

# Remerciements

---

Comme un tel travail ne s'effectue jamais seul, j'aimerais au début remercier par ces quelques mots tous ceux qui, de près ou de loin, ont contribué à le rendre possible et m'ont aidé à le réaliser.

Je tiens à remercier tout particulièrement MR. Tadriss R et MR. Mamoune pour avoir accepté l'encadrement scientifique de ce mémoire

Je leur suis reconnaissant pour m'avoir dirigé et assisté tout au long de ce travail. Je les remercie également pour leurs nombreux conseils et remarques qui m'ont été d'une grande utilité.

Je tiens à remercier aussi le président et les membres de jury, qui m'ont fait l'honneur d'accepter de juger ce travail.

Je remercie également tous mes collègues et amis sans oublier personne.

Que toutes et tous ceux qui ont fait que ce travail aboutisse trouvent ici l'expression de mes remerciements les plus sincères

# *Dédicace*

*Je dédie ce travail à mes très chers parents pour le courage et le soutien pendant  
mes années d'études*

# Introduction générale

---

## Introduction générale

Les concepteurs de systèmes de commandes modernes s'investissent dans le control de projets associant plusieurs disciplines et technologies. Pour répondre aux différents besoins notamment la conception des systèmes de commande, le recours aux outils de **Conception Assistée par Ordinateur (CAO)** est plus qu'indispensable. L'avantage des méthodes de CAO est de faire une conception matérielle et logicielle simultanément afin de réduire le temps de développement et d'augmenter la fiabilité par le test des prototypes virtuels avant la réalisation sur un circuit intégré. Ce dernier sera le lieu d'implantation de la solution finale. Dans des approches pluridisciplinaires en particulier la mécatronique, il est indispensable de mettre en place des méthodes et des outils facilitant l'intégration de solution analogiques, numériques et mixtes. L'introduction et l'accumulation d'innovations technologiques sur les circuits intègres ainsi que leurs expansion a permis l'automatisation des taches considérées complexes auparavant. L'approche classique de programmation séquentielle qui est une solution logicielle est considérée insuffisante pour la commande en temps réel car les exigences de temps d'exécution ne cessent d'augmenter. L'approche par la programmation architecturale qui est une solution matérielle permet de surpasser relativement cet inconvénient. Afin de développer des systèmes de commande de très hautes performances et d'améliorer la dynamique des convertisseurs dc /dc, la conversion d'énergie d'une forme à une autre a pris une importance considérable ces dernières décennies et constitue un domaine de recherche très dynamiques. Mais la complexité croissante des algorithmes de commande des systèmes électriques ne cessent d'augmenter à cause des contraintes liées aux modeles (non-linéarités, couplage, variation de paramètres....etc.) et d'autres contraintes liées aux performances (temps d'exécution, précision...etc.). Motivé par les exigences grandissantes en puissance de traitement afin de répondre aux exigences strictes concernant les performances exigeantes en puissance de calcul, les FPGA représentent une alternative pour compenser les DSP qui sont devenus classique avec une grande souplesse et de bonnes performances (parallélisme de traitement, vitesse, surface, consommation....etc.). La logique programmable FPGA permet l'intégration de la circuiterie numérique de

commande dans les systèmes automatisés et particulièrement la commande en temps réel des convertisseurs. Alors, les FPGA présentent beaucoup de perspectives pour l'implantation d'algorithmes de contrôle de ces derniers.

Pour ces FPGA, les outils de conception assistée par ordinateur servent à passer directement d'une description fonctionnelle en VHDL à un schéma en portes logiques. Ces outils ont révolutionné la conception des circuits numériques tels que les ASIC et les FPGA. Afin de mieux exploiter les avantages de ces circuits, nous avons ainsi opté dans notre travail pour ces derniers pour la commande des convertisseurs dc-dc.

Ce travail de thèse a pour objectif l'élaboration d'une étude complète avec conception, Simulation et implémentation d'un circuit numérique d'une commande pour les convertisseurs DC/DC à base d'un circuits reconfigurables FPGA en vue d'améliorer les performances et la fiabilité de ces derniers. C'est dans ce contexte que cette thèse est élaborée avec la mise au point d'une solution architecturale pour des algorithmes en temps réel. La démarche scientifique adoptée dans cette thèse suit cette structuration:

- Le premier chapitre est consacré à l'état de l'art des circuits reconfigurables FPGA. On traitera l'architecture interne de ces derniers et une comparaison des FPGA avec les circuits séquentiels classiques puis la contribution des FPGA pour la commande des convertisseurs.
- Le second chapitre traitera du rôle des convertisseurs en général dans la conversion d'énergie, ainsi leur rôle dans le domaine de l'industrie et nous présentons brièvement la modélisation et leur commande.
- Dans le troisième chapitre, nous présentons des approches numériques de la commande par mode de glissement dédiée aux convertisseurs DC/DC.
- Quant au quatrième et dernier chapitre, il sera consacré à l'implémentation sur cible FPGA de la commande de deux convertisseurs à savoir l'abaisseur (Buck) et l'élévateur (Boost) par mode de glissement.
- Enfin, nous clôturons ce document avec une conclusion générale ainsi que des perspectives d'exploitation de la technologie programmable FPGA qui est prometteuse et en pleine maturité.

Introduction générale .....	9
-----------------------------	---

## **Chapitre 1 : Etat de l'art des FPGA**

1.1 INTRODUCTION .....	11
1.2 FACTEURS D'EVOLUTION DES CIRCUITS NUMERIQUES .....	12
1.3 LES TECHNOLOGIES DE MEMORISATION .....	13
1.4 LES CIRCUITS PROGRAMMABLES .....	14
1.5 LES CIRCUITS LOGIQUES PROGRAMMABLES .....	16
1.6 Les Circuits Logiques Programmables du type FPGA.....	17
1.6.1 Critères de choix du circuit programmable FPGA .....	17
1.6.2 Différent domaines d'application des FPGA .....	18
1.6.3 Principaux fondateurs d'FPGA .....	18
1.6.4 Configuration et reconfiguration des FPGA .....	19
1.6.5 Technologies de programmation des FPGA .....	19
1.6.5.1 Technologie à base de RAM (XILINX et ALTERA) .....	20
1.6.5.2 Technologie à base d'EEPROM ou FLASH (LATTICE et ACTEL).....	20
1.6.5.3 Technologie à base d'ANTI-FUSIBES(ACTEL).....	21
1.6.6 Architecture interne des FPGA .....	21
1.6.7 Architecture MULTI-COMPOSANTS .....	23
1.6.7.1 Association de plusieurs FPGA .....	23
1.6.7.2 Association d'un microprocesseur et d'un FPGA.....	24
1.6.8 Avantages et inconvénients des FPGA .....	24
1.6.9 Les deux grandes familles architecturales d'FPGA.....	25
1.6.9.1 Les circuits FPGA à base de « LUT » (Look Up Tables) .....	25
1.6.9.2 Les circuits FPGA à base de multiplexeurs « MUX » .....	25
1.7 LA CONFIGURATION DES FPGA PAR LES OUTILS CAO .....	26
1.7.1 De l'algorithmique à la conception CAO .....	26
1.7.2 Méthodologie de conception .....	27
1.8 Conclusion.....	29

## **Chapitre 2 : Modélisation de convertisseurs DC/DC**

2.1 Introduction .....	30
2.2 Convertisseurs DC/DC (hacheurs).....	30
2.3 Différents types d'hacheurs .....	30

2.3.1 Convertisseur Boost .....	31
2.3.2 Convertisseur Buck.....	34
2.3.3 Convertisseur Buck-Boost .....	36
2.3.4 Convertisseur Cuk.....	38
2.4 Récapitulatif d'analyse et choix du convertisseur .....	39
2.5 Conclusion.....	41

## **Chapitre3 : Commande des Convertisseurs DC-DC par Mode Glissant**

3.1. Introduction .....	42
3.2. Système à structure variable .....	42
3.3. Commande par mode Glissant.....	43
3.3.1. Principe .....	43
3.3.2. Objectif de la commande par mode glissant.....	45
3.3.3. Choix de la surface de glissement .....	45
3.3.4. Condition d'existence du glissement .....	46
3.4 Commande de Convertisseur DC-DC Buck par mod Glissant.....	46
3.4.1. Modélisation du système .....	46
3.4.2 Conception de la commande par MG .....	48
3.4.3 Conditions d'existence de la surface de glissement :.....	48
3.4.4 Commande équivalente : .....	49
3.4.5 Équation d'état en MG .....	49
3.4.6 Résultats de simulation .....	50
3.5 Commande par MG du convertisseur Boost DC/DC .....	52
3.5.1 Résultats de simulation .....	54
3.6 Conclusion.....	57

## **CHAPITRE 4 : Modèle Xilinx System Generator**

4.1 Brève description .....	58
4.2 Méthode de développement.....	58
4.2.1 Développement de bloc individuel et validation de premier niveau.....	59
4.2.1.1 méthodes simultanées .....	60
4.2.1.2 la sainte croyance .....	61
4.3 Développement d'ensemble, liaisons et validation de deuxième niveau .....	62
4.4 Contrôle vectoriel .....	62
4.5 Simulation du convertisseur Buck sous l'environnement XSG .....	63

4.6 Résultat de simulation (XSG).....	64
4.7 Simulation du convertisseur Boost sous l'environnement XSG .....	65
4.8 Comparaison : .....	66
Conclusion Générale .....	69

# Liste des Figures

---

## Chapitre :1

<i>Figure 1.1.</i> Diagramme d'évolution des coûts des dernières années. ....	12
<i>Figure 1.2.</i> les différents type de mémoires .....	15
<i>Figure 1.3.</i> Schéma comparatif d'un DSP et d'un FPGA. ....	16
<i>Figure 1.4.</i> Diagramme des différents types de circuits logiques programmables.....	16
<i>Figure 1.5.</i> Critères de choix du circuit logique programmable FPGA.....	19
<i>Figure 1.6.</i> Statistiques du marché par les vendeurs d'FPGA.....	19
<i>Figure 1.7.</i> Reprogrammabilité sur site d'un FPGA. ....	19
<i>Figure 1.8.</i> Caractéristiques des technologies SRAM. ....	20
<i>Figure 1.9.</i> Caractéristiques des technologies FLASH.....	21
<i>Figure1.10.</i> Architecture interne d'un FPGA. ....	22
<i>Figure 1.11.</i> Exemples d'association entre FPGA. ....	23
<i>Figure 1.12.</i> Exemples d'association entre FPGA et Microprocesseur.....	24
<i>Figure 1.13.</i> Exemple d'implémentation sur LUT. ....	25
<i>Figure 1.14.</i> Exemple d'implémentation sur des multiplexeurs.....	26
<i>Figure 1.15.</i> Mode d'exécution matériel de la CAO.....	27

## Chapitre :2

<i>Figure 2.1.</i> Formes d'ondes des courants et tensions du convertisseur Boost.....	31
<i>Figure 2.2.</i> Circuit équivalent du Boost interrupteur fermé.....	32
<i>Figure 2.3.</i> Circuit équivalent du Boost interrupteur ouvert . ....	32
<i>Figure 2.4.</i> Formes d'ondes des courants et tensions du convertisseur Buck. ....	34
<i>Figure 2.5.</i> Formes d'ondes des courants et tensions du convertisseur Buck-Boost.....	37
<i>Figure 2.6.</i> Formes d'ondes des courants et tensions du convertisseur Cuk. ....	38
<i>Figure 2.7.</i> Evolution du gain en tension en fonction du rapport cyclique des convertisseurs .....	39
<i>Figure 2.8.</i> Evolution du gain en tension en tenant compte des éléments parasites des convertisseurs.....	40

## Chapitre :3

<i>Figure 3.1.</i> Configuration par commutation au niveau de la contre-réaction d'état. ....	43
<i>Figure 3.2.</i> Configuration par commutation au niveau de l'organe de Commande. ....	43
<i>Figure 3.3.</i> Principe de la commande par mode glissant.....	44
<i>Figure 3.4.</i> Circuit de Commande du Convertisseur DC-DC Buck par Mode Glissant.....	47
<i>Figure 3.5.</i> Représentation schématique d'un convertisseur Buck. ....	47



<i>Figure 3.6.</i> Bloc de Commande à boucle fermée.....	47
<i>Figure 3.7.</i> Tension de sortie (Buck) à boucle ouvert .....	50
<i>Figure3.8.</i> Courant d'inductance (Buck) .....	50
<i>Figure 3.9.</i> Tension de sortie (Buck) à boucle fermée .....	51
<i>Figure3.10.</i> Tension d'entrée (Buck) .....	51
<i>Figure 3.11.</i> Circuit de Commande du Convertisseur DC-DC Boost par Mode Glissant.....	53
<i>Figure 3.12.</i> Représentation schématique d'un convertisseur Boost .....	54
<i>Figure 3.13.</i> Bloc de commande à boucle fermée .....	54
<i>Figure3.14.</i> Tension de sortie à boucle ouverte (Boost).....	55
<i>Figure 3.15.</i> Courant d'inductance (Boost).....	55
<i>Figure3.16</i> Tension de sortie à boucle fermée (Boost).....	56
<i>Figure 3.17.</i> Tension d'entrée (Boost) .....	56

#### **Chapitre :4**

<i>Figure 4.1.</i> Propriétés d'un bloc XSG .....	64
<i>Figure 4.2.</i> Simulation de convertisseur Buck sous (XSG).....	64
<i>Figure 4.3.</i> Bloc la commande sous xilinx system generator (XSG) .....	65
<i>Figure 4.4.</i> Tension de sortie (Buck) sous (XSG) .....	65
<i>Figure 4.5.</i> Courant d'inductance (Buck) sous XSG.....	67
<i>Figure 4.6.</i> Simulation de convertisseur Boost en (XSG) .....	67
<i>Figure 4.7.</i> Bloc la commande sous xilinx system generator (XSG).....	67
<i>Figure 4.8.</i> Tension de sortie (Boost) sous (XSG).....	68
<i>Figure 4.9.</i> Courant d'inductance (Boost) sous XSG .....	68
<i>Figure 4.10.</i> Comparaison tension de sortie(Buck) XSG, Boucle fermée et Boucle ouvert .....	68
<i>Figure 4.11.</i> Comparaison tension de sortie(Boost) XSG, Boucle fermée et Boucle ouvert .....	69

## Liste des acronymes

---

**ASIC** : Application Specific Integrated Circuit.

**CAO** : Conception Assistée par Ordinateur

**CLB** : Configurable Logique Bloc

**CPLD** : Complex Programmable Logic Device

**DSP** : Digital Signal Processor.

**EPROM** : Erasable Programmable Read Only Memory.

**EEPROM** : Electrically Erasable Programmable Read Only Memory

**EEPLD** : Electrically Erasable Programmable Logic Device

**FPGA** : Field Programmable Gate Array

**PLD** : Programmable Logic Device

**PROM** : Programmable Read Only Memory.

**ROM** : Read Only Memory.

**SRAM** : Static Random Access Memory.

**SOC** : Systems On Chips

**VHDL** : Very high speed integrated circuit Hardware Description Language .

**XSG** : Xilinx Système Generator

**DC/DC** : Conversion continu/continu

## Liste des tableaux

*Tableau 1.1.* Comparaison des différentes solutions numériques.

*Tableau 2.1.* Récapitulatif des caractéristiques des convertisseurs continu-continu

Tableau 4.1 Comparaison tension de sortie(Buck) XSG, Boucle fermée et Boucle ouverte

Tableau 4.2 Comparaison tension de sortie(Boost) XSG, Boucle fermée et Boucle ouverte

# Chapitre 1 : Etat de l'art des FPGA

---

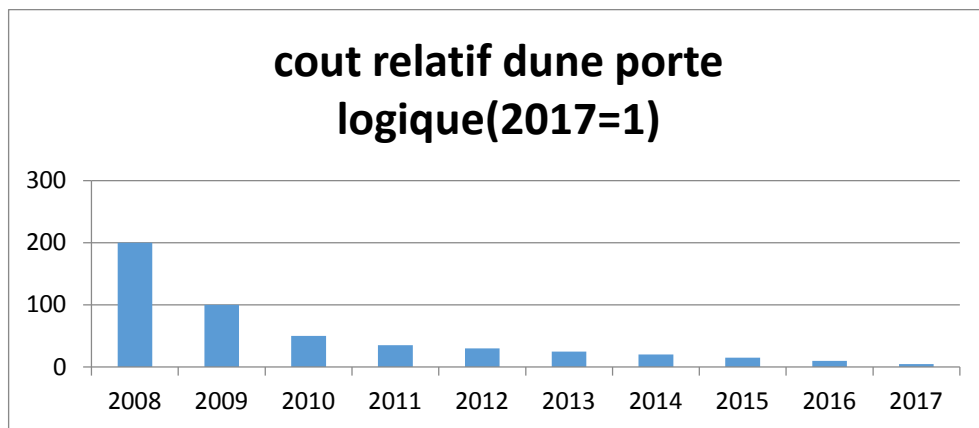
## 1.1 INTRODUCTION

L'histoire des circuits intégrés a commencé après l'invention du transistor en 1947 par les laboratoires Bell. Relativement, l'apparition du premier circuit intégré est au cours des années 1958 et 1959 grâce à un jeune ingénieur du nom de Jack Kilby de la firme Texas Instruments qui a intégré sur un même substrat de silicium plusieurs éléments électroniques (transistors, résistances, capacités) ce qui est qualifié à cette époque le premier circuit intégré. Juste après, dans les années 60, et plus précisément en 1965, un des fondateurs de la compagnie Intel nommé Gordon Moore a fait une étude concernant l'évolution du secteur des circuits intégrés, ce qui a lui permis de prédire que le taux d'intégration des transistors dans ces circuits double tous les deux ans. A nos jours, ce constat reste vérifiée sur les circuits FPGA qui n'ont pas échappé à cette loi. Jusqu'au début des années 80 et même à une époque plus récente, la conception d'un système sur puce (SOC) n'était accessible qu'aux firmes et sociétés spécialisées a cause de la complexité des circuits et des fonctions à intégrer qui demandent divers efforts et compétences. Par conséquence des coûts élevés et cette technologie est inaccessible au grand public.

Aujourd'hui, l'avènement des dernières générations d'FPGA a permis de mettre la technologie SOC à la portée d'un public nettement plus large. Ceci est particulièrement depuis que les FPGA sont proposés à un prix très faible et raisonnable. Ce prodigieux essor a été rendu possible grâce aux progrès concernant les technologies de fabrication des transistors et les méthodes de conception assistée par ordinateur (CAO). Le rôle des FPGA est d'intégrer des circuits logiques complexes. Ces circuits sont susceptibles d'être reconfigurés (Architecture programmée modifiable) partiellement ou entièrement suivant l'application. A cet effet, ce premier chapitre s'inscrit dans un contexte qui traite les FPGA et leurs positions au sein des autres systèmes digitaux. Nous allons ensuite décrire l'apport de cette technologie programmable sur la commande des moteurs. **[1]**

## 1.2 FACTEURS D'ÉVOLUTION DES CIRCUITS NUMÉRIQUES [1]

Les circuiteries numériques (où la porte logique représente l'unité de base) reposent sur les trois aspects des circuits logiques qui sont : le combinatoire, le séquentiel et l'hybride des deux. Des statistiques qui ont été faites ces dernières années ont montrés qu'au bout de 10ans, le prix d'une porte logique a été divisé par 200 où l'intérêt économique du numérique. A cet effet, le prix d'une porte logique est divisé se réduit de 40% par an. Le diagramme suivant montre ces statistiques d'évolution des couts durant ces dernières années:



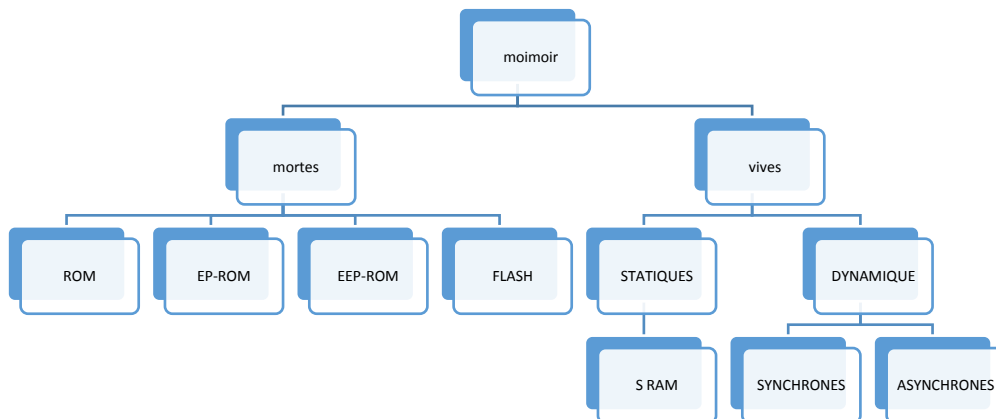
**Figure 1.1.** Diagramme d'évolution des coûts ces dernières années.

Cette évolution est le fruit d'un ensemble de facteurs qui sont récapitulés comme suit :

1. Besoins croissants en circuits spécialisés.
  - Produits de plus en plus complexes.
  - Contraintes (performance, coût...).
2. Evolution rapide de la technologie.
  - *Généralisation des « Systèmes sur Puce ».*
  - Espace de conception trop grand.
3. Evolution des outils de conception "haut-niveau".
  - Produire une architecture à partir d'un algorithme.
  - Exploration automatique de l'espace de conception.
  - Outils d'estimations (performances, surface, etc.).

## 1.3 LES TECHNOLOGIES DE MEMORISATION [2]

Voici d'une manière générale, l'arborescence des mémoires disponibles à nos jours :



**Figure 1.2.** Les différents types de mémoires

L'ensemble des caractéristiques de ces mémoires sont récapitulées comme suit :

- **Les ROM (Read Only Memory):** Mémoires figées par le concepteur à lecture seule et non modifiables.
- **Les P-ROM (Programmable Read Only Memory):** Mémoires programmables une fois  
Par l'utilisateur avec un équipement spécialisé (tableau de fusibles).
- **Les EP-ROM (Erasable Programmable Read Only Memory):** Mémoires Programmables électriquement et effacement par des rayons ultra-violet au bout d'un certain temps (quelques minutes).
- **Les EEP-ROM (Electrically Erasable Programmable Read Only Memory):**  
Mémoires programmables électriquement à lecture seule, effaçables électriquement  
(quelques millisecondes).
- **Les mémoires FLASH:** Elles sont une version plus évoluée des EEP-ROM avec l'avantage d'être plus facile à programmer et à effacer.
- **Les S-RAM (Static Random Memory):** Mémoires volatiles avec cellule de base à plusieurs transistors (accès rapide, consommation plus, coûteux). La volatilité correspond au non-disponibilité de l'information lorsqu'il n'y a pas d'alimentation.
- **Les RAM dynamiques:** Mémoires volatiles qui nécessitent un rafraîchissement

Périodique de l'information afin de la conserver avec cellule de base à un transistor (densité forte, accès lent).

