

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



UNIVERSITE AMAR TELIDJI DE LAGHOUCAT
FACULTE DE TECHNOLOGIE
Département d'Electrotechnique

Mémoire de fin d'étude

En vue de l'obtention du diplôme de **MASTER** en Génie électrique.

Option : Automatique et systèmes

Présenté par :

Labadi Ayoub

Nebeg Mohamed Adam

THEME

Contribution à la Commande d'un Convertisseur DC-DC

Soutenu devant le jury composé de :

MR. GUIBADJE MOSADEK	UAT-Laghouat	Président
Mr. SID AHMED BESSSEDIK	UAT-Laghouat	Examineur
Mr. AISSA AMMEUR	UAT-Laghouat	Encadreur

Promotion 2020

ملخص

تهدف هذه الرسالة الى دراسة شاملة لأنظمة التحكم القوية لمحوالات مستمر /مستمر DC/DC وتعرض فيها:

- التحكم في مستبدل رافع التوتر المستمر بدل رافع التوتر المستمر لمنظمات PID والمنطق الضبابي الشبكات العصبية
 - تقنية لتحسين التحكم القوي في مستبدل رافع التوتر المستمر لمنظمات PID والمنطق الضبابي والشبكات العصبية ثم استخدامه لبلوغ مرجع ثابت مع ضمان الأداء الجيد واستقرار النظام من ال تحقيق ذلك. تم استخدام خوارزمية النمل ACO و الخوارزمية الوراثةية GA لتحسن معاملات وحدة التحكم PID, المنطق الضبابي .. لتحسين أداء حلقة التحكم.
- الكلمات المفتاحية: منطق ضبابي – الشبكات العصبية – محول مستمر/مستمر -الخوارزميات التطورية -دفعه PID-محول رافع.

Abstract

This letter aims to study the synthesis of robust control systems for converter DC/DC series parallel or parallel series.

A robust control optimization technique of the Boost type converter; associating a PID regulator, fuzzy logic was used to achieve a constant reference for ensuring good performance and system stability. A GA and ant colony genetic algorithm was used to optimize the parameters of the PID controller to improve the performance of the control loop.

Key words: - RLF- RNA - ACO - DC / DC converter - PID converter Boost.

Résumé.

Ce mémoire visé la synthèse de systèmes de commande robustes pour les convertisseurs continue/continu. DC/DC de type série, parallèle out série parallèle.

Une technique d'optimisation de la commande robustes du convertisseur de type Boost ; associant un régulateur PID, la logique floue ; a été utilisée pour atteindre une référence constante pour en assurant des bonnes performances et la stabilité du système.

Un algorithme génétique GA et colonie de fourmis a été utilisé pour optimiser les paramètres du régulateur PID afin d'améliorer les performances de la boucle de régulation.

Mots clés : - RLF- RNA - ACO - convertisseur DC/DC - PID convertisseur Boost.

Remercîments

Tout d'abord je remercie dieu tout puissant qui m'a gardé en bonne santé afin de faire ce modeste travail et qui m'a permis de vivre ce bonheur.

*Tous mes vifs remerciement vont vert notre promoteur **Mr. Ameur Aïssa** Pour ces conseils chaleureux son encadrement, sa disponibilité et son aide précieuse pour réaliser ce modeste travail, et je n'oublie pas **Mr. Bessedî Sid Ahmed** Pour son aide, ses conseil précieux.*

Pour cela je l'assure de ma sincère gratitude et je respecte le plus profond.

Je remercie les membres du jury qui m'ont fait l'honneur d'examiner mon travail.

Merci à toute personne ayant contribué de près ou de loin a la concrétisation de ce projet.

Dédicace

Grace à ma bonne volonté, mon acharnement, dieu tout puissant m'a donné la force et le courage pour la réalisation de mon mémoire.

Je dédie ce travail à ma très chère famille. À mes très chers partants, en témoignage et en gratitude de leurs dévouements et leurs soutiens permanent durant tous mes années d'études, leurs sacrifices illimités, leur réconfort moral et tous les efforts qu'ils ont consentis pour mon éducation et mon instruction pour me voir réussir un jour. Mes frères

Abréviation

PI	proportional integral
RNA	Réseaux de neurone
RLF	Logique floue
DC	Direct courant
ACO	Ant Colonie Optimisation
GA	Génétique Algorithmes

Symboles

L	Inductance
C	Capacité
R	Résistance
Vref	Tension de référence
e(t)	L'erreur entre la variable et sa référence
x1,x2	Le courant dans l'inductance
Ts	Période de commutation
fs	Fréquence de commutation
Ton	Temps de l'état on (transistor)
Toff	Temps de l'état off (transistor)
u	Rapport cyclique
Vs	Tension de sortie
Vl	Tension d'inductance
Vc	Tension aux bornes de la capacité
Vd	Tension aux bornes de la diode
D	Diode
E	Source de tension

Liste des tableaux

Titre	Page
Tableau II.7. Synthèse sur les actions PID.	24
Tableau II.9.1. les paramètres de système et le régulateur	25
Tableau II.4. Les paramètres de système et le régulateur.	37
Tableau III .1. Paramètres de ACO	44

Liste de figure

Titre	Page
Chapitre I	
Figure I .1. Le schéma de principe d'un hacheur.	3
Figure I.2 : Principe de base d'un hacheur	4
Figure I.3: Schéma de base d'un hacheur Buck.	4
Figure I.4: la période de commutation de transistor Q.	5
Figure I.5: Schéma équivalent du hacheur buck , état ON (Séquence1).	5
Figure I.6 : Schéma équivalent du hacheur Buck , l'état OFF (Séquence 2).	6
Figure I.7 : Schéma de base d'un hacheur Boost.	8
Figure I.8 : Schéma équivalent de l'hacheur boost, l'état ON. (Séquence 1).	9
Figure I.9 : Schéma équivalent du hacheur boost, l'état OFF (séquence 2).	10
Figure I.10 : Schéma de base d'u hacheur Buck-Boost.	12
Figure I.11 : Schéma équivalents du hacheur buck-Boost , l'état ON (séquence 1).	12
Figure I.12: Schémas équivalents du hacheur buck-boost, l'état OFF (séquence 2).	13
Chapitre II	
Figure II .1Schéma de principe d'une boucle de régulation	16
Figure II.2. Système en Boucle ouverte	17
Figure II.3. Système en Boucle fermé	17
Figure II.4. Représentation de (Tm, Ts, D)	18
Figure II.5. Régulateur PID	19
Figure II.6. La structure parallèle	20
Figure II.7. La structure mixte	21
Figure II.8. La structure série	21
Figure II.9. Schéma bloc de système de 3eme ordre commandé par les différentes structures de PID.	22
Figure II.10. La réponse du système de 3eme ordre commandé par PID	23
Figure II.11. Réponse d'un système avec tous les autres régulateurs	23
Figure II.12. Simulation d'hacheur boost avec PI control	26
Figure II.13. Tension de sortie du système	26

Figure II.14. Courant de l'inductance du système	27
Figure II.15. : représentation du neurone artificiel	28
Figure II.16. Simulation d'hacheur boost avec commande RNA	30
Figure II.17. Tension de sortie (RNA)	31
Figure II.18. Les tensions du sortie pour (PI , RNA)	31
Figure II.19. Structure de base d'un contrôleur flou	33
Figure II.20. Schéma synoptique d'un système flou	33
Figure II.21. Définition de la structure globale (entrée, sortie, type d'opérateurs)	35
Figure II.22. Définition des fonctions d'appartenances pour(V), (Vref) et(Y). Les règles sont saisies à l'aide d'un formulaire représenté sur la Figure. II.23	35
Figure II.23. Écran de saisie des règles	36
Figure II.24. Simulation d'hacheur boost avec commande RLF	36
Figure II.25. Tension de sortie du système (RLF)	37
Figure II.26. Courant de l'inductance du système (RLF)	38
Figure II.26. Les tensions de la sortie pour (PI, RLF)	38
Chapitre III	
Figure III.1 - Fonctionnement courant d'un algorithme génétique	46
Figure III.2. Tension de sortie de system avec PI	47
Figure III.3. Tension de sortie de system avec PI a optimise par ACO	47
Figure III.4. Tension de sortie de system avec PI a optimise par GA	48
Figure III.5. Les tensions de sortie pour (PI, PI ACO, PI GA)	48
Figure III.6. Les tensions de sortie pour (PI, PI ACO, PI GA)	49
Figure III.7. Tension de sortie de system avec RLF optimise par GA	49
Figure III.8. Tension de sortie de system avec RLF optimise par ACO	50
Figure III.9. Les tensions de sortie pour (RLF, RLF ACO, RLF GA)	50
Figure III.10. Les tensions de sortie pour (RLF, RLF ACO, RLF GA)	51

Sommaire

Titre	Page
Résumé	I
Remercîment	IV
Dédicace	V
Abréviation	VI
Symboles	VII
Introduction général	1
Chapitre I	
I.1. Introduction	3
I.2. Les convertisseurs DC/DC (Hacheurs)	3
I.3. L'hacheur Buck	4
I.3.1 Topologie	4
I.3.2. Principe de fonctionnement	5
I.4. Modélisation	7
I.5. L'hacheur boost	8
I.5.1. Topologie	8
I.5.2. Principe de fonctionnement	8
I.6. Modélisation	10
I.7. L'hacheur Buck-Boost	11
I.7.1 Topologie	11
I.7.2. Principe de fonctionnement	12
I.8. Modélisation	13
I.9. Conclusion	15
Chapitre II	
II .1. Introduction	16
II .2. Eléments constitutifs d'une boucle de régulation	16
II .2.1. Une boucle de régulation	16
II .3. Les types de boucles	17
II 3.1. Système en boucle ouverte	17
II 3.2. Système en boucle fermée	17

II .4 Caractéristiques dynamiques d'un procédé	18
II .4.1. Temps de réponse	18
II .4.2. Temps de montée	18
II.4.3. Dépassement	18
II .5. Les régulateurs PID	19
II.5.1 L'algorithme de base	20
II .6. Les différentes structures du PID	20
II .6.1. La structure parallèle	20
II .6.2. La structure mixte	21
II .6.3. La structure série	21
II .6.4. Schéma bloc	21
✓ A réponse d'un système	23
II .7. Synthèse sur les actions PID	24
II .8. La Commande PI	24
A. Action proportionnelle	24
B. Action intégrale	24
II .8.1. Caractéristiques des régulateurs PI	25
II .9. Résultat et simulation	25
II .9.1. Simulation du convertisseur Boost	25
II .9.2. Interprétation	27
II .10. La commande par réseaux de neurones artificiels (RNA)	27
II .10.1. Neurone formel	27
II .10.2. La fonction d'activation (transfert)	28
II .10.3. Définition des réseaux de neurones artificiels (RNA)	29
II .10.4. Apprentissage	29
II .10.5. Apprentissage supervisé	29
II .10.6. Apprentissage non supervisé	29
II .10.7. La rétropropagation	30
II .11. La commande par logique floue.	32
II .11.1 Les différentes parties de la commande floue.	32
II .11.2 Structure d'un système flou.	33

II .11.3. Application de la commande floue.	34
II.11.4. Simulation d'hacheur Boost avec RLF	36
II.12. Conclusion	39
Chapitre III	
III.1 Introduction	40
III.2 Optimisation par colonie de fourmis (ACO)	40
III.2.1. Historique	40
III.2.2. Similarités et analogies avec les fourmis réelles	40
A. Points communs	41
B. Analogie	42
III.2.3. Algorithme de base AS : 'Ant system'	42
• Algorithme d'optimisation ACO : « Ant Colony Optimization »	44
III.3. Algorithmes génétiques GA	44
III.3.1. Principes et généralités	44
III .3.2. Fonctionnement d'algorithme génétique	45
III .3.3. Résultats des simulations	47
III.4. la Comparaisent des résultats de simulation (PI, PI ACO, PI GA)	48
III.5. la Comparaisent des résultats de simulation (RLF, RLF ACO, RLF GA)	50
III.6. Interprétation	51
III .7. Conclusion	51
Conclusion général	52
Reference bibliographie	53
Annexe	a

Introduction générale

Introduction générale

L'utilisation des convertisseurs de puissance pour des différentes applications devient de plus en plus importante. Parmi ces domaines d'application on cite : l'industrie automobile (avec l'apparition de véhicules hybrides et électriques), les applications domestiques (téléphone mobile, ordinateurs, électroménager), l'aéronautique (l'avion électrique), le domaine ferroviaire, les énergies renouvelables (panneaux solaires photovoltaïques, éoliennes) et réseaux de transport d'énergie électrique (les liaisons à courant continu). En plus ces convertisseurs sont caractérisés par leur faible cout, leur grande simplicité et leur disponibilité [1].

Les convertisseurs DC-DC sont des dispositifs qui sont utilisés chaque fois que nous voulons changer une alimentation électrique à courant continu d'un niveau de tension à un autre. Ces convertisseurs utilisent des commutateurs électroniques (MOS FET) pour la conversion de tension. L'objectif principal est de modifier l'énergie en courant continu à partir d'un niveau particulier à d'autres niveaux tout en minimisant les pertes.

La nécessité pour les convertisseurs de ce type a augmenté en raison du fait que les transformateurs sont incapables de fonctionner sur courant continu.

La commande des convertisseurs DC-DC est un sujet très discuté et très investi. En effet, plusieurs techniques de commande ont été, alors, appliquées pour commander ces convertisseurs, ces commandes ont souvent pour objectifs : une excellente stabilité globale du système en boucle fermée, une meilleure poursuite de la référence d'entrée et un bon fonctionnement.

Dans notre travail, on exploitera plusieurs techniques des commandes (RNA et RLF) pour commander ces convertisseurs DC-DC, avec des méthodes d'optimisation.

Le mémoire est structuré de la manière suivante :

- ✓ Le premier chapitre porte sur la modélisation détaillée des convertisseurs DC-DC, on présentera, au cours de ce chapitre, les différentes topologies de ces convertisseurs buck, boost et buck-boost, on expliquera également leurs principes de fonctionnement
- ✓ Le deuxième chapitre présente les techniques de l'intelligence artificielle, la commande par ' logique floue, réseaux de neurones ' pour développer les régulateur classique PI.

- ✓ Le troisième chapitre sera consacré pour la présentation des résultats obtenus par simulation après l'optimisation par la méthode (ACO et GA) et par expérience avec des interprétations, comparaison des résultats de simulation et validation expérimentale avec le régulateur PI.

Chapitre I : Généralité sur les

I.1. Introduction

Le présent chapitre traite de la conversion DC/DC, qui a pour but d'assurer l'influence d'énergie entre une source de tension continue et une source de courant continu.

Pour ce type de convertisseur statique, la gamme de puissance que l'on peut « traiter » s'étend de quelques watts, à une trentaine de kilowatts environ. Cependant, l'apparition de composants de puissance sans cesse plus performants, tout comme la définition de nouvelles structures (multi niveau en particulier), permettent d'envisager le traitement de puissances plus élevées. Nous distinguons deux types de convertisseurs continus/continus :

Les convertisseurs appelés ((hacheurs)), qu'ils soient abaisseurs, éleveurs ou bien dévolteurs {survolteurs, qui correspondent aux applications moyennes et fortes puissances, les alimentations à découpages, qui correspondent aux applications petites puissances de la conversion DC/DC .

Dans ce chapitre on traite les différents types d'hacheurs.

I .2. Les convertisseurs DC/DC (Hacheurs) :

L'hacheur continue/continue est un convertisseur permettant de convertir une énergie continue d'un niveau donné de tension (ou du courant) pour un autre niveau d'énergie continue de la tension (ou du courant). Il est utilisé pour stocker l'énergie photovoltaïque dans des batteries, ainsi que pour alimenter une charge continue se compose de condensateurs, d'inductance et de commutateurs. Dans le cas idéal, tous ces dispositifs ont un bon rendement, ils ne consomment aucune puissance active,

Les hacheurs sont des convertisseurs continus qui procèdent par découpage d'une grandeur d'entrée continue, tension ou courant et dont la grandeur de sortie est également continue ou à faible ondulation [2].

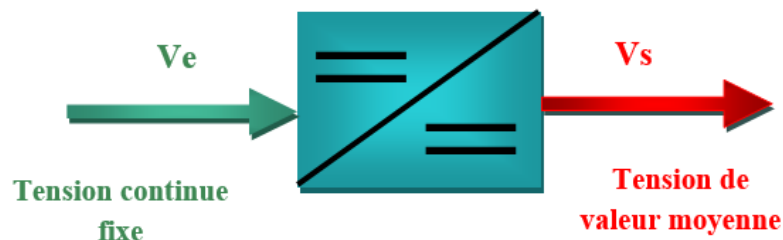


Figure I .2. Le schéma de principe d'un hacheur.

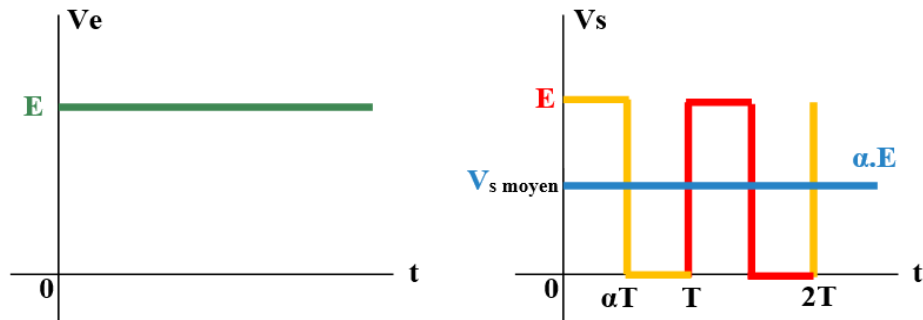


Figure I.2.1 : Principe de base d'un hacheur

Les différents types des hacheurs :

Selon la position du commutateur et de l'hacheur, différents types de convertisseurs de tension peuvent être réalisés :

- ✓ Abaisseur de tension « Buck »
- ✓ Élévateur de tension « Boost »
- ✓ Abaisseur élévateur de tension « Buck-Boost »

I.3. L'hacheur Buck

I.3.1 Topologie

L'hacheur série (Buck) permet de transformer une tension continue fixe en une tension d'alimentation plus faible réglable par l'utilisateur. **La Figure I.3.1** présente le schéma de base d'un hacheur Buck. Cet hacheur est formé d'une source de tension continue V_0 à l'entrée, d'une diode de roue libre D , d'un transistor Q commandable de puissance, d'une inductance L , d'un condensateur C et d'une résistance de charge R [5].

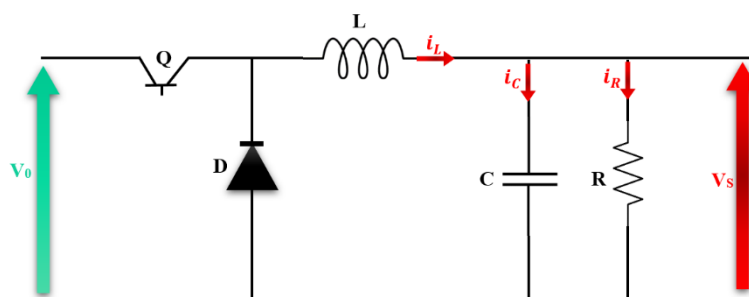


Figure I.3.1: Schéma de base d'un hacheur Buck.

I.3.2. Principe de fonctionnement

Le transistor Q est commandé en commutation à la période T_s avec un rapport cyclique u comme l'indique La **Figure I.3.2** représentant le signal de commande MLI. Ce rapport cyclique est défini comme $u = t_{on} / T_s$, tel que t_{on} : temps de fermeture de Q

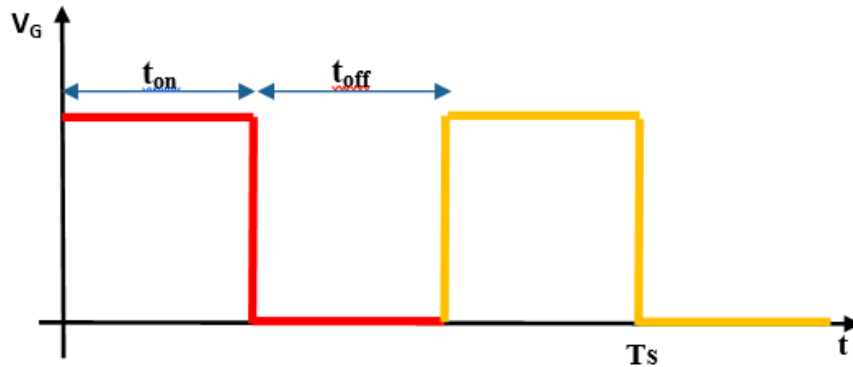


Figure I.3.2: la période de commutation de transistor Q.

