

**République Algérienne Démocratique et Populaire**

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Mouloud MAMMERY, Tizi-Ouzou**



Faculté de Génie Electrique et d'Informatique

**MEMOIRE DE FIN D'ETUDES**

En vue de l'obtention du diplôme de

***MASTER ACADEMIQUE EN AUTOMATIQUE  
OPTION: COMMANDE DES SYSTEMES***

***Thème***

***Étude et Réalisation d'une transmission sécurisée  
de donnée par le chaos sur une carte FPGA.***

Proposé et Dirigé par :

***M. KARA***

Présenté par :

***BOUALEM TAOUS  
BOUKSIL NAIMA***

Soutenu le : / 09 /2013

***Promotion 2013***

# Remerciements

# *Remerciement*

*EN premier lieu, nous tenons à remercier notre DIEU pour nous avoir donné la force et le courage pour accomplir ce travail.*

*Nous tenons à remercier notre promoteur, Monsieur KARA, à qui nous sommes reconnaissantes pour ses remarques et ses conseils.*

*Nous somme aussi reconnaissantes envers tous les Enseignants du département automatique, également l'ensemble du personnel du laboratoire de conception et conduite des systèmes de production L2CSP.*

*Nos tenons à remercier chaleureusement les membres du jury de nous faire l'honneur d'accepter d'évaluer ce Projet.*

Dédicace

# *Dédicace*

*Je dédie le fruit de mes années d'études à mes très chers parents qui m'ont tout donné leur amour et leurs sacrifices éternels pour que je puisse suivre mes études dans des bonnes conditions et qui ne cessent pas de m'encourager et de veiller pour mon bien, sans leurs soutient ce travail n'aurait jamais vu le jour.*

*À la Mémoire de Mes très Chères sœurs.*

*À ceux qui sont la source de mon inspiration et mon courage, à qui je dois de l'amour et de la reconnaissance :*

- ✓ *Mes frères : Mohamed et sa famille, Ali et sa famille, Amar et sa famille, Abdelleh et sa famille, Nassar et Hamza.*
- ✓ *Mes Amies : samira, Nabila, Amar, Sofiane, Brahim, Ahcen, Hani, hayat, Amina, Nouara, lylia, Yasmine, Kamylia, Amel, Chahinaz, Aicha, Dyhia, Nacira, Kenza, Razika, et Tous mes amis sans exception.*
- ✓ *À Mes chères Malaké et hocine.*
- ✓ *Ma Binôme NAIMA.*
- ✓ *À toutes ces personnes et à celles que j'ai peut-être oubliées j'adresse mes sentiments les plus chaleureux.*

TAOUS

# *Dédicace*

*Je dédie le fruit de mes années d'études à mes très chers parents qui m'ont tout donné leur amour et leurs sacrifices éternels pour que je puisse suivre mes études dans des bonnes conditions et qui ne cessent pas de m'encourager et de veiller pour mon bien, sans leurs soutient ce travail n'aurait jamais vu le jour.*

*À ceux qui sont la source de mon inspiration et mon courage, à qui je dois de l'amour et de la reconnaissance :*

*+ Mes très chères sœurs : Kahina, Hamida, Chafia, Dyhia, Malika et son époux Djilali.*

*+ Mes frères : Tayeb, Tahar, Akli et Karim.*

*+ Mes neveux : Hacen, Luis et Elissa.*

*+ Mes Amies : samira, Nouara, Hafida, Nabila, lylia, Nacira, Kamylia, Aicha, samira, Chahinaz, Ghania, Amina, Kenza, Razika, Amel, Souad, Samia, Nanou, Amar, Nourdin et Tous mes amis sans exception.*

*+ Ma Binôme et aime Taous qui m'a accompagné tout au long de cette période ainsi sa famille.*

*+ À toutes ces personnes et à celles que j'ai peut-être oubliées j'adresse mes sentiments les plus chaleureux,*

*NAIMA*

# Notation

## *Notation :*

---

$\mathbb{R}$  : Ensemble des nombres réels.

$\mathbb{R}^+$  : Ensemble des nombres positifs.

$\mathbb{R}^n$  : Espace vectorielles de dimension  $n$ .

$\mathbb{R}^p$  : Espace vectorielles de dimension  $p$ .

$\mathbb{N}$  : Ensemble des nombres naturels.

$\mathbb{N}^*$  : Ensemble des entiers naturels non nuls.

$\mathbb{Z}$  : Ensemble des nombres relatifs (entiers négatifs et positifs).

$L_f h(x) = \sum_{i=1}^n f_i(x) \frac{\partial h}{\partial x_i}(x)$  : La dérivée de  $h$  log de  $f$ .

$L_f^i h$  :  $i$ ème dérivée du vecteur d'état.

$dL_f^i h$  : Le différentiel de  $L_f^i h$ .

$\dot{x}$  : Dérivée du vecteur d'état  $x$ .

$\lambda_i$  : Variation d'exposant de Lyapunov selon  $i$ .

$\hat{x}$  : Vecteur  $x$  estimé.

$\dot{\hat{x}}$  : Dérivée du vecteur  $x$  estimé.

$\|x\|$  : Norme euclidienne du vecteur  $x$ .

# Liste des acronymes et abréviations

## Table des Figures

---

Figure (I.1) : Illustration de la propriété de sensibilité aux conditions initiales pour l'état $x(t)$ ...	4
Figure (I.2) : Attracteurs réguliers .....	6
Figure (I.3) : Attracteur de Lorenz en 2 et 3 dimensions.....	7
Figure (I.4) : Effet ailes de papillon. ....	8
Figure (I.5) : État chaotique $x$ de système de Rössler .....	9
Figure (I.6) : Attracteur chaotique de Rössler .....	9
Figure (I.7) : Exposant de Lyapunov .....	12
Figure (I.8) : Exposant de Lyapunov pour le système de Lorenz.....	13
Figure (I.9) : État $x(t)$ du système de Lorenz comparé à un signal périodique.....	14
Figure (I.10) : le spectre de puissance en régime périodique et en régime chaotique.....	15
Figure (I.11) : Le principe de la section de Poincaré.....	16
Figure (I.12) : Section de Poincaré.....	17
Figure (I.13) : Intersections de la trajectoire de l'attracteur de Rössler avec un plan p d'équation $y = 0$ ( $x \leq 0$ ). ....	18
Figure (I.14) : Attracteurs du modèle de Lorenz pour différentes valeurs de ces Paramètres .....	19
Figure (I.15) : dédoublement de période de l'attracteur du modèle de Rössler.....	20
Figure (I.16) : Le Diagramme de Bifurcation. ....	21
Figure (II.1) : système de communication chaotique .....	24
Figure (II.2) : système maître esclave pour réaliser la synchronisation.....	25
Figure (II.3) Synchronisation par un contrôle en boucle fermée .....	26
Figure (II.4) : Synchronisation par l'inversion du système.....	26
Figure (II.5) : Structure de synchronisation par décomposition en sous-système proposée par Pocora et Carroll.....	27
Figure(II.6) : synchronisation impulsive .....	28
Figure (II.7) : principe de la synchronisation à base d'observateurs .....	28
Figure (II.8) : Principe de l'observateur .....	29
Figure(II.9) : Principe de la synchronisation par boucle fermée .....	32
Figure (II.10) : Cryptographie et cryptanalyse.....	36
Figure(II.11) : Cryptage par la méthode additive .....	37
Figure(II.12) : Cryptage par la méthode d'inclusion .....	37
Figure(II.13) : cryptage par commutation.....	38
Figure(II.14) : Transmission à deux voies .....	39
Figure(III.1) : représentation d'un oscillateur .....	40
Figure (III.2) : L'exposant de Lyapunov pour le système de $QI$ .....	41
Figure (III.3) : Le spectre de Puissance pour le système de $QI$ .....	42
Figure(III.4) : Schéma de réalisation du système hyperchaotique de $QI$ .....	43
Figure (III.5) : Graphe de l'état $x_1(t)$ du système de $QI$ .....	47
Figure (III.6) : Graphe d'état $x_2(t)$ du système de $QI$ .....	47
Figure (III.7) : Graphe d'état $x_3(t)$ du système de $QI$ .....	48
Figure (III.8) : Graphe d'état $x_4(t)$ du système de $QI$ .....	48
Figure (III.9) : Les attracteurs de phase du système de $QI$ observés sous Matlab.....	49
Figure (III.10) : Attracteur en trois dimensions .....	50
Figure (III.11) : Graphe d'état $x_{1s}$ .....	60
Figure (III.12) : Graphe d'état $x_{2s}(t)$ .....	60
Figure (III.13) : Graphe d'état $x_{3s}$ .....	61

## Table des Figures

---

Figure (III.14): Graphe d'état $x_{4s}$ .....	61
Figure (III.15): Allures des erreurs entre les états des deux systèmes .....	62
Figure (III.16) : Comparaison entre le message et la sortie de récepteur .....	64
Figure (VI.1) : Les différents types de mémoires .....	65
Figure (VI.2) : Classification des circuits numériques .....	66
Figure (IV.3) : le diagramme de différents types de circuits logiques programmables .....	67
Figure (VI.4) : La structure de base d'une PLDs .....	68
Figure (VI.5) : Physionomie d'un CPLD .....	69
Figure (IV.6) : physionomie d'un FPGA .....	70
Figure (IV.7) : Structure générale d'une FPGA .....	70
Figure (VI.8) : Statistique de marché occupée par les vendeurs des FPGAs .....	71
Figure (VI.9) : Exemple d'implémentation sur LUT .....	73
Figure (VI.10) : Exemple d'implémentation sur des multiplexeurs .....	74
Figure (VI.11) : Classification des circuits FPGAs selon leurs configurations .....	74
Figure (VI.12) : La carte DE2 .....	76
Figure (V.1) : Présentation de la bibliothèque DSP-Builder .....	78
Figure (V.2) : le schéma de simulation de l'émetteur .....	80
Figure (V.3): Graphe de l'état $x_1(t)$ du l'émetteur .....	81
Figure (V.4) : Graphe de l'état $x_2(t)$ du l'émetteur .....	81
Figure (V.5) : Graphe d'état $x_3(t)$ du l'émetteur .....	82
Figure (V.6) : Graphe d'état $x_4(t)$ du l'émetteur .....	82
Figure (V.7) : Les attracteurs en deux dimensions .....	83
Figure (V.8) : Les attracteurs en trois dimensions .....	85
Figure (V.9) : Le schéma de simulation Émetteur-Récepteur .....	87
Figure (V.10) : le graphe d'état $x_{1s}$ .....	88
Figure (V.11) : le graphe d'état $x_{2s}(t)$ .....	88
Figure (V.12) : le graphe d'état $x_{3s}$ .....	89
Figure (V.13) : le graphe d'état $x_{4s}(t)$ .....	89
Figure (V.14) : Comparaison entre le message et la sortie de récepteur .....	90
Figure (V.15) : Le block TestBench .....	91
Figure (V.16) : La configuration du bloc TestBench .....	91
Figure (V.17) : Simulation du programme sur le block TestBench .....	92
Figure (V.18) : Généré le code VHDL .....	93

# Liste des Figures

## *Liste des acronymes et abréviations*

---

*ASIC: Application Specific Integrated Circuit.*

*CI: Condition Initial.*

*CPLD ou EPLD: Erasable Programmable Logic Device.*

*CLB: Configurable Logic Bloc.*

*EP-ROM: Erasable Programmable Read Only Memory.*

*EER-ROM: Electrically Erasable Programmable Read Only Memory.*

*FPGA: Field Programmable Gate Array.*

*GAL: Generic Array logic.*

*ISP: In System Programmable.*

*IOB: Input Output Bloc.*

*IP: Proportionnel Intégral.*

*LUT: Look Up Tables.*

*MLT: Modulation de Largeurs d'Impulsions.*

*MUX : Multiplexeur.*

*ROM: Read Only Memory.*

*S-RAM: Static Random Memory.*

*SOS: Systems on Chips.*

*OLMC: output Logic Macro Cell.*

*PAL: programmable Array Logic.*

*P-Rom: Programmable Read Only Memory.*

*PLD: Programmable Logic Device.*

# Sommaire

# Sommaire

---

Introduction générale .....	01
-----------------------------	----

## Chapitre I : Généralité sur les systèmes chaotiques

I.1 Introduction .....	03
I.2 La théorie de chaos .....	03
I.3 caractérisation de chaos .....	03
A- La sensibilité aux conditions initiales .....	04
B- l'attracteur étrange .....	05
C- L'exposant de lyapunov .....	10
D- Le déterminisme et l'imprédictibilité .....	15
E- L'aspect aléatoire .....	15
F- Le spectre de puissance .....	16
G- La fonction d'auto corrélation .....	16
I.4 Les sections de Poincaré .....	17
I.5 La Bifurcation .....	19
I.6 Conclusion.....	22

## Chapitre II : Les méthodes de synchronisation des systèmes chaotique

II.1 Introduction .....	23
II.2 La synchronisation.....	23
II.2.1 Définition.....	23
II.2.2 Système de communication chaotique .....	24
II.2.3 Méthode de synchronisation .....	25
A- Synchronisation par la boucle fermée .....	25
B- Synchronisation par l'inversion du système .....	26
C- Synchronisation identique .....	27
D- Synchronisation par couplage.....	28
E- Synchronisation impulsive .....	28
F- Synchronisation à l'aide d'observateur.....	28
II.3 Le chaos dans la transmission sécurisée.....	35
II.3.1 La définition de cryptage .....	35
II.3.2 Les techniques de cryptage .....	36
a) Le cryptage par addition .....	36
b) Le cryptage par inclusion .....	37
c) Le cryptage par commutation .....	37
II.3.3 La transmission à deux voies .....	38
II.4 Conclusion .....	39

# Sommaire

---

## Chapitre III : Synchronisation de système hyperchaotique de Qi

III.1 Introduction .....	40
III.2 Emetteur .....	40
III.2.1 Définition d'un oscillateur électronique .....	40
III.2.2 Système hyperchaotique de Qi .....	41
III.2.2.1 Caractérisation du chaos dans le système de Qi .....	41
a- Exposant de lyapunov .....	41
b- Spectre de puissance .....	42
c- Le schéma électrique du système Qi .....	42
d- Etude du montage électrique .....	43
III.2.3 Simulation sur Matlab .....	47
a- Visualisation des états .....	47
b- Visualisation des Attracteurs .....	49
c- Visualisation des Attracteurs .....	50
III.3 Le Récepteur .....	51
III.3.1 Condition du rang d'observabilité .....	51
III.3.2 Synchronisation du système de Qi .....	53
III.3.3 Simulation sous Matlab .....	60
a) Visualisation des états estimés .....	60
b) Visualisation des écarts entre les états des deux systèmes .....	62
c) Récupération du message .....	64
III.4 Conclusion .....	64

## Chapitre IV : Présentation des FPGAs

IV.1 Introduction .....	65
IV.2 Les techniques de mémorisation .....	65
IV.3 Les techniques de mémorisation .....	66
IV.4 Les circuits logiques programmables .....	67
IV.5 Présentation de FPGAs .....	69
IV.6 Principaux fondeurs d'FPGA .....	71
IV.7 Classification des FPGAs .....	71
IV.8 Critères du choix des FPGAs .....	72
IV.9 Différents domaines d'applications des FPGAs .....	72
IV.10 Les familles architecturales des FPGAs .....	73
IV.11 Programmation et configuration des circuits FPGAs .....	74
IV.12 Avantages et Inconvénients des FPGAs .....	75
IV.13 Architecteur de la carte FPGA DE2 .....	76
IV.14 Conclusion .....	77

# Sommaire

---

## Chapitre V : La Réalisation

V.1 Introduction .....	78
V.2 Présentation de DSP-Builder .....	78
V.3 Simulation sous Matlab .....	79
V.3.1 Emetteur .....	79
a) Shémas de simulation de l'émetteur .....	80
b) Visualisation des états .....	81
c) Visualisation des attracteurs .....	85
d) Visualisation en trois dimensions .....	86
V.3.2 Récepteur .....	88
a) Le schéma de Simulation .....	88
b) Visualisation des états .....	90
c) Récupération du message .....	93
V.4 Visualisation des signaux sur la carte FPGA .....	95
V.5 Conclusion .....	97
Conclusion générale .....	98
Annexe A .....	99
Annexe B .....	103
Annexe C .....	115
Bibliographie	

# Introduction Générale

# Introduction Générale

---

L'emploi du chaos pour la transmission sécurisée de l'information a été considéré dans les dernières années comme une solution très prometteuse pour augmenter les performances des systèmes de transmission actuels. Ainsi, on trouve dans la littérature une multitude d'applications et d'études réalisées concernant plusieurs aspects de la transmission. Grâce à ses caractéristiques quasi stochastiques, le chaos offre une solution possible pour les systèmes à probabilités réduites de détection et d'interception.

La sécurisation de la chaîne de transmission devient de plus en plus nécessaire avec l'évolution des communications en termes de nombre d'utilisateurs et de la nature d'information à transmettre. Actuellement, tout système de communication performant nécessite un système de sécurisation afin de le protéger des attaques possibles. Pour cela, de nouvelles méthodes de cryptage sont développées. Le cryptage des informations est maintenant utilisé pour interdire l'accès ou la modification des informations sensibles et garantir la confidentialité dans les communications. Certaines de ces nouvelles méthodes utilisent le chaos dans les systèmes de communication.

En 1963, le météorologue Edward Lorenz expérimentait une méthode lui permettant de prévoir les phénomènes météorologiques. C'est par hasard qu'il observa qu'une modification minime des données initiales pouvait changer de manière considérable ses résultats. Lorenz venait de découvrir le phénomène de sensibilité aux conditions initiales. Les systèmes répondant à cette propriété seront à partir de 1975 dénommés : systèmes chaotiques. C'est donc au cours des années soixante-dix que la théorie du chaos a pris son essor, cependant, les travaux de certains scientifiques menés bien avant cette découverte vont être très utiles à la compréhension de la dynamique chaotique. En effet, vers la fin du XIXe siècle le mathématicien, physicien et philosophe français Henri Poincaré avait déjà mis en évidence le phénomène de sensibilité aux conditions initiales lors de l'étude astronomique du problème des trois corps (une étoile et deux planètes). On trouve dans le calcul des probabilités de Henri Poincaré l'affirmation suivante :

« Une cause très petite, qui nous échappe, détermine un effet considérable que nous ne pouvons pas ne pas voir, et alors nous disons que cet effet est dû au hasard ».

Toujours au XIXe siècle, le mathématicien russe Alexandre Lyapunov effectue des recherches sur la stabilité du mouvement. Il introduit l'idée de mesurer l'écart entre deux trajectoires ayant des conditions initiales voisines, lorsque cet écart évolue exponentiellement on parle de sensibilité aux conditions initiales.

Les travaux de Lyapunov, d'abord tombés dans l'oubli, seront plus tard très précieux pour étudier certains aspects de la théorie du chaos. Les travaux des prédécesseurs de Lorenz ont donc été très importants pour la compréhension du chaos déterministe, mais il faut souligner que ce qui va permettre aux scientifiques une compréhension plus accrue des systèmes chaotiques c'est l'ordinateur. En effet, les équations différentielles régissant un système chaotique sont nécessairement non linéaires et sans ordinateur, leur résolution est en général impossible.

L'idée d'utilisation du chaos dans les systèmes de communication a été inspirée de la découverte de Picora et Carroll en 1990. Ils ont montré que deux systèmes chaotiques identiques avec conditions initiales différentes peuvent éventuellement se synchroniser s'ils sont couplés d'une certaine manière convenables, c'est-à-dire sous certaines conditions. Le développement des systèmes de communication utilisant le chaos a commencé donc des schémas de synchronisations très simples de circuit électroniques. Par la suite, plusieurs méthodes de synchronisation ont été découvertes. On peut citer : la méthode impulsive, la méthode par observateur, synchronisation par couplage (unidirectionnel, bidirectionnel) ...

Il est bien connu que les systèmes de communication traditionnels comportaient deux parties, respectivement appelées émetteur et récepteur. Le signal de sortie de l'émetteur est

# Introduction Générale

---

transmis par le canal public au récepteur, ce signal en question contient comme information le message confidentiel.

Enfin, ce message peut être recouvert en synchronisant le récepteur avec le signal envoyé par l'émetteur. Dans la théorie de l'observation, le récepteur est un observateur, qui est une copie du modèle dynamique de l'émetteur plus un terme de correction. De nombreux schémas de transmission ont été proposés par le passé : la méthode par addition, par inclusion, par commutation.

Dans ce travail, nous utiliserons le chaos pour chiffrer une information dans une transmission sécurisée de donnée que nous allons concevoir théoriquement et réaliser sur une carte électronique "FPGA". Pour ce faire, nous utiliserons un système hyperchaotique appelé système de Qi. Deux systèmes de Qi identiques seront donc utilisés comme émetteur et récepteur de signaux chaotiques. L'émetteur étant un générateur de signaux chaotiques et le récepteur sera un circuit identique à l'émetteur, les deux systèmes sont synchronisés à l'aide d'observateur afin de récupérer en sortie le signal chaotique généré par l'émetteur. Pour sécuriser un message à l'aide de ce système de transmission, nous allons noyer ce message (par exemple un signal sinusoïdal) dans le signal chaotique généré par l'émetteur, en l'additionnant dans un état de l'émetteur. Le signal reçu par le récepteur sera donc un signal chaotique dans lequel le message est caché. L'extraction du message se fera par sa soustraction du signal chaotique.

Ce travail est organisé de la façon suivante :

Le chapitre I est une introduction à la théorie du chaos, on parlera alors de ses différentes propriétés et de son comportement spécifique.

Dans le chapitre II, nous allons, dans une première partie, étudier la technique de synchronisation que nous utiliserons pour concevoir notre transmission, qui se fera à l'aide d'observateur. En deuxième partie nous allons donner des généralités sur la transmission sécurisée et les méthodes de cryptage pouvant s'appliquer à notre système.

Le chapitre III est consacré à l'étude théorique du système hyperchaotique de Qi, puis à la conception du récepteur et à l'insertion du message puis son extraction. Le système de transmission sécurisée obtenu sera simulé sous Matlab pour anticiper la possibilité du fonctionnement de la réalisation pratique.

Le chapitre IV présente les circuits logiques programmables avec ses différentes classes et en particulier les circuits FPGA, leur architecture et leur principe de fonctionnement.

Dans le dernier chapitre, nous allons réaliser en pratique la transmission du système simulé dans le chapitre III sur une carte FPGA.

Et enfin, nous terminerons par une conclusion générale.

# Chapitre I

## ***1.1- Introduction :***

Les systèmes dynamiques chaotiques (étranges) sont depuis longtemps connus dans le domaine des mathématiques, mais c'est seulement au cours de la dernière décennie que les applications concrètes se sont multipliées. Depuis quelques années la théorie des systèmes chaotiques a été appliquée dans le domaine des communications et notre étude se focalise sur l'usage du chaos pour transmettre de l'information. Dans cette perspective, ce chapitre est destiné à l'introduction de quelques outils de base associés à ces deux domaines dont l'intersection fait l'objet de notre étude. Ce chapitre peut être vu comme une synthèse bibliographique compacte sur les transmissions chaotiques, dont l'intérêt principal est de mieux mettre en évidence l'originalité des résultats que nous proposerons par la suite. Le lecteur avisé pourra directement consulter le chapitre suivant.

Dans ce chapitre, nous présentons les aspects généraux des systèmes dynamiques chaotiques et ses caractéristiques.

## ***1.2 La théorie de chaos :***

### ***1- Définition :***

Il n'existe pas de définition à la fois formelle et générale du chaos. Cependant, le chaos est défini généralement comme un comportement particulier d'un système dynamique qui inclut :

- La non-linéarité : Si un système est linéaire, il ne peut pas être chaotique.
- Le déterminisme : Un système chaotique a des règles fondamentales déterministes (capacité à prédire le futur d'un phénomène à partir d'un événement passé ou présent).
- La sensibilité aux conditions initiales : de très petits changements sur l'état initial peuvent mener à un comportement radicalement différent dans son état final.
- La non-intégrabilité : La solution est toujours une courbe, on ne peut donc pas expliciter l'équation analytique.

Le chaos est défini généralement comme un comportement particulier d'un système dynamique déterministe non linéaire.

## ***1.3- Caractérisation du chaos :***

Il existe plusieurs définitions possibles du chaos. Ces définitions ne sont pas toutes équivalentes, mais elles convergent vers certains points communs caractérisant le chaos.

Les caractéristiques des systèmes chaotiques sont :

- a- la sensibilité aux conditions initiales.
- b- L'attracteur étrange.
- c- L'exposant de Lyapunov.
- d- Le déterminisme et l'imprédictibilité.
- e- L'espace aléatoire.
- f- Le spectre de puissance.
- g- La fonction d'auto corrélation.

## A- La sensibilité aux conditions initiales [14][25] :

Les systèmes chaotiques sont extrêmement sensibles aux perturbations, on peut illustrer ce fait par l'effet papillon, découvert par Poincaré et popularisé par le météorologue Edward Lorenz. L'évolution d'un système dynamique chaotique est imprédictible en ce sens qu'elle est sensible aux conditions initiales, pour deux conditions initiales arbitraires très voisines initialement, les deux trajectoires correspondantes à ces données initiales divergentes exponentiellement par suite les deux trajectoires sont incomparables, ce qui rend les systèmes chaotiques imprédictibles à long terme.

### ❖ Illustration par le modèle de Lorenz :

il s'agit d'un système dynamique non linéaire de dimensions 3, ce système est défini par :

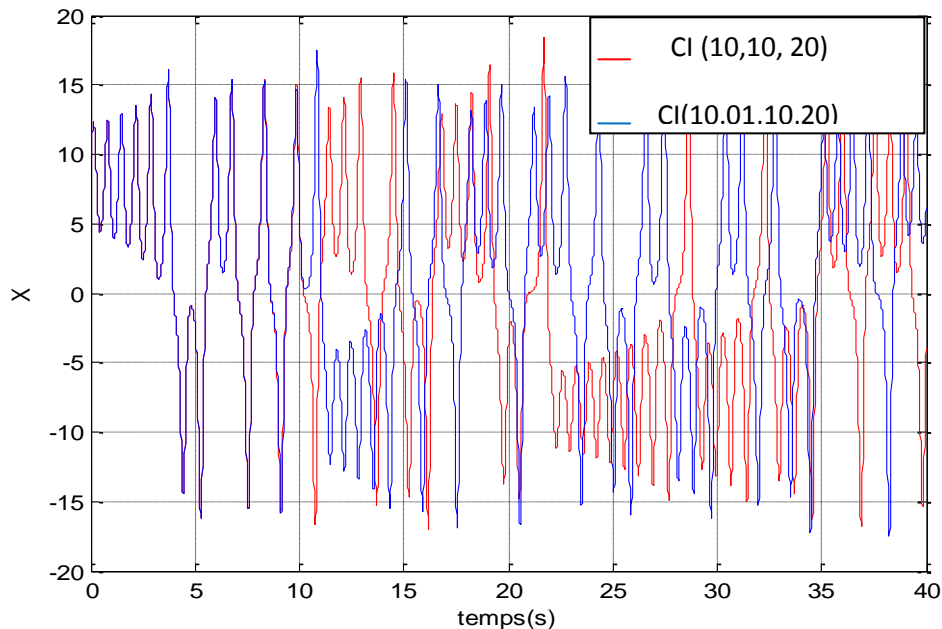
$$\begin{cases} \frac{dx}{dt} = a(y-x) \\ \frac{dy}{dt} = bx-y-xz \\ \frac{dz}{dt} = xy-cz \end{cases} \quad (I.1)$$

$(x,y,z)^T$  le vecteur d'état, a, b et c des paramètres du système.

a = 10, b=28, c=8/3 sont les valeurs pour lesquels le système présente un aspect aléatoire.

Pour deux conditions initiales très proches  $(x_0, y_0, z_0)=(10,10,20)$  et  $(x_0, y_0, z_0)=(10.01,10,20)$

Nous allons simuler ce modèle sous MATLAB



Figure(I.1) : illustration de la propriété de sensibilité aux conditions initiales pour l'état  $x(t)$

# Chapitre I : Généralité sur les systèmes chaotiques

---

## ❖ *Observation :*

On remarque que pour deux conditions initiales très proches les deux trajectoires évoluent au départ de la même manière ensuite et très rapidement elles divergent. Donc les systèmes chaotiques montrent une sensibilité exponentielle aux conditions initiales c'est-à-dire une petite différence entre deux conditions initiales conduits à une divergence rapide des trajectoires au cours du temps comme est montré à la figure (I.1).

## *B- l'attracteur étrange [25] :*

### *Définition :*

Un attracteur est la zone de l'espace des phases qui attire les trajectoires d'un système dynamique quelconque. L'attracteur le plus simple est un point, c'est un attracteur d'un système qui évolue à taux constant ; d'autres, attracteurs, peuvent inclure des cycles qui se répètent au cours du temps. Dans les systèmes chaotiques, les trajectoires convergentes vers une région particulière de l'espace appelé attracteur étrange qui est une signature du chaos, c'est ce qui différencie un signal chaotique d'un signal aléatoire. En effet, si un mouvement est aléatoire, les points de la trajectoire remplissent l'espace de phase de manière aléatoire.

Un attracteur est caractérisé par

- son bassin d'attraction.
- Sa dimension fractale.

