

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Projet de Fin d'Études

présenté par :

OUSSAID Med AMINE

Pour l'obtention du diplôme Master en Électronique option Systèmes de Vision Robotique

Thème

Implémentation d'un Contrôleur PID pour Convertisseur DC-DC «Buck» sur FPGA, Méthodes Ziegler Nicholas et Hurwitz

Proposé par : Mr. Tadrict R.

Année Universitaire 2014-2015

Remerciements

Nous remercions dieu de nous avoir donnée patience et courage pour accomplir ce travail.

Nous remercions nos parents et nos frères et soeurs pour leur soutien, leur amour et leur encouragement.

Nous tenons à remercier notre encadreur Mr. Tadrist R qui s'est toujours montré à l'écoute et très disponible tout au long du travail, ainsi que son aide et le temps qu'il nous a consacré afin que notre travail soit bien fait et bien présenté.

Nous remercions aussi tous nos amis et nos proches qui se demandaient sur le moindre avancement du travail.

Enfin, nous n'oublions pas à remercier tous le staff du département d'électronique à l'université de BLIDA.

ملخص:

اصبح تنفيذ رقمنة دوائر الاستطاعة على FPGA هدفا للمحاكاة في الوقت الحقيقي تحديا يتعين التغلب عليها. العمل المقدم في هذه المذكرة، هو تنفيذ محول DC-DC من نوع BUCK مع وحدة تحكم PID على أساس اثنين يطبق على متن FPGA. ان دراسة و تصميم محول SEPIC شرطان اساسيان لتطوير هندسة النموذج في الموارد من مقبول بعدد تحقق Spartan 3E لوحة في pid نموذج ادماج . XSG و SIMULINK البيئتين الكلمات الرئيسية: تحويل PID ؛ ماتلاب / السيميولينك؛ FPGA؛ XSG
كلمات المفاتيح: المحول BUCK. XSG. FPGA.

Résumé :

La numérisation des circuits de puissance en vu de leur implémentation sur cible FPGA pour une simulation temps réel est devenue un défi à franchir.

Le travail présenté dans ce mémoire est consacré à l'étude et la comparaison des performances d'un régulateur PID basé sur deux méthodes différentes. La premier étant la méthode de "Ziegler-Nicolas" et la deuxième étant la méthode de "Hurwitz-Polynomial" pour le contrôle d'un convertisseur DC-DC "BUCK"

L'étude et le dimensionnement du convertisseur BUCK avec le régulateur «PID» sont nécessaires pour l'élaboration des architectures dans les environnements SIMULINK et XSG. L'implémentation du modèle de convertisseur BUCK est élaborée avec un nombre de ressources acceptable.

Mots clés : Convertisseur DC-DC; régulateur PID ; FPGA ;XSG.

Abstract :

The digitalization of high tension circuits in regards of their implementation on FPGA circuits for a real time implementation has become a real challenge to overcome.

The work presented in the paper is the implementation of a DC-DC model of type Buck with a PID regulator applied on a FPGA card. The study and the characteristics of the Buck converter with the PID regulator are necessary for the elaboration of the architectures and the environments SIMULINK and XSG, The implementation of the model of the Buck converter is done with an acceptable number of recourses.

Keywords : ; DC-DC converter; PID regulator ; FPGA ;XSG.

Listes des acronymes et abréviations :

DC/DC	Convertisseur continu-continu.
PID	proportionnel intégral dérivé
$G_c(s)$	Fonction de transfert associée à la consigne
$W(s)$	La consigne
$Y(s)$	La sortie
$E(s)$	L'erreur
$U(s)$	La commande
K_p	Gain proportionnel
K_i	Gain intégral
K_d	Gain dérivation
K_0	Gain statique du processus
K_{cr}	Gain critique
K_n	Gain normalisé
k	Gain relatif
f	Fréquence de travail
T_i	Temps d'intégration
T_d	Temps de dérivation
τ	Retard apparent
T	Constante de temps apparente

R	Résistance
K_e	Constante de vitesse
U(t)	Commande
ε	Erreur statique
1/ Ti	Action Intégral
Y	Mesure
FPGA	Field Programmable Gate Array
VHDL	Very Hardware Description Language
LCA	Logic Cell Array
SRAM	Logic Cell Array
DCM	Static Random Access Memory
D	Double Data-Rate
CPLD	Complex Programmable Logic Device
EPLD	Erasable Programmable Logic Device
PAL	Programmable Array Logic
GAL	Generic Array Logic
MOS	Metal Oxide Semiconductor
SRAM	Static Read Access Memory
CLB	Configurable Logic Bloc (bloc logique configurable)

IOB	Inptu Output Bloc (bloc d'entrées sorties)
LUT	Luck Up Table
α	Rapport cyclique
MOSFETs	Transistors à effet champ d'oxyde de métal
CLB	Configurable logic Bloc
IOB	Input Output Bloc

Table des matières :

Introduction générale	1
-----------------------------	---

CHAPITRE 1 : LES CONVERTISSEURS DC-DC .

1.1 INTRODUCTION	3
1.2 Les convertisseurs DC-DC	3
1.3 La conception du convertisseur DC-DC Buck	8
1.4 Dimensionnement de L et de C	11
1.5 Conclusion	15

CHAPITRE 2 : Le régulateur PID .

2.1 Introduction	16
2.2 Les types de régulateurs	17
2.2.1 L'Algorithme p.....	17
2.2.2 L'Algorithme I.....	18
2.2.3 L'Algorithme PID	18
2.3 Principe général d'un correcteur PID	19
2.4 Réglage d'un PID	20
2.5 Les paramètres du PID influencent sur la réponse du système de la manière suivante	21
2.6 Les caractéristiques du régulateur PID	21
2.7 Structure des régulateurs PID	22
2.7.1 Type Parallèle.....	22
2.7.2 Type Mixte	22
2.7.3 Type Série.....	23

2.8 Correction en boucle ouverte	23
2.9 Correction en boucle fermée.....	24
2.10 Les actions PID	24
2.11 La synthèse par la réponse indicielle et la réponse fréquentielle	25
2.12 Réponse indicielle	25
2.13 La réponse fréquentielle.....	26
2.14 Méthodes de Ziegler et Nichols (ZN)	27
2.14.1 Méthode de la réponse indicielle	28
2.14.2 Méthode du point critique	28
2.15 Méthodes de Horowitz	29
2.16 Conclusion	30

CHAPITRE 3 :Résultats de simulation du convertisseur DC/DC «Buck» avec un régulateur PID conçu sous SIMULINK .

3.1 Introduction	31
3.2 Calcule des valeurs de la self et de la capacité de convertisseur	31
3.3 Simulation sous SIMULINK du convertisseur DC/DC Buck	32
3.4 Calcule des paramètres de régulateur PID	33
3.4.1 Méthode de Hurwitz.....	34
3.4.2 Méthode Ziegler et Nichols (ZN)	35
3.5 Résultats de simulations et interprétations	36
3.6 conclusion	38

CHAPITRE 4 : Simulation Sous System Generator .

4.1 Introduction	39
4.2 Circuits FPGA.....	40
4.2.1 Définition.....	40
4.2.2 Structure générale des FPGAs.....	41
a. Les blocs logiques configurables(CLB)	42
b. Les blocs d'entrées/sorties.....	44
c. Les interconnexions	44
4.3 Langage de description de matériel VHDL	46
3.3.1 Définition.....	46
4.3.2 Logiciel Xilinx ISE 12.3	46
4.3.3 Utilité du VHDL	49
4.4 Plateforme de développement utilisée	49
4.4.1 Logiciel utilisé	49
a. Définition du bloc System Generator(SG).....	50
b. La précision et les passerelles Xilinx	50
4.4.1 Le choix du circuit FPGA	51
4.5 La description XSG du «régulateur PID	53
4.6 Résultats et interprétations	55
4.7 Conclusion	57
Conclusion générale	58

Liste des figures

Figure 1.1 : Convertisseur continu-continu	5
Figure 1.2 : Convertisseur Buck	6
Figure 1.3 : Convertisseur Boost	6
Figure 1.4 : Convertisseur Buck-Boost	6
Figure 1.5 Convertisseur cuk	7
Figure 1.6 : Convertisseur SEPIC	7
Figure 1.7: Circuit idéal du Buck.	8
<i>Figure 1.7(a): Schéma équivalent du hacheur série quand $t \in [0, t_{on}]$</i>	<i>9</i>
<i>Figure 1.7(b) : Schéma équivalent du hacheur série quand $t \in [t_{on}, T_s]$</i>	<i>9</i>
<i>Figure 1.8(a) : La tension dans la self</i>	<i>10</i>
<i>Figure 1.8(b) : Le courant circulant la self.....</i>	<i>10</i>
<i>Figure 1.9 : Forme de l'ondulation du courant dans l'inductance</i>	<i>12</i>
<i>Figure 1.10 : Ondulation de la tension V_o.....</i>	<i>14</i>
Figure 2.1 : Correcteur PID	19
Figure 2.2 : Réponse d'un système du second ordre	20
Figure 2.3 : régulateur parallèle	22
Figure 2.4 : régulateur mixte	22
Figure 2.5 : régulateur série.....	23
Figure 2.6 : Schéma fonctionnel d'un système asservi mono-variable (correction en boucle fermée)	24
Figure 2.7 : Réponse indicielle d'un processus apériodique	26

Figure 2.8 : Réponse fréquentielle d'un processus apériodique d'ordre 3	27
Figure 3.1 : Simulation de convertisseur DC/DC Buck	33
Figure 3.2 : Le principe de convertisseur DC/DC Buck avec un régulateur PID	33
Figure 3.3 : Simulation de convertisseur DC/DC Buck avec un régulateur PID par la méthode de Hurwitz	35
Figure 3.4 : Simulation de convertisseur DC/DC Buck avec un régulateur PID par la méthode de Ziegler et Nichols (ZN)	36
Figure 3.5 : Les tensions de sortie de convertisseur Buck pour une boucle ouverte avec les régulations PID pour la méthode de ziegler et celle de Hurwitz	37
Figure 3.6 : Les Courant dans la bobine L de convertisseur Buck pour une boucle ouverte et les régulations PID pour la méthode de Ziegler et la méthode de Hurwitz	37
Figure 4.1 : Classification des circuits logiques programmables	40
Figure 4.2 : Structure générale d'un FPGA.	42
Figure 4.3 : Bloc logique d'un FPGA	43
Figure 4.4 : Principe de travail d'un LUT.....	43
Figure 4.5 : Bloc d'E/S.	44
Figure 4.6 Interconnexions commandés par les cellules SRAMs.	45
Figure 4.7 : Flot de conception sur FPGA	48
Figure 4.8 : Le bloc SG.....	50
Figure 4.9 : La limite FPGA.	51
Figure 4.10 L'architecture générale de la série Spartan-3E.....	51
Figure 4.11 L'architecture générale de la série Virtex-II	52
Figure 4.12 : L'architecture du "Virtex-5 SXT DSP48E Slice"	53
Figure 4.13 : Modèle du « régulateur PID » sous XSG	54
Figure 4.14 : Architecture détaillée du bloc Derivé (du /dt)	54
Figure 4.15: Le bloc intégrateur en utilisant des éléments de la bibliothèque Xilinx ...	55
Figure 4.17: la superposition du signal de sortie Vo du model du «régulateur PID »	

sous XSG et Simulink	55
<i>Figure 4.18</i> : Boite de dialogue de bloc System Generator.....	56

Liste des tableaux

Tableaux 1.1 : Différents convertisseurs DC-DC utilisées dans les applications solaires.	8
Tableaux 2.1 : Paramètres PID obtenus à partir d'une réponse indicielle (ZNt).....	28
Tableaux 2.2 : Paramètres PID obtenus à partir du point critique (ZNf).....	29
Tableaux 3.1 : Les valeurs des composants du convertisseur DC-DC Buck	32
Tableaux 4.1 : utilisation de périphériques sommaire (valeurs estimées)	57

Conclusion générale

Le travail présenté dans ce mémoire est consacré à l'étude et la comparaison des performances d'un régulateur PID basé sur deux méthodes différentes. La première étant la méthode de "Ziegler-Nicolas" et la deuxième étant la méthode de "Hurwitz-Polynomial" pour le contrôle d'un convertisseur DC-DC "BUCK". Au premier chapitre, on a fait un rappel sur l'état de l'art des convertisseurs DC-DC, avec le développement de l'étude du convertisseur abaisseur "Buck". Au deuxième chapitre, on a présenté une étude des correcteurs analogiques PID, avec leurs différentes formes ainsi que les transformations qui permettent d'obtenir leur version numérique. Au troisième chapitre, on a calculé les valeurs de la self et de la capacité du convertisseur Buck, et nous avons simulé notre système sous SIMULINK/MATLAB ainsi que le calcul des paramètres du régulateur PID. En quatrième chapitre, on a implémenté le régulateur PID sur une carte FPGA en utilisant l'environnement de développement XSG, le langage de programmation VHDL est obtenu automatiquement.

Pour réaliser un tel travail, on a commencé par implémenter le circuit du convertisseur "BUCK" avec une boucle ouverte. Comme le signal de sortie du convertisseur a présenté beaucoup de dépassement importants dans le régime transitoire, on a utilisé les deux méthodes de la régulation PID par "Ziegler-Nicols" et par "Hurwitz-Polynomial" pour réduire ces dépassements. On a ensuite passé à l'implémentation de ces méthodes de régulation sur l'environnement XSG, pour enfin faire une comparaison avec celui de SIMULINK.

Les résultats obtenus sont semblables à ceux obtenus par la simulation sous Simulink. Ceci confirme la bonne description XSG du PID et nous garantit la faisabilité de l'implémentation sur la carte FPGA.

CHAPITRE1 :

LES CONVERTISSEURS

« DC-DC »

CHAPITRE 2 :
Le régulateur
« PID »

CHAPITRE 3 :

**Résultats de simulation du
convertisseur DC/DC «Buck»
avec un régulateur PID conçu
sous SIMULINK**

CHAPITRE 4 :
Simulation Sous System
Generator

Chapitre4 Simulation Sous System Generator

4.1 Introduction :

Il y a quelques années, la réalisation d'un montage en électronique numérique impliquait l'utilisation d'un nombre important de circuits intégrés logiques. Ceci avait pour conséquences un prix de revient élevé, une mise en œuvre complexe et un circuit imprimé de taille importante. Dans notre projet, l'utilisation de l'FPGA a pour but une simulation dynamique d'un régulateur PID avec deux méthodes pour contrôler un convertisseur « Buck » conçu précédemment.

Le développement des systèmes sur les FPGA est différent du développement de systèmes conventionnels puisqu'il fait appel à la conception physique mais également à la conception logicielle. L'utilisation d'un langage de description de matériel, qui partage certains aspects avec les langages de programmation conventionnels, n'est pas banale. Ce langage sert à définir physiquement le circuit au niveau d'abstraction choisi par le concepteur [31].

Pour cela, dans ce chapitre, l'architecture interne des FPGAs est présentée, puis le langage de description de matériel utilisé : *Very Hardware Description Language* (VHDL) et la génération du code VHDL par Xilinx System Generator. Finalement, les architectures du « régulateur PID » conçu sous XSG sont présentés et les résultats obtenus lors de la simulation sont discutés et comparés aux résultats de la simulation sous SIMULINK/MATLAB.

Vient enfin la programmation du circuit et la vérification du fonctionnement sur la carte. Si la simulation et la vérification ont été faites correctement, aucune erreur de fonctionnement ne doit apparaître.

4.2 Circuits FPGA :

De nombreuses familles de circuits programmables et reprogrammables sont apparues et ont évolué d'une manière étonnante depuis les années soixante dix avec des noms très divers suivant les constructeurs. Il existe dans la littérature plusieurs manières de les classer. Nous choisirons l'approche illustrée à la figure 4.1 consistant à séparer les PLDs (Programmable LogicDevice) en trois sous-classes, à savoir les SPLDs (Simple Programmable LogicDevice), les CPLDs (Complex Programmable LogicDevice) et les FPGAs (Field Programmable GateArray).

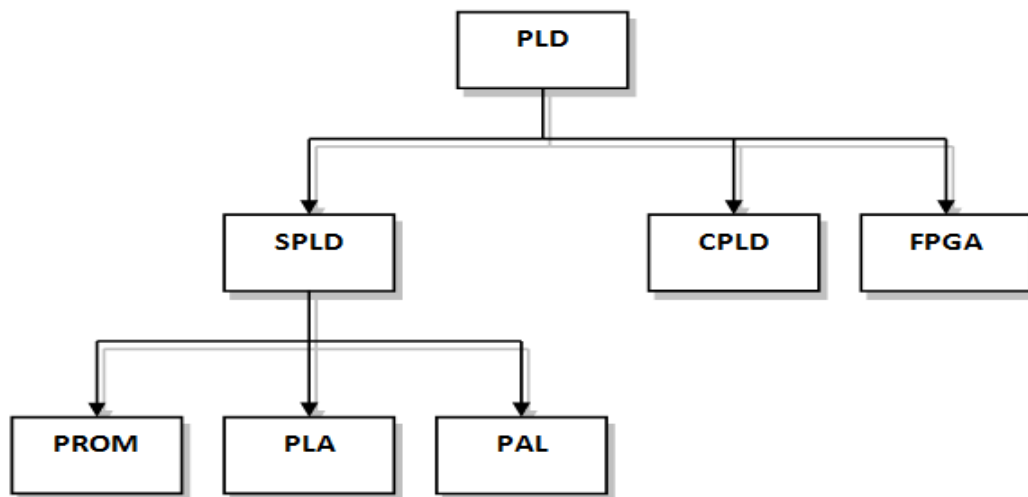


Figure 4.1 : Classification des circuits logiques programmables.

Les circuits FPGA (Field Programmable Gate Array) sont certainement les circuits programmables ayant le plus succès dans le domaine des circuits intégrés vu qu'ils sont entièrement reconfigurables par programmation qui permettent d'implanter physiquement, par simple description, n'importe quelle fonction logique.

4.2.1 Définition :

Les FPGA sont des circuits à architecture programmable qui ont été inventés par la société XILINX en 1985. Ils sont entièrement reconfigurables et ne demandent donc pas de fabrication spéciale en usine, ni de systèmes de développement coûteux, ceci permet de les reprogrammer à volonté afin d'accélérer notablement certaines phases de calculs. Un autre avantage de ces circuits est leur grande souplesse qui permet de

les réutiliser à volonté dans des algorithmes différents en un temps très court (quelques millisecondes).

Field Programmable Gate Array ou réseau de portes programmable par l'utilisateur ; c'est un ensemble de blocs logiques élémentaires que l'utilisateur peut interconnecter pour réaliser les fonctions logiques de son choix. On distingue deux types:

- FPGA à anti-fusible: programmables électriquement par l'utilisateur, non effaçables.
- FPGA à SRAM: ou encore LCA (Logic Cell Array) introduit en 1985 par Xilinx à base des cellules SRAM pour configurer les connexions entre les blocs logiques.

Les FPGA sont utilisés dans de nombreuses applications, on en cite dans ce qui suit quelques unes:

- Prototypage de nouveaux circuits ;
- Fabrication de composants spéciaux en petite série ;
- Adaptation aux besoins rencontrés lors de l'utilisation ;
- Systèmes de commande à temps réel ;
- DSP (Digital Signal Processor) ;
- Imagerie médicale[32].

4.2.2 Structure générale des FPGAs :

Les FPGAs à la différence des CPLDs sont programmables par l'utilisateur. La puissance de ces circuits est telle qu'ils peuvent être composés de plusieurs milliers jusqu'à des millions de portes logiques et de bascules. Ils sont composés de blocs logiques élémentaires (plusieurs milliers de portes) qui peuvent être interconnectés. Ces circuits sont bien distincts des autres familles tout en offrant le plus haut niveau d'intégration logique [32].

On appelle les FPGA parfois LCA (LOGIC Cell ARRAY) qui signifie : réseau de cellules logiques. Pour implémenter un système dans un FPGA il est indispensable de bien connaître sa structure interne et ses limites du point de vue performances. Un FPGA est un circuit intégré composé d'un réseau de blocs logiques programmables CLB (Configurable logic Bloc) et des blocs d'entrées/sorties IOB (Input Output Bloc) répartis régulièrement et

reliés entre eux grâce à des connexions qui forment une matrice de routage programmable.

Puisque tous les éléments sont programmables, le FPGA peut émuler et réaliser n'importe quel circuit à condition que celui-ci n'épuise pas les ressources logiques et de routage du FPGA [33]. La figure suivante illustre sa structure générale.

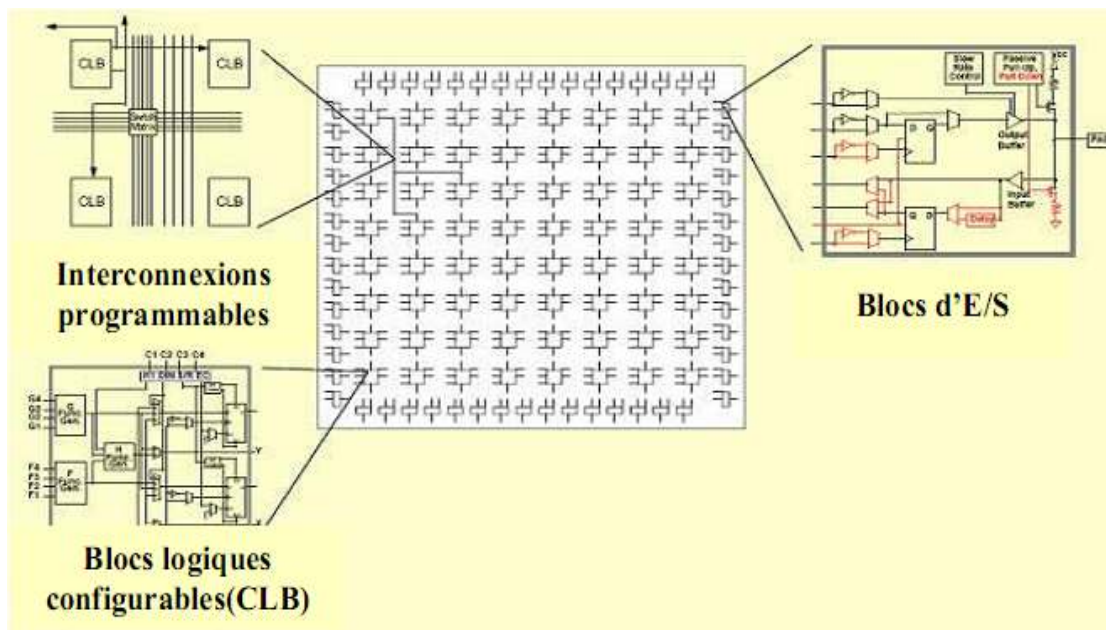


Figure 4.2 : Structure générale d'un FPGA.