On considère que le transistor Q et la diode D sont parfaits. On peut alors distinguer deux phases de fonctionnement :

La phase active : lorsque le transistor Q est à l'état ON pendant la période t_{on} . La source d'entrée fournit l'énergie à la charge R à travers l'inductance L. le schéma équivalent durant la phase active est illustré sur **La Fig. I.3.3** [7].

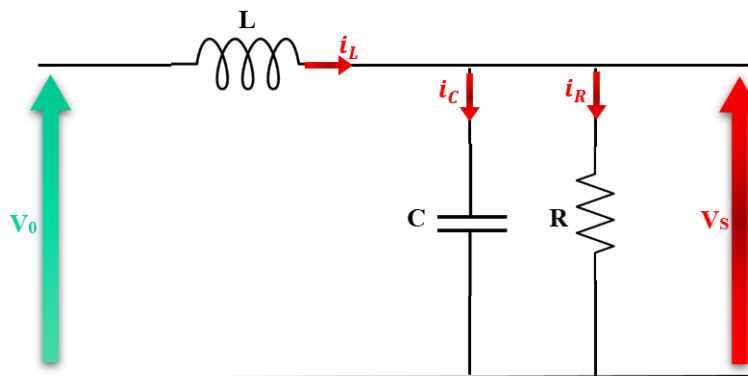


Figure I.3.3: Schéma équivalent du hacheur buck , état ON (Séquence1).

Séquence 1 : À l'instant $t = 0$, l'interrupteur Q est fermé pendant une durée t_{ON} on obtient :

$$V_L(t) = L \frac{di}{dt} = V_0 - V_C \quad (1.1)$$

$$C \frac{dv}{dt} = i_1 - \frac{v_C}{R} \quad (1.2)$$

La phase de roue libre : lorsque le transistor Q est bloqué, la diode D assure la continuité du courant dans l'inductance L, l'énergie emmagasinée dans cette inductance est alors déchargée dans le condensateur C et la résistance de charge R. le schéma équivalent durant la phase de roue libre est illustré sur **la Fig I.3.4**.

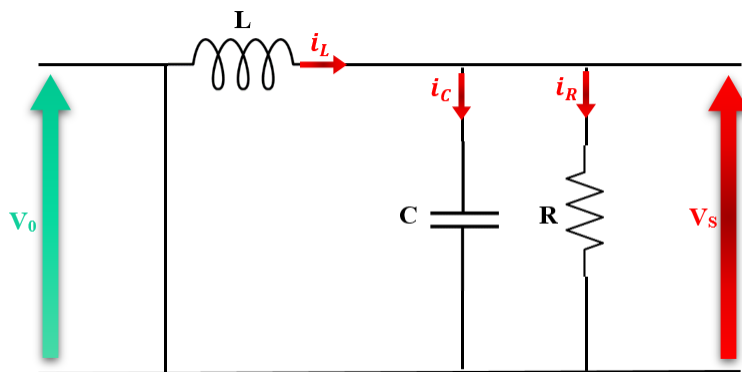


Figure I.3.4 : Schéma équivalent du hacheur Buck , l'état OFF (Séquence 2).

Séquence 2 : À l'instant $t = t_{ON}$, l'interrupteur Q est à l'état OFF pendant une durée de T_S ($1 - u$).

$$V_L(t) = L \frac{di}{dt} = V_C \quad (1.3)$$

$$C \frac{dv}{dt} = i_1 - \frac{v_C}{R} \quad (1.4)$$

On engendre aussi que la tension moyenne aux bornes de l'inductance $\langle V_L \rangle$ est nulle, Par conséquent on a

$$V_S = uV_0 \quad (1.5)$$

Avec le rapport cyclique u étant comprise entre zéro et un, et la tension de sortie V_S peut-être varier entre zéro et V_0

I.4. Modélisation

Dans de tels systèmes, l'ordre d'équations d'états est égal au nombre d'éléments de stockage d'énergie. Dans le cas présent l'ordre $n=2$. Dans le cas où le système considéré est linéaire, la représentation d'état se met sous la forme ci-après [6] :

$$\dot{x} = Ax + BU \quad (1.6)$$

$$y = C_1x \quad (1.7)$$

Où : x est le vecteur d'état ; $x = [x_1, x_2] = [i_l, V_C]$.

U : vecteur de commande, A : matrice d'état, B : matrice d'entrée et C : matrice de sortie.

Equations d'état prendre la forme suivante en fonction de la séquence en cours :

Séquence 1 : $t \in [0, t_{on}]$

$$\begin{cases} V_L(t) = L \frac{di}{dt} = V_0 - V_C \\ C \frac{dv}{dt} = i_l - \frac{V_C}{R} \end{cases} \quad (1.9)$$

$$\begin{cases} \dot{x}_1 = \frac{1}{L}V_0 - \frac{1}{L}x_2 \\ \dot{x}_2 = \frac{1}{C}x_1 - \frac{1}{RC}x_2 \end{cases} \quad (1.10)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} V_0 \quad (1.11)$$

Séquence 2 : $t \in [t_{on} \text{ à } T_s]$

$$\begin{cases} V_L(t) = L \frac{di}{dt} = -V_0 \\ C \frac{dv}{dt} = i_l - \frac{V_C}{R} \end{cases} \quad (1.12)$$

$$\begin{cases} \dot{x}_1 = -\frac{1}{L}x_2 \\ \dot{x}_2 = \frac{1}{C}x_1 - \frac{1}{RC}x_2 \end{cases} \quad (1.13)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} V_0 \quad (1.14)$$

À partir des séquences 1 et 2 on obtient le modèle dynamique d'un convertisseur Buck :

$$\dot{x}_1 = -\frac{1}{L}x_2 - \mu \frac{V_0}{L} \quad (1.15.a)$$

$$\dot{x}_2 = \frac{1}{C}x_1 - \frac{1}{RC}x_2 \quad (1.15.b)$$

Où x_1 et x_2 représentant le courant I_L dans l'inductance et la tension V_C aux bornes du condensateur.

I.5. L'hacheur boost

I.5.1. Topologie

L'hacheur parallèle (boost) est un convertisseur statique permettant de convertir une tension continue constante à une autre tension continue réglable de valeur moyenne plus grande. La **Figure I.5.1** présente schéma de base d'un hacheur Boost. Il est constitué d'un transistor Q commandable de puissance en parallèle avec la tension d'entrée V_0 , ainsi d'une diode D, d'une inductance L, d'un condensateur C et d'une résistance de charge R.

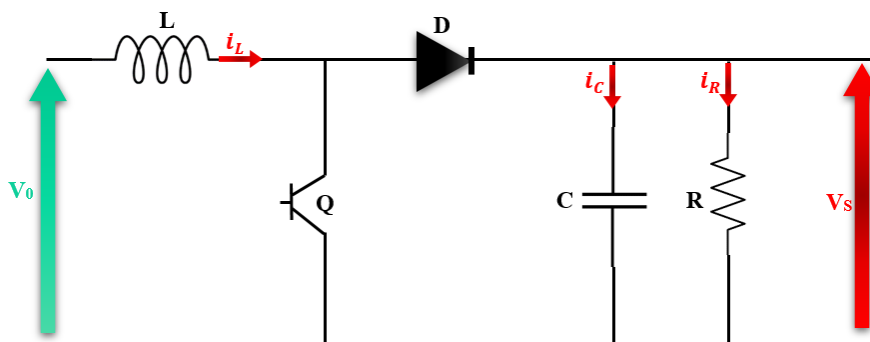


Figure I.5.1 : Schéma de base d'un hacheur Boost.

I.5.2. PRINCIPE DE FONCTIONNEMENT :

Le transistor Q est commandé en commutation à la période T_S avec un rapport cyclique μ comme l'indique la figure précédente. La **Figure I.5.1**, on admet que les composants Q transistor et la diode D sont parfaits, on peut distinguer deux phases de fonctionnement :

La phase active : lorsque le transistor Q est à l'état on et la diode D est ouverte. Durant cette séquence, le courant traversant l'inductance L va augmenter linéairement. Le condensateur C fournit de l'énergie à la charge R. le schéma équivalent durant la phase active est illustré sur La **Figure I.7.[7]**.

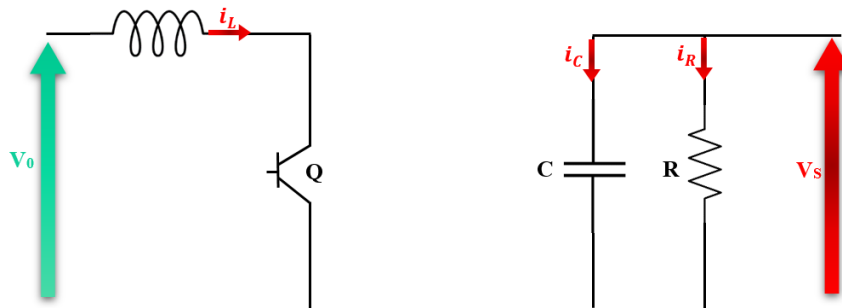


Figure I.7 : Schéma équivalent de l'hacheur boost, l'état ON. (Séquence 1).

Séquence 1 : À l'instant $t = 0$, le transistor Q est fermé pendant une durée t_{ON} , La tension aux bornes de la diode D est alors égale à $V_D = V_A - V_K$. On a $V_D = 0$, ce qui implique que la diode est polarisée en inverse $V_D = -V_0$. Dans ces conditions, on obtient :

$$v_l(t) = l \frac{di}{dt} = v_D \quad (1.16)$$

$$C \frac{dV}{dt} = -\frac{V_S}{R} \quad (1.17)$$

La phase de roue libre : lorsque Le transistor a l'état OFF et la diode D est fermée. Durant cette séquence, l'énergie emmagasinée dans l'inductance L est restituée au condensateur et à la charge R. Cette décharge n'est possible que $V_S > V_0$, lors de cette phase, le fait que l'inductance L soit en série avec la source de tension d'entrée permet d'obtenir un montage survolteur ou élévateur.

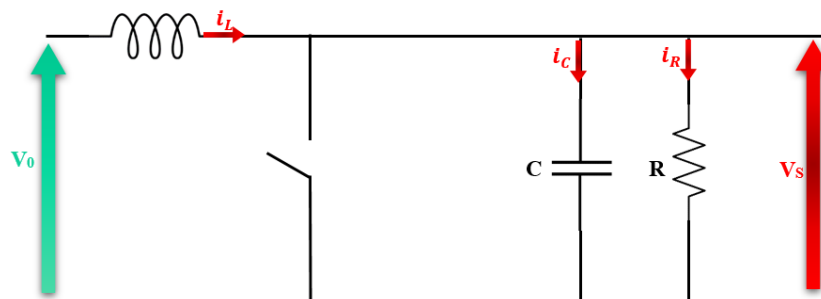


Figure I.8 : Schéma équivalent du hacheur boost, l'état OFF (séquence 2).

Séquence 2 : À l'instant $t = t_{ON}$, on et pendant une durée $T_S (1 - u)$ on obtient les équations (1.1) et (1.2)

On engendre aussi que la tension moyenne aux bornes de l'inductance $\langle V_L \rangle$ est nulle, Ce qui impose, Par conséquent on a :

$$V_s = \frac{V_0}{1-u} \quad (1.18)$$

I.6. MODELISATION

L'ordre de l'équation d'état de ce type de convertisseur est aussi égal au nombre d'éléments de stockage ($n=2$). Dans le cas où le système considéré est linéaire. Equations d'état :

Séquence 1 : $t \in [0 \text{ à } t_{ON}]$

$$\begin{cases} V_L(t) = L \frac{di}{dt} = V_D \\ C \frac{dv}{dt} = -\frac{V_c}{R} \end{cases} \quad (1.19)$$

$$\begin{cases} \dot{x}_1 = \frac{1}{L} V_0 \\ \dot{x}_2 = -\frac{1}{RC} x_2 \end{cases} \quad (1.20)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{-1}{RC} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V_0 \quad (1.21)$$

Séquence 2 : $t \in [t_{ON} \text{ à } T_S]$

$$\begin{cases} V_L(t) = L \frac{di}{dt} = V_0 - V_c \\ C \frac{dv}{dt} = i_l - \frac{V_c}{R} \end{cases} \quad (1.22)$$

$$\begin{cases} \dot{x}_1 = \frac{1}{L} V_0 + \frac{1}{L} V_c \\ \dot{x}_2 = \frac{1}{C} x_1 - \frac{1}{RC} x_2 \end{cases} \quad (1.23)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V_0 \quad (1.24)$$

À partir des séquences 1 et 2 on obtient le modèle dynamique d'un convertisseur boost :

$$\dot{x}_1 = (1 - \mu) \frac{1}{L} x_2 + \frac{1}{L} E \quad (1.25a)$$

$$\dot{x}_2 = -(1 - \mu) \frac{x_1}{C} + \frac{x_2}{RC} \quad (1.25b)$$

Où x_1 et x_2 représentant le courant i_L dans l'inductance et la tension V_C aux bornes du condensateur.

I.7. L'HACHEUR BUCK-BOOST

I.7.1 TOPOLOGIE

Le hacheur série parallèle ou inverseur est un convertisseur indirect DC-DC à stockage inductif. Son rôle principal est de convertir la tension constante d'entrée en une tension supérieure ou inférieure mais de polarité inverse. La source d'entrée est de type tension continue (filtrage capacitif en parallèle avec une source de tension) et la Charge de sortie continue de type source de tension (condensateur en parallèle avec la charge résistive). Le transistor doit être commandé par blocage et l'amorçage. La **Figure I.9** présente Schéma de base d'un hacheur (Buck-Boost).

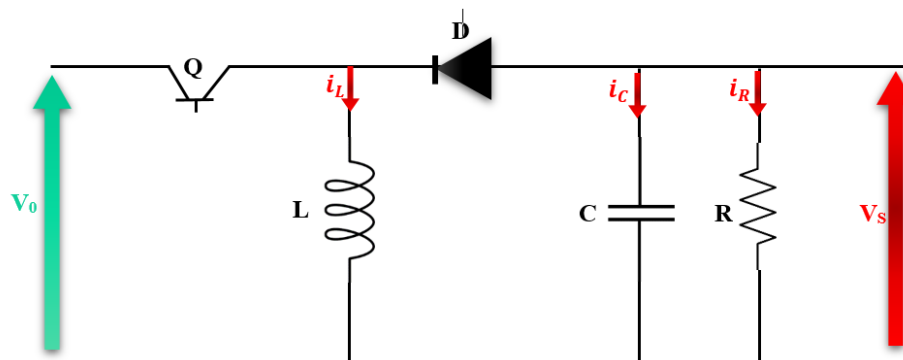


Figure I.9 : Schéma de base d'u hacheur Buck-Boost.

I.7.2. PRINCIPE DE FONCTIONNEMENT

Le fonctionnement d'un convertisseur Buck-Boost peut être divisé en deux configurations suivant l'état de transistor Q . On peut distinguer deux phases de fonctionnement.

La phase active :

Le transistor a l'état on, conduisant ainsi à une augmentation de l'énergie stockée dans l'inductance. La charge est déconnectée du montage parce que la diode D est bloquée, le schéma équivalent durant la phase active est illustré sur la **Figure I.10**

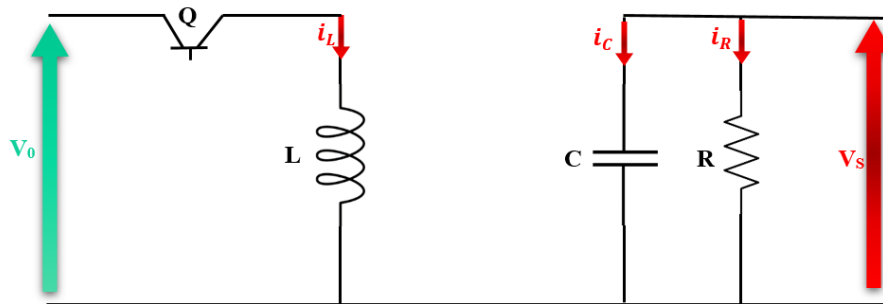


Figure I.10 : Schéma équivalents du hacheur buck-Boost , l'état ON (séquence 1).

Séquence 1 : À l'instant $t = 0$, le transistor Q est à l'état ON pendant une durée t_{ON} , La dynamique du circuit est décrite par les équations ci-dessous :

$$V_L(t) = L \frac{di}{dt} = V_0 \quad (1.25)$$

$$C \frac{dv}{dt} = -\frac{V_S}{R} \quad (1.26)$$

La phase de roue libre :

Le transistor a l'état OFF, la diode D assure la continuité du courant dans l'inductance. L'énergie emmagasinée dans cette dernière est alors déchargée dans le condensateur et la résistance de charge R, le schéma équivalent durant la phase active est illustré sur **La Figure I.11**.

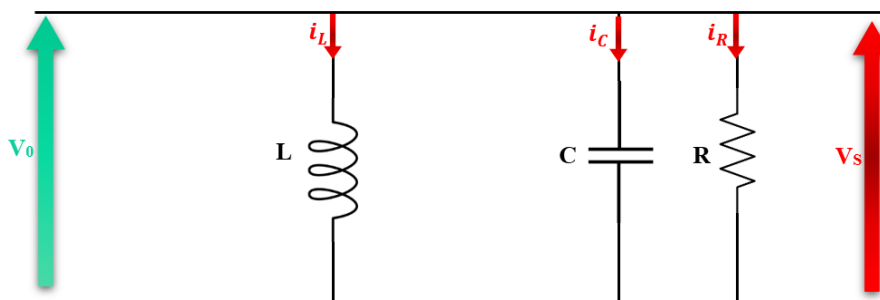


Figure I.11: Schémas équivalents du hacheur buck-boost, l'état OFF (séquence 2).

Séquence 2 : À l'instant $t = t_{ON}$, on et pendant une durée $T_S (1 - u)$ on obtient :

$$V_L(t) = L \frac{di}{dt} = V_s \quad (1.27)$$

$$C \frac{dv}{dt} = -i_t - \frac{V_s}{R} \quad (1.28)$$

On engendre aussi que la tension moyenne aux bornes de l'inductance est nulle, Par conséquent on a :

$$V_s = -\frac{u}{1-u} V_D \quad (1.29)$$

I.8. MODELISATION

L'ordre de l'équation d'état de ce type de convertisseur est aussi égal au nombre d'éléments de stockage ($n=2$). Dans le cas où le système considéré est linéaire.

Séquence 1 : $t \in [0 \text{ à } t_{ON}]$

$$\begin{cases} V_L(t) = L \frac{di}{dt} = V_0 \\ C \frac{dv}{dt} = -\frac{V_C}{R} \end{cases} \quad (1.30)$$

$$\begin{cases} \dot{x}_1 = \frac{1}{L} V_0 \\ \dot{x}_2 = -\frac{1}{RC} x_2 \end{cases} \quad (1.31)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V_D \quad (1.32)$$

Séquence 2 : $t \in [t_{ON} \text{ à } T_S]$

$$\begin{cases} V_L(t) = L \frac{di}{dt} = V_0 \\ C \frac{dv}{dt} = -i_l - \frac{V_C}{R} \end{cases} \quad (1.33)$$

$$\begin{cases} \dot{x}_1 = \frac{1}{L}x_2 \\ \dot{x}_2 = -\frac{1}{C}x_1 - \frac{1}{RC}x_2 \end{cases} \quad (1.34)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{L} \\ -\frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} V_0 \quad (1.35)$$

À partir des séquences 1 et 2 on obtient le modèle dynamique d'un convertisseur buck-boost :

$$\dot{x}_1 = -(1 - \mu) \frac{x_2}{L} + \frac{E}{L} \quad (1.36)$$

$$\dot{x}_2 = -(1 - \mu) \frac{x_1}{C} + \frac{x_2}{RC} \quad (1.37)$$

Où x_1 et x_2 représentant le courant I_L dans l'inductance et la tension V_C aux bornes du condensateur.