### ❖ *Le bassin d'attraction :*

Le bassin d'attraction est l'ensemble des points de l'espace des phases qui sont sous l'effet de l'attracteur. Toutes les trajectoires qui commencent à ces points tendent vers l'attracteur après un temps fini. Donc le bassin d'attraction correspond à l'ensemble des points dont les trajectoires convergent vers l'attracteur.

### ❖ *Dimension fractale :*

La dimension d'un espace euclidien correspond au nombre minimum de coordonnées nécessaires, pour spécifier de manière unique la position d'un point ou encore la dimension.

La dimension fractale de l'attracteur selon Hausdorff-Besicovitch se représente comme suit :

Soit un ensemble de points dans l'espace des phases à  $n$  dimensions, cherchons à recouvrir cet ensemble par des hyper cubes de côté  $\epsilon$ , soit le nombre minimal de cubes  $N(\epsilon)$  nécessaire à cette opération qui varie comme :

$$N(\epsilon) = \epsilon^{-D}$$

$D$  étant la dimension fractale, elle est donnée par :

$$D = \lim_{\epsilon \rightarrow 0} \frac{\ln N(\epsilon)}{\ln(1/\epsilon)}$$

Pour  $\epsilon$  petit, le nombre minimal de cubes  $N(\epsilon)$  nécessaires pour recouvrir l'ensemble de point varie avec  $\epsilon$  comme  $\epsilon^{-D}$ , dans le cas où l'ensemble se réduit à un point unique

$$N(\epsilon) = 1$$

Donc  $D = 0$  est la dimension de Hausdorff du point.

# Chapitre I : Généralité sur les systèmes chaotiques

Si l'ensemble est celui des points d'un segment de longueur L.

$$N(\epsilon) = L\epsilon^{-1}$$

D'où  $D = 1$ , alors que pour l'ensemble des points d'une surface d'aire S :

$$N(\epsilon) = S\epsilon^{-2}$$

D'où  $D=2$ .

Par conséquent : le point fixe a une dimension nulle, une oscillation périodique stable est de dimension 1, et un attracteur de période 2 a une dimension 2.

C'est parce que la structure de ces attracteurs est très régulière que leur dimension est entière. Par contre, la dimension d'un attracteur étrange est non entière par exemple la dimension de l'attracteur de Lorenz est de 2.06.

## ❖ Les différents types d'attracteurs:

### a) Attracteurs réguliers :

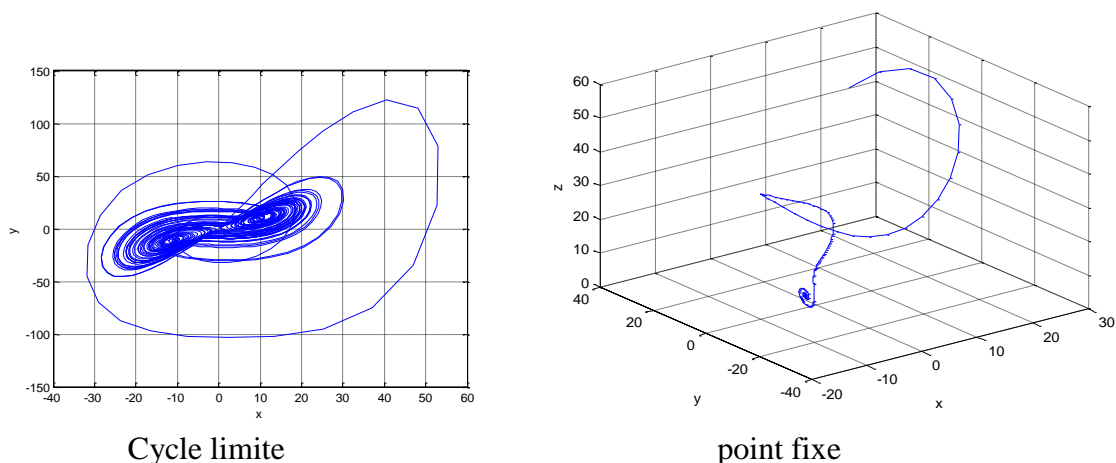
Les attracteurs réguliers caractérisent l'évolution de systèmes non chaotiques, et peuvent être de trois sortes :

- **le point fixe** : c'est le cas le plus courant, et plus simple d'attracteur, dans lequel le système évolue vers un état de repos (point). Notons que seuls les points peuvent être des attracteurs

- **le cycle limite périodique** : il peut arriver que les trajectoires de phase se referment sur elles-mêmes. L'évolution temporelle est alors cyclique, le système présente des oscillations permanentes.

- **le cycle limite pseudo-périodique** : c'est presque un cas particulier du précédent. Le système présente au moins deux périodes. La trajectoire de phase ne se referme pas sur elle-même, mais s'enroule sur une variété de dimension 2 par exemple un tore.

**Exemple :**



Figure(I.2) : Attracteurs réguliers.

## b) Attracteurs étranges :

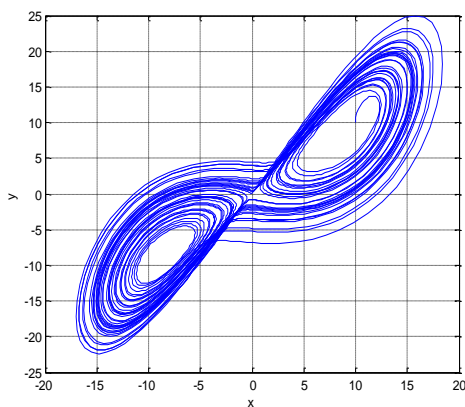
Le terme d'attracteur étrange a été utilisé par la première fois par David Ruelle et Floris Takens en 1971, afin de décrire l'attracteur obtenu par une série de bifurcations d'un système modélisant le courant d'un liquide. En fait, avant l'article de Ruelle et Takens, des attracteurs avaient déjà fait l'objet de publications, mais ils sont restés ignorés. La structure géométrique des trajectoires générées par un système chaotique est extrêmement complexe à cause des étirements, repliements et contractions s'opérant dans une région bornée de l'espace d'état. En effet cette appellation d'attracteur étrange fait appel à leurs propriétés peu communes, qui sont :

- sensibilité aux conditions initiales : deux trajectoires de l'attracteur initialement voisines finissent par s'écarter l'une de l'autre.
- la dimension  $d$  de l'attracteur est fractale (non entière) avec  $2 < d < n$ ,  $n$  est la dimension de l'espace des phases.
- dans l'espace des phases, l'attracteur est de volume nul.
- presque toute trajectoire sur l'attracteur a la propriété de ne jamais passer deux fois sur le même point : chaque trajectoire est presque sûrement apériodique.

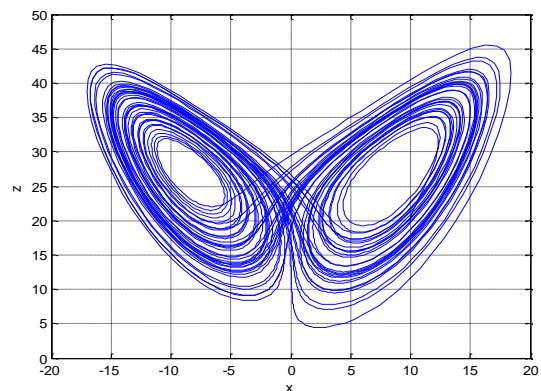
### Exemple1 :

Nous avons choisi l'attracteur de Lorenz, nous allons simuler sur Matlab le système de Lorenz avec  $a=10$ ,  $b=28$ ,  $c=8/3$   $[x_0, y_0, z_0]=[10, 10, 20]$ , et on obtient :

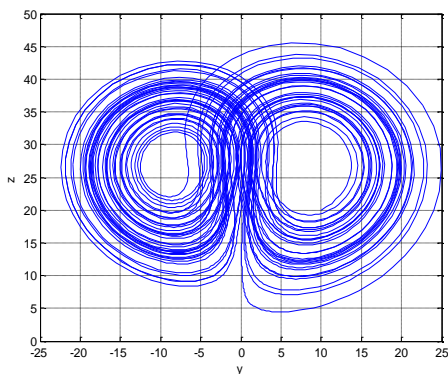
Attracteur en 2 dimensions(x,y)



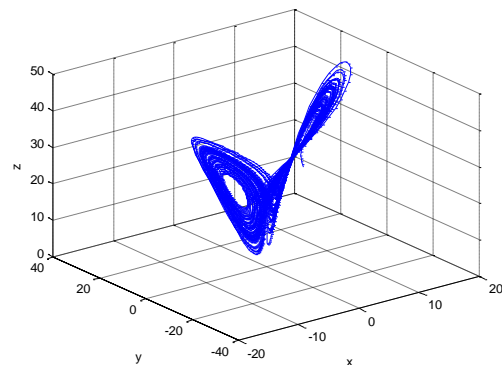
Attracteur en 2 dimensions(x,z)



Attracteur en 2 dimensions (y,z)



Attracteur en 3 dimensions (x,y,z)

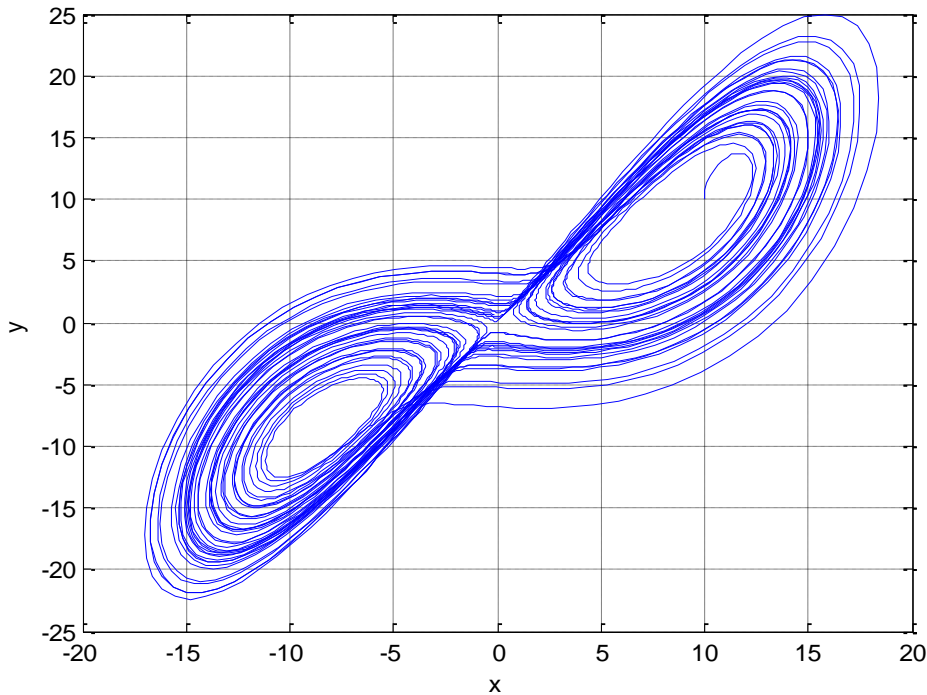


Figure(I.3) : Attracteur de Lorenz en 2 et 3 dimensions.

## ❖ *Effet papillon :*

La figure suivante représente la projection de l'attracteur de Lorenz sur le plan (x,y) avec les paramètres et conditions initiales ci-dessous :

$a=10, b=28, c=8/3$   $[x_0, y_0, z_0]=[10, 10, 20]$ .



**Figure (I.4) : Effet ailes de papillon.**

Lorenz appela cet effet *ail de papillon*. En fait, le titre d'une de ses conférences s'intitulait : « *le battement d'une aile de papillon au Brésil peut-il déclencher une tornade au Texas* ». On remarquera avec amusement que l'attracteur de Lorenz a une forme d'ailes de papillon.

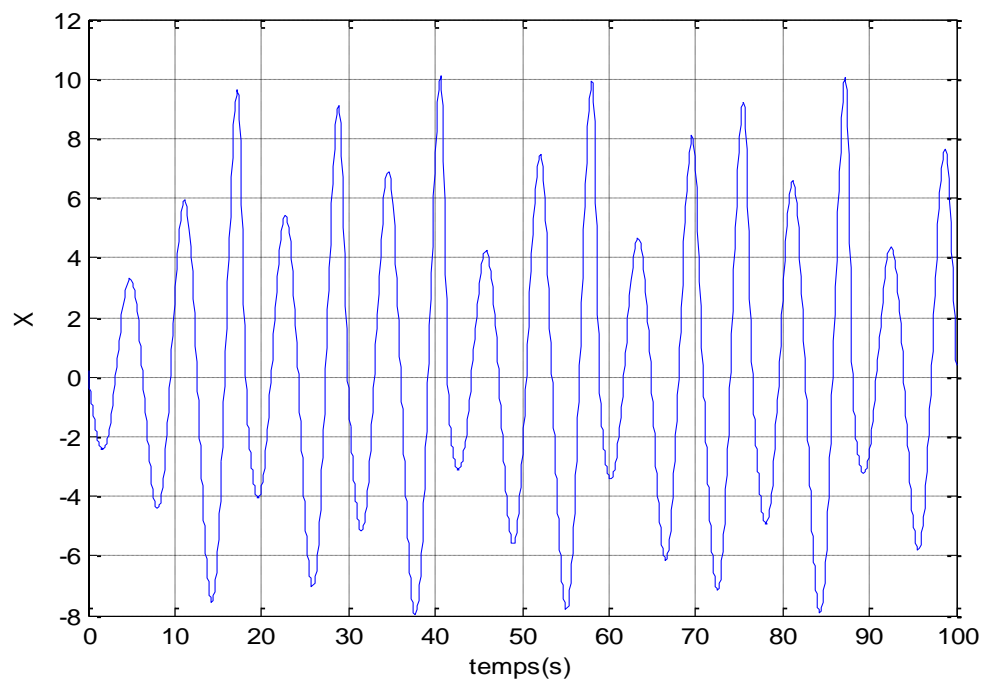
## **Exemple2 :**

Nous avons choisi l'attracteur de Rössler qui est décrit par le système d'équations suivant :

$$\left\{ \begin{array}{l} \frac{dx}{dt} = -(y+z) \\ \frac{dy}{dt} = x+ay \\ \frac{dz}{dt} = b+z(x-y) \end{array} \right. \quad (I.2)$$

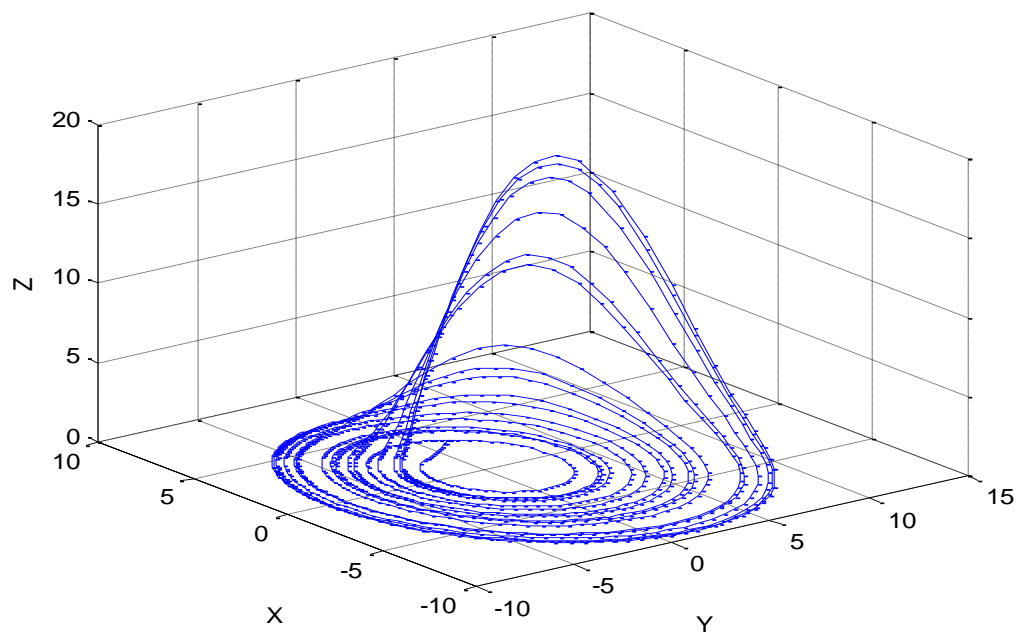
$a = b = 0.2, c = 5$  sont les paramètres du système et  $(x, y, z)$  est le vecteur d'état. Les conditions initiales choisies comme suit :  $[x_0, y_0, z_0] = [0.2, 2, 1]$

- *Évolution dans le temps de l'état  $x$*



**Figure (I.5) : Etat chaotique  $x$  de système de Rössler.**

**-Attracteur de Rössler :**



**Figure (I.6) : Attracteur chaotique de Rössler.**

### C- Exposant de Lyapunov [14][25] :

Certains systèmes dynamiques (systèmes qui décrivent dans l'espace un état qui évolue dans le temps) sont très sensibles aux petites variations de leur condition initiale (SCI). Ces variations peuvent rapidement prendre d'énormes proportions. Le mathématicien russe Alexander Lyapunov s'est penché sur ce phénomène et a développé une quantité permettant de mesurer la vitesse à laquelle ces petites variations peuvent s'amplifier. Cette quantité appelée "exposant de Lyapunov" mesure en fait le degré de sensibilité d'un système dynamique alors les exposants de Lyapunov sont des coefficients qui permettent de mesurer la sensibilité aux conditions initiales d'une série temporelle.

#### ❖ Définition :

Un exposant de Lyapunov est le taux exponentiel moyen de divergence de trajectoires voisines de l'espace des phases, il qualifie le degré de divergence des trajectoires d'un système dynamique non linéaire soumis à des conditions initiales différentes. Les exposants de Lyapunov caractérisent les attracteurs ou le comportement du système non linéaire et notamment son caractère chaotique ou hyperchaotique. Si le système représente un exposant de Lyapunov positif alors le système est chaotique et s'il a deux exposants de Lyapunov positifs alors le système est hyperchaotique.

$$\boxed{\text{Exposant de Lyapunov} = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \ln |(f' x_{i-1})|} \quad (\text{I.3})$$

Où  $x$  est un vecteur de dimension  $n$ ,  $f$  est une fonction vectorielle de dimension  $n$  continue et dérivable.

Considérons un système dynamique quelconque, dont la condition initiale  $x_0$  est affectée d'une erreur infinitésimale  $E_0$ . Après  $n$  itérations, l'erreur initiale  $E_0$  sera donc amplifiée d'un facteur  $\left| \frac{E_n}{E_0} \right|$ . Notons que l'erreur diminue lorsque le facteur est inférieur à 1 et augmente s'il est supérieur à 1.

Puisque  $\left| \frac{E_n}{E_0} \right| = \left| \frac{E_n}{E_{n-1}} \right| \left| \frac{E_{n-1}}{E_{n-2}} \right| \left| \frac{E_{n-2}}{E_{n-3}} \right| \dots \dots \dots \left| \frac{E_1}{E_0} \right|$ , il suffit alors de calculer ce produit pour déterminer la façon dont s'amplifie l'erreur initiale.

Malheureusement, calculer un très grand nombre de produits à l'aide d'un ordinateur amène quelques difficultés. Le logarithme d'un produit correspond à une somme de logarithmes. Utilisons plutôt le logarithme du produit pour compléter cette étude.

$$\ln \left| \frac{E_n}{E_0} \right| = \ln \left( \left| \frac{E_n}{E_{n-1}} \right| \left| \frac{E_{n-1}}{E_{n-2}} \right| \left| \frac{E_{n-2}}{E_{n-3}} \right| \dots \dots \dots \left| \frac{E_1}{E_0} \right| \right)$$

$$\begin{aligned} \ln \left( \left| \frac{E_n}{E_0} \right| \right) &= \ln \left( \frac{E_n}{E_{n-1}} \right) + \ln \left( \frac{E_{n-1}}{E_{n-2}} \right) + \ln \left( \frac{E_{n-2}}{E_{n-3}} \right) + \dots \dots + \ln \left( \frac{E_1}{E_0} \right) \\ &= \sum_{i=1}^n \ln \left| \frac{E_i}{E_{i-1}} \right| \end{aligned}$$

# Chapitre I : Généralité sur les systèmes chaotiques

---

Avant de faire tendre cette dernière quantité vers l'infini, calculons d'abord la moyenne de la somme obtenue. On arrive ainsi à l'exposant de Lyapunov.

$$\lim_{n \rightarrow \infty} \frac{1}{n} \left( \sum_{i=1}^n \ln \left| \frac{E_i}{E_{i-1}} \right| \right)$$

$E_i$  et  $E_{i-1}$  étant de très petites valeurs, le rapport  $\left| \frac{E_i}{E_{i-1}} \right|$  correspond à la dérivée de la fonction associée à l'équation utilisée si naturellement la fonction est dérivable. En effet soit  $f(x_i)$  cette fonction.

$$E_i = f(x_{i-1} + E_{i-1}) - f(x_{i-1})$$

et 
$$\frac{E_i}{E_{i-1}} = \frac{f(x_{i-1} + E_{i-1}) - f(x_{i-1})}{E_{i-1}}$$

Puisque  $\lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} = f'(x)$  alors si  $f(x_i)$  est dérivable, on a

$$\frac{E_i}{E_{i-1}} = \frac{f(x_{i-1} + E_{i-1}) - f(x_{i-1})}{E_{i-1}} = f'(x), \text{ lorsque } E_{i-1} \rightarrow 0$$

Par conséquent

$$\text{Exposant de Lyapunov} = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \ln |(f' x_{i-1})|$$

Lorsque l'exposant de Lyapunov est positif,  $\ln \frac{E_n}{E_0} > 0$  et par conséquent  $\frac{E_n}{E_0} > 1$ .

L'erreur infinitésimale du début ira donc en augmentant. Le système sera dans ce cas sensible aux très petites variations de sa condition initiale, une des caractéristiques des systèmes chaotiques. Si au contraire l'exposant de Lyapunov est négatif, l'erreur infinitésimale du début ira en diminuant. L'erreur initiale n'aura dans ce cas aucun effet à long terme.

### *Remarque :*

Pour le calcul de ces exposants il existe plusieurs méthodes par contre il est difficile de le déterminer par les méthodes analytiques, c'est pour cela qu'on a recours aux simulations numériques.

Le nombre des exposants de Lyapunov est égal à la dimension du système.

### *Exemple :*

Soit le système suivant :

$$\begin{cases} \dot{x}_1 = x_1 - x_1^3 \\ \dot{x}_2 = -x_2 \end{cases}$$

La trajectoire  $\emptyset(x_0, t)$  considérée est définie à partir des conditions initiales  $0 < x < 1$  et  $x_2 = 0$ .

# Chapitre I : Généralité sur les systèmes chaotiques

---

L'intégration de la première variable donne :

$$\begin{aligned}\frac{dx_1}{dt} &= x_1(t) - x_1^3(t) \\ \frac{dx_1}{dt} &= x_1(t)(1 - x_1^2(t))\end{aligned}$$

$$\frac{dx_1(t)}{x_1(t)(1-x_1^2)} = dt$$

Posant  $\varphi(t) = (1 - x_1^2(t)) \Rightarrow d\varphi(t) = -2x_1(t)dx_1(t)$

Soit  $dx_1(t) = \frac{d\varphi(t)}{-2\sqrt{1-\varphi(t)}}$ , d'où

$$\frac{d\varphi(t)}{dt} = \frac{d\varphi(t)}{dx_1(t)} \frac{dx_1(t)}{dt} = -2\sqrt{1-\varphi(t)}\sqrt{1-\varphi(t)}\varphi(t) = -2(1-\varphi(t))\varphi(t)$$

En écrivant:  $\frac{d\varphi(t)}{(1-\varphi(t))\varphi(t)} = -2dt$ , on trouve :  $\ln\left(\frac{1}{\varphi(t)} - 1\right) = -2t$

Et alors  $x_1^2(t) = \frac{e^{-2t}}{1+e^{-2t}}$

La linéarisation du système d'équation autour d'un point  $x_p$  donne

$$\begin{aligned}\delta\dot{x}_1 &= (1 - 3x_1^2)\delta x_1 \\ \delta\dot{x}_2 &= -1\delta x_2\end{aligned}$$

$$\text{Donc } \begin{pmatrix} \delta\dot{x}_1 \\ \delta\dot{x}_2 \end{pmatrix} = \begin{bmatrix} 1 - 3x_1^2 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix}$$

$$\frac{\delta\dot{x}_2}{\delta x_2} = -1 \Rightarrow x_2 = e^{-t}$$

On remplace  $x_1(t)$  (et donc sa dérivée par calcul) par la solution obtenue précédemment puis on intègre ce dernier ce qui permet d'obtenir la matrice de la solution fondamentale  $\Phi_t$

On obtient ainsi le système suivant :

$$\frac{\delta\dot{x}_1}{\delta x_1} = \left(1 - 3\frac{e^{-2t}}{1+e^{-2t}}\right), \text{ soit par intégration : } \ln x_1 = \left(t + \frac{3}{2} \ln(e^{-2t} + 1)\right)$$

**Par conséquent :**

$$x_1 = e^t e^{3/2 \ln(e^{-2t}+1)} = e^t (e^{-2t} + 1)^{3/2}$$

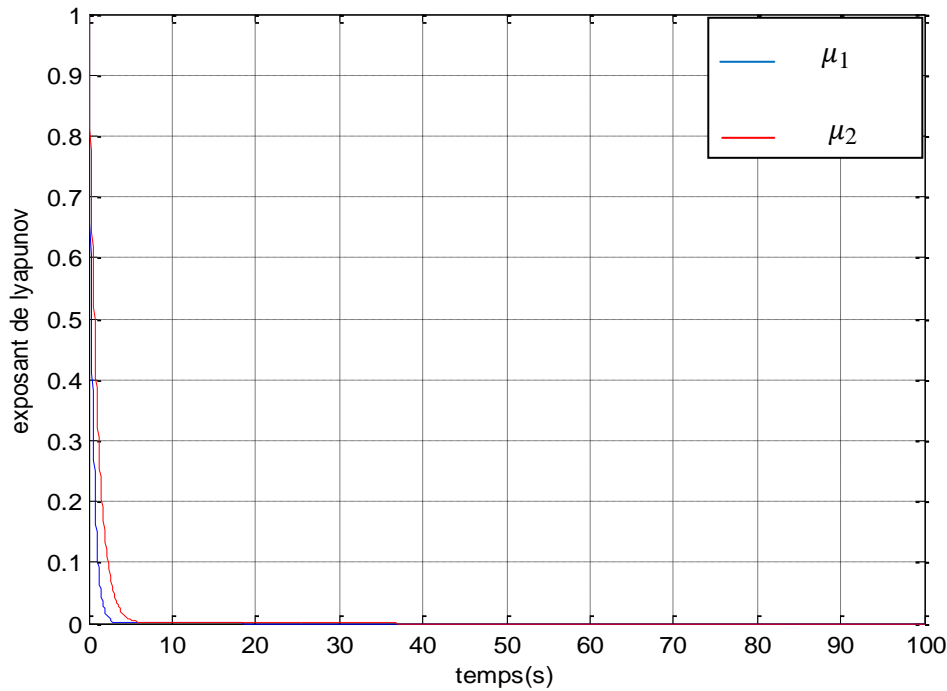
$$\Phi_t = \begin{bmatrix} e^t ((e^{-2t} + 1))^{3/2} & 0 \\ 0 & e^{-t} \end{bmatrix}$$

# Chapitre I : Généralité sur les systèmes chaotiques

Les valeurs propres de cette matrice sont  $\mu_1=e^t((e^{-2t} + 1))^{3/2}$  et  $\mu_2=e^{-t}$ , les exposants de Lyapunov sont donnés par  $\lambda_i=\lim_{t \rightarrow \infty} \frac{1}{t} \ln \mu_i(t)$

Alors on trouve

$$\begin{cases} \lambda_1=-2 \\ \lambda_2=-1 \end{cases}$$



**Figure (I.7) : Exposant de lyapunov.**

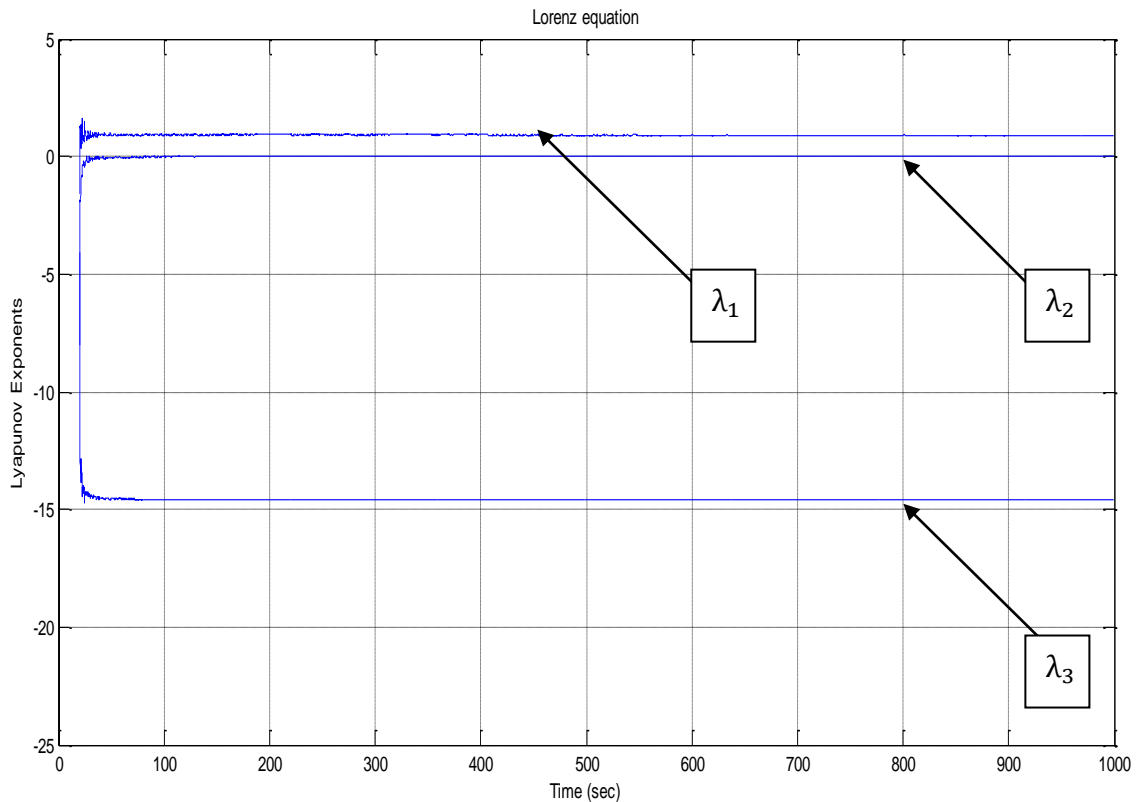
❖ **Observation :**

On remarque que la trajectoire qui décrit l'exposant de lyapunov négatif  $\lambda_1$  même  $\lambda_2$  diminue à une vitesse exponentielle au début, puis se stabilise à une valeur fixe.