## 1.4 LES CIRCUITS PROGRAMMABLES [3]

La plupart des circuits numériques programmables sont issues de la célèbre Architecture « Architecture Von-Neumann » proposée par John Von Neumann en 1945 et porte son nom. Ensuite une autre architecture qui vient pour compenser les lacunes de la Précédente dans certains domaines et afin d'améliorer la cadence de calcul où le facteur Temps d'exécution est le plus important. Cette architecture nommée « Architecture Harvard » qui porte le nom de l'université américaine qui la propose sachons que parallèlement l'architecture « Von-Neumann » n'est pas figée mais on évolue. Mais le besoin croissant de composants très rapides a orienté les chercheurs à développer une autre solution qui sera complètement différente des deux précédentes architectures. Cette solution réside dans le mode de programmation qui est devenu architectural à logique câblée inversement aux deux premières architectures précédentes où la programmation est séquentielle

D'une manière générale, il existe deux alternatives ou solutions qui sont:

- **Une solution logicielle:** Elle est nommée aussi solutions programmables du type « Processeur » où un traitement séquentiel relativement lent et programmation dépendante du composants (DSP, Microprocesseur et Microcontrôleur...).
- **Une solution matérielle:** Elle est nommée aussi solutions programmables du type « Logique » où un traitement parallèle en temps réel et une programmation conception CAO) indépendante du composant (ASIC et FPGA...).

**Les caractéristiques de ces circuits sont :**

**1. Les circuits du type DSP/Microprocesseurs :** Un rapport performance/coût faible, un temps de conception très court et une grande souplesse d'utilisation.

**2. Les circuits du type spécialisé ASIC :** Très performants mais avec un cycle de conception long et une architecture figée.

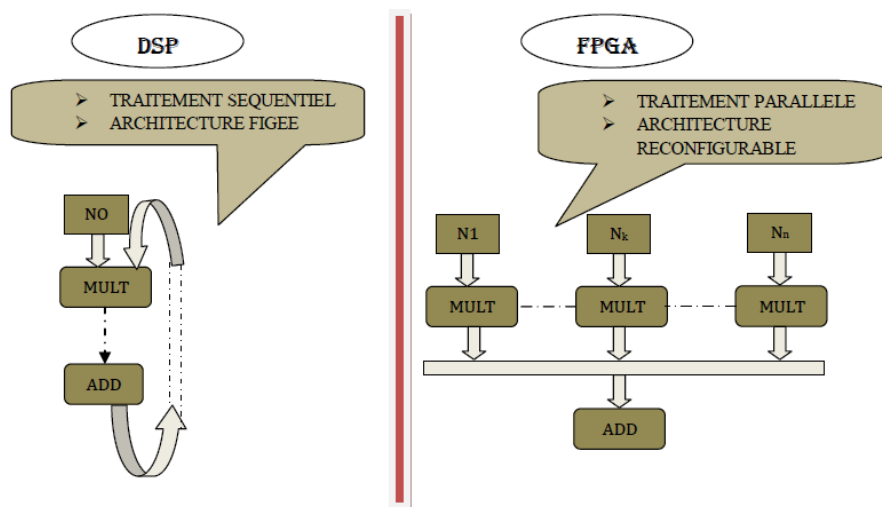
**3. Les circuits du type FPGA:** Des performances proches des ASIC, un coût unitaire intermédiaire et un cycle de conception moyen et une architecture modifiable.

Voici un tableau récapitulatif de comparaison des différentes solutions numériques

-----	ASIC	FPGA	DSP
PERFORMANCES	Très élevées	Elevées	Faible
TAILLE	Faible	moyenne	Elevée
CONSOMMATION	Faible	moyenne	Très élevées
SOUPLESSE	Système sur puce	Système sur puce	Composant supplémentaires
MISE ON OEUVR	complexe	Complexité moyenne	Complexité moyenne
COÛT DE COMPOSANT	Très élevées	Moyen	Faible

**Tableau 1.1.** Comparaison des différentes solutions numériques.

Les fréquences de fonctionnement du mode séquentiel dépassent aujourd'hui 2 GHz, la réduction du temps de cycle ne suffira pas à compenser les insuffisances de ce mode de fonctionnement. La fréquence d'horloge ou de fonctionnement des FPGA est relativement faible devant les microprocesseurs et les DSP et ne dépasse pas quelques centaines de Mégahertz, mais cette faiblesse est largement compensée et même surpassée grâce au parallélisme de traitement. L'exploitation du parallélisme est une technique en pleine expansion dans les circuits numériques FPGA qui sont une alternative des DSP.



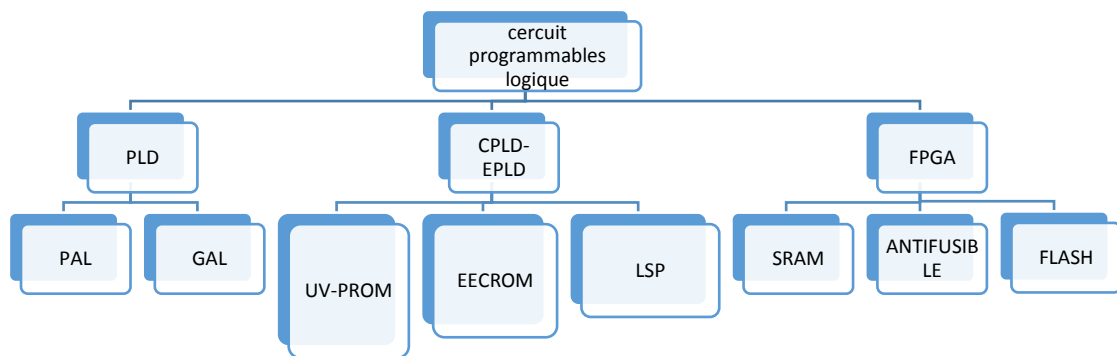
**Figure 1.3.** Schéma comparatif d'un DSP et d'un FPGA.



Les circuits séquentiels ont une architecture matérielle figée et inversement pour les circuits logiques reconfigurables FPGA, on fait une adaptation de l'architecture du composant en fonction des l'algorithme.

## 1.5 LES CIRCUITS LOGIQUES PROGRAMMABLES [2]

Alors qu'auparavant la distinction était nette entre le logiciel et le matériel, le circuit logique programmable FPGA est venu s'introduire comme un hybride entre les deux approches. Les circuits logiques programmables et reprogrammables Architecturalement sont classifiés en trois grandes familles les PLD, CPLD et FPGA. La figure suivant illustre les différents types suivant la technologie utilisée.



**Figure 1.4.** Diagramme des différents types de circuits logiques programmables.

- **Les PLD (Programmable Logic Device):** Famille des circuits programmables qui comprend les PAL, GAL.
- **PAL (Programmable Array Logic):** Circuits logiques programmables dans lesquels seules les fonctions ET sont programmables, les fonctions OU ne le sont pas.
- **GAL (Generic Array Logic):** Circuits logiques PAL reprogrammables à technologie CMOS
- **Les CPLD ou EPLD (Erasable Programmable Logic Device):** Circuits logiques reprogrammables.
- **ISP (In System Programmable):** Circuit que l'on peut programmer même lorsqu'il est en place sur l'application.

- **Les FPGA (Field Programmable Gate Array)** : Ces circuits sont une évolution des CPLD. Récemment, ils intègrent également des mémoires entières, des multiplieurs et même des noyaux de processeur.

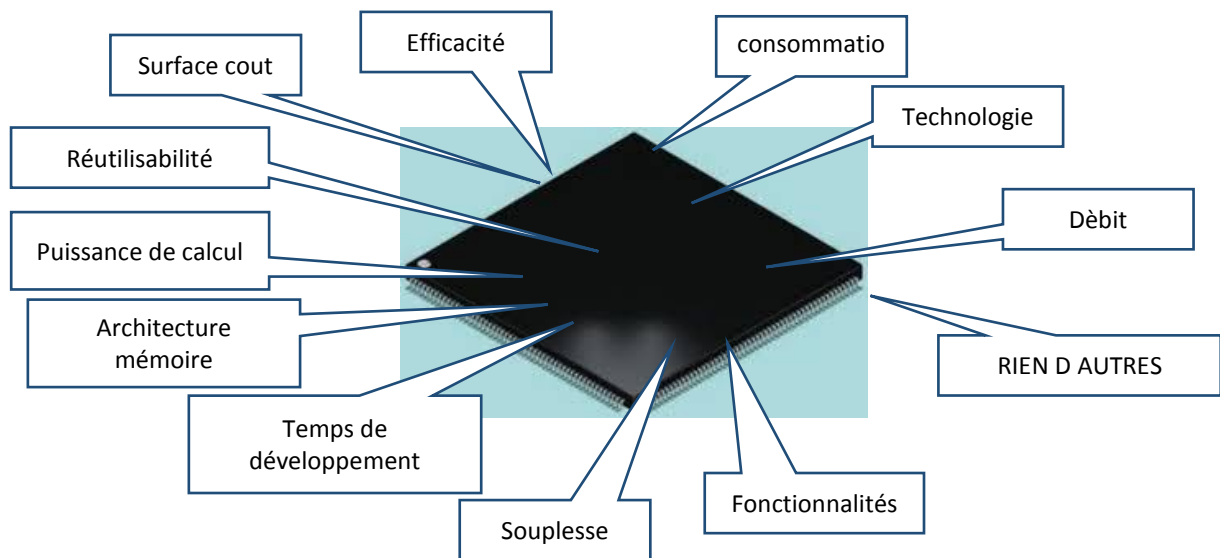
## **1.6 Les Circuits Logiques Programmables du type FPGA**

### **1.6.1 Critères de choix du circuit programmable FPGA [5]**

Les FPGA sont développés récemment grâce aux progrès de la technologie VLSI, l'apparition de ce type de circuits est une révolution des systèmes digitaux et ouvrants des perspectives de traitement numérique inaccessibles auparavant. La fin des années 80 a vu l'apparition des premiers circuits FPGA qui sont des circuits intégrés que l'on peut configurer en un temps relativement court pour réaliser n'importe quelle fonction logique « câblée » à bas coût par une programmation de ses cellules logiques et ses interconnexions avec une restriction de ne pas épuiser les ressources du FPGA. Typiquement, un circuit FPGA haute densité peut contenir jusqu'à plusieurs millions d'éléments programmables. Pour réussir une application à base d' FPGA et afin d'obtenir un système plus performant, consommant un minimum de puissance, il est nécessaire de respecter un certain nombre de règles comme :

- Bien connaître les caractéristiques du FPGA ciblé pour assurer son adéquation avec les besoins du projet.
- Elaborer une méthodologie de conception.
- Maîtriser les outils d'implémentation et de choisir des outils de synthèse de qualité.

La conception sur les circuits FPGA est un challenge dans lequel l'objectif est de trouver le bon compromis entre densité, flexibilité et performances temporelles.



**Figure 1.5.** Critères de choix du circuit logique programmable FPGA.

### 1.6.2 Différent domaines d'application des FPGA [6]

Les FPGA ont fait révolutionner certains domaines de contrôle numérique et de plus en plus utilisés pour intégrer des architectures numériques complexes. Ils sont devenus les plus populaires en matière d'implantation et de prototypage des circuits numériques après leur apparition sur le marché en 1984. La clé maîtresse de leurs réussites est l'aspect de programmation de ces derniers. Leurs utilisations actuelles couvrent les deux domaines : civil et militaire. Parmi ces applications nous citons :

**1-Informatique** : Périphériques spécialisés. **2-Machinerie industrielle** : Contrôleur pour machines. **3-Télécommunications** : Traitement d'images, filtrage. **4-Instrumentation** : équipement médical, prototypage. **5-Transport** : Contrôle d'avions et métros. **Aérospatiale** : Satellites. **Militaire** : Radar, Communication protégée, la détection ou la surveillance. Autres.

### 1.6.3 Principaux fondateurs d'FPGA [7]

Les fabricants des FPGA ne cessent pas d'améliorer leurs produits par l'efficacité et la puissance. L'ensemble des firmes (Principaux fondateurs) qui conçoivent ce type de circuits sont : Actel , Altera , Atmel , Cypress, Lattice, Minc , QuicLogic, Xilinx et d'autres

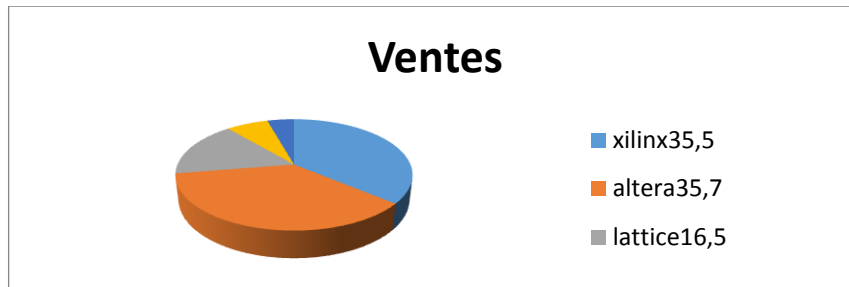


Figure 1.6. Statistiques du marché par les vendeurs d’FPGA.

### 1.6.4 Configuration et reconfiguration des FPGA [8]

Un système reconfigurable est un système qui est constitué de composants ou entités à architecture modifiable afin de répondre à un objectif bien déterminé. Ce système reconfigurable dispose d’un mécanisme permettant de choisir une nouvelle configuration et de la mettre en place dans le cadre du processus de reconfiguration. Les circuits FPGA sont un type de ces circuits reconfigurables. Ils sont programmables ou configurables sur les cartes sur lesquelles ils sont implantés par l’utilisateur. Cette reconfigurabilité est une propriété nécessaire face aux systèmes à charges et contraintes variables. Le FPGA est une abréviation anglaise de qui signifie réseau des portes programmables sur site ce qui est décrit dans la figure suivante.

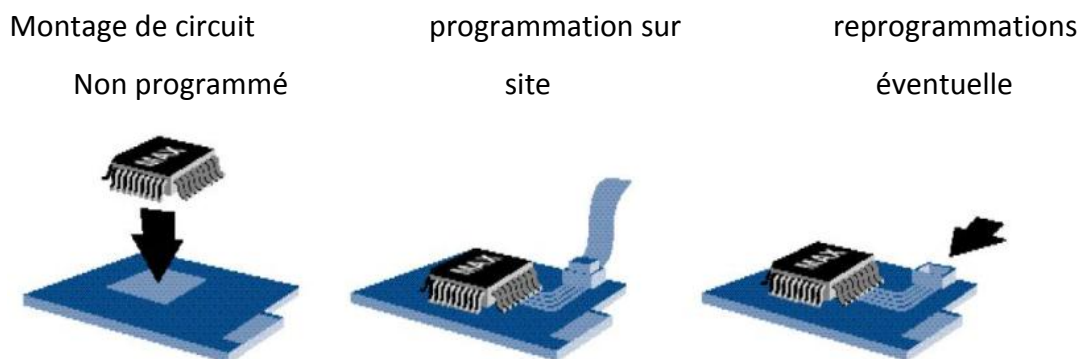


Figure 1 .7. Reprogrammabilité sur site d’un FPGA.

### 1.6.5 Technologies de programmation des FPGA [9]

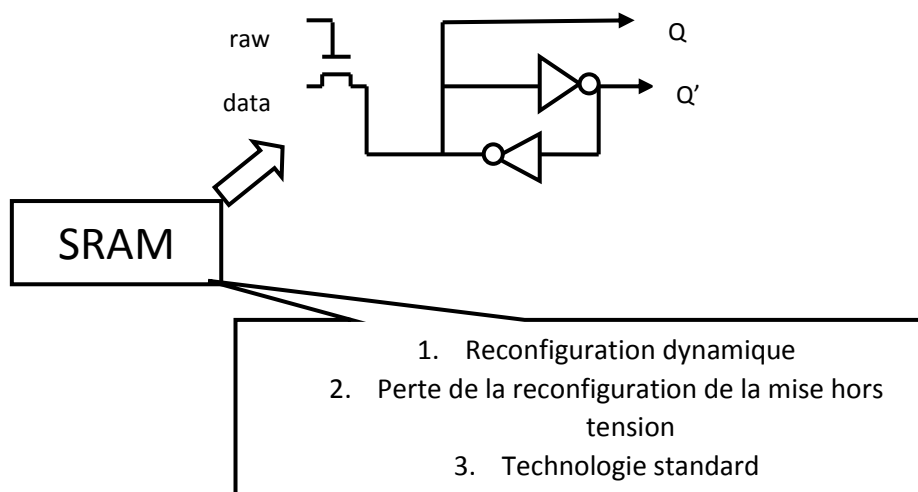
Pour franchir les inconvénients susmentionnés des mémoires, et dans le but de faire un ensemble de technologies complémentaire adaptable suivant l’environnement des cahiers de charges, il existe trois types d’FPGA reprogrammables suivant la technologie de mémorisation pour répondre aux différentes applications.

Ces trois principales technologies d’FPGA sont :

- Technologie de programmation par **RAM**.
- Technologie de programmation par **EEPROM ou FLASH**.
- Technologie de programmation par **ANTI-FUSIBLE**.

### 1.6.5.1 Technologie à base de RAM (XILINX et ALTERA)

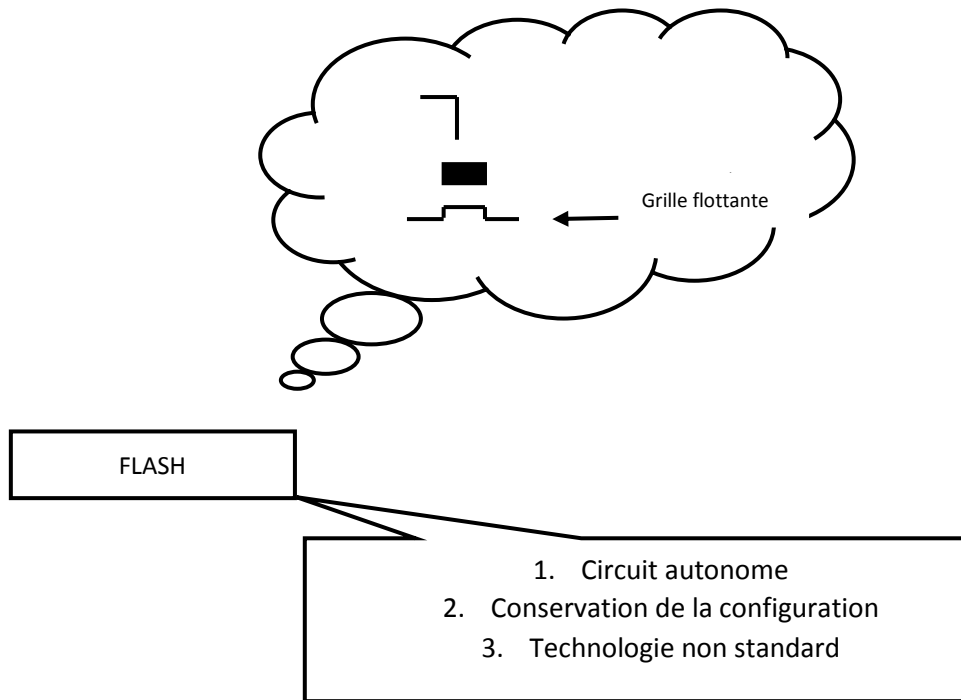
Cette technologie permet d'avoir une reconfiguration rapide des FPGA .Les points de connexions sont des ensembles de transistors commandés. L'inconvénient majeur de cette technologie c'est qu'elle nécessite beaucoup de place et il est nécessaire de sauvegarder le design du FPGA dans une autre mémoire Flash.



**Figure 1.8.** Caractéristiques des technologies SRAM.

### 1.6.5.2 Technologie à base d'EEPROM ou FLASH (LATTICE et ACTEL)

Cette technologie garde sa configuration mais un nombre limité de configuration avec une configuration plus lente par rapport à SRAM.



**Figure 1.9.** Caractéristiques des technologies FLASH.

### 1.6.5.3 Technologie à base d'ANTI-FUSIBLES(ACTEL)

Les points de connexions sont du type ROM, c'est-à-dire que la modification du point est irréversible. Pour comprendre le mécanisme de connexion sans rentrer dans les détails des semi-conducteurs, on considère que le point de connexion est le point de rencontre de deux segments conducteurs ou lignes conductrices. Le non anti-fusible vient du fait que l'état initial du fusible ou la couche isolante est présent et il n'y a pas de contact pour l'établir il faut détruire le fusible ce qui est contradictoire au fonctionnement habituel d'un fusible. Des composants moins génériques mais plus petits et plus rapides ont été développés.

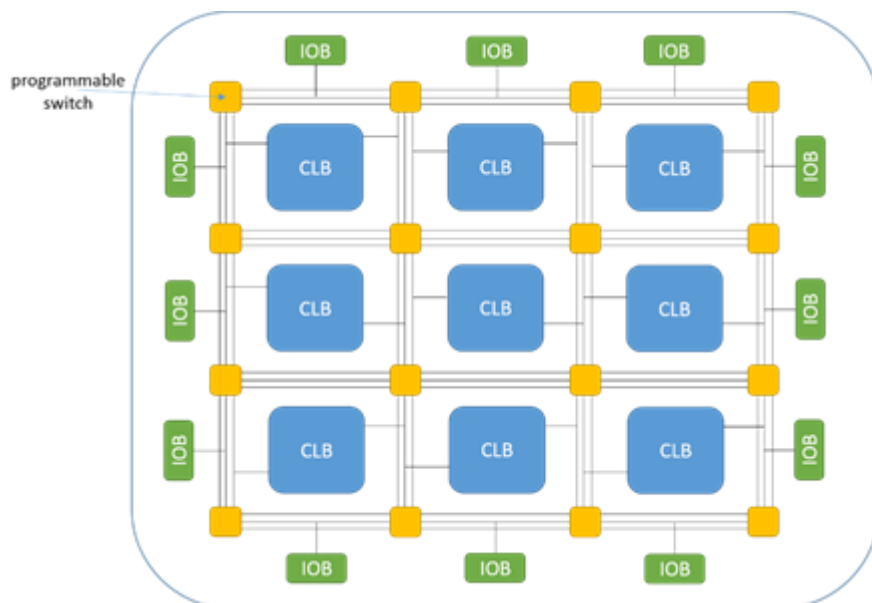
### 1.6.6 Architecture interne des FPGA [10]

On appelle les FPGA quelques fois LCA, abréviation anglaise de « Logic Cell Array » signifiant réseau de cellules logiques. Pour réussir à implanter un système dans un FPGA de manière efficace, il est indispensable de bien connaître sa structure interne et ses limites du point de vue performances. Les composants logiques programmables sont des circuits composés de nombreuses cellules logiques élémentaires librement assemblables. Celles-ci sont connectées de manière définitive ou réversible par

programmation afin de réaliser les ou les fonctions numériques désirées. Un FPGA (Field-Programmable Gate Array) est un circuit intégré avec une structure adaptable par l'utilisateur et composée d'un réseau de cellules élémentaires ou d'éléments logiques programmables CLB et IOB répartis régulièrement et reliés entre eux grâce à des connections qui forment une matrice de routage programmable pour obtenir un comportement spécialisé du circuit dans sa globalité. Puisque tous les éléments sont programmables, le FPGA peut émuler et réaliser n'importe quel circuit à l'unique condition que celui-ci n'épuise pas les ressources logiques et de routage du FPGA.

L'ensemble des systèmes reconfigurables FPGA est subdivisé en trois catégories suivant les fonctions préexistantes et des possibilités de les interconnectées. Ces catégories sont : des systèmes reconfigurables nommés "**grain fin**", des systèmes reconfigurables nommés "**grain moyen**" et des systèmes reconfigurables nommés "**grain large**".