I.9. Conclusion

Au cours de ce chapitre nous avons présenté les différentes topologies des convertisseurs DC-DC, ainsi leur principe de fonctionnement, nous avons aussi présenté la modélisation détaillée des hacheurs Buck, boost et Buck-Boost dans les deux intervalles du temps de signal de commande (commutateur fermé puis commutateur ouvert), Dans le chapitre qui suit, nous allons présenter les nouvelles techniques de commande.

Chapitre II : Techniques de commande d'un convertisseur DC/DC

II .1. Introduction

Avec la complexité des nouvelles installations industrielles, les commandes classiques deviennent incapables de gérer ces installations, pour surmonter ce problème, la tendance des recherches actuelles est vers les techniques de l'intelligence artificielle.

Dans ce chapitre nous allons discrète sur du développement théorique des régulateurs classiques PI et les régulateurs basé sur les techniques de l'intelligence artificielle, la commande par ' logique floue, réseaux de neurones '.

II .2. Eléments constitutifs d'une boucle de régulation

II .2.1. Une boucle de régulation

Doit comporter au minimum les éléments suivants

- Un capteur de mesure,
- Un transmetteur souvent intégré au capteur,
- Un régulateur ; un actionneur.

Elle est souvent complétée par : un enregistreur, des convertisseurs, des sécurités.

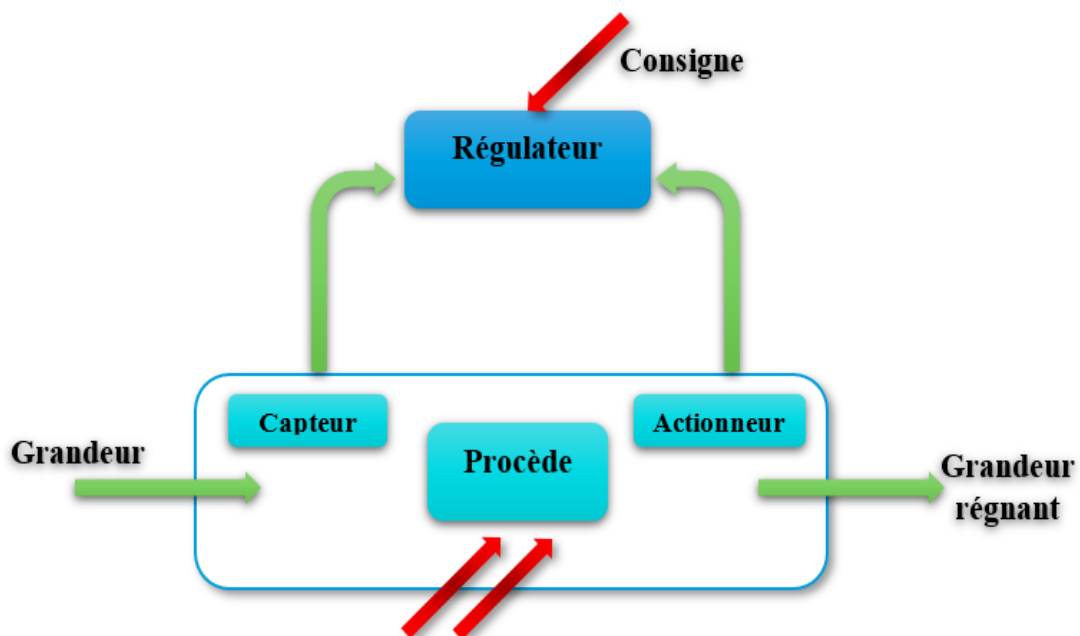


Figure II .1Schéma de principe d'une boucle de régulation.

Le régulateur reçoit deux informations :

- Le signal de mesure provenant du capteur,
- La consigne (qui peut être locale ou externe),

Deux valeurs et de l'algorithme de calcul pour lequel il a été configuré, il délivre un signal de sortie dirigé vers l'actionneur afin d'annuler cet écart et de ramener la mesure vers la valeur de consigne. [8]

II .3. Les types de boucles

II 3.1. Système en boucle ouverte

Un système en boucle ouverte est système qui ne comporte pas de contre-réaction (feedback) entre la sortie et l'entrée, classiquement il est composé du processus physique, d'un capteur pour en mesurer la sortie et d'un actionneur pour agir sur la grandeur d'entrée du processus.



Figure II.2. Système en Boucle ouverte.

II 3.2. Système en boucle fermée

Une boucle est dite fermée si la grandeur de la mesure affecte la grandeur de la correction (grandeur manipulée).

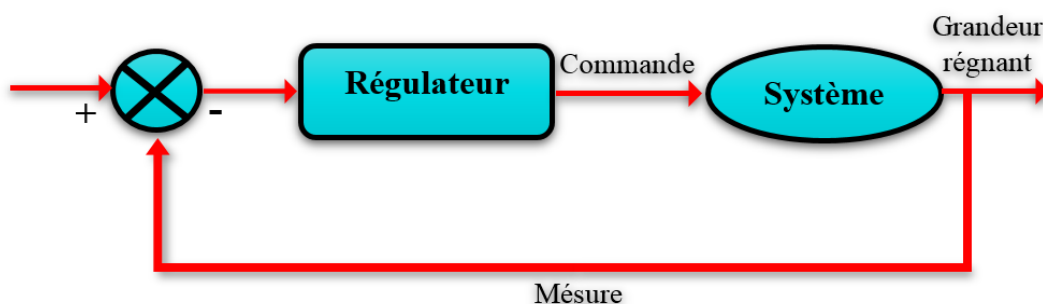


Figure II.3. Système en Boucle fermé.

II .4 Caractéristiques dynamiques d'un procédé

II .4.1. Temps de réponse

C'est l'aptitude du procédé à suivre les variations de la consigne. Dans le cas d'un échelon de la consigne ; la croissance de la grandeur réglée définit les différents temps de réponse. Dans l'exemple ci-dessous de la figure, on mesure le temps de réponse à $\pm 5\%$ qui est égal à t_1-t_0 .

II .4.2. Temps de montée

Le temps de montée est l'intervalle de temps séparant les instants auxquels la réponse indicielle vaut 10% et 90% de la valeur finale.

II.4.3. Dépassement

Le premier dépassement permet de qualifier la stabilité d'un système. Plus celui-ci sera important, plus le système sera proche de l'instabilité. Dans certaines régulations, aucun dépassement n'est toléré, dans d'autre un dépassement inférieur à 15 % est considéré comme acceptable. Dans la réponse indicielle ci-dessous, le premier dépassement est de 14%.

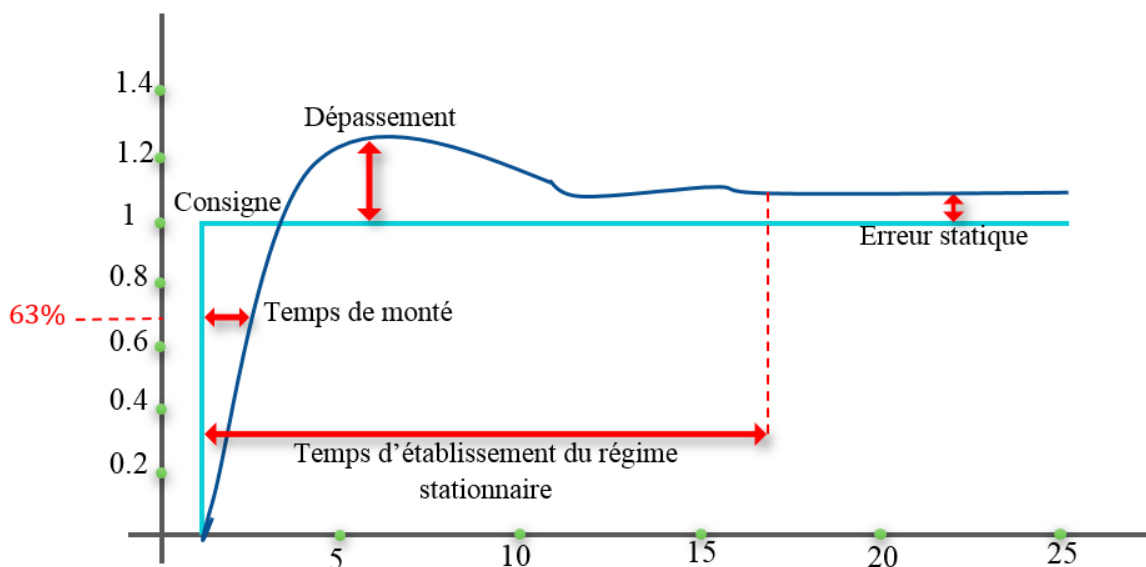


Figure II.4. Représentation de (T_m , T_s , D).

II .5. Les régulateurs PID

Le régulateur PID (Proportionnel, Intégral, Dérivé) C'est un système d'auto régulation (boucle fermée), qui cherche à réduire l'erreur entre la consigne et la mesure. [9]

$e = \text{consigne} - \text{mesure}$

Les régulateurs PID répondent à plus du 90% 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 P, I et D ne sont pas toujours satisfaisantes, ni adaptées au processus à régler. L'histoire des régulateurs est déjà longue et il peut être intéressant de rappeler quelques étapes importantes. Les premiers régulateurs de type centrifuge apparaissent vers 1750 pour régler la vitesse des moulins à vent, suivi en 1788 du fameux contrôleur de vitesse d'une machine à vapeur de James Watt.

En 1942, Ziegler et Nichols ont proposé deux démarches permettant de trouver facilement les paramètres optimums pour une installation donnée. Au fil des ans, les propositions de Ziegler et Nichols ont été adaptées ou modifiées selon les besoins. [9]

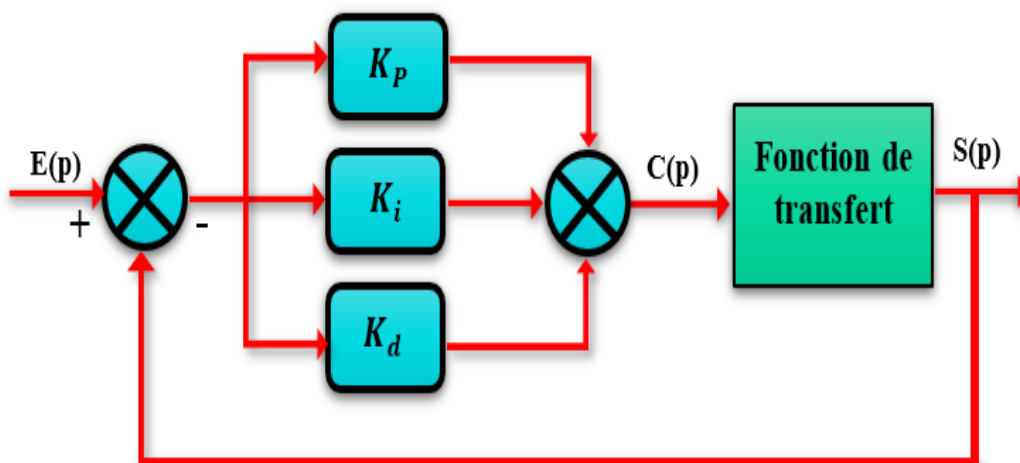


Figure II.5. Régulateur PID.

II.5.1 L'algorithme de base

L'équation du PID est donnée comme suit :

$$U(t) = K[e(t) + \frac{1}{T_i} \int_0^t e(s) ds + T_d \frac{de(t)}{dt}] \quad (2.2)$$

U : la variable de commande.

e : le signal d'erreur.

T_i, T_d : constant de temps.

K : le gain de la partie proportionnelle.

II .6. Les différentes structures du PID

II .6.1. La structure parallèle

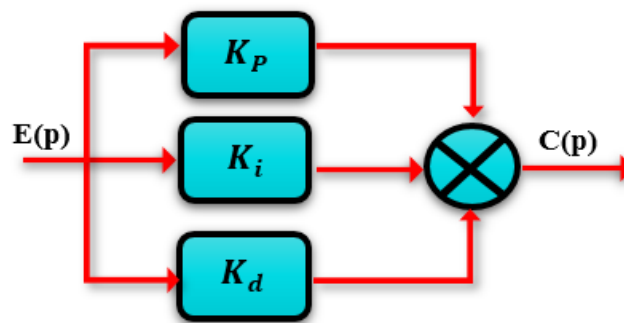


Figure II.6. La structure parallèle.

La loi de commande est de la forme :

$$C(t) = K_p \cdot e(p) + K_i \cdot \int_0^t e(p) \cdot f\tau + K_d \cdot \frac{de(p)}{dt} \quad (2.3)$$

Soit d'après la transformée de Laplace :

$$C(p) = K_p \cdot e(p) + K_i \cdot \frac{e(p)}{p} + K_d \cdot p \cdot e(p) \quad (2.4)$$

C'est la structure la plus utile dans l'industrie.

II .6.2. La structure mixte

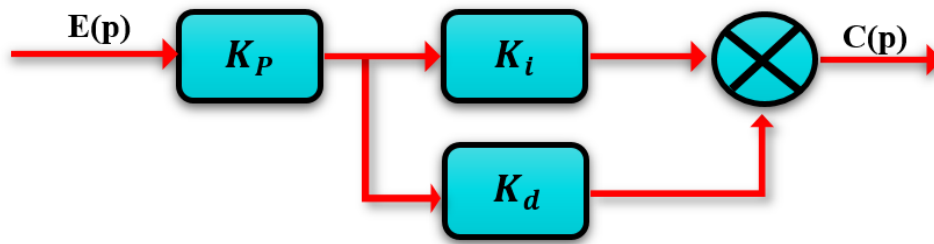


Figure II.7. La structure mixte.

Soit d'après la transformée de Laplace, La loi de commande est de la forme :

$$C(p) = K_p \cdot e(p) \cdot \left[K_i \cdot \frac{1}{p} + K_d \cdot p \right] \quad (2.5)$$

II .6.3. La structure série

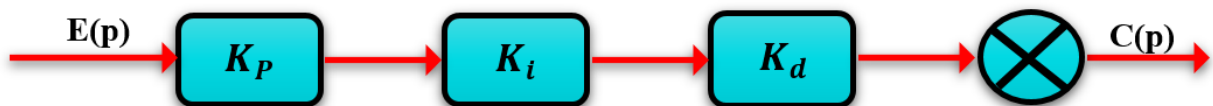


Figure II.8. La structure série.

Soit d'après la transformée de Laplace, La loi de commande est de la forme :

$$C(p) = e(p) \cdot K_p \cdot \left[K_i \cdot \frac{1}{p} \right] \cdot [K_d \cdot p] \quad (2.5)$$

II .6.4. Schéma bloc

Le schéma bloc du system avec des différentes structures de PID, On prend un exemple du système de 3éme ordre de la fonction de transfert suivant

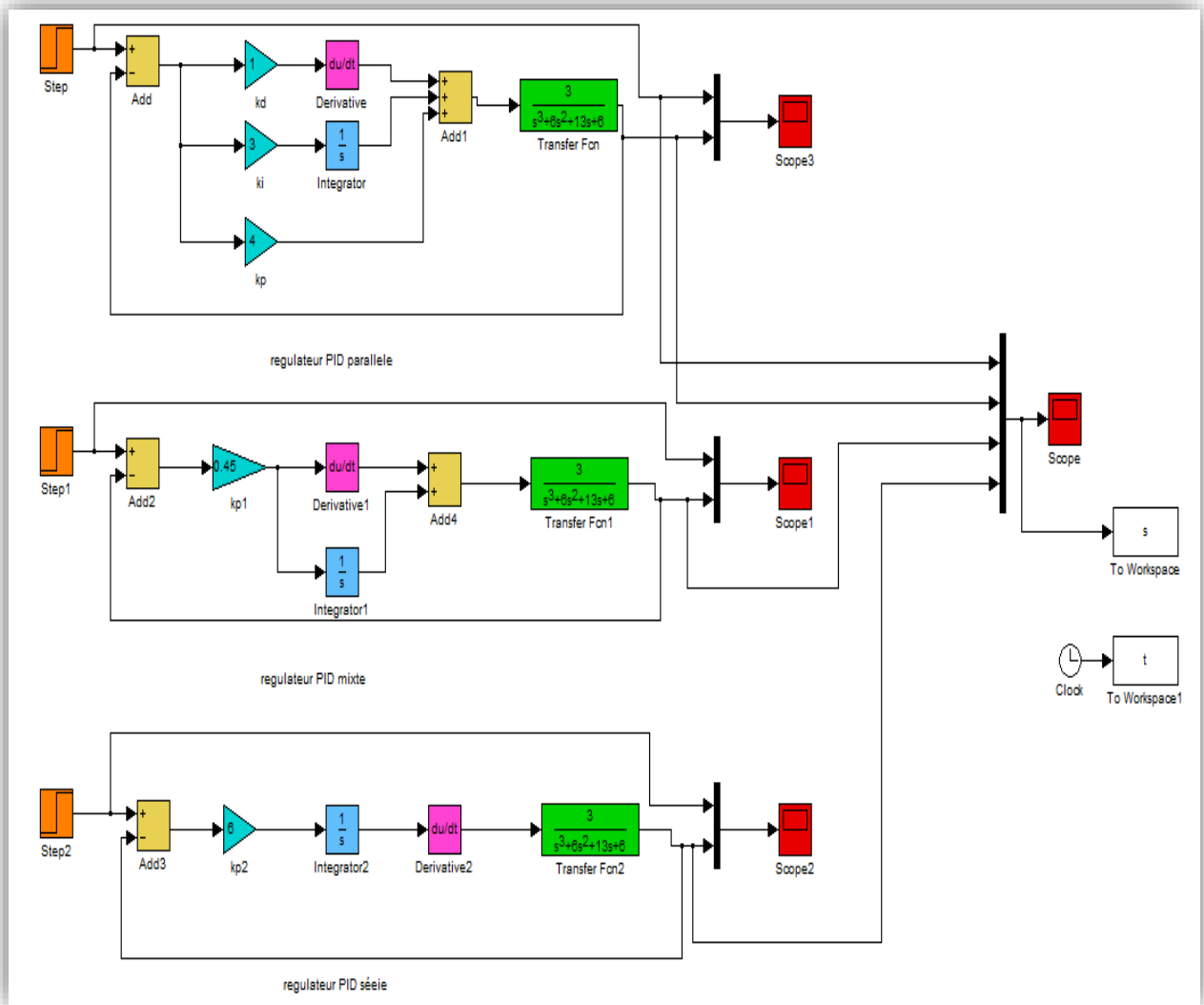


Figure II.9. Sch ema bloc de syst eme de 3eme ordre command e par les diff erentes structures de PID.

✓ A. réponse d'un système

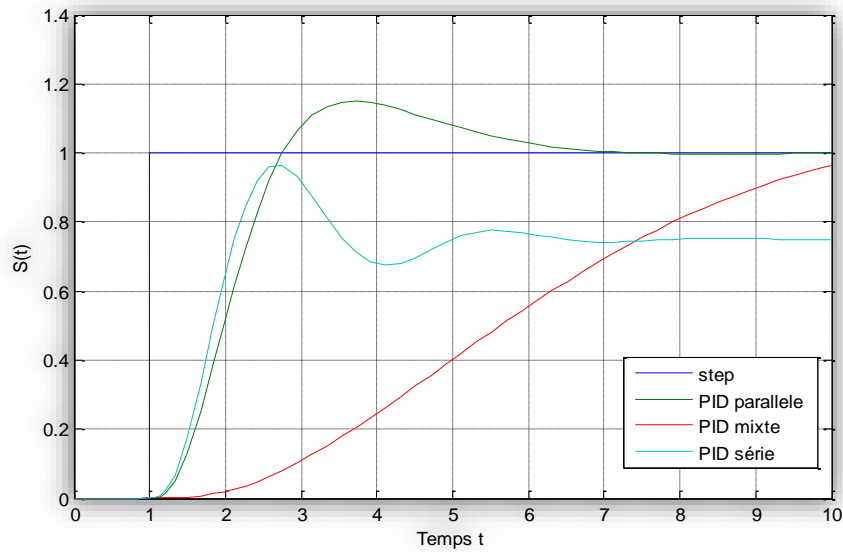


Figure II.10. La réponse du système de 3eme ordre commandé par PID

✓ **Remarque :** on observe que la meilleure réponse est la réponse de la structure parallèle.