La plupart des FPGA modernes sont basés sur des cellules SRAM (Static Random Access Memory) aussi bien pour le routage du circuit que pour les blocs logiques à interconnecter.

a Les blocs logiques configurables (CLB) :

Les blocs logiques configurables sont les éléments déterminants des performances du circuit FPGA (figure 4.3). Chaque CLB est un bloc de logique combinatoire composé de générateurs de fonction à quatre entrées (LUT) et d'un bloc de mémorisation/synchronisation composé de bascules D. Quatre autres entrées permettent d'effectuer les connexions internes entre les différents éléments du CLB. La structure du bloc logique change selon le fabricant et la famille. Aujourd'hui, la structure la plus utilisée est basée sur une Look-Up Table (RAM) pour implémenter une

fonction combinatoire plus une bascule D, la figure suivante montre un bloc logique de base.

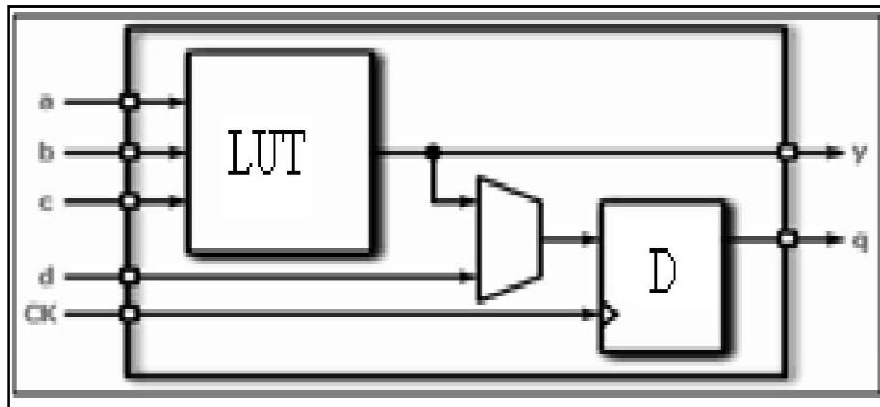


Figure 4.3 : Bloc logique d'un FPGA.

- La LUT (Look-Up Table) utilisée est une mémoire à base de SRAM, sa fonction est de stocker la table de vérité de la fonction à implémenter dans le bloc logique.

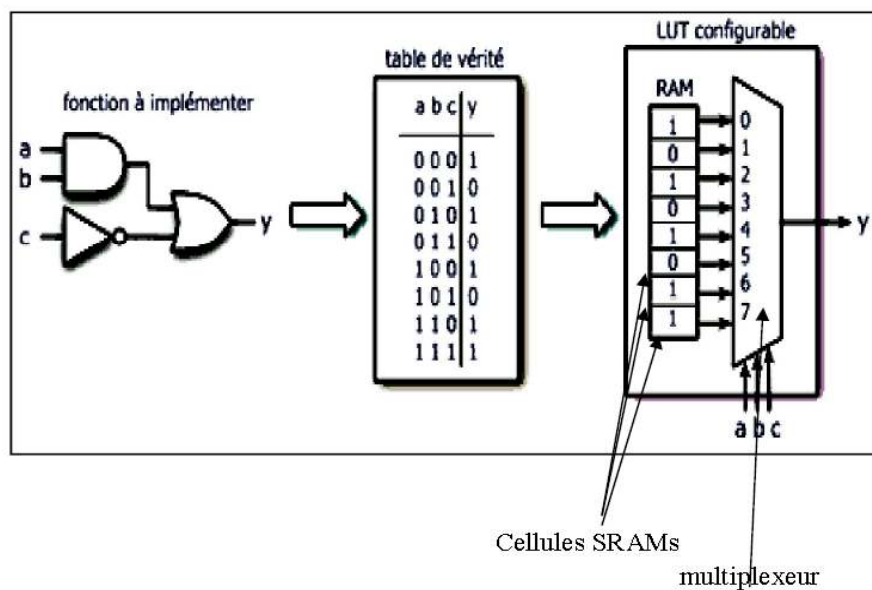


Figure 4.4 : Principe de travail d'un LUT.

b Les blocs d'entrées/sorties :

Les blocs d'entrée/sortie ou IOB (Input Output Bloc) permettent l'interface entre les broches du FPGA et la logique interne développée à l'intérieur (figure 4.5). Ils sont présents sur toute la périphérie du circuit FPGA. Chaque bloc IOB contrôle une broche du composant et il peut être défini en entrée, en sortie, en signal bidirectionnel ou être inutilisé (Ils ont une configuration programmable). [32]

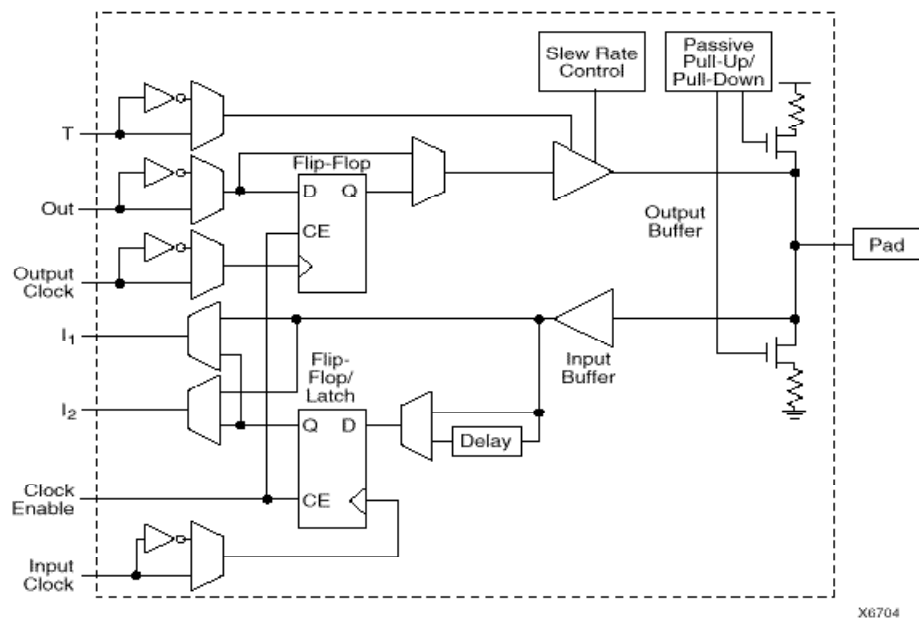


Figure 4.5: Bloc d'E/S.

Les IOB peuvent s'adapter à un grand nombre standards de communication: LVTTTL, LVCMOS, PCI, LVDS, LVPECL, etc. Ils sont regroupés en banques, et chaque banque a sa propre tension d'alimentation. Ceci permet de combiner des standards incompatibles entre eux sur le même circuit FPGA en utilisant des banques différentes [34]

c Les interconnexions :

Les blocs logiques configurables (CLB) et les blocs d'entrée/sortie (IOB) sont connectés entre eux par un réseau d'interconnexions programmable. Ces connexions internes sont composées de segments métallisés distribués horizontalement et verticalement entre les divers CLB. Parallèlement à ces lignes, nous trouvons des

matrices de routage programmables (matrices de commutation, switch matrix) réparties sur la totalité du circuit [31]. Elles permettent d'avoir des connexions entre les diverses lignes, celles-ci sont assurées par des transistors CMOS contrôlés par la configuration lors de la phase de placement-routage (selon la description) et ils sont commandés par des cellules SRAMs [32]. La figure suivante montre un réseau d'interconnexion.

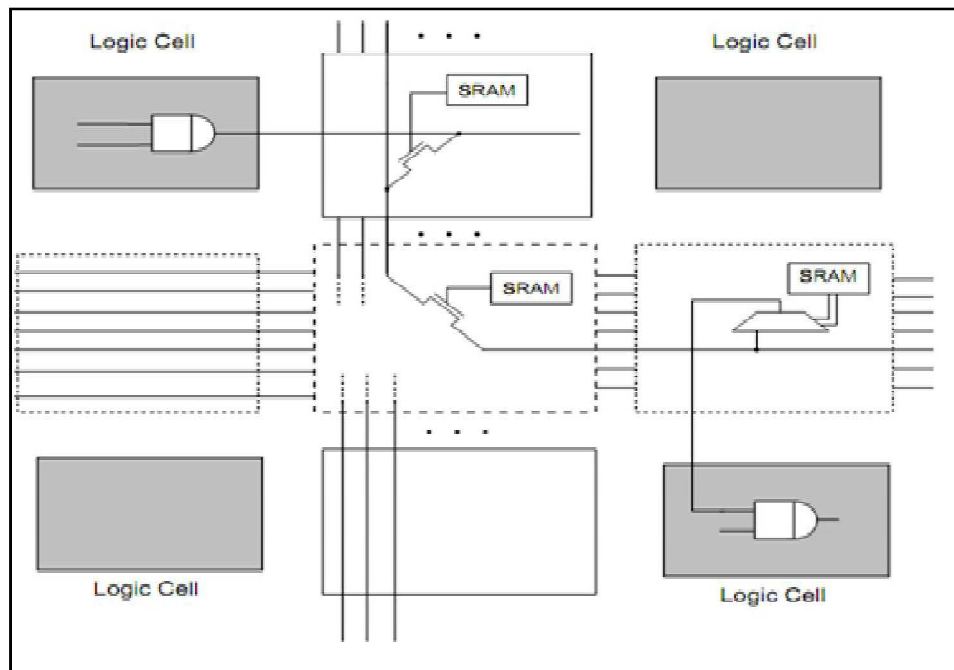


Figure 4.6 : Interconnexions commandés par les cellules SRAMs.

Le routage programmable à l'aide des matrices d'interconnexion permet la configuration à volonté du composant, mais occupe une place importante sur le silicium ce qui justifie le coût élevé des composants FPGA. Le rôle des interconnexions est de relier avec un maximum d'efficacité les blocs logiques et les blocs d'entrées/sorties afin que le taux d'utilisation dans un circuit donné soit le plus élevé possible.

Pour la conception sur FPGA, les deux langages de description les plus utilisés sont le Verilog et le VHDL. Tous deux font l'objet de standards IEEE afin d'uniformiser ces langages autant pour les concepteurs que les manufacturiers [31].

Le langage utilisé dans le cadre de ce projet est le VHDL et il sera présenté au cours de cette section.

4.3 Langage de description de matériel VHDL :

4.3.1 Définition :

Le VHDL est un langage de description matériel destiné à représenter le comportement ainsi que l'architecture d'un système électronique numérique. Son nom complet est: VHSIC Hardware Description Language (VHSIC: Very High Speed Integrated Circuit). L'intérêt d'une telle description réside dans son caractère exécutable : une spécification décrite en VHDL peut être vérifiée par simulation, avant que la conception détaillée ne soit terminée. En outre, les outils de conception assistée par ordinateur permettant de passer directement d'une description fonctionnelle en VHDL à un schéma en porte logique ont révolutionnés les méthodes de conception des circuits numériques comme les FPGA.

4.3.2 Logiciel Xilinx ISE 12.3 :

L'offre logicielle dans le domaine de conception des circuits numériques est très varié et l'un de ces environnements que nous allons exploiter au cours de ce travail est Xilinx ISE qui est un logiciel de création et de gestion de projet pour les circuits FPGA. C'est un logiciel multi taches qui possède dans son soft de différents outils permettant la création de système sous circuits numériques, l'introduction de projets de manière textuelle ou graphique en vue d'une intégration dans un circuit logique programmable.

Ce logiciel Xilinx ISE permet la simulation de la descriptions et la synthèse du circuit logique équivalent puis le placement et le routage du circuit sur un prototype correspondant a une technologie FPGA bien précise et enfin lorsque toutes les vérification sont faites vient l'implantation sur un FPGA réel ce qui correspond à

générer le fichier de configuration du circuit cible choisi afin d'établir les interconnexions des cellules logiques correspondantes au circuit logique conçu avec optimisation de ressources disponibles au niveau du circuit programmable FPGA.

D'une manière générale, le Xilinx ISE permet de réaliser toutes les étapes de conception et de programmation des circuits FPGA de Xilinx et même pour d'autres circuits programmables tel que les CPLD.

La conception de circuits sur Xilinx ISE met en œuvre 4 outils :

- **Un éditeur de texte ou entrée graphique** : Permet la description dans les logiciels CAO c'est-à-dire de dessiner ou de décrire le circuit avec une interface graphique ou textuelle.
- **Un simulateur** : La simulation du système est faite pour vérifier la validité du code avant-synthèse, après synthèse et placement –routage.
- **Un synthétiseur** : L'étape de synthèse et routage succéderons par la suite où la synthèse consiste à faire la transcription de la description d'une forme texte vers une graphique (RTL) à base de portes logiques.
- **Un placeur routeur** : Cette étape est une adaptation du circuit logique synthétisé sur les ressources disponibles dans le circuit FPGA ciblé. La figure 4.7 montre les différentes étapes pour réaliser un circuit sur FPGA. [34]

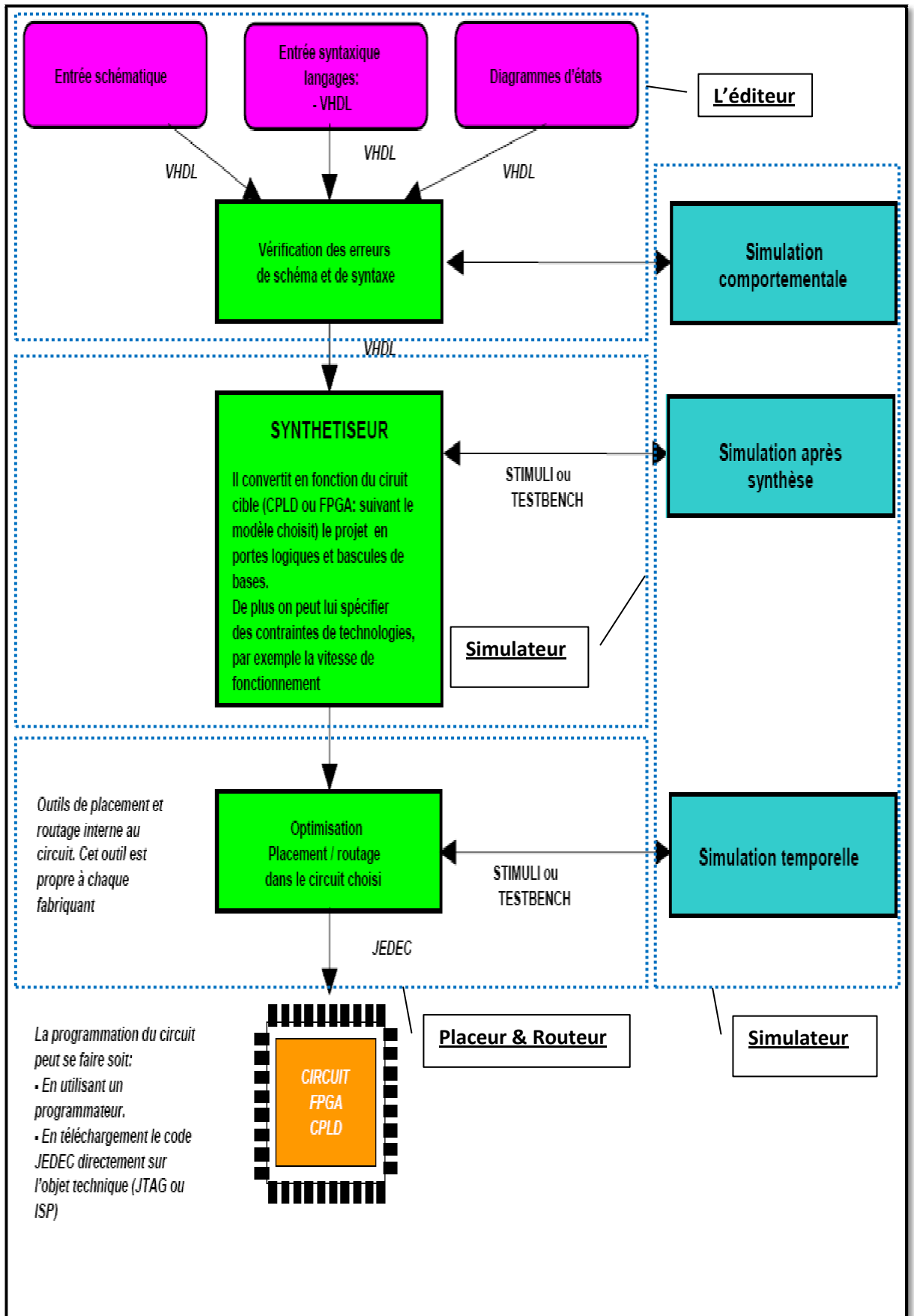


Figure 4.7: Flot de conception sur FPGA.

Pour le présent projet, la description VHDL sera faite d'une manière différente : par la bibliothèque Xilinx System Generator dans la bibliothèque Simulink /Matlab. Cette dernière génère le programme en langage VHDL relatif au model du « régulateur PID conçu dans le Xilinx System Generator /simulink.

4.3.3 Utilité du VHDL :

Le VHDL est un langage de spécification, de simulation et également de conception:

- Il est tout à fait possible de décrire un circuit en un VHDL standard pour qu'il soit lisible de tous;
- Il est également un langage de simulation. La possibilité de simuler avec des programmes VHDL devrait considérablement faciliter l'écriture de tests avant la programmation du circuit et éviter ainsi de nombreux essais sur un prototype qui sont beaucoup plus coûteux et dont les erreurs sont plus difficiles à trouver;
- Il permet la conception de circuits avec une grande quantité de portes. Il est compatible avec les circuits actuels qui comprennent, pour les FPGA par exemple, entre 500 et 100 000 portes et ce nombre augmente rapidement[31].

4.4 Plateforme de développement utilisée :

4.4.1 Logiciel utilisé :

Dans le cadre de ce projet, afin d'implémenter le régulateur PID pour contrôler le convertisseur conçu au chapitre précédent, le logiciel Xilinx System Generator (XSG) est utilisé.

Il s'agit d'un 'toolbox' développé par Xilinx pour être intégré dans l'environnement SIMULINK/MATLAB, il qui facilite la création des systèmes hautement parallèles pour FPGA. Les modèles créés sont affichés sous forme de blocs, et peuvent être raccordés aux autres blocs et autres 'toolbox' de SIMULINK/MATLAB. 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 doit 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. 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.

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 en suite 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 opérands est employée. L'outil XSG est utilisé afin de produire un modèle qui va tout de suite fonctionner sur le matériel une fois achevé et validé[34].

a .Définition du bloc System Generator(SG) :

Le bloc System Generator permet de contrôler les paramètres du système et la simulation, et est utilisé pour appeler le générateur de code. Tous les modèles SIMULINK contenant les blocs de XILINX doivent contenir au moins un bloc SG. Une fois ce dernier ajouté à un modèle, il est possible de spécifier la manière dont la génération de code et de simulation doivent être manipulée (figure 4.8)[35].



Figure. 4.8 : Le bloc SG.

B. La précision et les passerelles Xilinx :

SG fonctionne avec les modèles SIMULINK standard. Deux blocs appelés "Gateway In" et "Gateway Out" définissent la limite du FPGA du modèle simulé en SIMULINK. La passerelle dans le bloc convertit l'entrée à un nombre à virgule fixe [34].

définir la limite FPGA

- la limite est définie par FPGA de Xilinx "Gateway In" et "Gateway Out" de blocs.
- "Gateway In" bloc convertit la virgule flottante à point fixe
 - modes de saturation et arrondissement sont définis
- le "Gateway Out" bloc convertit le FPGA sorties **bloc** à la double précision

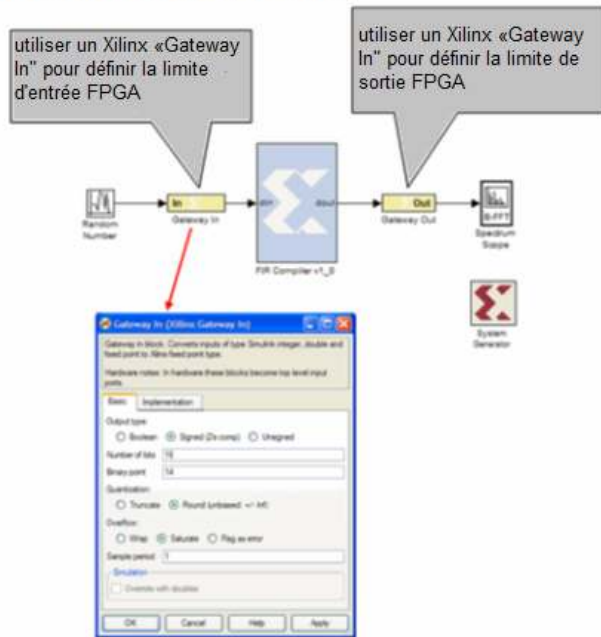


Figure 4.9 : La limite FPGA.