L'architecture interne des FPGA est différente d'un fondeur à un autre et même entre les différentes gammes du même constructeur mais rien n'empêche que leurs ressemblances peuvent être rassemblées dans le schéma représentatif de la figure suivante :



**Figure1.10.** Architecture interne d'un FPGA.

- ❖ Les macro-cellules internes sont appelées :
  - Soit **CLB** qui est la dénomination adoptée par Xilinx et abréviation anglaise de «**C**onfigurable **L**ogic **B**lock », signifiant bloc logique configurable.
  - Soit LC qui est le nom choisi par Cyprès et abréviation anglaise de « **L**ogic **C**ell », signifiant cellule logique.
  - Soit LE qui c'est l'appellation d'Altéra abréviation anglaise de « **L**ogic **E**lement» signifiant élément logique.
- ❖ Les macro-cellules sur la périphérie sont appelées : **IOB** abréviation anglaise de «**I**nput **O**utput **B**lock », signifiant bloc logique d'entrées sorties.
- ❖ L'ensemble des points de connexion est appelé PIP, abréviation anglaise de« **P**rogramme **I**nterconnect **P**oints ».

La granularité des FPGA par les macro-cellules CLB nous permet d'implémenter des fonctions logiques « combinatoires ou séquentielles » complexes car chaque CLB est constitué d'une partie combinatoire et d'une partie séquentielle. Chaque fonction est décomposée en petites fonctions booliennes qui peuvent être contenues par de petites cellules élémentaires SLICES. Ces dernières comportent des LUT pour la partie combinatoire et une ou des bascules (généralement de type D) pour la partie séquentielle.

### 1.6.7 Architecture MULTI-COMPOSANTS [11]

Dans un environnement multi-composant, plusieurs structures sont possibles pour un certain nombre d'FPGA qui admettent l'association et comme exemple :

#### 1.6.7.1 Association de plusieurs FPGA

L'association de plusieurs FPGA prend différentes architectures suivant le besoin voici deux exemples d'association

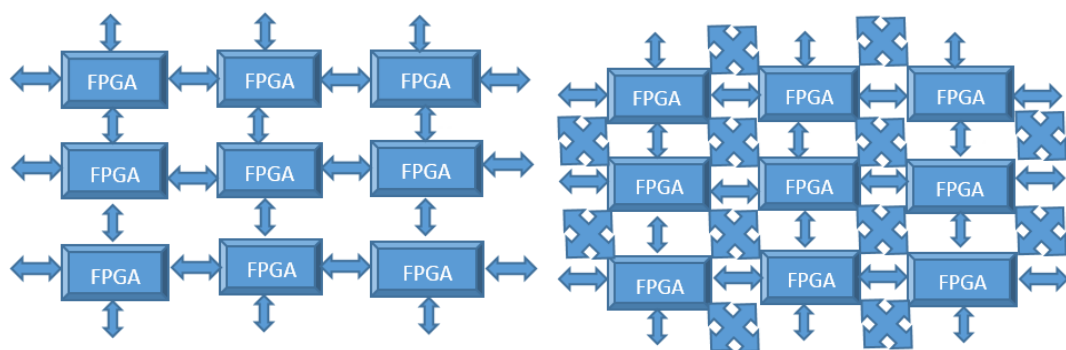


Figure 1.11. Exemples d'association entre FPGA.



### 1.6.7.2 Association d'un microprocesseur et d'un FPGA

La combinaison entre un **FPGA** et un processeur est possible dans certains cas comme accélérateurs matériels où le système reconfigurable est directement couplé à un processeur ce qui constitue un SOC. Le microcontrôleur ou le microprocesseur est généralement le hôte (organise, initialise, charge les programmes.....).

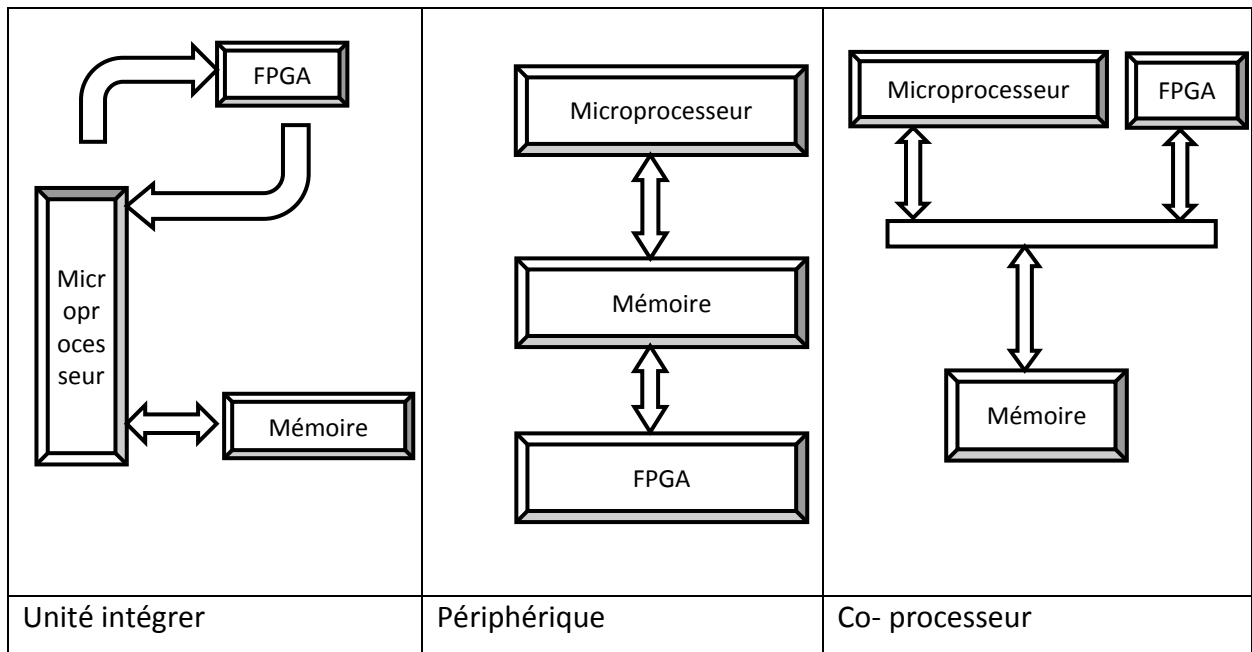


Figure 1.12. Exemples d'association entre **FPGA** et **Microprocesseur**.

### 1.6.8 Avantages et inconvénients des FPGA [12]

Les avantages et les inconvénients des FPGA sont multiples on trouve :

Avantage	Inconvénient
<ul style="list-style-type: none"> <li>-Technologie « facile » a maitriser.</li> <li>-Temps de développement réduit</li> <li>-Reprogrammable</li> <li>-Cout peu élevé</li> <li>-Parallélisme de traitement</li> <li>-Flexibilité et la possibilité de réduire fortement les délais de développement et commercialisation</li> <li>-La reconfiguration, parfois en temps réel</li> </ul>	<p>Temps de réponse long par rapport aux ASIC</p>

## 1.6.9 Les deux grandes familles architecturales d’FPGA

Les familles des FPGA peuvent se regrouper en deux groupes :

### 1.6.9.1 Les circuits FPGA à base de « LUT » (Look Up Tables)

Les LUT (Look Up Tables) ressemblent aux tables de vérité des fonctions logiques et réalisables par des mémoires de type SRAM. Aujourd'hui, la structure la plus utilisée est basée sur ce type (Look-Up Table) d’FPGA. Les possibilités offertes par les circuits programmables FPGA à SRAM permettent par ailleurs de mettre en œuvre le concept de prototypage (ou maquette) pour la vérification fonctionnelle de systèmes sur puce pour certaines applications. La fonction de la LUT est de stocker la table de vérité de la fonction combinatoire à implémenter comme le montre la figure suivante.

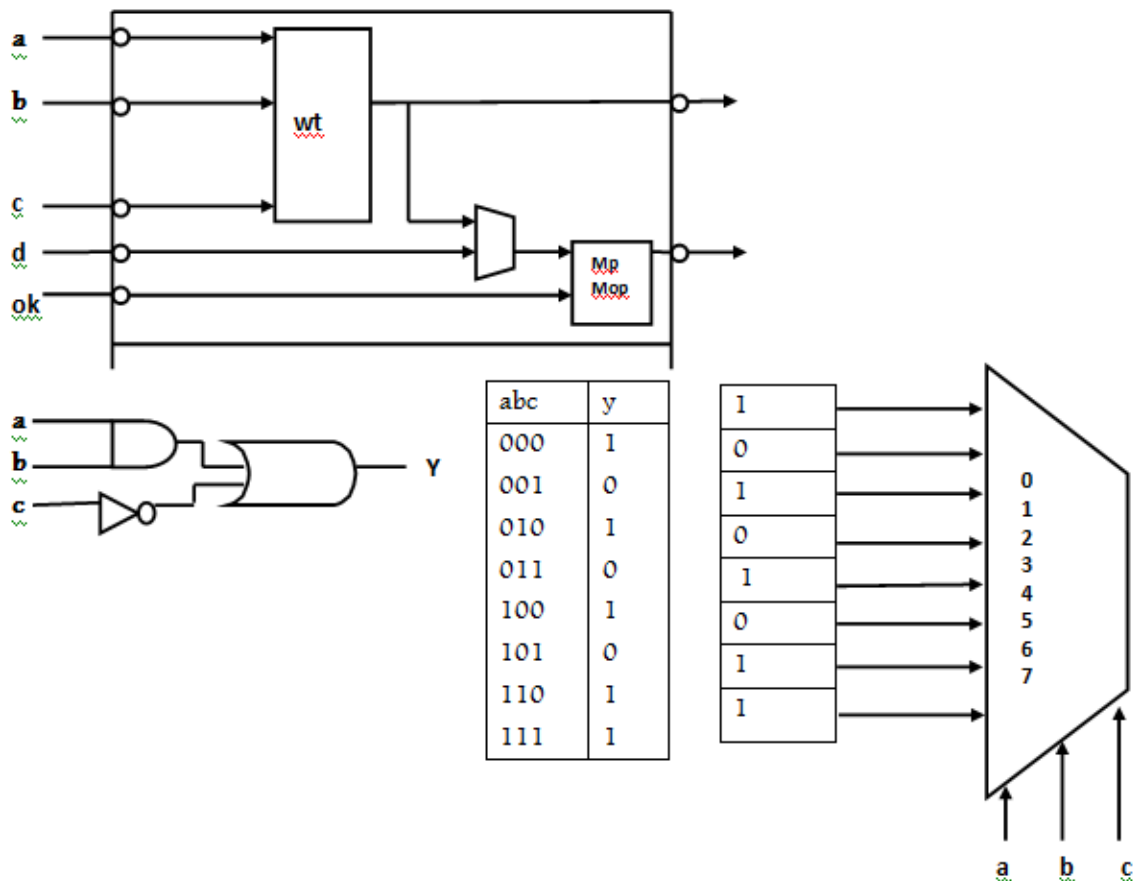
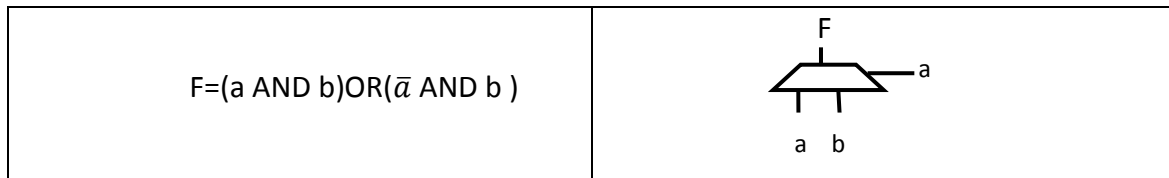


Figure 1.13. Exemple d’implémentation sur LUT.

### 1.6.9.2 Les circuits FPGA à base de multiplexeurs « MUX »

Les FPGA à base de multiplexeurs qui sont des microcellules à trois entrées capable de réaliser la fonction suivante :



**Figure 1.14.** Exemple d'implémentation sur des multiplexeurs.

## 1.7 LA CONFIGURATION DES FPGA PAR LES OUTILS CAO

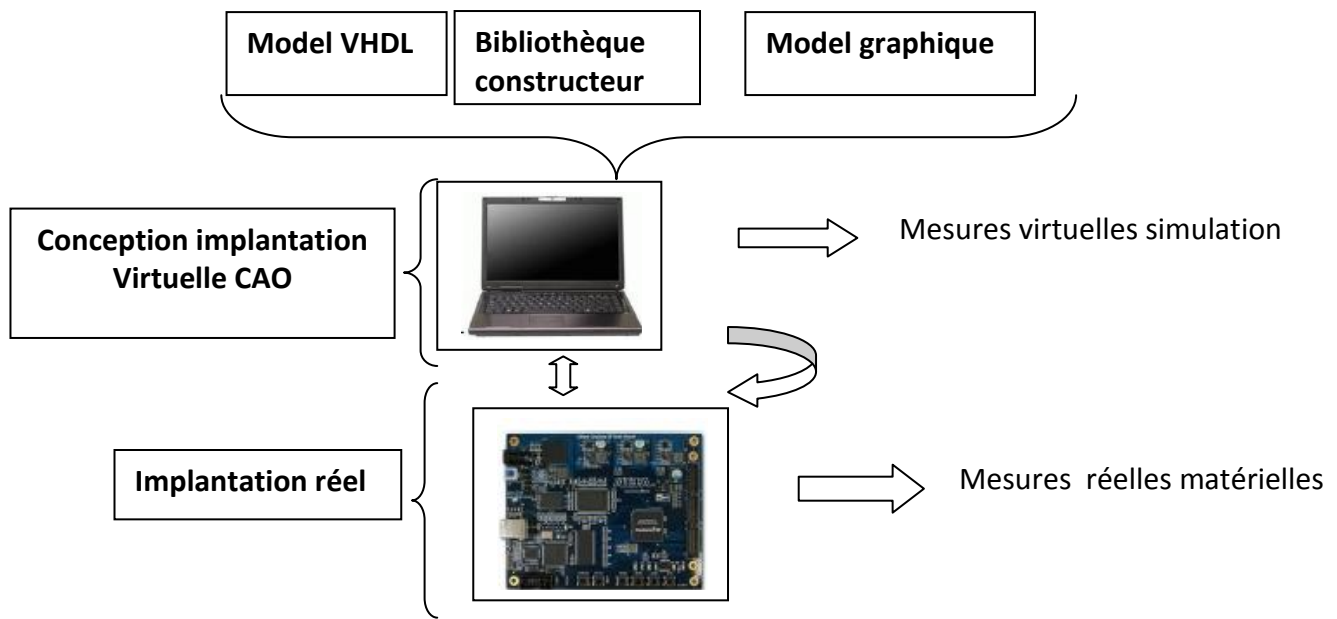
### 1.7.1 De l'algorithmique à la conception CAO [14]

L'origine du mot algorithme provient du nom latinisé d'Al-Khawarizmi (Abou Jafar Mohammed Ibn Musa al-Khawarizmi médecin arabe du moyen âge). Un algorithme est une suite ou séquence de raisonnements réalisés par un nombre fini d'opérations en termes de temps et de support matériel afin de fournir une solution à certains problèmes. Un algorithme est présenté sous forme d'une prescription qui peut avoir différentes formes (textuelle, graphique, formule mathématique, diagramme de séquence.....etc.). La plupart des algorithmes existants sont orientés vers une implémentation logicielle ce qui est contraire à notre application qui s'agit d'une implantation matérielle. L'objectif des algorithmes sont récapitulés comme suit:

- Transmettre un savoir faire.
- Décrire les étapes à suivre pour réaliser un travail.
- Expliciter clairement les idées.

Les techniques de conception CAO (**C**onception **A**ssistée par **O**rdinateur) sont aujourd'hui très éprouvées et largement employées afin de concevoir des circuits électroniques nécessaires

à mettre en pratique les connaissances algorithmiques .L'approche moderne pour la conception des circuits (logiques) électriques et la manière d'introduire une fonctionnalité sur un support physique sont confiées aux outils CAO. Les outils CAO sont utilisés pour générer le fichier de configuration des FPGA qui s'appelle (bit-Stream) à partir d'une description de haut niveau. Les principaux rôles confiés aux outils CAO sont : la description, la simulation, la synthèse, le placement et le routage. Un design peut être conçu à l'aide d'un éditeur schématique ou d'un outil de traitement de textes.



**Figure 1.15.** Mode d'exécution matériel de la CAO.

La conception de circuit simple peut se faire par l'approche schématique mais dans les circuits complexes elle cède le champ de conception à l'approche textuelle.

### 1.7.2 Méthodologie de conception [15]

Le flot de conception d'un système sur puce regroupe plusieurs niveaux d'abstraction. Dans chaque niveau, le concepteur s'intéresse à la résolution d'un problème. Les outils de

CAO sont utilisés intensivement et assurent la transition entre les différents niveaux d'abstraction. Nous pouvons traiter un système complexe de deux manières qui sont :

- L'approche dite « descendante » (ou « **top-down** » en anglais).
- L'approche dite « ascendante » (ou « **bottom-up** » en anglais).

**REMARQUE:** Il y a une grande similitude de conception pour les FPGA, les CPLD et les ASIC.

Le développement d'une application sur FPGA par des outils CAO suit l'enchaînement des étapes suivantes:

- **Spécification du design**
  - ❖ Le nombre de broches d'entrée-sortie et leur localisation dans la puce FPGA.
  - ❖ La spécification de la fréquence d'horloge du système.
  - ❖ La spécification de la mémoire requise pour l'application.

- **Développement du design**
- ❖ Spécification de la méthodologie de design (Outil de développement utilisé).
- ❖ La saisie du circuit Codage RTL (VHDL, Verilog...)
  - Graphique (Machine à états).
  - Saisie HDL (Hardware Description Language).
- ❖ La simulation (Pré et Post synthèse).
- **Synthèse**

La synthèse est le processus qui convertit la représentation du design à partir du code HDL fourni pour produire une représentation au niveau porte logique. Elle s'occupe de déterminer quelles sont les structures susceptibles de répondre au cahier des charges étudié et de produire un code booléen unique sous forme d'un fichier.

- **Placement et routage**

A partir des fichiers de synthèse, l'outil de conception procède au placement et routage. Un algorithme de routage est sensé de faire l'aiguillage des données qu'il reçoit vers leurs destination par action sur les nœuds de routage ce qui est équivalent a définir les chemins qui relient l'ensemble des **CLB** contenus dans la fonction désirée. Ces algorithmes de routage sont différents d'un concepteur à un autre. Plusieurs traitements sont nécessaires pour obtenir un fichier de configuration :

- **Partitionnement** : Les équations logiques sont partitionnées en un autre ensemble équivalent d'équations. Chaque équation de ce nouvel ensemble peut être implantée dans un seul bloc logique du composant cible **FPGA**.
- **Placement** : Des blocs logiques sont sélectionnés dans la matrice et affectés au calcul des nœuds du réseau booléen.
- **Routage** : Les ressources d'interconnexion sont affectées à la communication de l'état des nœuds du réseau vers les différents blocs logiques qui en ont besoin.
- **Génération des données numériques de configuration** : Les informations abstraites de routage, de placement et les équations implantées dans les blocs sont transformées en un ensemble de valeurs numériques, qui seront chargées sur le composant FPGA.

- **Intégration et implémentation**

L'implémentation est la réalisation proprement dite qui consiste à mettre en œuvre l'algorithme sur l'architecture du circuit configurable cible, c'est-à-dire à compiler, charger, puis lancer l'exécution sur un ordinateur ou calculateur. C'est une étape de programmation physique et de tests électriques qui clôture la réalisation du circuit. La figure suivante résume un peu l'ensemble de ces étapes.

## **1.8 Conclusion**

Ce premier chapitre est dédié à la présentation du contexte général de thèse et permet d'en déterminer les principaux objectifs. Au départ, nous avons présenté un survol des circuits programmables puis nous avons étudié l'état d'art des FPGA ce qui nous a permis de conclure que la technologie FPGA s'inscrit au sommet de l'évolution des composants logique et le besoin croissant de composants plus performants, plus économiques et disponibles en grandes quantités est les grands axes du progrès qui sont disponibles dans les FPGA. Ces derniers ouvrent de grandes perspectives en matière de contrôle en temps réel. La réalisation d'un système de contrôle en temps réel nécessite aussi une bonne maîtrise des outils fournis par la théorie de l'automatique lors de la phase de modélisation et de simulation, ainsi qu'une bonne maîtrise de l'informatique temps réel lors de la phase d'implantation.

## Chapitre 2 : Modélisation des convertisseurs DC/DC

---

### 2.1 Introduction [16]

L'utilisation des convertisseurs de puissance pour différentes applications devient de plus en plus importante. Les domaines sont principalement les applications domestiques (téléphone mobile, ordinateurs, électroménager), l'industrie automobile (avec l'apparition de véhicules hybrides et électriques), l'aéronautique (l'avion électrique), le domaine ferroviaire, les énergies renouvelables (panneaux solaires photovoltaïques, éoliennes), les réseaux de transport d'énergie électrique (les liaisons à courant continu), etc.

Les hacheurs sont des convertisseurs statiques continu-continu, permettant de générer une source de tension continue variable à partir d'une source de tension continue fixe. Ils se composent de condensateurs, d'inductance et d'interrupteurs statiques. Ils consomment moins de puissance. C'est pour cette raison que les hacheurs ont de très bons rendements.

### 2.2 Convertisseurs DC/DC (hacheurs)

Le convertisseur DC-DC permet d'alimenter une charge sous une tension continue réglable à partir d'une source continue fixe.

On obtient une tension de valeur moyenne variable en établissant et interrompant périodiquement l'alimentation de la charge par la source grâce à des interrupteurs.