On a une autre réponse d'un système avec tous les autres régulateurs précédents :

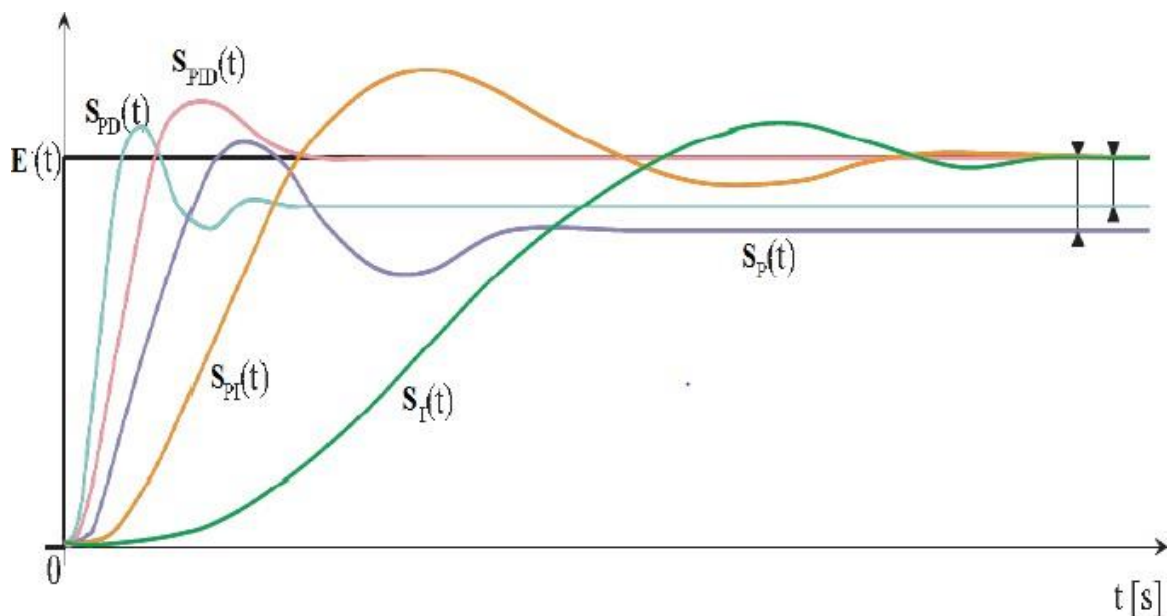


Figure II.11. Réponse d'un système avec tous les autres régulateurs.

II .7. Synthèse sur les actions PID

On résume les avantages et les limitations des actions de base des régulateurs PID [10].

Action	Avantage	Désavantage
P	<ul style="list-style-type: none"> • Action instantanée 	<ul style="list-style-type: none"> • Ne permet pas d'annuler une erreur statique mais permet de la réduire
I	<ul style="list-style-type: none"> • Annule l'erreur statique 	<ul style="list-style-type: none"> • Action lente ralentit le système (effet déstabilisant)
D	<ul style="list-style-type: none"> • Action très dynamique • Améliore la stabilité • Réduire la rapidité 	<ul style="list-style-type: none"> • Sensibilité aux bruits • Forte sollicitation de l'organe de commande

Tableau II.7. Synthèse sur les actions PID.

II .8. La Commande PI

Soit $V^*(t)$ le signal de référence (la consigne) et $V(t)$ le signal de sortie du système à contrôler. Le régulateur PI est connu par une loi de commande.

$$U(t) = K_p e(t) + K_i \int e(t) \quad (2.6)$$

A. Action proportionnelle

- Si K_p est grand, la correction est rapide. Le risque de dépassement et d'oscillation dans la sortie s'accroît.
- Si K_p est petit, la correction est lente, il y a moins de risque d'oscillations.

B. Action intégrale

L'action intégrale réagit lentement à la variation de l'erreur et assure un rattrapage progressif de la consigne.

Tant que l'erreur positive (ou négative) subsiste, l'action $U(t)$ augmente (ou diminue) jusqu'à ce que l'erreur s'annule.

II .8.1. Caractéristiques des régulateurs PI

A. Stabilité

Un système bouclé doit être stable. Seulement si les réactions du système de régulation soient énergétiques sans être disproportionnées avec l'erreur à corriger. Une correction trop forte ou tardive risque de conduire le système à une instabilité.

B. Précision

En régulation, la précision est obtenue par l'implantation d'intégration dans la boucle.

C. Rapidité

En générale, un système bouclé doit répondre rapidement à la variation de sa consigne (poursuite) et effacer rapidement les perturbations (régulation). Le temps de réaction est, bien entendu, en relation étroite avec l'inertie propre du processus.

II .9. Résultat et simulation.

II .9.1. Simulation du convertisseur Boost :

La régulation de la tension continue est assurée par une commande de type **PI**. Ce dernier corrige l'erreur entre la tension continue mesurée et sa référence.

Les valeurs numériques des paramètres du convertisseur sont regroupées dans le tableau suivant :

Les paramètres	Les valeurs
La tension (V)	40V
La tension de référence (Vref)	80V
L	8mH
R	80Ω
C	25μF
Kp	0.001
Ki	0.1
Kd	0

Tableau II.9.1. Les paramètres de système et le régulateur

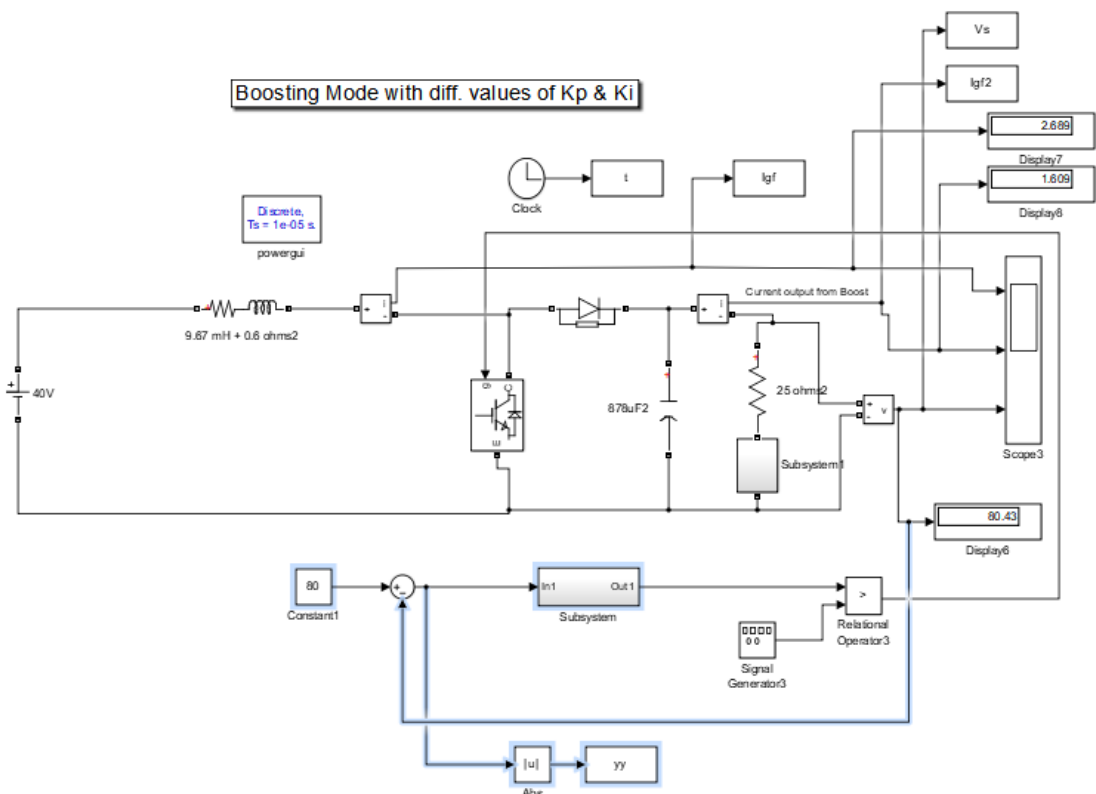


Figure II.12. Simulation d'hacheur boost avec PI control.

✓ **Résultat de simulation :**

Tension de sortie

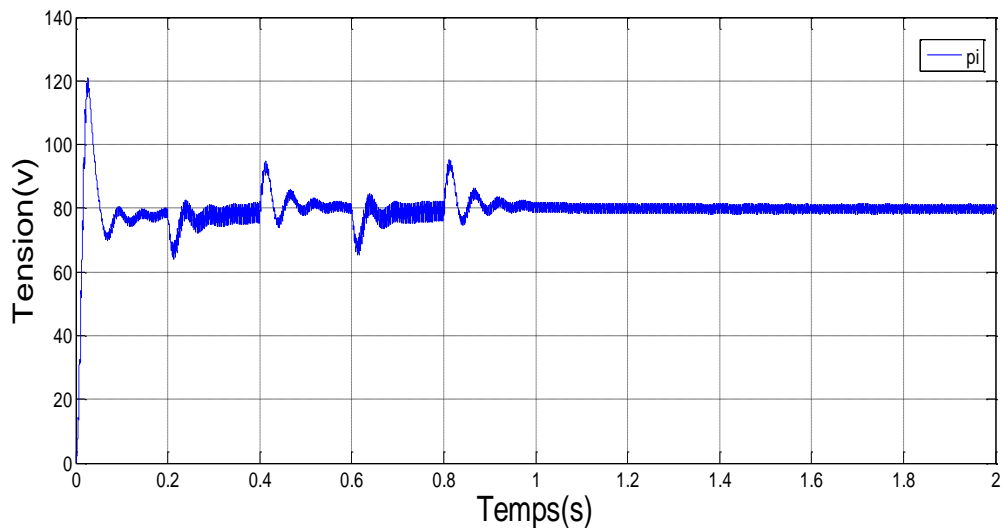


Figure II.13. Tension de sortie du système.

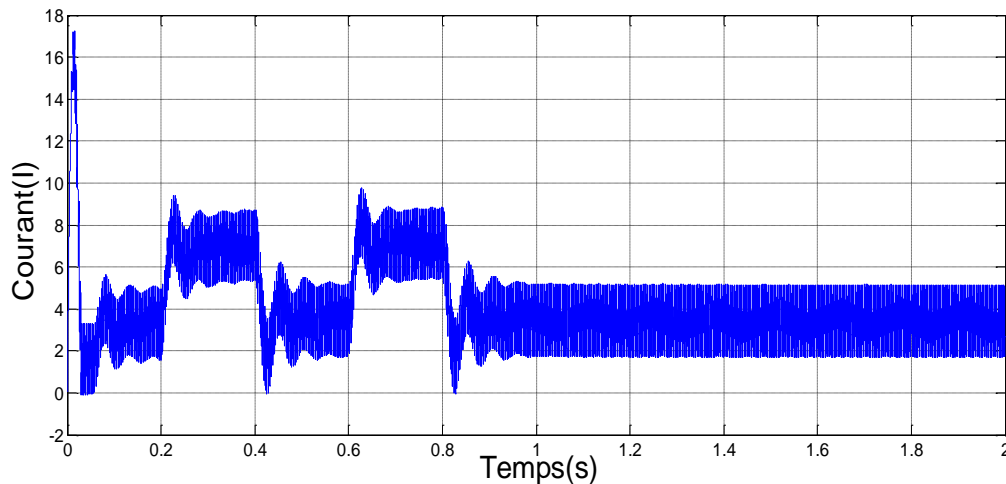


Figure II.14. Courant de l'inductance du système.

II .9.2. Interprétation :

Les Figures (II.13) et (II.14) présentent la réponse de la tension de sortie Y et courant de l'inductance de convertisseur Boost avec une tension de référence de 80V, on remarque que les performances de commande sont presque identiques. Le temps de réponse est de 0.125s, l'erreur statique est presque nul, le dépassement est de 40V soit un pourcentage de 8.2% ce qui est assez important et peu acceptable dans certaines applications sensibles aux variations de la tension. Il y a aussi présence d'oscillations et le temps de stabilisation est d'à peu près 0.13s.

II .10. La commande par réseaux de neurones artificiels (RNA)

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur calcule une sortie unique sur la base des informations qu'il reçoit.

II .10.1. Neurone formel

Le premier modèle du neurone formel a été présenté par Culloch et Pitts (figure IV.8). D'une façon générale, un neurone formel est un élément de traitement qui fait une sommation pondérée de n entrées p_1, \dots, p_R . Si cette somme dépasse un certain seuil (fonction d'activation f), le neurone est activé et transmet une réponse dont la valeur est celle de son activation. Si le neurone n'est pas activé, il ne transmet rien, comme il est indiqué par la figure suivante :

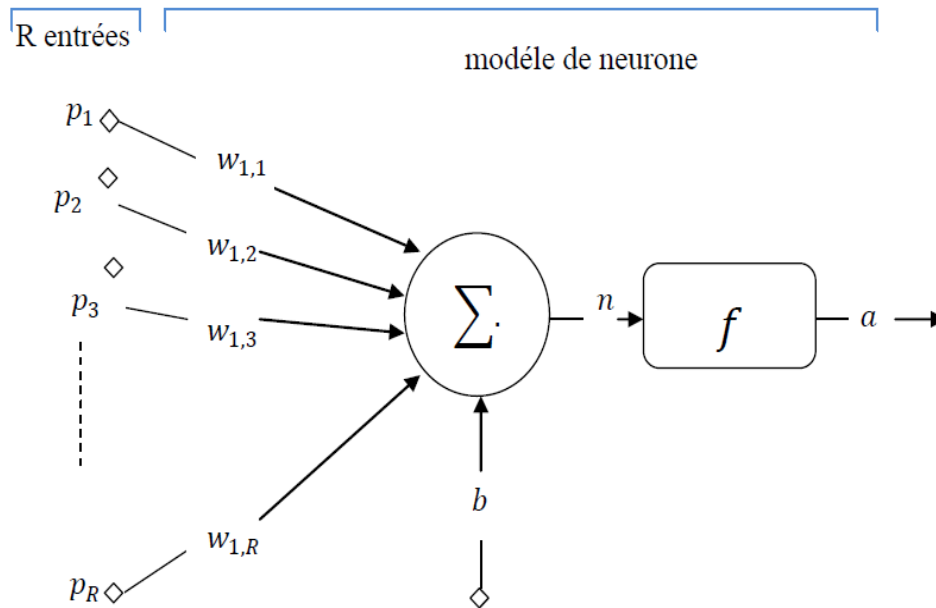


Figure II.15. : représentation du neurone artificiel

- P_1, \dots, R : entrées.
- $W_{1,1}, \dots, R$: poids sur les entrées.
- b : biais; (déplacement horizon. def).
- $n=net$: niveau d'activation.
- f : fonction de transfert.
 - $a = out$: sortie du neurone

Les entrées du neurone sont désignées par P_j ($j = 1, n$). Les paramètres W_j reliant les entrées aux neurones sont appelées poids synaptique ou tout simplement poids. La somme pondérée des signaux d'entrée constitue l'activation du neurone. C'est une fonction qui définit l'activité du neurone, elle est appelée aussi fonction de seuillage ou de transfert. [11].

II .10.2. La fonction d'activation (transfert)

Dans le domaine des réseaux de neurones artificiels, la fonction d'activation est une fonction mathématique appliquée à un signal en sortie d'un neurone artificiel. Le terme de "fonction d'activation" vient de l'équivalent biologique "potentiel d'activation", seuil de stimulation qui, une fois atteint entraîne une réponse du neurone. L'objectif de cette fonction dite aussi fonction d'activation est de rendre la sortie bornée.

II .10.3. Définition des réseaux de neurones artificiels (RNA)

Un réseau de neurones est une structure de traitement parallèle et distribuée d'information, comportant plusieurs éléments, de traitement (neurones), avec topologie spécifique d'interconnexion entre ces éléments, et une loi d'apprentissage pour adapter les poids des connexions. Chaque élément de traitement à une sortie unique branchée à plusieurs connexions collatérales qui transmettent le même signal, qui est la sortie du neurone. Dans un réseau de neurones donné, l'information est traitée par un grand nombre très important d'autres processeurs.

II .10.4. Apprentissage

L'apprentissage est donc défini comme étant tout algorithme d'ajustement des coefficients synaptiques (poids). Donc, les variables modifiées pendant la phase d'apprentissage sont généralement les poids des connexions entre neurones. Le réseau est testé plusieurs fois au fur et à mesure que l'on ajuste les poids, avant qu'il satisfasse à une réponse désirée. Une fois le but est réalisé les poids seront fixé et on peut alors passer à la phase d'utilisation du réseau, principalement il existe deux types :

II .10.5. Apprentissage supervisé

Dans ce type d'apprentissage, on présente au réseau une entrée et la sortie désirée correspondante, les coefficients synaptiques sont alors ajustés dans le but de minimiser un critère de coût. Une fois l'apprentissage est effectué, le réseau est apte à accomplir la tâche prévue. Les performances du réseau sont évalués à l'aide d'un ensemble d'exemples (de même nature que l'ensemble apprentissage ou d'entraînement) dit *ensemble de test*. La méthode la plus utilisée pour ce type d'apprentissage est la rétropropagation.

II .10.6. Apprentissage non supervisé :

Dans ce type d'apprentissage l'adaptation des coefficients synaptiques n'est pas basée sur la comparaison avec une certaine sortie désirée, mais c'est le réseau qui organise lui-même les entrées qui lui sont présentées de façon à optimiser une certaine fonction de coût, sans lui fournir d'autres éléments de réponses désirées. Cette propriété est dite Auto-organisation (self organisation). [11].

II .10.7. La rétropropagation :

L'algorithme de rétro-propagation du gradient qui est le plus connu pour réaliser l'adaptation des réseaux multicouches. Il s'agit d'une méthode d'apprentissage supervisé, fondée sur la modification des poids du réseau dans le sens contraire à celui du gradient de l'erreur par rapport à ces poids.

✓ Schéma block de système :

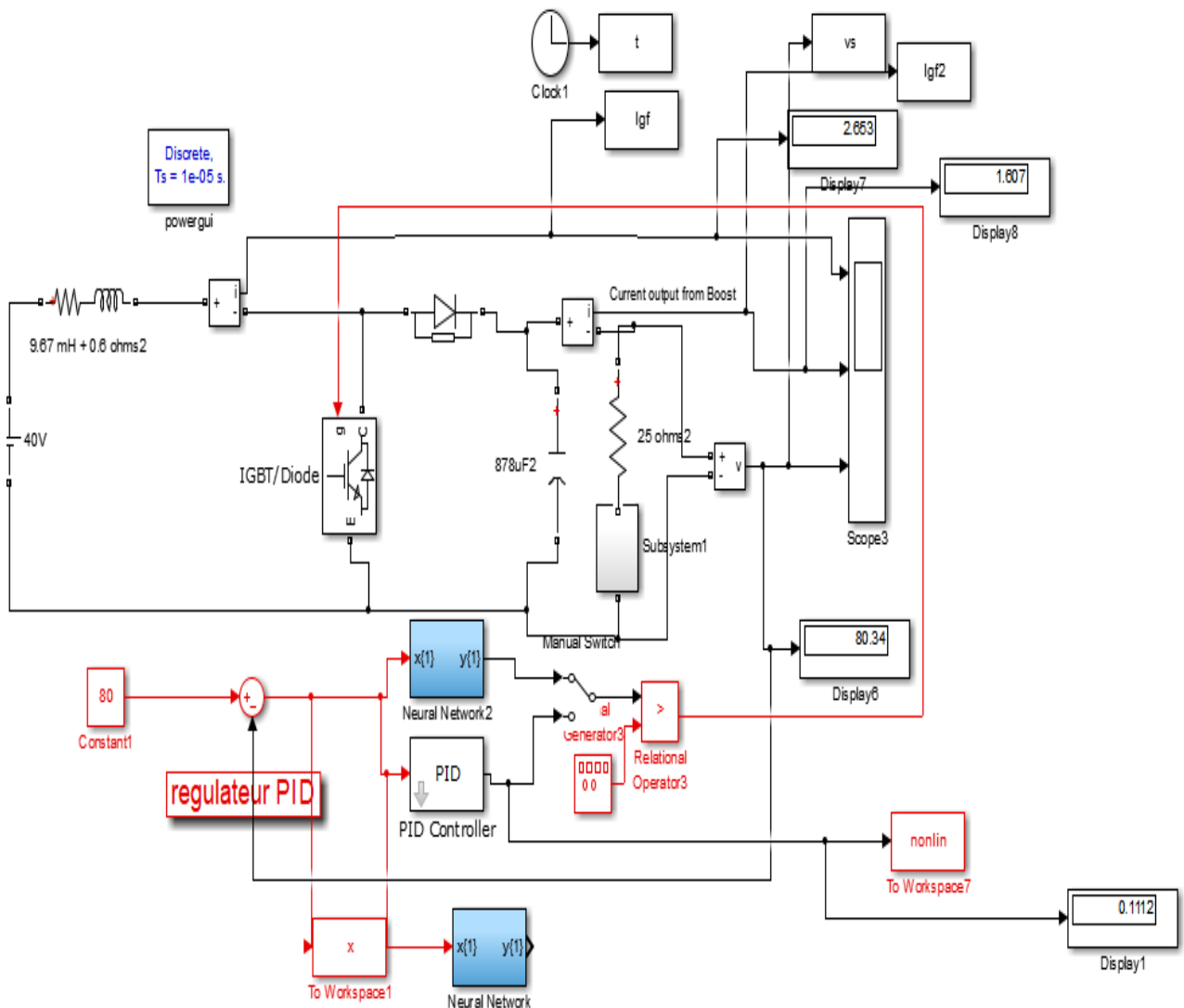


Figure II.16. Simulation d'hacheur boost avec commande RNA .

