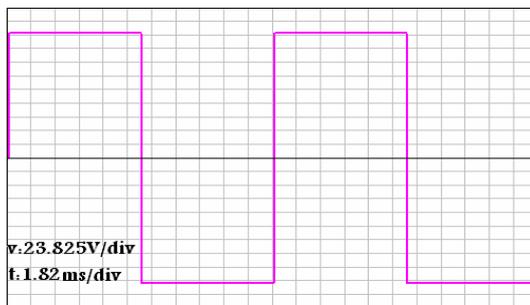
Figure(IV.26): Courant, et tension ondulés.
 $E_S=220V$, $R=10\Omega$, $L=20mH$, $t=40ms$. et $f=50Hz$.

Charge RLE(t):

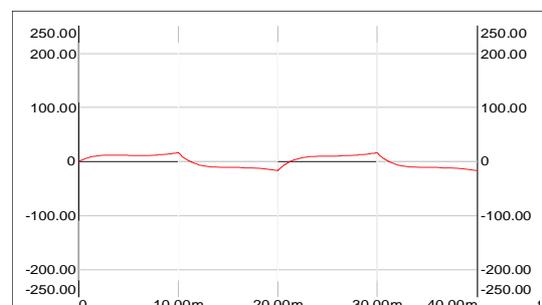
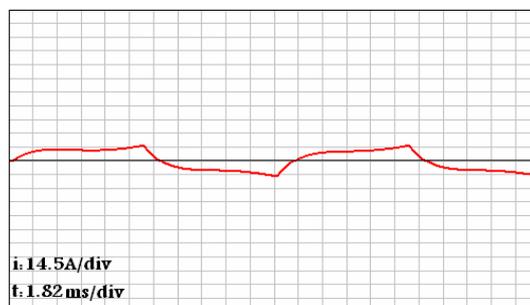
ELTSIM

SIMPLORER

Tension ondulée



Courant ondulé

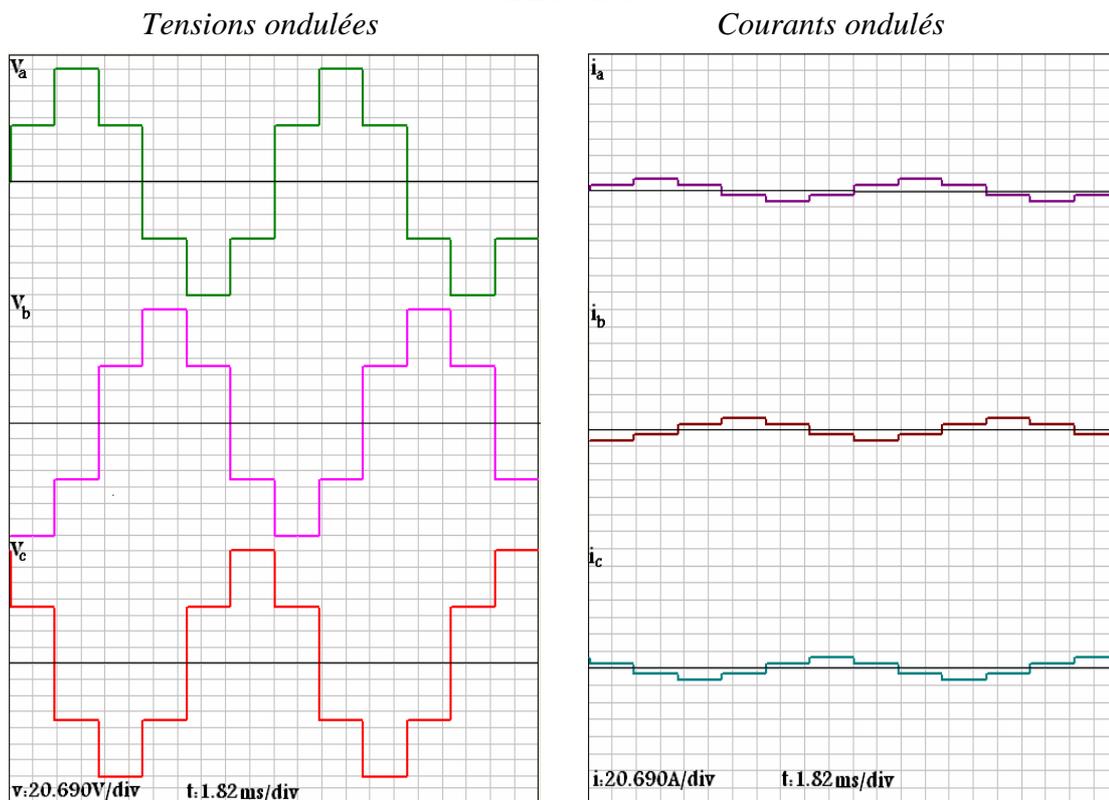


Figure(IV.27): Courant, et tension ondulés.
 $E_S=220V$, $R=10\Omega$, $L=20mH$, $E=110\sin(100\pi t)$, $t=40ms$. et $f=50Hz$.