### 2.3 Différents types d'hacheurs [17]

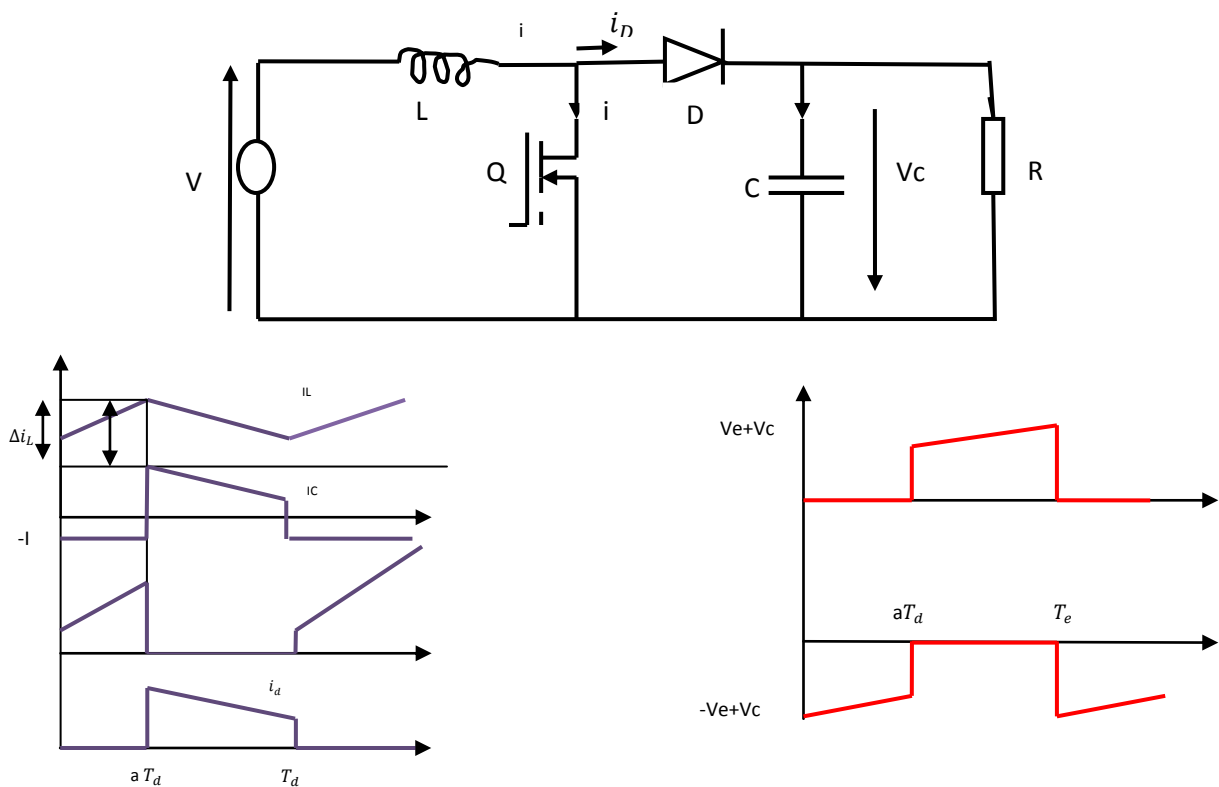
Selon la position du commutateur et du hacheur, différents types de convertisseurs de tension peuvent être réalisés, nous citons à titre d'exemple:

- ✓ L'abaisseur de tension « Buck »
- ✓ L'élévateur de tension « Boost »
- ✓ L'abaisseur élévateur de tension « Buck-Boost »
- ✓ Le Cuk.

Dans ce qui suit, nous abordons la modélisation des ces quatre convertisseur.

### 2.3.1 Convertisseur Boost

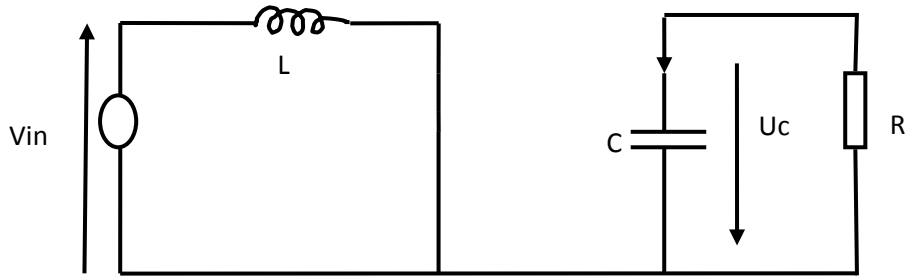
On considère un convertisseur DC-DC de type Boost. Il est composé essentiellement d'un interrupteur  $K$  (comme IGBT ou MOSFET) et d'une diode  $D$ . L'interrupteur  $K$  est commandé par un signal à modulation de largeur d'impulsion (MLI) de période de découpage fixe  $T_d$  et de rapport cyclique variable  $D$ . La conduction des deux interrupteurs est complémentaire, quand  $K$  est fermé  $D$  est ouvert ; et quand  $K$  est ouvert,  $D$  est fermé. Durant chaque période,  $K$  est fermé de l'instant 0 à  $\alpha T_d$  et ouvert de  $\alpha T_d$  à  $T_d$ . On distingue deux modes de fonctionnement selon que le courant dans l'inductance s'annule (conduction discontinue) ou pas (conduction continue). On s'intéresse au second cas qui est le plus important. La figure 2.4 donne le schéma de principe de ce convertisseur, les intervalles de conduction de l'interrupteur et de la diode ainsi que les formes d'ondes de ses courants (à gauche) et tensions (à droite).



**Figure 2.1.** Formes d'ondes des courants et tensions du convertisseur Boost

Lorsque l'interrupteur  $K$  est fermé ( $0 < t < aT_e$ ), le circuit du convertisseur devient (Figure 2.2) :





**Figure 2.2.** Circuit équivalent du Boost l'interrupteur fermé.

$$L \frac{di_1}{dt} = V_e \quad (2.1)$$

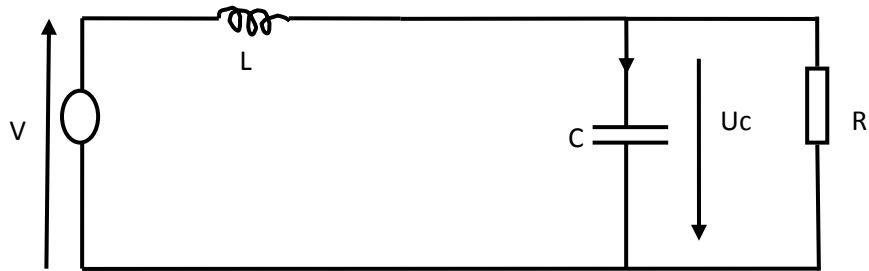
$$i_1 = I_{mn} + \frac{V_e}{L} t \quad (2.2)$$

Avec  $I_{min}$  la valeur minimale du courant dans l'inductance.

A l'instant  $t = aT_d$ , le courant dans l'inductance atteint sa valeur maximale  $I_{max}$

$$I_{max} = I_{min} + \frac{V_e}{L} aT_d \quad (2.3)$$

Lorsque l'interrupteur  $K$  est ouvert ( $aT_d < t < T_d$ ), le circuit du convertisseur sera comme montré dans la Figure 2.3.



**Figure 2.3.** Circuit équivalent du Boost l'interrupteur ouvert .

$$L \frac{di_L}{dt} = V_e - V_s \quad (2.4)$$

$$i_L = I_{max} + \frac{V_e - V_s}{L} (t - a T_d) \quad (2.5)$$

A l'instant  $t = T_d$ , le courant dans l'inductance revient à sa valeur minimale  $I_{min}$

$$i_{min} = I_{max} + \frac{V_e - V_s}{L} (t - a T_d) \quad (2.6)$$

Soit  $\Delta i_{L=i_{\max}-i_{\min}}$  l'ondulation du courant dans l'inductance.

En faisant l'égalité des valeurs de l'ondulation du courant tirées des équations (2.3) et (2.6), on peut déduire la valeur moyenne de la tension de sortie  $V_s$  :

$$V_s = \frac{1}{1-a} V_e \quad (2.7)$$

On remarque qu'on peut contrôler la tension de sortie du convertisseur en faisant varier sa tension d'entrée ou son rapport cyclique. Celui-ci étant toujours compris entre 0 et 1, alors le montage fonctionne en élévateur de tension.

En appliquant le principe de conservation de puissance entre l'entrée et la sortie du convertisseur, on peut établir la valeur moyenne du courant dans l'inductance en fonction du courant moyen dans la charge et du rapport cyclique :

$$I_L = \frac{1}{1-a} I_s \quad (2.8)$$

De l'expression (2.3), on exprime l'ondulation de courant dans l'inductance :

$$\Delta i_L = \frac{a V_e}{L} T_d = \frac{a V_e}{L f} \quad (2.9)$$

Avec  $f$  la fréquence de découpage.

Pendant la première séquence de fonctionnement ( $0 < t < a T_d$ ), seul le condensateur donne de l'énergie à la charge, on peut écrire :

$$C \frac{dv_s}{dt} = -i_s \quad (2.10)$$

Alors l'ondulation de la tension de sortie peut être exprimée par:

$$\Delta V_s = \frac{a I_s}{C f} T_d = \frac{a V_e}{(1-a) R C f} \quad (2.11)$$

D'après l'analyse des deux séquences de fonctionnement, on constate que les contraintes sur l'interrupteur commandé et la diode sont les mêmes en courant et en tension

$$i_{k,\max} = i_{d,\max} = I_{\max} = I_L + \frac{\Delta i_L}{2} = \frac{I_s}{1-a} + \frac{\Delta i_L}{2} \quad (2.12)$$

$$V_{k,\max} = |V_{d,\max}| = V_{s,\max} = V_s + \frac{\Delta V_s}{2} = \frac{V_e}{1-a} + \frac{\Delta V_s}{2} \quad (2.13)$$

### 2.3.2 Convertisseur Buck

On procède de la même manière dans l'étude du convertisseur Buck. La figure 2.4 montre la structure et les formes d'ondes du courant et de la tension de ce convertisseur qui est caractérisé par la disposition en série de l'interrupteur avec la source.

La tension de sortie en valeur moyenne, l'ondulation de courant dans l'inductance et l'ondulation de la tension de sortie sont donnés respectivement par:

$$V_s = a \cdot V_e \quad (2.14)$$

$$\Delta i_L = \frac{a(1-a) \cdot V_e}{L \cdot f} \quad (2.15)$$

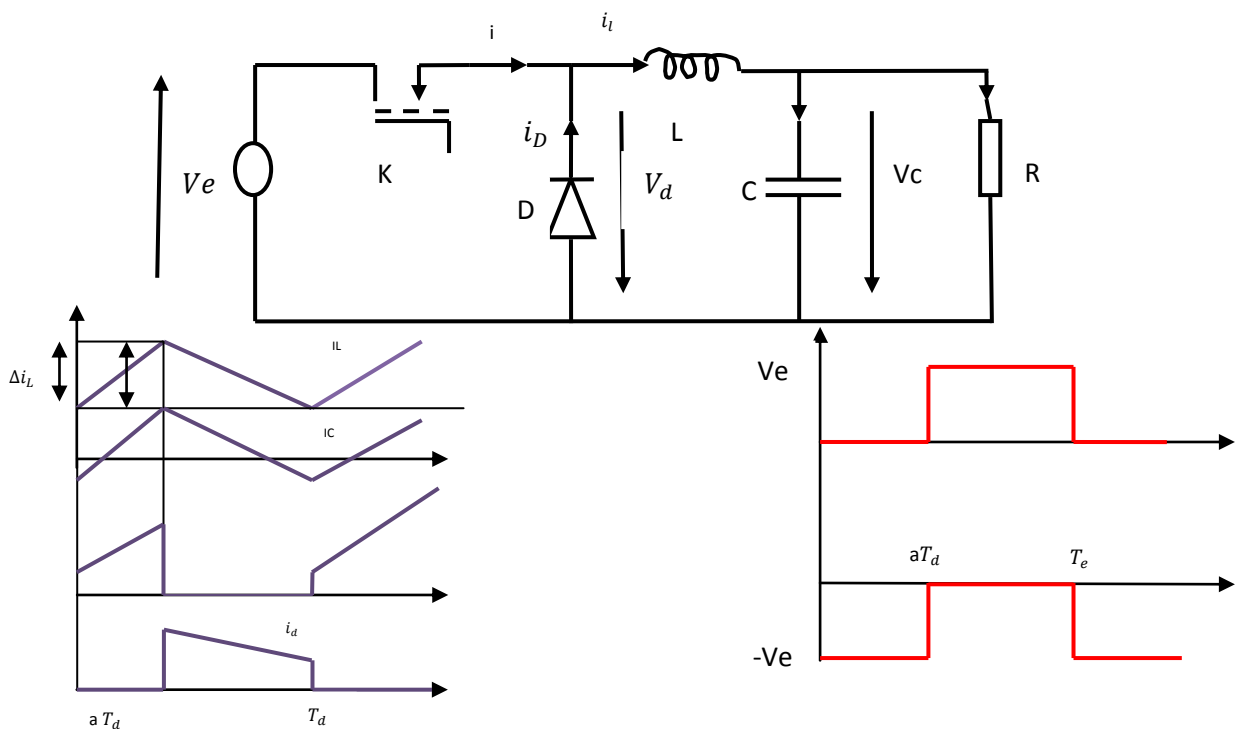
$$\Delta V_s = \frac{\Delta i_L}{8Cf} = \frac{a(1-a)V_e}{8CLf^2} \quad (2.16)$$

Le convertisseur est donc abaisseur en tension.

Le courant moyen traversant l'inductance est égal au courant moyen dans la charge :

$$I_L = I_s \quad (2.17)$$

Les contraintes sur l'interrupteur commandé et la diode sont les mêmes.



**Figure 2.4.** Formes d'ondes des courants et tensions du convertisseur Buck.

En tension, on a :

$$V_{k,max} = |V_{d,max}| = V_e \quad (2.18)$$

Et en courant :

$$i_{k,max} = i_{d,max} = I_{L,max} = I_L + \frac{\Delta i_L}{2} \quad (2.19)$$

### Calcul de C:

Le condensateur de sortie C sert à filtrer la tension de sortie. Plus la valeur de C est grande, plus le signal de sortie n'est lissé, mais d'autre part, le courant absorbé atteint des pics plus considérables. Les paramètres dimensionnant d'un condensateur sont sa capacité et la tension max qu'il peut supporter. La valeur du condensateur nous sera donnée à partir de valeur crête-à-crête du signal Vs en régime permanent dont il dépend

$$\text{On a } i_c(t) = i_L(t) - I_s \quad (2.20)$$

$$\Delta V_s = V_2 - V_1 = \frac{1}{C} \int_{t_1}^{t_2} \delta I_L(t) dt \quad (2.21)$$

$$\text{Donc } \Delta V_s = \frac{1}{C} A \quad (2.23) \quad \text{et } A = \frac{1}{2} \times I_{cmax} \times (t_2 - t_1) \quad (2.22)$$

$$\text{On calcule : } t_2 - t_1 \text{ et } I_{cmax}, I_{cmin} \quad (2.24)$$

• Pendant la Première phase de conduction ( $0 < t < aT$ ):

$$t = 0 \rightarrow \begin{cases} i_L = 0 \\ I_s = \frac{1}{2} I_{Lmax} (a - a') \end{cases} \rightarrow I_{cmin} = -\frac{I_{Lmax}}{2} (a - a') \quad (2.25)$$

$$t = aT \rightarrow \begin{cases} i_L = I_{Lmax} \\ I_s = \frac{1}{2} I_{Lmax} (a - a') \end{cases} \rightarrow I_{cmax} = \frac{I_{Lmax}}{2} (2 - (a - a')) \quad (2.26)$$

On peut écrire l'équation du courant de C :

$$i_c = \frac{I_{cmax} - I_{cmin}}{aT} t + I_{cmin} \quad (2.27)$$

$$i_c = 0 \rightarrow \frac{I_{Lmax}}{aT} t + \frac{I_{Lmax}}{2} (a - a') = 0 \quad (2.28)$$

$$t = t_1 = \frac{aT}{2} (a - a') \quad (2.29)$$

• Pendant la deuxième phase de conduction ( $aT < t < (a - a')T$ ):

$$t = aT \rightarrow \begin{cases} i_L = I_{Lmax} \\ I_s = \frac{1}{2} I_{Lmax} (a - a') \end{cases} \rightarrow I_{cmax} = \frac{I_{Lmax}}{2} (2 - (a - a')) \quad (2.30)$$

$$t = (a - a')T \rightarrow \begin{cases} i_L = 0 \\ I_s = \frac{1}{2} I_{Lmax} (a - a') \end{cases} \rightarrow I_{cmin} = -\frac{I_{Lmax}}{2} (a - a') \quad (2.31)$$

On peut écrire l'équation du courant dans C :

$$i_c = \frac{I_{cmin} - I_{cmax}}{a'T} (t - aT) + I_{cmin} \quad (2.32)$$

$$i_c = -\frac{I_{Lmax}}{a'T} (t - aT) + \frac{I_{Lmax}}{2} (2 - (a + a')); a'T = (a' + a)T - aT \quad (2.33)$$

$$i_c = -\frac{I_{Lmax}}{a'T} (t - aT) + \frac{I_{Lmax}}{2} (2 - (a + a')) = 0 \quad (2.34)$$

$$t = t_2 = (a + a') \left(1 - \frac{a}{2}\right) T \quad (2.35)$$

On calcule :  $t_1 - t_2$  :

$$(t_2 - t_1) = (a + a') \left(1 - \frac{a}{2} - \frac{a'}{2}\right) T = \frac{(2 - (a + a'))}{2} (a + a') T \quad (2.36)$$

$$A = \frac{1}{2} \times I_{cmax} \times (t_2 - t_1) = \frac{1}{2} \frac{I_{Lmax}}{2} (2 - (a + a')) \frac{(2 - (a + a'))}{2} (a + a') T \quad (2.37)$$

$$I_{Lmax} = \frac{V_s a' T}{L} \quad \text{et} \quad V_s = \frac{a V_e}{(a + a')} \quad I_{Lmax} = \frac{a' a V_e T}{(a + a') L} \quad (2.38)$$

$$\text{Donc :} \quad A = \frac{a' a V_e (2 - (a + a'))^2}{8 L f^2} \quad (2.39)$$

$$\text{Alors :} \quad \Delta V_s = \frac{1}{C} A = \frac{a' a V_e (2 - (a + a'))^2}{8 L f^2} \quad (2.40)$$

$$C = \frac{a' a V_e (2 - (a + a'))^2}{8 L \Delta V_s f^2} \quad (2.41)$$

### Calcule l'inductance L :

$$\text{On a :} \quad I_{Lmax} = \frac{V_s a' T}{L} \quad (2.41)$$

$$\text{Alors :} \quad I_{Lmin} \geq \frac{V_s a' T}{L} \quad (2.42)$$

### 2.3.3 Convertisseur Buck-Boost

La topologie et les formes d'ondes de courant et de tension obtenues de ce convertisseur sont présentées à la figure 2.5. La tension de sortie moyenne est donnée par:

$$V_s = \frac{a V_e}{1 - a} \quad (2.43)$$

L'ondulation de courant dans l'inductance est donnée par :

$$\Delta I_L = \frac{a V_e}{L f} \quad (2.44)$$

et l'ondulation de la tension de sortie:

$$\Delta V_s = \frac{a I_s}{C.f} = \frac{a^2 V_e}{(1-a)RCf} \quad (2.45)$$

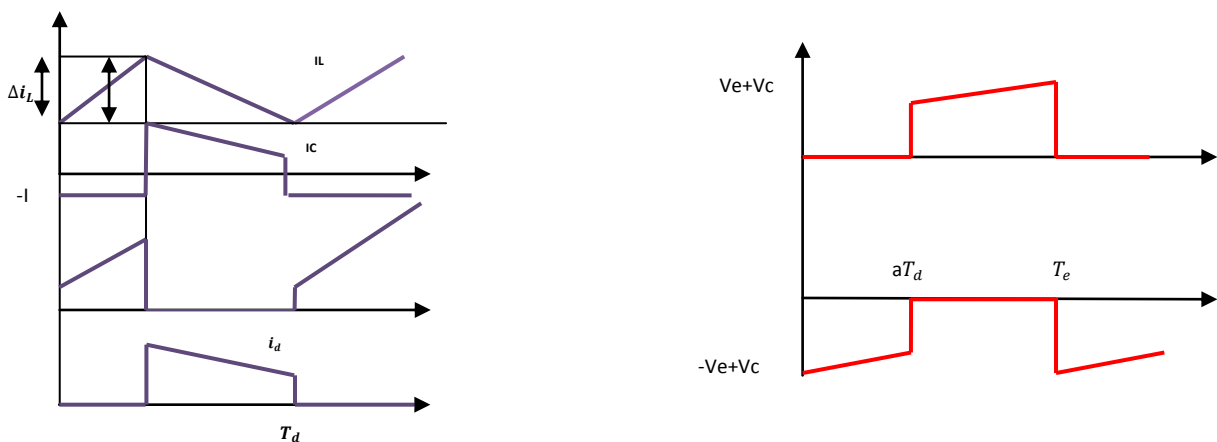
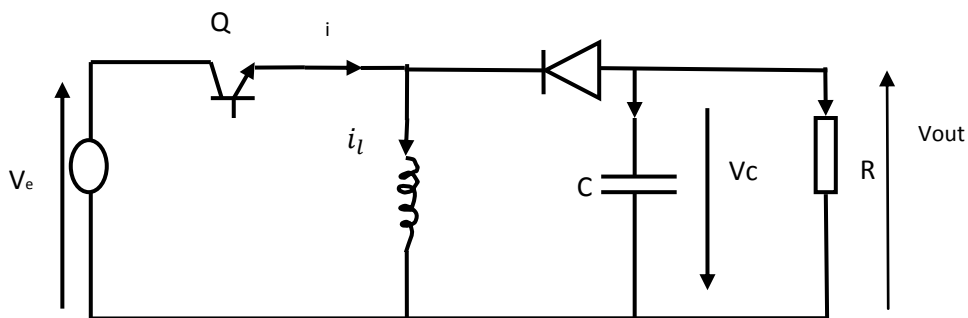
La tension de sortie du convertisseur Buck-Boost est négative par rapport à la tension d'entrée. Son amplitude peut être supérieure ou inférieure à celle de la tension d'entrée selon la valeur du rapport cyclique. C'est un abaisseur-élevateur-inverseur en tension. Le courant moyen traversant l'inductance est donné par :

$$I_L = \frac{I_s}{1-a} \quad (2.46)$$

Les contraintes en tension et en courant sur l'interrupteur commandé et la diode sont les mêmes :

$$V_{k,max} = |V_{d,max}| = V_{s,max} + V_e = \frac{V_e}{1-a} + \frac{\Delta V_s}{2} \quad (2.47)$$

$$i_{k,max} = i_{d,max} = I_{L,max} = \frac{I_s}{1-a} + \frac{\Delta i_L}{2} \quad (2.48)$$



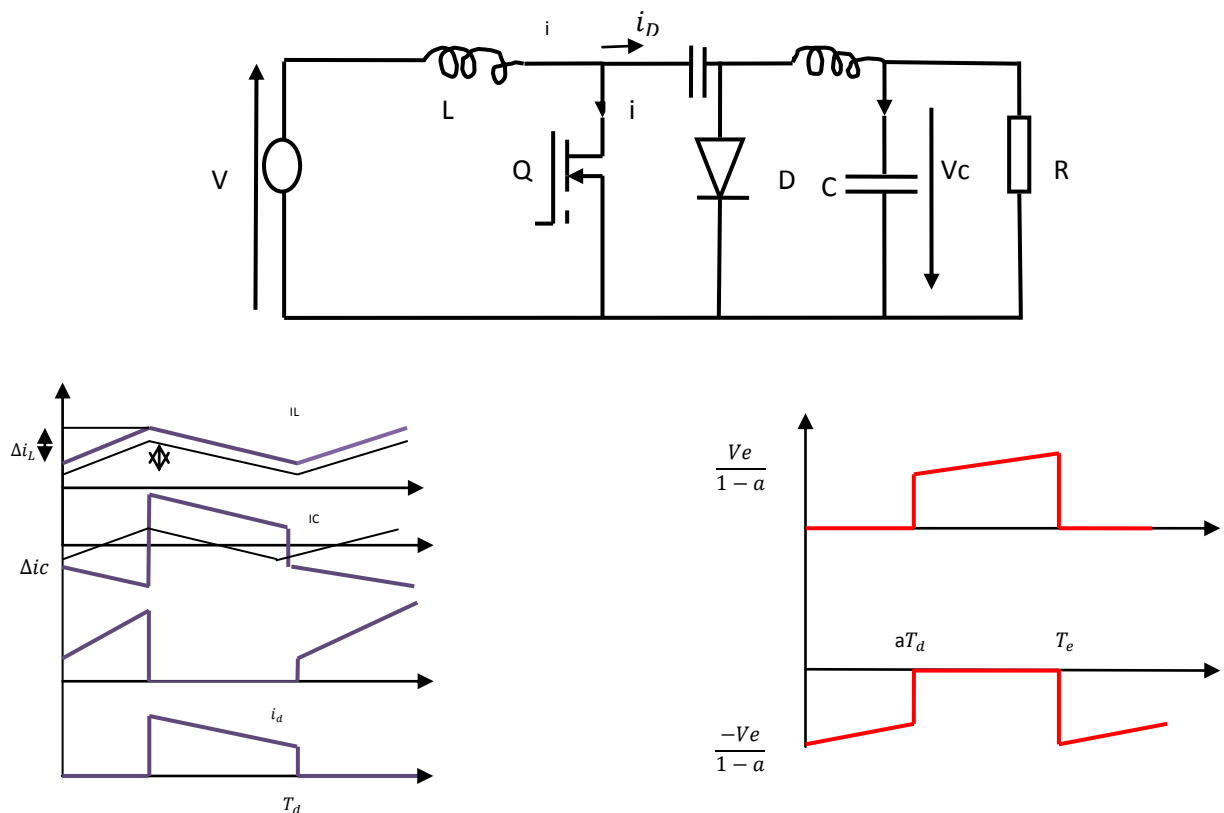
**Figure 2.5.** Formes d'ondes des courants et tensions du convertisseur Buck-Boost.