❖ **Comportement d'un système dynamique non linéaire en fonction des exposants de lyapunov :**

- $\lambda_n \leq \dots \leq \lambda_1 < 0$  existence d'un point fixe (les exposants de lyapunov négatifs).
- $\lambda_n \leq \dots \leq \lambda_2 < 0, \lambda_1 = 0$  l'attracteur est un cycle limite (régime périodique).
- $\lambda_n \leq \dots \leq \lambda_3 < 0, \lambda_1 = \lambda_2 = 0$  l'attracteur est un tore d'ordre 2 (régime quasi-périodique à 2 fréquences).
- $\lambda_1 = \lambda_2 = \dots = \lambda_k, \lambda_n \leq \dots \leq \lambda_{k+1} < 0$  l'attracteur est un tore d'ordre k (régime quasi-périodique à k fréquences).
- $\lambda_n \leq \dots \leq \lambda_2 < 0, \lambda_1 > 0$  l'attracteur est chaotique.
- $\lambda_n \leq \dots \leq \lambda_{k+1} < 0, \lambda_1 > \lambda_2 > \dots > \lambda_k > 0$  l'attracteur est hyper chaotique.

**Exemple :**



**Figure (I.8) : Exposant de lyapunov pour le système de Lorenz.**

Le premier exposant  $\lambda_1 = 0.906799$ , le deuxième  $\lambda_2 = -0.00180499$  et le troisième  $\lambda_3 = -14.5733$

On remarque que la trajectoire que décrit le premier exposant de Lyapunov ( $\lambda_1 = 0.906799$ ) augmente à une vitesse exponentielle au début puis se stabilise à une valeur fixe. Le système de Lorenz contient un exposant de Lyapunov positif alors le système de Lorenz est un *système chaotique*.

❖ **Les exposants de Lyapunov pour un attracteur étrange :**

Une des particularités du chaos est son extrême sensibilité aux conditions initiales. Un attracteur étrange possédera toujours au moins un exposant de Lyapunov positif avec la propriété  $\sum_{k=0}^n \lambda_k < 0$ . De plus, pour un attracteur étrange, un des exposants de Lyapunov est toujours nul. Cela signifie que pour respecter la condition  $\sum_{k=0}^n \lambda_k < 0$  un attracteur étrange doit avoir au minimum trois exposants de Lyapunov. Donc, un système continu dans le temps doit être au moins de dimension trois pour produire du chaos.

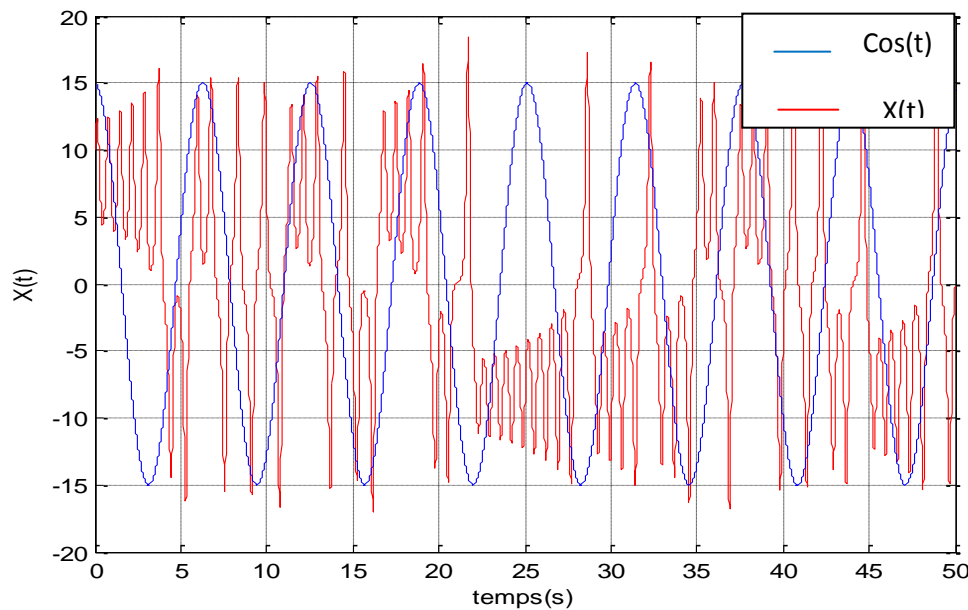
**D-Le déterminisme et l'imprédictibilité :**

# Chapitre I : Généralité sur les systèmes chaotiques

Pour un système déterministe, si l'on connaît l'état du système à un instant donné on est en mesure de prévoir son état à un instant ultérieur puisqu'il existe une seule solution pour chaque condition initiale. Cependant, deux conditions initiales très proches peuvent avoir des évolutions complètement différentes. L'évolution du système devient alors imprédictible, car une petite erreur de mesure ou un simple arrondi conduit à des résultats complètement faux au bout d'un certain temps. C'est le chaos déterministe à long terme, on ne peut pas savoir, même approximativement quelle sera la valeur d'un système chaotique.

## *E-L'aspect aléatoire :*

Il est difficile de distinguer un comportement chaotique d'un comportement aléatoire sur les trajectoires des solutions dans l'espace des temps. Donc un système chaotique a un aspect aléatoire.



**Figure (I.8) :** Etat  $x(t)$  du système de Lorenz comparé à un signal périodique.

$x(t)$  est l'état du système de Lorenz décrit par l'équation (I.1).

Notant que les systèmes chaotiques semblent évoluer de manière aléatoire, mais ils obéissent tout de même aux lois de la physique, si nous nous plaçons dans l'approximation de la physique classique.

## *F-Le spectre de puissance [21] :*

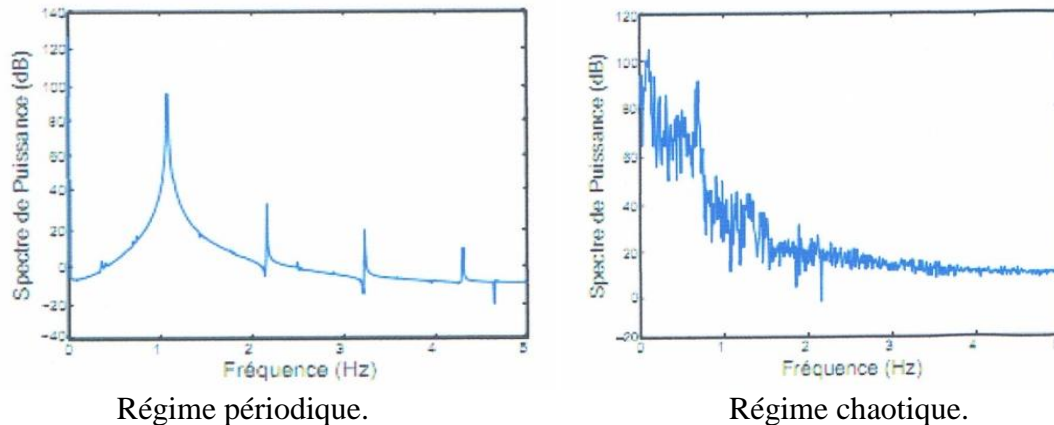
Une approche pratique et porteuse d'information est le calcul du spectre du signal, il s'agit de l'extraction des composantes fréquentielles d'un signal. Elle utilise la transformée de Fourier. La fréquence, l'amplitude et la phase de chaque composante sinusoïdale sont calculées en fonction du signal temporel selon la relation suivante en faisant intervenir les deux variables réciproques,  $t$  le temps, et  $f$  la fréquence :

$$\tilde{x}(f) = \int_{-\infty}^{+\infty} x(t) e^{-2j\pi f t} dt \quad (I.4)$$

La transformée de Fourier  $\tilde{x}(f)$  est complexe.

# Chapitre I : Généralité sur les systèmes chaotiques

Le module de cette grandeur représente le spectre d'amplitude ; l'argument le spectre de phase et le spectre de puissance représente le carré du module  $|\tilde{x}(f)|^2$ . Une façon simple de caractériser le chaos consiste à calculer le spectre de Fourier de l'évolution temporelle d'une des variables du système. Le spectre de Fourier d'un signal périodique ou quasi périodique est constitué de raies distinctes correspondant aux périodes et harmoniques du système. Pour un signal chaotique, on obtient un spectre continu.



*Figure(I.9) : Le spectre de puissance en régime périodique et en régime chaotique.*

## G-La fonction d'auto corrélation [21] :

La fonction de corrélation entre deux signaux est un outil mathématique permettant d'analyser le degré de ressemblance entre ces deux signaux, à une translation près.

Le spectre de puissance est la transformée de Fourier de la fonction d'autocorrélation notée

$$C(\tau) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} x(t)x(t + \tau)dt \quad (I.5)$$

La fonction d'autocorrélation est construite en faisant varier progressivement l'intervalle  $\tau$ , par conséquent si  $x(t)$  est constant, périodique ou quasi périodique  $C(\tau)$  reste non nul quand  $\tau \rightarrow \infty$  car le spectre de puissance est formé de raies distincts. En particulier pour les oscillations périodiques.

$C(\tau)$  oscille entre 1 et -1, dans le cas des oscillations chaotiques ou le spectre de puissance comporte une partie continue,  $C(\tau) \rightarrow 0$ , un processus chaotique est tel que sa fonction de corrélation décroît vers 0 quand l'horizon tend vers l'infini. Cette propriété assure que les solutions sont divergentes les unes des autres.

Si la fonction de corrélation est nulle pour des horizons non nuls alors c'est un processus non corrélé, et on parle de « bruit blanc déterministe »

## I.4- Les sections de Poincaré [15] :

Henri Poincaré a apporté une contribution très utile pour l'étude des systèmes chaotiques. Parmi ces contributions, on trouve les sections de Poincaré. Faire une section de Poincaré revient à couper la trajectoire dans l'espace des phases par un plan à la dimension 3 ou par une droite en dimension 2, afin d'étudier les intersections de cette trajectoire avec ce plan ou cette droite, ces intersections sont faites avec un sens prédéfini.

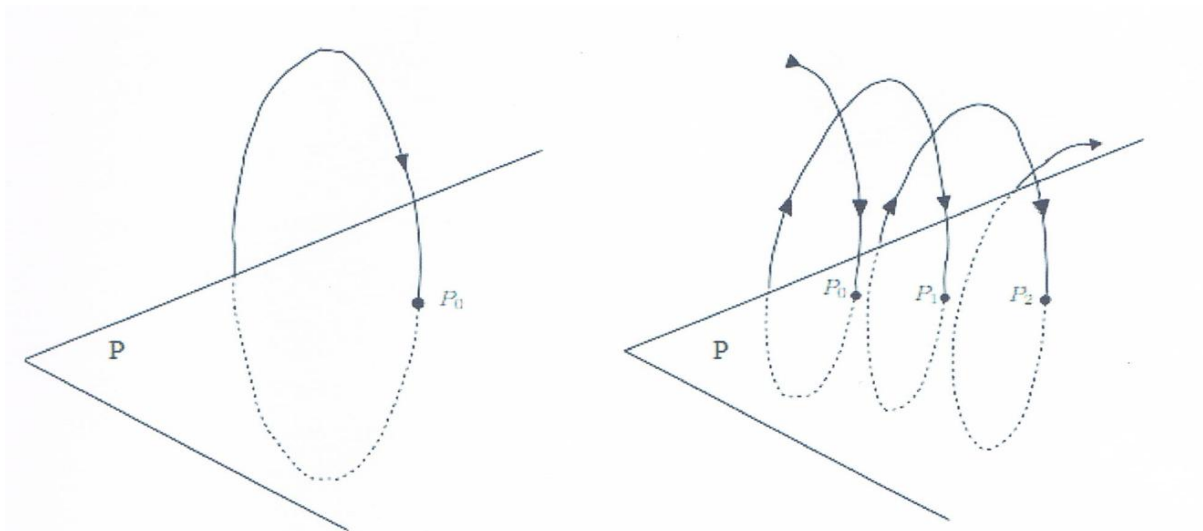
# Chapitre I : Généralité sur les systèmes chaotiques

On passe alors d'un système dynamique à temps continu à un système dynamique à temps discret via une réduction d'une unité de l'ordre du système. Les mathématiciens ont bien sûr démontré que les propriétés du système sont conservées après la réalisation d'une section de Poincaré judicieusement choisie.

On constate qu'à chaque type d'attracteur correspond un seul type de topologie dans le plan.

Le plan de la section doit être choisi de manière à garantir l'existence d'intersections avec la trajectoire et de telle sorte que celle-ci le traverse alternativement dans un sens puis dans l'autre.

La figure suivante montre le principe de la section de Poincaré, pour une solution périodique et une solution non périodique.



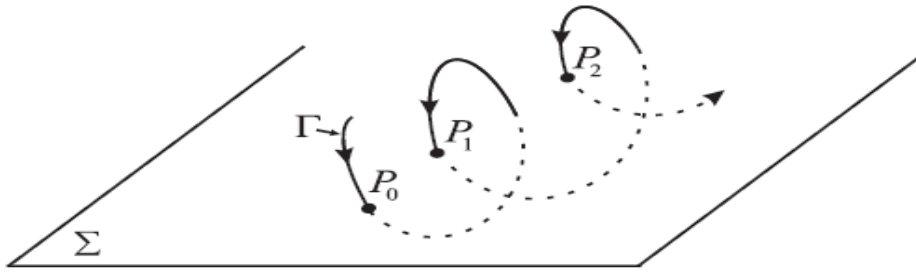
Section de Poincaré d'une solution périodique.

Section de Poincaré d'une solution non périodique.

**Figure (I.10) : le Principe de la section de Poincaré.**

En effectuant une section de Poincaré, on peut identifier le régime de fonctionnement d'un système dynamique en observant l'allure obtenue sur cette section. Par exemple, lorsque le régime est périodique, la section de Poincaré est un point (l'attracteur est un cycle limite), pour un régime bipériodique la section de Poincaré est une courbe fermée (l'attracteur est un tore) et lorsque le régime est chaotique les points de la section sont répartis sur une surface.

Exemple :



**Figure (I.11) : Section de Poincaré.**

Avec

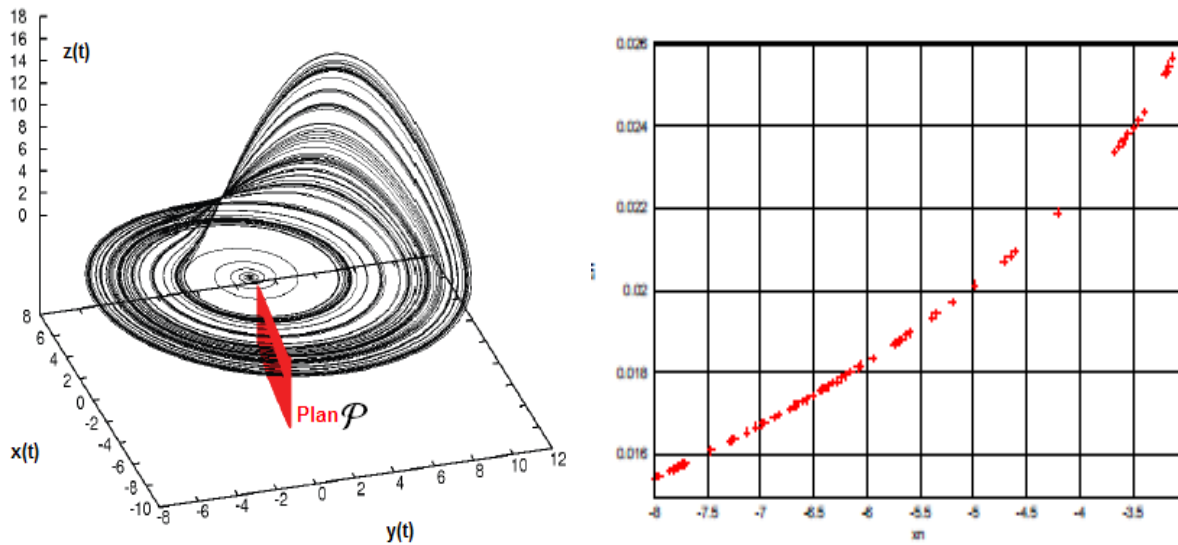
$\Gamma$  : la trajectoire

$\Sigma$  : le plan

La trajectoire  $\Gamma$  coupe le plan  $\Sigma$  en des points  $P_n$ . Ces points appartiennent à la section de Poincaré.

❖ **Exemples de sections de Poincaré :**

La section de Poincaré la plus naïve est de couper la trajectoire dans l'espace des phases par un plan (en dimension trois) ou par une droite (en dimension deux).



**Figure (I.12): Intersections de la trajectoire de l'attracteur de Rössler avec un plan  $p$  d'équation  $y = 0$  ( $x \leq 0$ ).**

La section de Poincaré obtenue par l'intersection de l'attracteur de Rössler en trois dimensions avec un plan est un arc assimilable à une parabole.

**1.5- La Bifurcation [21] :**

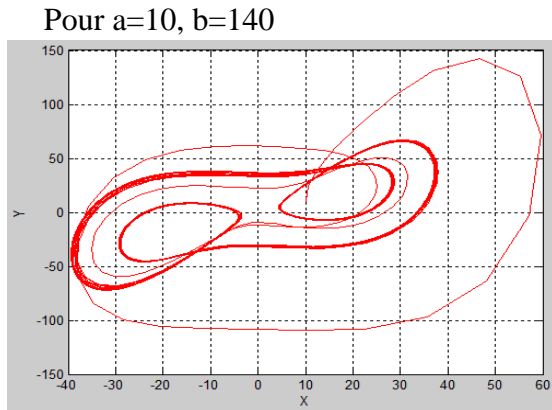
Dans les systèmes linéaires, les modifications des paramètres entraînent des changements quantitatifs, mais ne modifient pas le comportement (état stationnaire) du système. Dans les

# Chapitre I : Généralité sur les systèmes chaotiques

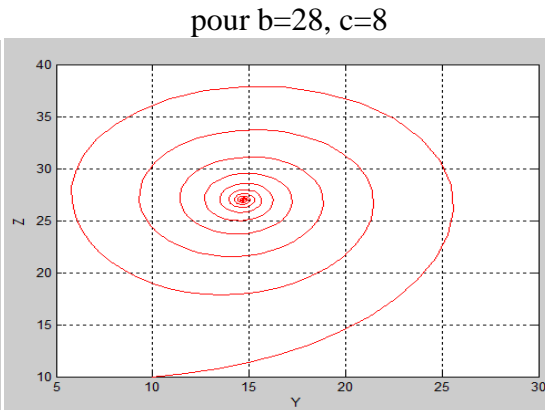
systemes dynamiques non linéaires et dans certains cas une petite variation de certains paramètres de contrôle, peut dans des conditions bien définies, au voisinage d'une valeur critique, provoquer un changement complet de comportement à l'équilibre du système. C'est ce qu'on appelle *une bifurcation*.

**Exemple :**

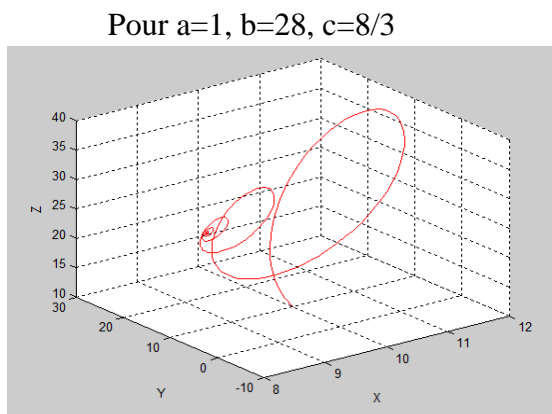
Le modèle de Lorenz représente des comportements différents suivant la valeur des paramètres  $a$ ,  $b$  et  $c$  comme le montre les figures suivantes.



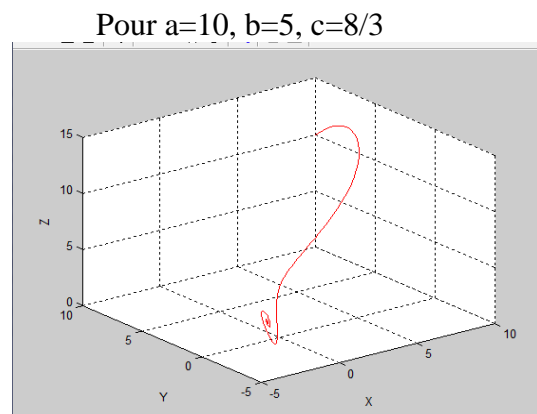
Cycle limite



cycle limite



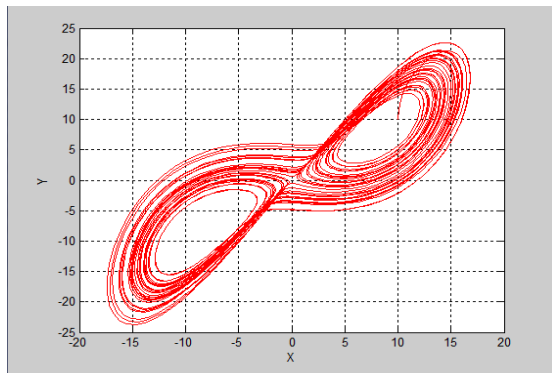
Point fixe



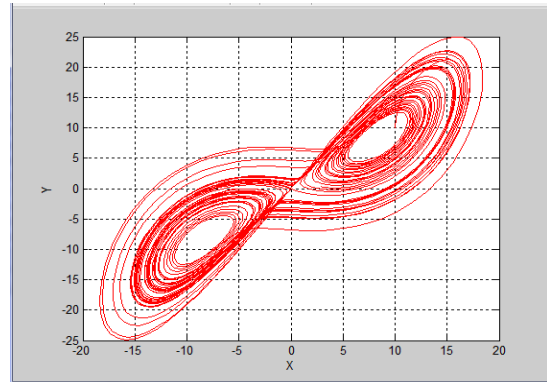
Point fixe

Pour  $a=9, b=28$

Pour  $a=10, b=20$



Attracteur chaotique



Attracteur chaotique

**Figure (I.13):** *Attracteurs du modèle de Lorenz pour différentes valeurs de ces Paramètres.*

❖ **Observation :**

On remarque pour le même système différents types d'attracteurs sont obtenus, par le changement des paramètres du système.

❖ **Étude de diagramme de bifurcation en utilisant une suite logistique :**

Feigenbaum a redécouvert une route vers le chaos qui avait été découverte dans les années 60 par Myrberg. Aujourd'hui, cette route est appelée « cascade de dédoublements de période » pour décrire la transition entre un comportement périodique et un attracteur chaotique. Ce scénario est observé par exemple avec la suite logistique.

En mathématiques, une suite logistique est un exemple simple de suite dont la récurrence n'est pas linéaire. Souvent citée comme exemple de la complexité pouvant surgir de simple relation non linéaire, cette suite fut popularisée par le biologiste Robert May en 1976.

Sa relation de récurrence est :

$$x_{n+1} = kx_n(1 - x_n) \quad (I.6)$$

Où  $n=0, 1, 2, \dots$ , dénote le temps discret,  $x$  l'unique variable dynamique et  $0 < k < 4$  un paramètre de contrôle.

Elle conduit, suivant les valeurs de  $k$ , à une suite convergente, une soumise à oscillations ou une suite chaotique.

**Exemple :**

L'exemple suivant représente le dédoublement de période du modèle de Rössler décrit par l'équation (I.2), par variation du paramètre  $c$ .

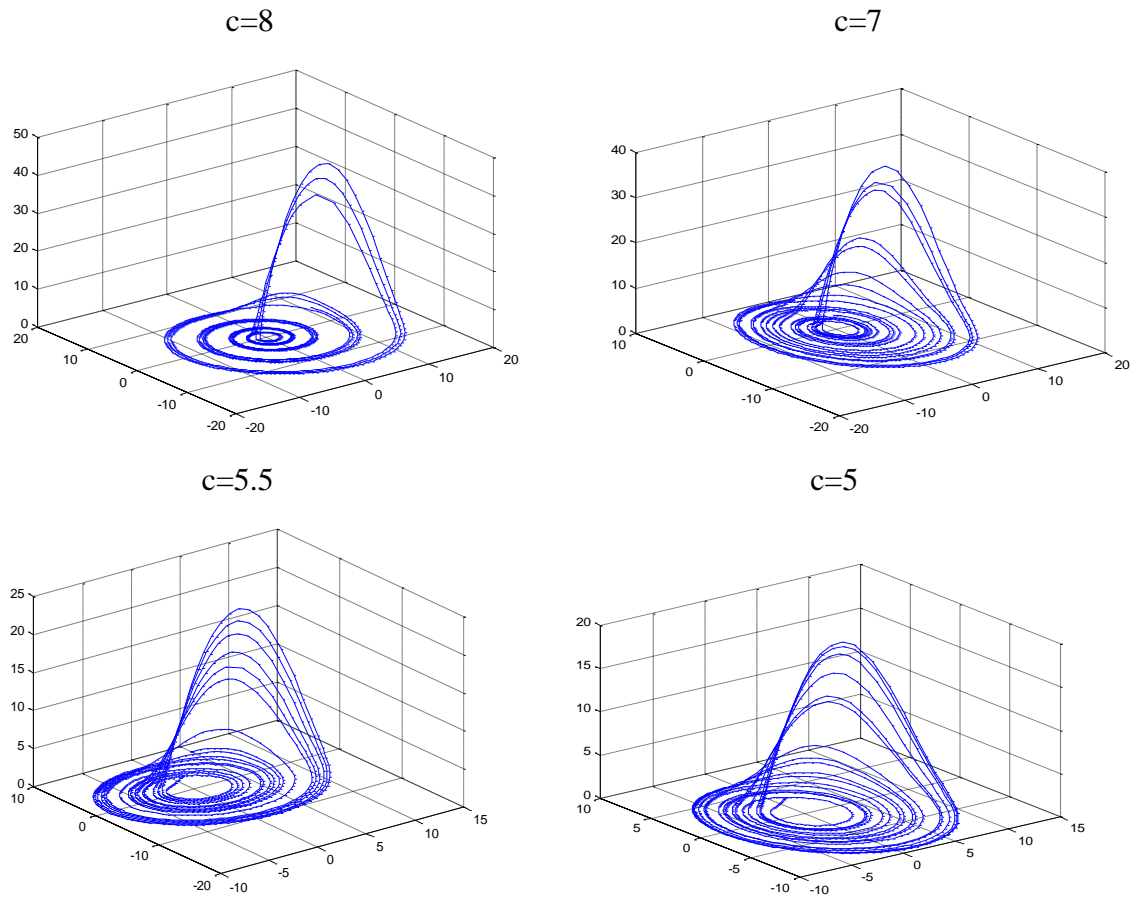


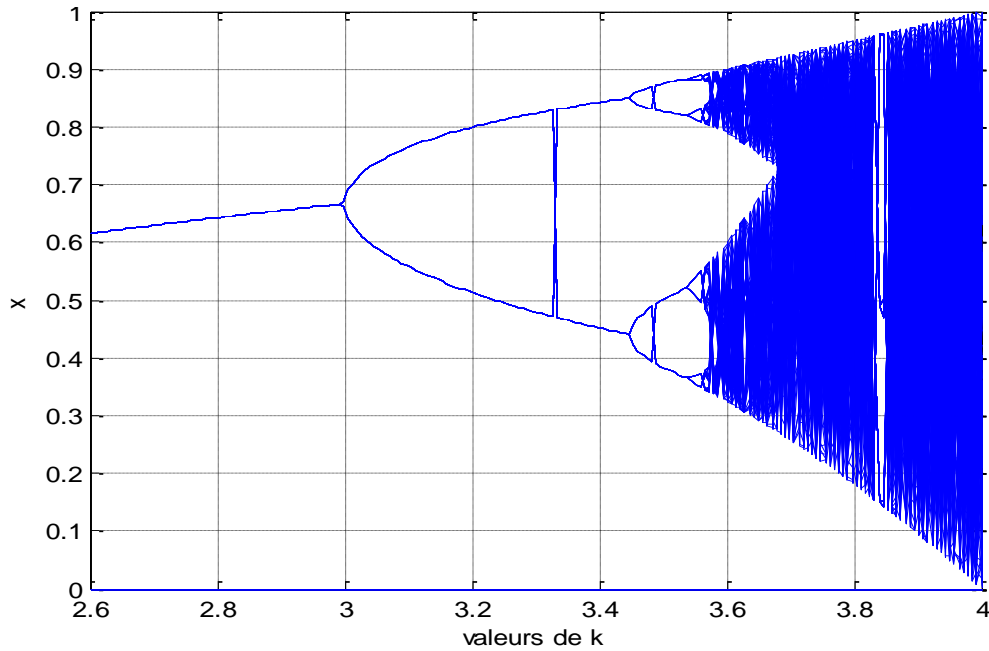
Figure (14) : dédoublement de période de l'attracteur du modèle de Rössler.