✓ Résultat de simulation :

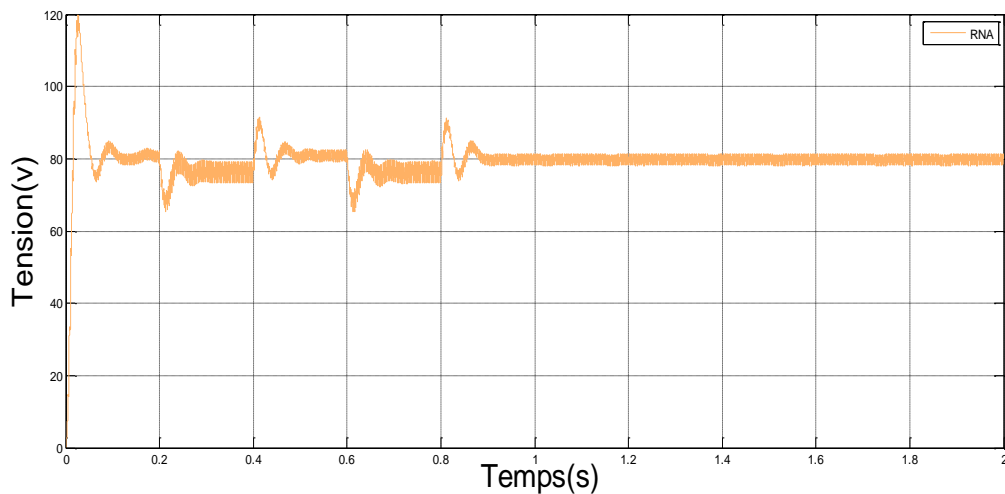


Figure II.17. Tension de sortie (RNA)

✓ Interprétation

On constate d'après le résultat de simulation que la tension de charge est presque identique à la tension de référence. Le temps de réponse est de 0.12s, l'erreur statique est presque nul, le dépassement est de 40V soit un pourcentage de 7.9% ce qui est assez important et peu acceptable dans certaines applications sensibles aux variations de la tension. Il y a aussi présence d'oscillations et le temps de stabilisation est d'à peu près 0.125s.

✓ Etude comparative de régulateur PI et réseaux de neurones

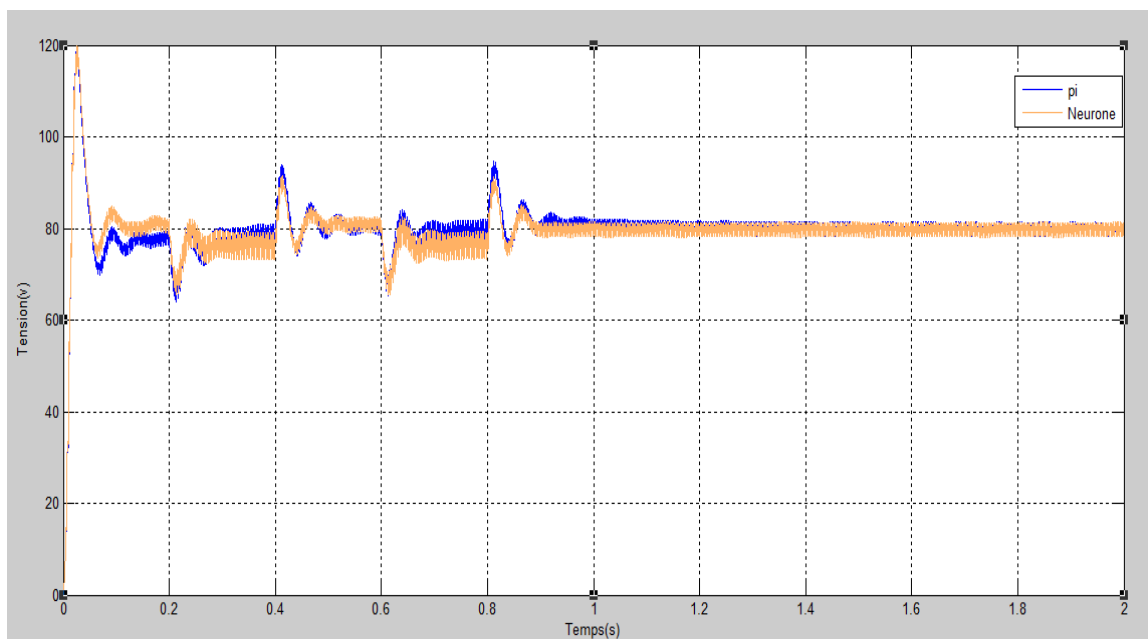


Figure II.18. Les tensions du sortie pour (PI , RNA)

D'après les résultats de simulation et les résultats expérimentaux obtenus, on remarque après on vas utiliser la commande de RNA une amélioration de la tension de sortie et aussi une petite augmentation de pics de tension par rapport à la commande PI. Autrement dit, cette commande est robuste avec une réponse très remarquable, vis-à-vis de la tension de charge.

II .11. La commande par logique floue.

La logique floue a été créée en 1969 par Lotfi Zadeh de l'Université de Berkeley en Californie. Basée sur la théorie des ensembles flous formalisés par Zadeh, la logique floue étend la logique des ensembles classiques pour prendre en compte des ensembles plutôt imprécis. La force de la logique floue vient donc de sa capacité à traiter des données imprécises, mouvantes et des concepts imprécis, contrairement à la logique traditionnelle, la logique crispée, qui nécessite des données exactes et fixes. L'être humain, sans le savoir, utilise dans la majorité de ses actions des raisonnements comparables à ceux de la logique floue. Les premières applications au niveau des systèmes de commande apparaissent dès 1975. Les Japonais commencèrent à utiliser la logique floue dans les produits industriels pour résoudre des problèmes de réglage et de commande à partir de 1985.

II .11.1 Les différentes parties de la commande floue.

Un contrôleur flou est un système à base de connaissances particulier, utilisant un raisonnement en profondeur limité, dans une procédure de chaînage avant règles (activation des règles par les prémisses). On peut distinguer plusieurs étapes dans le traitement des règles. Un schéma représentatif peut être le suivant : Les variables caractéristiques du système à commander et les consignes définissent les variables d'entrée du contrôleur flou. Les variables caractéristiques sont en général, les grandeurs de sortie du processus et, le cas échéant, d'autres mesures déterminantes pour saisir l'évolution dynamique du processus. Les variables de sortie du contrôleur flou sont les commandes à appliquer au processus. La base de connaissances est composée d'une base de données et d'une base de règles. La base de données regroupe:

- Les ensembles flous associés aux variables d'entrée et de sortie du contrôleur flou
- Les facteurs d'échelle (gains) en entrée (normalisation) et en sortie (dénormalisation).

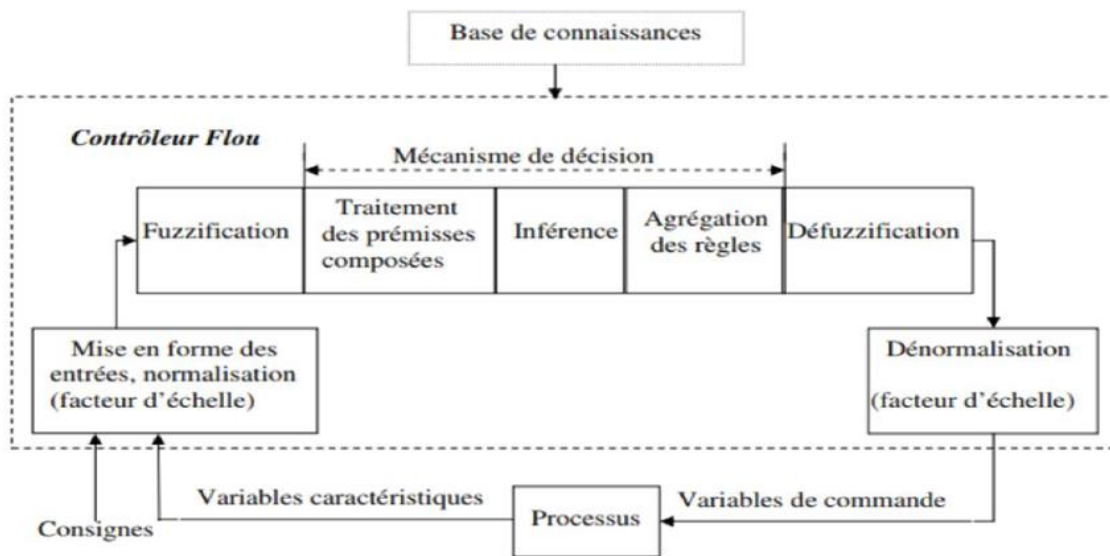


Figure II.19. Structure de base d'un contrôleur flou.

II .11.2 Structure d'un système flou.

Le schéma synoptique d'un système flou est représenté par la figure (II.20)

Ce schéma est composé [17] de :

- Une interface de fuzzification à l'entrée.
- Une base de connaissance.
- Une logique de prise de décision.
- Une interface de défuzzification à la sortie.

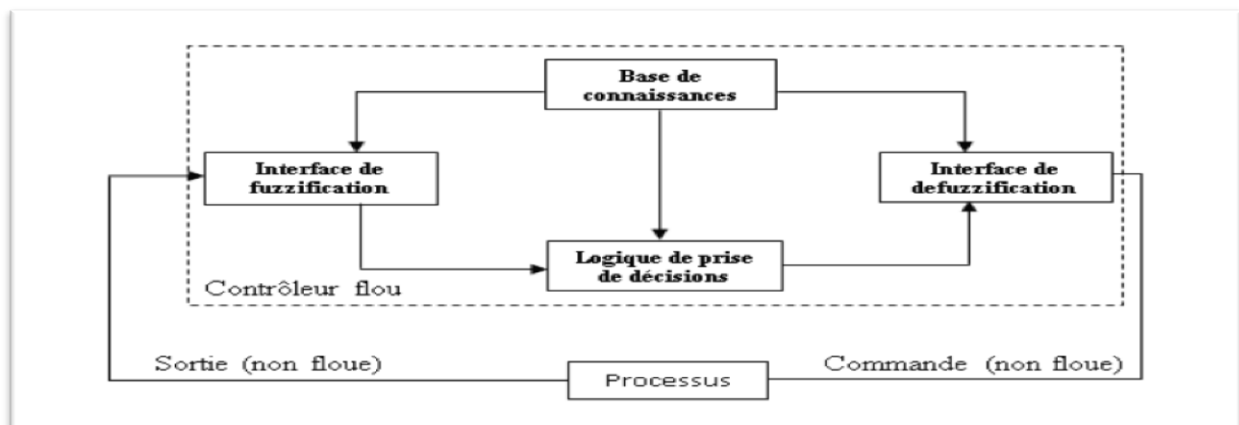


Figure II.20. Schéma synoptique d'un système flou.

- **Fuzzification**

Elle sert à la transformation des variables réelles données (physiques) en variables linguistiques (floues).

- **Moteur (mécanisme) d'inférence**

Un mécanisme d'inférence est la méthode d'interprétation des opérateurs flous et de traitement de l'information linguistique en se basant sur l'expérience de l'opérateur humain. Il existe plusieurs méthodes d'inférence : Max-Min, Max- Produit et Somme-Produit.

- **Défuzzification**

C'est l'opération de transformation des résultats obtenus (conclusions linguistiques) en variables réelles. Il existe plusieurs méthodes de défuzzification parmi lesquelles on peut citer : le centre de gravité, la méthode de hauteurs pondérées, la méthode de la valeur maximum [annexe]et la méthode de la moyenne des centres.

II .11.3. Application de la commande floue.

Nous avons synthétisé le régulateur à logique Floue du type MAMDANI pour le réglage. Le signal de sortie est déterminé en fonction des signaux d'entrée par l'intermédiaire des règles.

Les classes sont notées comme suit :

- NG : négatif grand.
- NM : négatif moyen.
- N : négatif.
- Z : environ zéro.
- P : positive.
- PM : positive moyen.
- PG : positif grand.

Les fonctions d'appartenances sont du type triangulaire et trapézoïdal sur les limites. L'intervalle d'intérêt des variables d'entrées est subdivisé en trois classes pour la tension V , tension de référence V_{ref} et la sortie Y .

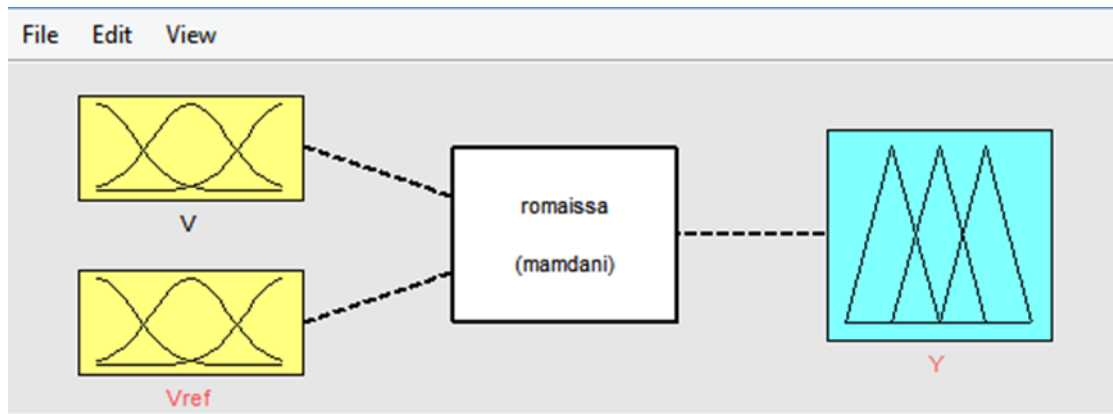


Figure II.21. Définition de la structure globale (entrée, sortie, type d'opérateurs).

Chaque grandeur est caractérisée par sa fonction d'appartenance. On utilise ici un éditeur graphique, illustré par la Figure II.22. Pour définir ces fonctions.

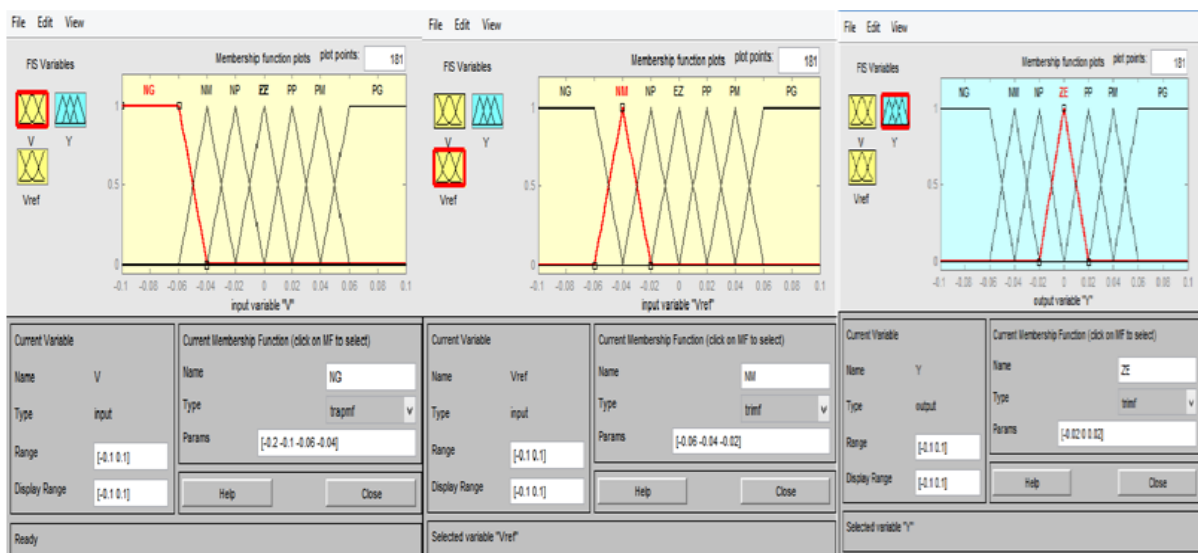


Figure II.22. Définition des fonctions d'appartenances pour(V), (V_{ref}) et(Y). Les règles sont saisies à l'aide d'un formulaire représenté sur la Figure. II.23.

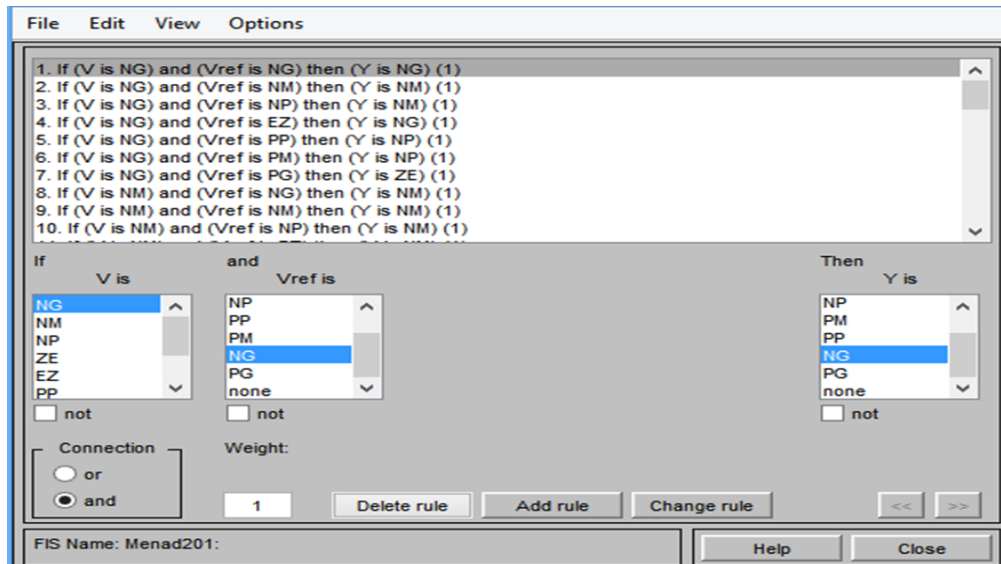


Figure II.23. Écran de saisie des règles.