IV.15. Onduleur triphasé:

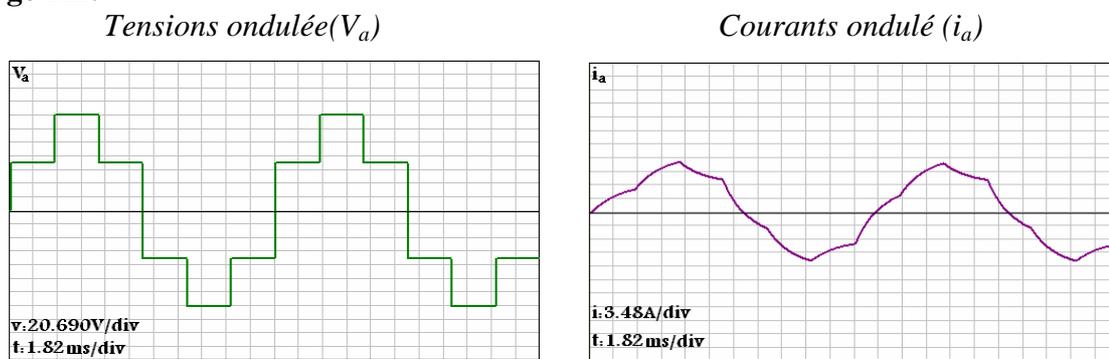
Charge R:

ELTSIM



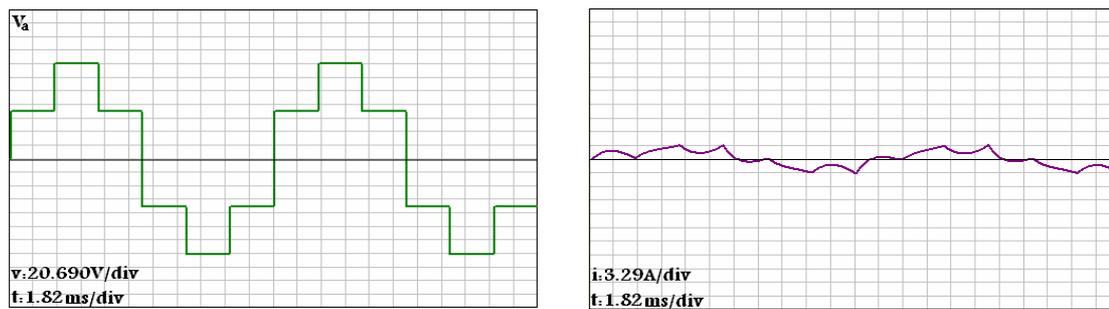
Figure(IV.28): Courants, et tensions ondulés.
 $E_S=220V, R=10\Omega, f=50Hz, \text{ et } t=40ms$

Charge RL:



Figure(IV.29): Courant, et tension ondulés.
 $E_S=220V, R=10\Omega, L=20mH, t=40ms. \text{ et } f=50Hz.$

Charge RLE(t):



Figure(IV.30): Courant, et tension ondulés.
 $E_S=220V, R=10\Omega, L=20mH, E=110\sin(100\pi t), t=40ms. \text{ et } f=50Hz.$

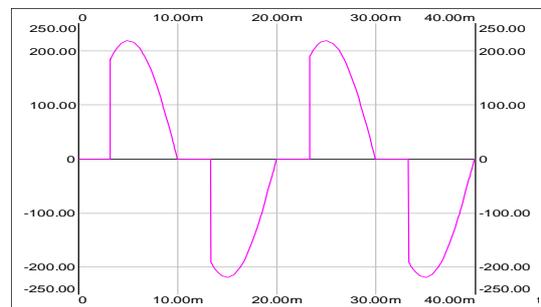
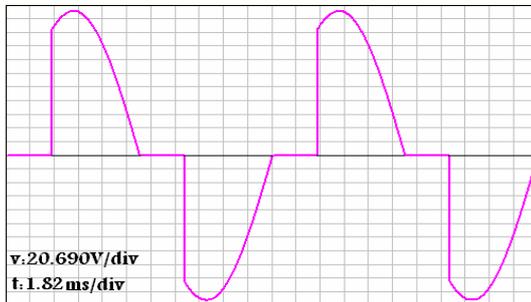
IV.16. Gradateur monophasé:

Charge R:

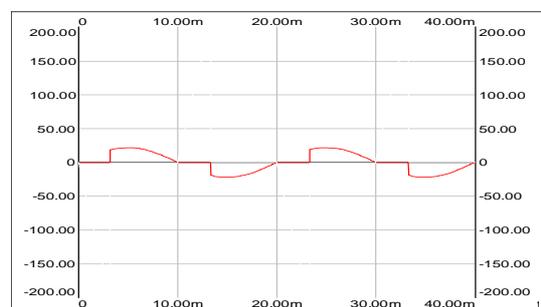
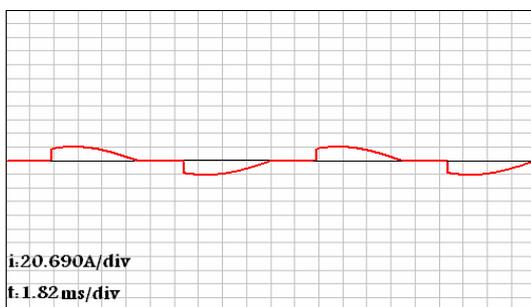
ELTSIM

SIMPLORER

Tensions ondulé



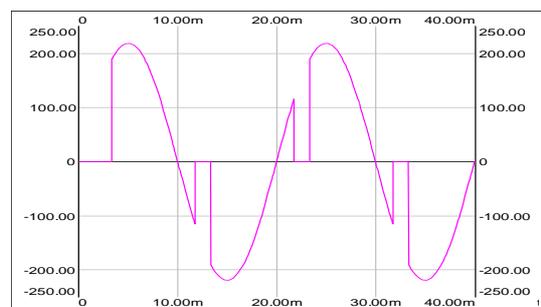
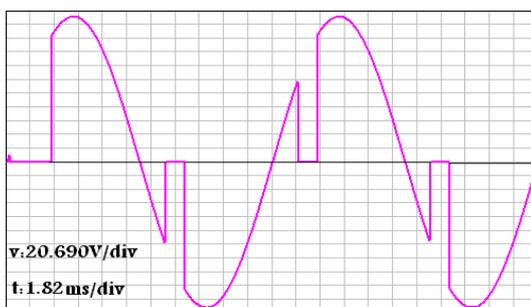
Courants ondulé