4.4.1 Le choix du circuit FPGA :

Actuellement, XILINX, le premier fabricant de FPGA, propose principalement deux familles, Virtex et Spartan, toutes deux de type SRAM. Les deux familles contiennent des multiplieurs des blocs de RAM, pouvant être utilisés en single ou dual port. La figure 3.10 présente l'architecture générale de la série Spartan-3E.

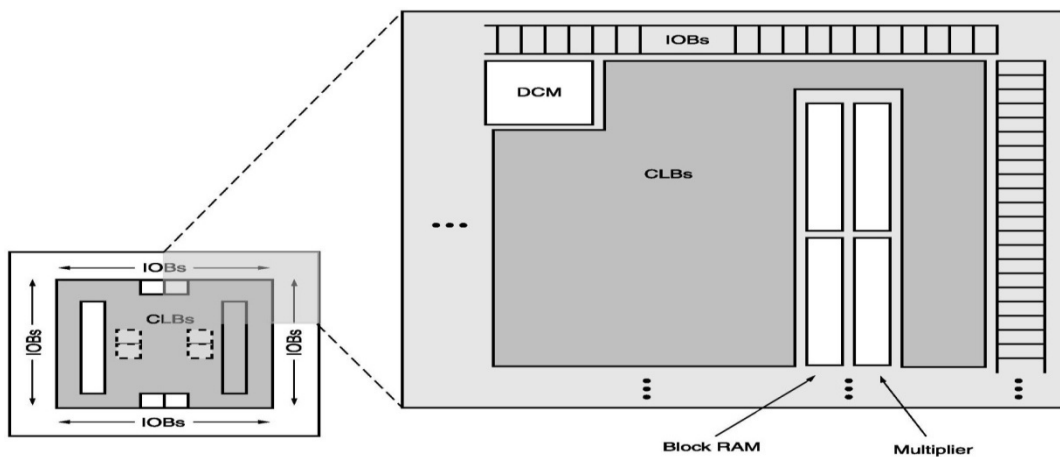


Figure 4.10 : L'architecture générale de la série Spartan-3E.

- Le bloc DCM (Digital Clock Manager Blocks) offre un auto-calibrage, temporisation, multiplication, division et décalage de phase des signaux d'horloge.
- Les IOBs supportent un échange de données bidirectionnelles et la logique 3 états. Les registres DDR (Double Data-Rate) sont inclus.
- Les blocs de RAM fournissent un stockage de données sous forme de blocs 18-Kbit avec un accès double.
- Les blocs de multiplicateurs acceptent deux nombres binaires de 18 bit comme entrées et assurent le calcul du produit.

Le Virtex-II offre la même architecture avec des technologies nouvelles. La figure 4.11 présente l'architecture générale de la série Virtex-II. Les Virtex-4 et Virtex-5 représentent les deux dernières versions des FPGA de Xilinx.

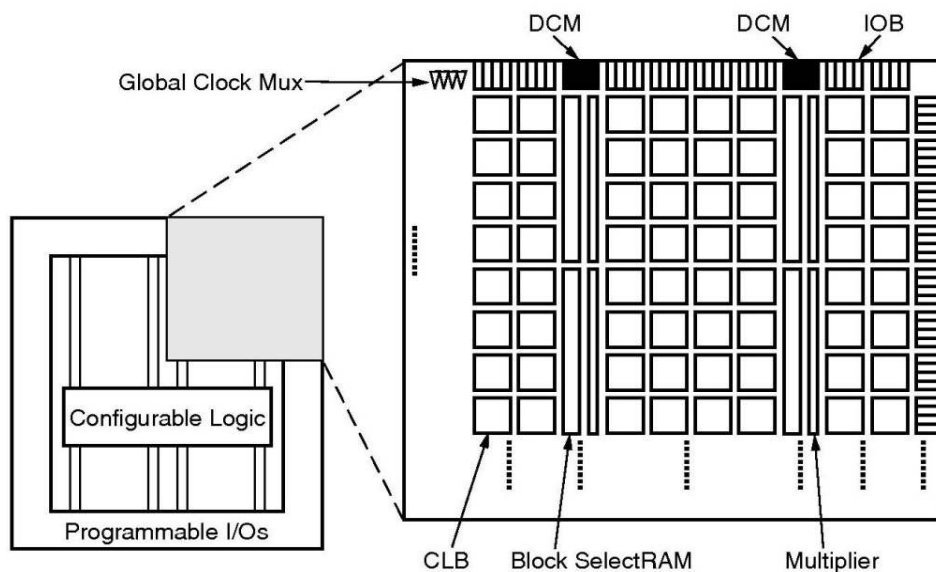


Figure 4.11 : L'architecture générale de la série Virtex-II

Les Virtex-4 sont fabriqués avec une technologie de 90 nm et contiennent des blocs multiplieurs accumulateurs (MAC) de 18×18 bits et des blocs de RAM de 18 Kbits. Cette série possède jusqu'à 512 "500 Mhz DSP48 Slice".

La technologie de 65 nm est employée pour la fabrication des FPGA de la série Virtex-5. Ces derniers contiennent des blocs multiplieurs accumulateurs (MAC) de 25×18bits, des blocs de RAM de 36 Kbits et jusqu'à 640 "550 Mhz DSP48E Slice". La figure4.12 illustre l'architecture du "Virtex-5 SXT DSP48E Slice". [31]

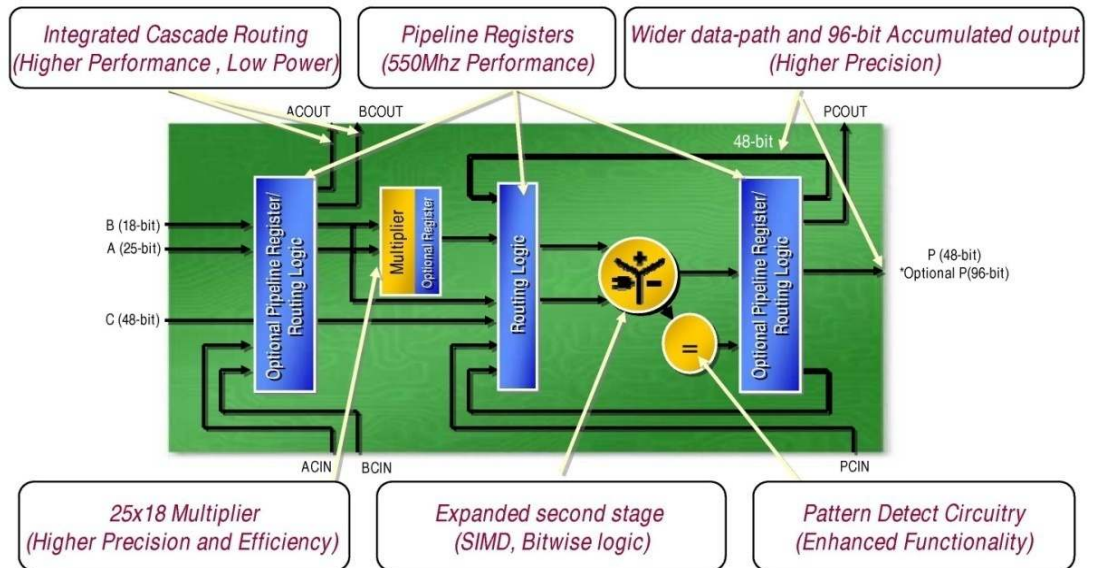


Figure 4.12 : L'architecture du "Virtex-5 SXT DSP48E Slice"

4.5 La description XSG du régulateur PID :

Le système générateur de XILINX fournit un ensemble de blocs de SIMULINK (modèles) pour plusieurs opérations de matériel qui pourraient être mises en application sur divers XILINX FPGAs. Ces blocs peuvent être employés pour simuler la fonctionnalité du système de matériel utilisé dans l'environnement de SIMULINK.

L'objectif à ce stade est d'effectuer des simulations aboutissant à des résultats comparables à ceux obtenus sous SIMULINK. Cette approche a été retenue afin de réduire le temps de conception sur le FPGA.

La modélisation XSG du régulateur PID était basée sur les équations décrivant son comportement (équation : 2.8) du deuxième chapitre. L'architecture adoptée est présentée dans la figure 4. 13.

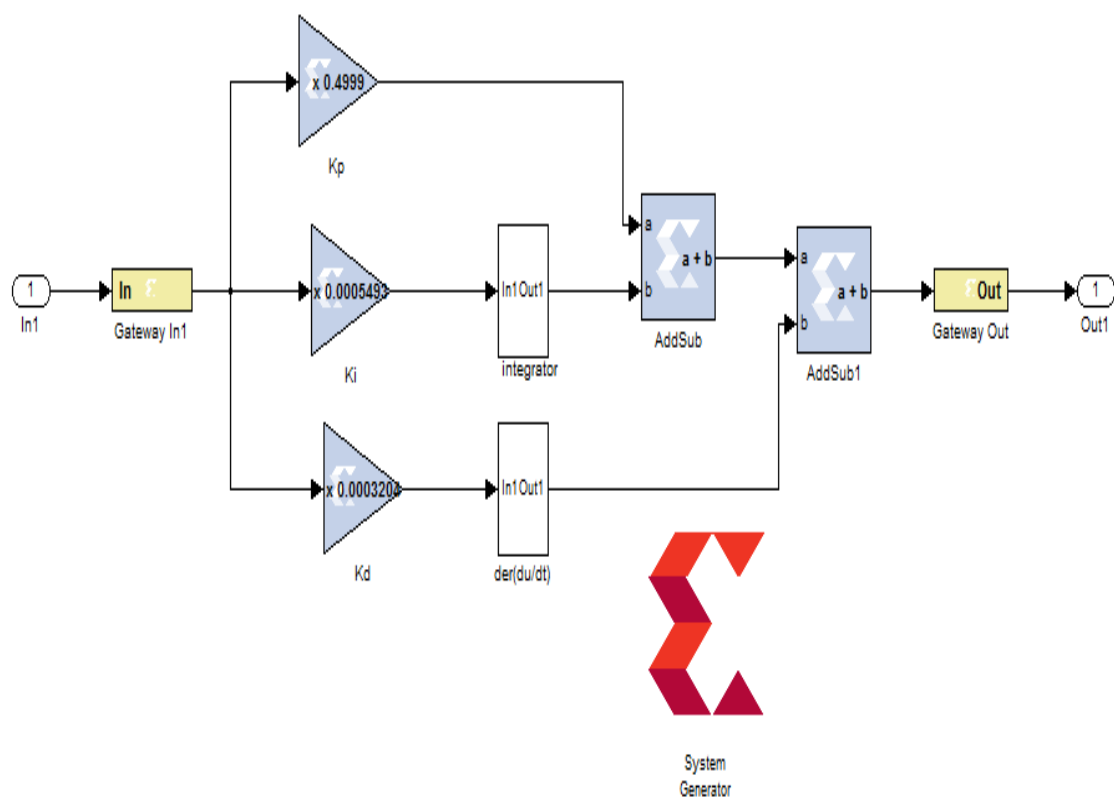


Figure 4.13 : Modèle du « régulateur PID » sous XSG.

Tandis que les deux fonctions : intégrateur et dérivateur sont des fonctions principales dans notre système qui ne sont pas disponibles dans la bibliothèque Xilinx Blockset. Par conséquent, elle sera mise en œuvre à l'aide d'autres éléments blocs comme il est montré dans la figure 4.13 et par figure 4.14 la suite en utilisant des blocs de XILINX .

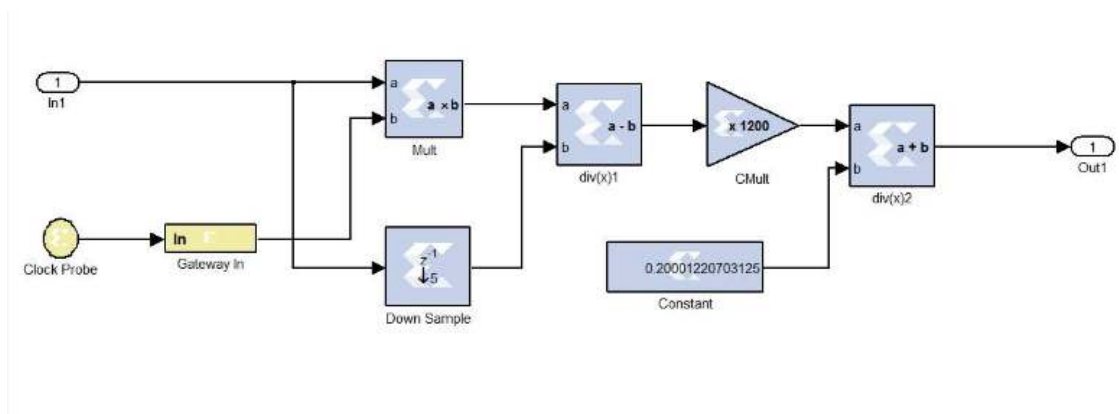


Figure 4.14 : Architecture détaillée du bloc Dérivé (du /dt)

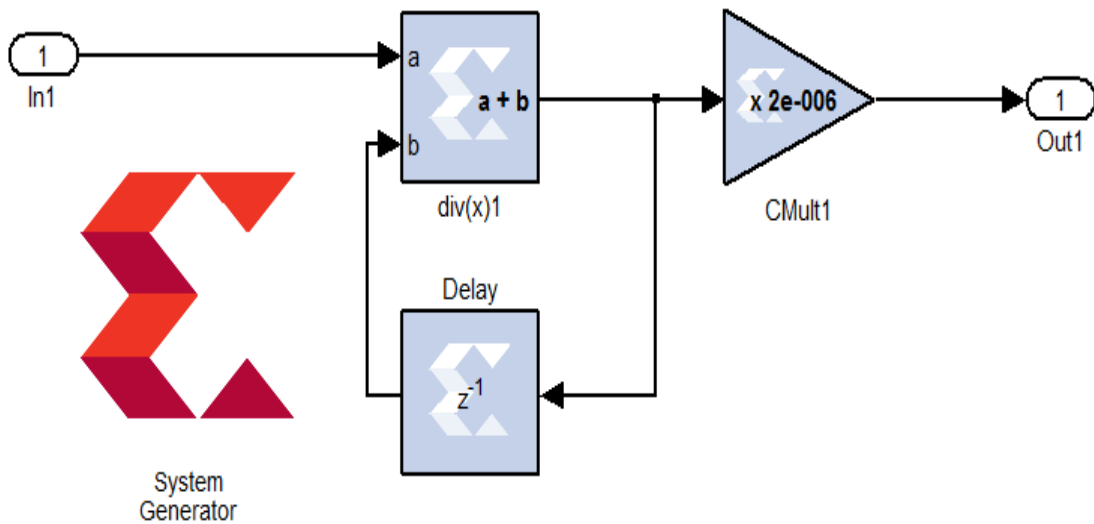


Figure 4.15: Le bloc intégrateur en utilisant des éléments de la bibliothèque Xilinx.

4.6 Résultats et interprétations :

Après une description XSG du régulateur PID, la simulation de ce dernier et les résultats obtenus sont comparés avec les résultats de simulations donnés par MATHLAB/SIMULINK comme le montre la figure suivante :

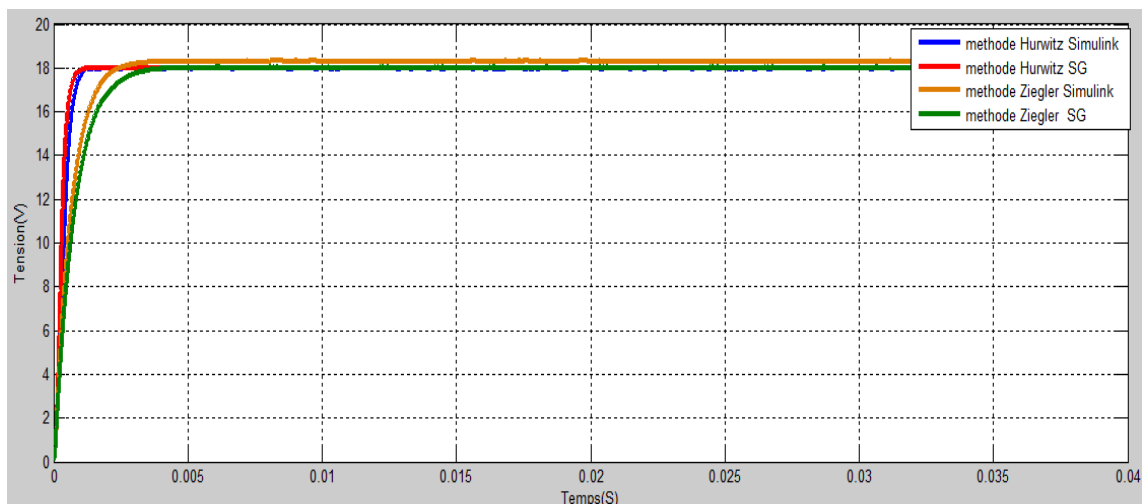


Figure 4.17: Tension à la sortie du convertisseur contrôlé par un «régulateur PID » sous XSG et SIMULINK.

L'analyse des tensions de sortie du convertisseur contrôlé par un régulateur PID basé sur les deux méthodes cités ci-dessus montre une similitude des résultats obtenus par les deux environnements: Simulink/ Matlab et XSG . Le suivi de la consigne pour chaque cas est parfaitement réalisé. Nous constatons un meilleur temps de montée

pour le cas de la méthode de Hurwitz. Les conclusions tirées dans le chapitre précédent sont aussi valables dans ce cas.

Après la simulation sous le XSG la Co-simulation et l'implémentation sur FPGA sera aisée. En commençant par choisir une cible de compilation comme indiqué par la boîte de dialogue du bloc System Generator comme le montre la figure suivante :

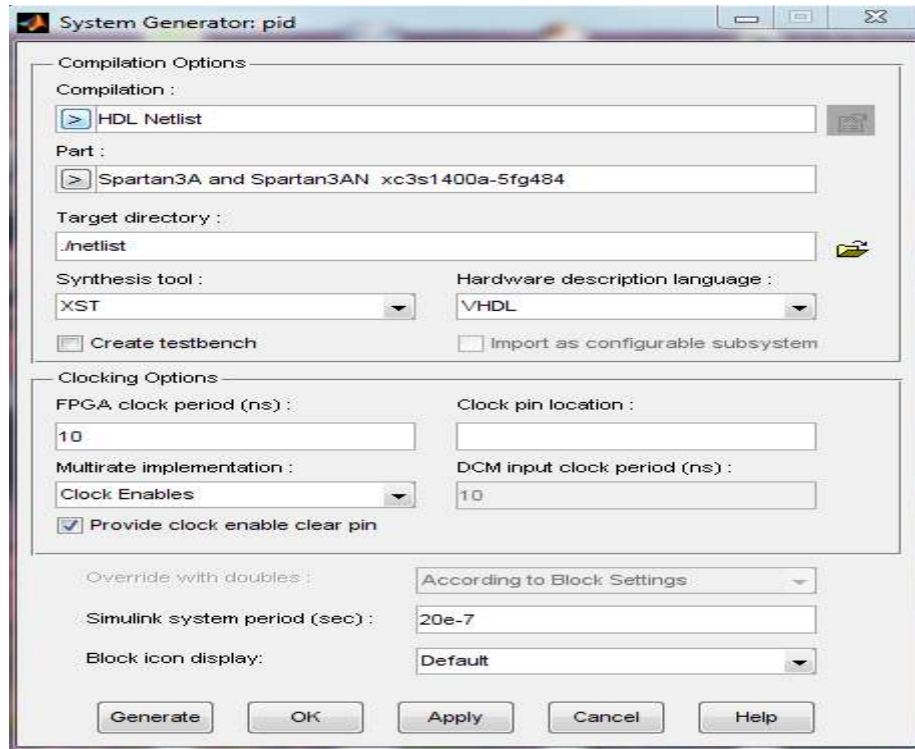


Figure 4.18 : Boite de dialogue de bloc System Generator.

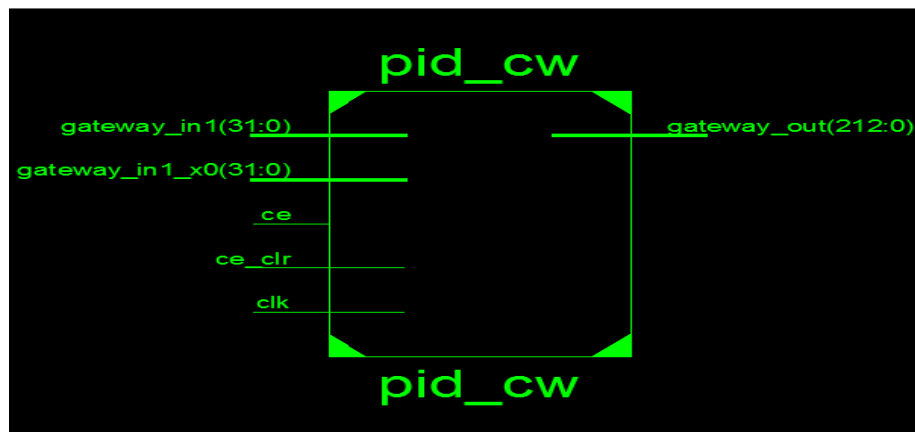


Figure 4.18 : Schématique du modèle du PID



Figure 4.16. Architecture du Modèle de PID sur circuit FPGA.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	123	22,528	1%	
Number of 4 input LUTs	1,229	22,528	5%	
Number of occupied Slices	836	11,264	7%	
Number of Slices containing only related logic	836	836	100%	
Number of Slices containing unrelated logic	0	836	0%	
Total Number of 4 input LUTs	1,557	22,528	6%	
Number used as logic	1,229			
Number used as a route-thru	328			
Number of bonded IOBs	279	375	74%	
Number of BUFGMUXs	1	24	4%	
Number of MULT18X18SIOs	9	32	28%	
Average Fanout of Non-Clock Nets	1.73			

Tableau 4.1 : Utilisation de périphériques sommaire (valeurs estimées)

4.7 Conclusion :

Dans ce chapitre, la description et la simulation sous XSG du «PID » est élaborée. La comparaison des résultats obtenus avec ceux obtenus par SIMULINK/MATLAB montrent que la description SGX donne de bons résultats et son implémentation sur FPGA devient très aisée.