II.11.4. Simulation d'hacheur Boost avec RLF

Schéma block

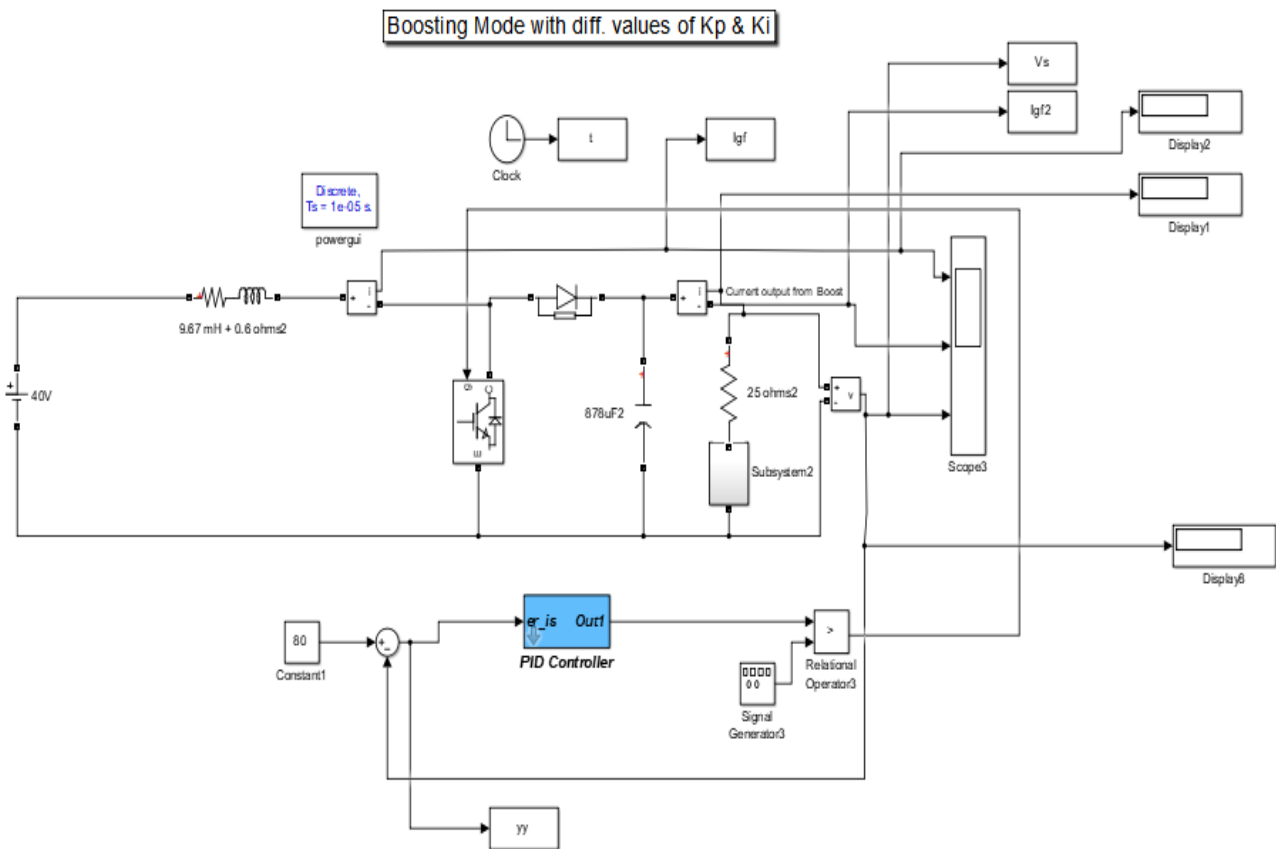


Figure II.24. Simulation d'hacheur boost avec commande RLF.

On utilise ces paramètres suivants :

Les paramètres	les valeurs
La tension (V)	40V
La tension de référence (Vref)	80V
L	8mH
R	80Ω
C	25μF
Kp	2.95
Ki	0.2
Kd	1 /2920

Tableau II.4. Les paramètres de système et le régulateur.

On remarque qu'à chaque fois on intervient sur les gains d'entrées de notre RLF on agit sur la rapidité du système et à chaque fois qu'on intervient sur le gain de la sortie du système on agit sur la précision.

Après plusieurs réglages effectués, notre système devient performant (plus rapide et plus précis) avec les valeurs suivantes : $K_p = 2.95$, $K_i = 0.2$, $K_d = 1 /2920$, associés au régulateur flou on a eu la réponse représentée par la Figure II.25.

✓ Résultat de simulation.

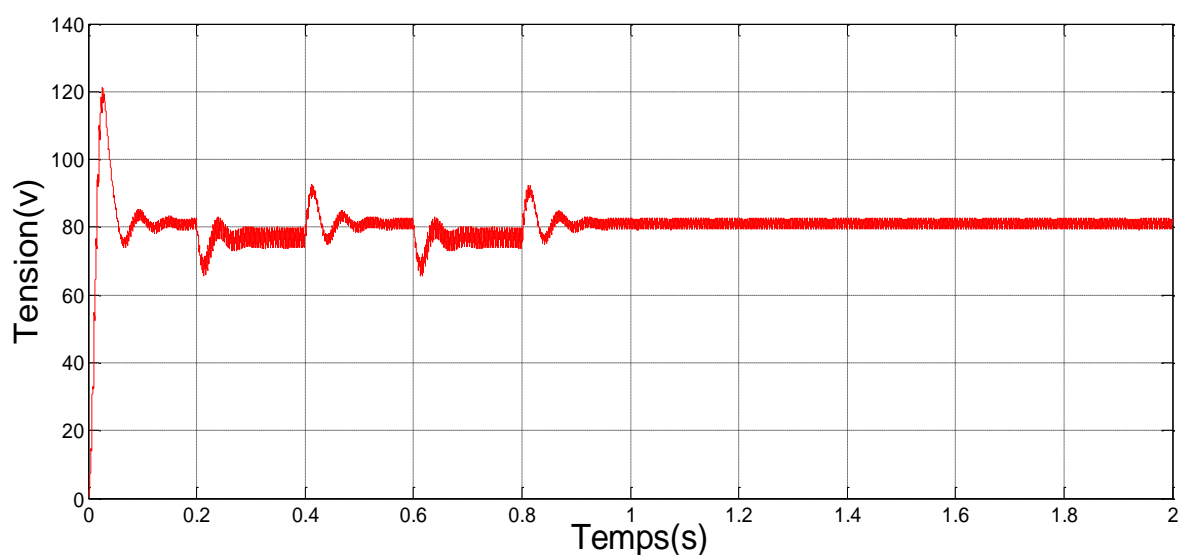


Figure II.25. Tension de sortie du système (RLF)

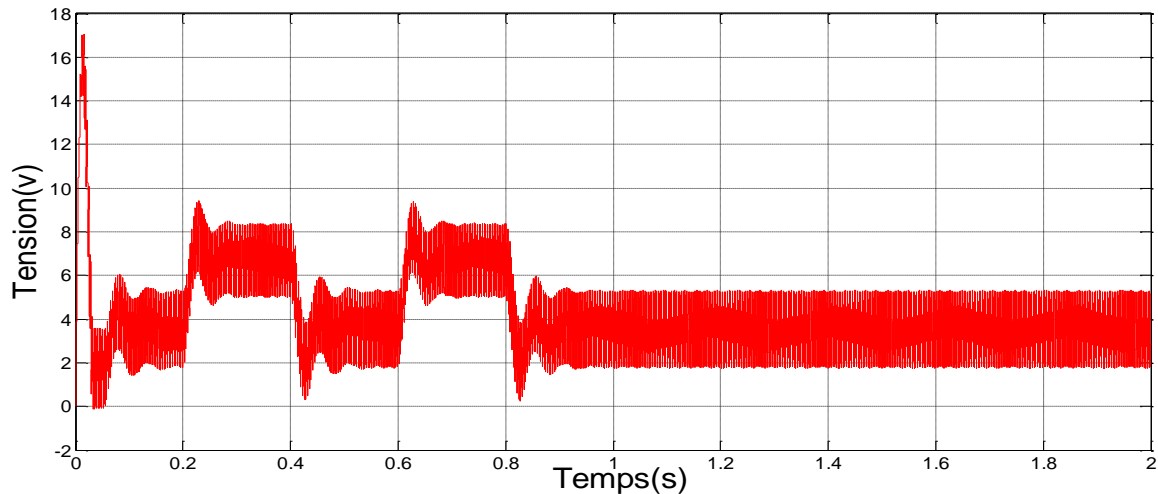


Figure II.26. Courant de l'inductance du système (RLF)

✓ **Interprétation :**

D'après les figures (II.25) et (II.26), précédente on voit que :

La tension de sortie est identique à la tension de référence avec un dépassement, une réponse rapide et le système est stabilisé en temps $t= 0.12s$.

✓ **Etude comparative de régulateur PID et la logique floue.**

On introduit le convertisseur boost dans une boucle de régulation avec les corrections de PID et la logique floue, on obtient la figure (II.26).

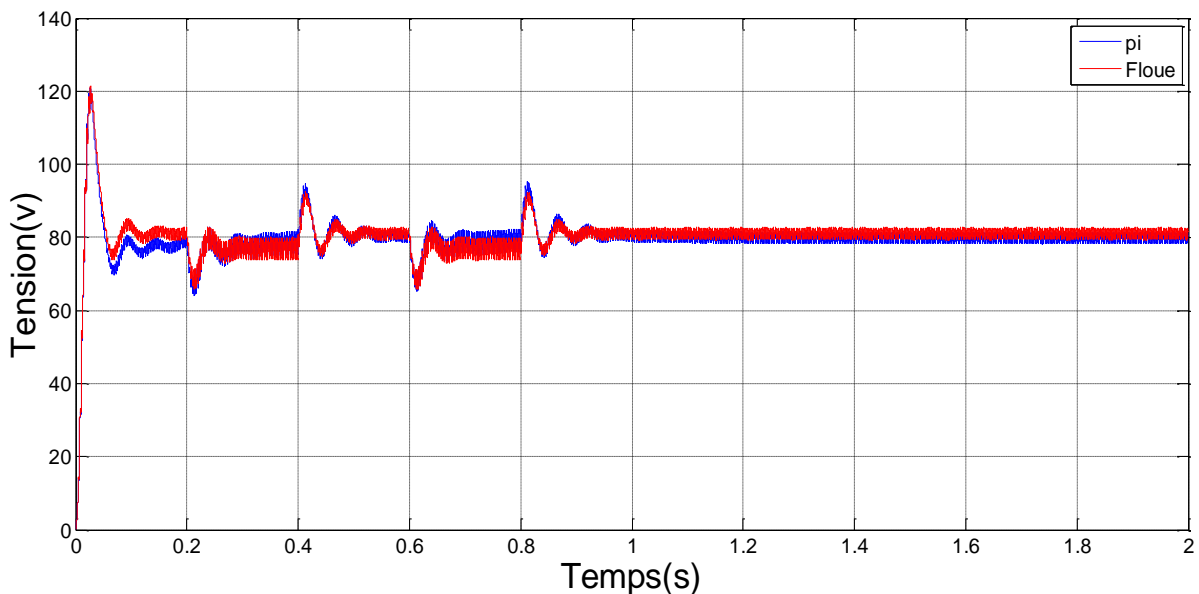


Figure II.26. Les tensions de la sortie pour (PI, RLF)

✓ **Interprétation des résultats.**

En observant les résultats précédents, on voit que le comportement de régulateur flou est identique durant le régime permanent par rapport à régulateur PI, ainsi que le régulateur Flou présente un net avantage :

- Un meilleur temps de montée.
- Une réponse rapide avec un peu de dépassement.
- Ce dernier corrige l'erreur entre la tension continue mesurée et sa référence.

II.12. Conclusion

Nous avons présenté dans ce chapitre les résultats de simulation et les résultats expérimentaux des différentes techniques de commande et leurs applications, Ces différentes techniques sont appliquées à un convertisseur DC/DC (Boost).

Le choix des gains de la commande est minutieux pour obtenir des hautes performances Ces commandes présentes la facilité d'implantation en pratique.

Les résultats de simulation nous donnent une idée des performances de cette commande appliquée sur un hacheur boost.

Dans le chapitre suivant, nous allons nous plus intéressé par les méthodes d'optimisation pour améliorer la performance.

Chapitre III :
Optimisation du PID par
la méthode ACO et GA

III.1 Introduction

L'optimisation est un sujet central en recherche opérationnelle, un grand nombre de problèmes d'aide à la décision pouvant en effet être décrits sous la forme de problèmes d'optimisation. Les problèmes d'identification, la recherche du plus court chemin sont, par exemple, des problèmes d'optimisation.

Dans ce chapitre nous présentons les résultats de simulation de l'optimisation avec deux méthodes différentes :

- La méthode dite 'Ants Colony Optimization' ACO.
- La méthode Algorithmes génétiques GA.

III.2 Optimisation par colonie de fourmis (ACO)

III.2.1. Historique

Dans les sections qui viendront plus tard, nous présenterons la méta-heuristique ACO, pour le "Ant Colony Optimization". Toutes ces idées abstraites sont inspirées des travaux de Deneubourg sur les fourmis. Cette méta-heuristique est relativement récente. Elle a été introduite en 1991 par Coloni, Dorigo et Maniezzo pour résoudre le problème du Voyageur de commerce. Elle s'est popularisée, puis a été l'objet d'améliorations dès 1995 et a été appliquée avec succès à d'autres problèmes d'optimisation dès 1994. Nous allons tout d'abord exposer les différences et les points communs entre les fourmis virtuelles et les fourmis réelles, avant d'exposer en termes plus abstraits la méta-heuristique proprement dite. Ceci expliquera comment les fourmis virtuelles peuvent être exploitées pour résoudre un problème d'optimisation combinatoire. [13].

III.2.2. Similarités et analogies avec les fourmis réelles

Les fourmis virtuelles ont une double nature. D'une part, elles modélisent les comportements abstraits de fourmis réelles, et d'autre part, elles peuvent être enrichies par des capacités que ne possèdent pas les fourmis réelles, afin de les rendre plus efficaces que ces dernières. Nous allons maintenant synthétiser ces ressemblances et différences.

A. Points communs

Colonie d'individus coopérants. Comme pour les fourmis réelles, une colonie virtuelle est un ensemble d'entités non-synchronisés, qui se rassemblent ensemble pour trouver une

"bonne" solution au problème considéré. Chaque groupe d'individus doit pouvoir trouver une solution même si elle est mauvaise.

- **Pistes de phéromones :**

Ces entités communiquent par le mécanisme des pistes de phéromone. Cette forme de communication joue un grand rôle dans le comportement des fourmis : son rôle principal est de changer la manière dont l'environnement est perçu par les fourmis, en fonction de l'historique laissé par ces phéromones.

- **Évaporation des phéromones :**

La méta-heuristique ACO comprend aussi la possibilité d'évaporation des phéromones. Ce mécanisme permet d'oublier lentement ce qui s'est passé avant. C'est ainsi qu'elle peut diriger sa recherche vers de nouvelles directions, sans être trop contrainte.

- **Recherche du plus petit chemin :**

Les fourmis réelles et virtuelles partagent un but commun : recherche du plus court chemin reliant un point de départ (le nid) à des sites de destination (la nourriture).

- **Déplacement local :**

Les vraies fourmis ne sautent pas des cases, tout comme les fourmis virtuelles. Elles se contentent de se déplacer entre sites adjacents du terrain.

- **Choix aléatoire lors des transitions.**

Lorsqu'elles sont sur un site, les fourmis réelles et virtuelles doivent décider sur quel site adjacent se déplacer. Cette prise de décision se fait au hasard et dépend de l'information locale déposée sur le site courant. Elle doit tenir compte des pistes de phéromones, mais aussi du contexte de départ (ce qui revient à prendre en considération les données du problème d'optimisation combinatoire pour une fourmi virtuelle).

B. Analogie

Les fourmis virtuelles possèdent certaines caractéristiques que ne possèdent pas les fourmis réelles : Elles vivent dans un monde non-continu. Leurs déplacements consistent en des transitions d'état

- **Mémoire de la fourmi :**

État interne Les fourmis réelles ont une mémoire très limitée. Tandis que nos fourmis virtuelles mémorisent l'historique de leurs actions. Elles peuvent aussi retenir des données supplémentaires sur leurs performances.

- **Nature de la phéromone déposée**

Les fourmis réelles déposent une information physique sur la piste qu'elles parcourent, là où les fourmis virtuelles modifient des informations dans les variables d'états associées au problème. Ainsi, l'évaporation des phéromones est une simple décrémentation de la valeur des variables d'états à chaque itération.

- **Retard dans le dépôt de phéromone.**

Les fourmis virtuelles peuvent mettre à jour les pistes de phéromones de façon non immédiate : souvent elles attendent d'avoir terminé la construction de leur solution. Ce choix dépend du problème considéré bien évidemment.

- **Capacités supplémentaires.**

Les fourmis virtuelles peuvent être pourvues de capacités artificielles afin d'améliorer les performances du système. Ces possibilités sont liées au problème et peuvent être : 1. l'anticipation : la fourmi étudie les états suivants pour faire son choix et non seulement l'état local. 2. le retour en arrière : une fourmi peut revenir à un état déjà parcouru car la décision qu'elle avait prise à cet état a été mauvaise.

III.2.3. Algorithme de base AS : 'Ant system'

Dans l'algorithme AS, à chaque itération t ($1 \leq t \leq \max t$), chaque fourmi k ($k=1, \dots, m$) parcourt le graphe et construit un trajet complet de $n = |N|$ étapes (on note $|N|$ le cardinal de l'ensemble N). Pour chaque fourmi, le trajet entre une ville i et une ville j dépend de 1. La liste des villes déjà visitées, qui définit les mouvements possibles à chaque pas, quand la fourmi k est sur la Ville i : J_i^k

2. l'inverse de la distance entre les villes : $\eta_{ij} = \frac{1}{d_{ij}}$, appelée visibilité. Cette information "statique" est utilisée pour diriger le choix des fourmis vers des villes proches, et éviter les villes trop lointaines ;

3. la quantité de phéromone déposée sur l'arête reliant les deux villes, appelée l'intensité de la piste. Ce paramètre définit l'attractivité d'une partie du trajet global et change à chaque passage d'une fourmi. C'est, en quelque sorte, une mémoire globale du système, qui évolue par apprentissage. La règle de déplacement (appelée "règle aléatoire de transition proportionnelle" par les auteurs [Bonabeau et al., 1999]) est la suivante :

$$P_{ij}^k = \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{i \in J_j^k} (\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta} \quad (1,1)$$

Où α et β sont deux paramètres contrôlant l'importance relative de l'intensité de la piste $\tau_{ij}(t)$, et de la visibilité η_{ij} .

Après un tour complet, chaque fourmi laisse une certaine quantité de phéromone $\Delta\tau_{ij}(t)$ sur l'ensemble de son parcours, quantité qui dépend de la qualité de la solution trouvée :

$$\Delta\tau_{jk}^i = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i, j) \in T^k(t) \\ 0 & \text{si } (i, j) \notin T^k(t) \end{cases} \quad (1,2)$$

Où $(i, j) \in T^k(t)$ est le trajet effectué par la fourmi k à l'itération t , $L^k(t)$ la longueur de la tournée et Q un paramètre fixé. L'algorithme ne serait pas complet sans le processus d'évaporation des pistes de phéromone.

$$T_{ij}(t+1) = (1 - \rho) \cdot T_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (1,3)$$

Où m est le nombre de fourmis et ρ le taux d'évaporation.

La quantité initiale de phéromone sur les arêtes est une distribution uniforme d'une petite quantité $\tau_0 \geq 0$. [14].

• **Algorithme d'optimisation ACO : « Ant Colony Optimization »**

Pour $t = 1, \dots, \max t$

Pour chaque fourmi $k = 1, \dots, m$

Choisir une ville au hasard

Pour chaque ville non visitée i

Choisir une ville j , dans la liste k i J i des villes restantes, selon la formule 1.1

Fin Pour

Déposer une piste Δk i τj (t) sur le trajet () k T t conformément à l'équation 1.2

Fin Pour

Evaporer les pistes selon la formule 1.3

Fin Pour

III .3. Optimisation des paramètres par ACO :

Pour évaluer les performances des contrôleurs proposés (PI, RLF), on présentera les résultats des techniques d'optimisation ACO de la tension de sortie des convertisseurs boost, et les paramètres de simulation utilisés sont les suivants :

Choix des paramètres	Nombre of Ant	pheromene	Evaporation Parametre
Valeurs	35	0.06	0.95
Positive phéromone	Négative phéromone	Max tour	Min value
0.2	0.3	100	-10

Tableau III .1. Paramètres de ACO

III.3. Algorithmes génétiques GA :

III.3.1. Principes et généralités :

Les algorithmes génétiques sont généralement utilisés pour rechercher les optima d'un critère d'optimisation défini sur un espace de recherche. Leur mise en œuvre nécessite :

Un codage des données qui associe à chaque point de l'espace de recherche une structure de données spécifique, appelée génotype ou ensemble de chromosomes, qui caractérisera chaque individu de la population.