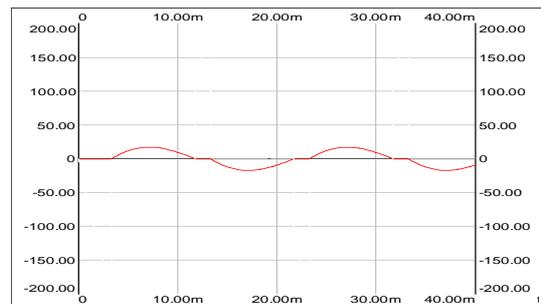
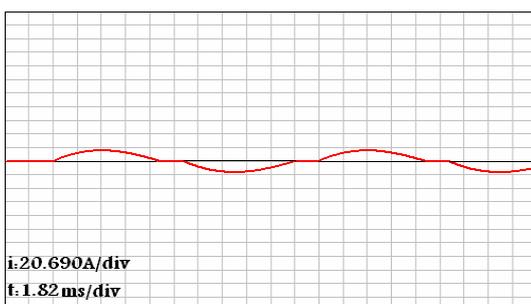
Figure(IV.31): Courant, et tension délivrés par un gradateur monophasé. $V_m=220V$, $R=10\Omega$, $f=50Hz$, $t=40ms$, et $\alpha=60^\circ$.

Charge RL:

Tensions ondulé



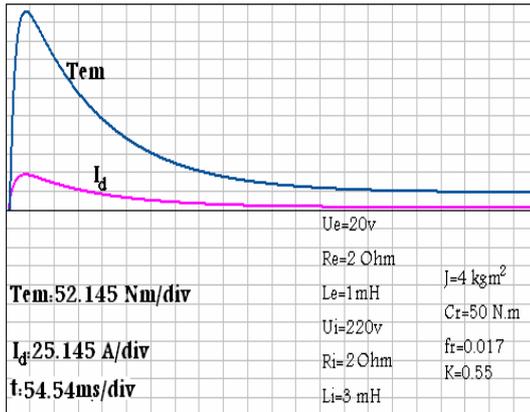
Courants ondulé



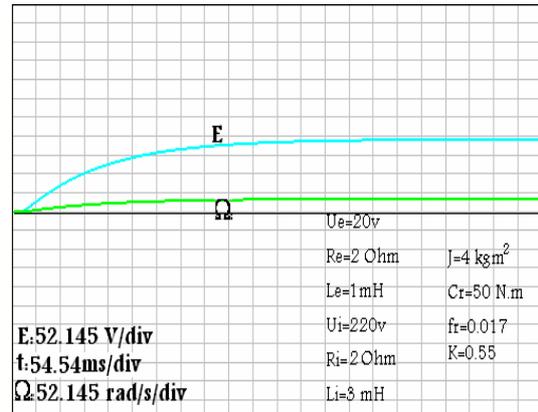
Figure(IV.32): Courant, et tension délivrés par un gradateur monophasé. $V_m=220V$, $R=10\Omega$, $L=20mH$, $f=50Hz$, $t=40ms$, et $\alpha=60^\circ$.

IV.17. Machine à courant continu:

- Alimentée par une tension continue :
Couple et courant de l'induit

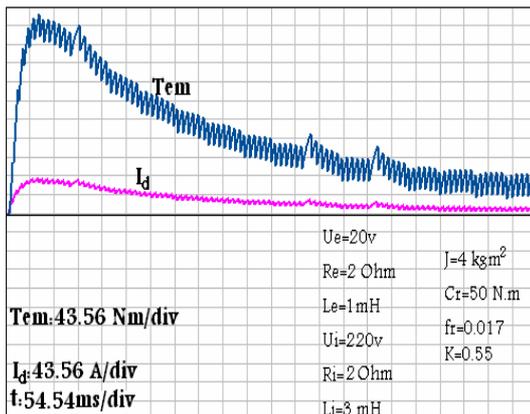


Vitesse et fem

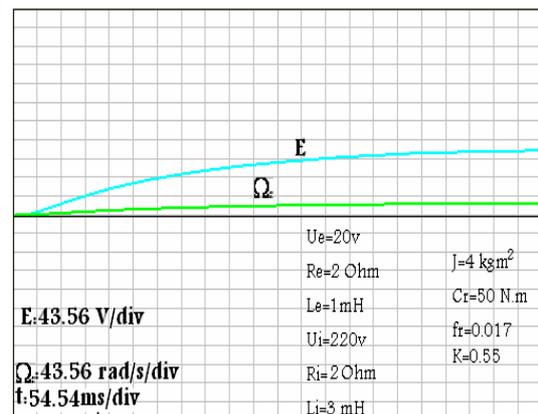


Figure(IV.33): Courant d'induit, tension d'induit, couple, et vitesse de MCC alimentée par une tension continue, $t=1,6s$.