### 2.3.4 Convertisseur Cuk

La figure 2.6 illustre le circuit électrique du convertisseur Cuk et ses formes d'ondes de courant et de tension. La tension de sortie moyenne est donnée par:

$$V_s = \frac{aV_e}{1-a} \quad (2.49)$$

On remarque que le rapport de transformation est similaire à celui du Buck-Boost. De même, on peut dire que c'est un montage abaisseur-élevateur-inverseur en tension.



**Figure 2.6.** Formes d'ondes des courants et tensions du convertisseur Cuk.

L'expression reliant les courants moyens est :

$$I_L = \left(\frac{a}{1-a}\right) I_L = \left(\frac{a}{1-a}\right) I_s \quad (2.50)$$

Les ondulations de courant dans les deux inductances :

$$\Delta i_L = \frac{aV_e}{L_f} \quad (2.51)$$

$$\Delta i_{L'} = \frac{aV_e}{L'f} \quad (2.52)$$

L'ondulation de la tension de sortie:

$$\Delta V_s = \frac{\Delta i_{L'}}{8C.f} = \frac{a V_e}{8L' Cf^2} \quad (2.53)$$

L'ondulation de la tension aux bornes de  $C'$

$$\Delta V_{c'} = \frac{(1-a)I_L}{C'.f} = \frac{a^2 V_e}{(1-a)RCf} \quad (2.54)$$

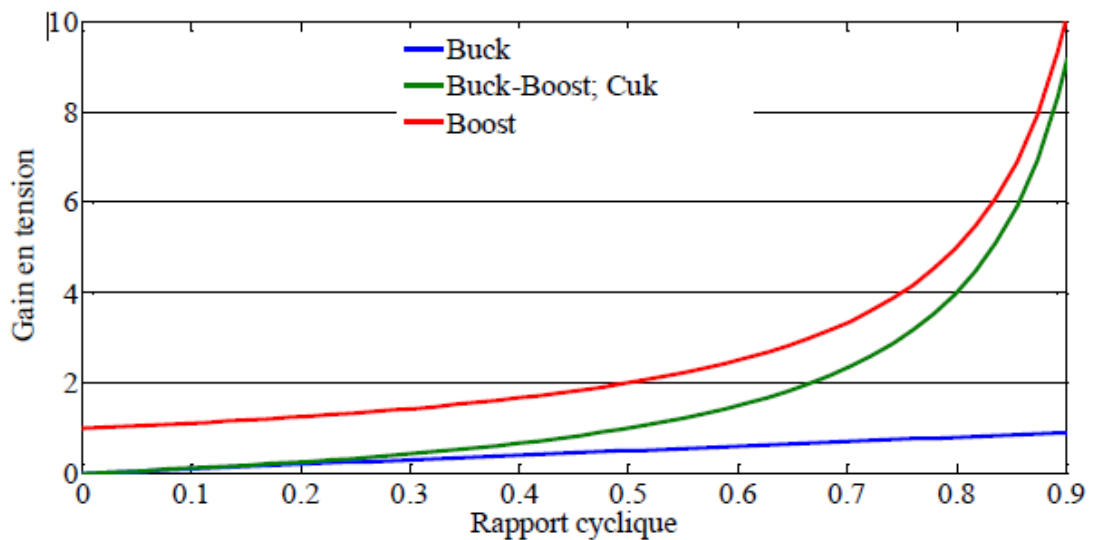
Les contraintes sur l'interrupteur commandé et la diode sont :

$$V_{k,max} = |V_{d,max}| = V_{c',max} = \frac{V_e}{1-a} + \frac{\Delta V_{c'}}{2} \quad (2.55)$$

$$i_{k,max} = i_{d,max} = I_L - I_{L'} = \frac{\Delta i_L + \Delta i_{L'}}{2} \quad (2.56)$$

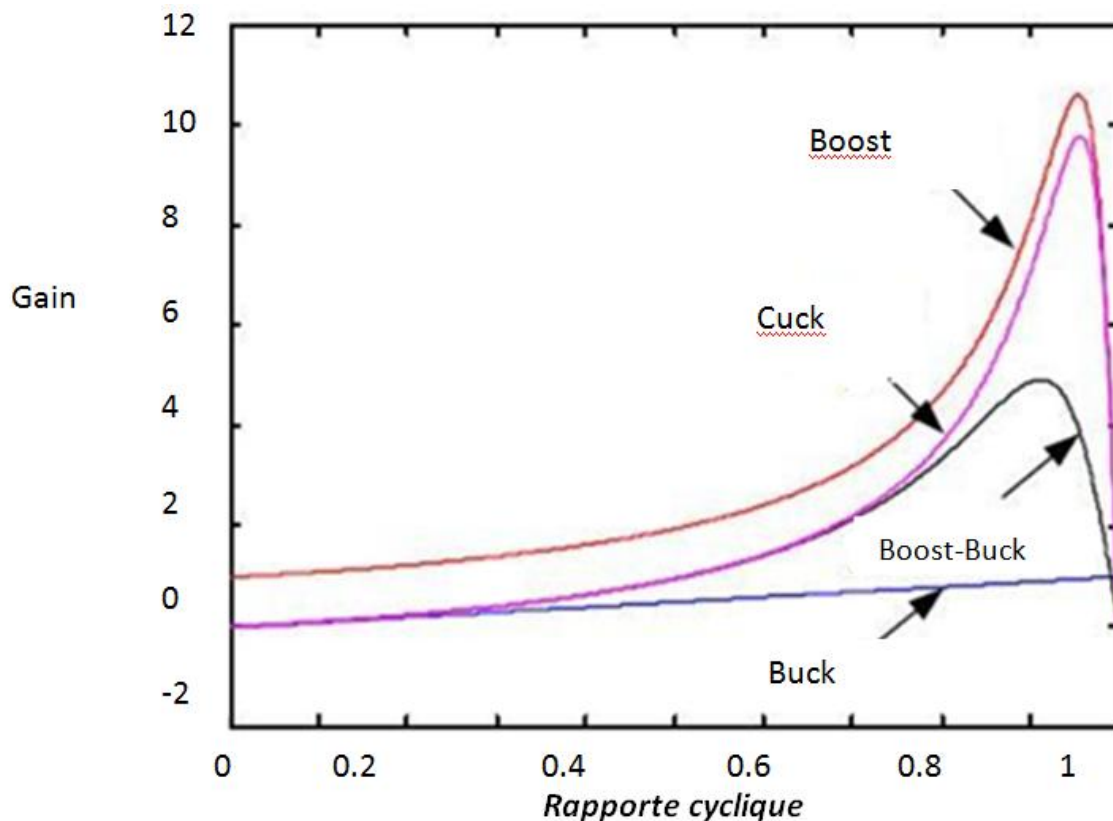
## 2.4 Récapitulatif d'analyse et choix du convertisseur

. Tous les convertisseurs cités précédemment excepté le Buck, peuvent jouer le rôle d'élevateur de tension. D'ailleurs, les structures Buck-Boost, Cuk sont élévatrices pour un rapport cyclique supérieur à un demi. Si on regarde à la complexité des structures, le Boost est le plus simple. Les contraintes en tension et en courant sur les interrupteurs et les diodes sont presque similaires pour les différentes topologies élévatrices.



**Figure 2.7.** Evolution du gain en tension en fonction du rapport cyclique des convertisseurs





**Figure 2.8.** Evolution du gain en tension en tenant compte des éléments parasites des convertisseurs

Si on les compare du point de vue gain en tension, le Boost prend la première place comme le montre la figure 2.7 qui donne l'évolution du gain en tension en fonction du rapport cyclique.

Enfin, une comparaison de ces convertisseurs est donnée par le Tableau 2.1 qui récapitule l'analyse faite auparavant sur les différents convertisseurs. On peut remarquer aussi que la diode D présente dans le convertisseur Boost peut jouer le rôle de protection du GPV ce qui permet de dispenser de la diode anti-retour et de faire des économies par rapport aux autres topologies. Pour mieux différencier entre les convertisseurs, HUANG a tracé le gain en tension en fonction du rapport cyclique (Figure 2.8) en tenant compte des éléments parasites des convertisseurs

**Tableau 2.1.** Récapitulatif des caractéristiques des convertisseurs continu-continu

Paramètres convertisseurs	Gain en tension $\frac{V_s}{V_e}$	Contraintes en tension $V_{k,max} =  V_{d,max} $	Contraintes en courant $i_{k,max} =  i_{d,max} $
Boost	$\frac{1}{1-a}$	$\frac{V_e}{1-a} + \frac{\Delta V_s}{2}$	$\frac{I_s}{1-a} + \frac{\Delta i_L}{2}$
Cuk	$\frac{a}{1-a}$	$\frac{V_e}{1-a} + \frac{\Delta V_{c'}}{2}$	$I_L + I_{L'} + \frac{\Delta i_L}{2}$
Buck- Boost	$\frac{a}{1-a}$	$\frac{V_e}{1-a} + \frac{\Delta V_s}{2}$	$\frac{I_s}{1-a} + \frac{\Delta i_L}{2}$
Buck	$a$	$V_e$	$I_L \frac{\Delta i_L}{2}$

## 2.5 Conclusion

Dans ce chapitre nous avons présenté les caractéristiques électriques des convertisseurs statiques (convertisseur DC-DC). Nous avons montré que la valeur moyenne de la tension de sortie peut être ajustée en jouant sur la valeur du rapport cyclique.

# Chapitre 3 : commande des convertisseurs DC/DC par Mode Glissant

---

## 3.1. Introduction

La commande par mode glissant est un contrôle de type non linéaire. Elle a été introduite initialement pour le contrôle des systèmes à structure variable (SSV). Elle est caractérisée par la discontinuité de la commande aux passages par une surface de commutation appelée : surface de glissement. L'avantage principal de la commande à structure variable avec le mode glissant est la robustesse vis-à-vis du changement des paramètres ou des perturbations. De plus, la commande par mode glissant est relativement facile à mettre en œuvre par rapport à d'autres types de commandes non linéaires. Ces propriétés font que cette loi de commande soit adaptée à de nombreuses applications industrielles, comme dans les domaines de l'automobile ou de l'aéronautique [18].

## 3.2. Système à structure variable

Un SSV est le système dont la structure change pendant son fonctionnement. Il est caractérisé par le choix d'une fonction et d'une logique de commutation. Ce choix permet au système de commuter d'une structure à une autre à tout instant. Dans la commande des SSV par mode de glissement, la trajectoire d'état est amenée vers une surface, puis à l'aide de la loi de commutation, elle est obligée de rester au voisinage de cette surface de glissement. Des contrôleurs à structure variable ont fait leur apparition dans la littérature soviétique [20, 21].

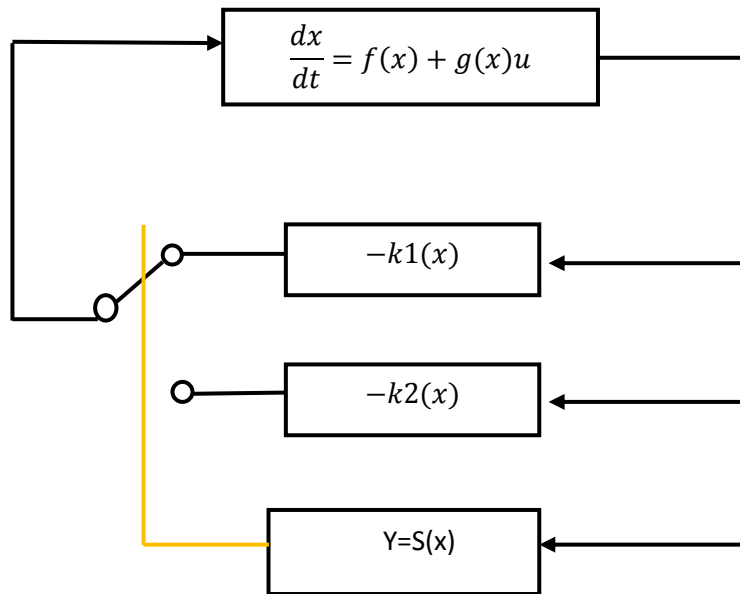
Dans les SSV, on peut trouver deux configurations de base différentes. La première change la structure par commutation entre deux retours d'état différents (Figure 3.1). La deuxième configuration appelée : « configuration par commutation au niveau de l'organe de commande » est la plus simple. Elle modifie la structure du système par simple commutation d'interrupteurs Figure 3.2, ce qui est le cas de tous les convertisseurs statiques [22].

Pour la première configuration, la commande  $u$  est donnée par :

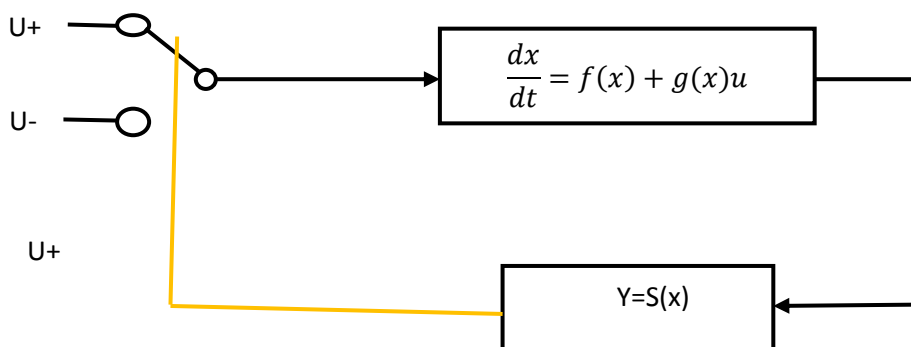
$$u = \begin{cases} -k_1(x) & \text{si } S(x) > 0 \\ -k_2(x) & \text{si } S(x) < 0 \end{cases} \quad (3.1)$$

En mode glissant, le système évolue sur la surface de glissement, par conséquent  $S(x)=0$ . La seconde configuration nécessite un organe de commande qui possède une action à deux positions avec une commutation rapide d'une position à l'autre. La commutation entre ces deux valeurs est imposée par la loi de commutation selon :

$$u = \begin{cases} u^+ & \text{si } S(x) > 0 \\ u^- & \text{si } S(x) < 0 \end{cases} \quad (3.2)$$



**Figure 3.1.** Configuration par commutation au niveau de la contre-réaction d'état.



**Figure 3.2.** Configuration par commutation au niveau de l'organe de Commande.

### 3.3 Commande par mode glissant

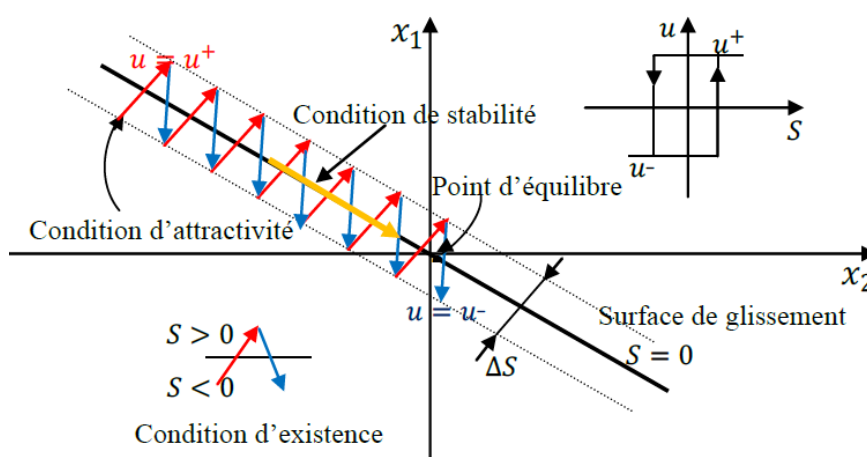
#### 3.3.1 Principe

La commande par mode glissant est une commande robuste basée sur le concept de changement de structure du contrôleur avec l'état du système afin d'obtenir la réponse

désirée [21]. Le contrôleur par mode glissant se base sur l'hypothèse d'une hystérésis nulle sur la surface de glissement  $S(x, t) = 0$  et donc sur une fréquence de commutation variable et théoriquement infinie. Il est clair que, du point de vue pratique, il n'est pas possible de vérifier cette hypothèse. En raison des limitations technologiques liées à l'utilisation de fréquences de commutation élevées, il est préférable de limiter cette fréquence.

L'idée est de diviser l'espace d'état par une frontière de décision appelée : « surface de glissement ». Cette surface délimite deux sous-espaces correspondant à deux états possibles de l'organe de commande (Figure 3.3). La stabilisation sur la surface de glissement est obtenue à l'aide d'une commutation à chaque franchissement de la frontière de décision.

Ce principe de commande repose donc essentiellement sur l'utilisation d'une commande discontinue afin de maintenir l'évolution du système sur une surface de glissement judicieusement choisie. La synthèse doit donc viser à rendre la surface de glissement attractive (condition d'attractivité) depuis tout point de l'espace d'état. Une fois la surface atteinte, il faut assurer le glissement le long de cette surface (condition de glissement) et la stabilité du système (condition de stabilité). En d'autres termes, il faut trouver la condition pour laquelle la dynamique du système glisse sur la surface vers le point d'équilibre désiré (Figure 3.3). Sur la surface, la dynamique du système est indépendante de celle du processus initial, ce qui implique que ce type de contrôle entre dans le domaine des commandes robustes. Ces notions de stabilité sont démontrées en tenant compte du principe de stabilité suivant le critère de LYAPUNOV (théorème 1).



**Figure 3.3.** Principe de la commande par mode glissant.

Pour chaque interrupteur, la commande est effectuée par un comparateur à hystérésis. Le sens de l'hystérésis doit être choisi de manière à ce que le changement d'état ramène la trajectoire à l'intérieur de la fourchette. Ainsi, on peut être amené à choisir :

$$u = \begin{cases} 1 & \text{si } S(x) > +\frac{1}{2}\Delta S \\ 0 & \text{si } S(x) < -\frac{1}{2}\Delta S \end{cases} \quad (3.3)$$

### Théorème 1 :

Soit  $V(x)$  une fonction différentiable de  $R^n$  dans  $R^n$ , dite fonction de LYAPUNOV, qui satisfait les conditions suivantes [23] :

$$\begin{cases} V(0) = 0 \\ V(x) > 0 \quad \forall x \neq 0 \\ \dot{V}(x) \leq 0 \quad \forall x \neq 0 \end{cases} \quad (3.4)$$

Si ces trois conditions sont satisfaites,  $x = 0$  est un point d'équilibre stable. Si la dernière condition devient  $\dot{V}(x) < 0$  pour  $x \neq 0$ , le point  $x = 0$  est asymptotiquement stable.

Dans le cas de la commande par mode glissant, cette fonction de LYAPUNOV est déduite à l'aide d'une pseudo-sortie qui est la surface de glissement  $S(x, t) = 0$ .

### 3.3.2. Objectif de la commande par mode glissant

L'objectif de la commande par mode glissant se résume en deux points essentiels

- Synthétiser une surface  $S(x, t)$  telle que toutes les trajectoires du système obéissent à un comportement désiré de poursuite, régulation et stabilité.
- Déterminer une loi de commande (commutation)  $u(x, t)$  qui est capable d'attirer toutes les trajectoires d'état vers la surface de glissement et les maintenir sur cette surface.

### 3.3.3. Choix de la surface de glissement

La surface ( $S(x)$ ) représente le comportement dynamique désiré du système. SLOTINE propose une forme d'équation générale pour déterminer la surface de glissement qui assure la convergence d'une variable vers sa valeur désirée [24]:

$$S(x) = \left( \frac{\partial}{\partial t} + \lambda_x \right)^{r-1} e(x) \quad (3.5)$$

Ou :

$e(x)$ : Écart de la variable à régler,  $e(x) = x_{ref} - x$

$\lambda_x$  : Constante positive qui interprète la bande passante du contrôle désiré.

$r$  : Degré relatif, égal au nombre de fois qu'il faut dériver la sortie pour faire apparaître la commande.

$S(x) = 0$  : est une équation différentielle linéaire dont l'unique solution est  $e(x) = 0$ .

### 3.3.4. Condition d'existence du glissement

La condition d'existence du régime glissant  $S(x, t) = 0$  se traduit par [25, 19] :

$$\lim_{s \rightarrow 0} S \cdot \dot{S} < 0 \quad (3.6)$$

Ces conditions sont déduites du théorème (1) en appliquant le critère de stabilité de LYAPUNOV dans un voisinage de la surface de glissement et en prenant  $V(x) = \frac{s^2}{2}$  comme fonction candidate de LYAPUNOV. Dans ce cas, la dérivée de la fonction de LYAPUNOV  $\dot{V}$  est égale à  $S \cdot \dot{S}$ .

Les conditions de LYAPUNOV énoncées dans le théorème 1 sont vérifiées si  $S$  et  $\dot{S}$  sont de signes opposés. A noter que ces dernières conditions deviennent des conditions suffisantes pour assurer l'attractivité de la surface si elles sont valables sur tout l'espace d'état et non seulement dans une région proche de la surface de glissement.

## 3.4 Commande de Convertisseur DC-DC Buck par Mode Glissant [26]

La commande par mode de glissement est basée sur la théorie des systèmes à structure variable [3] - [4], elle présente plusieurs avantages à savoir une bonne réponse dynamique et une forte robustesse due aux grandes variations de la charge et de la tension d'entrée. La commande par mode de glissement fonctionne d'une manière simple, elle est décrite comme suit: une surface de glissement est définie avec le point d'équilibre, le système est forcé d'être maintenu dans la surface de glissement (condition d'existence) et enfin le système doit atteindre le point d'équilibre (stabilité).