La définition d'une fonction d'adaptation à maximiser, définie sur l'espace de recherche. Elle est également appelée fonction d'évaluation ou encore fitness. Elle se définit en fonction du critère d'optimisation du problème : les extrema recherchés du critère correspondent aux maximas de cette fonction.

Une population initiale, regroupant un ensemble d'individus, dont dériveront les futures générations.

Des opérateurs d'évolution de la population, permettant l'exploration, plus ou moins large, de l'espace de recherche.

Un processus de sélection des individus les mieux adaptés, qui sera appliqué sur chaque nouvelle génération d'individus [15].

III .3.2. Fonctionnement d'algorithme génétique :

Le fonctionnement d'un GA est basé sur les phases suivantes :

- 1. Initialisation : Une population initiale de N chromosomes est tirée aléatoirement.
- 2. Évaluation : Chaque chromosome de la population initiale est décodé, puis évalué.
- 3. Sélection : Création d'une nouvelle population de chromosomes par l'utilisation d'une Méthode de sélection appropriée (regroupement des chromosomes par paires).
- 4. Reproduction : Possibilité de croisement et de mutation au sein de la nouvelle population.
- 5. Retour à la phase d'évaluation (étape 2) tant que la condition (ou critère d'arrêt) du problème n'est pas satisfaite [16.17].

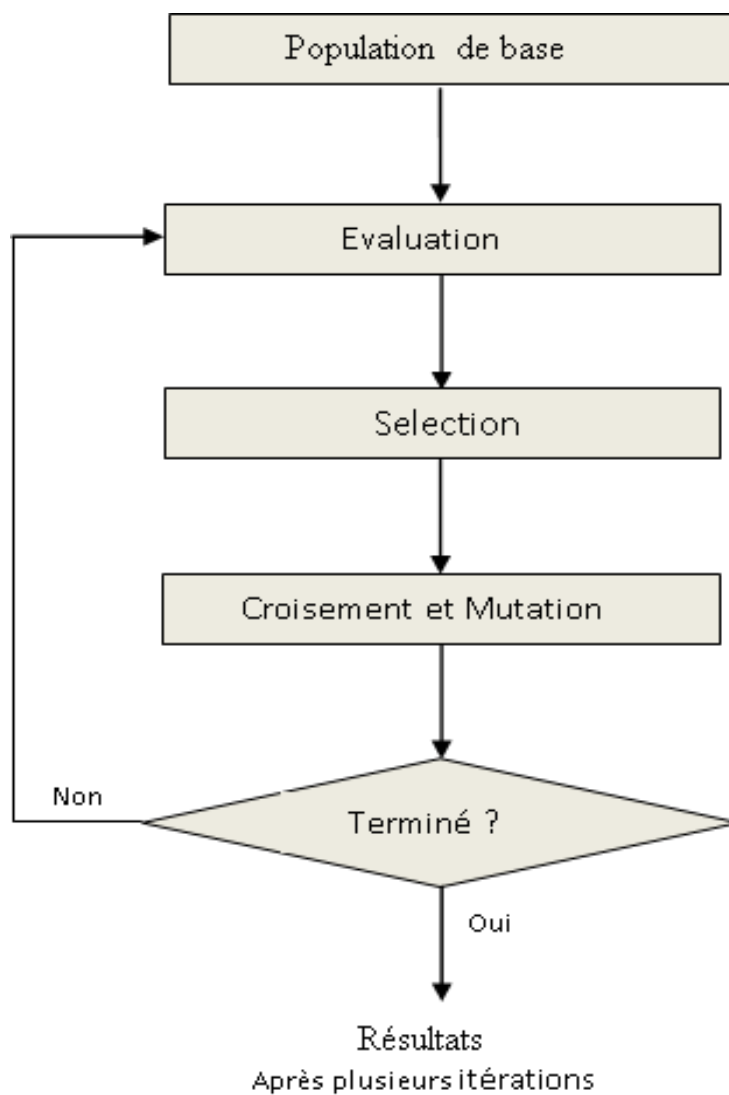


Figure III.1 - Fonctionnement courant d'un algorithme génétique [18]

III .3.3. Résultats des simulations :

Tension de sortie

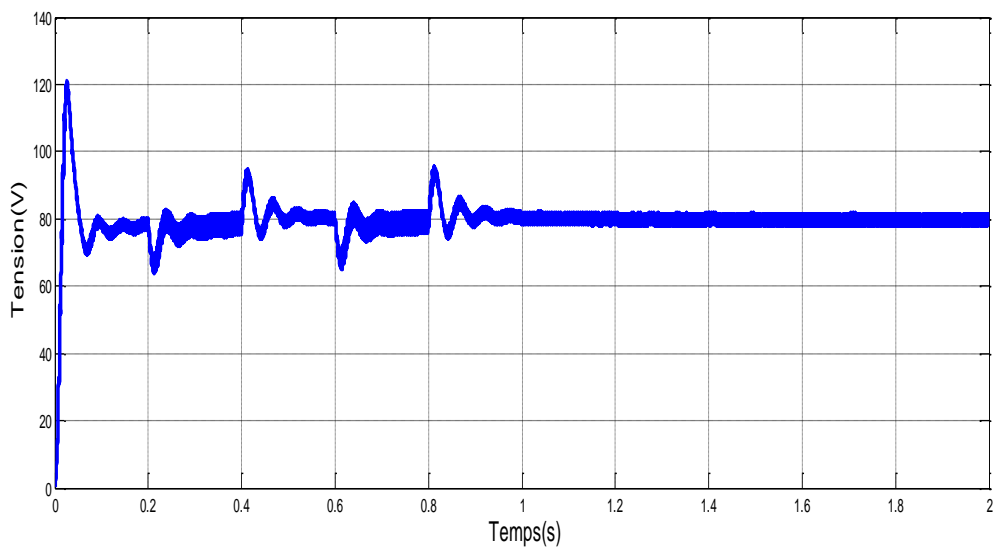


Figure III.2. Tension de sortie de system avec PI

PI-ACO

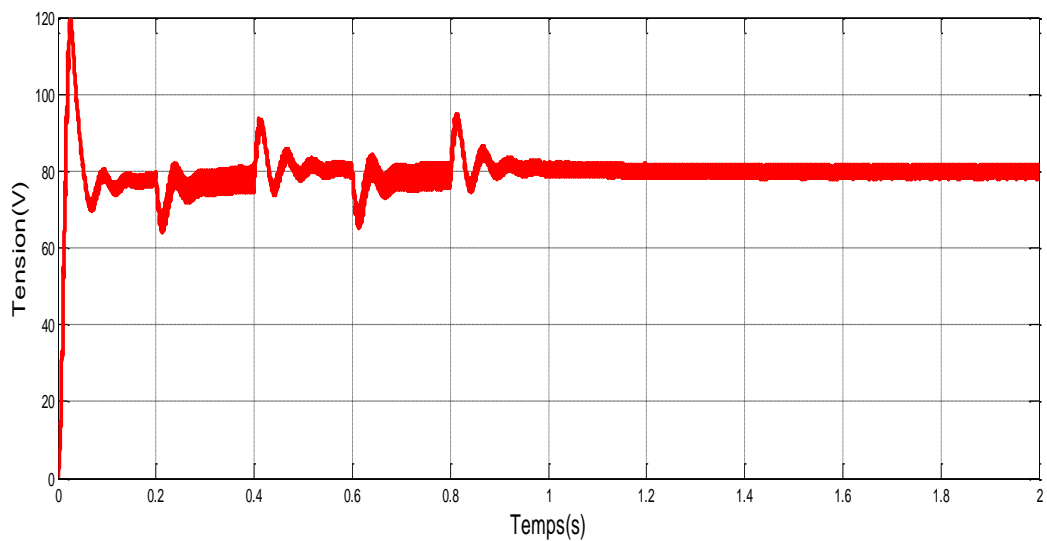


Figure III.3. Tension de sortie de system avec PI a optimise par ACO

PI-GA

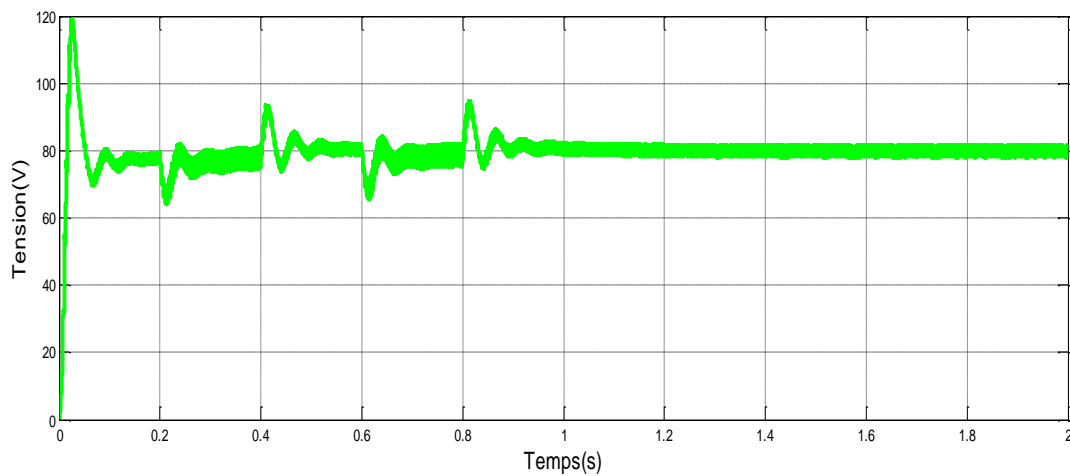


Figure III.4. Tension de sortie de system avec PI a optimise par GA

III.4. la Comparaisent des résultats de simulation (PI, PI ACO, PI GA) :

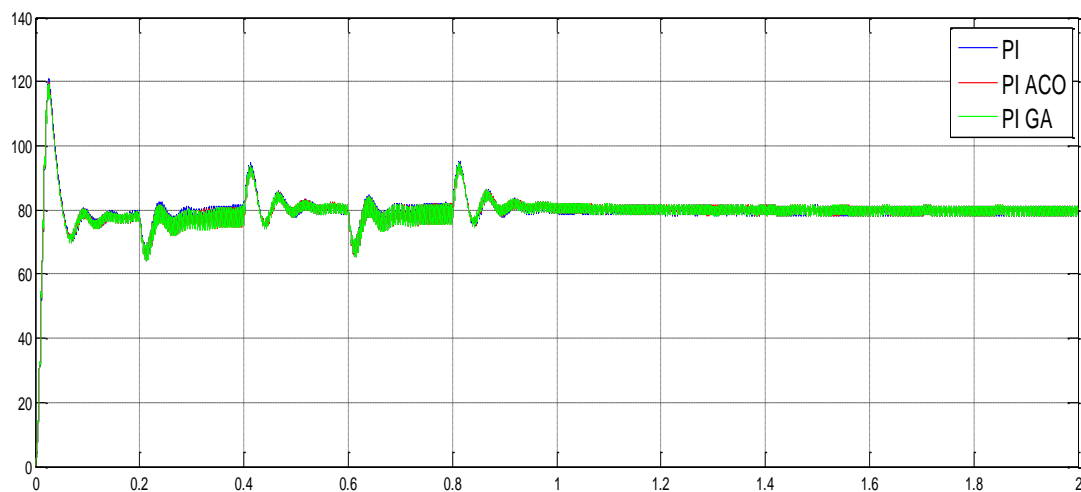


Figure III.5. Les tensions de sortie pour (PI, PI ACO, PI GA)

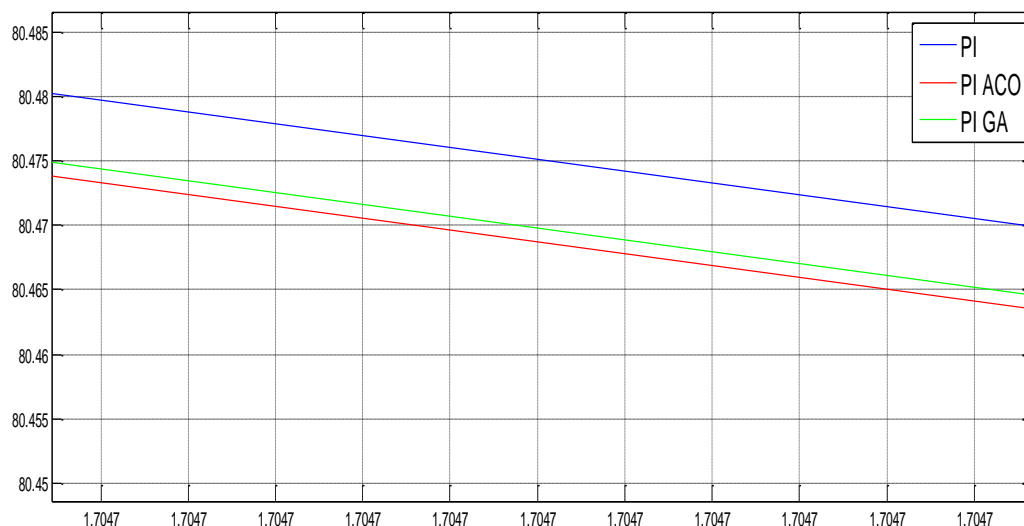


Figure III.6. Les tensions de sortie pour (PI, PI ACO, PI GA)

Floue-GA

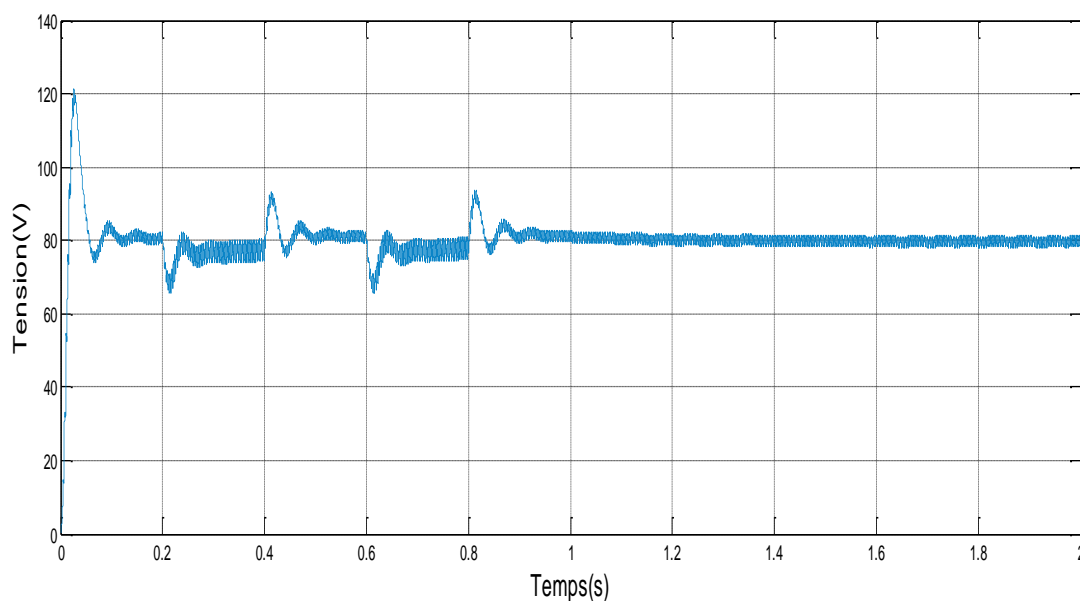


Figure III.7. Tension de sortie de system avec RLF optimise par GA

Floue-ACO

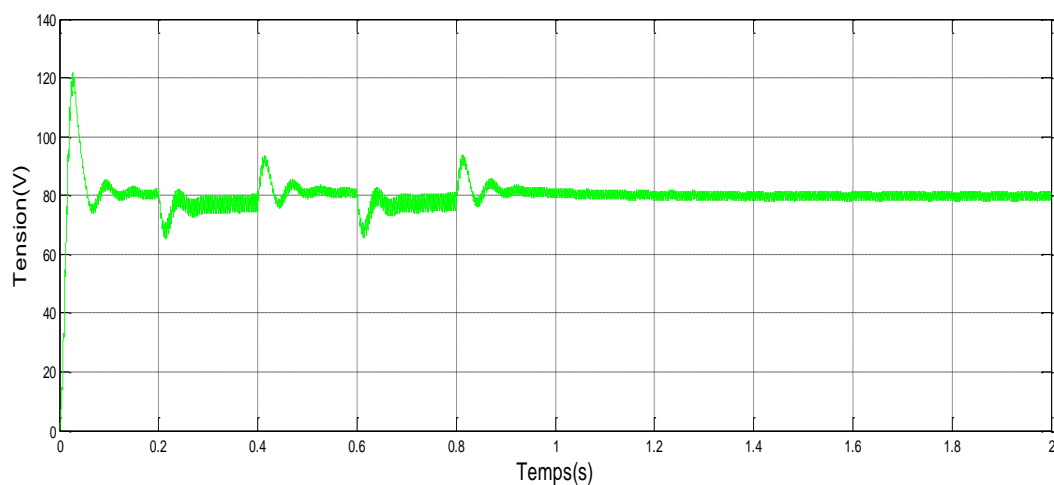


Figure III.8. Tension de sortie de system avec RLF optimise par ACO

III.5. La Comparaisent des résultats de simulation (RLF, RLF ACO, RLF GA) :

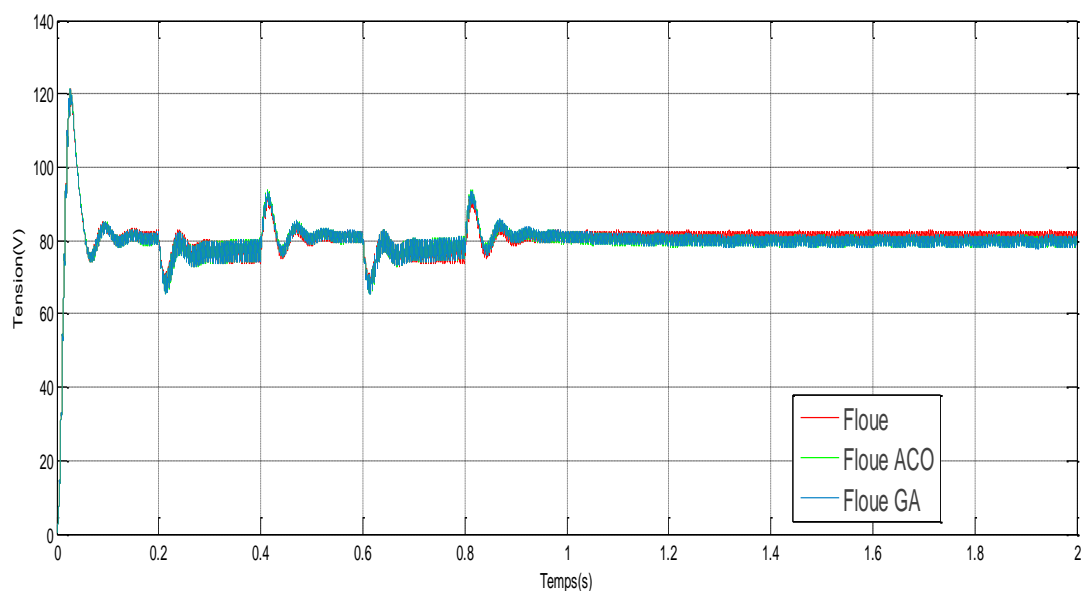


Figure III.9. Les tensions de sortie pour (RLF, RLF ACO, RLF GA)

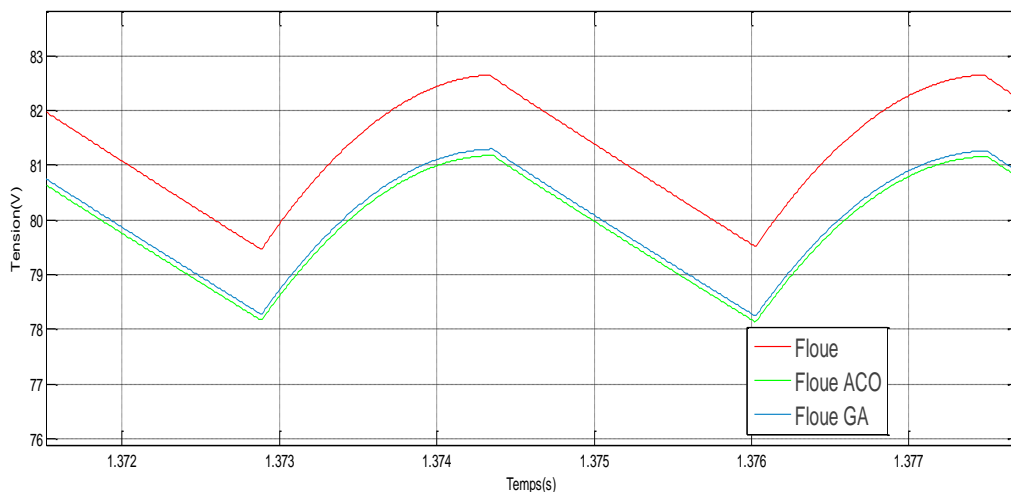


Figure III.10. Les tensions de sortie pour (RLF, RLF ACO, RLF GA)

III.6. Interprétation :

D'après les figures (III.2 III.3 III.4 III.7 III.8 III.9), précédentes on voit que : les résultats de simulations après l'optimisation par ACO et GA , montrent l'efficacité de cette technique , et présente une amélioration des performances de réponse au régime transitoires traduit par la diminution des oscillations au niveau de signal de sortie et au niveau du courant de l'inductance , ainsi une vitesse de convergence meilleure .