1.1 Introduction

Les hacheurs sont les convertisseurs statiques continu-continu permettant de fabriquer une source de tension continue variable à partir d'une source de tension continue fixe. Ce chapitre est consacré au rappel sur les différents types de convertisseurs, à la conception plus précisément d'un convertisseur DC-DC de type « Buck » et à sa modélisation.

1.2 Les convertisseurs DC-DC

Les convertisseurs de puissance sont organisés en grandes familles fondées sur des principes de conversion : continu-continu Figure 1.1[1], continu-alternatif (onduleurs), alternatif-continu (redresseurs) et alternatif-alternatif . Dans ce chapitre, nous nous intéressons à la famille des convertisseurs continu-continu car elle représente en quelque sorte la base des topologies connues en électronique de puissance (abaisseur, élévateur, inverseur, flyback , forward, etc.) et des phénomènes physiques essentiels à la compréhension de tous les autres convertisseurs (notion de base de cellule de commutation). Les convertisseurs continu-continu ou commercialement appelés DC-DC constituent une partie essentielle des alimentations de puissance. En effet, on trouve ces convertisseurs dans de nombreuses applications à différents niveaux de puissance.

Par exemple, à des faibles puissances (quelques watts) dans des chargeurs et alimentations de téléphones portables, à des puissances moyennes (quelques centaines de watts) dans des alimentations industrielles, et aussi à des fortes puissances (quelques KW) pour des applications de soudage électrique par exemple. Les convertisseurs DC-DC ont donc un rôle très important dans les systèmes de

conversion d'énergie. Pour ce type de fonction, plusieurs topologies de circuits sont proposées. Nous pouvons les classer généralement par les topologies simples, qui sont les convertisseurs DC-DC de second ordre du point de vue de la modélisation mathématique, et les topologies complexes, qui sont ceux d'ordre plus élevé. Pour les topologies simples nous citons les convertisseurs de second ordre de type Buck (abaisseur), Boost (élevateur) et Buck-Boost (inverseur). Le mode Buck est employé pour réduire la tension de sortie, tandis que le mode Boost augmente la tension de sortie. Dans le mode Buck-Boost, la tension de sortie peut être maintenue supérieure ou inférieure que la tension d'entrée mais dans la polarité opposée [2]. Le tableau 1.1 représente ces différents types de convertisseurs DC-DC. Pour ceux d'ordre élevé, nous citons les convertisseurs Cuk et SEPIC (de l'anglais : Single-Ended Primary Inductor Converter) de quatrième ordre. Du point de vue de l'électronique de puissance, chacun de ces convertisseurs a des avantages et des inconvénients. Les schémas des différents convertisseurs DC-DC cités sont donnés sur les Figure. 1.2 à 1.6 [3] [4] [5]. Parmi les convertisseurs DC-DC de base, le SEPIC présente beaucoup d'avantages et très peu d'inconvénients par rapport aux autres. En effet, cette topologie SEPIC présente plusieurs éléments intéressants. Le premier et le plus important, est le fonctionnement abaisseur-élevateur avec une tension de sortie ayant la même polarité que la tension d'entrée. Ensuite, la commande de l'interrupteur est référencée par rapport à la masse, ce qui simplifie l'électronique de commande rapprochée du MOSFET (interrupteur).

Les convertisseurs DC/DC sont apparus dans les années soixante dix, cependant leur utilisation massive dans les alimentations à découpage n'a pris son essor que depuis une vingtaine d'années. En effet depuis longtemps les alimentations classiques restaient préférées pour la facilité de la mise au point de leur étage de régulation. Malgré leurs deux grands défauts, le gros volume du transformateur abaisseur de la tension AC du secteur qu'ils exigent ainsi que les très grandes pertes dissipées dans le régulateur et dans l'étage ballast, les alimentations classiques ont prospéré jusqu'aux années 90 tant que le courant ne dépassait pas une dizaine d'ampères. Cependant, au moment où le besoin de miniaturisation des dispositifs électroniques s'est fait sentir, les difficultés de contrôle des convertisseurs DC/DC abaisseurs de tension (convertisseur Buck, convertisseur Flyback, etc..) sont devenus obsolètes car

avec le temps sont apparus des circuits intégrés pour la régulation de courant et de tension qui parfois sont tellement miniatures qu'ils sont intégrés dans le transistor de puissance (exemple du circuit KA5H0385R) du convertisseur. En effet en plus des faibles pertes que présentent les convertisseurs DC/DC, ceux-ci n'exigent pas la présence d'un transformateur abaisseur de la tension AC de gros volume et donc la tension AC du secteur est directement redressée et filtrée avant d'être présentée au convertisseur DC/DC. En outre, l'utilisation de fréquence de commutation de plus en plus élevée permettra l'utilisation de bobines (convertisseur Buck) et de transformateur (pour les convertisseurs Flyback) de dimension très réduite.

Les convertisseurs DC/DC en général réalisent deux fonctions : modifier le niveau de tension (élever ou abaisser) et réguler la tension. L'électronique de conversion se trouve à l'interface entre les batteries (la source d'énergie en général) et l'ensemble des blocs constitutifs du système considéré, des circuits électroniques numériques et analogiques, des écrans, des actionneurs, des claviers, etc.

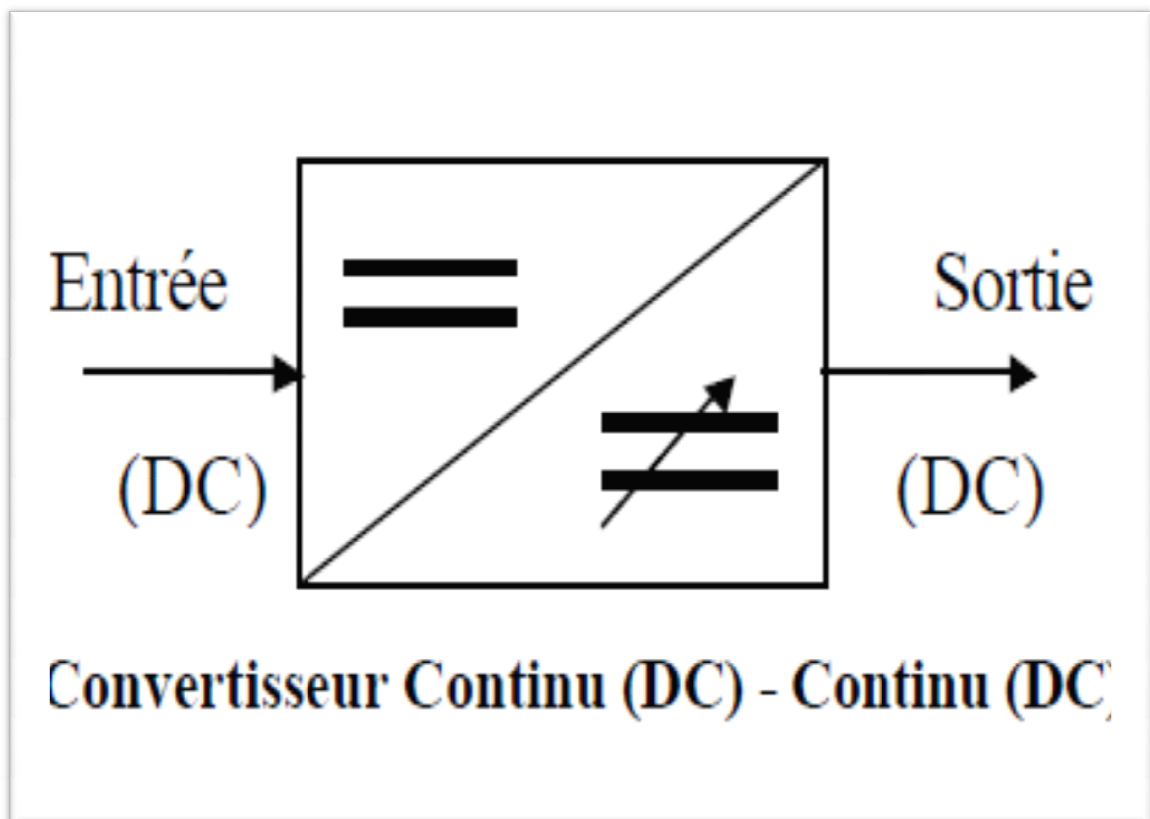


Figure 1.1 : Convertisseur continu-continu.

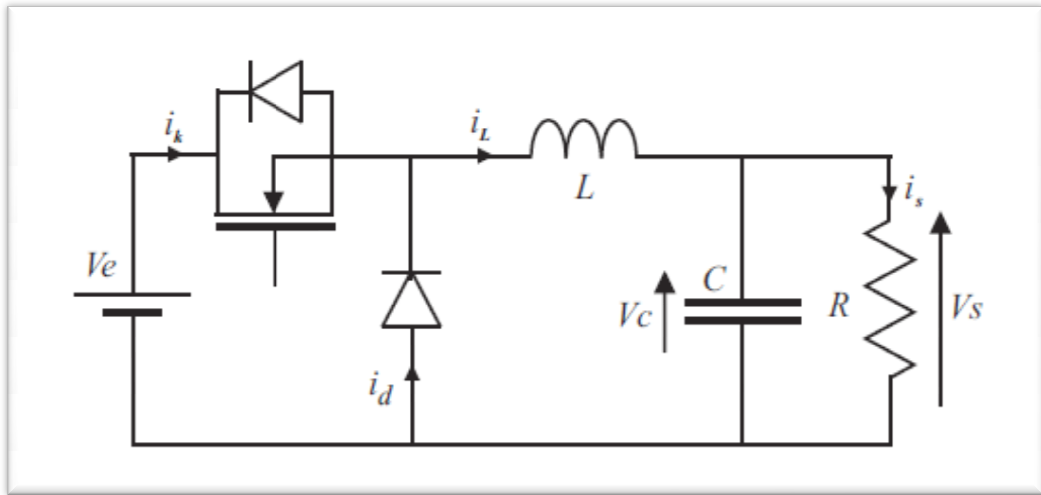


Figure 1.2 : Convertisseur Buck.

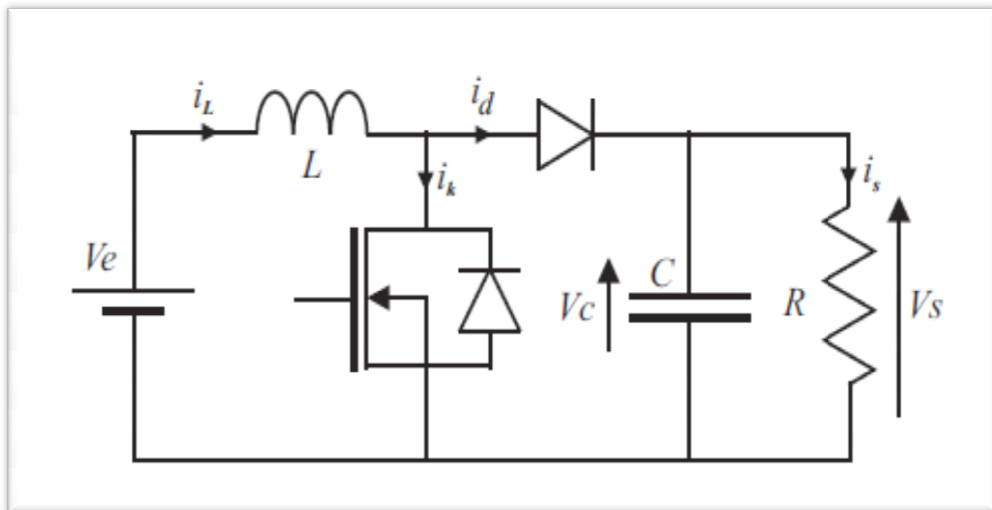


Figure 1.3 : Convertisseur Boost.

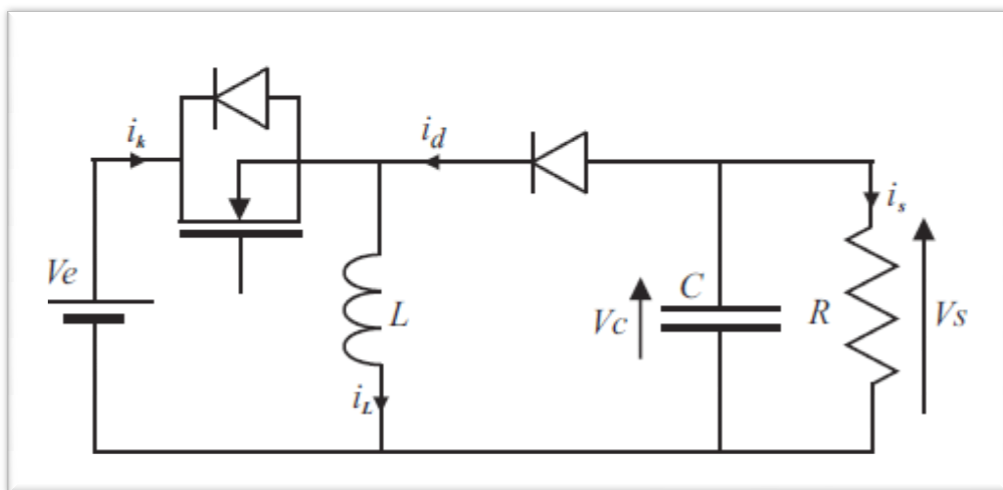


Figure 1.4 : Convertisseur Buck-Boost.

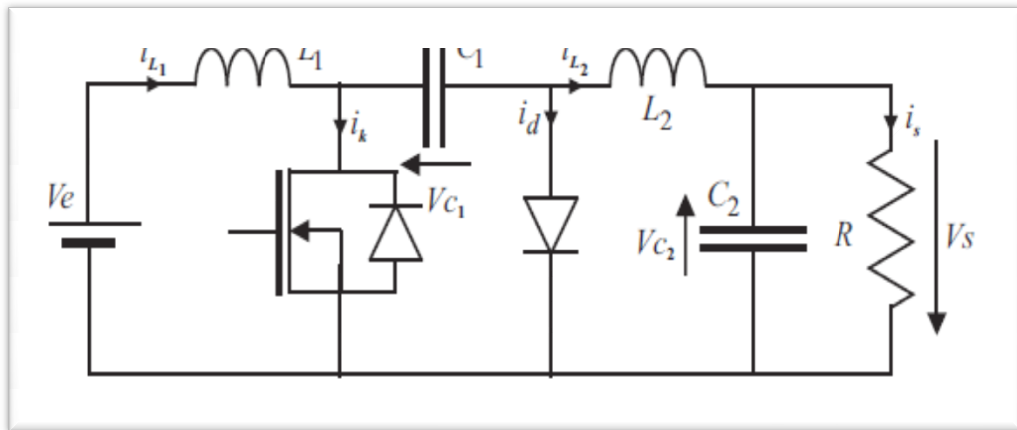


Figure 1.5 : Convertisseur cük.

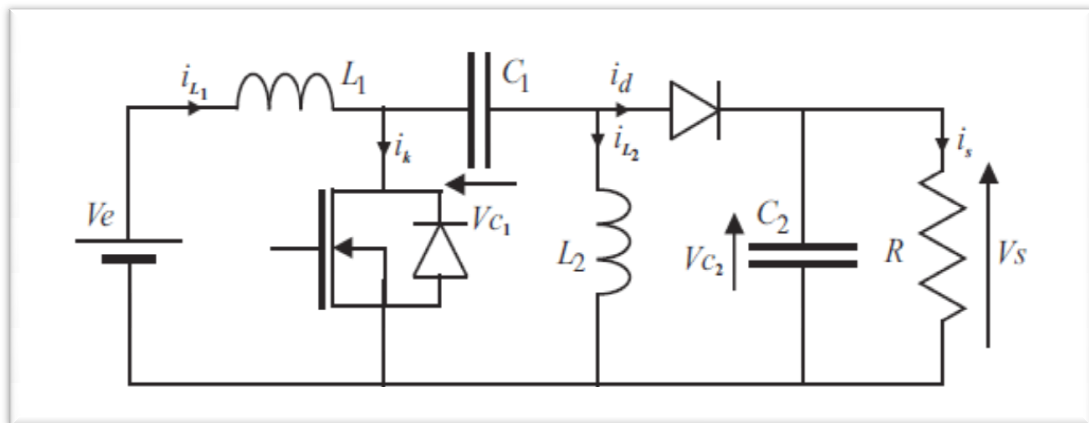


Figure 1.6 : Convertisseur SEPIC.

Type	Fonction	Rapport de transformation $M(\alpha)$
Série (Buck)	Abaisseur	α
Parallèle (Boost)	Élévateur	$\frac{1}{1 - \alpha}$
Série-parallèle (Buck-boost)	Abaisseur Élévateur	$\frac{\alpha}{1 - \alpha}$

Tableau 1.1: Différents convertisseurs DC-DC utilisés dans les applications solaires.

Dans notre étude, nous nous intéressons uniquement aux convertisseurs de types «Buck». On donne ici son étude avec un peu de détail.

1.3 La conception du convertisseur DC-DC Buck

Le convertisseur Buck est constitué d'une source de tension continue V_i , un commutateur K (qu'on peut contrôler), une diode D, une inductance L, un condensateur C et une charge résistive R. Son fonctionnement peut être divisé en deux configurations suivant l'état du commutateur K:

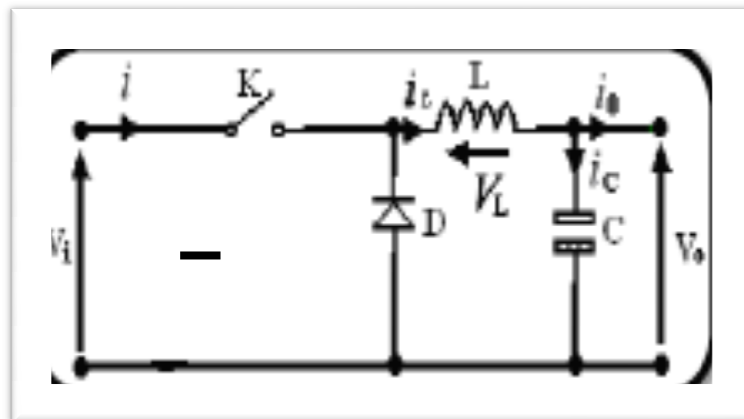


Figure 1.7 : Circuit idéal du Buck.

- Dans l'état passant (figure 1.7(a))

Le commutateur K est fermé pendant un temps t_{on} , le courant traversant l'inductance augmente linéairement et le condensateur se charge et maintient la tension entre ses

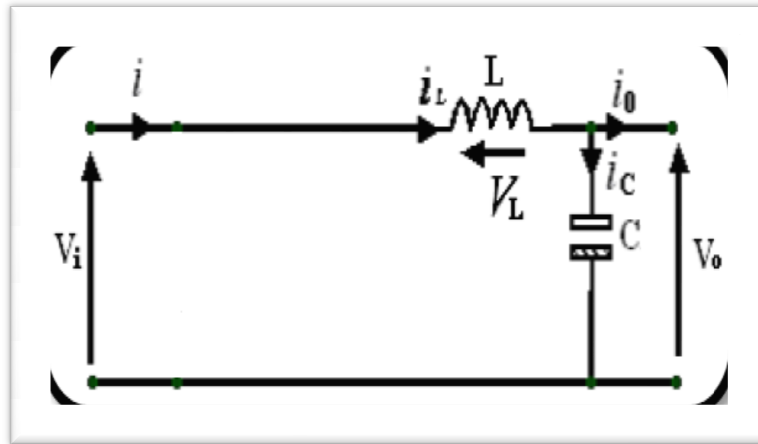


Figure 1.7(a) : Schéma équivalent du hacheur série quand $t \in [0, t_{on}]$

bornes jusqu'à l'ouverture du commutateur K. La tension aux bornes de la diode étant négative, aucun courant ne la traverse.

L'application des lois de Kirchhoff sur le circuit équivalent pour cette phase donne:

$$V_L(t) = L \cdot \frac{di_L}{dt} = V_i(t) - V_o(t) \quad (1-1)$$

$$I_c(t) = C \cdot \frac{dV_o}{dt} = I_i(t) - I_o(t) \quad (1-2)$$

- Dans l'état bloqué (figure 1.7(b))

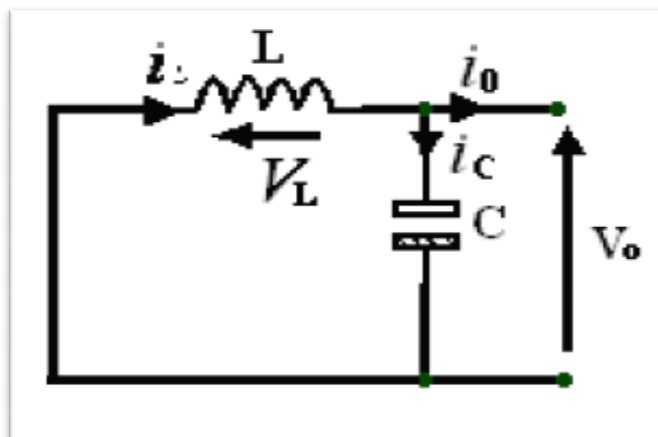


Figure 1.7(b) : Schéma équivalent du hacheur série quand $t \in [t_{on}, T_s]$.