## ❖ Diagramme de Bifurcation :

Le diagramme de Bifurcation est un tracé repérant la nature des différentes solutions du système et leur stabilité lorsqu'un paramètre varie.

Il est composé d'intervalles sur lesquelles les solutions asymptotiques (ou les ensembles limites qui leur correspondent) évoluent continûment avec le paramètre. Il classe donc les valeurs du paramètre sur l'axe d'abscisse et des valeurs particulières d'une des variables d'état sur l'axe des ordonnées. C'est un outil efficace pour évaluer rapidement l'ensemble des solutions possibles d'un système en fonction des variations de l'un de ses paramètres. Il permet de repérer les valeurs particulières du paramètre qui induisent des bifurcations.

La figure suivante illustre le diagramme de bifurcation.



*Figure (I.12) : Le Diagramme de Bifurcation.*

## ***I.6- Conclusion :***

Dans ce chapitre, nous avons vu que le chaos est obtenu à partir de systèmes non linéaires; il correspond à un comportement borné, de ces systèmes, ce qui le fait apparaître comme du bruit pseudo aléatoire. Un système est chaotique s'il est régi par des lois déterministes, mais que son évolution échappe à toute prévision à long terme; la caractéristique principale du chaos déterministe est la dépendance aux conditions initiales qui est l'origine de la théorie du chaos, la représentation des systèmes chaotique dans l'espace de phases est une figure géométrique de structure infiniment complexe qui s'appelle l'attracteur étrange. Un chaotique est un système déterministe à aspect aléatoire qui a au moins un exposant de Lyapunov positif, les signaux chaotiques ont des fonctions d'auto corrélation très étroites et des spectres de puissance à large bande proche du bruit blanc.

Dans ce chapitre, nous avons vu aussi la section de Poincaré ainsi que la notion de bifurcation avec quelques exemples pour chaque point.

# Chapitre II

# Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

---

## **II.1- Introduction :**

L'idée d'utiliser les signaux aléatoires pour la communication sécurisée a été mise en œuvre en 1926 par Vernam. Il proposa dans son article d'utiliser un alphabet binaire et coder chaque bit à l'aide d'un bit de la clé, choisi de façon arbitraire. Plus tard, dans les années 90, cette idée a été développée dans le contexte des signaux chaotiques à cause de la nature imprédictible à long terme du chaos. On a cru pendant longtemps que le chaos serait inutilisable et incontrôlable, mais depuis quelques années les chercheurs ont réussi à modéliser le chaos par des équations différentielles et montré qu'il existe un côté déterministe dans ce phénomène qui apparaît aléatoire à première vue. C'est cette nature ressemble aux signaux aléatoires qui a motivé les chercheurs d'essayer de camoufler un message confidentiel à l'aide d'un signal chaotique de façon à ne pas pouvoir le distinguer. Différentes méthodes ont été proposées afin de masquer le message dans un système chaotique et ensuite le restaurer. Ces méthodes sont toutes basées sur la synchronisation des systèmes chaotiques et ont été améliorées au fil des années dans le but d'augmenter de plus en plus la sécurité et la rapidité de transmission d'information. Ces méthodes sont parfois appelées méthodes de cryptographie chaotique.

Dans les systèmes de communication, la synchronisation est une clé très importante pour une transmission réussie. La synchronisation classique employée dans les systèmes de télécommunication cherche à reproduire juste le signal périodique de la porteuse. Par contre, la synchronisation chaotique au niveau du récepteur cherche à dupliquer le signal chaotique envoyé de l'émetteur. Cela veut dire que deux signaux chaotiques seront dits synchronisés s'ils sont asymptotiquement identiques lorsque le temps  $t$  tend vers l'infini. Dans le premier chapitre, nous avons déjà vu la sensibilité du chaos aux conditions initiales, il apparaît alors que la synchronisation chaotique n'est pas si simple à établir et pose plus de contraintes qu'une synchronisation classique.

Un signal chaotique se présente sous la forme d'un bruit blanc que ce soit dans le domaine temporel ou le domaine fréquentiel. Ce qui différencie un signal chaotique d'un bruit aléatoire est la notion de déterminisme. En effet, le bruit ne peut être décrit que comme un processus aléatoire alors qu'un système chaotique est représentable par des équations différentielles, ce qui explique qu'il sera possible de synchroniser deux systèmes chaotiques.

Ce chapitre est organisé de la manière suivante ; dans la première partie du chapitre nous présentons le concept de la synchronisation du chaos, dans la deuxième partie nous exposerons les principales implémentations utilisant des systèmes chaotiques pour les communications.

## **II.2. La synchronisation :**

### **II.2.1- Définition :**

La synchronisation est un phénomène qui caractérise de nombreux systèmes non linéaires. La synchronisation de deux systèmes dynamiques signifie que chaque système évolue en suivant le comportement de l'autre système.

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

Considérons les deux systèmes suivants :

$$\begin{cases} \dot{x} = f_1(x, u) \\ \dot{y} = f_2(x, u) \end{cases} \quad (\text{II.1})$$

Avec  $x \in \mathbb{R}^n$ ;  $y \in \mathbb{R}^n$  et  $f_i : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  sont des champs de vecteurs non linéaires.

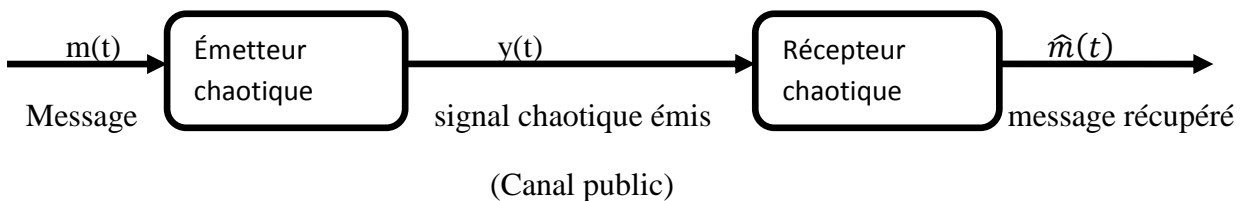
Les deux systèmes sont dits synchronisés si :

$$e = |x(t) - y(t)| = 0 \text{ quand } t \rightarrow \infty \quad (\text{II.2})$$

Avec  $e$  : l'erreur de synchronisation [14].

### II.2.2- Système de communication chaotique :

Soit la figure suivante :



**Figure (II.1) : système de communication chaotique.**

Le signal  $m(t)$  contient l'information à transmettre qui est injectée à l'entrée de l'émetteur chaotique, celui-ci génère un signal chaotique  $y(t)$  qui est transmis au récepteur, ce dernier est piloté par le signal  $y(t)$  et réalise l'opération inverse de celle effectuée à l'émission afin de récupérer le message transmis. Pour transmettre le message par l'émetteur, on mélange ce message avec un signal chaotique. De ce fait, il apparaît difficile pour le récepteur de récupérer l'information contenue dans le signal chaotique.

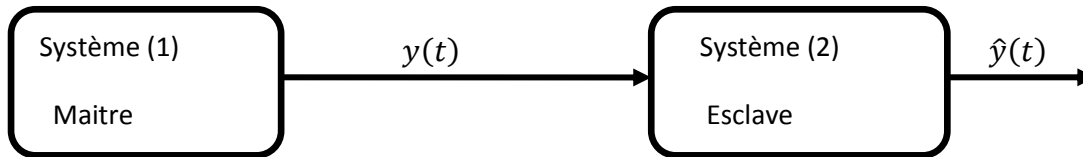
En général le récepteur est similaire à l'émetteur et peut présenter un comportement chaotique lorsqu'il n'est pas piloté par le signal  $y(t)$ . Pour pouvoir extraire l'information du signal reçu, il est nécessaire de synchroniser les deux systèmes.

Deux systèmes sont synchronisés si leur période ainsi que leur phase sont identiques, mais cela n'est pas valable pour les systèmes chaotiques. Alors, deux signaux chaotiques sont dits synchronisés s'ils sont asymptotiquement identiques.

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

### ❖ *Système maître esclave pour la communication chaotique :*

Soit la figure suivante :



**Figure (II.2) : système maître esclave pour réaliser la synchronisation.**

Le système (1) est dit maître et le système (2) est dit esclave. Le système esclave sera synchronisé avec le système maître si :

$$\lim_{t \rightarrow \infty} |\hat{y}(t) - y(t)| = 0 \quad (\text{II.3})$$

Il paraît impossible d'arriver à synchroniser deux exemplaires d'un même système chaotique, parce que dans les systèmes réels il est extrêmement difficile de construire deux circuits à l'identique à cause de la tolérance sur les composants ainsi que du bruit présent dans tous les systèmes électroniques.

Si on suppose que l'on dispose de deux circuits identiques, il se pose le problème de la sensibilité aux conditions initiales qui se traduit par une instabilité au sens de Lyapunov. Une infime différence entre les conditions initiales des deux circuits conduira à des signaux totalement différents. Cela signifie qu'il est impossible de reproduire ces conditions initiales dans un système réel.

### **II.2.3- Méthode de synchronisation [14] :**

Supposons deux systèmes chaotiques identiques oscillent de façon totalement indépendante. Si par un moyen quelconque, on leur permet d'échanger de l'énergie, action qui l'on nomme « couplage », les deux systèmes finiront par céder la place à un comportement commun donc les deux systèmes se synchronisent. Il est possible de coupler les systèmes chaotiques dans un sens c'est le couplage unidirectionnel ou dans l'autre sens «le couplage bidirectionnel », en plus de couplage unidirectionnel et bidirectionnel d'autres méthodes ont être proposées pour la synchronisation. On peut citer la méthode par décomposition du système, la synchronisation par la boucle fermée, la synchronisation impulsive et la synchronisation à l'aide d'observateur. Dans la majorité des cas, les deux systèmes doivent avoir des structures identiques, ce qui n'est pas tout à fait réalisable en pratique.

Un petit écart entre les valeurs des composants peut entraîner un écart considérable entre les comportements des deux circuits et détruire le phénomène de synchronisation.

#### **A- Synchronisation par la boucle fermée [10] :**

Une autre technologie de la synchronisation est basée sur la boucle fermée qui peut être illustrée par la figure (II.3) où nous employons l'erreur entre l'émetteur et le récepteur pour corriger le comportement du récepteur afin de réaliser la synchronisation.

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

---

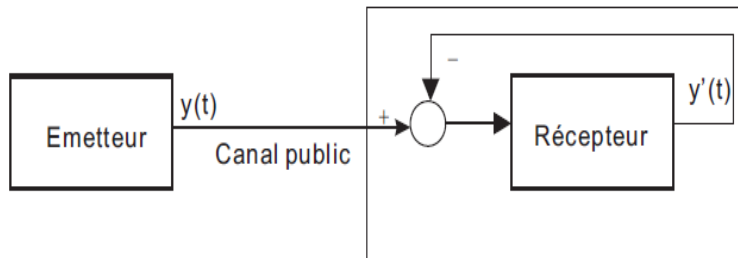
Supposons que l'émetteur s'écrit comme suit :

$$\begin{cases} \dot{x} = f(x) \\ y = h(x) \end{cases} \quad (\text{II.4})$$

et que le récepteur peut être décrit comme suit :

$$\begin{cases} \dot{x}' = f(x') + g(y - y') \\ y' = h(x') \end{cases} \quad (\text{II.5})$$

Avec  $g$  est une fonction de l'erreur entre  $y$  et  $y'$  et que cette fonction est choisie afin de garantir la synchronisation entre l'émetteur et le récepteur. En fait, ce type de récepteur peut être considéré comme la conception d'un observateur.



**Figure(II.3) Synchronisation par un contrôle en boucle fermée.**

### **B- Synchronisation par l'inversion du système [10] :**

Jusqu'à présent, toutes les approches mentionnées ont pour but de synchroniser seulement les états du système, et elles ne concernent pas la synchronisation (ou plus exactement l'estimation) des entrées inconnues du système. Cependant, la possibilité d'estimer les entrées inconnues est évidemment essentielle à la transmission chaotique de données puisque l'entrée inconnue est généralement le message confidentiel.

Soit la figure (II.4), ou l'émetteur peut être écrit de la façon suivante :

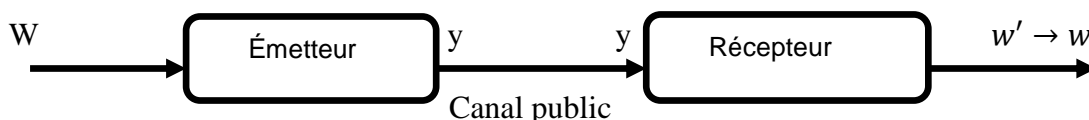
$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases} \quad (\text{II.6})$$

$x \in \mathbb{R}^n$  est le vecteur des états du système,  $u \in \mathbb{R}^m$  est le vecteur des entrées inconnues.

$f: \mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^m$ ,  $h: \mathbb{R}^n \rightarrow \mathbb{R}^p$  sont des vecteurs des fonctions analytiques.

Pour le récepteur, son vecteur d'entrée est le vecteur de sortie de l'émetteur.

Nous essayons de concevoir un récepteur tel que son vecteur de sortie convergera au moins asymptotiquement vers le vecteur d'entrée de l'émetteur. Ce problème s'appelle l'inversion à gauche du système. Il est à noter que l'inversion du système exige des conditions additionnelles par rapport au problème d'observabilité, telle que le degré relatif.



**Figure (II.4): Synchronisation par l'inversion du système.**

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

### C- Synchronisation identique [10] :

Pour illustrer la méthode de synchronisation par couplage entre deux systèmes chaotiques, on a choisi de présenter la synchronisation identique proposée par Picora et Carroll. Celle-ci a l'avantage de représenter une solution simple et performante de synchronisation dont l'objectif est que l'esclave reproduise le plus fidèlement possible l'état du maître, après un régime transitoire.

Considérons un système dynamique autonome, en temps continu, de dimension  $n$ , représenté par la relation suivante :

$$\dot{x}(t) = f(x(t)) \quad (\text{II.7})$$

Si on divise le système initial en deux sous-systèmes sys(1) et sys(2) alors on obtient :

$$\begin{cases} \text{Sys(1)} : \dot{x}_1(t) = f_1(x_1(t), x_2(t)) \\ \text{Sys(2)} : \dot{x}_2(t) = f_2(x_1(t), x_2(t)) \end{cases} \quad (\text{II.8})$$

Où :  $x = [x_1, \dots, x_n]^T$ , les états et les dynamiques définis conformément aux relations suivantes :  $x = [x_1, x_2]^T$ ,  $x_1 = [x_1, \dots, x_m]^T$ ,  $x_2 = [x_{m+1}, \dots, x_n]^T$ ;  $f(x) = [f_1(x), f_2(x)]$ .

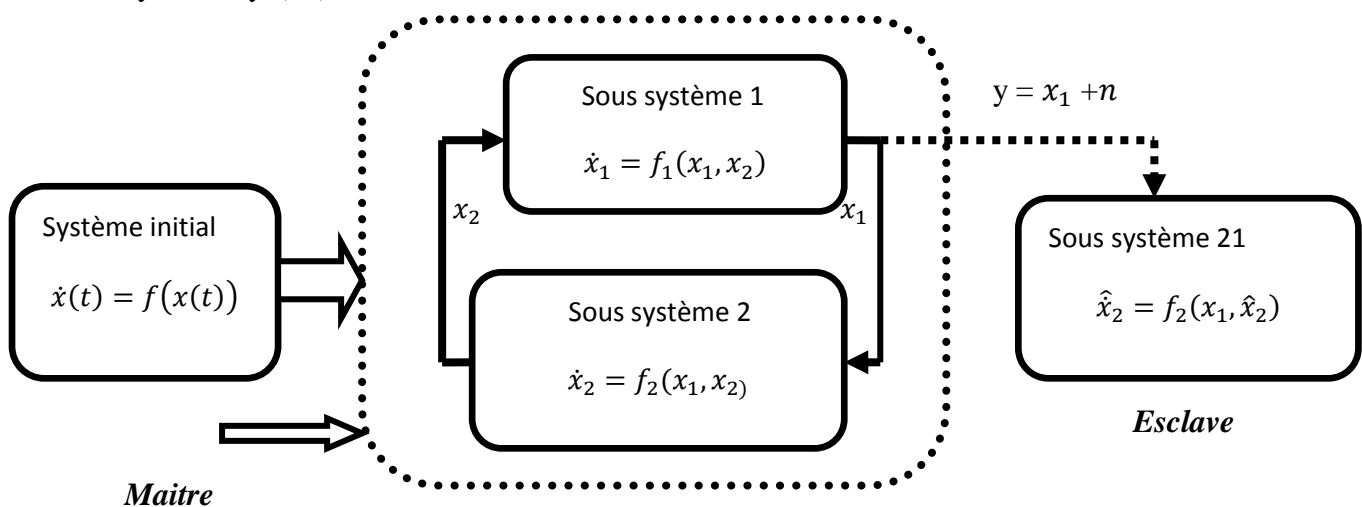
Cette opération peut être réalisée de manière arbitraire avec une réorganisation des variables d'état dans un ordre quelconque. On considère un deuxième sous système syst(21) caractérisée par une dynamique identique à  $f_2$ , et un vecteur d'état  $\hat{x}_2(t)$ .

$$\text{Sys(21)} : \dot{\hat{x}}_2(t) = f_2(x_1(t), \hat{x}_2(t)) \dots \dots \dots (\text{II.9})$$

On peut dire que ce sous-système sys(21) est un candidat susceptible de se synchroniser avec la dynamique complète initiale. Picora et Carroll ont démontré que la condition nécessaire et suffisante pour que cette proposition soit vraie est que le sous-système soit stable, hypothèse qui est équivalent avec la condition que l'ensemble des exposants de Lyapunov de sous système soient négatif, alors une synchronisation parfaite peut alors être accomplie, les trajectoires étant asymptotiquement convergentes :

$$\lim_{t \rightarrow \infty} |\hat{x}_2(t) - x_2(t)| = 0 \quad (\text{II.10})$$

Dans la figure (II.5) , on présente graphiquement le processus de décomposition en sous-systèmes , cette fois avec la notation  $y = x_1 + n$  de la variable d'état qui commande le système sys(21) avec  $n$  est un éventuel bruit associé au canal de communication.



**Figure(II.5) : Structure de synchronisation par décomposition en sous-système Proposée par Pocora et Carroll.**

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

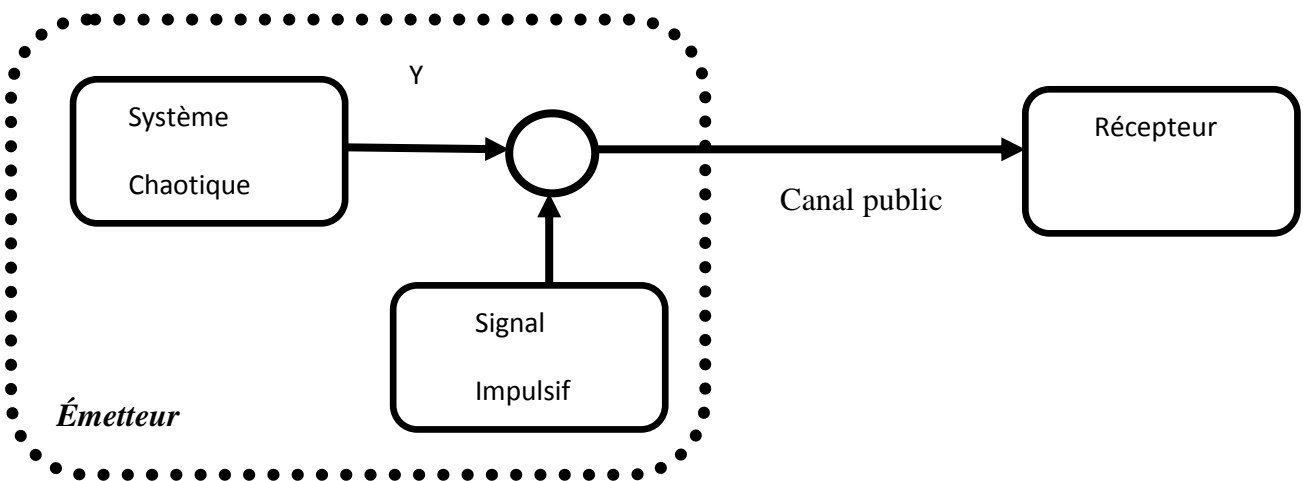
### D- Synchronisation par couplage :

Il est possible de coupler les systèmes chaotiques par le couplage unidirectionnel, ou par le couplage bidirectionnel. Dans le premier cas, l'énergie est transférée d'un système à l'autre, à l'aide d'un élément de couplage fonctionnel, dans un seul sens.

Dans le couplage bidirectionnel, l'élément de couplage permet d'échanger de l'énergie dans les deux sens. Ces deux types de couplages peuvent aussi être appliqués aux systèmes non identiques.

### E- Synchronisation impulsive [10] :

Dans un schéma de transmission usuel, un des états du système dynamique est transmis afin de réaliser la synchronisation par le récepteur. Dans le but de réduire la redondance du signal transmis, pour cela le signal de transmission est divisé en petits intervalles (impulsions). La synchronisation est analogue à la synchronisation échantillonnée.



Figure(II.6) : synchronisation impulsive.

### F- Synchronisation à l'aide d'observateur [12] :

La synchronisation peut également être réalisée en employant un observateur.

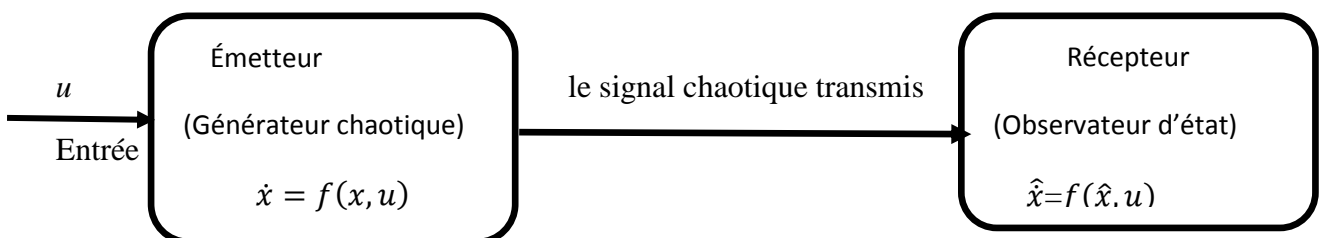


Figure (II.7) : principe de la synchronisation à base d'observateurs.

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

Le système maître est un système chaotique quelconque et le système esclave est un observateur d'état correspondant.

### ❖ Définition :

De manière générale, un observateur est un système dynamique qui permet la reconstruction de l'état d'un système dynamique, à partir de ses entrées, de ses sorties, et de la connaissance de son modèle, qui sont les seules informations disponibles. C'est une méthode typique afin d'estimer les états inconnus d'un système dynamique qui ne peuvent pas être mesurés directement : soit par ce qu'ils sont inaccessible, soit par économie.

La figure suivante représente le principe d'observateur.

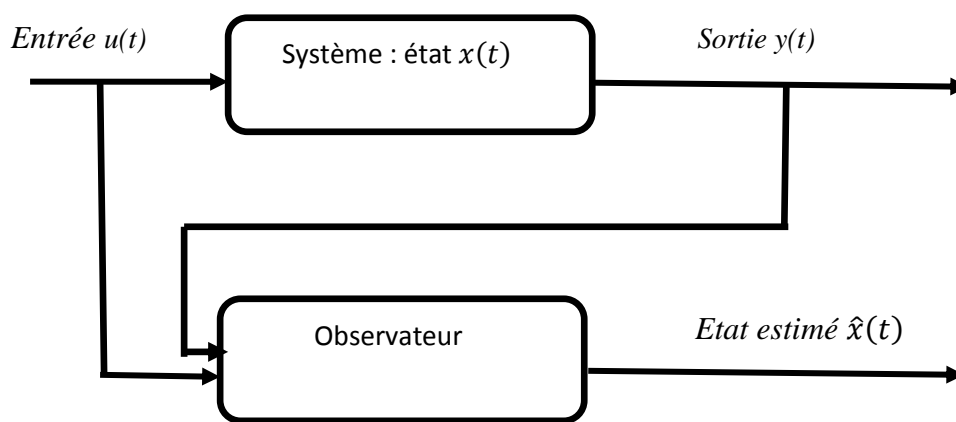


Figure (II.8) : Principe de l'observateur.

Le problème de la conception d'un système (non linéaire) donnée est posé comme suit.

$$|x(t) - \hat{x}(t)| \rightarrow 0 \text{ quand } t \rightarrow \infty \quad (\text{II.11})$$

Où  $x(t)$  est l'état du système et  $\hat{x}(t)$  l'état estimé.

### a- Observabilité des systèmes :

#### ❖ Observabilité d'un système linéaire

Considérons le système linéaire défini par les équations suivantes :

$$\begin{cases} \dot{x}(t) = Ax \\ y(t) = Cx \end{cases} \quad (\text{II.12})$$

Où  $x \in \mathbb{R}^n$  et  $y \in \mathbb{R}^p$  et  $A, C$  sont des matrices de dimensions appropriées. Le système est observable, c'est-à-dire qu'on peut retrouver les états d'origine du système à partir de ses sorties, lorsque :

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

---

$$\text{rang}(0) = \text{rang} \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} = n \quad (\text{II.13})$$

❖ **Observabilité d'un système non linéaire :**

Considérons maintenant le système non linéaire :

$$\begin{cases} \dot{x} = f(x) \\ y = h(x) \end{cases} \quad (\text{II.14})$$

$x \in \mathbb{R}^n$  est le vecteur d'état,  $y \in \mathbb{R}^p$  est le vecteur de sorties, les fonctions  $f, h$  sont des vecteurs de fonctions analytiques de dimensions appropriées.

❖ **Condition du rang d'observabilité :**

Pour donner la condition du rang d'observabilité, il faut d'abord définir la dérivée de Lie, qui est une notion largement utilisée dans l'étude d'observabilité des systèmes non linéaire.

❖ **Définition de la dérivée de Lie :**

Considérons  $h$  une fonction de  $\mathbb{R}^n$  dans  $\mathbb{R}$ . On définit la dérivée de Lie de  $h$  dans la direction de  $f$ , notée  $L_f h$ , la dérivée de  $h$  le long de la courbe intégrale de  $f$  en  $t = 0$ .

$$L_f h(x) = \sum_{i=1}^n f(x) \frac{\partial h}{\partial x_i}(x) \quad (\text{II.15})$$

Par définition on décrit :  $L_f^0 h = h$  et  $L_f^k h = L_f(L_f^{k-1} h), \forall k \geq 1$ .

Le système (II.15) satisfait la condition du rang d'observabilité si :

$$\text{rang} \begin{pmatrix} dh \\ dL_f h \\ \vdots \\ dL_f^{n-1} h \\ \vdots \\ dL_f^\infty h \end{pmatrix} = n \quad (\text{II.16})$$

Si la condition du rang d'observabilité est vérifiée en  $x, \forall x \in U, U$  étant un voisinage de  $x^0$ , le système (II.14) est localement observable en  $x^0$

Pour l'application non linéaire, nous définissons la condition du rang d'observabilité 'pratique' comme suit :

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

---

$$\text{rang} \begin{pmatrix} dh \\ dL_f h \\ \vdots \\ dL_f^{n-1} h \end{pmatrix} = n \quad (\text{II.17})$$

### **b- Observabilité de Luenberger :**

La théorie de l'observation de Luenberger repose essentiellement sur des techniques de placement de pôles.

Soit le système linéaire suivant :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \quad (\text{II.18})$$

Où  $x(t) \in \mathbb{R}^n$ ,  $u(t) \in \mathbb{R}^m$ ,  $y(t) \in \mathbb{R}^p$ , les matrices du système sont de dimensions appropriées, et les conditions initiales sont définies par  $x(t_0) = x_0$ .

Luenberger propose l'observateur suivant pour le système suivant :

$$\begin{cases} \dot{\hat{x}} = A\hat{x}(t) + Bu(t) + k(y(t) - C\hat{x}(t)) \\ \hat{y} = C\hat{x}(t) \end{cases} \quad (\text{II.19})$$

Où  $\hat{x}(t)$  est l'estimé de  $x(t)$ ,  $\hat{y}(t)$  est l'estimé de  $y(t)$ .