- Alimentée par un hacheur série:
Couple et courant de l'induit

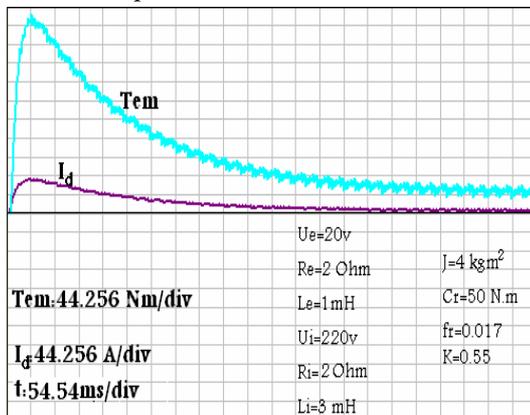


Vitesse et fem

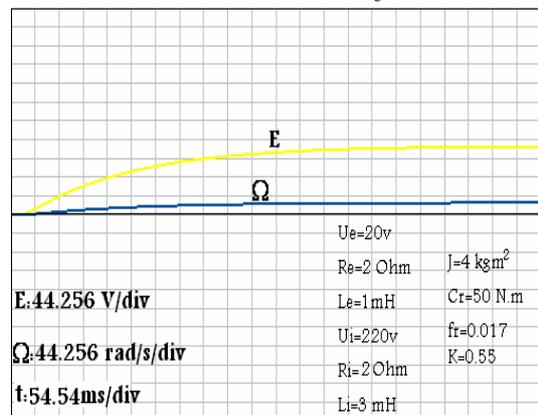


Figure(IV.34): Courant d'induit, tension d'induit, couple, et vitesse de MCC alimentée par un hacheur série, $f=2000Hz$, $\alpha =0.8$, et $t=1,2s$.

- Alimentée par un hacheur série:
Couple et courant de l'induit



Vitesse et fem



Figure(IV.35): Courant d'induit, tension d'induit, couple, et vitesse de MCC alimentée par un redresseur triphasé parallèle simple, $t=1,2s$.

IV.18. Interprétation des résultats:

Le couple et le courant d'induit d'une part, et la fem d'induit et la vitesse de rotation de la machine d'une autre part sont proportionnel, qui vérifient les deux équations $T_{em} = kI$, et $E = k \Omega$

Au démarrage le couple est élevé qui demande un courant d'induit élevé pour vaincre l'inertie de la machine et les frottement, et la vitesse augmente progressivement sous forme d'un exponentiel. Au régime permanent le couple se stabilise à la valeur $T_u = T_r + T_f$.

L'alimentation de la MCC par un hacheur série ou par un redresseur triphasé parallèle simple provoque des ondulations au niveau du couple de la machine, cette ondulation est liée à l'ondulation du courant d'induit (exponentiel croissant et décroissant). Les amplitudes des ondulations pour le hacheur série sont liés au rapport cyclique du hacheur série.

Dans les trois types d'alimentations le régime permanent est atteint pour l'alimentation en tension continue, redresseur triphasé parallèle simple, et puis par le hacheur série.

IV.19. Calculs des puissances:

Type de montage	Type de charge	Valeurs de simulation	Résultats	à diodes		à thyristors ($\alpha=60^\circ$)	
				ELTSIM	SIMPLORER	ELTSIM	SIMPLORER
Redresseur monophasé simple	R	10 Ω	$U_{dmoy}[V]$	70.028	/	53.16	/
			$I_{dmoy}[A]$	7.003	/	5.32	/
			P[W]	1210	1202.80	985.55	981.02
			Q[VAr]	1210	1202.59	1189.00	1183.07
			S[VA]	1711.19	1700.87	1544.35	1536.90
			Fp	0.707	0.707	0.638	0.638
	RL	10 Ω 20 mH	$U_{dmoy}[V]$	65.344	/	46.90	/
			$I_{dmoy}[A]$	6.46	/	4.75	/
			P[W]	962.09	942.06	670.65	642.89
			Q[VAr]	1169.58	1173.92	1061.92	1065.2
			S[VA]	1514.44	1505.18	390.51	1244.17
			Fp	0.635	0.626	0.534	0.517
	RLE	10 Ω 20 mH 110 V	$U_{dmoy}[V]$	130.97	/	126.31	/
			$I_{dmoy}[A]$	2.05	/	1.69	/
			P[W]	366.26	356.32	301.46	295.147
			Q[VAr]	4443.55	441.25	416.16	415.95
			S[VA]	575.23	567.157	513.88	510.03
			Fp	0.637	0.628	0.587	0.579

Type de montage	Type de charge	Valeurs de simulation	Résultats	à diodes		à thyristors ($\alpha=60^\circ$)	
				ELTSIM	SIMPLORER	ELTSIM	SIMPLORER
Redresseur monophasé en pont	R	10 Ω	$U_{dmoy}[V]$	140.06	/	106.31	/
			$I_{dmoy}[A]$	14.01	/	10.63	/
			P[W]	2420	2394.37	1971.10	1937.38
			Q[VAr]	0	10.803	940.65	945.77
			S[VA]	2420	2394.4	2184.04	2155.90
			Fp	1	0.99	0.902	0.899
	RL	10 Ω 20 mH	$U_{dmoy}[V]$	139.69	/	98.16	/
			$I_{dmoy}[A]$	13.45	/	9.36	/
			P[W]	2058.25	2056.91	1255.63	1297.42
			Q[VAr]	849.75	792.46	1269.45	1169.11
			S[VA]	2226.76	2204.29	1785.53	1746.46
			Fp	0.924	0.933	0.703	0.743
	RLE	10 Ω 20 mH 110 V	$U_{dmoy}[V]$	149.57	/		/
			$I_{dmoy}[A]$	4.09	/		/
			P[W]	691.49	705.64		
			Q[VAr]	428.49	365.50		
			S[VA]	813.49	794.68		
			Fp	0.850	0.888		