### 3.4.1. Modélisation du système

La figure (3.4) montre le diagramme schématique de la commande par mode de glissement d'un convertisseur BUCK. Cette partie illustre les aspects théoriques de cette commande. La méthode pour déterminer les coefficients de la surface glissement est également introduite. Dans la figure (3.4)  $C, L$  et  $RL$  sont respectivement la capacitance, l'inductance et la résistance de charge;  $V_{ref}, V_i, \beta, V_o$  et  $I_L$  sont respectivement la tension de référence, la

tension d'entrée, la tension de sortie et le courant dans l'inductance. Avec  $U = 1$  ou  $0$  est l'état de commutation de l'interrupteur de puissance.

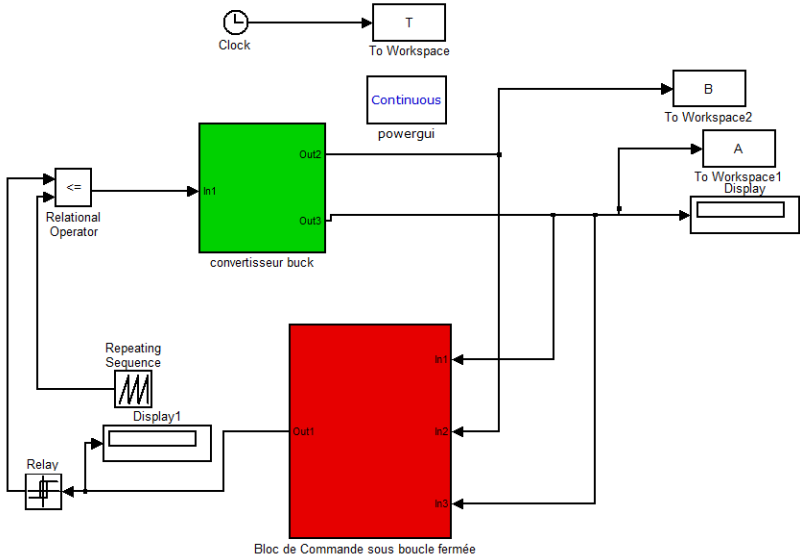


Figure 3.4. Circuit de Commande du Convertisseur DC-DC Buck par Mode Glissant

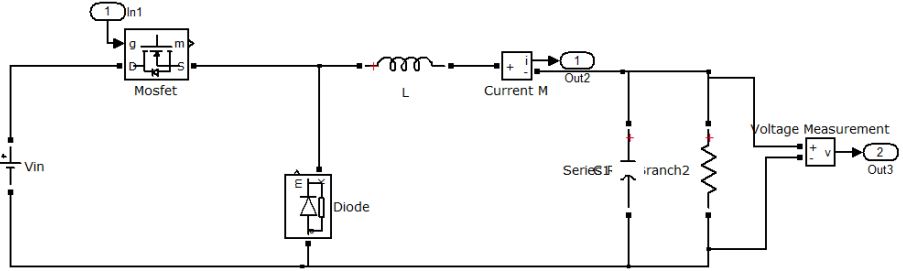


Figure 3.5. Représentation schématique d'un convertisseur Buck.

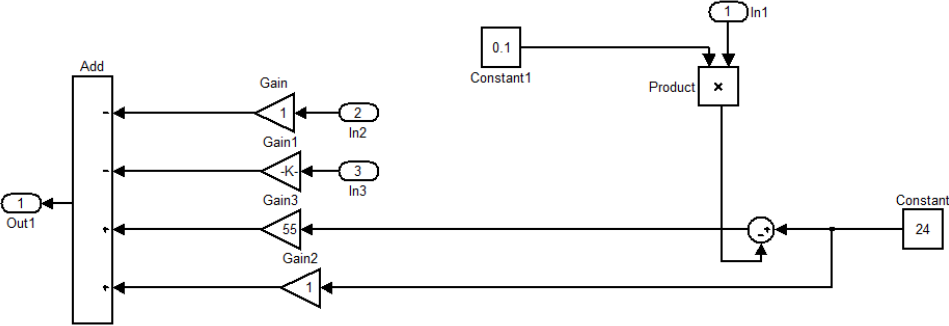


Figure 3.6. Bloc de Commande a boucle fermée



$$\begin{cases} \dot{V}_r = -\beta V_0 + V_{\text{ref}} \\ \dot{V}_0 = -\frac{1}{R_L C} V_0 + \frac{1}{C} I_L \\ \dot{I}_L = -\frac{1}{L} V_0 + \frac{1}{L} V_i U \end{cases} \quad (3.7)$$

Le modèle d'équations d'état du système peut être décrit comme indiqué en (3.8):

$$\dot{x} = Ax + BU + D \quad (3.8)$$

Où

$$A = \begin{bmatrix} 0 & -\beta & 0 \\ 0 & -\frac{1}{R_L C} & \frac{1}{C} \\ 0 & -\frac{1}{L} & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ \frac{V_i}{L} \end{bmatrix}, D = \begin{bmatrix} V_{\text{ref}} \\ 0 \\ 0 \end{bmatrix} \text{ et } x = \begin{bmatrix} V_r \\ V_0 \\ I_L \end{bmatrix}$$

### 3.4.2 Conception de la commande par MG

L'idée de base de la commande par MG est de concevoir une surface de glissement dont la loi de commande orientera la trajectoire vers une origine souhaitée [5]. La fonction de la commande par MG est décrite la relation suivante:

$$U = \begin{cases} 1 & \text{si } S > 0 \\ 0 & \text{si } S < 0 \end{cases} \quad (3.9)$$

Où  $S$  est la trajectoire des variables d'état instantanées tend vers zéro, il est décrit comme suit:

$$S = -K_i I_L - K_v V_0 + K_r V_r + K_{\text{ref}} V_{\text{ref}} \quad (3.10)$$

Avec  $J^T = [K_i \ K_v \ K_r \ K_{\text{ref}}]$  et  $K_i, K_v, K_r$  et  $K_{\text{ref}}$  des paramètres de la commande appelés coefficients de la surface de glissement. Les coefficients de la surface de glissement sont déterminés par la méthode dite placement de pôles robustes.

### 3.4.3 Conditions d'existence de la surface de glissement :

Pour s'assurer de l'existence de la surface par MG d'un système, une condition d'existence doit être réalisée:

$$\lim_{s \rightarrow 0} S \cdot \dot{S} < 0 \quad (3.11)$$

Dérivant l'équation (3.10) et remplaçant  $s$  et  $\dot{s}$  par leur expression nous obtenant :

$$\begin{cases} \dot{S}_{s \rightarrow 0+} = J^T Ax + J^T BU_{s \rightarrow 0+} + J^T D < 0 \\ \dot{S}_{s \rightarrow 0-} = J^T Ax + J^T BU_{s \rightarrow 0-} + J^T D > 0 \end{cases} \quad (3.12)$$

Une démonstration est illustrée pour le cas du convertisseur BUCK.

- Cas 1 :  $S \rightarrow 0^+$      $\dot{S} < 0$  :

Remplaçant  $U_{S \rightarrow 0^+} = \bar{U} = 0$  l'équation (3.12) devient :

$$K_r(V_{ref} - \beta V_0) - k_v \left( \frac{1}{C} I_L - \frac{1}{R_L C} V_0 \right) + k_i \frac{1}{L} V_0 < 0 \quad (3.13)$$

- Cas 2 :  $S \rightarrow 0^-$      $\dot{S} > 0$  :

Remplaçant  $U_{S \rightarrow 0^-} = \bar{U} = 1$  l'équation (3.12) devient :

$$K_r(V_{ref} - \beta V_0) - k_v \left( \frac{1}{C} I_L - \frac{1}{R_L C} V_0 \right) + k_i \left( \frac{1}{L} V_0 - \frac{V_i}{L} \right) > 0 \quad (3.14)$$

### 3.4.4 Commande équivalente :

L'analyse de stabilité de la commande par MG est réalisée avec la commande dite : commande équivalente. Cette dernière est obtenue si le système vérifie la condition suivante :

$$\dot{s} = 0$$

Et on remplaçant  $u$  par  $u_{\acute{e}q}$ . L'expression de la commande équivalente est donc donnée par :

$$U_{eq} = \frac{L}{V_i} \left[ \left( \frac{1}{R_L C} \frac{k_v}{k_i} - \beta \frac{k_r}{k_i} + \frac{1}{L} \right) V_0 - \frac{1}{C} \frac{k_v}{k_i} I_L + \frac{k_r}{k_i} V_{ref} \right] \quad (3.15)$$

Dans ce cas, une condition nécessaire et suffisante pour l'existence locale de MG :

$$-1 < U_{eq} < 1 \quad (3.16)$$

### 3.4.5 Équation d'état en MG

Les équations d'état en MG sont obtenues en remplaçant dans l'équation (3.8) la commande  $u$  par  $u_{\acute{e}q}$ . On obtient alors :

$$\begin{bmatrix} \dot{V}_r \\ \dot{V}_0 \\ \dot{I}_L \end{bmatrix} = \begin{bmatrix} 0 & -\beta & 0 \\ 0 & -\frac{1}{R_L C} & \frac{1}{C} \\ 0 & \frac{k_v}{k_i} \frac{1}{R_L C} - \beta \frac{k_r}{k_i} & -\frac{1}{C} \frac{k_v}{k_i} \end{bmatrix} \begin{bmatrix} V_r \\ V_0 \\ I_L \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ \frac{k_r}{k_i} \end{bmatrix} V_{ref} \quad (3.17)$$

La dynamique du système en MG est donnée par l'équation caractéristique suivante :

$$P(s) = \det(sI - A^*) = s \left[ s^2 + \left( \frac{1}{C} \frac{k_v}{k_i} + \frac{1}{R_L C} \right) s + \frac{\beta k_r}{C k_i} \right] = 0 \quad (3.18)$$

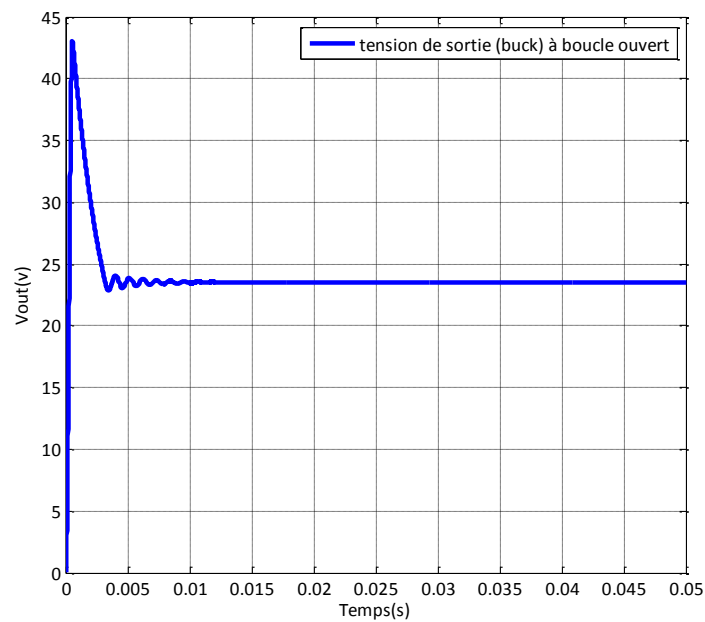
Notons ici qu'une racine de l'équation caractéristique est nulle en raison de la dépendance linéaire des variables d'état lorsque le système est en MG. La dynamique du système est donc influencée par les paramètres de la surface de glissement :  $K_v / k_i$ ,  $k_r / k_i$  et la constante de temps du filtre  $RLC$ .

Cependant, la dynamique du système n'est pas affectée par les variations de la tension  $V_i$  et de l'inductance  $L$ . Ceci est dû à une qualité intrinsèque des MG qui est la robustesse.

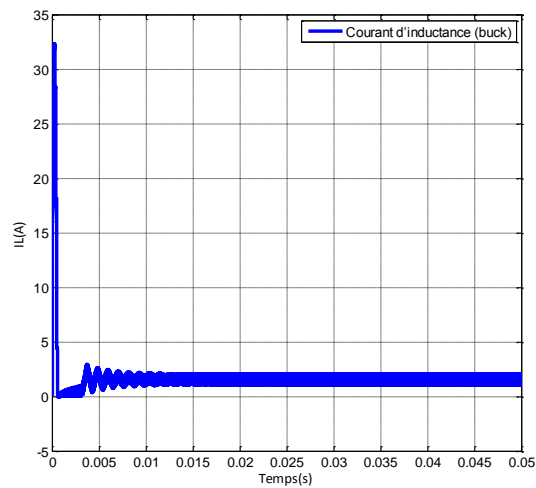
### 3.4.6 Résultats de simulation

Les valeurs adapte pour la simulation dans le cas du convertisseur Buck sont résumes dans le tableau suivant

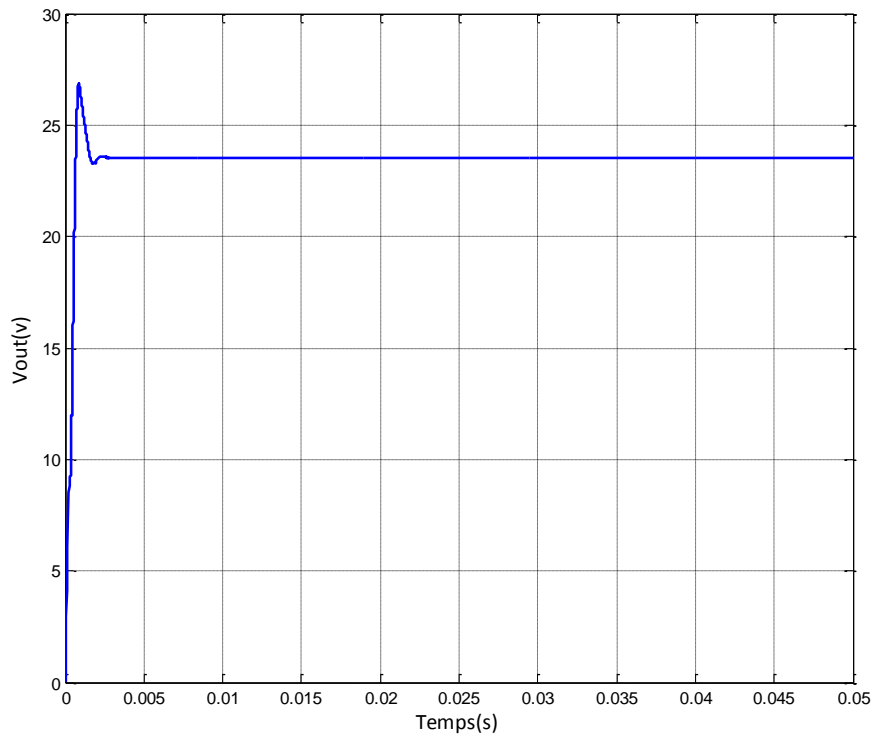
Ki	Kref	Kv	Kr	B	Vref	Vi	Vo	C	L	F
1	1	0.045	1	0.1	24	48V	24V	250uF	125UF	100KHZ



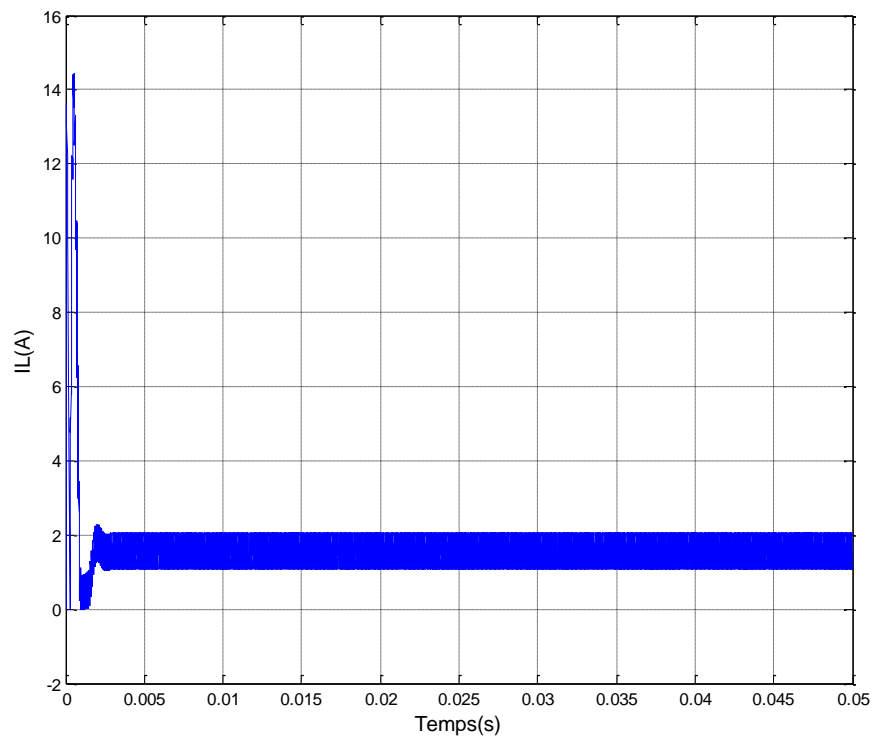
**Figure 3.7.** Tension de sortie (Buck) à boucle ouvert



**Figure3.8.** Courant d'inductance (buck)



**Figure 3.9.** Tension de sortie (Buck) boucle fermée



**Figure3.10.** Courant d'inductance (Buck) à boucle fermée

### 3.5 Commande par MG du convertisseur Boost DC/DC [27]

Considérons  $x$  l'erreur sur la tension, et  $Y$  dérivée alors  $Z$  son intégrale. En mode de conduction continu [7] nous obtenant les expressions suivantes :

$$X = (V_{ref} - \beta V_0) \quad (3.24)$$

$$Y = \dot{X} = \frac{\beta}{C} \left[ \frac{v_0}{RL} - \int 1 \frac{uV_i - V_0}{RL} dt \right] \quad (3.25)$$

$$Z = \int X dt \quad (3.26)$$

$$X_{boost} = \begin{bmatrix} V_{ref} - \beta V_0 \\ \frac{\beta}{C} \left[ \frac{v_0}{RL} - \int 1 \frac{uV_i - V_0}{RL} dt \right] \\ \int V_{ref} - \beta V_0 \end{bmatrix} \quad (3.27)$$

$$X_{boost} = AX_{boost} + Bu \quad (3.28)$$

Où,

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & \frac{1}{R1C} & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (3.29)$$

$$B = \begin{bmatrix} 0 \\ \frac{\beta V_0}{LC} - \frac{\beta V_i}{LC} \\ 0 \end{bmatrix} \quad (3.30)$$

Pour notre système, il est judicieux de choisir une commande par MG appropriée basée sur la fonction de commutation telle que :

$$\begin{cases} u = 1 & \text{si } S > 0 \\ u = 0 & \text{si } S < 0 \end{cases} \quad (3.31)$$

$$u = \frac{1}{2(1 + \sin s)} \quad (3.32)$$

$S$  est défini comme étant la trajectoire de la variable d'état instantanée, elle est décrit comme :

$$S = \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3 = J^T X \quad (3.33)$$

$$\text{Avec, } J^T = [\alpha_1 \quad \alpha_2 \quad \alpha_3] \quad (3.34)$$

Où  $\alpha_1$ ,  $\alpha_2$  et  $\alpha_3$  représentent les paramètres de la commande par MG appelés coefficient de la surface de glissement.

La surface de glissement peut être obtenue en posons  $S = 0$ . Le signal de commande  $u$  est défini comme le rapport entre la tension  $V_c$  et la tension  $V_{ramp}$  tel que :

$$0 < d = \frac{V_c}{V_{ramp}} < 1 \quad (3.35)$$

Les tensions  $V_c$  et  $V_{ramp}$  sont données par les relations suivantes :

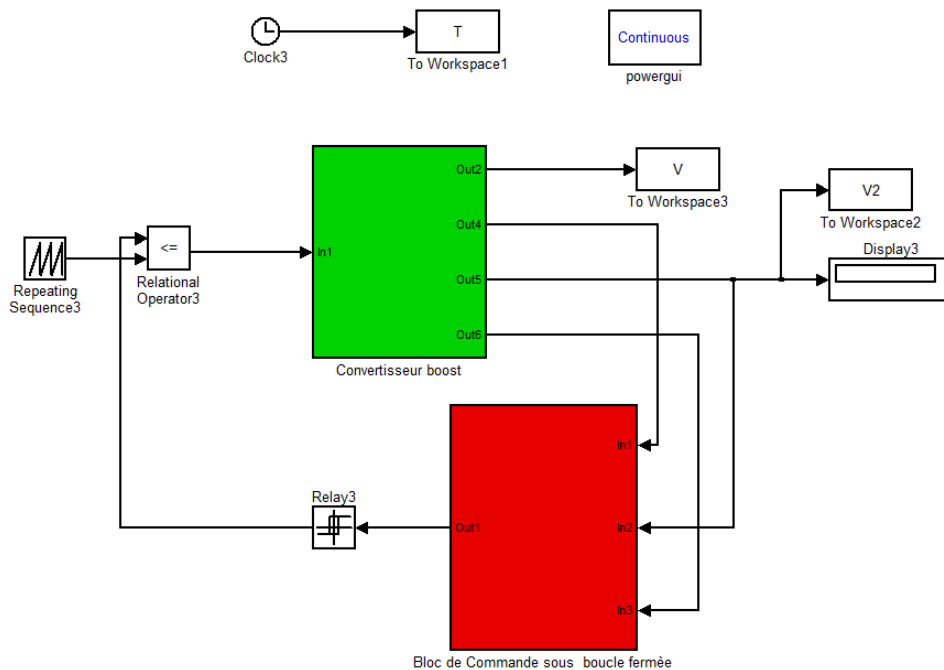
$$V_c = U_{equ} = -\beta L \left[ \left( \frac{\alpha_1}{\alpha_2} \right) - \left( \frac{1}{R1C} \right) \right] i_c + LC \left( \frac{\alpha_3}{\alpha_2} \right) (V_{ref} - \beta V_0) + \beta (V_0 - V_i) \quad (3.36)$$

$$V_c = -kp1 i_c + kp2 (V_{ref} - \beta V_0) + \beta (V_0 - V_i) \quad (3.37)$$

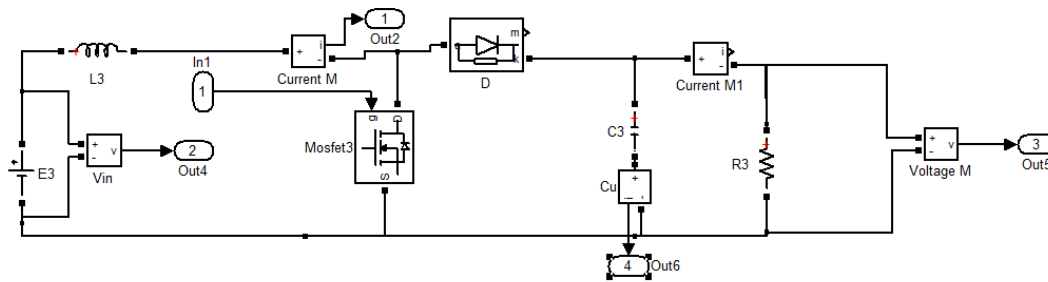
$$kp1 = \left( \frac{\alpha_1}{\alpha_2} \right) - \left( \frac{1}{R1C} \right) \text{ et } kp2 = LC \left( \frac{\alpha_3}{\alpha_2} \right) \quad (3.38)$$

$$V_{ramp} = \beta (V_0 - V_i) \quad (3.39)$$

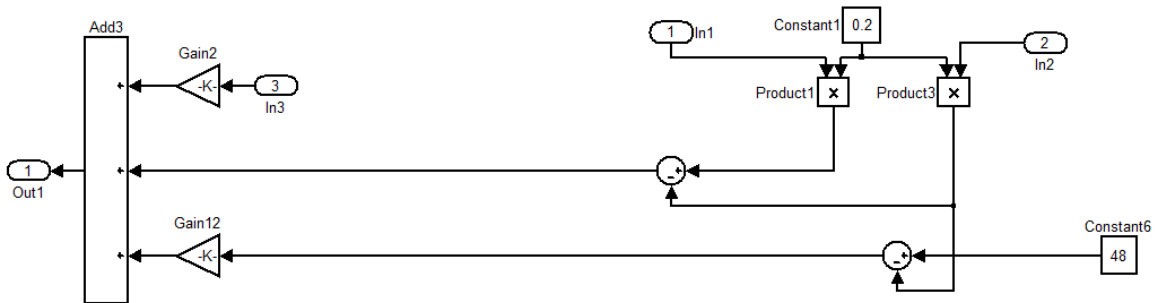
En utilisant la commande par MG, l'architecture adoptée pour le contrôle du convertisseur Boost est illustrée par la figure suivante (3.5) :