La dynamique de l'erreur d'estimation  $e(t) = x(t) - \hat{x}(t)$  a pour expression :

$$\dot{e}(t) = (A - kC)e(t) \quad (\text{II.20})$$

En utilisant une technique de placement de pôles, il suffit alors de choisir le gain  $K$  de l'observateur de telle sorte que les valeurs propres de la matrice  $A - KC$  soient dans le demi-plan complexe gauche.

### **c- Observabilité à grand gain :**

Les techniques dite grand gain peuvent être appliquées sans transformation du système initiale : dans ce cas, la conception de l'observateur se fait directement à partir de la structure du système. Cette technique utilise la théorie de stabilité de Lyapunov pour adapter les techniques développées dans le cas linéaire. La méthode présentée dans [25] donne des conditions suffisantes de convergence de l'état estimé vers l'état réel du système, pour la classe des systèmes non linéaires décrits par le modèle suivant :

$$\begin{cases} \dot{x} = f(x) = Ax + f(x(t), u(t)) \\ y = h(x) = Cx \end{cases} \quad (\text{II.21})$$

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

---

La dynamique de l'état comporte une partie linéaire non commandée et une partie non linéaire commandée. Vérifiant en général la condition de Lipschitz par rapport à  $x$

$$\|f(x_1) - f(x_2)\| \leq k\|x_1 - x_2\| \quad (\text{II.22})$$

L'observateur à grand gain possède la structure suivante :

$$\dot{\hat{x}}(t) = A\hat{x}(t) + f(\hat{x}(t), u(t)) + k(y(t) - C\hat{x}(t)) \quad (\text{II.23})$$

L'appellation grande gain provient de la structure de l'observateur :

Lorsque la fonction non linéaire possède une grande constante de Lipschitz, la moindre erreur entre l'état réel et l'état estimé va se répercuter et croître. Par conséquent, le gain  $k$  de l'observateur (II.23) doit être important pour compenser cette amplification de l'erreur.

La dynamique de l'erreur d'estimation  $e(t) = x(t) - \hat{x}(t)$  se réduit de (II.21) et (II.23)

$$\dot{e}(t) = (A - kC)e(t) + f(x(t), u(t)) - f(\hat{x}(t), u(t)) \quad (\text{II.24})$$

Il a été démontré dans [26] que si le gain  $k$  est choisi telle que :

$$k < \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} \quad (\text{II.25})$$

Où  $k$  est la constante de Lipschitz de  $f(x(t), u(t))$ ,  $\lambda_{\min}(Q)$  et  $\lambda_{\max}(P)$  sont des valeurs propre maximales et minimales .  $Q$  est une matrice symétrique définie positive et  $P$  est une matrice définie positive, solution de l'équation de Lyapunov :

$$(A - kC)^T P + (A - kC) = -Q \quad (\text{II.26})$$

Alors (II.23) est un observateur asymptotique du système non linéaire (II.21).

La synthèse de l'observateur consiste à ajouter les matrices  $P$ ,  $Q$  et  $k$  de façon à garantir l'égalité (II.26). Ensuite, on vérifie si la condition (II.25) est satisfaite. Ainsi, et malgré l'intérêt du résultat, cette méthode n'est pas constructive. Elle ne donne aucune indication sur le choix d'un gain satisfaisant la condition (II.25). Cette approche a été étendue par plusieurs auteurs.

Pour simplifier le problème on peut remplacer la matrice  $Q$  par une matrice identité, et ce la revient à choisir un gain  $k$  qui satisfait :  $k < \frac{1}{\lambda_{\max}(P)}$  (II.27)

Ces techniques dites à grand gain sont très répandues dans le littérature. Il s'agit principalement de techniques de vérification, qui permettent d'établir des conditions suffisantes de convergence de l'état estimé vers l'état réel. La structure de l'observateur non linéaire est une structure de Luenberger étendue au cas non linéaire.

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

---

### *d- Commande par retour d'état :*

Pour stabiliser l'écart  $\dot{e}(t)$  Nous utilisons dans ce travail un retour d'état.

#### ❖ *Définition de la commandabilité:*

La commandabilité d'un système caractérise sa capacité à voir son comportement dynamique évoluer sous l'action du signal de commande  $u(t)$ . Un système linéaire est commandable s'il est possible de le conduire, en appliquant un signal d'entrée admissible  $u(t)$ .

Soit le système linéaire décrit par les équations (II.28) :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \quad (\text{II.28})$$

La matrice de contrôlabilité  $M$  est :

$$M = [B \ AB \ A^2B \ \dots \ A^{n-1}B]$$

Le système est commandable si le rang de la matrice de commandabilité est égal à la dimension du système,  $\text{rang}(M) = n$ .

#### ❖ *Retour d'état :*

L'objectif est de stabiliser le système (s'il n'est pas stable) ou de modifier son comportement dynamique en introduisant une contre-réaction déterminée à partir du vecteur d'état  $x(t)$ . En supposant que l'on a accès à toutes les variables d'état (variables mesurées). En désignant par  $v(t)$  le signal de référence (consigne), la loi de commande s'exprime de la manière suivante :

$$u(t) = -kx(t) + v(t) \quad (\text{II.29})$$

Compte tenu de l'expression de la commande  $u(t)$  et du modèle du système en boucle ouverte, celui en boucle fermée est donné par :

$$\begin{cases} \dot{x}(t) = (A - Bk)x + B(t)v(t) \\ y(t) = (C - Dk)x(t) + Dv(t) \end{cases} \quad (\text{II.30})$$

Le problème posé ici consiste à trouver un vecteur de gain  $k$  (gain de retour d'état) de telle sorte que la matrice d'état du système en boucle fermée  $(A - Bk)$  Possède les valeurs propres désirées. Pour la stabilisation, les valeurs propres doivent être, dans la partie complexe négative (valeurs propres à partie réelle négative). Pour ce faire, on utilise souvent la méthode de placement de pôles.

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

---

**Exemple :**

Soit le système

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -5 & -4 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -5 & -4 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

1) Vérifier si le système est commandable

Matrice de commandabilité M :

$$M = [B \ AB \ A^2B \ \dots \ A^{n-1}B] \quad \text{comme } n=3 \text{ alors } M = [B \ AB \ A^2B]$$

$$M = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -4 \\ 1 & -4 & 11 \end{bmatrix}$$

Rang(M) = 3 = n alors le système est commandable.

2) Déterminer les coefficients  $a_1, a_2, \dots, a_n$

À partir du polynôme caractéristique de la matrice A.

$$\det(sI - A) = \det \left( s \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -5 & -4 \end{bmatrix} \right)$$

$$\det(sI - A) = s^3 + 4s^2 + 5s + 2$$

$$\det(sI - A) = s^n + a_1s^{n-1} + \dots + a_{n-1}s + a_n$$

Par identification:  $a_1 = 4$

$$a_2 = 5$$

$$a_3 = 2$$

La matrice de transformation  $T = MW$

Matrice W

$$W = \begin{bmatrix} a_2 & a_1 & 1 \\ a_1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 6 & 1 \\ 6 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\text{Alors la matrice } T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -4 \\ 1 & -4 & 11 \end{bmatrix} * \begin{bmatrix} 5 & 6 & 1 \\ 6 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -8 & 2 & 1 \end{bmatrix}.$$

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

---

3) *Pôles en boucle fermée (choisir les valeurs propres désirées) :*

Le polynôme caractéristique correspondant.

$(s - \mu_1)(s - \mu_2) \dots (s - \mu_n) = s^n + \alpha_1 s^{n-1} + \dots + \alpha_{n-1} s + \alpha_n$  ce qui mène aux coefficients  $\alpha_1 \alpha_2 \dots \alpha_n$

Si on choisit les pôles suivants :  $s \pm 4j, s = -10$

Cela mène à

$(s + 2 - 4j)(s + 2 + 4j)(s + 10) = s^3 + 14s^2 + 60s + 200.$

4) *Calcul de la matrice de gain k*

$$k = -[(\alpha_n - a_n)(\alpha_{n-1} - a_{n-1}) \dots (\alpha_1 - a_1)]T^{-1}$$

$$k = -[(200 - 2) (60 - 5) (14 - 4)] \begin{bmatrix} 1 & 2 & -8 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$k = [198 \ 451 \ -1464]$$

### II.3- le chaos dans la transmission sécurisée

En transmission sécurisée d'information binaire, le message appelé texte est transformé de manière à le rendre imprédictible. Ce processus est appelé *chiffrement* ou *cryptage*. Par ailleurs, le destinataire doit engager un processus appelé *déchiffrement* ou *décryptage*, pour reconstruire le message à partir du texte chiffré. Pour cela, des algorithmes sont utilisés, qui sont en effet des fonctions mathématiques destinées au chiffrement et déchiffrement du message. Afin de transmettre le message d'une manière sûre, un élément appelé *clé* de cryptage est introduit, qui est utilisé par l'expéditeur et le destinataire. Cette clé peut prendre des valeurs parmi un grand nombre de valeurs possibles. Certains algorithmes utilisent des clés différentes pour le chiffrement et le déchiffrement. Avec ces algorithmes, toute la sécurité réside dans les clés et non pas dans l'algorithme. Alors même si un espion connaît l'algorithme il ne peut pas détecter le message, car il ne connaît pas la clé.

On distingue deux types de clés: clé secrète et clé publique. Dans l'algorithme à clé secrète, la clé de chiffrement est calculée à partir de la clé de déchiffrement et vice versa. Dans la plus part des cas les deux clés sont identiques, l'expéditeur et le destinataire se mettent d'accord sur une clé avant d'échanger des messages, si cette clé secrète est dévoilée n'importe qui peut lire ce message.

Par contre, un algorithme à clé publique utilise des clés différentes au niveau du chiffrement. Ces clés ne peuvent pas être calculées l'une à partir de l'autre. La clé de chiffrement peut être rendue publique, mais seul celui qui possède la clé de chiffrement peut lire ce message. Alors, la clé de chiffrement est appelée *clé privée*.

En parallèle avec le chiffrage d'informations discrètes ou binaires, des recherches ont été effectuées afin de pouvoir appliquer ce principe de cryptographie aux informations continues. Grâce aux résultats obtenus en synchronisation des systèmes chaotiques, il a été possible

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

d'employer des systèmes chaotiques continus comme porteur d'informations. Dans ce cas, le message est codé par l'émetteur et il est décodé et extrait du signal chaotique par le récepteur. Parmi les méthodes de transmission chaotique, on peut citer le cryptage par addition, le cryptage par commutation et le cryptage par inclusion.

### II.3.1. La définition de cryptage [11] :

La cryptologie [du grec (cryptos) : caché et (graphein) : écrire] est une science mathématique qui comporte deux branches : la cryptographie et la cryptanalyse.

Le principe de la cryptographie chaotique est de noyer le message à transmettre dans un chaos et de l'envoyer à un récepteur qui connaît les caractéristiques du générateur de chaos, qui pourra donc soustraire le chaos au signal reçu et ainsi en extraire le message.

La cryptographie traditionnelle est l'étude des méthodes permettant de transmettre des données de manière confidentielle. Afin de protéger un message, on lui applique une transformation qui le rend incompréhensible ; c'est ce qu'on appelle *le chiffrement*.

Le déchiffrement est l'action qui permet de reconstruire le texte en clair à partir du texte chiffré.

La cryptographie moderne, les transformations en question sont des fonctions.

Le déchiffrement est l'action qui permet de reconstruire le texte en claire à partir du texte chiffré. Ces transformations en question sont des fonctions mathématique, appelée *algorithme cryptographique* qui dépend d'un paramètre appelé *clés*.

La cryptanalyse est l'étude des procédés cryptographiques dans le but de trouver des faiblesses et en particulier, de pouvoir décrypter des textes chiffrés.

Le décryptement est l'action consistant à retrouver le texte en clair sans connaître la clé de déchiffrements.

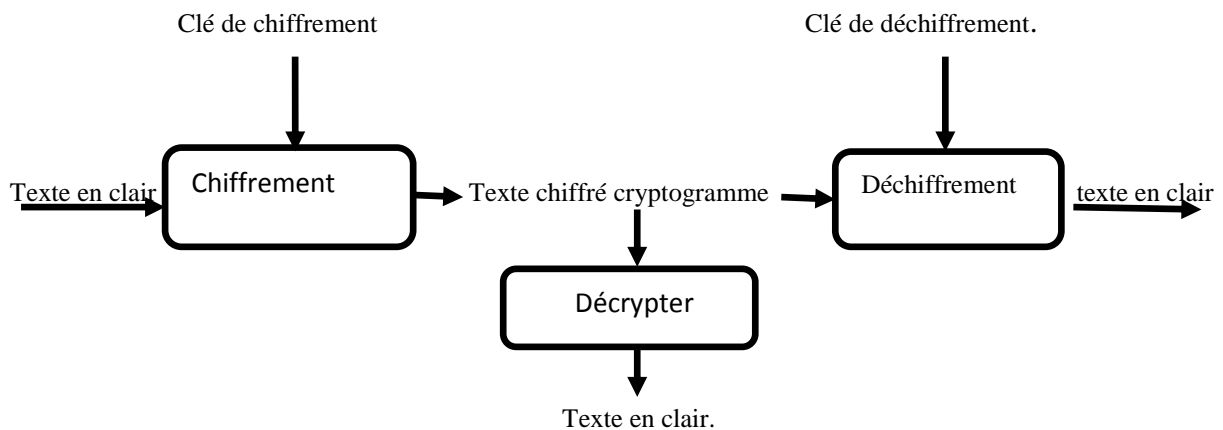


Figure (II.9) : Cryptographie et cryptanalyse.

### II.3.2- Les techniques de cryptage [11][12] :

Nous avons plusieurs techniques de cryptage, parmi ces techniques on trouve :

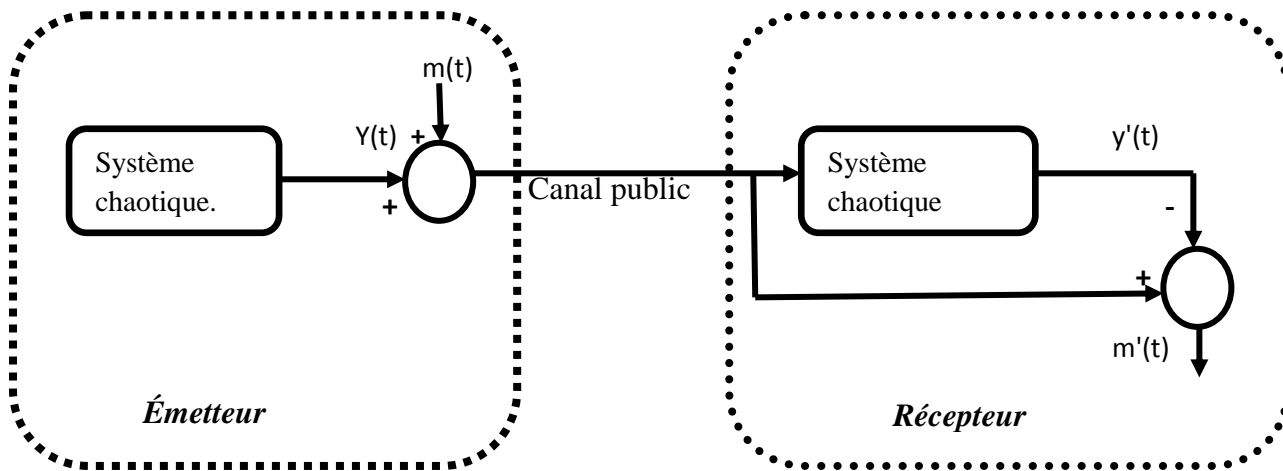
#### a) Le cryptage par addition :

Dans le cryptage additif le message  $m(t)$  est tout simplement additionné au signal chaotique  $y(t)$  ; et le signal résultant est envoyé au récepteur pour la synchronisation.

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

En conséquence, après la synchronisation, le message confidentiel peut être récupéré par une simple opération de soustraction entre la sortie du récepteur et le signal émis sur le canal public.

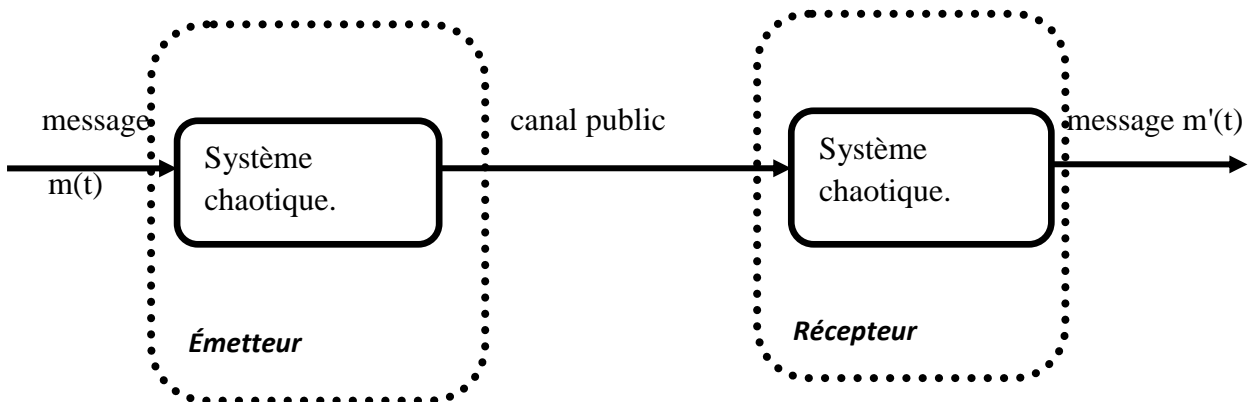
*L'inconvénient de cette méthode* est qu'afin de garantir la synchronisation, le message doit être au moins de 20 à 30 dB inférieurs à la sortie de l'émetteur, or en présence d'un bruit de canal d'une puissance proche de celle du message, il devient difficile de détecter l'information. De plus, cette méthode reste sensible aux attaques extérieures et l'usage du canal de transmission est inefficace du point de vue de l'énergie transmise par rapport à la qualité d'information fournie.



Figure(II.10) : Cryptage par la méthode additive.

### b) Le cryptage par inclusion :

Dans le cryptage par inclusion, le message source du système est inclus dans la structure du système chaotique du côté de l'émission. Dans ce cas, un observateur doit être utilisé à la réception pour récupérer le message original. Cette méthode nécessite un seul canal de transmission.



Figure(II.11) : Cryptage par la méthode d'inclusion.

## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

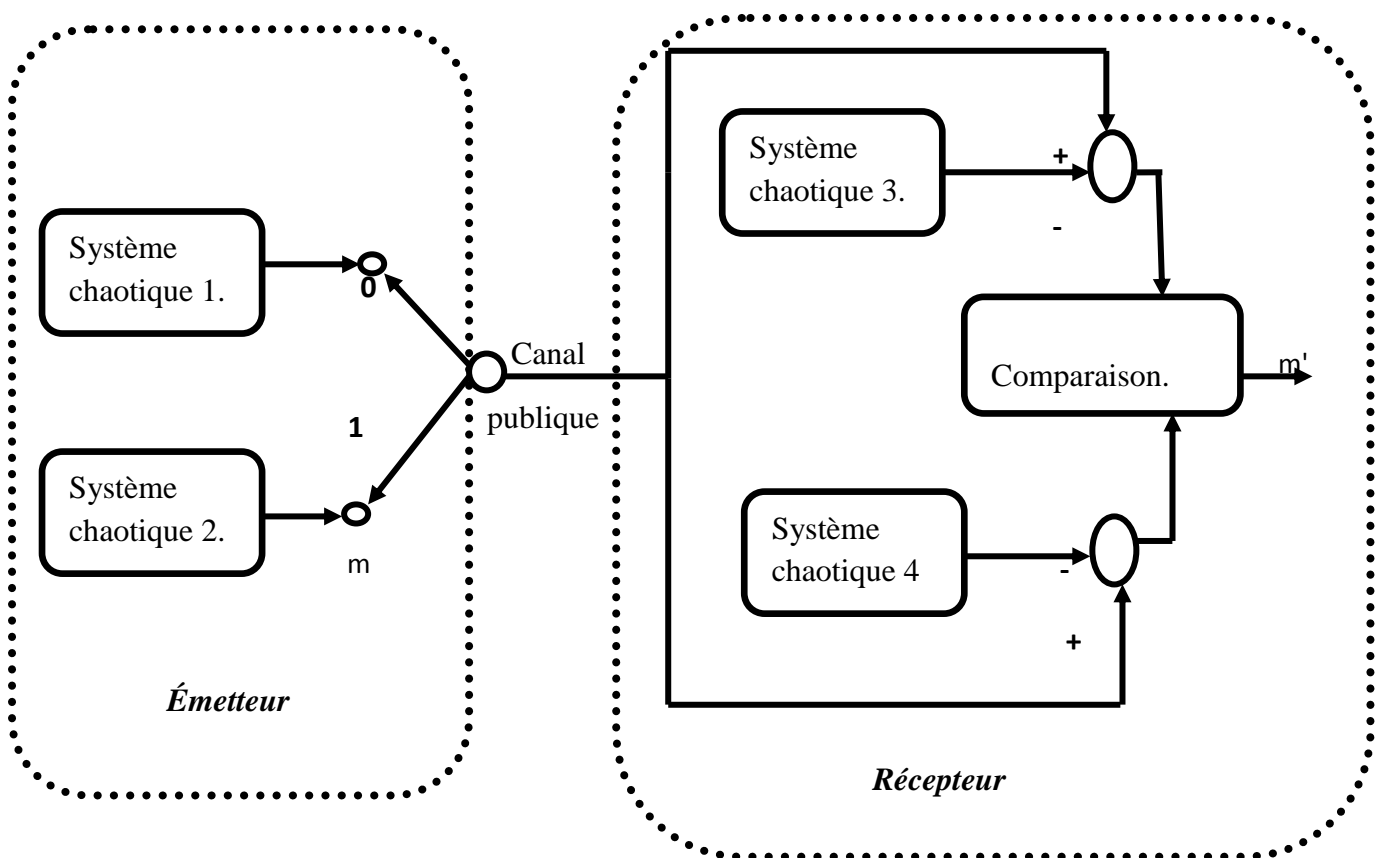
### c) Le cryptage par commutation :

Un autre schéma de transmission chaotique de donnée est la communication chaotique, qui exige que le message soit binaire, et s'inspire de la modulation à saut de fréquence.

Le diagramme de cette approche est illustré en figure (II.12), où une opération de commutation est employée selon la valeur du message binaire :

Si sa valeur est 0 alors le système chaotique 1 est choisi et le signal de sortie est transmis, sinon la sortie du système chaotique 2 est transmise. Dans ce sens, le message binaire commute l'émetteur entre deux attracteurs étranges correspondants à deux systèmes chaotiques.

Du côté récepteur, il y a deux sous-systèmes chaotiques 3 et 4 qui correspondent respectivement à 1 et 2. Supposant que le canal est parfait, et que le signal transmis est 0 alors le sous-système 3 se synchronisera avec le système chaotique 1, mais le sous-système 4 ne pourra pas être synchronisé. Selon les erreurs de synchronisation (1,3) et (2,4), le signal peut-être récupéré avec succès.



Figure(II.12) : cryptage par commutation.

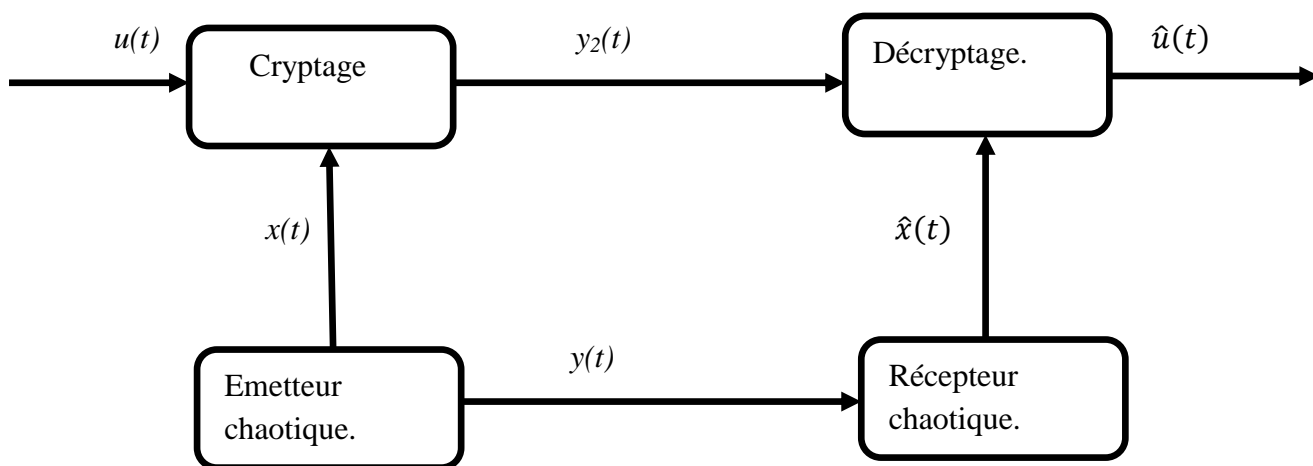
## Chapitre II : Les méthodes de synchronisation des systèmes chaotiques

### II.3.3 La transmission à deux voies :

Dans le schéma présenté à la figure (II.13), l'émetteur envoie deux signaux au récepteur. Le premier,  $y$ , est une fonction à valeurs réelles de l'état  $x$  du système chaotique émetteur, dont l'unique but est de permettre la synchronisation du récepteur.

Le seconde,  $y_2$  envoyée éventuellement sur un autre canal (est un signal chaotique qui contient l'information à transmettre). Parmi les avantages de cette méthode, on peut souligner, d'une part que le signal  $y$  ne contient aucune information, par conséquent la synchronisation peut s'établir de façon optimale. D'un autre côté, le second signal  $y_2$  contient l'information qui peut être soit cryptée par une fonction non linéaire de l'état  $x$ , soit simplement masquée par un signal chaotique généré par l'émetteur, qui sert de porteuse.

On peut noter également que les deux étapes de synchronisation et de cryptage étant totalement indépendantes, le décryptage n'est pas nécessairement effectué, au niveau du récepteur, en même temps que la synchronisation.



Figure(II.13) : Transmission à deux voies.

### II.4- Conclusion :

Ce chapitre a permis de faire le lien entre les systèmes dynamiques chaotiques et les systèmes de communication. Nous avons abordé le phénomène de synchronisation et présenté par la suite les principales méthodes utilisées pour la synchronisation des systèmes chaotiques. Le concept de la synchronisation du chaos de Picora et Carroll a ouvert la voie à l'utilisation du chaos dans les télécommunications. Ces derniers ont montré que deux systèmes chaotiques identiques peuvent se synchroniser s'ils sont couplés sous certaines conditions.

Les principales méthodes de transmission basée sur la synchronisation de systèmes chaotiques ont fait également l'objet de ce chapitre. Dans la méthode de cryptage par addition nous avons signalé ses principaux inconvénients qui sont sa sensibilité aux attaques extérieures ainsi que la difficulté pour détecter le message utile en cas de présence de bruit de canal de transmission. La méthode de cryptage par inclusion reste moins sensible aux perturbations extérieures.

# Chapitre III

# Chapitre III : Synchronisation de système hyperchaotique de Qi

## III.1. Introduction :

L'idée de synchroniser deux systèmes chaotiques dans différentes conditions initiales a été présentée la première fois par Poccora et Carrol en 1989. La synchronisation du chaos, comme matière importe en science non linéaire, a été largement étudiée dans une variété de champs, tels que la physique, la chimie et la sécurité de la communication. Par conséquent, il a été proposé une grande variété d'approches pour la synchronisation des systèmes chaotiques avec la commande linéaire ou non linéaire, et le contrôle adaptatif. Guoyuan QI et autre ont présenté un nouveau système à quatre dimensions, nommé système de QI. Les propriétés de base du nouveau système 4D ont été analysées au moyen de l'exposant de Lyapunov. Leurs études prouvent que le système peut se produire divers attracteurs chaotiques complexes quant les paramètres sont choisis pour les deux cas c'est-à-dire un cas autonome et un cas non autonome.

Ce chapitre est consacré à la synchronisation de deux systèmes chaotiques identiques (émetteur et récepteur). Dans un premier temps, nous allons définir un oscillateur appelé <<système de QI>>, par la caractérisation de chaos dans ce dernier par les exposants de Lyapunov et le spectre de Puissance. Par le suit nous allons synthétiser un observateur en alinéatisant le système d'erreur par un retour d'état, puis nous définirons les équations mathématiques de notre récepteur. Puis nous le simulerons sur Matlab.

## III.2. Émetteur :

Afin de générer un signal chaotique et noyer un message, nous avons besoin d'un oscillateur électronique.

### III.2.1. Définition d'un oscillateur électronique :

Un oscillateur électronique est un montage électrique autonome, dont la fonction principale est de produire un signal périodique lors de la mise sous la tension. La forme du signal peut être sinusoïdale, carrée, ou quelconque.