Type de montage	Type de charge	Valeurs de simulation	Résultats	à diodes		à thyristors ($\alpha=60^\circ$)	
				ELTSIM	SIMPLORER	ELTSIM	SIMPLORER
Redresseur triphasé parallèle simple	R	10 Ω	$U_{dmoy}[V]$	181.92	/	106.51	/
			$I_{dmoy}[A]$	18.19	/	10.65	/
			P[W]	1140.51	1134.46	637.27	601.49
			Q[VAr]	1208.00	1202.77	1065.87	1042.51
			S[VA]	1661.33	1653.38	1241.85	1203.59
			Fp	0.687	0.686	0.513	0.500
	RL	10 Ω 20 mH	$U_{dmoy}[V]$	181.61	/	90.27	/
			$I_{dmoy}[A]$	17.10	/	8.62	/
			P[W]	1007.67	1034.24	272.49	329.92
			Q[VAr]	1140.99	1137.95	884.08	826.95
			S[VA]	1522.25	1537.72	925.12	890.34
			Fp	0.662	0.673	0.295	0.371
	RLE	10 Ω 20 mH 110 V	$U_{dmoy}[V]$	185.08	/	133.06	/
			$I_{dmoy}[A]$	1.427	/	2.41	/
			P[W]	92.61	103.82	124.84	141.45
			Q[VAr]	152.08	147.82	302.47	284.78
			S[VA]	178.06	180.63	327.22	317.98
			Fp	0.52	0.575	0.382	0.445

Type de montage	Type de charge	Valeurs de simulation	Résultats	Série f=2kHz		Réversible à quatre quadrants f=2kHz	
				ELTSIM	SIMPLORER	ELTSIM	SIMPLORER
Hacheur $\alpha=0.5$	R	10 Ω	$U_{dmoy}[V]$	110	/	0.00	/
			$I_{dmoy}[A]$	11	/	0.00	/
			P[W]	2420	2411.44	4840	4805.17
			Q[VAr]	2420	2410.96	0.00	53.72
			S[VA]	3422.40	3409.95	4840.00	4805.47
			Fp	0.707		1.00	0.999
	RL	10 Ω 20 mH	$U_{dmoy}[V]$	110	/	0.37	/
			$I_{dmoy}[A]$	2.86	/	1.08	/
			P[W]	254.24	1005.23	12.92	17.426
			Q[VAr]	323.81	1082.43	296.59	181.27
			S[VA]	411.70	1477.21	296.87	182.11
			Fp	0.618	0.681	0.044	0.096
	RLE	10 Ω 20 mH 110 V	$U_{dmoy}[V]$	115.70	/	0.367	/
			$I_{dmoy}[A]$	0.600	/	-1.26	/
			P[W]	66.516	77.95	130.10	23.47
			Q[VAr]	88.31	97.49	427.22	2287.26
			S[VA]	110.56	124.83	446.59	2287.38
			Fp	0.602	0.625	0.291	0.01

Type de montage	Type de charge	Valeurs de simulation	Résultats	en demi- pont		en pont	
				ELTSIM	SIMPLORER	ELTSIM	SIMPLORER
Onduleur f=50Hz	R	10 Ω	U _{dmoy} [V]	0.00	/	0.00	/
			I _{dmoy} [A]	0.00	/	0.00	/
			P[W]	605.00	601.02	4840	4805.17
			Q[VAr]	605.00	600.54	0.00	53.72
			S[VA]	855.60	849.63	4840	4805.47
			Fp	0.770	0.707	1.00	0.999
	RL	10 Ω 20 mH	U _{dmoy} [V]	0.183	/	0.367	/
			I _{dmoy} [A]	0.598	/	0.664	/
			P[W]	413.27	423.491	3866.43	3150.29
			Q[VAr]	551.05	542.03	2003.34	2156.84
			S[VA]	688.80	687.85	4354.61	3817.89
			Fp	0.600	0.616	0.888	0.825
	RLE	10 Ω 20 mH 110 V	U _{dmoy} [V]	0.183	/	0.37	/
			I _{dmoy} [A]	0.34	/	0.93	/
			P[W]	111.46	120.46	1893.13	1995.69
			Q[VAr]	183.23	179.34	1512.59	1361.94
			S[VA]	214.46	216.04	2423.19	2416.12
			Fp	0.520	0.558	0.781	0.826

Type de montage	Type de charge	Valeurs de simulation	Résultats	monophasé ($\alpha=60^\circ$)		à thyristors	
				ELTSIM	SIMPLORER	ELTSIM	SIMPLORER
Gradateur	R	10 Ω	U _{dmoy} [V]	0.00	/	/	/
			I _{dmoy} [A]	0.00	/	/	/
			P[W]	1971.10	1946.96	/	/
			Q[VAr]	940.65	945.96	/	/
			S[VA]	2184.04	2164.60	/	/
			Fp	0.902	0.899	/	/
	RL	10 Ω 20 mH	U _{dmoy} [V]	-3.375	/	/	/
			I _{dmoy} [A]	0.26	/	/	/
			P[W]	1306.36	1292.1	/	/
			Q[VAr]	1193.68	1177.62	/	/
			S[VA]	1769.59	1748.23	/	/
			Fp	0.738	0.729	/	/
	RLE	10 Ω 20 mH 110 V	U _{dmoy} [V]	/	/	/	/
			I _{dmoy} [A]	/	/	/	/
			P[W]	/	/	/	/
			Q[VAr]	/	/	/	/
			S[VA]	/	/	/	/
			Fp	/	/	/	/

IV.20. Interprétation des résultats des calculs

La comparaison des résultats obtenus par *Eltsim* et celui obtenus par *Simplorer* a contribué à les remarques et les constatations suivantes:

- Les résultats obtenus dans les montages redresseurs sont presque les mêmes avec tous les types de charges, sauf avec le redresseur triphasé parallèle simple, où il y a de petites différences.
- Les résultats obtenus dans les montages hacheurs avec les charges R et RLE, sont presque les mêmes, par contre avec la charge RL il y a une grande différence.
- Les onduleurs soit en demi-pont ou en pont donnent de très bons résultats, qui sont très proches aux résultats obtenus par *Simplorer*.
- Le gradateur monophasé donne aussi des résultats proches avec les charges R et RL.