**Figure 3.11.** Circuit de Commande du Convertisseur DC-DC Boost par Mode Glissant



**Figure 3.12.** Représentation schématique d'un convertisseur Boost

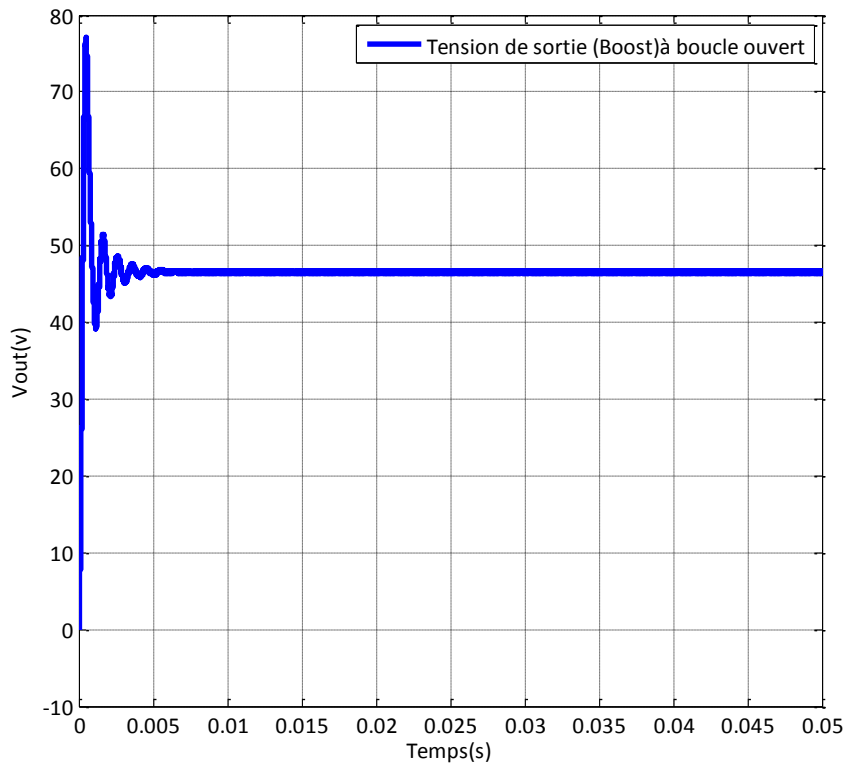


**Figure 3.13.** Bloc de Commande à boucle fermée

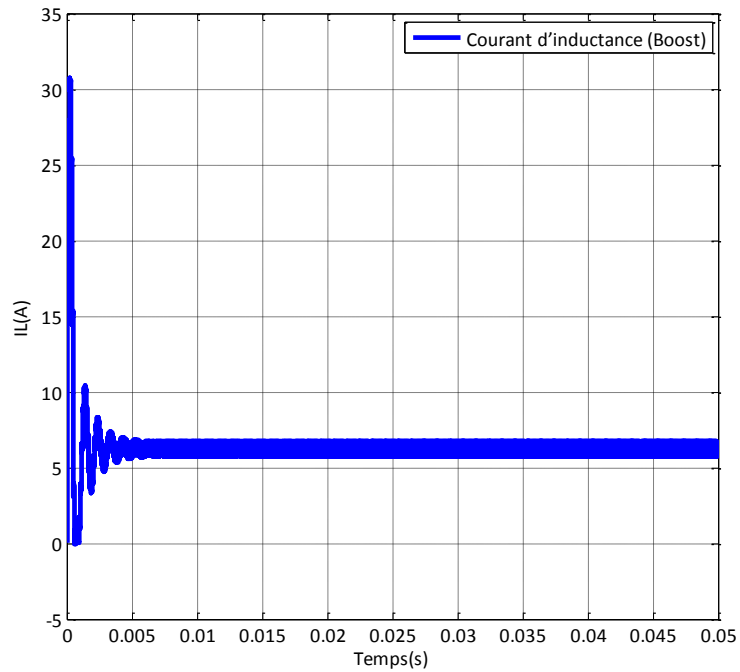
### 3.5.1 Résultats de simulation

Les valeurs adapte pour la simulation dans le cas du convertisseur Boost sont résumes dans le tableau suivant :

Description	Paramètre	Valeur minimal
Input voltage	Vin	24V
Capacitance	C	50 Uf
Inductance	L	120Uh
Switching frequency	F	100 KHz
Load resistance	R1	50
Sliding mode controller gain	Kp1	0.149
	Kp2	1.35
PID	Kp	25
Integral constant	Ki	12
Derivative constant	Kd	0.05
Pi	Kp	0.17
Integral constant	Ki	15
Expected voltage	VO	48

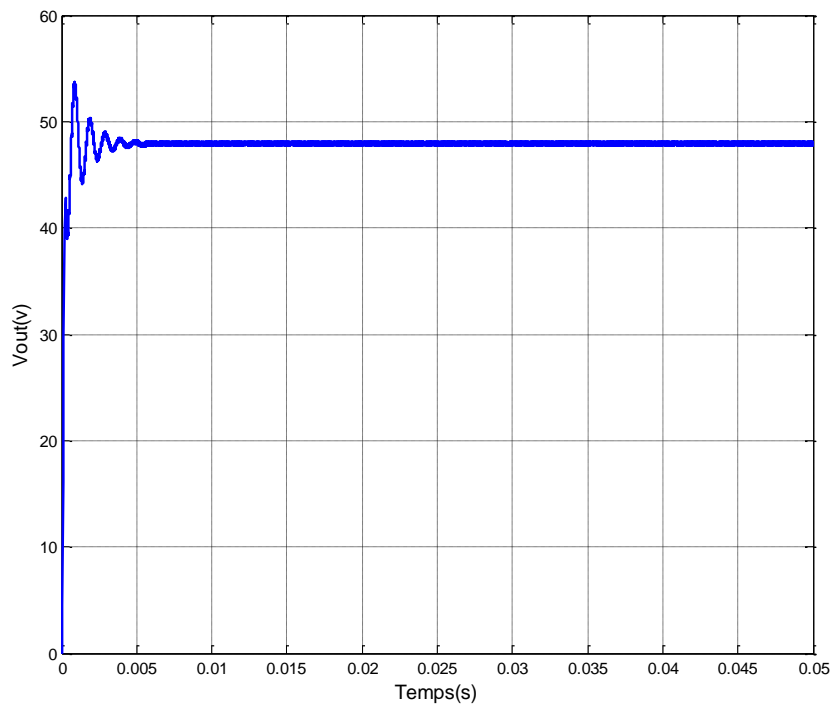


**Figure 3.14.** Tension de sortie à boucle ouvert (Boost)

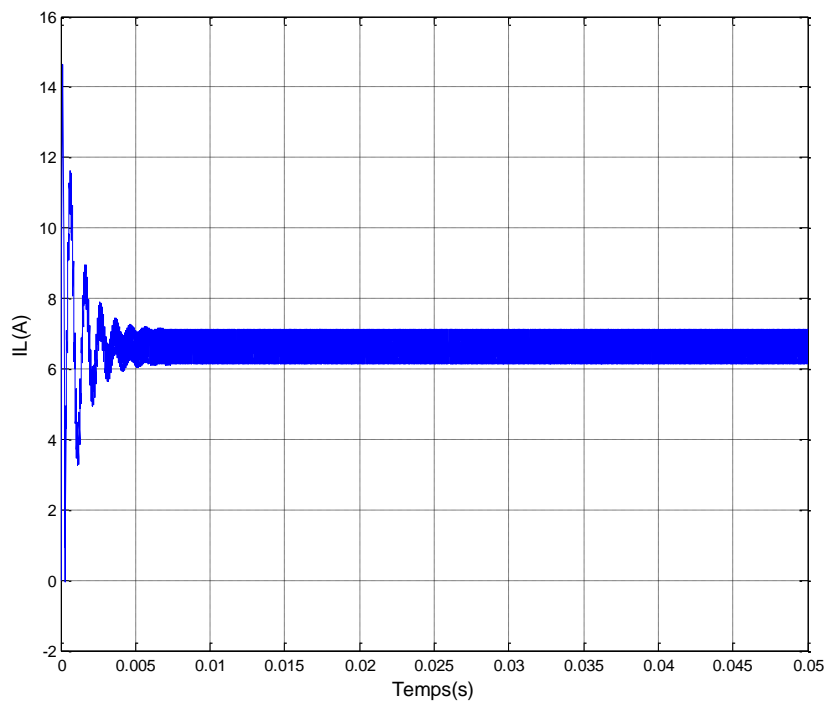


**Figure 3.15.** Courant d'inductance (Boost)





**Figure 3.16** Tension de sortie (Boost) à boucle fermée



**Figure 3.17.** Courant d'inductance (Boost)

### **3.6 Conclusion**

Dans ce chapitre, on a décrit les principes de base de la théorie de la commande par mode de glissement appliquée aux convertisseurs DC/DC. Cette technique s'adaptant bien aux cas des systèmes à structure variables et sur tout connue pour sa robustesse vis-à-vis des changements des paramètres ou des perturbations des systèmes. Les résultats des simulations confirment bien cet aspect théorique et démontrent que l'approche de commande par mode glissant.

# Chapitre 4 : Modèle Xilinx System Generator

---

## 4.1 Brève description

Dans le cadre de cette recherche, le développement du prototype s'effectue avec le logiciel XSG. Il s'agit d'un toolbox développé par Xilinx pour être intégré dans l'environnement Matlab Simulink et qui laisse l'utilisateur créer des systèmes hautement parallèles pour FPGA. Les modèles créés sont affichés sous forme de blocs, et peuvent être raccordés aux autres blocs et autres toolboxes de Matlab-Simulink comme SPS. Une fois le système complété, le code VHDL généré par l'outil XSG reproduit exactement le comportement observé dans Matlab. Pour le prototypage rapide, le choix de cet outil est facilement explicable. Le système de contrôle devant être vérifié et simulé souvent et rapidement pendant tout le développement, il est beaucoup plus simple d'analyser les résultats avec Matlab qu'avec les outils habituellement associés au VHDL, tel que Modelsim. Aussi, le modèle peut ensuite être couplé à des moteurs virtuels (à l'aide du toolbox SPS) et des simulations en boucle fermée sont réalisables. Quand le prototype fonctionne, le passage vers la plateforme matérielle pour des tests sur le terrain est rapide, ce qui rend la validation du prototype un projet réalisable à court terme.

## 4.2 Méthode de développement

Avec une banque de résultats provenant de la référence établie, le développement des blocs du prototype peut se faire. L'outil XSG est utilisé afin de produire un modèle qui va tout de suite fonctionner sur le matériel une fois achevé et validé.

### 4.2.1 Développement de bloc individuel et validation de premier niveau

L'avantage d'utiliser XSG pour le prototypage rapide devient plus évident lorsqu'il est nécessaire de tester un bloc achevé. Il suffit de brancher aux entrées les données intermédiaires obtenues de la référence. En simulant, on recueille les données à la sortie pour ensuite les comparer avec les données de la référence. Notons qu'il peut y avoir une certaine erreur, car une précision arbitraire selon le nombre de bits des opérandes est employée. L'outil GAPPA va plus tard permettre d'ajuster cette précision. Pour l'instant, il faut seulement vérifier le comportement du bloc.

Lorsqu'un bloc se comporte bien, ce test s'effectue assez rapidement, car le temps de simulation est plus court. Nous avons déjà mentionné que les signaux intermédiaires ne sont pas emmagasinés en trop grand nombre, de façon à rendre le temps de simulation acceptable. Cependant, lorsqu'un bloc ne répond pas comme il se doit, le travail devient plus important. Dans le cas de la commande vectorielle, plusieurs blocs sont d'une complexité importante comme le bloc découplage ou le bloc estimateur de flux du rotor. La provenance de l'erreur à l'intérieur du bloc est souvent difficile à identifier. Il faut alors reprendre notre méthode en traitant le bloc comme un algorithme séparé. On retourne vers la référence afin d'identifier les signaux que l'on retrouve à l'intérieur du bloc problématique, et une nouvelle collecte de données est effectuée. Ensuite, de nouvelles sorties temporaires sont instanciées à l'intérieur du bloc fautif pour tenter d'identifier la source du problème.

Afin de rapidement identifier si le comportement du modèle XSG correspond au modèle théorique Simulink, les résultats des deux modèles sont comparés. Cependant, il s'agit de tests qui ne serviront pas seulement lors du développement initial, mais qui seront sollicités lors de toutes les phases de débogage. Ainsi, il est important d'établir certaines règles afin d'accéder rapidement à cet environnement lorsqu'un problème survient. Deux méthodes sont ici proposées : la méthode simultanée et la sainte croyance

Voyons premièrement les conditions communes aux deux méthodes

- **le bloc XSG utilise le même fichier d'initialisation que le modèle final**

Il est important d'utiliser le même fichier d'initialisation que le modèle XSG final afin de ne pas avoir à créer d'autres copies de ce fichier. Chaque fichier supplémentaire d'initialisation doit être modifié si un changement de constante, de précision, etc. est effectué. Lorsqu'un modèle est composé de nombreux blocs, maintenir un nombre égal de scripts d'initialisation devient une source d'erreurs trop importante

- **Les données (entrées et sorties) sont toutes dans une même source**

Dans les deux méthodes proposées, les entrées proviennent de l'enregistrement des signaux lors d'un test en boucle fermée du système de référence (Simulink et SPS). Comme tous ces signaux sont enregistrés au même endroit (ex. : fichier MAT), il est important de toujours utiliser cette source,

même si une seule portion des signaux est utilisée. Encore une fois, la maintenance de multiples fichiers doit être évitée pour minimiser les causes d'erreur et faciliter la réutilisation de l'environnement de bloc individuel lors de débogage avancé

- **Les entrées respectent la nomenclature des signaux du modèle de référence Simulink**

Dans un système complexe, plusieurs signaux sont insérés dans des blocs différents afin de subir plusieurs traitements. Il convient donc d'utiliser le même nom pour un signal injecté dans plusieurs blocs différents. Dans l'environnement de bloc individuel, les entrées sont souvent des signaux provenant du workspace ou d'un fichier. Comme la deuxième règle spécifie que l'on doit utiliser un seul et même fichier de source, les noms des entrées vont devoir correspondre dans chaque environnement de test. Les sorties ne vont pas se plier exactement à cette règle selon la méthode choisie.

- Les blocs sont encapsulés exactement comme dans le modèle final ou de référence

Il est important d'utiliser le même routage de fils afin de pouvoir supprimer et remplacer un bloc rapidement lors de modifications à vérifier.

- **L'analyse des résultats se trouve dans un fichier Matlab unique au bloc, exécutable en tout temps**

Chaque bloc doit avoir son fichier Matlab d'analyse qui va générer les résultats (et souvent, les graphiques) permettant de valider le bloc XSG. Un seul fichier par bloc assure que l'utilisateur n'a pas besoin de faire fonctionner les autres blocs (ou de commenter des lignes du fichier Matlab) pour faire une analyse. Le fichier doit également pouvoir produire une analyse en tout temps, afin que l'utilisateur puisse faire une pause dans sa simulation et analyser les résultats obtenus jusqu'à cette pause, sans devoir compléter un temps donné de simulation

#### **4.2.1.1 méthodes simultanées**

L'environnement contient une source d'entrée, deux blocs et deux ensembles de sorties. La source d'entrée commune aux deux blocs provient du modèle de référence Simulink. Ces signaux sont ensuite injectés dans le bloc XSG ainsi que son bloc

Sinnulink correspondant, en parallèle. Les deux sorties (XSG et référence) sont ensuite enregistrées et comparées à l'aide du fichier d'analyse Matlab créé pour le bloc. Cette méthode est la plus efficace pour tester la correspondance des deux blocs, car ceux-ci sont tous les deux isolés et doivent traiter les mêmes entrées. Les connexions et délais entre les blocs du système ne sont donc pas une cause d'erreur possible dans ce scénario cependant, la comparaison des deux systèmes occasionne souvent des problèmes de « timing ». Le nombre de pas nécessaire pour le traitement n'est pas forcément le même pour les deux blocs, ce qui engendre l'insertion de blocs supplémentaires pour synchroniser les résultats à comparer et introduit conséquemment une nouvelle source d'erreur. Mais le plus gros désavantage de cette méthode est que, advenant un changement dans le bloc XSG, les contraintes temporelles peuvent être appelées à changer, ce qui signifie que l'utilisateur doit modifier le mécanisme de synchronisation entre les deux blocs. Un tel ouvrage ralentit l'utilisateur lors de tests de débogage et détériore la facilité de réutilisation de l'environnement

#### **4.2.1.2 la sainte croyance**

Cette méthode ne contient qu'un seul bloc par environnement, soit le bloc XSG. Pour des entrées qui proviennent de la simulation du modèle Simulink théorique entier, le bloc XSG traite ces données et produit des résultats qui sont ensuite comparés aux résultats enregistrés aux sorties du même bloc dans le modèle théorique entier. On suppose ici que le bloc du modèle théorique est juste, et que les sorties théoriques utilisées sont fiables pour l'analyse.

L'avantage d'une telle méthode est la rapidité d'utilisation. Un seul bloc doit être simulé par l'ordinateur, un seul ensemble de sorties doit être enregistré, et l'environnement ne change jamais, peu importe les modifications apportées au bloc XSG. La réutilisation de cet environnement est donc très grande, et le débogage s'effectue rapidement.

Cependant, un changement des constantes et paramètres de n'importe quel bloc du système affecte les résultats du système de référence Simulink. Donc, pour chaque changement de constante, la simulation du modèle de référence en entier doit à nouveau être effectuée. Cela n'est pas toujours nécessaire en utilisant la méthode simultanée, car pour une même entrée, les deux traitements sont simulés. Pour un modèle complexe ce désavantage peut coûter du temps plutôt qu'en sauver. Le plus

grand désavantage est au niveau de la précision d'identification, car cette méthode ne permet pas de détecter un problème qui vient des interconnexions entre les blocs. Par exemple, un usager pourrait chercher longtemps le problème de correspondance entre deux sorties, alors qu'avec la méthode simultanée, l'erreur serait la même à la sortie des deux traitements, ce qui confirme que le bloc XSG démontre le même comportement que

### **4.3 Développement d'ensemble, liaisons et validation de deuxième niveau**

Le test de chaque bloc individuel peut sembler laborieux, mais le temps investi dans cette étape se rentabilise lors du débogage du système complet. En effet, il serait laborieux de chercher le bloc fautif qui rend le système complet inopérant. Plus le test des blocs individuels est rigoureux, moins le test du système complet est laborieux. Souvent, comme dans le cas de la commande vectorielle, le défi est alors de bien synchroniser les blocs entre eux. C'est ici que des signaux « START » et « READY » peuvent être implémentés afin de s'assurer que le traitement des données se fait dans le bon ordre. Il ne s'agit plus ici d'un problème d'ordre mathématique

### **4.4 Contrôle vectoriel**

Le contrôle vectoriel XSG est divisé selon les mêmes blocs schématisés lors de la prise de connaissance de l'algorithme de contrôle. De cette façon, il est plus facile de comparer et de tester les blocs de manière individuelle durant le développement. Des schémas internes de chaque sous système composant le contrôle vectoriel se trouvent à l'annexe E., mais plutôt d'ordre logistique.