Aussi, un oscillateur comporte toujours un élément actif (circuit amplificateur) associé à un circuit passif (un filtre). L'élément actif est souvent un transistor bipolaire ou un amplificateur opérationnel. Sa structure est celle d'un système bouclé dans lequel une fonction du signal de sortie est ramenée à l'entrée pour l'auto entretien des oscillations. D'une façon générale, un oscillateur est représenté par le schéma bloc ci-dessous :

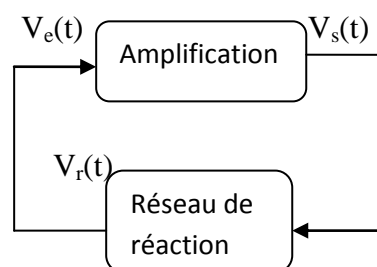


Figure (III.1) : représentation d'un oscillateur.

## Chapitre III : Synchronisation de système hyperchaotique de Qi

Si le signal  $V_r(t)$  est identique à  $V_e(t)$ , alors on peut le boucler à l'entrée de l'amplificateur. Cette condition n'est satisfaite que pour une fréquence bien définie appelée la fréquence d'oscillation. La structure d'un tel oscillateur se reconnaît par le fait que la sortie de l'amplificateur est rebouclée sur l'entrée via le réseau de réaction qui est un circuit passif.

### III.2.2. Système hyperchaotique de Qi [13] :

Le système de Qi est un oscillateur, décrit par les équations suivantes :

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1) + x_2x_3 \\ \dot{x}_2 = b(x_1 + x_2) - x_1x_3 \\ \dot{x}_3 = -cx_3 - ex_4 + x_1x_2 \\ \dot{x}_4 = -dx_4 + fx_3 + x_1x_2 \end{cases} \quad (\text{III.1})$$

$x_1, x_2, x_3, x_4$  sont des variables d'état ;  $a, b, c, d, e, f$  sont des constantes positives.

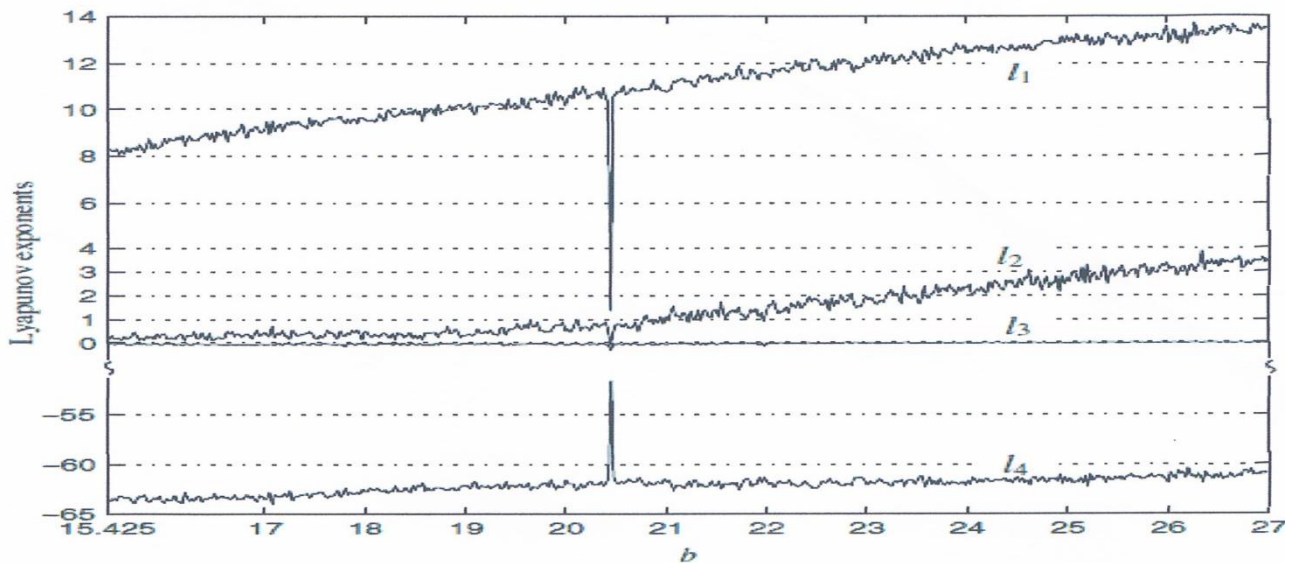
#### III.2.2.1 Caractérisation du chaos dans le système de Qi :

##### a- Exposant de Lyapunov [13] :

$15.425 \leq b \leq 27, a=50, c=13, d=8, e=33, f=30$ .

On peut observer qu'il y a deux exposants de Lyapunov positifs sur tout un intervalle de valeurs de paramètres. Pendant que le paramètre  $b$  varie, comme on le voit sur la figure, le premier exposant  $L_1$  est très grand ( $8.358 \leq L_1 \leq 13.4632$ ) le deuxième exposant de Lyapunov  $L_2$  est assez grand ( $12 \leq L_2 \leq 3.4781$ ), le troisième exposant est approximativement nul, et le quatrième est négatif ( $L_4 < -60$ ).

La figure (III.2) représente l'exposant de Lyapunov.

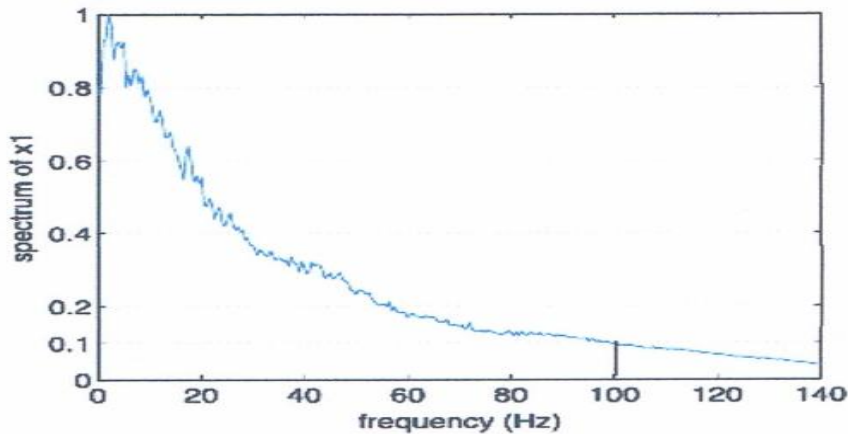


**Figure (III.2) : L'exposant de Lyapunov pour le système de Qi [13].**

### *b- Spectre de puissance :*

Le spectre de puissance est la transformée de Fourier de la fonction d'autocorrélation. Le spectre de puissance d'un système chaotique est continu et évolue de manière exponentielle vers zéro.

La figure (III.3) représente le spectre de puissance de système de Qi.



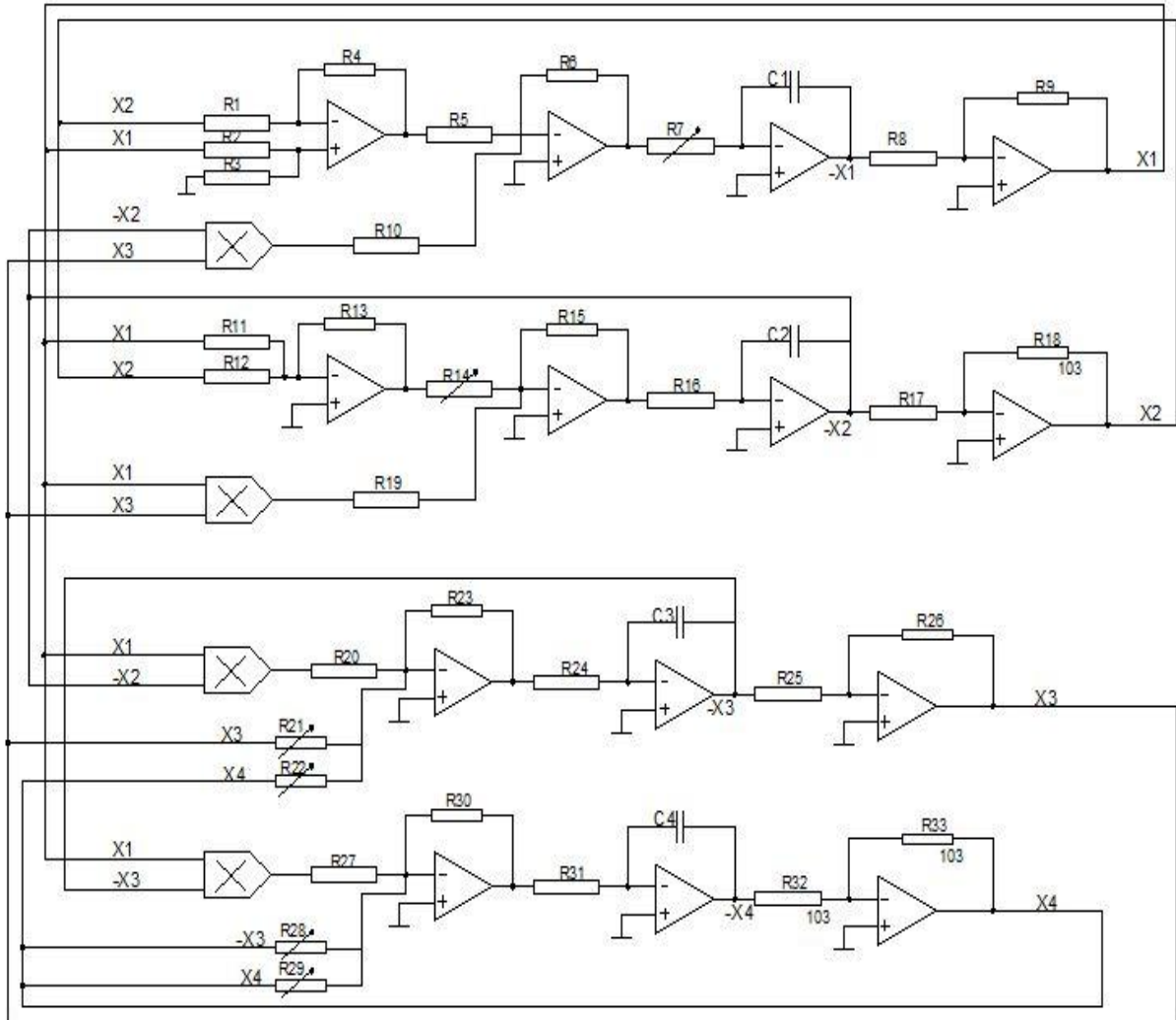
*Figure (III.3) : Le spectre de Puissance pour l'état  $x_1$  de système de Qi [13].*

### *c- Le schéma électrique du système Qi :*

$R_1, R_2, R_3, R_4, R_8, R_9, R_{15}, R_{17}, R_{18}, R_{20}, R_{22}, R_{26}, R_{27}, R_{30}, R_{32}, R_{33} = 10k\Omega$ .

$R_5, R_6, R_{16}, R_{23}, R_{31} = 100k\Omega$  ;  $R_{16}, R_{24}, R_{31} = 100k\Omega$  ;  $R_7 = 20k\Omega$  ;  $R_{14} = 4.17k\Omega$  ;  
 $R_{21} = 7.69k\Omega$  ;  $R_{22} = 3.03k\Omega$  ;  $R_{28} = 3.33k\Omega$  ;  $R_{29} = 12.5k\Omega$  ;  $R_{10} = 500\Omega$  ;  $R_{19}, R_{20},$   
 $R_{27} = 100\Omega$ .

$C_1 = C_2 = C_3 = C_4 = 5nF$ .



**Figure(III.4) : Schéma de réalisation du système hyperchaotique de Qi.**

**d- Étude du montage électrique :**

On prend :  $S_1$  sortie du première AOP,  $S_2$  sortie du deuxième AOP, et ainsi de suite.

➤ **Pour le premier étage :**

$$V^+ = \frac{R_3}{R_2 + R_3} x_1 \tag{1}$$

Comme  $R_2 = R_3$  alors l'équation (1) devient :  $V^+ = V^- = \frac{1}{2} x_1$  (2)

$$S_1 - V^- = -R_4 \cdot i \tag{3}$$

$$x_2 - V^- = R_1 \cdot i \Rightarrow i = \frac{x_2 + V^-}{R_1} \tag{4}$$

## Chapitre III : Synchronisation de système hyperchaotique de Qi

---

On remplace (4) dans (3):

$$S_1 = V^- - R_4 \frac{x_2 + V^-}{R_1} \quad (5)$$

On remplace (2) dans (5):

$$S_1 = \frac{1}{2}x_1 - (x_2 \frac{1}{2}x_1) = (x_2 - x_1)$$

$$S_2 = R_6 \cdot i \quad (6)$$

$$\text{On a: } i = i_1 + i_2 \quad (7)$$

$$\text{Avec: } i_1 = \frac{S_2}{R_5}, \quad i_2 = \frac{x_2 x_3}{R_{10}}$$

$$i = -\frac{(x_2 - x_1)}{R_5} + \frac{x_2 x_3}{R_{10}}$$

$$S_2 = -\frac{R_6}{R_5}(x_2 - x_1) + \frac{R_6}{R_{10}}x_2 x_3$$

$$S_3 = -\frac{1}{C_1} \int i dt, \quad i = \frac{S_2}{R_7}$$

$$\dot{x}_1 = \frac{R_6}{R_5}(x_2 - x_1) + \frac{R_6}{R_{10}}(x_2 x_3) \quad (8)$$

$$x_1 = \frac{1}{R_7 C_1} \int \left( \frac{R_6}{R_5}(x_2 - x_1) + \frac{R_6}{R_{10}}(x_2 x_3) \right) dt$$

➤ **Pour le deuxième étage:**

$$S_1 = R_{13} i \quad (9)$$

$$i = i_1 + i_2 \quad (10)$$

Avec:

$$i_1 = \frac{x_1}{R_4}, \quad i_2 = \frac{x_2}{R_{12}}$$

On remplace (9) dans (8) et on obtient:

$$S_1 = R_{13} \left( \frac{x_1}{R_4} + \frac{x_2}{R_{12}} \right) = -(x_1 + x_2)$$

$$S_2 = R_{15} \cdot i \quad (11)$$

$$i = i_1 + i_2 \quad (12)$$

Avec :

$$i_1 = \frac{S_2}{R_{14}}$$

$$i_2 = \frac{x_1 x_2}{R_{19}}$$

On remplace (11) dans (10) et on obtient :

$$S_2 = R_{15} \left( \frac{S_2}{R_{14}} + \frac{x_1 x_3}{R_{19}} \right)$$

$$S_3 = -\frac{1}{C_2} \int i dt, \quad i = \frac{S_2}{R_{16}}$$

$$\dot{x}_2 = \frac{R_{15}}{R_{14}} (x_2 + x_1) - \frac{R_{15}}{R_{19}} (x_2 x_3) \quad (13)$$

$$x_2 = -\frac{1}{R_{16} C_2} \int \left( \frac{R_{15}}{R_{14}} (x_2 + x_1) - \frac{R_{15}}{R_{19}} (x_2 x_3) \right) dt$$

➤ **Pour le troisième étage:**

$$S_1 = R_{23} i \quad (14)$$

$$i = i_1 + i_2 \quad (15)$$

Avec:

$$i_1 = \frac{x_1 x_2}{R_{20}}$$

$$i_2 = \frac{x_3}{R_{21}} + \frac{x_4}{R_{22}}$$

$$\text{Donc: } i = \frac{x_1 x_2}{R_{20}} + \frac{x_3}{R_{21}} + \frac{x_4}{R_{22}}$$

L'équation (13) deviente :

$$S_1 = R_{23} \left( \frac{x_1 x_2}{R_{20}} + \frac{x_3}{R_{21}} + \frac{x_4}{R_{22}} \right)$$

$$S_2 = -\frac{1}{C_3} \int i dt$$

$$\dot{x}_3 = \frac{R_{23}}{R_{20}} x_3 - \frac{R_{23}}{R_{20}} x_4 + \frac{R_{23}}{R_{20}} (x_1 x_2) \quad (16)$$

$$\dot{x}_3 = -\frac{1}{R_{24}C_3} \int \left( -\frac{R_{23}}{R_{20}} x_3 - \frac{R_{23}}{R_{22}} x_4 + \frac{R_{23}}{X_{20}} (x_1 x_2) \right) dt$$

➤ Pour le quatrième étage :

$$S_1 = -R_{30} \dot{i} \quad (17)$$

$$\dot{i} = \dot{i}_1 + \dot{i}_2 \quad (18)$$

Avec:

$$\dot{i}_1 = \frac{x_1 x_2}{R_{27}}$$

$$\dot{i}_2 = \frac{x_3}{R_{28}} + \frac{x_4}{R_{29}}$$

$$\text{Donc: } \dot{i} = \frac{x_1 x_2}{R_{27}} + \frac{x_3}{R_{28}} + \frac{x_4}{R_{29}}$$

L'équation (13) devient :

$$S_1 = -R_{30} \left( \frac{x_1 x_2}{R_{27}} + \frac{x_3}{R_{28}} + \frac{x_4}{R_{29}} \right)$$

$$S_2 = -\frac{1}{C_4} \int i dt$$

$$\dot{x}_4 = \frac{R_{30}}{R_{28}} x_3 + \frac{R_{30}}{R_{29}} x_4 + \frac{R_{30}}{X_{27}} (x_1 x_2) \quad (19)$$

$$\dot{x}_4 = -\frac{1}{R_{24}C_3} \int \left( \frac{R_{30}}{R_{28}} x_3 + \frac{R_{30}}{R_{29}} x_4 + \frac{R_{30}}{X_{27}} (x_1 x_2) \right) dt$$

Notre oscillateur comporte quatre étages et on a pu extraire ses équations :

$$\dot{x}_1 = \frac{R_6}{R_5} (x_2 - x_1) + \frac{R_6}{R_{10}} (x_2 x_3) \quad (\text{III.2})$$

$$\dot{x}_2 = \frac{R_{15}}{R_{14}} (x_2 + x_1) - \frac{R_{15}}{R_{19}} (x_2 x_3) \quad (\text{III.3})$$

$$\dot{x}_3 = \frac{R_{23}}{R_{20}} x_3 - \frac{R_{23}}{R_{20}} x_4 + \frac{R_{23}}{R_{20}} (x_1 x_2) \quad (\text{III.4})$$

$$\dot{x}_4 = \frac{R_{30}}{R_{28}} x_3 + \frac{R_{30}}{R_{29}} x_4 + \frac{R_{30}}{X_{27}} (x_1 x_2) \quad (\text{III.5})$$

On remarque si on varié les valeurs des résistances variables on obtient les mêmes coefficients que les équation autonomes(III.2),(III.3),(III.4),(III.5) du système de Qi .

## Chapitre III : Synchronisation de système hyperchaotique de Qi

### III.2.3. Simulation sur Matlab :

On insère les équations du système de Qi dans Matlab, on observe le comportement de chaque étage durant la simulation, la simulation se fera pour les valeurs suivantes :

$a=50$ ,  $b=20$ ,  $c=13$ ,  $d=8$ ,  $e=33$ ,  $f=30$ ; on obtient :

#### a) Visualisation des états :

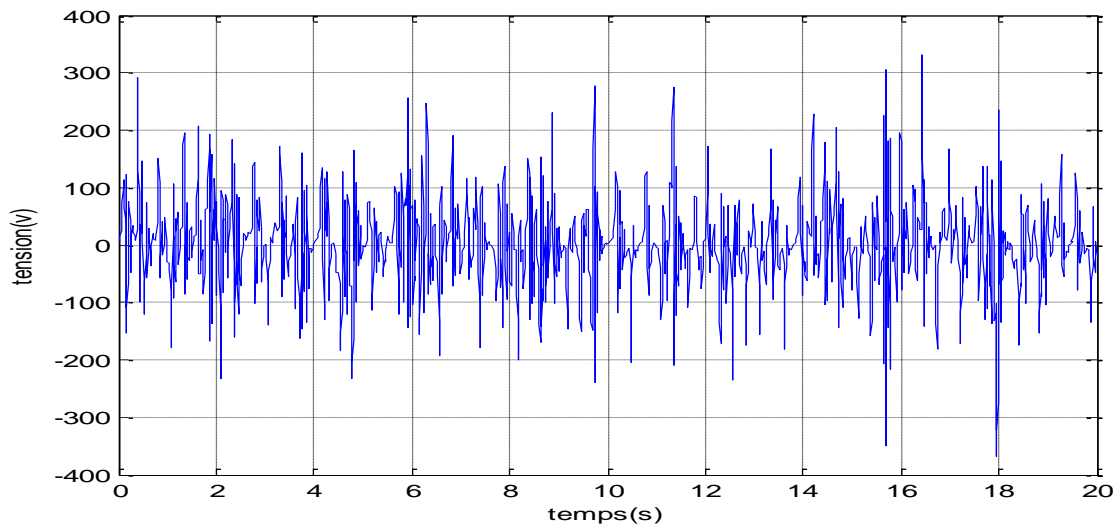


Figure (III.5): Graphe de l'état  $x_1(t)$  du système de Qi.

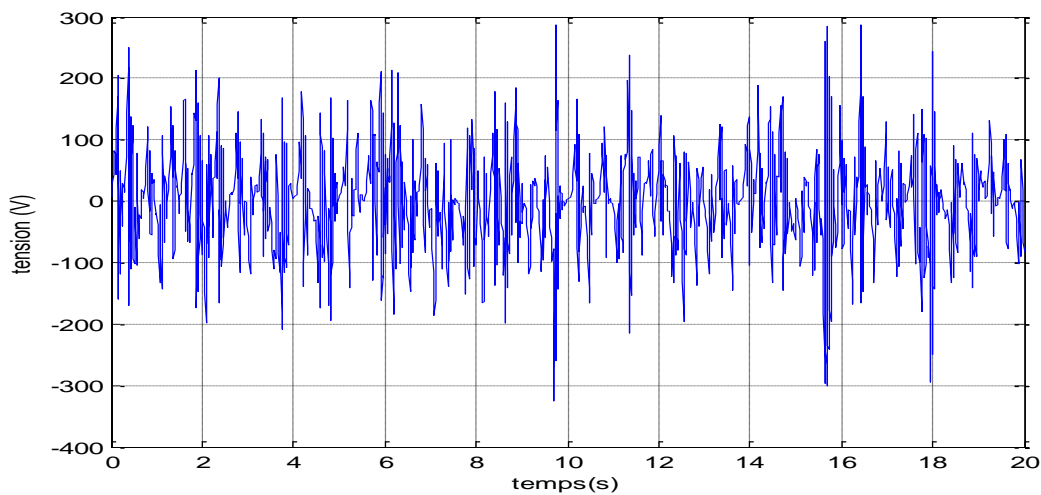
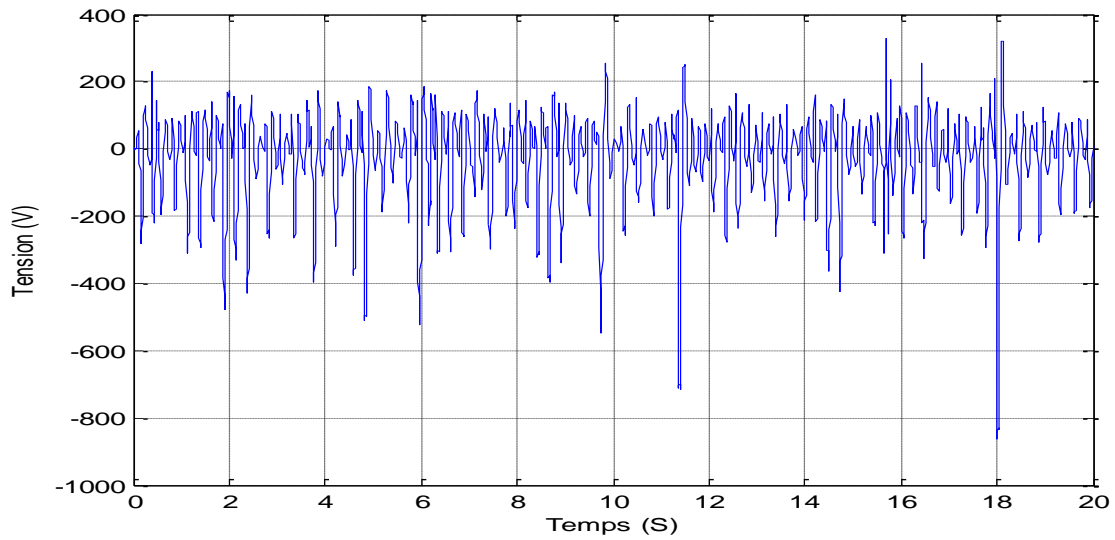
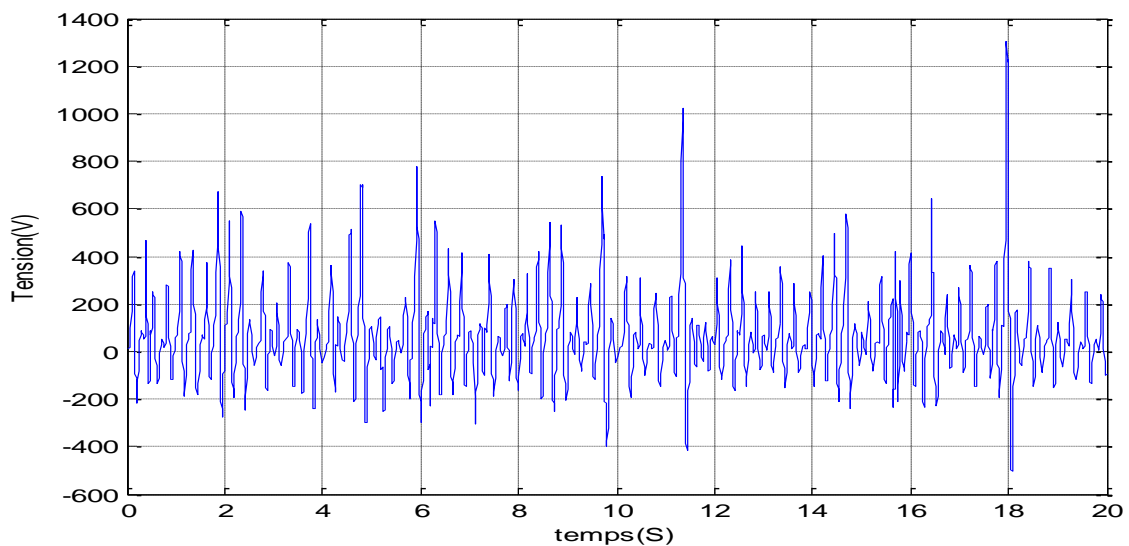


Figure (III.6) : Graphe d'état  $x_2(t)$  du système de Qi.



*Figure (III.7) : Graphe d'état  $x_3(t)$  du système de Qi.*



*Figure (III.8): Graphe d'état  $x_4(t)$  du système de Qi.*

❖ **Observation :**

Nous remarquons la présence d'oscillations apériodiques et irrégulières dans ces figures alors les signaux  $x_1$ ,  $x_2$ ,  $x_3$ ,  $x_4$  sont des signaux chaotiques.