Le commutateur est ouvert pendant un temps t_{off} . La diode devient passante afin d'assurer la continuité du courant dans l'inductance. Le courant traversant l'inductance décroît et le condensateur se décharge à travers la diode sur la charge.

$$V_L(t) = -V_o(t) \quad (1-3)$$

$$I_c(t) = C \cdot \frac{dV_o}{dt} = I_L(t) - I_o(t) \quad (1-4)$$

Lors de son fonctionnement, le commutateur est commandé par un signal PWM (Pulse Width Modulation), à une fréquence f_s fixe avec un temps de fermeture égal à $(\alpha \cdot T_s)$ et un temps d'ouverture égal à $((1 - \alpha) \cdot T_s)$, α est le rapport cyclique variable ($\alpha \in [0,1]$) et T_s est la période de commutation.

Quand un convertisseur Buck travaille en mode de conduction continue où $T_s = t_{on} + t_{off}$, le courant I_L traversant l'inductance ne s'annule jamais. La figure (1.8(a)) et (1.8(b)) montre les formes d'ondes de la tension et du courant dans un convertisseur Buck.

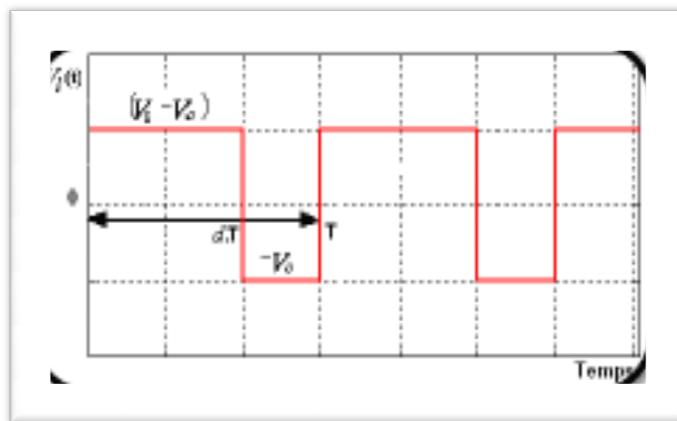


Figure 1.8(a) : La tension dans la self.

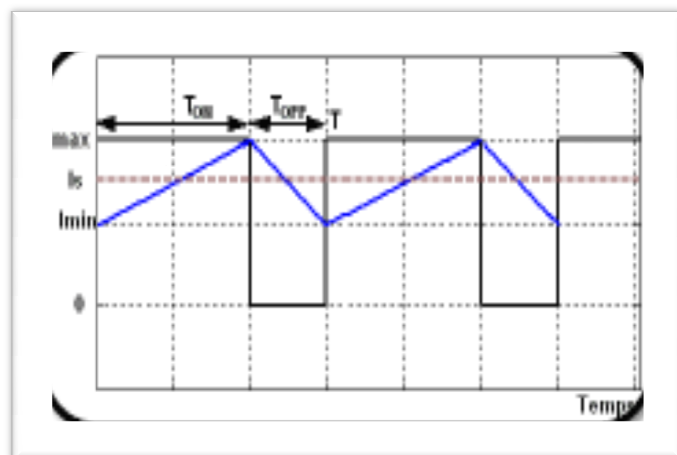


Figure 1.8(b) : Le courant circulant la self.

Le taux d'accroissement de I_L est donné par:

$$V_L = L \cdot \frac{dI_L}{dt} \quad (1-5)$$

Par conséquent, l'augmentation de I_L durant l'état passant est donnée par:

$$\Delta I_{L\text{on}} = \int_0^{\alpha.Ts} dI_L = \int_0^{\alpha.Ts} \frac{V_L}{L} dt = \frac{(V_i - V_o) \cdot \alpha \cdot Ts}{L} \quad (1-6)$$

De même, la baisse du courant dans l'inductance durant l'état bloqué est donnée par:

$$\Delta I_{L\text{off}} = \int_{\alpha.Ts}^{Ts} dI_L = \int_{\alpha.Ts}^{Ts} \frac{V_L}{L} dt = \frac{-V_o \cdot (1 - \alpha) \cdot Ts}{L} \quad (1-7)$$

Si on considère que le convertisseur est en régime permanent, l'énergie stockée dans chaque composant est la même au début et à la fin de chaque cycle de commutation. En conséquence, le courant I_L traversant l'inductance est le même au début et à la fin de chaque cycle de commutation. Ce qui peut s'écrire de la façon suivante:

$$\Delta I_{L\text{on}} + \Delta I_{L\text{off}} = 0 \quad (1-8)$$

En remplaçant $\Delta I_{L\text{on}}$ et $\Delta I_{L\text{off}}$ par leur expression, on obtient:

$$\frac{(V_i - V_o) \cdot \alpha \cdot Ts}{L} + \frac{-V_o \cdot (1 - \alpha) \cdot Ts}{L} = 0 \quad (1-9)$$

Ce qui peut se réécrire de la façon suivante:

$$V_o = \alpha \cdot V_i \quad (1-10)$$

Grâce à cette équation, on peut voir que la tension de sortie varie linéairement avec le rapport cyclique. Le rapport cyclique étant compris entre 0 et 1, la tension de sortie V_o est toujours inférieure à celle d'entrée. C'est pour cela que l'on parle d'un *dévolteur*. [6]

1.4 Dimensionnement de L et de C

Les équations (1-7), (1-8), et (1-9) ont été retirées après avoir considéré que la variation des deux paramètres: L et C est linéaire, c'est-à-dire approximer l'exponentielle à un segment et sa dérivé sera constante ($a \cdot e^x \cong a + x; x \lll 1$).

Donc, les deux intervalles: $(\alpha.T_s)$ et $((1-\alpha).T_s)$ doivent être très faibles devant les constantes de temps du circuit: L/R_L et RC relativement. Dans ce cas la forme exponentielle du courant qui parcourt l'inductance et de la tension aux bornes du condensateur pourra prendre la forme linéaire (figure.1.7(b)).

Ce sont les critères prises en considération dans la conception du circuit et le dimensionnement des composants ainsi, dont le but de diminuer les ondulations du courant et de la tension sans faire un surdimensionnement ce qui accroîtrait le prix du circuit, un calcul de ses composants en fonction des ondulations voulues est nécessaire.

Nous avons vu que le courant I_L dans l'inductance L est constitué d'une valeur moyenne $\langle I_L \rangle$, et comporte une ondulation \tilde{I}_L autour de cette valeur moyenne. [7]

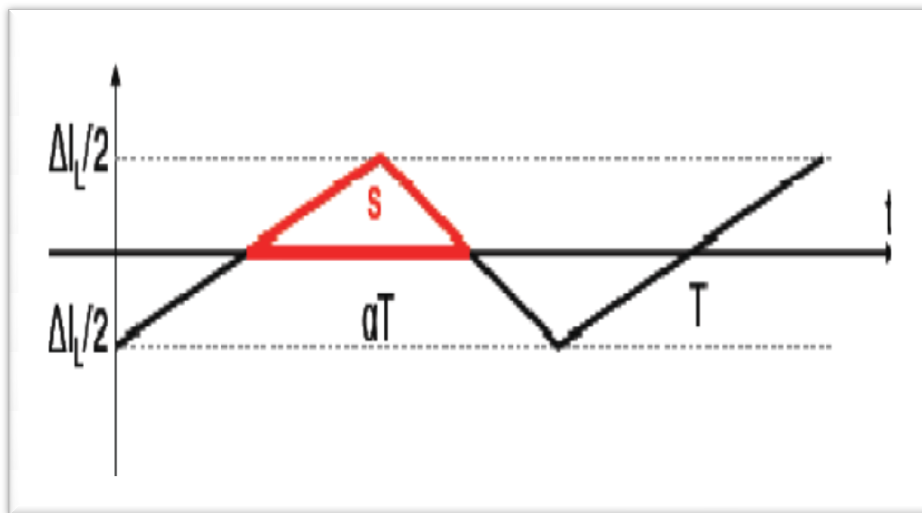


Figure 1.9: Forme de l'ondulation du courant dans l'inductance.

Compte tenu des hypothèses de travail adoptées, le courant moyen $\langle I_L \rangle$ est fixé par la valeur du courant I_o dans la charge R_S :

$$I_o = \langle I_L \rangle = \frac{V_o}{R} = \frac{\alpha.V_i}{R} \quad (1.11)$$

Le courant moyen de sortie est ainsi fixé par:

- La valeur de la source de tension continue,
- Le rapport cyclique,
- La résistance de charge.

En outre, ce courant est régi par la relation :

$$I_L = I_c + I_o \quad (1.12)$$

Tel que la valeur moyenne du courant I_L est fixée par le courant I_o dans la charge résistive R_s , alors que l'ondulation \tilde{I}_L est absorbée par le condensateur ce qui permet de déduire:

$$\langle I_L \rangle = I_o \quad (1.13)$$

$$\tilde{I}_L = I_c \quad (1.14)$$

L'analyse d'une seule des phases de conduction décrites précédemment suffit pour déterminer l'amplitude des ondulations ΔI_L . L'analyse de la phase relative à l'état passant du transistor K, et d'après les relations (1-6) et (1-10) permet d'écrire:

$$\Delta I_L = \frac{(1-\alpha)\alpha \cdot T_s \cdot V_i}{4L} \quad (1.15)$$

$$\Rightarrow L \geq \frac{(1-\alpha)\alpha \cdot V_i}{4f_s \cdot \Delta I_{Lmax}} \quad (1.16)$$

Les paramètres V_i , L et f_s sont des paramètres constants. En revanche, le rapport cyclique est susceptible de varier suivant le point de fonctionnement recherché: il est en effet ajusté pour obtenir les valeurs de V_o et/ou I_o exigées par la consigne de la commande du convertisseur.

D'après l'équation (1.15), l'ondulation du courant I_o sera maximale pour $\alpha = 0,5$. C'est à partir de cette valeur de α que l'on devra dimensionner la valeur de L , afin d'être sûr que l'ondulation \tilde{I}_L n'excédera jamais la valeur spécifiée. En effet, tout dimensionnement de L se fait à partir d'une valeur de ΔI_{Lmax} imposée. Si l'exploitation de la relation (1.16) conduit à des valeurs de L_{min} excessives, on devra augmenter la valeur de la fréquence de commutation afin de pouvoir réduire la valeur de L en proportion.

En ce qui concerne l'ondulation de la tension en sortie V_o , soit \tilde{V}_o , nous l'avons négligé en la considérant minime devant les grandeurs continues. Nous devons cependant définir une relation qui permette de lier cette ondulation à la valeur du condensateur C , afin d'être en mesure de le dimensionner.

Compte tenu de la relation courant/tension définissant le condensateur C et l'équation (1.14), on peut écrire:

$$I_c = \tilde{I}_L = C \cdot \frac{d\tilde{V}_o}{dt} \Rightarrow \tilde{V}_o = \frac{1}{C} \int_0^T \tilde{I}_L dt \quad (1.18)$$

Les variations de la tension V_o sont l'intégrale des ondulations du courant dans l'inductance L. Nous les avons représentées dans la (figure.1.10). Les variations \tilde{V}_o sont ainsi des arcs de paraboles.

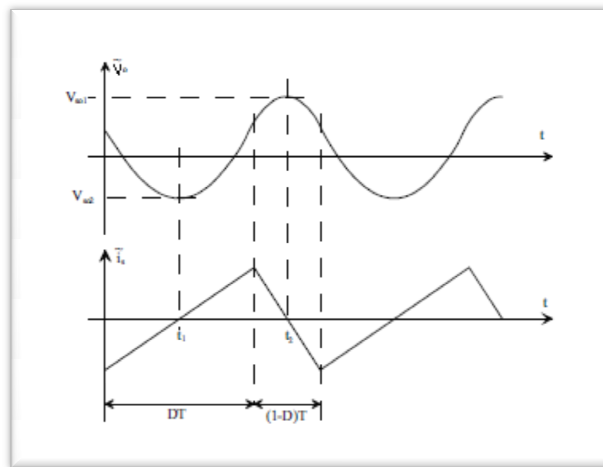


Figure 1.10 : Ondulation de la tension V_o .

Nous définissons ΔV_o l'amplitude des ondulations \tilde{V}_o . D'après les notations adoptées dans la (figure 1.10), il vient:

$$\Delta V_o = V_{o2} - V_{o1} = \tilde{V}_o(t_2) - \tilde{V}_o(t_1) = \frac{1}{C} \int_{t_1}^{t_2} \frac{d\tilde{I}_L}{dt} \quad (1.19)$$

Compte tenu de l'évolution linéaire du courant dans L (Fig.1.9), le calcul de cet intégral est aisé:

$$\Delta V_o = \frac{1}{C} \left[\frac{\alpha T_s}{8} \Delta I_L + \frac{(1-\alpha)T_s}{8} \Delta I_L \right] = \frac{\Delta I_L}{8Cf_s} = \frac{(1-\alpha)\alpha V_i}{8Cf_s^2 L} \quad (1.20)$$

$$\Rightarrow C \geq \frac{(1-\alpha)\alpha V_i}{8Lf_s^2 \Delta V_{o\max}} \quad (1.21)$$

Comme pour ΔI_L , les paramètres V_i , L , C et f_s sont des paramètres constants. En revanche, le rapport cyclique est susceptible de varier. Pour juger de son influence, on se reportera à la (figure 1.9), puisque l'influence de α est identique sur ΔI_L et sur ΔV_o .

On notera toutefois l'influence de la fréquence de commutation f_s du convertisseur. Il est très avantageux de travailler à une fréquence de commutation élevée, car une valeur plus faible de C permet d'obtenir une même amplitude de ΔV_o . [8]

1.5 Conclusion

Dans ce chapitre un rappel sur l'état de l'art des convertisseurs DC-DC a été envisagé, l'étude du convertisseur abaisseur (Buck) a été développée dans le but d'entamer sa régulation dans le prochain chapitre.

2.1 Introduction :

Le régulateur le plus utilisé dans l'industrie est le régulateur PID (proportionnel intégral dérivé). Il permet de régler à l'aide de ses trois actions les performances d'un système en boucle fermée (amortissement, temps de réponse). La prédominance incontestée de ce type de régulateurs provient, outre de la simplicité de sa structure et le nombre restreint de paramètres à ajuster, des performances qu'il peut offrir aux systèmes en boucle fermée, satisfaisant très souvent les cahiers des charges, si ses paramètres sont choisis judicieusement.

Plusieurs méthodes de synthèse d'un régulateur PID, offrant une complexité et des performances très variables, sont disponibles dans la littérature [14]. Cependant, un PID peut être considéré comme un outil dont l'ajustement peut être accompli en considérant indépendamment l'effet produit par chacun de ses paramètres. Ceci a conduit à des règles et formules empiriques d'ajustement très utiles.

Les régulateurs PID répondent à plus de 95% des besoins industriels et le nombre de régulateurs installés dans une usine pétrolière, par exemple, se compte par milliers. Malheureusement, malgré l'expérience acquise au fil des ans, les valeurs choisies pour les paramètres des actions P, I et D ne sont pas toujours satisfaisantes, ni adaptées au processus à régler.

En 1942, Ziegler et Nicholas [14] ont proposé deux démarches permettant de trouver facilement les paramètres optimaux pour une installation donnée. Au fil des ans, les propositions de Ziegler et Nicholas ont été adaptées ou modifiées selon les besoins.

En 1963, Hurwitz [15] a ajouté un degré de liberté supplémentaire au régulateur PID afin de

mieux contrôler les dépassements obtenus lors d'une réponse indicielle. Ce nouveau degré de liberté consiste, en particulier, à ne réinjecter vers le terme proportionnel qu'une partie du signal de sortie.

Au début des années 1990 et dans le but de fournir des règles d'ajustement simples mais plus performantes que celles de Ziegler-Nichols, Aström [16] et ses collaborateurs ont analysé le comportement dynamique d'un grand nombre de processus. Cette analyse a conduit à l'établissement de tableaux servant aux calculs des paramètres des actions P, I et D à partir de mesures simples.

Dans ce chapitre, après avoir présenté les régulateurs PID classiques et modifiés nous présentons les différentes méthodes de synthèse : par compensation des pôles, celles de Ziegler et Nicholas, et celles de Hurwitz [17] [18]. Une application de la synthèse par placement des pôles est présentée en fin de chapitre.

2.2 Les types de régulateurs :

- Régulateur proportionnel P.
- Régulateur proportionnel dérivé PD.
- Régulateur proportionnel intégral PI.
- Régulateur proportionnel intégral dérivé PID.
- Régulateur logique floue.

2.2.1 L'Algorithme P :

Le rôle de l'action proportionnelle est de minimiser l'écart ε entre la consigne et la mesure, et elle réduit le temps de monter et le temps de réponse. On constate qu'une augmentation du gain k_p du régulateur entraîne une diminution de l'erreur statique et permet d'accélérer le comportement global de la boucle fermée [19].

$$p = k_p \varepsilon(t) \quad (2.1)$$

On serait tenté de prendre des valeurs du gain élevé pour accélérer la réponse du Pprocédé mais on est limité par la stabilité de la boucle fermée. En effet, une valeur trop élevée du gain, implique une augmentation de l'instabilité du système et donne lieu à des oscillations. Dans le cas d'un régulateur P, et pour définir le sens d'action qui détermine le sens de variation de mesure et de consigne et on écrit :

$$u = u_0 \pm k_R(y_c - y) \quad (2.2)$$

Avec u_0 est la valeur centrale [19].

2.2.2 L'Algorithme I :

Le correcteur intégral est en général associé au correcteur proportionnel, il élabore Alors une commande qui peut être donnée par la relation suivante:

$$u(t) = k_p \left(\varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(\tau) d\tau \right) \quad (2.3)$$

La fonction de transfert du correcteur est alors donnée par :

$$C(p) = k_p \frac{1+T_{iP}}{T_{iP}} \quad (2.4)$$

Pour un régulateur intégral pur, le régime dynamique est relativement long. D'un autre côté, le régulateur proportionnel réagit immédiatement aux écarts de réglage mais il n'est pas en mesure de supprimer totalement l'erreur statique. La combinaison des actions proportionnelle et intégrale permettent d'associer l'avantage du régulateur PID, c'est-à-dire La réaction rapide à un écart de réglage, à l'avantage du régulateur I qui est la compensation exacte de la grandeur pilote.

La transposition de correcteur continu consiste à discrétiser un correcteur continu afin de l'utiliser dans une commande numérique [19].

2.2.3 L'Algorithme PID :

Le régulateur PID classique relie directement le signal de commande $u(t)$ au signal d'écart $e(t)$. Sa description temporelle est la suivante :

$$u(t) = k_p \left(e(t) + \frac{1}{T_i} \int_{-\infty}^t \varepsilon(\tau) d\tau + T_d \frac{d e(t)}{d t} \right) \quad (2.5)$$

Avec l'écart défini comme suit :

$$e(t) = w(t) - y(t) \quad (2.6)$$

Sa fonction de transfert s'écrit :

$$G_C(s) = \frac{u(s)}{E(s)} = k_p \left(1 + \frac{1}{sT_i} + sT_d \right) \quad (2.7)$$

Cette combinaison des termes P, I et D est aussi désignée sous le nom de forme parallèle ou non-interactive (Fig.2.1)[19].

2.3 Principe général d'un correcteur PID:

L'erreur observée est la différence entre la consigne et la mesure. Le PID permet trois actions en fonction de cette erreur:

- Une action Proportionnelle: l'erreur est multipliée par un gain k_p ;
- Une action Intégrale: l'erreur est intégrée sur un intervalle de temps s , puis multipliée par un gain k_i ;
- Une action Dérivée: l'erreur est dérivée suivant un temps s , puis multipliée par un gain k_d .

Les actions dérivées et intégrales ne s'emploient jamais seules mais en combinaison avec l'action proportionnelle [20].

Il existe plusieurs architectures possibles pour combiner les trois effets (série, parallèle ou mixte), on présente ici une architecture parallèle:

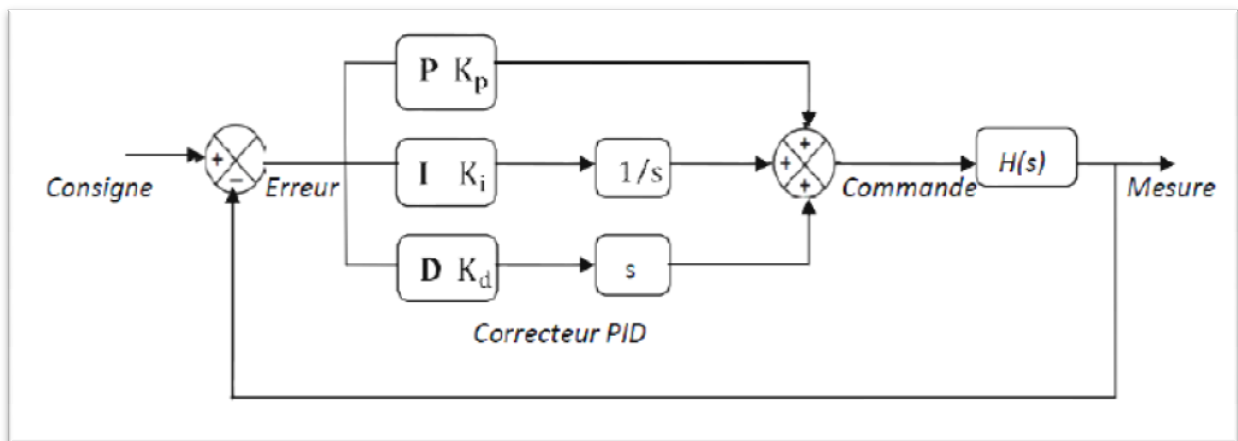


Figure 2.1 : Correcteur PID

La fonction de transfert avec la transformée de Laplace du régulateur PID parallèle est la somme des trois actions:

$$C(s) = k_p + k_i * \frac{1}{s} + K_d S = \frac{k_d S^2 + k_p S + 1}{s} \text{ avec } S = \frac{du}{dt} \quad (2.8)$$

2.4 Réglage par un PID :

Le réglage par un PID consiste à déterminer les coefficients K_p , K_i et K_d afin d'obtenir une réponse adéquate du procédé et de la régulation. L'objectif est d'être robuste, rapide et précis. Il faut pour cela limiter le/ou les éventuels dépassements (*over shoots*).