En incluant le calcul des délais d'activation [*firing*), le contrôle demande 122 pas d'horloge avec une précision adéquate pour donner les résultats le contrôle ne s'effectue que deux fois pour une période de commutation, ce qui signifie qu'à une vitesse d'horloge de 62.5 MHz (environs 2 us), il a amplement le temps de se compléter. Advenant le cas où il serait désirable d'effectuer des contrôles en série, il faut considérer qu'on doit attendre 70 pas d'horloge entre chaque échantillonnage. Ceci est dû au fait que l'opération de division et celle de racine carrée présentes dans le système doivent produire un résultat avant de traiter une nouvelle valeur, La figure 4.1 illustre les propriétés d'un bloc d'opération XSG où la précision peut être changée

par une valeur ou une variable. La précision des signaux d'entrée de tout le système (courants, tensions, vitesse) est représentée sur 16 bits, car il s'agit de la précision offerte par la carte AIO utilisée au laboratoire

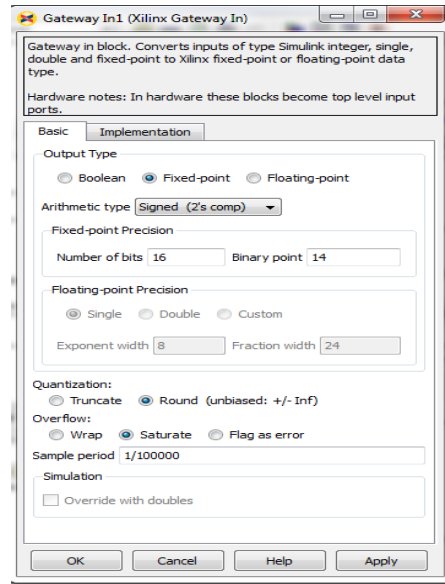


Figure 4.1 - propriétés d'un bloc XSG

#### 4.5 Simulation convertisseur Buck sous l'environnement XSG

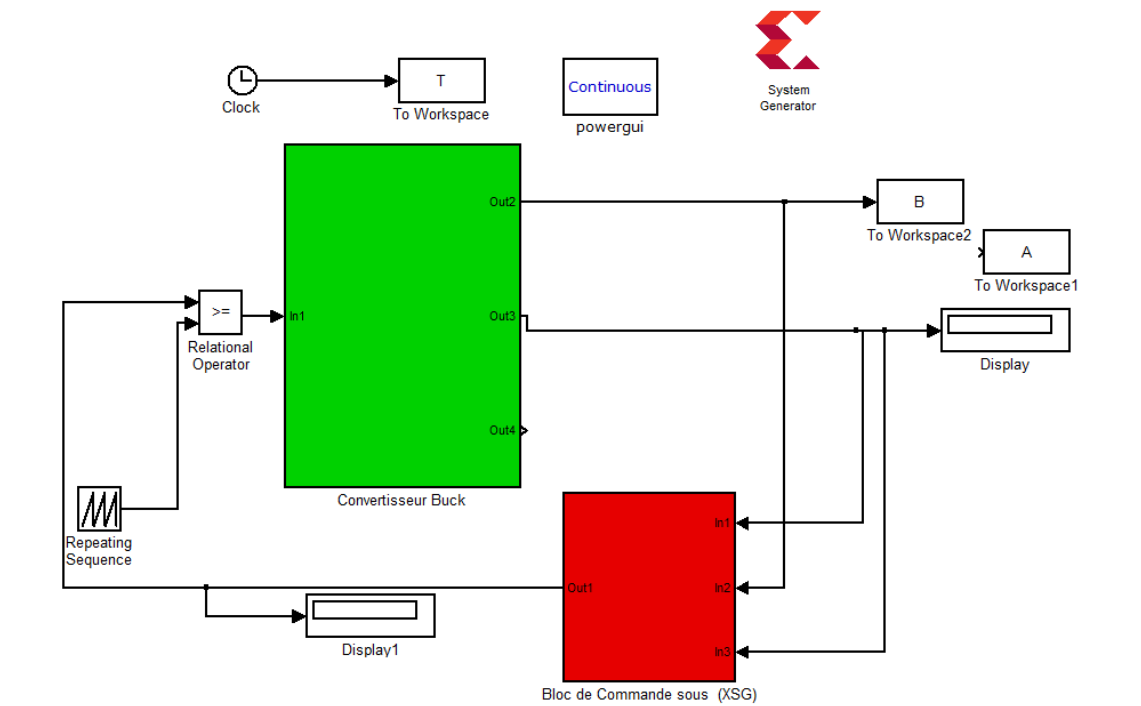
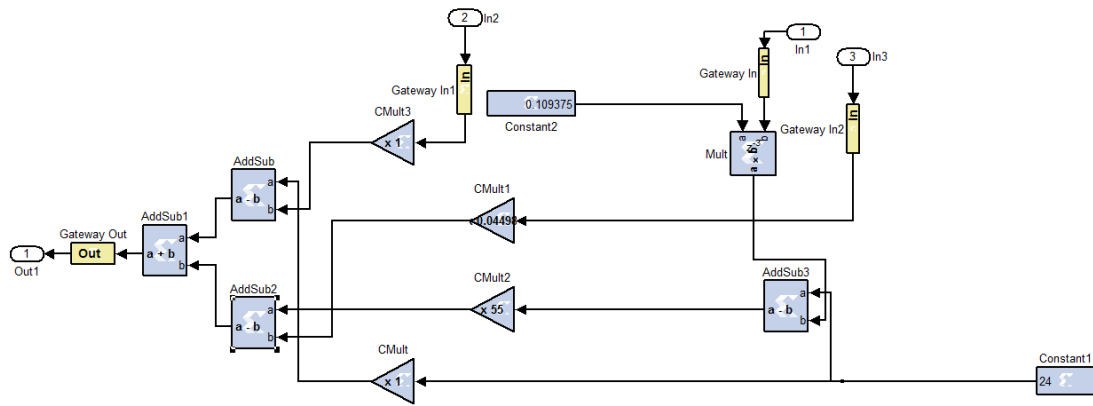


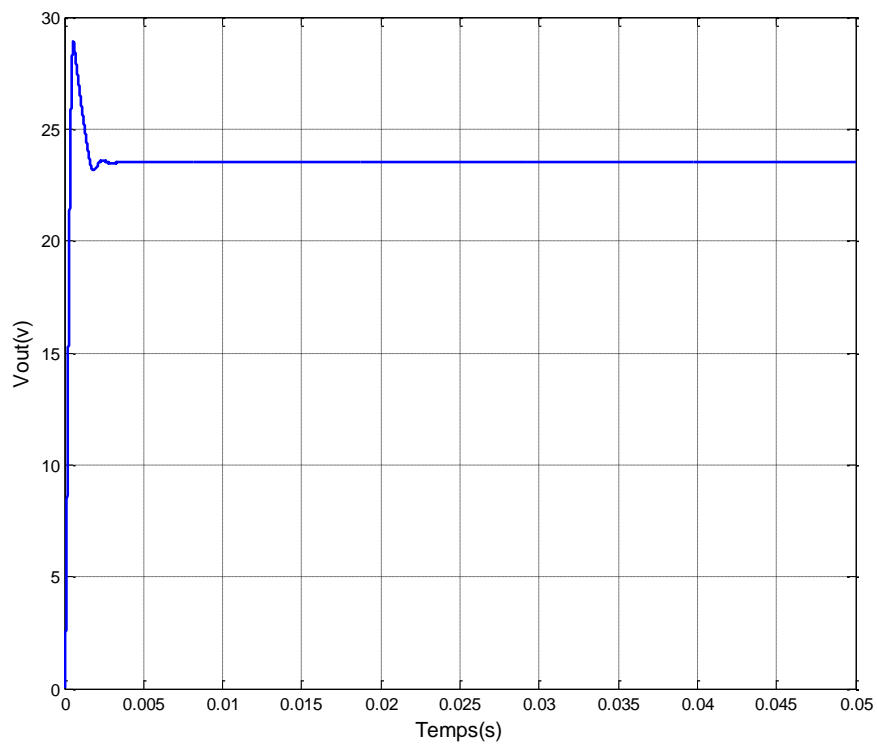
Figure 4.2: simulation de convertisseur Buck sous (XSG)





**Figure 4.3.** Bloc de Commande sous xilinx system generator (XSG)

## 4.6 Résultat de simulation en (XSG)



**Figure4.4.** Tension de sortie (Buck) sous (XSG)

## 4.5 Simulation convertisseur Boost sous l'environnement XSG

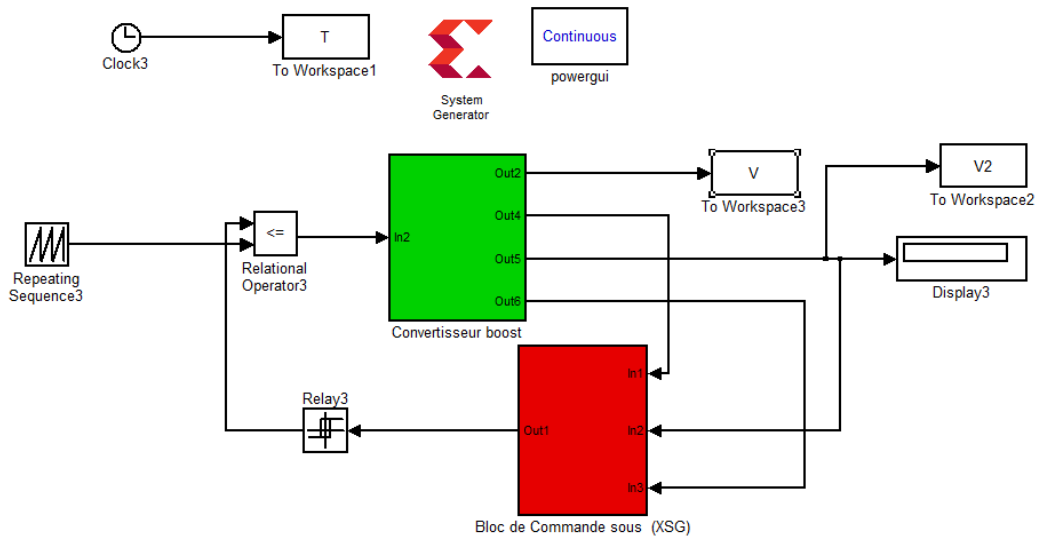


Figure 4.5. Simulation de convertisseur Boost sous (XSG)

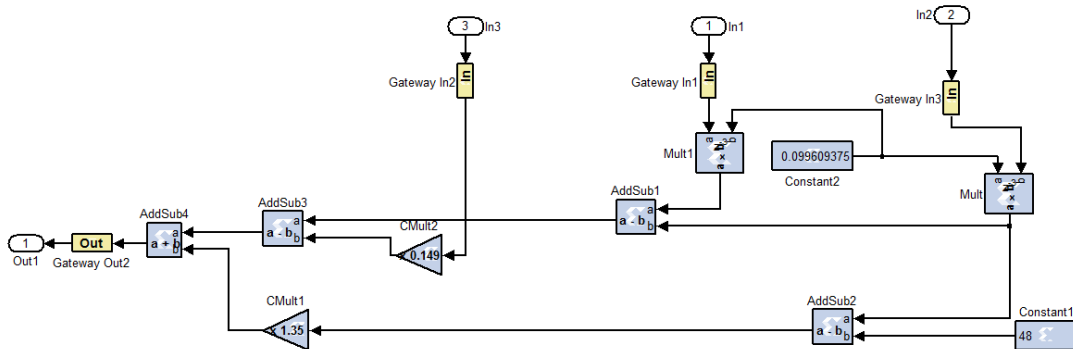
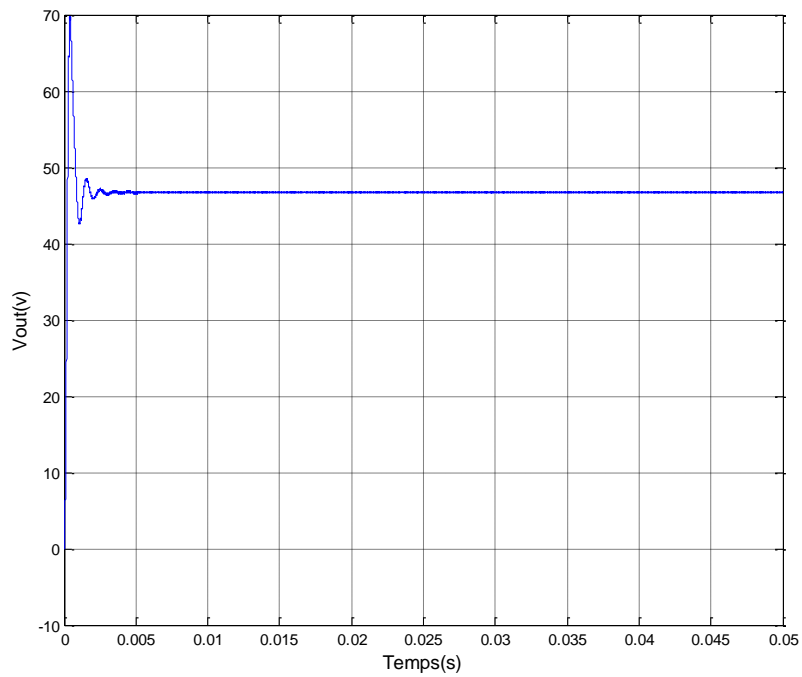
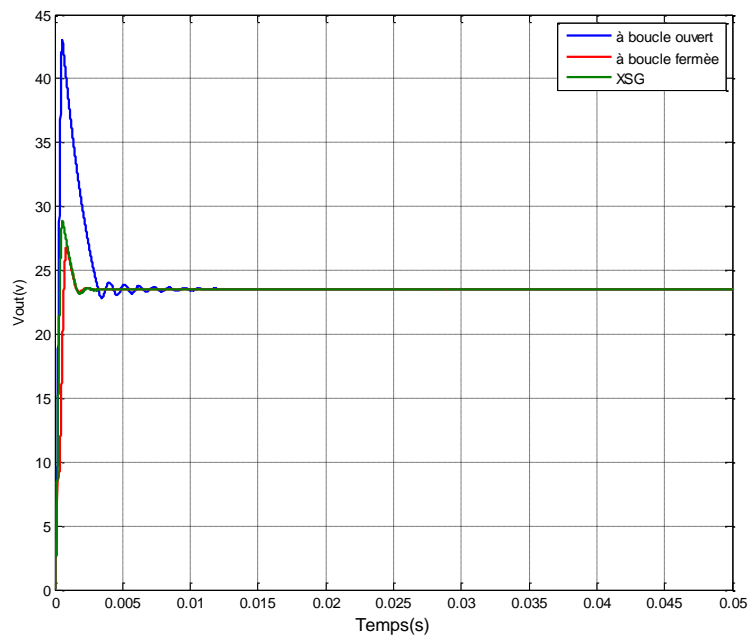


Figure 4.6. Bloc de Commande sous xilinx system generator (XSG)



**Figure 4.7.** Tension de sortie (Boost) sous (XSG)

**Comparaison :**



**Figure4.8.** Comparaison tension de sortie(Buck) XSG, Boucle fermée et Boucle ouvert

Parametre	Boucle ouvert	Boucle fermée	XSG
Temp de montée(Tm)	0.0025	0.001	0.0015
Temps du pick(Tp)	0.0006	0.001	0.0008
Maximaun de dépassement (Md)	43	25	30
Temps de stabilisation (Ts)	0 .015	0.006	0.005

Tableau 4.1 Comparaison tension de sortie(Buck) XSG, Boucle fermée et Boucle ouvert

Remarque :

D'après les résultats dressés dans le tableau ci-dessus nous remarquons que le suivi de la référence dans le cas de la boucle fermée est très par rapport de XSG quand à la stabilité des systèmes nous remarquons que la boucle fermée est très stable par rapport XSG

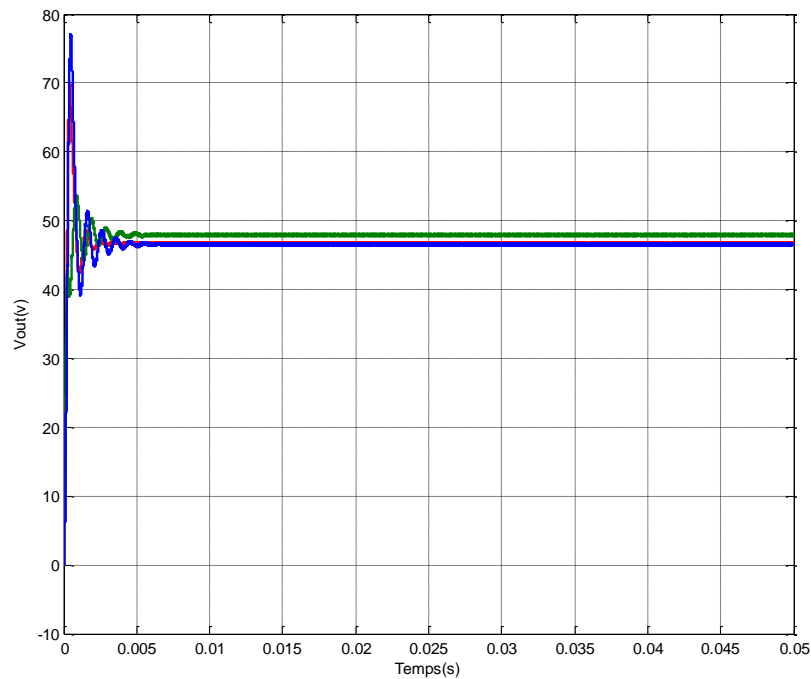


Figure 4.9. Comparaison tension de sortie(Boost) XSG, Boucle fermée et Boucle ouvert

Parametre	Boucle ouvert	Boucle fermée	XSG
Temp de montée(Tm)	0.0015	0.007	0.0010
Temps du pick(Tp)	0.0003	0.0007	0.001
Maximaun de dépassement (Md)	77	50	67
Temps de stabilisation (Ts)	0 .0065	0.005	0.004

Tableau 4.2 Comparaison tension de sortie(Boost) XSG, Boucle fermée et Boucle ouvert

Remarque :

D'après les résultats dressés dans le tableau ci-dessus nous remarquons que le suivi de la tension de référence dans le cas de la boucle fermée est très par rapport de XSG quand à la stabilité des systèmes nous remarquons que la boucle fermée est très stable par rapport à XSG

# Conclusion générale

---

## Conclusion Générale

Dans Le premier chapitre, Nous constatons que la recherche dans ce domaine (la conception des commandes implémentables sur les circuits numériques à architecture reconfigurable FPGA) est complexe, car elle nécessite non seulement une maîtrise des technologies relatives aux FPGA mai aussi une très bonne connaissance des applications et de leurs environnements. nous avons présenté un survol des circuits programmables puis nous avons étudié l'état d'art des FPGA ce qui nous a permis de conclure que la technologie FPGA s'inscrit au sommet de l'évolution des composants logique et le besoin croissant de composants plus performants, plus économiques et disponibles en grandes quantités est les grands axes du progrès qui sont disponibles dans les FPGA.

Nous constatons que la recherche dans ce domaine (la conception des commandes implémentables sur les circuits numériques à architecture reconfigurable FPGA) est complexe, car elle nécessite non seulement une maîtrise des technologies relatives aux FPGA mai aussi une très bonne connaissance des applications et de leurs environnements.

Dans le second chapitre, nous avons présenté le rôle des convertisseurs en général dans la conversion d'énergie et traités les types des convertisseurs, ainsi leur rôles dans le domaine de l'industrie et nous présentons brièvement la modélisation et leur commande. Nous avons montré que la valeur moyenne de la tension de sortie peut être ajustée en jouant sur la valeur du rapport cyclique .

Dans le troisième chapitre, une étude comparative des deux techniques de commande à savoir : la commande par modes glissants, appliquées à la mise en forme du courant et de la tension dans un convertisseur DC/DC, a été élaborée. Dans ce contexte nous avons étudié séparément chaque technique afin de définir les avantages et les inconvénients de chacune d'elle.

Dans le quatrième chapitre, nous avons implémenté sur l'FPGA la commande de deux convertisseurs à savoir l'abaisseur (Buck) et l'élévateur (Boost) par mode de glissement dans

l'environnement System Generator (XSG), les résultats obteniez confirmé les résultats dans SIMULINK plus précisé et plus stable que l'environnement (XSG).

---

# REFERENES BIBLIOGRAPHIQUES

## THESES

- [1] ZAHIR AIT OUALI : « Application des FPGA à la commande d'un moteur asynchrone » Mémoire Pour l'obtention du diplôme de Magister en Automatique, Option : Automatique des systèmes continus et productique,
- [2] J.WEBER et M.MEAUDRE « Circuits numériques et synthèse logique » un outil VHDL", Edition Masson collection technologie, 2006.
- [3] *ETIENNE MESSERLI* « Manuel VHDL synthèse et simulation », Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud (heig-vd), Version partielle septembre 2007.
- [4] JEREMIE DETREY « Arithmétiques réelles sur FPGA virgule fixe, virgule flottante et système logarithmique ». Thèse doctorat Ecole Normale Supérieure de Lyon, 15 janvier 2007.
- [5] *SEBASTIEN SNAIDERO* « Modélisation multidisciplinaire VHDL-AMS de systèmes complexes vers le prototypage virtuel ».Thèse présentée afin d'obtenir le grade de docteur de l'université Louis Pasteur Strasbourg I, décembre 2004.
- [6] *DAVID GUIHAL* « Modélisation en langage VHDL-AMS des systèmes Pluridisciplinaires. Thèse présentée au laboratoire d'analyse et d'architecture des systèmes du CNRS en vue de l'obtention du titre de docteur de l'université Toulouse, Mai 2007.
- [7] *Synopsys*, « SaberHDL : language-Independent Mixed –Signal Multi-Technology Simulator », Synposys Inc, Etats-Unis d'Amérique 2003.
- [8] *D.SMITH* « HDL Chip Design: A pratical Guide for Designing, Synthesis & Simulating Asics &FPGAs using VHDL or Verilog » Doone Pubns, 2006
- [9] *D. DECKERS* « Composants programmables » Cours de 3è Bachelier en Electronique appliquée, Institut Supérieur Industriel de Mons Haute Ecole de la Communauté française en Hainaut Unité Electronique 2008-2009



- [10] P.MELET. « Conception et réalisation d'un circuit numérique spécifique à étalement de spectre pour un système multicapteurs en milieu clos », Université Paul Sabatier, Toulouse. Dec. 2001.
- [11] C.GUILLEMINOT, L.ANDRIEUX, J.J.MERCIER « A contribution to a virtual prototyping for a spread spectrum telecommunication system using VHDL-AMS » IEEE BMAS'05, SanJose,Californie,USA, 22-23 Sept. 2005.
- [12] V. Betz and J.Rose « FPGA Routing Architecture: Segmentation and Buffering to Optimize Speed and Density ». In International Symposium on Field Programmable Gate Arrays (FPGA), 1999.
- [13] J. Arnold M. Gokhale, J. Stone and M. Kalinowski. «Stream Oriented FPGA Computing in the Streams-C High Level Language ». In Kenneth L. Pocek and Jeffrey Arnold, editors, *Symposium on FPGAs for Custom Computing Machines*. IEEE Computer Society Press, 2000.
- [14] J. Rose, A. El Gamal & A. Sangiovanni-Vincetelli. «Architecture of Field Programmable Gate Arrays ». Proceedings of the IEEE, vol. 81, July 1993.
- [15] V. Betz « Architectures and Methodologies for Dynamic Reconfigurable Logic (ADMREL) » - Information Societies Technology (IST) Program. Survey of existing fine-grain reconfigurable hardware platforms, November 2002. v2.0. 70
- [16] ZEMOURI AZEDDINE : « ETUDE EXPERIMENTALE D'UN CONVERTISSEUR DC/DC CONTROLER PAR UNE COMMANDE ROBUSTE »
- [17] BELKAID ABDELHAKIM : « Conception et implémentation d'une commande MPPT de haute performance pour une chaine de conversion photovoltaïque autonome »
- [18] W. Perruquetti and J. P. Barbot, "*Sliding mode control in engineering*", Marcek dekker, Ed. New York, USA, 2002.
- [19] H. Guldemir, "Sliding mode control of DC–DC boost converter", *Journal of Applied Sciences* 5 (3), 2005, pp.588-592.
- [20] H. Komurcugil, "Adaptive terminal sliding-mode control strategy for DC–DC buck converters", *ISA Transactions* vol.51, 2012, pp.673-681.
- [21] S. V. Emelyanov, "*Variable Structure Control Systems*", Moscow:Nauka, 1967
- [22] V. I. Utkin, "Variable structure systems with sliding mode", *IEEE Trans. Automat. Conir.*, vol. AC-22, no. 2, 1977, pp. 212-222

- [23] P. Liutanakul, "Stabilité des réseaux embarqués : Intéraction puissance-structure-commande", Thèse Doctorat de l'institut national polytechnique de Lorraine, France, 2007.
- [24] A. HIJAZI, "Modélisation électrothermique, commande et dimensionnement d'un système de stockage d'énergie par supercondensateurs avec prise en compte de son vieillissement : application à la récupération de l'énergie de freinage d'un trolleybus", Thèse de Doctorat de l'université de Lyon (France), 13 décembre 2010.
- [25] J.J.SLOTINE, « Sliding controller design for non linear systems », *IJC*, vol. 2, 1984, pp 421-434.
- [26] [J.Ackerman, << Parametrer space design of robust control systems>>. *IEEE Transaction On Automatic Control*, vol AC-25(1980).N°6, pp 1058-1072].
- [27] S. C. Tan, Y. M. *Lai*, and C. K. Tse, "Design of a PWM Based Sliding Mode Controlled Buck-Boost Converter in Continuous-Conduction-Mode" in proceedings, 11th European conference on Power Electronics And Applications (*EPE-2005*), September.
- [28] MÉMOIRE PRÉSENTÉ À L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI COMME EXIGENCE PARTIELLE DE LA MAÎTRISE EN INGÉNIERIE PAR JEAN-GABRIEL MAILLOUX 2008