## b) Visualisation des Attracteurs :

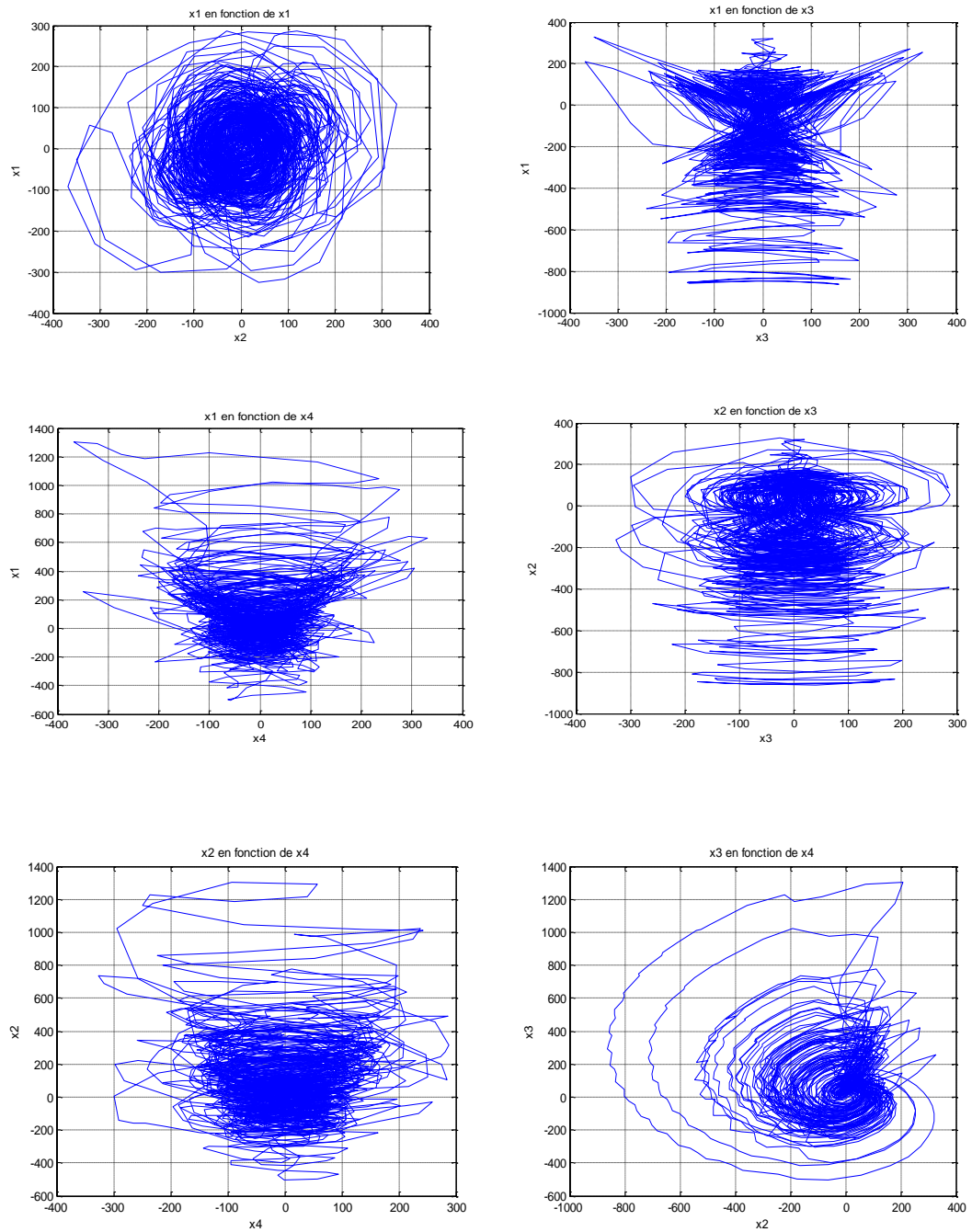
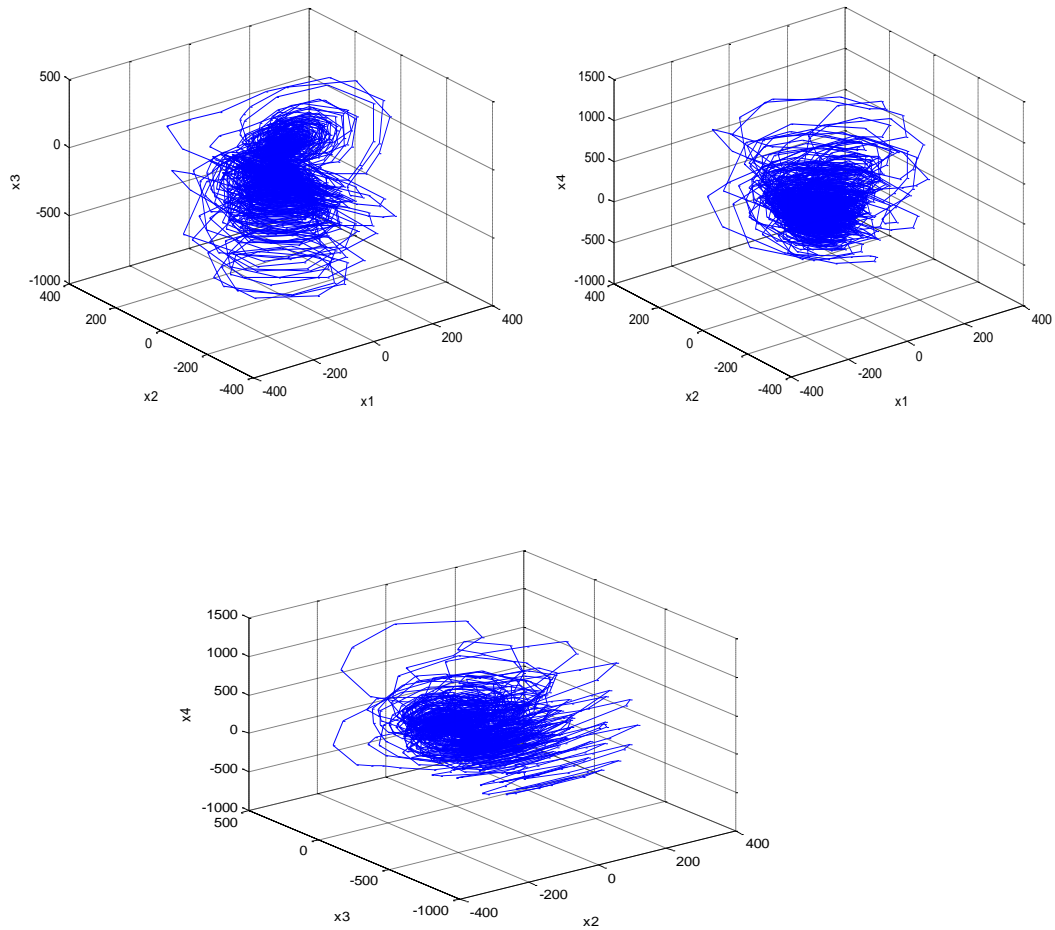


Figure (III.9) : Les attracteurs de phase du système de Qi observés sous Matlab.

c) *Visualisation des attracteurs en trois dimension :*



*Figure (III.10) : Attracteur en trois dimensions.*

❖ *Observation :*

Ces figures démontrent l'aspect hyper chaotique de système de Qi, et l'observation de ces attracteurs se fait à partir de plusieurs paramètres très sensibles aux changements.

### III.3. Le Récepteur :

Un système dynamique est dit observable si on peut récupérer toutes ses grandeurs par une combinaison de mesures et de leurs dérivées, Nijmeir et Mareels sont les premiers qui ont montré que la synchronisation unidirectionnelle de deux systèmes chaotiques peut être considérée comme un problème d'observateur non linéaire.

On peut utiliser un observateur pour estimer les états inconnus d'un système qui ne sont pas mesurable directement.

#### III.3.1. Condition du rang d'observabilité :

Un système est dit observable si on applique la dérivée de lie et si le rang de la matrice obtenue est égale au rang du système étudié.

$$\dot{x} = f(x)$$

$$y = c(x) = h(t)$$

$$L_f h = \frac{dh}{dx} f, \quad o = \begin{pmatrix} dh \\ dL_f h \\ dL_f^2 h \\ \vdots \\ dL_f^{n-1} h \end{pmatrix}, \quad \text{rang}(o) = n$$

On considère le système suivant :

$$f(x) = \begin{pmatrix} a(x_2 - x_1) + x_2 x_3 \\ b(x_1 + x_2) - x_1 x_3 \\ -c x_3 - e x_4 + x_1 x_2 \\ -d x_4 + f x_3 + x_1 x_2 \end{pmatrix}, \quad y = (0 \ 0 \ x_3 \ 0) = h(t).$$

$$h(t) = x_3$$

$$L_f h(t) = -c x_3 - e x_4 + x_1 x_2$$

$$L_f^2 h(t) = L_f(L_f h(t)) = 20x_1^2 - 26x_1 x_2 - x_1^2 x_3 + x_2^2 x_3 - 169x_3 + 533x_4.$$

$$L_f^3 h(t) = L_f(L_f^2 h(t)) = -x_1^3 x_2 + x_2^2 x_1 + 83x_1^2 x_3 - 2000x_1^2 - 8x_4 x_1^2 + x_3 x_2^2 - 8x_4 x_2^2 - 102x_1 x_2 x_3^2 + 2224x_1 x_2 - 60x_1 x_2 x_3 + 500x_3 + 296x_4.$$

Pour  $i=1\dots 4$

$$\frac{dh}{dx_i} = \left( \frac{dh}{dx_1} \quad \frac{dh}{dx_2} \quad \frac{dh}{dx_3} \quad \frac{dh}{dx_4} \right)$$

$$\frac{dL_f h}{dx_i} = \left( \frac{dL_f h}{dx_1} \quad \frac{dL_f h}{dx_2} \quad \frac{dL_f h}{dx_3} \quad \frac{dL_f h}{dx_4} \right)$$

$$\frac{dL_{f^2} h}{dx_i} = \left( \frac{dL_{f^2} h}{dx_1} \quad \frac{dL_{f^2} h}{dx_2} \quad \frac{dL_{f^2} h}{dx_3} \quad \frac{dL_{f^2} h}{dx_4} \right)$$

$$\frac{dL_{f^3} h}{dx_i} = \left( \frac{dL_{f^3} h}{dx_1} \quad \frac{dL_{f^3} h}{dx_2} \quad \frac{dL_{f^3} h}{dx_3} \quad \frac{dL_{f^3} h}{dx_4} \right)$$

$$\left\{ \begin{array}{l} \frac{dh}{dx_1} = 0 \\ \frac{dh}{dx_2} = 0 \\ \frac{dh}{dx_3} = 1 \\ \frac{dh}{dx_4} = 0 \end{array} \right. \quad (20)$$

$$\left\{ \begin{array}{l} \frac{dL_f h}{dx_1} = x_2 \\ \frac{dL_f h}{dx_2} = x_1 \\ \frac{dL_f h}{dx_3} = -13 \\ \frac{dL_f h}{dx_4} = -33 \end{array} \right. \quad (21)$$

$$\left\{ \begin{array}{l} \frac{dL_{f^2} h}{dx_1} = (-2x_1x_3 - 26x_2 + 40x_1) \\ \frac{dL_{f^2} h}{dx_2} = (2x_1x_3 - 26x_1) \\ \frac{dL_{f^2} h}{dx_3} = (-x_1^2 + x_2^2 - 169) \\ \frac{dL_{f^2} h}{dx_4} = (533) \end{array} \right. \quad (22)$$

$$\left\{ \begin{array}{l} \frac{dL_{f^3}h}{dx_1} = (3x_1^2x_2 + x_2^3 + 166x_1x_3 - 4000x_1 - 16x_4x_1 + 102x_2x_3^2 - 2224x_2 - 60x_3x_2) \\ \frac{dL_{f^3}h}{dx_2} = (x_1^3 + 3x_2^2 + 16x_4x_2 - 108x_2 - 102x_1x_3^2 + 2224x_1 - 60x_1x_3) \\ \frac{dL_{f^3}h}{dx_3} = (83x_1^3 + x_1^2 + x_2^3 + 204x_1x_2x_3 - 60x_1x_2 + 500) \\ \frac{dL_{f^3}h}{dx_4} = (-8x_2^2 + 296) \end{array} \right. \quad (23)$$

$$\left( \begin{array}{c} \frac{dh}{dx_i} \\ \frac{dL_f h}{dx_i} \\ \frac{dL_{f^2} h}{dx_i} \end{array} \right) = \left( \begin{array}{cccc} 0 & 0 & 1 & 0 \\ x_2 & x_1 & -13 & -33 \\ \frac{dL_{f^2} h}{dx_1} & \frac{dL_{f^2} h}{dx_2} & \frac{dL_{f^2} h}{dx_3} & \frac{dL_{f^2} h}{dx_4} \\ \frac{dL_{f^3} h}{dx_1} & \frac{dL_{f^3} h}{dx_2} & \frac{dL_{f^3} h}{dx_3} & \frac{dL_{f^3} h}{dx_4} \end{array} \right)$$

Le rang du système est égal à 4 donc le théorème de lie est vérifié alors notre système est **observable**.

### III.3.2. Synchronisation du système de Qi :

Nous allons réaliser une synchronisation entre deux systèmes de Qi identiques définis comme émetteur et récepteur.

On choisira le système maître (système de Qi), défini par :

$$\left\{ \begin{array}{l} \dot{x}_{1m} = a(x_{2m} - x_{1m}) + x_{2m}x_{3m} \\ \dot{x}_{2m} = b(x_{1m} + x_{2m}) - x_{1m}x_{3m} \\ \dot{x}_{3m} = -cx_{3m} - ex_{4m} + x_{1m}x_{2m} \\ \dot{x}_{4m} = -dx_{4m} + fx_{3m} + x_{1m}x_{2m} \end{array} \right. \quad (III.6)$$

Nous associons un système identique esclave, donné par :

$$\left\{ \begin{array}{l} \dot{x}_{1s} = a(x_{2s} - x_{1s}) + x_{2s}x_{3s} + U_1 \\ \dot{x}_{2s} = b(x_{1s} + x_{2s}) - x_{1s}x_{3s} + U_2 \\ \dot{x}_{3s} = -cx_{3s} - ex_{4s} + x_{1s}x_{2s} + U_3 \\ \dot{x}_{4s} = -dx_{4s} + fx_{3s} + x_{1s}x_{2s} + U_4 \end{array} \right. \quad (III.7)$$

$y = [0 \ 0 \ x_3 \ 0]$ , avec  $U(t) = [U_1(t), U_2(t), U_3(t), U_4(t)]^T$  vecteur de commande.

## Chapitre III : Synchronisation de système hyperchaotique de Qi

Pour avoir une synchronisation entre le système maître et le système esclave, on applique une commande pour forcer le système esclave à suivre le système maître.

Soit les écarts entre les états de l'esclave et du maître comme suit :

$$\begin{cases} e_1 = x_{1s} - x_{1m} \\ e_2 = x_{2s} - x_{2m} \\ e_3 = x_{3s} - x_{3m} \\ e_4 = x_{4s} - x_{4m} \end{cases} \quad (\text{III.8})$$

alors, on obtient :

$$\begin{cases} \dot{x}_{1s} - \dot{x}_{1m} = -a(x_{1s} - x_{1m}) + a(x_{2s} - x_{2m}) - x_{2m}x_{3m} + x_{2s}x_{3s} + U_1 \\ \dot{x}_{1s} - \dot{x}_{1m} = b(x_{1s} - x_{1m}) + b(x_{2s} - x_{2m}) - x_{1s}x_{3s} + x_{1m}x_{3m} + U_2 \\ \dot{x}_{1s} - \dot{x}_{1m} = -c(x_{3s} - x_{3m}) - e(x_{4s} - x_{4m}) + x_{1s}x_{2s} - x_{1m}x_{2m} + U_3 \\ \dot{x}_{1s} - \dot{x}_{1m} = f(x_{3s} - x_{3m}) - d(x_{4s} - x_{4m}) + x_{1s}x_{2s} - x_{1m}x_{2m} + U_4 \end{cases} \quad (\text{III.9})$$

Nous appliquons la commande qui linéarise le système (III.9)

$$\begin{cases} V_1 = -x_{2m}x_{3m} + x_{2s}x_{3s} + U_1 \\ V_2 = -x_{1s}x_{3s} + x_{1m}x_{3m} + U_2 \\ V_3 = x_{1s}x_{2s} - x_{1m}x_{2m} + U_3 \\ V_4 = x_{1s}x_{2s} - x_{1m}x_{2m} + U_4 \end{cases} \quad (\text{III.10})$$

Et  $V(t) = [V_1(t), V_2(t), V_3(t), V_4(t)]^T$  Le vecteur de commande.

Alors le système (III.9) devient :

$$\begin{cases} \dot{x}_{1s} - \dot{x}_{1m} = -a(x_{1s} - x_{1m}) + a(x_{2s} - x_{2m}) + V_1 \\ \dot{x}_{1s} - \dot{x}_{1m} = b(x_{1s} - x_{1m}) + b(x_{2s} - x_{2m}) + V_2 \\ \dot{x}_{1s} - \dot{x}_{1m} = -c(x_{3s} - x_{3m}) - e(x_{4s} - x_{4m}) + V_3 \\ \dot{x}_{1s} - \dot{x}_{1m} = f(x_{3s} - x_{3m}) - d(x_{4s} - x_{4m}) + V_4 \end{cases}$$

Et donc :

$$\begin{cases} \dot{e}_1 = -a(e_1) + a(e_2) + V_1 \\ \dot{e}_2 = b(e_1) + b(e_2) + V_2 \\ \dot{e}_3 = -c(e_3) - e(e_4) + V_3 \\ \dot{e}_4 = f(e_3) - d(e_4) + V_4 \end{cases} \quad (\text{III.11})$$

## Chapitre III : Synchronisation de système hyperchaotique de Qi

Donc système (III.11) à commander est un système linéaire en boucle ouverte qui peut s'écrire sous la forme matricielle:

$$\begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \end{pmatrix} = \begin{pmatrix} -a & a & 0 & 0 \\ b & b & 0 & 0 \\ 0 & 0 & -c & -e \\ 0 & 0 & f & -d \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} + \begin{pmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{pmatrix} \quad (\text{III.12})$$

Pour assurer la synchronisation entre les deux systèmes maître et esclave il faut stabiliser le système (III.11), en convergeant exponentiellement le système (III.8) vers zéro quand  $t \rightarrow \infty$  pour cela on applique la commande par retour d'état. Nous choisisant la constante  $k$  comme matrice à coefficients constants, avec  $V = -ke$

$$k = \begin{pmatrix} 1-a & a & 0 & 0 \\ b & 1+b & 0 & 0 \\ 0 & 0 & 1-c & -e \\ 0 & 0 & f & 1-d \end{pmatrix}$$

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{pmatrix} = - \begin{pmatrix} 1-a & a & 0 & 0 \\ b & 1+a & 0 & 0 \\ 0 & 0 & 1-c & -e \\ 0 & 0 & f & 1-d \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} \quad (\text{III.13})$$

On remplace le système (III.13) dans le système (III.12) précédent et on obtient

$$\begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \end{pmatrix} = \begin{pmatrix} -a & a & 0 & 0 \\ b & b & 0 & 0 \\ 0 & 0 & -c & -e \\ 0 & 0 & f & -d \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1-a & a & 0 & 0 \\ b & 1+b & 0 & 0 \\ 0 & 0 & 1-c & -e \\ 0 & 0 & f & 1-d \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix}$$

On remplace les valeurs de  $a, b, c, d, e, f$ .

$$\begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \end{pmatrix} = \begin{pmatrix} -50 & 50 & 0 & 0 \\ 20 & 20 & 0 & 0 \\ 0 & 0 & -13 & -33 \\ 0 & 0 & 30 & -8 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 49 & 50 & 0 & 0 \\ 20 & 21 & 0 & 0 \\ 0 & 0 & -12 & 33 \\ 0 & 0 & 30 & -7 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix}$$

$$\begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix}$$

Alors

$$\dot{e}_1 = e_1 \Rightarrow \frac{\dot{e}_1}{e_1} = -1; \ln e_1 = -1 \Rightarrow e_1 = e^{-t}.$$

$$\dot{e}_2 = e_2 \Rightarrow \frac{\dot{e}_2}{e_2} = -1; \ln e_2 = -1 \Rightarrow e_2 = e^{-t}. \quad (\text{III.14})$$

$$\dot{e}_3 = e_3 \Rightarrow \frac{\dot{e}_3}{e_3} = -1; \ln e_3 = -1 \Rightarrow e_3 = e^{-t}.$$

$$\dot{e}_4 = e_4 \Rightarrow \frac{\dot{e}_4}{e_4} = -1; \ln e_4 = -1 \Rightarrow e_4 = e^{-t}.$$

Cette matrice présente 04 valeurs propres négatives ; ce qui implique que tous les écarts du système convergent exponentiellement vers zéro en un temps fini. Et donc, la synchronisation de ses deux systèmes maître et esclave est réalisée.

Dans notre mémoire nous avons une réalisation pratique, et comme notre système comporte plusieurs variables alors faire une commande à chaque étage se relève un peu couteuse et en plus notre système comporte plusieurs variables d'état, l'alternative consiste à mesurer qu'un nombre réduit de variable d'état.

Alors il faut opter pour une commande passive afin de réduire le cout de la réalisation, par la suite on adopte une synchronisation unidirectionnel entre les eux systèmes qui on la même architecture électronique afin d'assurer une meilleure synchronisation et permettre à, l'émetteur de piloter le récepteur et résoudre du même coup le problème de non linéarité.

## Chapitre III : Synchronisation de système hyperchaotique de Qi

alors on applique une commande sur un seul étage.

$$\begin{cases} \dot{x}_{1s} = a(x_{2s} - x_{1s}) + x_{2m}x_{3m} \\ \dot{x}_{2s} = b(x_{1s} + x_{2s}) - x_{1m}x_{3m} + U \\ \dot{x}_{3s} = -cx_{3s} - ex_{4s} + x_{1m}x_{2m} \\ \dot{x}_{4s} = -dx_{4s} + fx_{3s} + x_{1m}x_{2m} \end{cases} \quad (\text{III.15})$$

Par soustraction des équations du système maître et celle du système (III.15) on obtient :

$$\begin{cases} \dot{x}_{1s} - \dot{x}_{1m} = -a(x_{1s} - x_{1m}) + a(x_{2s} - x_{2m}) \\ \dot{x}_{2s} - \dot{x}_{2m} = b(x_{1s} - x_{1m}) + b(x_{2s} - x_{2m}) + U \\ \dot{x}_{3s} - \dot{x}_{3m} = -c(x_{3s} - x_{3m}) - e(x_{4s} - x_{4m}) \\ \dot{x}_{4s} - \dot{x}_{4m} = f(x_{3s} - x_{3m}) - d(x_{4s} - x_{4m}) \end{cases} \quad (\text{III.16})$$

Alors :

$$\begin{cases} \dot{e}_1 = -a(e_1) + a(e_2) \\ \dot{e}_2 = b(e_1) + b(e_2) + U \\ \dot{e}_3 = -c(e_3) - e(e_4) \\ \dot{e}_4 = f(e_3) - d(e_4) \end{cases}$$

➤ **La représentation d'état :**

$$\begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \end{pmatrix} = \begin{pmatrix} -a & a & 0 & 0 \\ b & b & 0 & 0 \\ 0 & 0 & -c & -e \\ 0 & 0 & f & -d \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} U$$

Comme  $U = -ke_2$  alors

$$\begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \end{pmatrix} = \begin{pmatrix} -a & a & 0 & 0 \\ b & b & 0 & 0 \\ 0 & 0 & -c & -e \\ 0 & 0 & f & -d \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -k & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix}$$

## Chapitre III : Synchronisation de système hyperchaotique de Qi

On remplace les valeurs de a, b, c, d, e, f et on obtient :

$$\begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \end{pmatrix} = \begin{pmatrix} -50 & 50 & 0 & 0 \\ 20 & 20 & 0 & 0 \\ 0 & 0 & -13 & -33 \\ 0 & 0 & 30 & -8 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -k & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix}$$

Alors :

$$\begin{cases} \dot{e}_1 = -50(e_1) + 50(e_2) \\ \dot{e}_2 = 20(e_1) + 20(e_2) - k(e_2) \\ \dot{e}_3 = -13(e_3) - 33(e_4) \\ \dot{e}_4 = 30(e_3) - 8(e_4) \end{cases} \quad (\text{III.17})$$

➤ **Représentation d'état :**

$$\begin{pmatrix} \dot{e}_{1s} \\ \dot{e}_{2s} \\ \dot{e}_{3s} \\ \dot{e}_{4s} \end{pmatrix} = \begin{pmatrix} -a & a & 0 & 0 \\ b & b-k & 0 & 0 \\ 0 & 0 & -c & -e \\ 0 & 0 & f & -d \end{pmatrix} \begin{pmatrix} e_{1s} \\ e_{2s} \\ e_{3s} \\ e_{4s} \end{pmatrix}$$

$$\dot{x}_s = A' x_s.$$

$$A' = \begin{pmatrix} -50 & 50 & 0 & 0 \\ 20 & 20-k & 0 & 0 \\ 0 & 0 & -13 & -33 \\ 0 & 0 & 30 & -8 \end{pmatrix}$$

D'après le système (III.17) on remarque bien que nous avons construit un observateur avec un gain correctif, alors le problème de synchronisation est devenu celui de la stabilité du système d'erreur, donc nous cherchons la valeur de  $k$  qui permet de stabiliser le système.

## Chapitre III : Synchronisation de système hyperchaotique de Qi

pour garantir la stabilité de système d'erreur, et donc la synchronisation entre l'émetteur et le récepteur peut être réalisé.

D'abord, on doit déterminer les polynômes caractéristiques du système.

$$\Delta(A) = \det(sI - A').$$

$$\Delta(A) = \begin{pmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & s \end{pmatrix} - \begin{pmatrix} -50 & 50 & 0 & 0 \\ 20 & 20-k & 0 & 0 \\ 0 & 0 & -13 & -33 \\ 0 & 0 & 30 & -8 \end{pmatrix}$$

$$\Delta(A) = \begin{pmatrix} s+50 & -50 & 0 & 0 \\ -20 & s-20+k & 0 & 0 \\ 0 & 0 & s+13 & 33 \\ 0 & 0 & -30 & s+8 \end{pmatrix}$$

$$\det(sI - A') = s^4 + (51 + k)s^3 + (71k - 267)s^2 + (2144k - 9180)s + 54700k - 218800.$$

Pour déterminer la valeur de  $k$  on applique le critère de Routh.

$s^4$	1	71-276	-21800+54700k
$s^3$	51+k	2144k-9180	
$s^2$	$\alpha$	-21800+54700k	
$s^1$	$\beta$		
$s^0$			

$$\text{Avec } \begin{cases} \alpha = \frac{(71-276)(51+k) - (2144k-9180)}{51+k} \\ \beta = \frac{(2144k-9180)\alpha - (51+k)(-21800+54700)}{\alpha} \end{cases}$$

## Chapitre III : Synchronisation de système hyperchaotique de Qi

Pour que ce système soit stable :

$$\left\{ \begin{array}{l} 51+k > 0 \\ \alpha > 0 \\ \beta > 0 \end{array} \right. \Rightarrow k > -51 \Rightarrow k \in [-51, \infty[$$

### III.3.3. Simulation sous Matlab :

On insère les équations du récepteur dans Matlab, on additionne un message (signal sinusoïdale) dans l'émetteur dans l'un de ces étages et on observe le comportement de chaque comportement de chaque étage du récepteur.

Une simple opération de soustraction entre  $x_{3m}$  et  $x_{3s}$  permet de récupérer le message, (le message sera ajouté au niveau de  $x_{3m}$ ).

La simulation se fera pour  $a=50$ ,  $b=20$ ,  $c=13$ ,  $d=8$ ,  $e=33$ ,  $k=100$ .

#### a) Visualisation des états estimés :

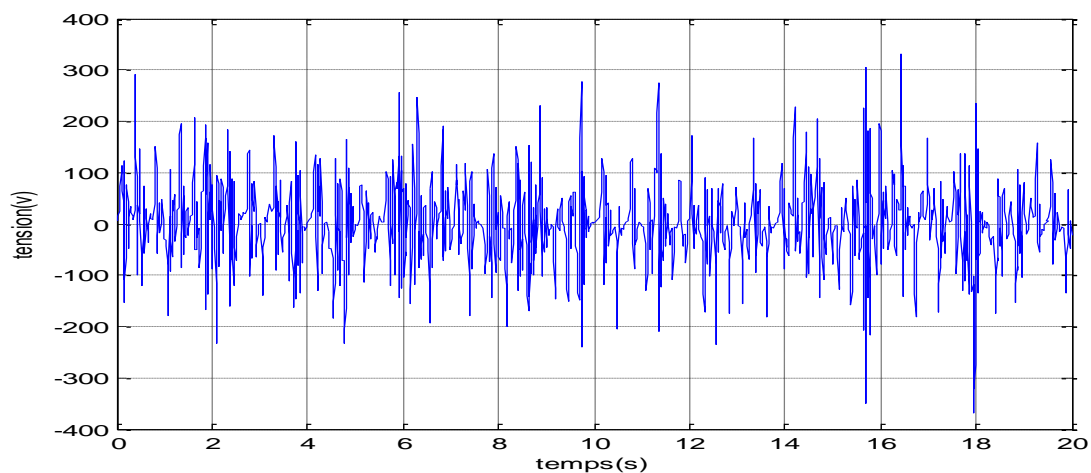


Figure (III.11): Graphe d'état  $x_{1s}(t)$

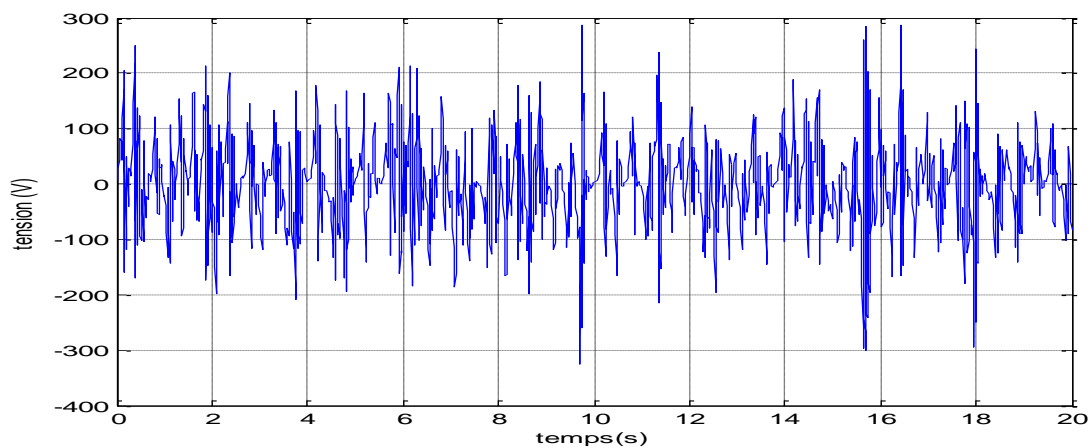
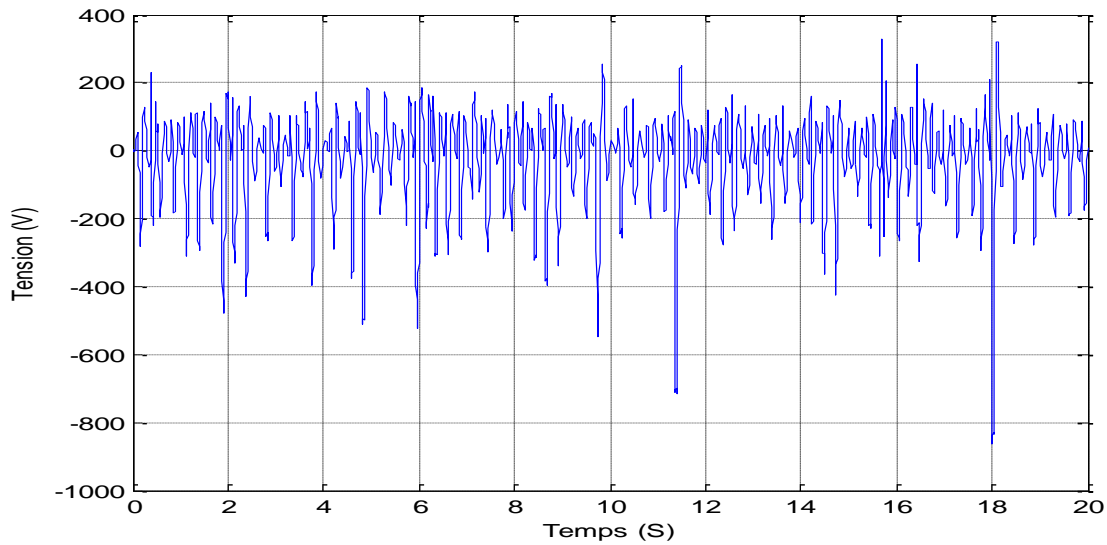
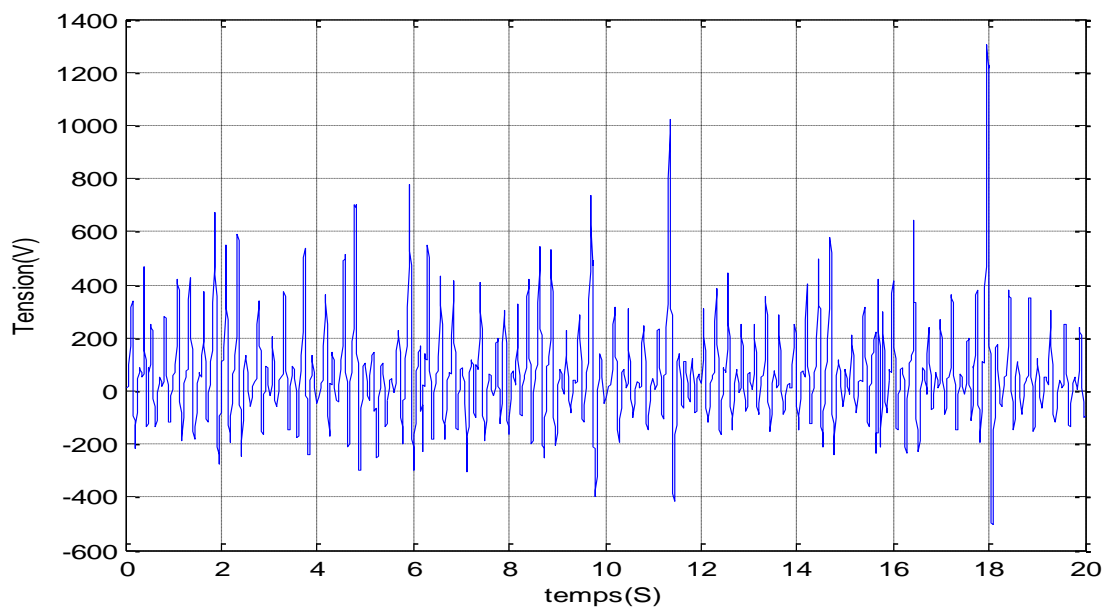


Figure (III.12): Graphe d'état  $x_{2s}(t)$ .