- La robustesse est sans doute le paramètre le plus important et délicat. On dit qu'un système est robuste si la régulation fonctionne toujours même si le modèle change un peu. Un régulateur doit être capable d'assurer sa tâche même avec ces changements afin de s'adapter à des usages non prévus.
- La rapidité du régulateur dépend du temps de montée et du temps d'établissement du régime stationnaire.
- Le critère de précision est basé sur l'erreur statique.

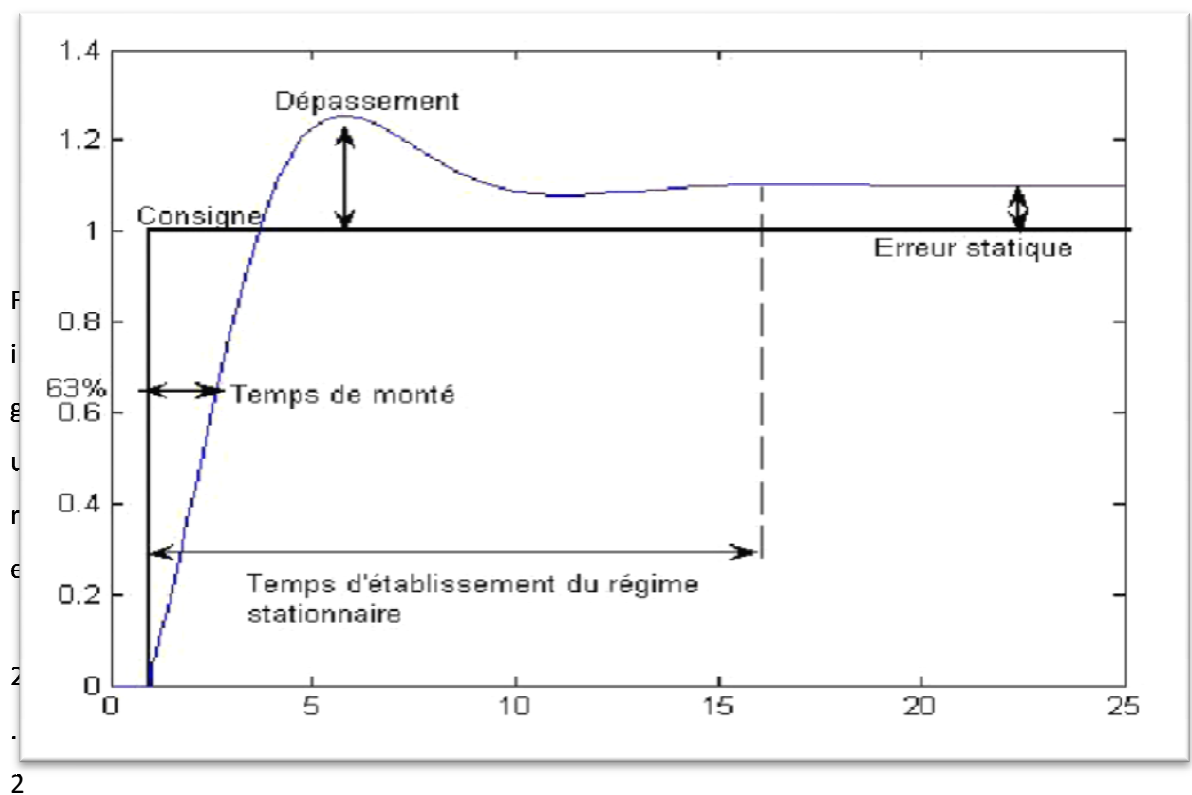


Figure 2.2 : Réponse d'un système du second ordre

2.5 Les paramètres du PID influant la réponse du système :

Les paramètres du régulateur PID influent sur la réponse du système de la manière suivante :

- k_p : Lorsque k_p augmente, le temps de montée (*rise time*) est plus court mais il y a un dépassement plus important. Le temps d'établissement varie peu et l'erreur statique se trouve améliorée.
- k_i : Lorsqu'il augmente, le temps de montée est plus court mais il y a un dépassement plus important. Le temps d'établissement au régime stationnaire s'allonge mais dans ce cas on assure une erreur statique réduite. Donc plus ce paramètre est élevé, moins l'erreur statique est grande, mais plus la réponse du système est ralentie.
- k_d : Lorsque k_d augmente, le temps de montée change peu mais le dépassement diminue. Le temps d'établissement au régime stationnaire est meilleur. Pas d'influences sur l'erreur statique. Si ce paramètre est trop élevé, le système anticipe trop et la consigne n'est pas atteinte dans des délais adéquats. Pour ces trois paramètres, le réglage au-delà d'un seuil trop élevé a pour effet d'engendrer une oscillation du système de plus en plus importante menant à l'instabilité.

2.6 Les caractéristiques du régulateur PID :

Le régulateur standard le plus utilisé dans l'industrie est le régulateur PID (proportionnel intégral dérivé), car il permet de régler à l'aide de ses trois paramètres les performances (amortissement, temps de réponse, ...) d'un processus modélisé par un deuxième ordre. Nombreux sont les systèmes physiques qui, même en étant complexes, ont un comportement voisin de celui d'un deuxième ordre. Par conséquent, le régulateur PID est bien adapté à la plupart des processus de type industriel et est relativement robuste par rapport aux variations des paramètres du procédé [21].

Si la dynamique dominante du système est supérieure à un deuxième ordre, ou si le système contient un retard important ou plusieurs modes oscillants, le régulateur PID n'est plus adéquat et un régulateur plus complexe (avec plus de paramètres) doit être utilisé, au dépend de la sensibilité aux variations des paramètres du procédé.

Il existe trois types d'algorithmes PID, le PID série, le PID parallèle et le PID mixte.

2.7 Structure des régulateurs PID :

2.7.1 Type Parallèle :

$$PID = K_p \left(1 + \frac{1}{sT_i} \right) (1 + sT_d) \tag{2.9}$$

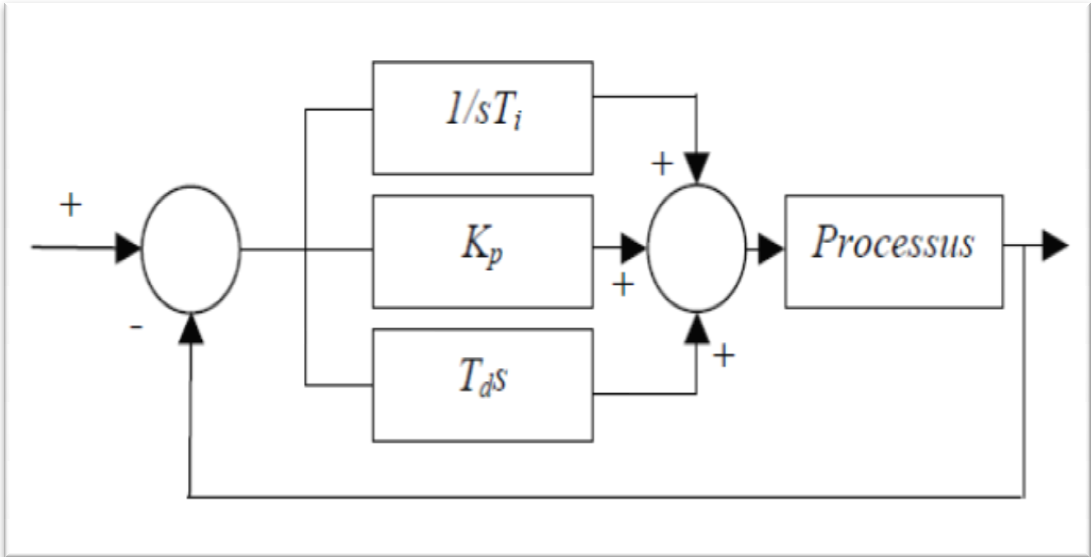


Figure 2.3 : régulateur parallèle

2.7.2 Type Mixte :

$$PID = k_p + \frac{1}{s+T_i} + sT_d \tag{2.10}$$

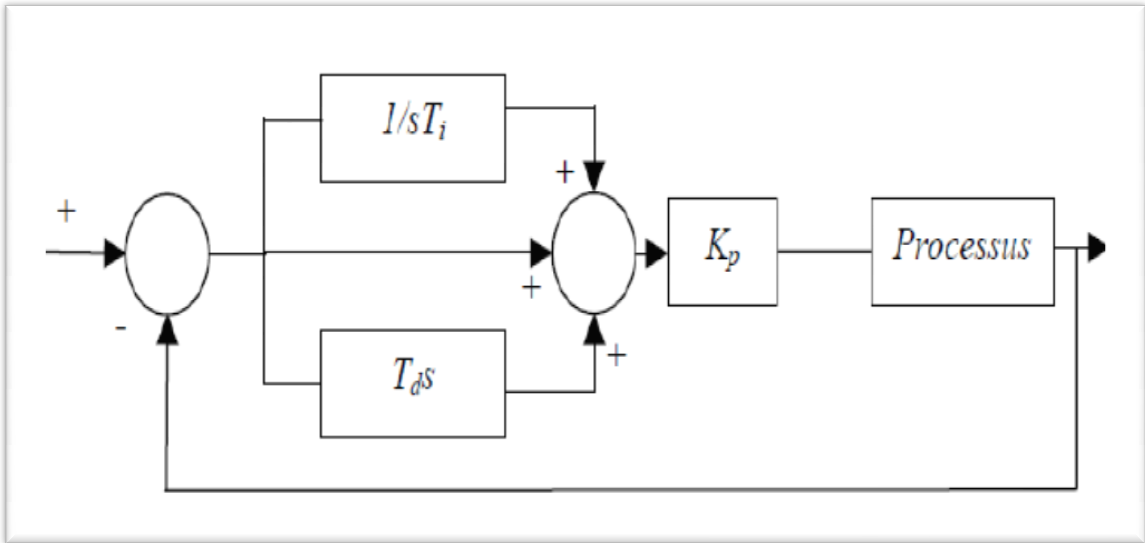


Figure 2.4 : régulateur mixte

2.7.3 Type Série :

$$PID = k_p \left(1 + \frac{1}{s + T_i}\right) (1 + sT_d) \quad (2.11)$$

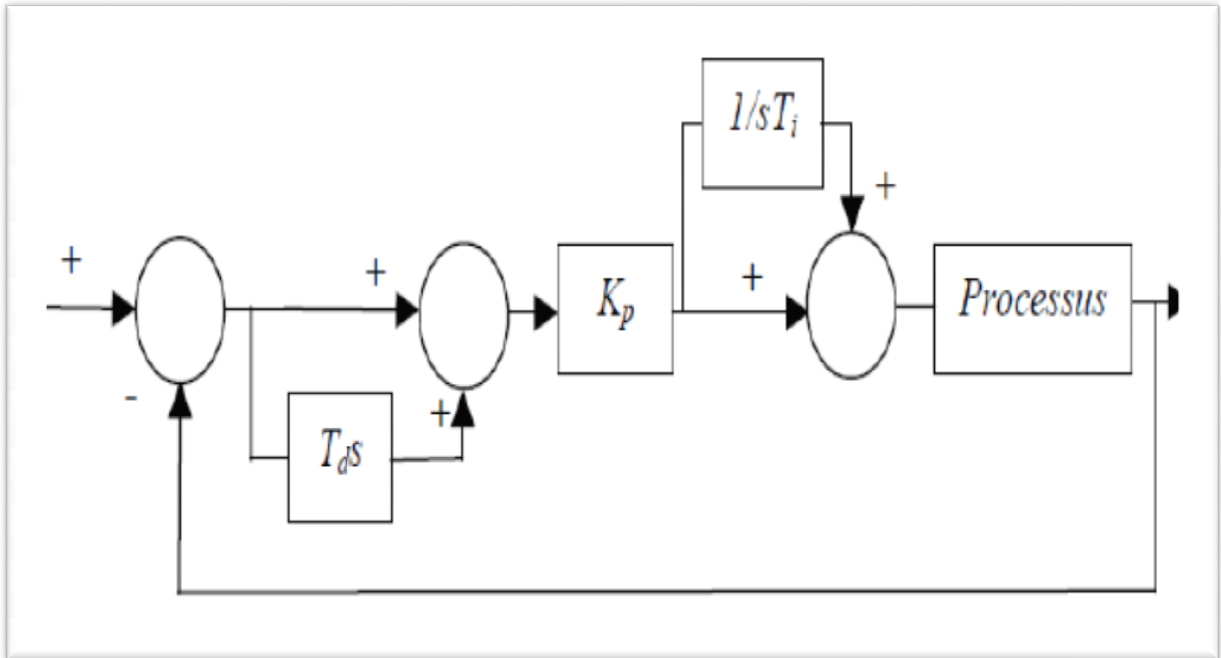


Figure 2.5 : régulateur série

2.8 Correction en boucle ouverte :

Dans une correction en boucle ouverte, la grandeur de commande ne dépend que du signal de référence w , et le correcteur ne reçoit pas d'informations sur la sortie réelle du système. Ce cas serait intéressant, si on sait qu'une commande quelconque appliquée au système va toujours engendrer la même réponse indépendamment du fait que les perturbations sont présentes ou non, ou de leurs importances (dans les cadres tolérés). L'avantage immédiat de cette approche réside dans le fait qu'un tel correcteur sera simple à concevoir en plus d'un coût très intéressant. Mais dans la majorité des cas réels l'image de la sortie est nécessaire pour le calcul de la commande adéquate. Ceci nous guide à l'introduction d'une boucle de retour qui donne cette image à un comparateur qui sert à calculer l'erreur ε .

Une application où on peut trouver ce type de correcteurs est la commande des moteurs pas à pas. Dans ce type de moteurs, pour chaque impulsion reçue par le moteur, ce dernier va tourner avec un angle spécifique quelle que soit sa charge (dans la limite de son fonctionnement). Donc suivant le nombre d'impulsions on connaît exactement sa position,

sans la nécessité d'un capteur. On peut aussi le trouver dans les systèmes où les perturbations sont négligeables, où le contrôle précis n'est pas nécessaire.

2.9 Correction en boucle fermée :

Le schéma classique d'une correction en boucle fermée est présenté dans la figure (2.6). Le régulateur, dont la fonction de transfert est désignée par $G_c(s)$, est situé en amont du système à régler $G_a(s)$

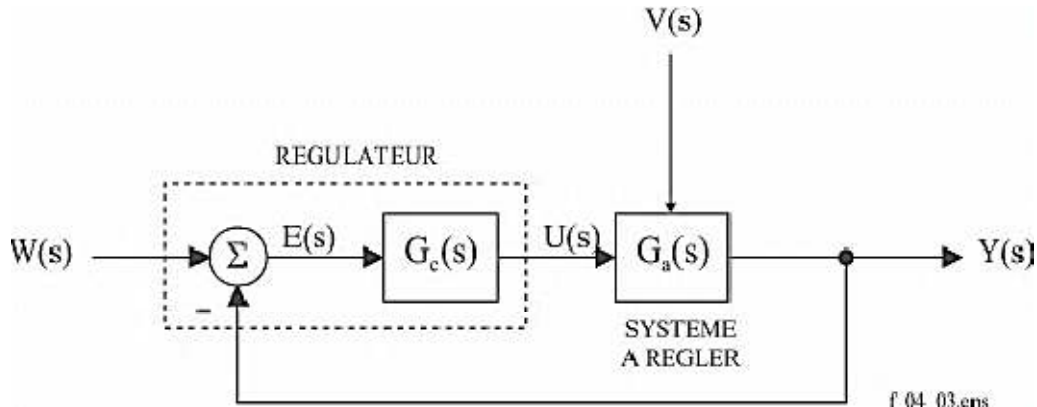


Figure 2.6 : Schéma fonctionnel d'un système asservi mono-variable (correction en boucle fermée).

Le système à régler $G_a(s)$ comprend, outre le processus, l'amplificateur de puissance, l'actionneur, le capteur et l'électronique de traitement de la mesure associée [22]. Appliquée au système à régler, la commande $u(t)$ provoque donc une modification de la grandeur réglée $y(t)$. Le régulateur en tenant compte pour former $u(t)$, on constate que $y(t)$ apparaît :

- à l'origine de l'action entreprise par le régulateur ;
- comme conséquence de cette action [22].

2.10 Les actions PID :

Un régulateur PID est obtenu par l'association de ces trois actions et il remplit essentiellement les trois fonctions suivantes :

- Fonction proportionnelle donne un système plus précis, plus rapide ;
- Fonction intégrateur élimine l'erreur statique ;
- Fonction dérivée accélère la correction.

2.11 La synthèse par la réponse indicielle et la réponse fréquentielle :

Il existe d'autres méthodes de synthèse qui se basent sur la réponse du processus, selon qu'elle soit temporelle ou fréquentielle. Avant d'entamer ces deux méthodes on présente ici les réponses d'un processus apériodique :

2.12 Réponse indicielle :

La réponse indicielle d'un processus apériodique est illustrée par la figure (2.7). On y a mis en évidence les instants t_1 , t_2 , t_3 qui nous permettent de définir les grandeurs suivantes:
Le retard apparent :

$$L=t_1 \quad (2.12)$$

La constante de temps apparente :

$$T=t_2 - t_1 \quad (2.13)$$

La pente de la tangente au point d'inflexion :

$$p \frac{y(\infty)}{t_3 - t_1} = \frac{a}{L} \quad (2.14)$$

Le temps mort relatif :

$$\tau = \frac{L}{L+T} = \frac{t_1}{t_2} \quad (2.15)$$

Le rapport entre la valeur asymptotique $y(\infty)$ et l'amplitude E du saut appliqué en entrée détermine le gain statique K_0 du processus :

$$K_0 = \frac{Y(\infty)}{E} \quad (2.16)$$

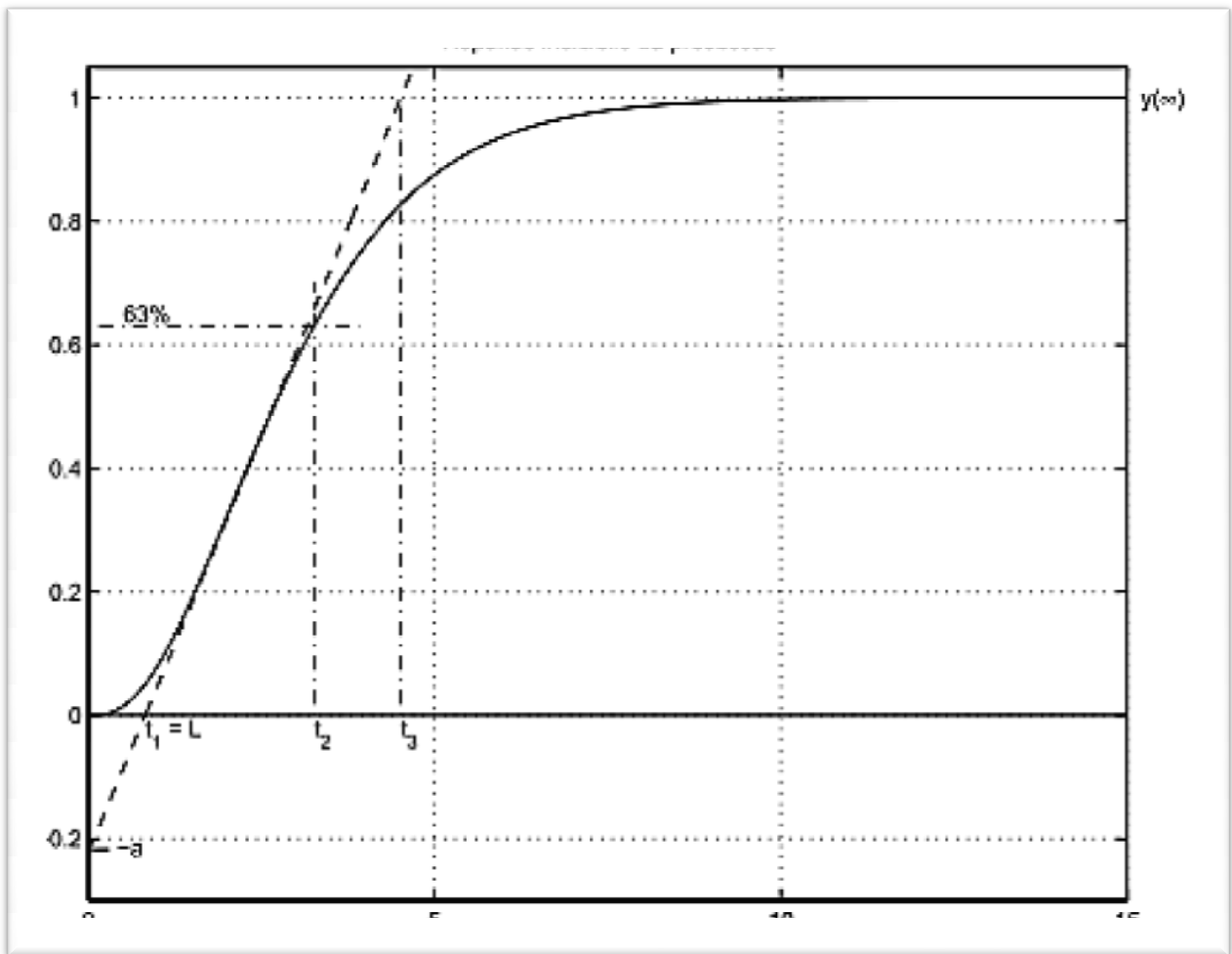


Figure 2.7 : Réponse indicielle d'un processus apériodique

2.13 La réponse fréquentielle :

La réponse fréquentielle d'un processus apériodique est illustrée par la figure (2.8). Sur cette réponse, on définit les grandeurs suivantes :