III .7. Conclusion :

On conclue que l'optimisation avec ACO et GA donne une bonne amélioration au niveau du temps de réponse, le dépassement, les oscillations et l'erreur statique et des meilleurs performances par rapport les autres régulateurs (PI , RLF).

Conclusion générale

Conclusion générale

Ce mémoire a eu pour objectif d'optimiser des contrôleurs **PI**, **RLF** et appliques a un convertisseur statique, le travail a été présenté :

Une brève représentation sur les convertisseurs **DC/DC** de types (**boost**, **Buck-boost**), leur principe de fonctionnement et leurs modelés mathématiques a été présentée.

L'application des trois commandes par régulateur **PI**, par logique floue et réseaux de neurone appliques au convertisseur **DC/DC**, puis une comparaison entre ces trois commandes pour illustrer la robustesse de chaque technique. On a jugé que le régulateur **RLF** possède de bonnes performances vis-à-vis les perturbations externes en présentant les avantages tels qu'une réponse rapide avec un léger dépassement lors les perturbations subit à la commande de ce type de convertisseur par rapport à la commande par **PI**. Mais la commande par réseaux de neurone montre une excellente performance une réponse rapide sans dépassement et un rejet de perturbation.

La méthode d'optimisation par **ant colony optimisation ACO** et **les algorithmes génétiques** a été applique aux deux commandes mentionnées au paravent pour améliorer ses performances et le choix des gains des deux régulateurs utilises pour commander **boost**. Les résultats de simulations en utilisant les méthodes **ACO** et **GA**, montrent l'efficacité de ces techniques qui présentent une oscillation des performances de réponse au régime transitoires traduit par la diminution des oscillations au niveau de signal de sortie et au niveau du courant de l'inductance, ainsi une vitesse de convergence meilleure.

On suggère dans le futur travail de recherche d'utiliser d'autre méthodes d'optimisation telles que MFO, CRAW, BAT, et de réaliser commandes en temps réels (benchmark).

**Reference
bibliographie**

Reference bibliographie

- [1] Zemouri Azeddine, « Etude Expérimental d'un Convertisseur DC/DC contrôler par une commande robuste », Mémoire de Magister, Université de Batna Faculté de Technologie Département Electrotechnique, 2014.
- [2] M. Rabia Benyacine ‘‘ contribution a la commande robuste des systèmes non linière. Projet d’autoamtique 2008.
- [3] Sahli Abdeslem, « Filtrage actif et contrôle de puissances : application aux systèmes photovoltaïques interconnectés au réseau », Mémoire de Magister, Université Ferhat abbas de Sétif Faculté de Technologie Département Electronique, 2012.
- [4] Amin Ayad, « Etude et réalisation d’une commande MLI pilotant un hacheur série », Mémoire de Master, Université Abou BekrBelkaid Tlemcen Faculté de Technologie Département Electrotechnique, 2014.
- [5] [https : www.astuces-pratiques.fr/electronique/le-hacheur-serie-principe-de-fonctionnement](https://www.astuces-pratiques.fr/electronique/le-hacheur-serie-principe-de-fonctionnement) (dernier accès : 02/04/2018) .
- [6] Tiar Mourad, « Commande Optimale d’un Système de Génération Photovoltaïque Hybride », Mémoire de Magister, Université Mohamed Khider Biskra Faculté des Sciences et de la technologie Département Electrotechnique, 2011.
- [7] Sahraoui Hamza, « Modélisation Et Commande Des Convertisseurs DC-DC Utilisés Dans Les Systèmes Photovoltaïques », Mémoire DOCTORAT 3ème cycle LMD, Université de Batna 2 Faculté des Sciences et de la technologie Département Electrotechnique, 2016.
- [8] C. le Lann, le PID utilise en regulation de position et/ou de vitesse de moteurs électriques. Project de fin d’étude 2007.
- [9]. M. Biswal, ‘‘control techniques for DC-DC buck converter with improved performance’’, Master Thesis,Nation institute of Technology, Rourkela,March2011
- [10]. Jean-Marie Flaus, la régulation industrielle ; régulateur PID, prédictif et flous. Edition HERMES ,Paris 1994.
- [11]. T, FAROUK Optimisation d’entraînement des réseaux de neurones RBF par la métaheuristique ACO .
- [12] L. A. Zadeh, "Fuzzy Sets Information and Control." pp. 29-44, 1965 .

Reference bibliographie

- [13] Johann, Dréo. (2006) Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues. Application en génie biomédical. Thèse de doctorat.
- [14] Marco Dorigo, Luca Maria Gambardella. Ant colonies for the travelling salesman problem. 11 october 1996.
- [15] A. Ben Mohamed. Résolution approchée du problème de bin-packing. Thèse de doctorat. Université du Havre (2009).
- [16] M. Assas, M. Djenane. Optimisation des Conditions de Coupe Basée sur Critère Combiné par la Méthode des Algorithmes Génétiques. 16 ème Congrès Français de Mécanique, Nice, 1-5 septembre (2003).
- [17] M. Assas, M. Djenane. Optimisation des conditions d'usinage basée sur un critère combiné. Journées de Mécanique EMP. Alger. 23/24 Décembre (2001).
- [18] L. Drdi. Les algorithmes génétiques. Thèse de doctorat (2005).

Annexe

Annexe.

A. optimisation des paramètres RLF

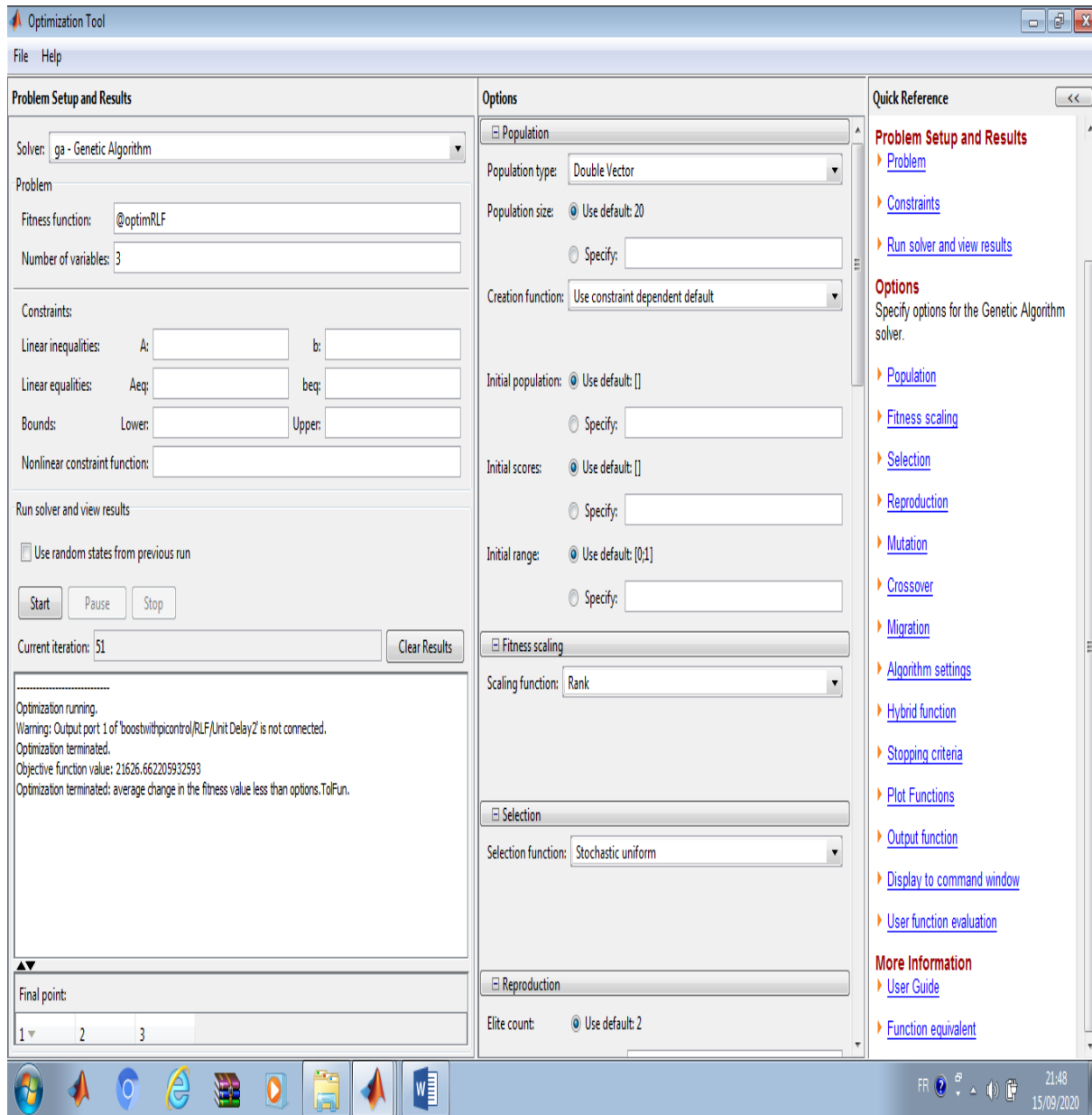


Figure.A.1. Écran de saisie des optimisation RLF-AGS

Un algorithme génétique a été utilisé dans cette étude dont le but d'ajuster des paramètres de régulateur logique floue (k_1 , k_2 , k_3).

On a obtenu des gains ($k_1=1.6910$; $k_2=0.04446$; $k_3=0.19729$;

Optimization running.

Warning: Output port 1 of 'boostwithpicontrol/RLF/Unit Delay2' is not connected.

Optimization terminated.

Objective function value: 21626.662205932593

Optimization terminated average change in the fitness value less than options.TolFun.

B. Reseaux de neurones

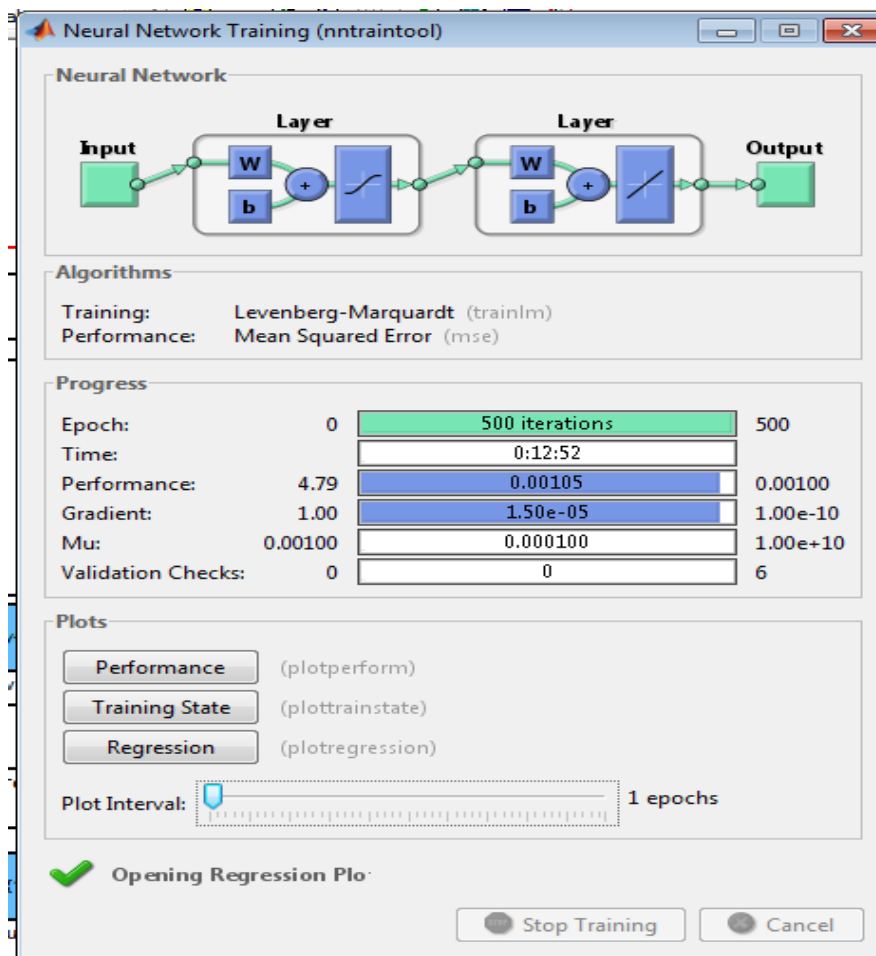


Figure.B.1. Neural Network Training

Après le choix des paramètres (les couches cachées) on a lancer l'apprentissage par le bouton **Train network** Le modèle d'apprentissage est enregistré, cette fenêtre représente les résultats de **neural network training** (nombre d'itérations, l'erreur...)

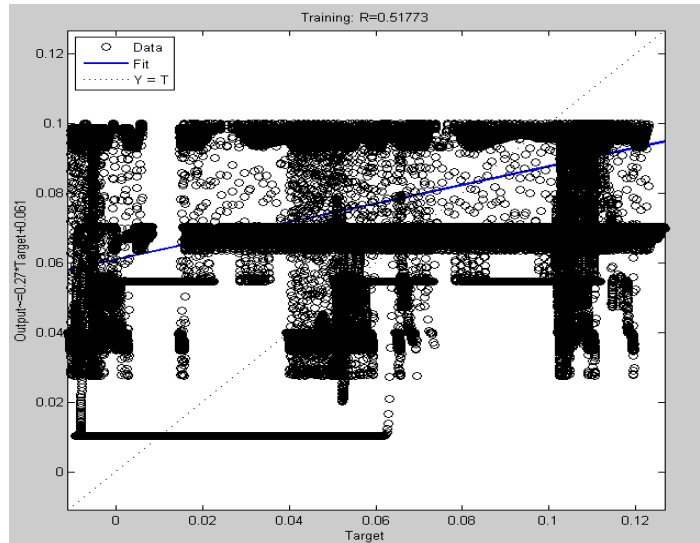


Figure.B.2. Intervalle de l'erreur

Cette figure représente l'intervalle de l'erreur

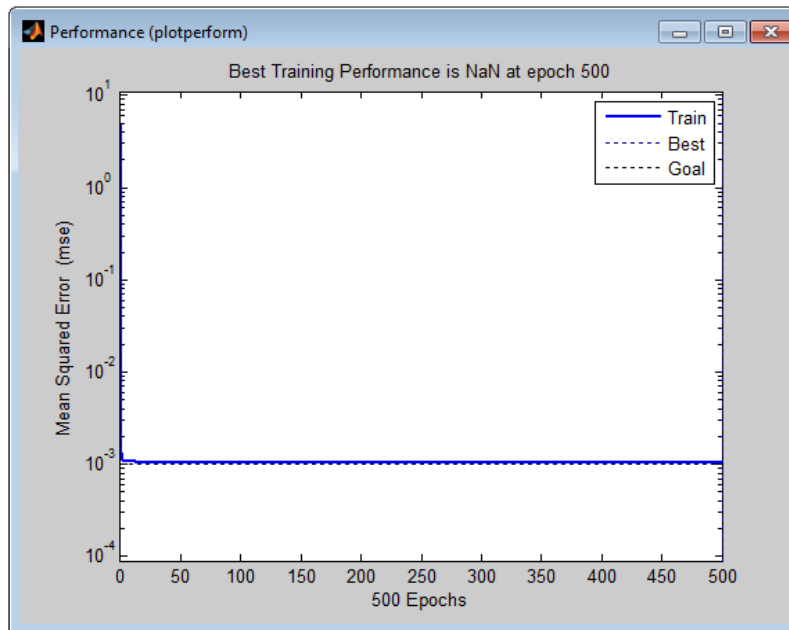


Figure.B.3. Performance

Cette figure représente **Best Training Performance**, on a le **Train DATA** a arrivez à la moyen **Best** et **Goal** ($10 \text{ e-}3$) et ça a l'aide des grands nombres de epoch (itérations).

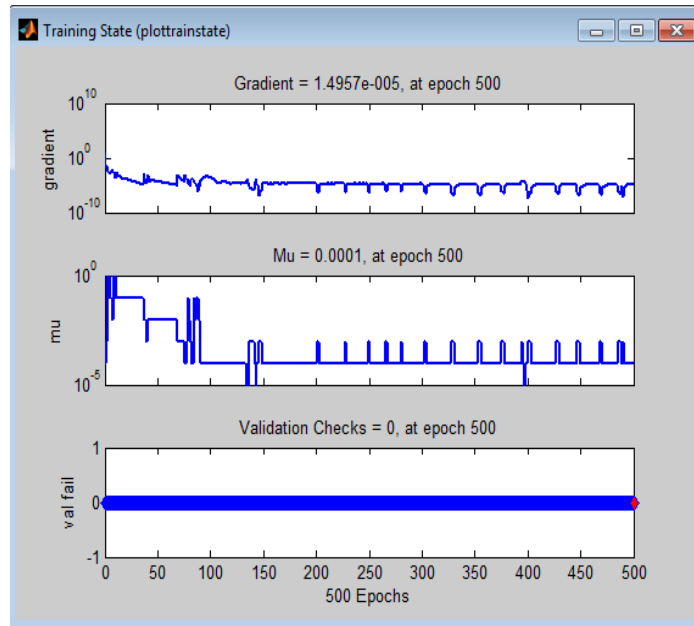


Figure.B.4. Training State

Cette fenêtre représente des figures de gradient avec les nombres de epoch. À chaque fois le gradients et petite le system Réduire le taux d'erreur.

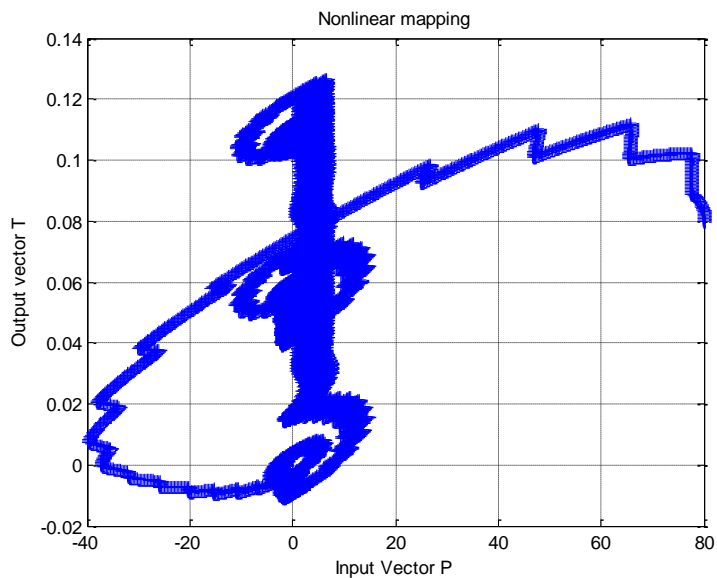


Figure.B.5. La sortie de la fonctionne de transfère

On a dans cette figure la sortie de la fonctionne de Transfer elle représente la division de la sortie de system avec l'entrée.