On constate que les petites différences entre les deux résultats sont liées aux méthodes de calcul, tel que l'intégration numérique et la résolution des équations différentielles, où *Eltsim* utilise les méthodes de Runge-Kutta et Simpson et *Simplorer* utilise la méthode de trapèze, et qui ont des régimes transitoires très courts.

Par contre les grandes différences, au plus des deux différentes méthodes de calcul, ils sont liés aux montages utilisés, c'est-à-dire qu'ils ont un temps de régime transitoire important, par exemple avec le hacheur série, le régime permanent est atteint après une huitième période complète (avec une fréquence de hachage de 2 kHz), où *Eltsim* utilise la première période comme référence de calcul.

CONCLUSION GENERALE

Nous avons vu la première version de *Elsim* qui présente un logiciel de simulation des convertisseurs statiques. Il a montré des résultats de simulation relativement les mêmes que ceux obtenus par *Simplorer*. Elsim suppose les semi-conducteurs idéaux. Il a été développé en utilisant le langage DELPHI.

Pour améliorer ce logiciel, on propose les idées suivantes :

- ✧ Restructurer les procédures et les fonctions programmées d'une manière généralisée pour plus de précision d'une part, et d'élever les capacités de calcul d'autre part, en ajoutant par exemple le pas de simulation variable

- ✧ Ajouter d'autres types de convertisseurs statiques (redresseur hexaphasé, redresseur triphasé parallèle double, hacheur parallèle, onduleur en courant, gradateur triphasé...etc.), d'autres types des charges alimentées (machine synchrone, machine asynchrone,...etc.);

- ✧ Ajouter la commande des machines électriques comme la régulation de vitesse, la limitation du courant de démarrage;

- ✧ Généraliser ce logiciel pour couvrir d'autres domaines de l'Electrotechnique tels que les réseaux électriques

ANNEXE

V.1. Autres composants de Delphi:



PopupMenu:

Utilisez *TPopupMenu* pour définir le menu surgissant qui apparaît quand l'utilisateur clique sur un contrôle avec le bouton droit de la souris. Pour qu'un menu surgissant soit disponible, affectez l'objet TPopupMenu à la propriété PopupMenu du contrôle.



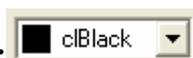
BitBtn:

TBitBtn est un contrôle bouton poussoir qui peut contenir une image sur sa face. Les boutons bitmap se comportent de la même manière que les contrôles bouton. Utilisez-les pour démarrer des actions dans des fiches ou des boîtes de dialogue.



PaintBox:

Utilisez *TPaintBox* pour ajouter des images personnalisées à une fiche. A la différence de TImage qui affiche une image stockée dans un bitmap ou un dessin, TPaintBox attend de l'application qu'elle dessine l'image directement dans un canevas. Utilisez le gestionnaire d'événement OnPaint pour dessiner dans la surface de dessin (Canevas) de la boîte à peindre.



ColorBox:

Utilisez *TColorBox* pour fournir à l'utilisateur une boîte à options déroulante dans laquelle il pourra sélectionner une couleur. Utilisez la propriété Style pour spécifier les couleurs devant être répertoriées par la boîte de couleurs. Chaque couleur est accompagnée d'un échantillon de la couleur correspondante.



ScrollBar:

Utilisez *TScrollBar* pour ajouter une barre de défilement indépendante dans une fiche. De nombreux contrôles disposent de propriétés ajoutant des barres de défilement qui font partie du contrôle même (objets TControlScrollBar). TScrollBar permet à l'utilisateur de faire défiler un groupe de contrôles ou a des contrôles ne disposant pas de barre de défilement intégrée ou groupement de contrôles.



TShape représente une forme géométrique pouvant être dessinée dans une fiche. Ajoutez un objet TShape à une fiche pour dessiner une forme géométrique simple dans une fiche.



TAnimate insérer un contrôle animation Windows. Un contrôle animation est une fenêtre affichant, silencieusement, une séquence AVI. Une séquence AVI est une série de plans bitmap similaire à un film.

V.2. Quelques relations:

< Plus petit

= Egal

<= Inférieur ou égal

> Plus grand

<> Différent

>= Supérieur ou égal

V.3. Moteur à courant continu (excitation séparée):

V.3.1. Description, rôle des éléments:

Le moteur à courant continu constitué d'un inducteur (stator) et d'un induit (rotor).

- Inducteur: est alimenté en courant continu (ou à aimant permanent). Il crée un champ magnétique dans l'entrefer.
- Induit: il est alimenté avec un générateur de tension continue variable à travers l'ensemble collecteur/balais. Les conducteurs de l'induit sont parcourus par un courant alternatif, dans un champ magnétique créé par l'inducteur. Ils sont soumis à des forces de Laplace, l'induit tourne.
- Collecteur/balais: il sert «d'interface» entre l'alimentation et les conducteurs de l'induit. [07]

V.3.2. Schéma équivalent

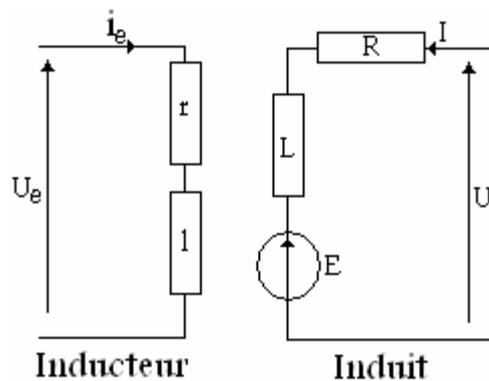


Figure (V.1): Schéma équivalent du moteur à courant continu

On peut écrire: $U_e = r i_e$ $U = RI + E$ (V.1)