- la pulsation w_π pour laquelle la phase vaut -180°
- le gain G_π correspondant à cette pulsation
- le gain critique K_{cr} qu'il faut introduire dans le système bouclé pour le rendre instable :

$$K_{cr} = \frac{1}{w_\pi} \quad (2.17)$$

- le gain relatif :

$$K = \frac{G_\pi}{G(0)} = \frac{1}{K_{cr}K_0} \quad (2.18)$$

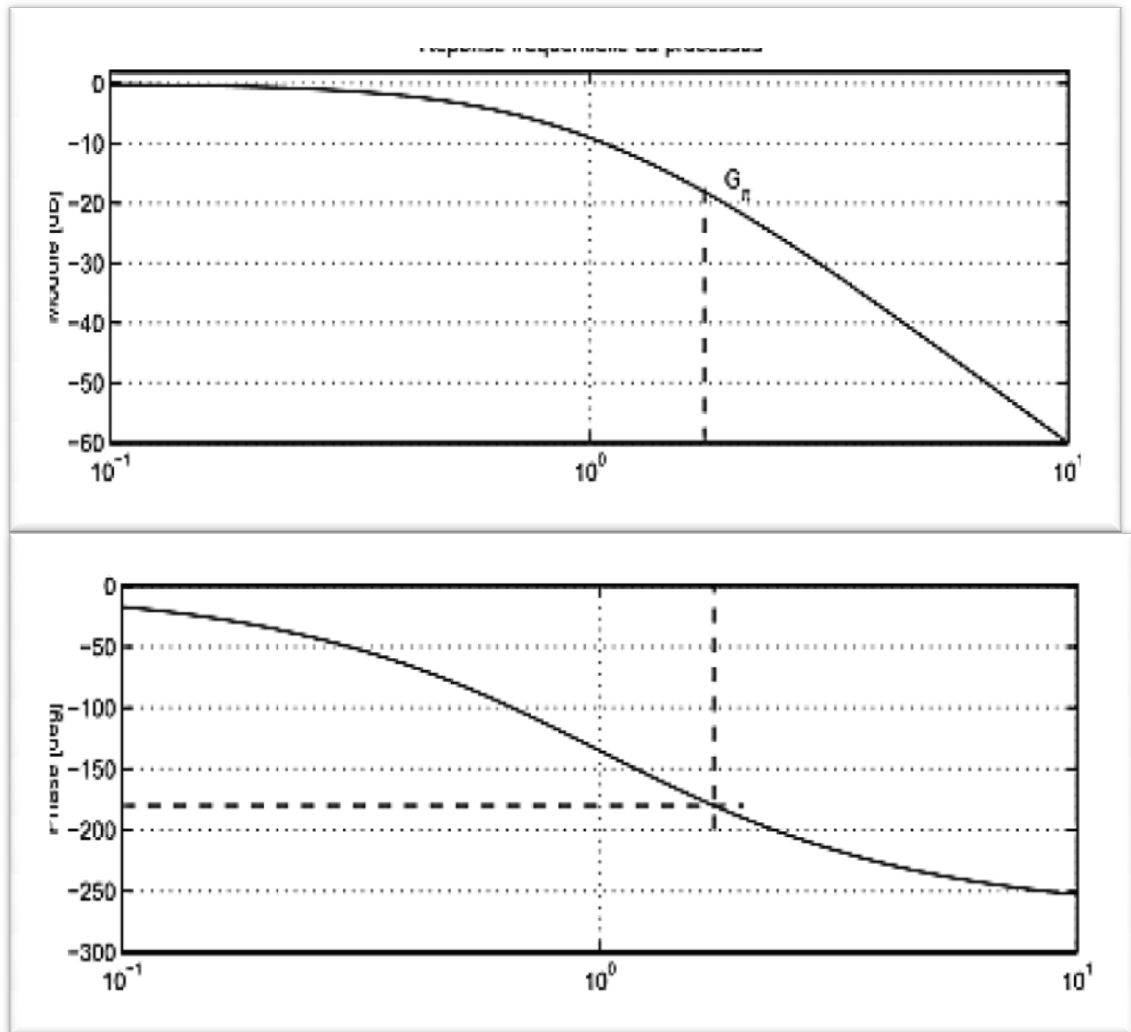


Figure 2.8 : Réponse fréquentielle d'un processus aperiodique d'ordre 3

2.14 Méthodes de Ziegler et Nicholas (ZN) :

Dans notre travail nous nous intéressons à deux méthodes de régulation, à savoir la méthode Ziegler et Nicholas [23]. En 1942, Ziegler et Nicholas ont proposé deux approches heuristiques basées sur leur expérience et quelques simulations pour ajuster rapidement les paramètres des régulateurs P, PI et PID. La première méthode nécessite l'enregistrement de la réponse indicielle en boucle ouverte, alors que la deuxième demande d'amener le système bouclé à sa limite de stabilité.

2.14.1 Méthode de la réponse indicielle :

Pour obtenir les paramètres du régulateur PID, il suffit d'enregistrer la réponse indicielle du processus seul (c'est-à-dire sans le régulateur), puis de tracer la tangente au point d'inflexion de la courbe. On mesure ensuite sa pente p , le retard apparent L correspondant au point d'intersection de la tangente avec l'abscisse et le gain $K_0 = \frac{Y_\alpha}{E}$ (Fig.2.6). On peut alors calculer les coefficients du régulateur choisi à l'aide du tableau (2.1). Généralement, les gains K_p proposés par Ziegler-Nicholas sont trop élevés et conduisent à un dépassement supérieur à 20%. Il ne faut donc pas craindre de réduire K_p d'un facteur deux (2) pour obtenir une réponse satisfaisante.

Type	K_p	T_i	T_d
P	$1/(PLK_0)=1/(aK_0)$		
PI	$0.9/(PLK_0)=0.9/(aK_0)$	3L	
PID	$1.2/(PLK_0)=1.2/(aK_0)$	2L	0.5L

Tab.2.1: Paramètres PID obtenus à partir d'une réponse indicielle (ZNt)

2.14.2 Méthode du point critique :

Cette méthode est basée sur la connaissance du point critique du processus. Expérimentalement, on boucle le processus sur un simple régulateur proportionnel dont on augmente le gain jusqu'à amener le système à osciller de manière permanente ; on se trouve ainsi à la limite de stabilité. Après avoir relevé le gain critique k_{cr} du régulateur et la période d'oscillation k_{cr} de la réponse, on peut calculer les paramètres du régulateur choisi à l'aide du tableau (2.2). Ici également, les valeurs proposées conduisent à un temps de montée relativement court malheureusement assorti d'un dépassement élevé. Cette situation n'étant pas toujours satisfaisante, on peut être amené à corriger les coefficients proposés et, en particulier, à diminuer le gain.

On notera que les paramètres T_i et T_d proposés par les deux méthodes de Ziegler-Nicholas sont dans un rapport constant égal à 4. Le régulateur possède donc deux zéros confondus valant :

$$-1/(2T_d) = -2/T_i \quad (2.20)$$

Type	K_P	T_i	T_d
P	$0.5K_{cr}$		
PI	$0.4K_{cr}$	$0.8T_{cr}$	
PID	$0.6K_{cr}$	$0.5T_{cr}$	$0.125T_{cr}$

Tab.2.2: Paramètres PID obtenus à partir du point critique (ZNf)

Dans notre étude, nous nous intéressons uniquement au régulateur PID de type Ziegler et Nicholas fréquentielle (ZNf)[24, 25, 26].

2.15 Méthodes de Hurwitz :

Pour le calcul des paramètres PID par rapport à Hurwitz méthode polynomiale [27], le transfert en boucle fermée est indispensable. L'utilisation de la fonction est la méthode de réduction de schéma-bloc, la fonction de transfert en boucle fermée est obtenue sous la forme :

$$\frac{C(s)}{R(s)} = \frac{(k_p T_i S^2 + k_p T_i S + k_p) \left(\frac{V_{in}}{LC}\right)}{S^3 + \left(\left(\frac{k_p T_d V_{in}}{Lc}\right) + \left(\frac{1}{Rc}\right)\right) S^2 + \left(\frac{k_p V_{in+1}}{Lc}\right) S + \left(\frac{k_p V_{in}}{Lc T_i}\right)} \quad (2.21)$$

La boucle fermée du polynôme caractéristique du système contrôlé PID est alors donnée par :

$$S^3 + \left(\left(\frac{k_p T_d V_{in}}{Lc}\right) + \left(\frac{1}{Rc}\right)\right) S^2 + \left(\frac{k_p V_{in+1}}{Lc}\right) S + \left(\frac{k_p V_{in}}{Lc T_i}\right) = 0 \quad (2.22)$$

Les coefficients K_p , T_i et T_d sont choisis de sorte que le polynôme Hurwitz de troisième degré, devient de l'ordre de deuxième de la forme suivante :

$$P(s) = (S^2 + 2Lw_n s + w_n^2)(S + \alpha) \quad (2.23)$$

Dont les coefficients ζ, ω_n , et α sont des quantités positives. On égalisant les deux équations (2.22) et (2.23), nous obtenons les valeurs des paramètres du Régulateur PID comme suit :

$$k_p = \frac{2\zeta w_n \alpha L C + w_n^2 L C - 1}{E} \quad (2.24)$$

$$T_i = \frac{E K_p}{L C \omega_n^2} \quad (2.25)$$

$$T_d = \frac{L C}{E k_p} \left(\alpha + 2\zeta W_n - \frac{1}{R C} \right) \quad (2.26)$$

2.16 Conclusion :

Dans ce chapitre, nous avons présenté une étude des correcteurs PID. En fait, nous avons présenté les différentes formes de ces régulateurs analogiques, les transformations qui permettent d'obtenir leur version numérique.

La régulation par correcteur PID est sans doute l'une des régulations les plus utilisées. Dans le prochain chapitre nous exploitons les équations cités ci-dessus, nous calculerons les différents coefficients de notre régulateur par les deux méthodes Hurwitz et Ziegler Nicholas (ZN).

Chapitre 3 Résultats de simulation du convertisseur DC/DC «Buck» avec un régulateur PID conçu sous SIMULINK

3.1 Introduction :

Ce chapitre sera consacré au dimensionnement des composants du convertisseur «Buck», au calcul des paramètres du régulateur PID on employant la méthode de Ziegler et la méthode de Hurwitz. Vu l'importance de la fréquence de travail f_s pour garantir une tension de sortie désirée, celle-ci sera bien choisie. Notre architecture adoptée sera implémentée sous l'environnement SIMULINK/MATLAB.

3.2 Calcule des valeurs de la self et de la capacité du convertisseur :

Les valeurs des composants du convertisseur DC-DC « Buck » sont résumées dans le tableau suivant on se servant des équations (3.2, 3.3 et 3.4)[28]:

<i>Description</i>		<i>Estimation</i>
La tension d'entrée	V_{in}	24V
La tension de sortie	V_o	18V
Condensateur	C	100 μ F
Inductance	L	1mH
Résistive	R	100 Ω
Rapport cyclique	δ	0.75
Fréquence de travail	f	48.828 kHz

Tableau 3.1 : Les valeurs des composants du convertisseur DC-DC << Buck>>

A partir des valeurs dressées dans le tableau ci-dessus on a :

$$V_{in} = 24V,$$

$$\Delta V_0/V_0=0.013\%,$$

$$f = 48.828\text{khz}$$

$$\Delta iL = 0.092 \text{ et } V_0=18V$$

L'équation de transfert est donnée par le rapport la tension de sortie sur rapport cyclique (3.1)[3, 4]et (3.2) [29].

$$\frac{V_0(s)}{\delta(s)} = \frac{V_s}{lcs^2 + \frac{L}{RS} + 1} \quad (3.1)$$

$$f = \frac{1}{(1+t_r)*(1+t_s)*(1+P)} \quad (3.2)$$

$$\delta = \frac{V_0}{V_{in}} \quad (3.3)$$

AN :

$$\delta = \frac{18}{24} = 0.75$$

$$L = \frac{(V_{in}-V_0)*d}{\Delta iL * f_{sw}} \quad (3.4)$$

AN :

$$L = \frac{(24-18)*0.75}{0.092*48.828} = 1\text{mH}$$

$$C = \frac{\Delta i \Delta iL}{8 * f_{sw} * \Delta V_0} \quad (3.5)$$

AN :

$$C = \frac{0.092*0.092}{8*48.828*0.013*18} = 100\mu\text{F}$$

3.3 Simulation sous SIMULINK du convertisseur DC/DC Buck :

A partir des équations d'états et les valeurs des composants du cahier de charge du convertisseur Buck, l'architecture adoptée à boucle ouverte pour la simulation sous SIMULINK/MATLAB est montrée dans la figure 3.3. Ci-dessous :

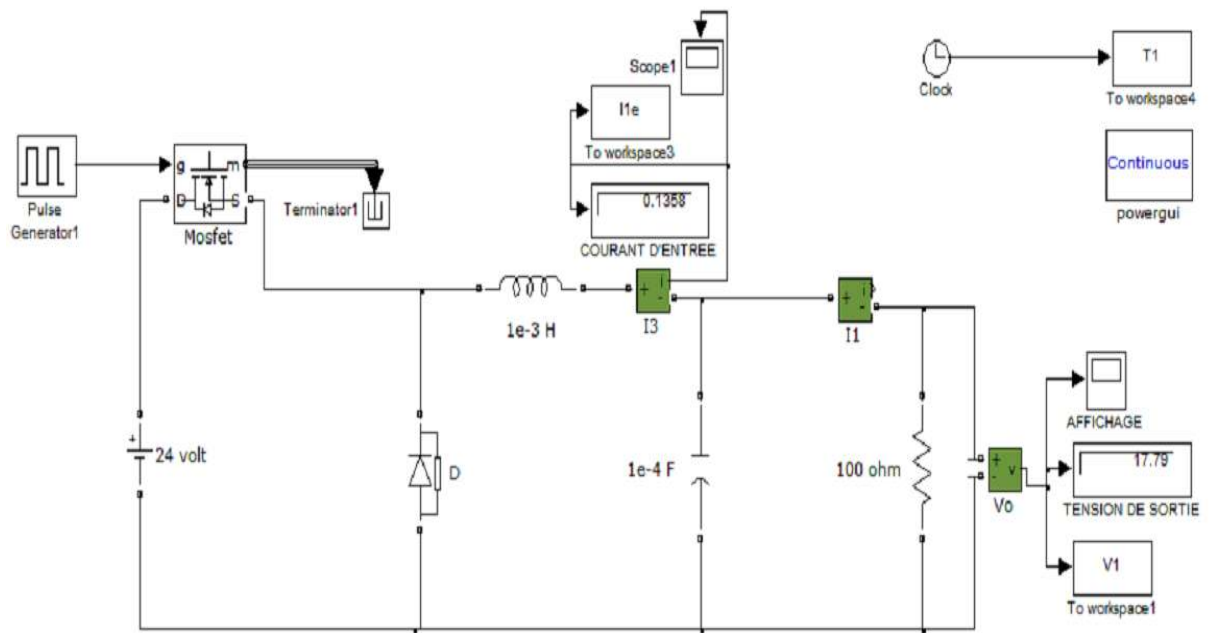


Figure 3.1: Simulation de convertisseur DC/DC Buck

3.4 Calcule des paramètres de régulateur PID :

Le schéma ci avant représente le principe de convertisseur DC/DC Buck avec un régulateur PID [30]:

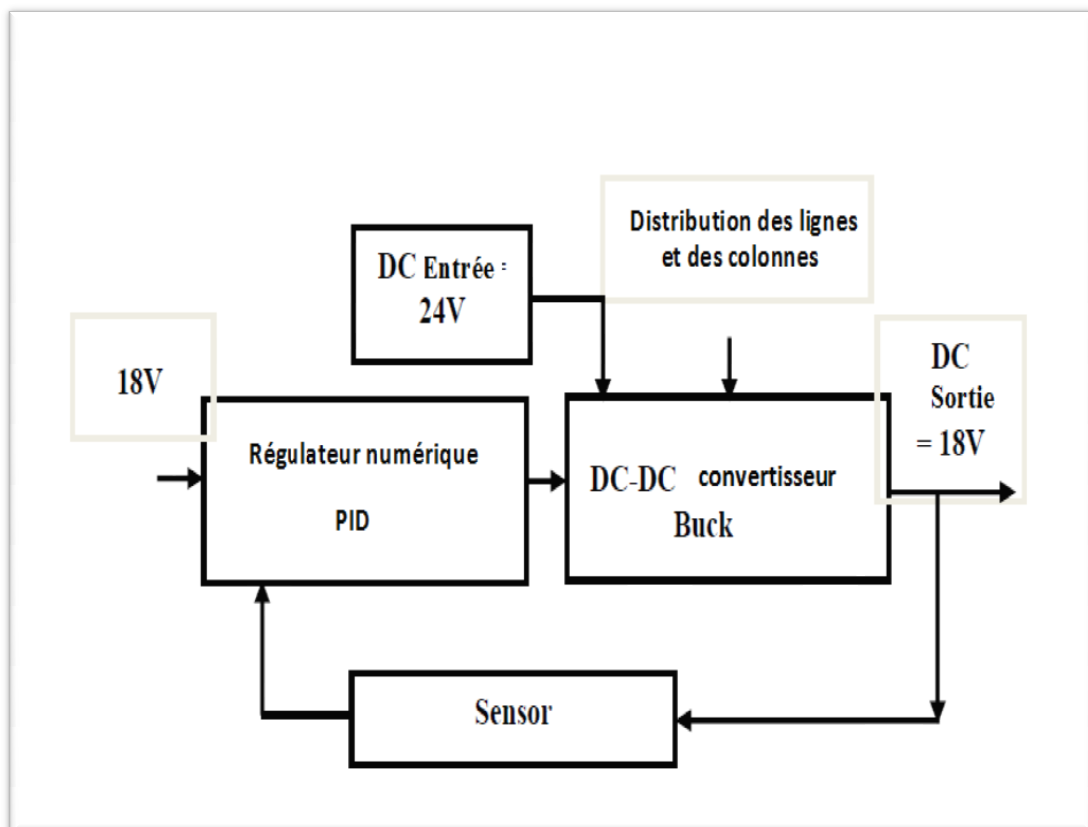


Figure 3.2 : Le principe du convertisseur DC/DC Buck avec un régulateur PID

3.4.1 Méthode de Hurwitz :

Avec l'utilisation des paramètres calculés précédemment nous calculons ces équations (2.24) et (2.25) et (2.26) [27] :

Avec:

$$\alpha = 35000,$$

$$\zeta = 0.707,$$

$$W_n = 2500 \text{ rad/sec}; [27].$$

$$k_p = \frac{2\zeta w_n \alpha Lc + w_n^2 Lc - 1}{E} \quad (3.6)$$

AN:

$$k_p = \frac{2 * 0.707 * 2500 * 35000 * 10^{-3} * 10^{-4} + 2500^2 * 10^{-3} * 10^{-4} - 1}{24}$$

$$k_p = 0.4998$$

$$T_i = \frac{EK_p}{Lcaw_n^2} \quad (3.7)$$

AN:

$$T_i = \frac{24 * 0.4998}{10^{-3} * 10^{-4} * 2500^2} = 5.4846 * 10^{-4}$$

$$T_d = \frac{Lc}{Ek_p} \left(\alpha + 2\zeta W_n - \frac{1}{RC} \right) \quad (3.8)$$

AN:

$$T_d = \frac{10^{-3} * 10^{-4}}{24 * 0.4998} \left(35000 * 2 * 0.707 * 25000 - \frac{1}{100 * 10^{-4}} \right)$$

$$T_d = 3.2036 * 10^{-4}$$

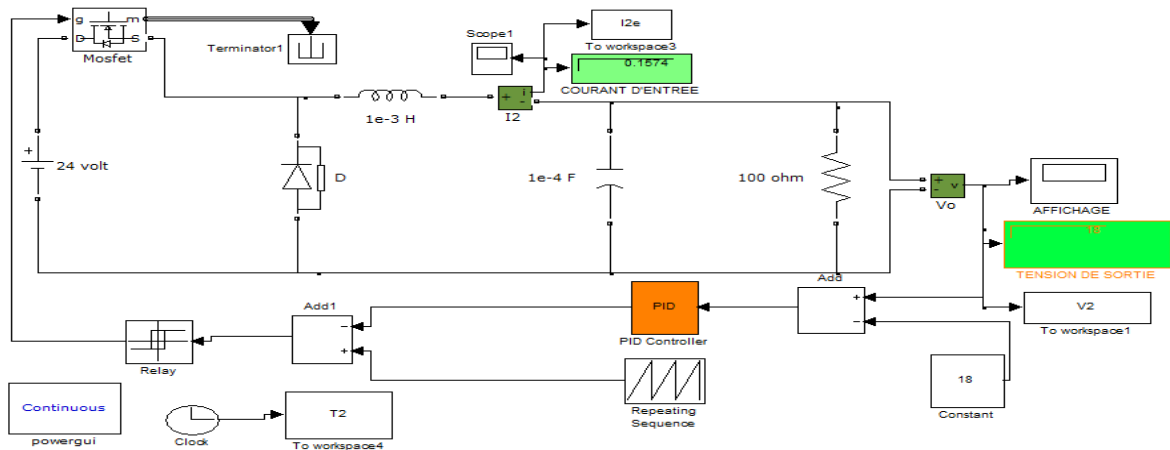


Figure 3.3 : Simulation de convertisseur DC/DC Buck avec un régulateur PID par la méthode de Hurwitz

3.4.2 Méthode Ziegler et Nicholas (ZN) :

Nous avons parlé de cette méthode dans le précédent chapitre, et nous avons choisi la méthode Ziegler et Nicholas fréquentielle (ZNf) (Tab.2.2) pour notre étude.

Avec :

$$k_p = 0.499 ;$$

$$T_i = 5.484610^{-4} ;$$

$$T_d = 3.2036 \cdot 10^{-4} ;$$

la pulsation critique : $w_{cr} = w_{\pi} = 1.73$ [rad/sec]

le gain critique :

$$K_{cr} = \frac{1}{w_{\pi}} \tag{3.9}$$

AN: $K_{cr} = \frac{1}{1.73} = 4.015$ [/]

la période critique :

$$T_{cr} = \frac{2\pi}{w_{\pi}}$$

$$T_{cr} = \frac{2 \cdot 3.14}{1.73} = 3.62 \text{ [sec]} \tag{3.10}$$

On se référant au tableau (2.2) ci-dessus, on calcule les trois paramètres du régulateur PID comme suit :

$$k_p = 0.6 \cdot k_{cr} = 2.41 \text{ [/]}; T_i = 0.5 \cdot T_{cr} = 1.81 \text{ [sec]}; T_d = 0.125 \cdot T_{cr} = 0.45 \text{ [sec]}$$

A partir des équations d'états et les valeurs des composants de cahier de charge du convertisseur Buck contrôlé par le régulations PID par la méthode de Ziegler, l'architecture adoptée sous SIMULINK/MATLAB est montrée dans la figure 3.5 . Ci-dessous :

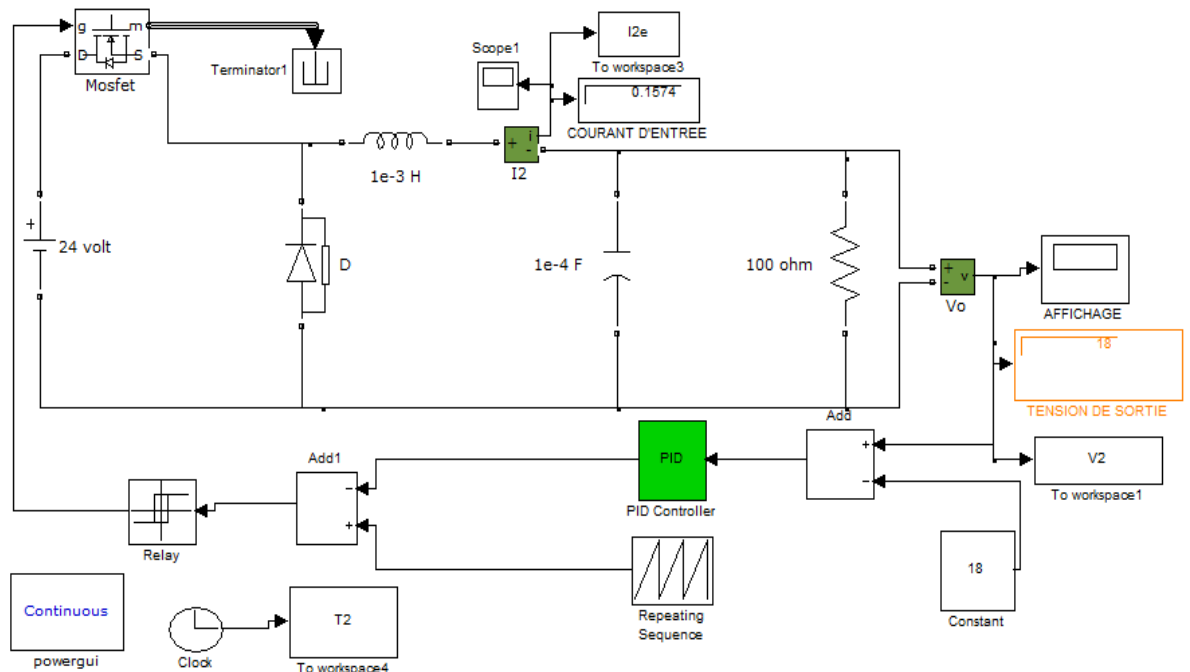


Figure 3.4 : Simulation de convertisseur DC/DC Buck contrôlé par un régulateur PID par la méthode de Ziegler et Nicholas (ZN)

3.5 Résultats de simulations et interprétations :

On analysons les résultats obtenus par la simulation sous SIMULINK, nous constatons ce qui suit :

La figure 3.5 illustre la tension de sortie de convertisseur Buck dans le cas de la boucle ouverte et avec le régulateur PID avec les deux méthodes : Ziegler et Hurwitz.

Il est apparent en boucle ouverte un dépassement important de 16 volts, un taux d'évion 80%, phénomène néfaste pour les charges en régime transitoire. Par contre dans les cas de contrôle par les deux méthodes Ziegler et Hurwitz, le dépassement est pratiquement nul. Nous remarquons aussi un meilleur temps de monté pour la méthode Hurwitz par apport à la méthode de Ziegler. De point de vu stabilité, la sortie suit correctement la consigne après un temps de 0.002s. La figure 3.6 représente la variation du courant dans la bobine, les remarques pour la tension sont valables pour courant.

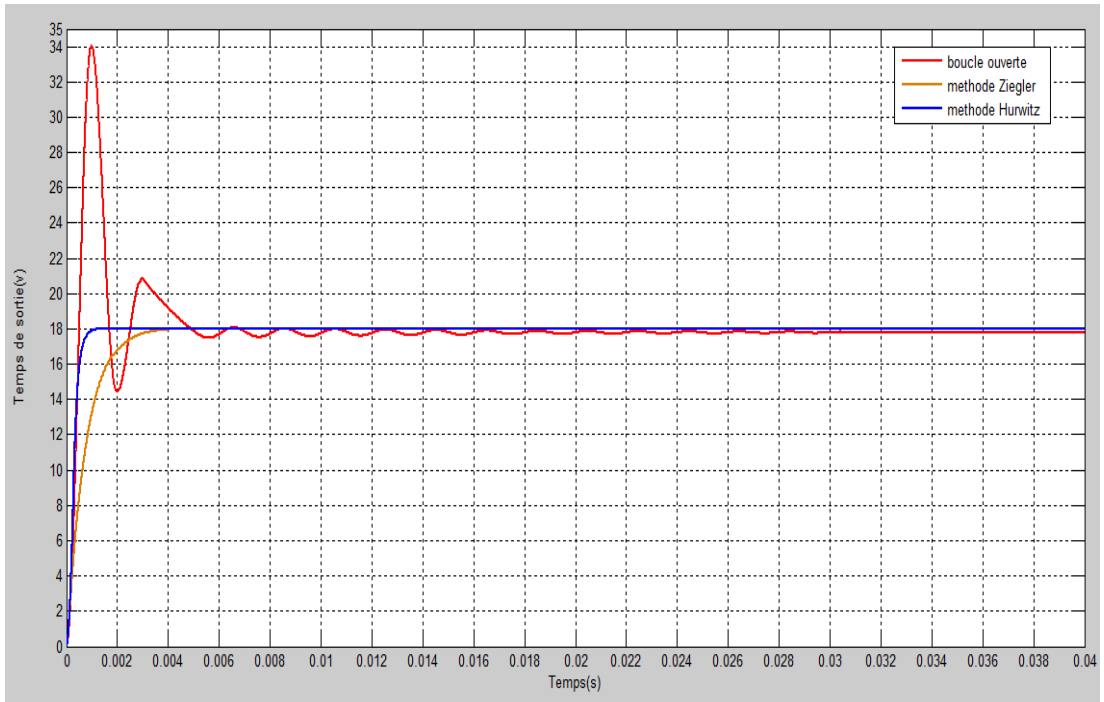


Figure 3.5 : Les tensions de sortie de convertisseur Buck pour une boucle ouverte avec les régulations PID pour la méthode de ziegler et celle de Hurwitz

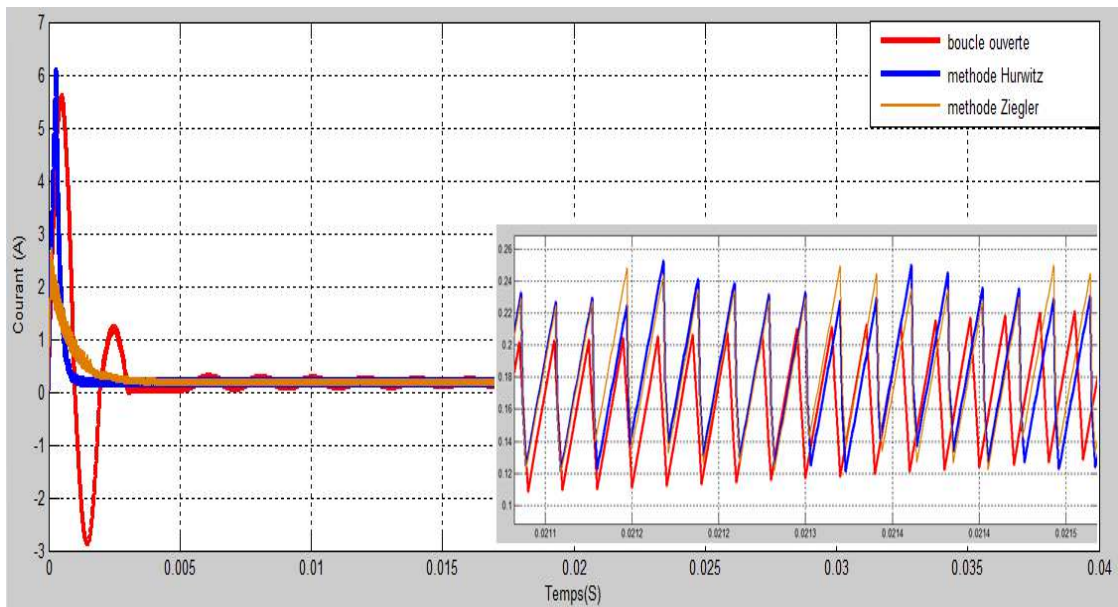


Figure 3.6 : Les Courant dans la bobine L de convertisseur Buck pour une boucle ouverte et les régulations PID pour la méthode de Ziegler et la méthode de Hurwitz

3.6 Conclusion :

Dans ce chapitre nous avons simulé notre système sous Simulink/MATLAB.

- Notre système est alimenté par une source de tension continue de 24V.
- Nous avons obtenu le signal de sortie par trois méthodes, la premier en boucle ouverte et enfin par les deux méthodes Ziegler et Hurwitz. La comparaison des différents cas est discutée.

Dans le prochain chapitre nous allons présenter les résultats de simulation sous l'environnement Xilinx System Generator.

Bibliographie

- [1] Ali JAAFAR : 'Contribution à la modélisation, l'analyse et l'optimisation de lois de commande pour convertisseurs DC-DC de puissance', thèse de doctorat en Physique, école d'doctorat STITS, 2011-17-TH.
- [2] Diaf.Said : ' Intégration des systèmes à sources renouvelables d'énergie pour la production décentralisée d'électricité en site isolé ', Mémoire de magister en électricité solaire de l'école Nationale Polytechnique, Alger, 2009.
- [3] J.-P. FERRIEUX et F. FOREST Alimentation à découpage. Convertisseur à résonance Masson, 1994. ISBN 2225845557.
- [4] R. BAUSIERE, F. LABRIQUE et G. SEGUIER "Les convertisseurs de l'électronique de puissance", volume 3, La conversion continu-continu. Technique & documentation, 1997, ISBN 2743001399.
- [5] H. FOCH, R. ARCHES, Y. CHÉRON, B. ESCAUT, P. MARTY, M. METZ Convertisseurs continu-continu non isolés. Technique de l'ingénieur, D 3 163.
- [6] Professeur A. Rufer et P. Barrade : 'Cours d'électronique de puissance Conversion DC/DC ', Ecole Polytechnique Fédérale de Lausanne.
- [7] Diaf.Said : " Intégration des systèmes à sources renouvelables d'énergie pour la production décentralisée d'électricité en site isolé ", Mémoire de magister en électricité solaire de l'école Nationale polytechnique, Alger,2009.
- [8] Professeur A. Rufer et P. Barrade : 'Cours d'électronique de puissance Conversion DC/DC ', Ecole Polytechnique Fédérale de Lausanne.
- [9]. J.G. Ziegler, N.B. Nichols : « Optimum settings for automatic controllers. »Trans. ASME, 64, pp. 759-768, 1942.
- [10]. I.M Horowitz (1963) : Synthesis of Feedback Systems. Academic Press,

New-York.

[11]. K.J. Aström, T. Häggglund: PID Controllers : Theory, Design and Tuning, Instrument Society of America, 2nd edition, 1995.

[12]. K.J. Aström, T. Häggglund, C.C. Hang, W.K. Ho: « Automatic Tuning and Adaptation for PID Controllers - A Survey. », Control Engineering Practice, 1:4, pp.699-714, 1993.

[13] Dancy A.P., Amirtharajah R. and Chandrakasan A.P., "High-Efficiency Multiple-Output DC–DC Conversion for Low-Voltage Systems", IEEE Trans. on Very Large Scale Integration (VLSI) Systems, Vol. 8, No. 3, June 2000: pp.252-263

[14]. Bouzid Aliane and Aladin Sabanovic, Design and implementation of digital bandpass FIR filter in FPGA, Computers in Education Journal, Vol. 14, 2004, pp. 76-81.

[15]. M. Canet, F. Vicedo, V. Almenar and J. Valls, FPGA implementation of an IF transceiver for OFDM-based WLAN, IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation, USA, 2004, pp. 227-232.

[16]. Xizhi Li and Tiejai Li, ECOMIPS: An economic MIPS CPU design on FPGA, in Proceedings of the 4th IEEE International Workshop on System-on-Chip for Real-Time Applications, Canada, 2004, pp. 291-294.

[17]. H. D. Maheshappa, R. D. Samuel and A. Prakashan, Digital PID controller for speed control of DC motors, IEEE Technical Review Journal, Vol. 6, No. 3, 1989, pp. 171-176.

[18]. J. Tang, PID controller using the TMS320C31 DSK with on-line parameter adjustment for real-time DC motor speed and position control, IEEE International Symposium on Industrial Electronics, Vol. 2, 2001, pp. 786-791.

[19] Mr.Rabi. Etude en simulation d'une regulation thermique par logique floue.University sidi mohamed ben abdallah , ecole superieur de technologie.2009/2010.

[20] ECET-462, « The Design of PID Controllers using Ziegler Nichols Tuning» -- Brian R Copeland; Purdue University, calumet; March, 2008

[21] Prof. Freddy Mudry- «Ajustage des Paramètres d'un Régulateur PID», ecole

d'ingénieurs du canton de vaud, mars 2006.

[22]. Michel Etique : « Régulation automatique », mars 2006, Yverdon- les Bains.

[23]. J.G. Ziegler, N.B. Nichols : « Optimum settings for automatic controllers. »Trans.

[24] A. Prodic, D. Maksimovic and R. W. Erickson, "Design and implementation of a Digital PWM controller for a high-frequency switching DC-DC power converter", The 27th Annual Conference of the IEEE Industrial Electronics Society (IECON'01), Vol. 2, pp. 893 - 898, November 2001.

[25] www.mathworks.com: "MATLAB/SIMULINK" Release 14 Notes, 2004.

[26] Ned Mohan, W.P. Robbins and T.M.Undeland, R. Nilssen and Olive Mo: "Simulation of Power Electronic and Motion Control Systems"; *Proceedings of the IEEE*, Vol.82, No.8, August 1994, pp. 1287 – 1292.

[27] HuiOuyang; JiguangYue; Yongqing Su, "Design and application of PID controllers based on interval computing theory," *International Conference on Electrical and Control Engineering (ICECE)*, 2011, vol., no., pp.1505,1510, 16-18 Sept. 2011.

[28] M.Gopal, *Digital control and state variable methods*, TATA McGraw Hill edition.

[29] A. Prodic, D. Maksimovic and R. W. Erickson, "Design and implementation of a Digital PWM controller for a high-frequency switching DC-DC power converter", The 27th Annual Conference of the IEEE Industrial Electronics Society (IECON'01), Vol. 2, pp. 893 - 898, November 2001

[30]. Hardware Implementation of Evolutionary Algorithm Assisted Digital PID Controllers for DC-DC Converters. S.Augustilindiya#, S.Palani*, K.Vijayarekha# and M.BreethiYeltsin Department of Electrical and Electronics Engineering, SASTRA University Tirumalaisamudram, Thanjavur, Tamilnadu 613401, India. Vol 5 No 4 Aug-Sep 2013.

[31] PHILIPPE LE-HUY : 'SIMULATION TEMPS RÉEL DE CONVERTISSEURS DE PUISSANCE À L'AIDE DE FPGA', Mémoire dans le cadre du programme de

maîtrise en génie électrique pour l'obtention du grade de maîtrises sciences (M.Se.), la Faculté des études supérieures de l'Université Laval, 2006. [32]Support du cours.

[33] Sylvie RENAUD, Patrice NOUEL, Lilian BOSSU et Sylvie RENAUD : 'Conception de circuits numériques Langage VHDL pour la synthèse et la modélisation', Filière Electronique, 2006.

[34]Jimmy Royer, Thomas Djiako, Eric Schiller, Bocar Sada Sy. « Le pompage photovoltaïque, Manuel de cours à l'intention des ingénieurs et des techniciens ». Université d'Ottawa, Eric Schiller, 1998.

[35] JEAN-GABRIEL MAILLOUX : ' PROTOTYPAGE RAPIDE DE LA COMMANDE VECTORIELLE SUR FPGA À L'AIDE DES OUTILS SIMULINK - SYSTEM GENERATOR', Mémoire dans le cadre du programme de maîtrise d'ingénierie, l'université du QUÉBEC à CHICOUTIMI, MARS 2008.

Introduction générale

Un convertisseur continu-continu ou hacheur est un convertisseur statique qui permet d'alimenter une charge sous une tension continue réglable, à partir d'une source de tension continue (batterie par exemple) [1].

Les convertisseurs continu-continu, appelés aussi convertisseurs DC-DC constituent une partie essentielle des alimentations de puissance. En effet, on trouve ces convertisseurs dans de nombreuses applications à différents niveaux de puissance. Par exemple, à des faibles puissances (quelques watts) dans des chargeurs et alimentations de téléphones portables, à des puissances moyennes (quelques centaines de watts) dans des alimentations industrielles, et aussi à des fortes puissances (quelques kW), par exemple pour des applications de soudage électrique.

Les convertisseurs DC-DC ont donc un rôle très important dans les systèmes de conversion d'énergie. Pour ce type de fonction, plusieurs topologies de circuits sont proposées. Elles sont classifiées généralement par les topologies simples, qui sont les convertisseurs DC-DC de second ordre du point de vue de la modélisation mathématique, et les topologies complexes, qui sont ceux d'ordre plus élevé. Pour les topologies simples nous citons les convertisseurs de second ordre de type Buck (abaisseur), Boost (élevateur) et Buck-Boost (inverseur). Pour ceux d'ordre élevé, nous citons les convertisseurs Cuk et SEPIC (de l'anglais : Single-Ended Primary Inductor Converter) de quatrième ordre. Du point de vue de l'électronique de puissance, chacun de ces convertisseurs a des avantages et des inconvénients [2].

Parmi les convertisseurs DC-DC de base, un convertisseur Buck, ou hacheur série, est une alimentation à découpage qui convertit une tension continue en une autre tension continue de plus faible valeur. Le Buck présente beaucoup d'avantages et très peu d'inconvénients par rapport aux autres, un convertisseur Buck, ou hacheur série, est une alimentation à découpage qui convertit une tension continue en une autre tension continue de plus faible valeur.

Le principe d'un tel convertisseur est d'assurer la fluence d'énergie entre une source de tension continue, et une source de courant continu.

Les contrôleurs PID (Proportionnel, intégrateur, dérivateur) ont été largement utilisés dans les dernières cinq décennies à cause de leur simplicité, robustesse, efficacité et applicabilité dans une large gamme de systèmes. Malgré de nombreuses approches qui ont apparait dans la littérature, il est estimé que, de nos jours, les contrôleurs PID sont utilisés dans plus de 95% des processus industriels [16]. Pour plusieurs décennies le contrôleur PID numérique a été beaucoup utilisé pour la commande numérique en temps réel. Le PID est largement utilisé dans les entraînements, la robotique, la commande de la température et l'électronique de puissance. D'une manière générale, le PID est un outil très important pour le synthétiseur de commande numérique des systèmes embarqués en temps réel. Il est souvent implémenté soit en hardware en utilisant des composants analogiques ou en logiciel en utilisant des systèmes numériques [14].

Ce travail est organisé comme suit :

Le premier chapitre :

Dans ce chapitre une étude élargie sur les convertisseurs DC-DC sera développée, nous nous consacrons plus au convertisseur abaisseur (Buck) qui l'objet de notre travail.

Le deuxième chapitre :

Dans ce chapitre, les types de régulateurs, spécifiquement le régulateur PID basé sur les deux méthodes Ziegler-Nicolas et Hurwitz Polynominal seront étudiés et développés. Le calcul des différents coefficients des deux régulateurs seront détaillé.

Le troisième chapitre :

Dans le troisième chapitre, nous présentons les résultats de simulation du convertisseur DC/DC «Buck» contrôlé par un régulateur PID pour deux méthodes différentes Ziegler-Nicolas et Hurwitz Polynominal sous l'environnement Simulink/MatLab. La comparaison pour les deux méthodes sera discutée.

Le quatrième chapitre :

Dans le quatrième chapitre, on se consacre à l'implémentation sur cible FPGA du régulateur PID par les deux méthodes citées ci-dessus pour le contrôle du convertisseur DC/DC «Buck». Après un rappel large sur les circuits FPGA, les étapes de l'implémentation de notre régulateur sera développée. Après routage et simulation de notre circuit, nous présentons les résultats de simulation avec une comparaison avec ceux obtenus sous SIMULINK.

Nous terminons notre travail par une conclusion générale et perspective.