Expression de la force électromotrice:

L'expression est: $E = K\phi \Omega$

ϕ : Flux utile sous un pôle [Wb]. Il dépend du courant d'excitation i_e . Lorsque la machine n'est pas saturée, ϕ est proportionnel à i_e

Ω : vitesse de rotation [rad/s], $\Omega = \frac{2\pi n}{60}$ (V.2)

, avec n vitesse en [tr/mn]

K : est une constante de la machine

Lorsque le courant d'excitation est constant, l'expression de E devient $E = k\Omega$ (V.3)

Avec $k = K\phi$ (V.4)

V.3.3. Expression du couple

Le moment du couple électromagnétique T_{em} [Nm] s'exprime (à flux constant) par:

$$T_{em} = \frac{EI}{2\pi n} = K\phi I \quad \text{Ou} \quad (\text{V.5})$$

$$T_{em} = kI$$

V.3.4. Point de fonctionnement:

La charge impose une couple résistant T_r ; en régime permanent, le moteur doit fournir un moment de couple utile:

$$T_u = T_r + T_f \quad (\text{V.6})$$

Généralement le couple de frottement est proportionnel à la vitesse de rotation

$$T_f = f_r \Omega, \quad \text{où } f_r [\text{N.m.s/rad}] \text{ est le coefficient de frottement} \quad (\text{V.7})$$

Les équations de fonctionnement en régime transitoire (machine non saturée) sont :

$$\begin{cases} U_e = r i_e + l \frac{di_e}{dt} \\ U = R i + L \frac{di}{dt} + K' i_e \Omega \\ K' i = T_r + f_r \Omega + J \frac{d\Omega}{dt} \\ K\phi = K' i_e \end{cases} \quad (\text{V.8})$$

Avec $l[H]$: est l'inductance propre de l'inducteur,

$L[H]$: est l'inductance propre de l'induit,

$r[\Omega]$: est la résistance de l'inducteur,

$R[\Omega]$: est la résistance de l'induit,

$J[kg.m^2]$: Le moment d'inertie de la machine.

V.4. Décomposition en série de Fourier:

Si f est une fonction périodique de période T , est intégrable, on peut la décomposer en série de Fourier comme suit:

$$f(x) = a_0 + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx) \quad (\text{V.9})$$

$$\text{Tel que: } a_0 = \frac{1}{2T} \int_{-T}^{+T} f(x) dx \quad (\text{V.10})$$

$$a_k = \frac{1}{2T} \int_{-T}^{+T} f(x) \cos kx dx \quad (\text{V.11})$$

$$b_k = \frac{1}{2T} \int_{-T}^{+T} f(x) \sin kx dx \quad (\text{V.12})$$

La fonction f peut être écrit aussi autrement:

$$f(x) = a_0 + \sum_{k=1}^{\infty} A_k \sin(kx + \varphi_k), \quad (\text{V.13})$$

$$\text{Où } A_k = \sqrt{\frac{a_k^2 + b_k^2}{2}} \quad (\text{V.14})$$

$$\varphi_k = \text{artg}\left(\frac{b_k}{a_k}\right) \quad (\text{V.15})$$

V.5. Intégration numérique:

Nous avons choisit la méthode de Simpson, qui utilise le polynôme d'interpolation d'ordre 2

$$I_2(f) = \frac{b-a}{6} [f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)] \quad (\text{V.16})$$

$$I_{2,m}(f) = \frac{h}{6} \sum_{k=0}^{m-1} [f(x_k) + 4f\left(\frac{x_k + x_{k+1}}{2}\right) + f(x_{k+1})] \quad (\text{V.17})$$

Exemple:

$$I = \int_0^{2\pi} x e^{-x} \cos(2x) dx$$

$$\text{La solution exacte est : } I = \frac{3(e^{2\pi} - 1) - 10\pi \cdot e^{2\pi}}{25} = -0.1221226046$$

La solution par la méthode **Simpson** est : $I = -0.12209214813$.

La solution par la méthode de **Trapèze** est : $I = -0.13055053965359$.

V.6. Résolution numérique d'équations (méthode de Runge- Kutta):**Méthode de Runge- Kutta pour une seule équation:**

$$\begin{cases} F_1 = F(t_0, x_0, U_0) \\ t_1 = t + \frac{h}{2} \\ x_1 = x + \frac{h}{2} F_1 \\ U_1 = U(t_1) \end{cases} \quad (\text{V.18})$$

$$\begin{cases} F_2 = F(t_1, x_1, U_1) \\ t_2 = t + \frac{h}{2} \\ x_2 = x + \frac{h}{2} F_2 \\ U_2 = U(t_2) \end{cases} \quad (\text{V.19})$$

$$\begin{cases} F_3 = F(t_2, x_2, U_2) \\ t_3 = t + h \\ x_3 = x + h F_3 \\ U_3 = U(t_3) \end{cases} \quad (\text{V.20})$$

$$F_4 = F(t_3, x_3, U_3) \quad (\text{V.21})$$

$$x_{n+1} = x_n + h[F_1 + 2F_2 + 2F_3 + F_4] \quad (\text{V.22})$$

Méthode de Runge- Kutta pour deux équations:

$$\begin{cases} t = t_0 \\ x = x_0 \\ U = U(t_0) \\ x' = x'_0 \\ U' = U'(t_0) \end{cases} \quad (\text{V.23})$$

$$\begin{aligned} f_1 &= f_1(t_0, x_0, x'_0, U(t_0), U'(t_0)) \\ f'_1 &= f'_1(t_0, x_0, x'_0, U(t_0), U'(t_0)) \end{aligned} \quad (\text{V.24})$$

$$\begin{cases} t = t_0 + \frac{h}{2} \\ x = x_0 + \frac{h}{2} f_1 \\ U = U(t_0 + \frac{h}{2}) \\ x' = x'_0 + \frac{h}{2} f \\ U' = U'(t_0 + \frac{h}{2}) \end{cases} \quad (\text{V.25})$$

$$f_2 = f_2(t_0 + \frac{h}{2}, x, x', U, U') \quad (V.26)$$

$$f_2' = f_1'(t_0 + \frac{h}{2}, x, x', U, U')$$

$$\left\{ \begin{array}{l} t = t_0 + \frac{h}{2} \\ x = x_0 + \frac{h}{2} f_2 \\ U = U(t_0 + \frac{h}{2}) \\ x' = x'_0 + \frac{h}{2} f_2' \\ U' = U'(t_0 + \frac{h}{2}) \end{array} \right. \quad (V.27)$$

$$f_3 = f_3(t + \frac{h}{2}, x, x', U, U') \quad (V.28)$$

$$f_3' = f_3'(t + \frac{h}{2}, x, x', U, U')$$

$$\left\{ \begin{array}{l} t = t_0 + h \\ x = x_0 + hf_3 \\ U = U(t_0 + h) \\ x' = x'_0 + hf_3' \\ U' = U'(t_0 + h) \end{array} \right. \quad (V.29)$$

$$f_4 = f_4(t + \frac{h}{2}, x, x', U, U') \quad (V.30)$$

$$f_4' = f_4'(t + \frac{h}{2}, x, x', U, U')$$

V.7. THD:

$$U_2 = U_{d0}^2 + U_h^2 \quad (V.31)$$

$$1 = \frac{U_{d0}^2}{U^2} + \frac{U_h^2}{U^2} \quad (V.32)$$

$$\text{Sachant que: } THD = \frac{U_h}{U} \quad (V.33)$$

$$THD = \sqrt{1 - \frac{U_{d0}^2}{U^2}} \quad (V.34)$$

V.8. Validation des résultats pour les harmoniques:

On prend le redresseur monophasé simple voie pour voir la similitude entre *Eltsim* et *Simplorer* sur l'amplitude des harmoniques.

Redressement monophasé simple à diode.

Charge R:

Le numéro D'harmonique	Fréquence (Hz)	Amplitude d'harmonique (Volts)	
		Eltsim	Simplorer
1	50	110.00	110.00
2	100	46.69	45.37
3	150	0.00	0.00
4	200	9.34	10.46
5	250	0.00	0.00
6	300	4.00	4.49
7	350	0.00	0.00
8	400	2.22	2.51
9	450	0.00	0.00
10	500	1.41	1.60

Charge RL:

Le numéro D'harmonique	Fréquence (Hz)	Amplitude d'harmonique (Volts)	
		Eltsim	Simplorer
1	50	113.50	113.68
2	100	53.99	54.78
3	150	8.75	9.85
4	200	13.52	13.67
5	250	7.74	8.57
6	300	6.05	6.06
7	350	6.40	6.85
8	400	3.57	3.72
9	450	4.93	5.07
10	500	3.14	3.49

BIBLIOGRAPHIE

- [01] M.C BELAID & BARDOU, « **DELPHI 5** » Edition BERTI, Algérie 2001.
- [02] M.C BELAID, « **Borland Delphi** » Edition Les pages bleus, Algérie 2005.
- [03] M.C BELAID «**Programmer les bases de données sous Delphi**» Edition Les pages Bleus, Algérie 2005.
- [04] Guy SEGUIER «**L'électronique de puissance**» Edition Dunod 2005.
- [05] Valérie LEGER, Alain JAMEAU «**conversion de l'énergie Electrotechnique, Electronique de puissance**» Edition ellipses 2004.
- [06] Christian ROMBAUT, Guy SEGUIER, Robert BAUSIERE «**Les convertisseurs de l'électronique de puissance, 2 La conversion alternatif- alternatif** » Edition Lavoisier 1986].
- [07] Laroche Jacques «**L'électronique de puissance, cours et exercices corrigés**» Edition Dunod 2004.
- [08] GUERTI Mhania, DOUKHALI Mourad «**Réseau de Petri**», PFE en Electronique, ENP Alger 1978.
- [09] R.SIACI «**Etude et commande d'une structure minimale de conversion directe AC-AC** », Magistère en Automatique ENP Alger 2001.
- [10] A. Abrous, A.Oukali «**Analyse, commande et protection d'un convertisseur matriciel alimentant une MAS triphasé**», PFE en G. électrique, ENP Alger 2004.
- [11] A. Bounceur, Cours Delphi.
- [12] Le programmeur apprenez Delphi3 en 14 jours, O.DAN- G.STEVE- B.STEVE, France, 1997.
- [13] Guide du langage pascal objet, Borland Delphi 5, Irlande, 1998.
- [14] Guide du développeur, Borland Delphi 5, Irlande, 1999.
- [15] Prise en main, Borland Delphi 5, Irlande, 1999.
- [16] Developer's guide, Borland Delphi 7, <http://www.borland.com>, U.S.A, 2002.
- [17] Delphi, <http://www.unrrai.com>, janvier 2003.
- [18] Site Internet: [http:// www.developpez.com](http://www.developpez.com).
- [19] Site Internet: [http:// www.phidels.com](http://www.phidels.com).
- [20] Site Internet: [http:// www.delphifr.com](http://www.delphifr.com).
- [21] Site Internet: [http:// www.delphipage.free.fr](http://www.delphipage.free.fr).
- [22] Site Internet: [http:// fr.wikipedia.org](http://fr.wikipedia.org).