*Figure (III.13) : Graphe d'état  $x_{3s}(t)$ .*



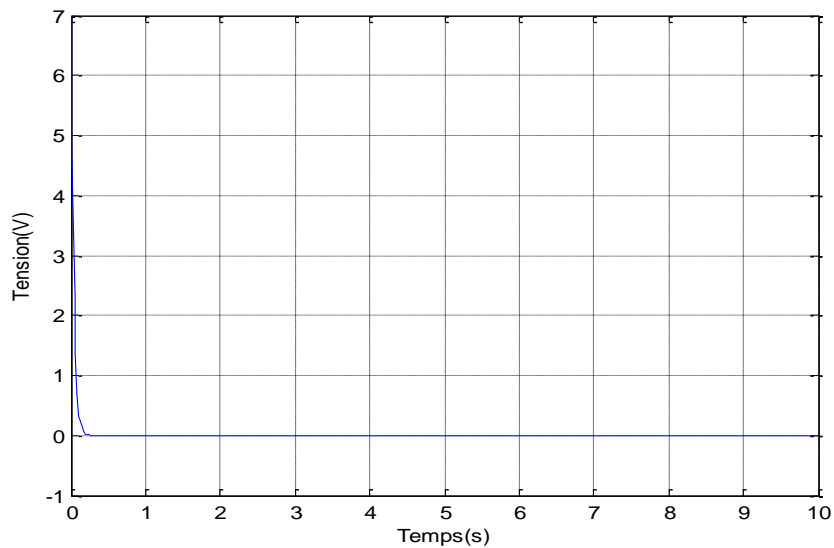
*Figure (III.14): Graphe d'état  $x_{4s}(t)$ .*

❖ **Observation :**

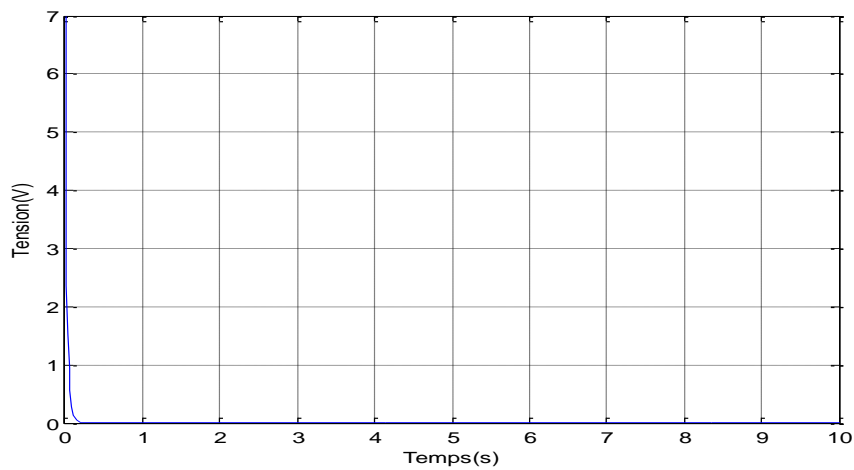
Nous remarquons la présence d'oscillations irrégulières et aperiodiques dans ces figures, donc les signaux  $x_{1s}$ ,  $x_{2s}$ ,  $x_{3s}$  et  $x_{4s}$  sont des signaux chaotiques.

b) Visualisation des écarts entre les états des deux systèmes :

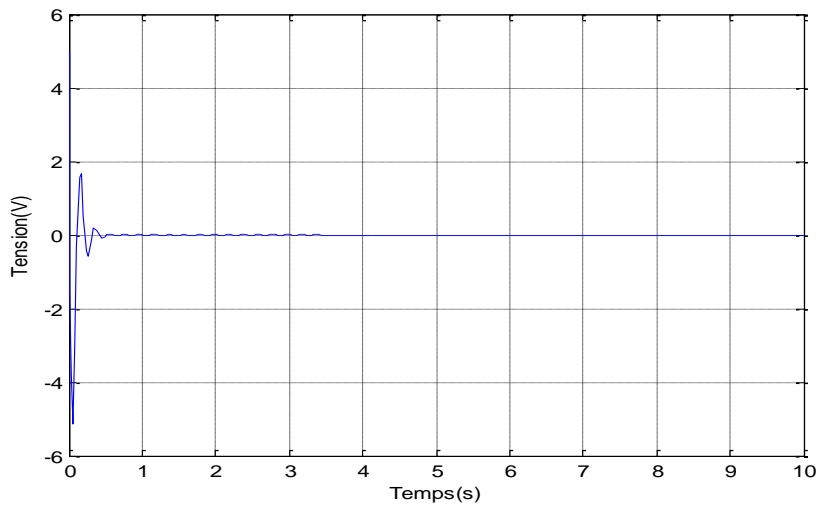
- Graphe de l'écart d'erreur entre  $x_{1m}(t)$  et  $x_{1s}(t)$



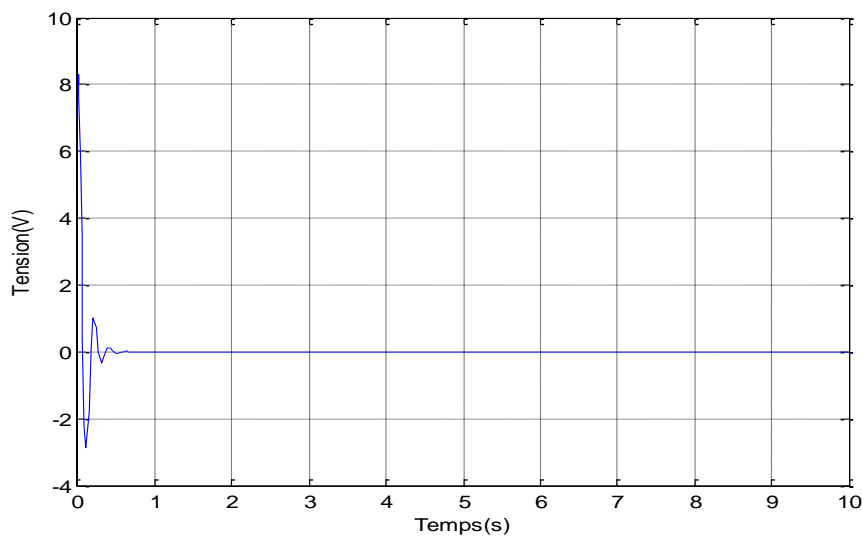
- Graphe de l'écart d'erreur entre  $x_{2m}(t)$  et  $x_{2s}(t)$  :



- Graphe de l'écart d'erreur entre  $x_{3m}(t)$  et  $x_{3s}(t)$  :



- Graphe de l'écart d'erreur entre  $x_{4m}(t)$  et  $x_{4s}(t)$  :



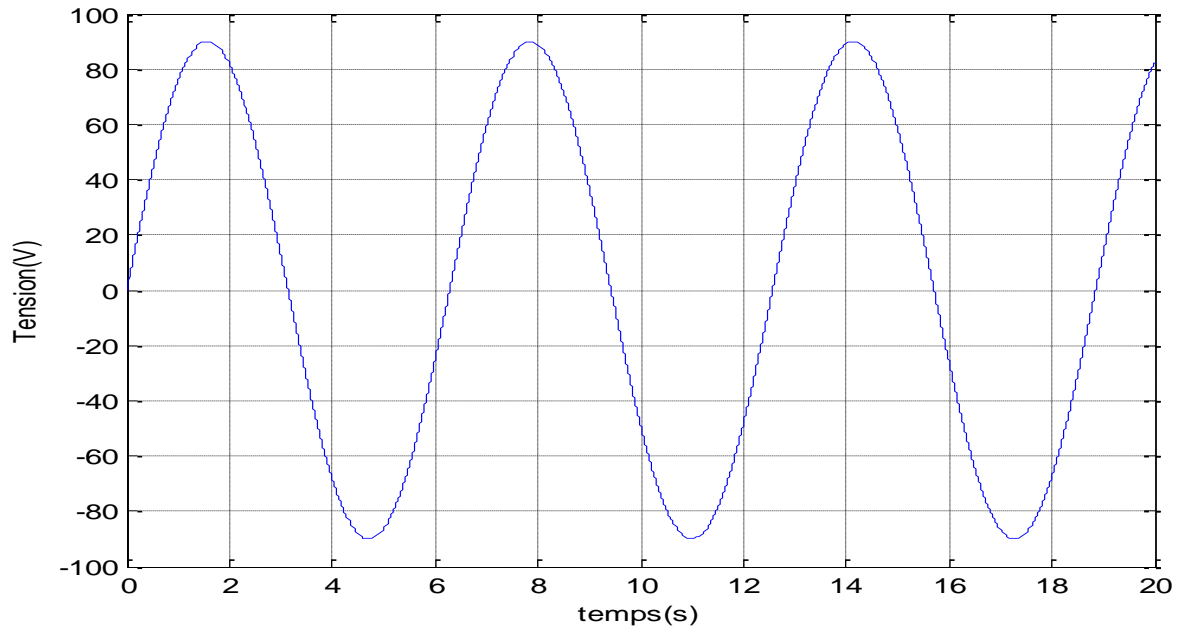
*Figure (III.15): Allures des erreurs entre les états des deux systèmes.*

❖ **Observation :**

La figure (III.15), nous montre la convergence des erreurs d'état vers zéro quant  $t$  tend vers l'infini, ce qui démontre la synchronisation des deux systèmes.

### C) Récupération du message :

Pour récupérer le message on fait une soustraction entre les états  $x_3$  et  $x_{3s}$ .



Figure(III.16) : Comparaison entre le message et la sortie de récepteur.

#### ❖ Observation :

La figure (III.16) montre bien que le message envoyé a bien été récupéré. Ce qui prouve l'efficacité de l'approche.

### III.3.4. Conclusion :

Dans ce chapitre nous avons synchronisé deux systèmes hyperchaotiques de Qi à l'aide d'un observateur, afin de réaliser un émetteur et un récepteur de signaux chaotiques pour une transmission sécurisée de données, en y ajoutant, par addition, le message (signal sinusoïdal) dans le signal chaotique généré par l'émetteur. La récupération du message s'est faite par soustraction de signal émis au signal reçu. Les résultats obtenus par Matlab prouvent la possibilité du procédé et montrent que la récupération du message est possible.

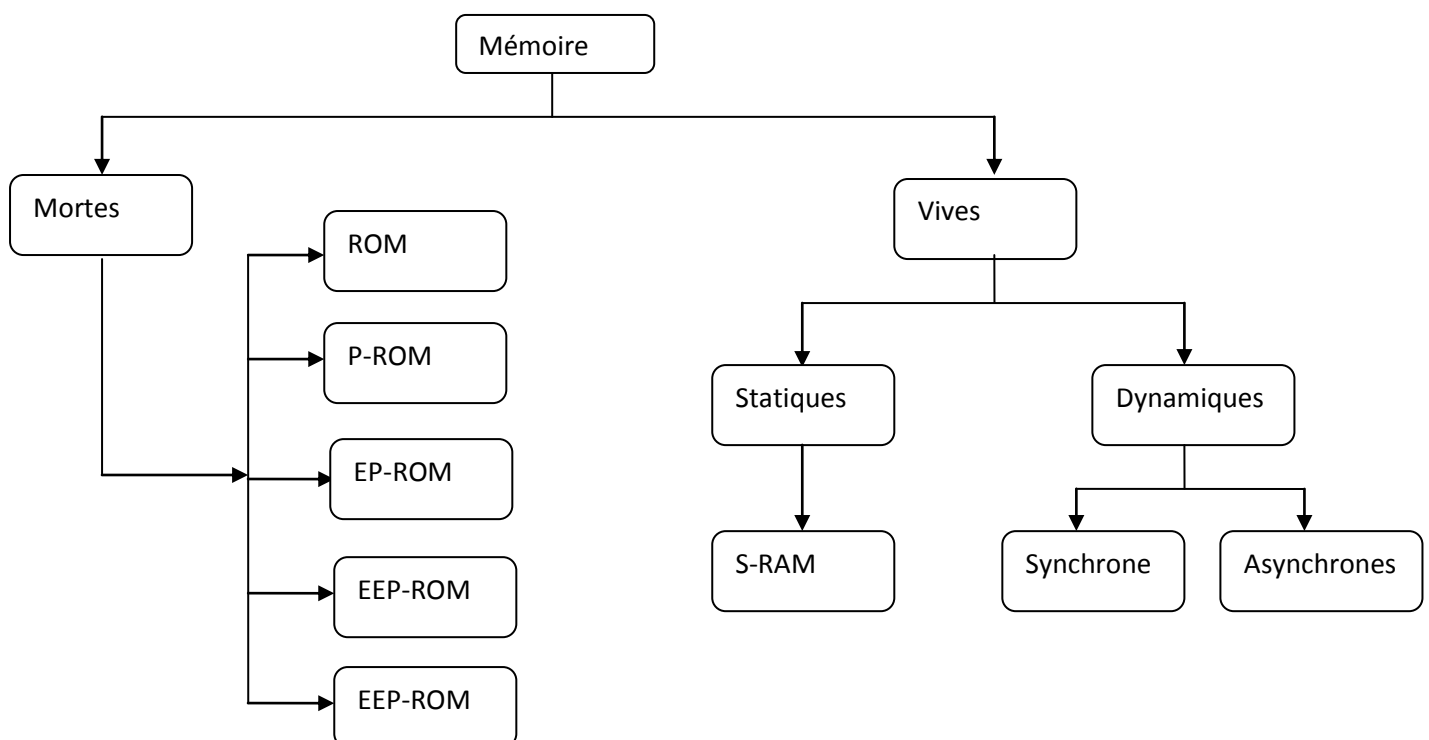
Dans un chapitre suivant nous allons réaliser ce système sur la carte FPGA.

# Chapitre VI

### VI. Introduction :

Il y a de cela quelques années, la réalisation des montages électroniques impliquait l'utilisation d'un nombre important de circuits intégrés logiques, ce qui avait pour conséquence une mise en œuvre complexe, un prix de revient important et un circuit imprimé de taille. Pour diminuer et minimiser ces circonstances, ces dernières années ont vu l'application d'un type de circuit intégrés à savoir, les circuits logiques programmables (PLDs : Programmable Logic Device). Ces derniers ont pour origine le développement des mémoires utilisées en informatique. Ce type de produit est caractérisé par sa simplicité de mise en œuvre et sa capacité d'intégrer, dans un seul circuit, plusieurs fonctions logiques programmables.

### IV.2. Les techniques de mémorisation [19] :



**Figure(VI.1) : Les différents types de mémoires.**

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

**Les ROM (Read Only Memory) :** Mémoires figées par concepteur à lecture seule et non modifiables.

**Les P-Rom (Programmable Read Only Memory) :** Mémoire programmable une fois par l'utilisateur.

**Les EP-ROM (Erasable Programmable Read Only Memory) :** Mémoires programmables électriquement et effaçables par des rayons-violet au bout d'un certain temps (quelques minutes).

## Chapitre IV: Présentation des FPGAs

**Les EEP-ROM** (*Electrically Erasable Programmable Read Only Memory*): Mémoires programmables électriquement à lecture seul, effaçables électriquement (quelques millisecondes).

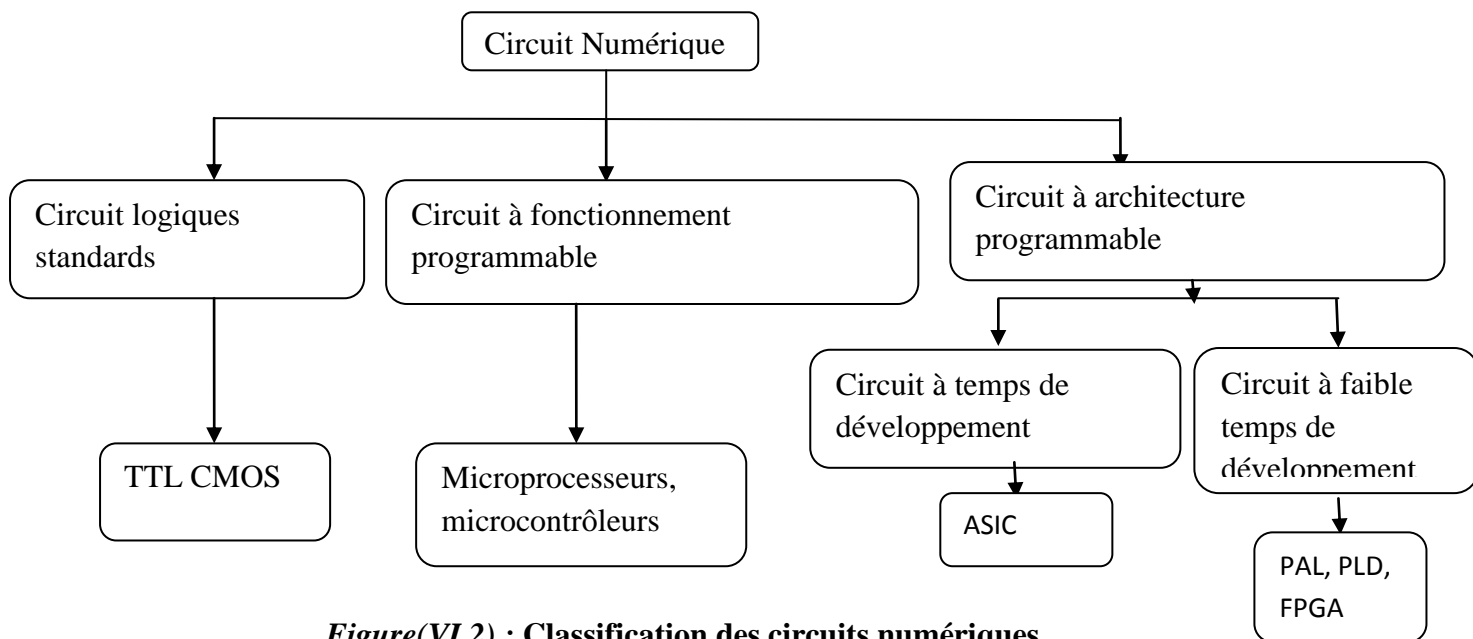
**Les FLASHs**: Elles sont une version plus évoluée des EEP-ROM avec avantage d'être plus faciles à programmer et à effacer.

**Les S-RAM** (*Static Random Memory*): Mémoires volatiles avec cellule de base à plusieurs transistors (accès rapide, consommation plus coûteux).

**Les RAM dynamiques**: Mémoires volatiles qui nécessitent rafraîchissement périodique de l'information afin de la conserver avec cellules de base à un transistor.

### VI.3. Les circuits logiques programmables:[17]

De nombreuses familles de circuits programmables et reprogrammables sont apparues depuis les années 70 avec des noms très divers suivant les constructeurs. **La figure VI.2** donne une classification possible des circuits numériques en précisant où se situe chaque type de ces circuits dans cette classification.



**Figure(VI.2) : Classification des circuits numériques.**

La technologie de la logique câblée permet de réaliser des circuits de faible complexité à un coût réduit, en interconnectant des circuits logiques classiques TTL (série 74xx) ou CMOS (série 4xxx). Lorsque la complexité sera plus importante, on préférera utiliser des circuits logiques programmables : FPLDs du type PLD, FPGA ou CPLD. Ces technologies permettent de réaliser des circuits logiques en configurant des matrices de portes logiques et de bascules.

### 1. Les ASICs (Application Specific Integrated Circuit): [2]

Par définition, les circuits ASIC regroupent tous les circuits dont la fonction peut être *personnalisée* d'une manière ou d'une autre en vue d'une application spécifique, par opposition aux circuits standards dont la fonction est définie et parfaitement décrite dans le catalogue des composants.

Les ASIC peuvent être classés en plusieurs catégories selon leur niveau d'intégration. En fait, un ASIC est défini par sa structure de base (réseau programmable, cellule de base, matrice, etc.).

D'une manière générale, l'utilisation d'un ASIC conduit à de nombreux avantages provenant essentiellement de la réduction de la taille des systèmes. Il en ressort, une réduction du nombre des composants sur le circuit imprimé. La consommation et l'encombrement s'en trouvent considérablement réduits.

Le concept ASIC par définition assure une optimisation maximale du circuit à réaliser. Nous disposons alors d'un circuit intégré correspondant réellement à nos propres besoins. La personnalisation du circuit donne une confidentialité au concepteur et une protection industrielle. Enfin, ce type de composant augmente la complexité du circuit, sa vitesse de fonctionnement et sa fiabilité.

Dans l'approche des circuits du type ASIC, l'inconvénient majeur réside dans le fait du passage obligatoire chez le fondeur ce qui implique des frais et un temps de développement élevés du circuit.

Les ASICs sont généralement plus convenables pour les productions en série de Conceptions déjà vérifiées et non pour les prototypes.

#### IV.4. Les circuits logiques programmables [7]:

Les circuits logiques programmables et reprogrammables architecturalement sont classifiés en trois grandes familles les PLD, CPLD, et FPGA. Le diagramme suivant illustre les différents types suivant la technologie utilisée.

##### ❖ les différents types de circuits logiques programmables :

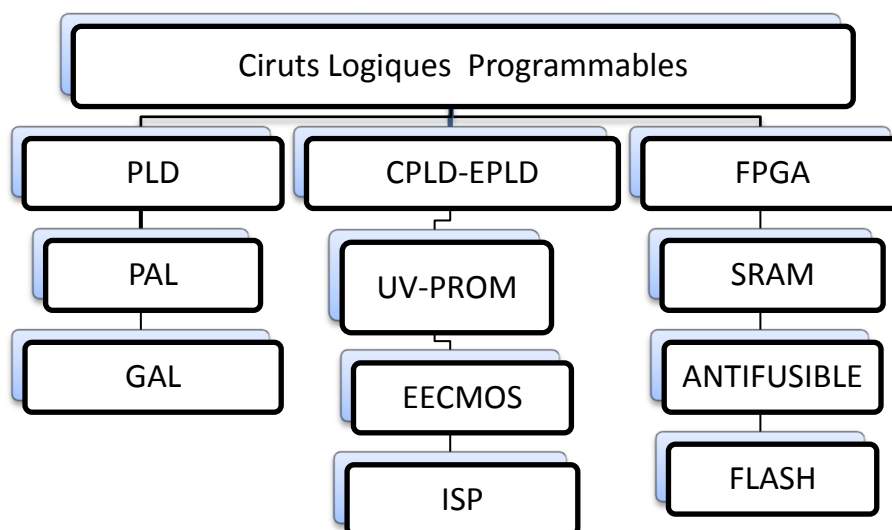


Figure (IV.3) : le diagramme de différents types de circuits logiques programmables.

## Chapitre IV: Présentation des FPGAs

**1-Les PLD (programmable logic Device) :** famille des circuits programmables qui comprend les PAL et les GAL.

- PAL (programmable Array Logic) : circuits logiques programmables dans lesquels seules les fonctions ET sont programmables, les fonctions OU ne le sont pas.
- GAL (Generic Array logic) : circuits logiques reprogrammables à technologie CMOS.
- **Structure de base d'un PLD :**

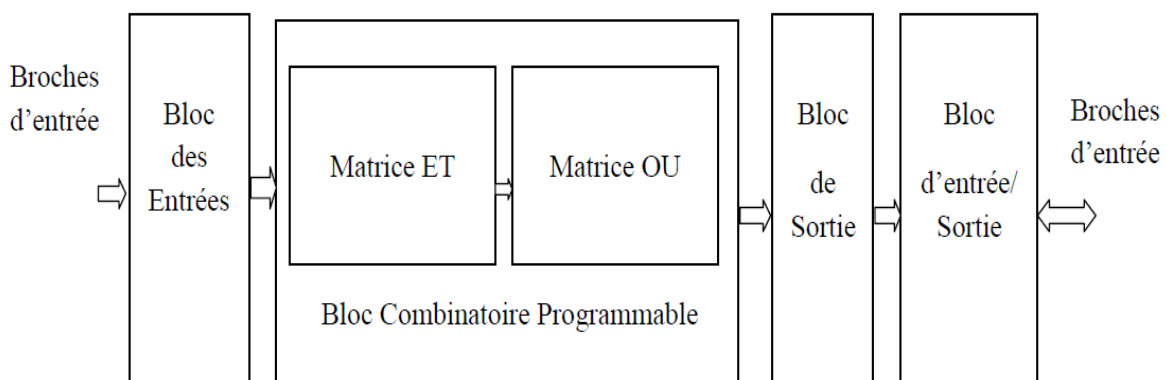
La plupart des circuits PLDs suivent la structure suivante (**Figure (IV.4)**) :

- Un bloc d'entrée qui permet de fournir au bloc combinatoire l'état de chaque entrée qui est son complément.
- Un ensemble d'opérateurs « ET » sur lesquels viennent se connecter, les variables d'entrées et leurs compléments.
- Un ensemble d'opérateur « OU » sur lequel les sorties des opérateurs « ET » sont connectées.
- Un bloc de sortie.
- Un bloc d'entrée-sortie, qui comporte une porte à 3 états et une broche d'entrée-sortie.

Le deuxième et le troisième ensemble forment chacun ce qu'on appelle une matrice. Les interconnexions de ces matrices doivent être programmables, et ceci est réalisé par des fusibles qui sont grillés lors de la programmation. Lorsqu'un PLD est vierge, toutes les connexions sont assurées.

Le bloc de sortie est souvent appelé macro-cellule que l'on nomme OLMC (abréviation anglaise de output Logic Macro Cell signifiant macro-cellule logique de sortie). Cette macro-cellule comporte :

- Une porte OU Exclusif, une bascule D.
- Des multiplexeurs qui permettent de définir différentes configurations et un dispositif de rebouclage sur la matrice ET.
- Des fusibles de configuration (dans les FPGAs on utilise plutôt des cellules de commande des points de connexion).

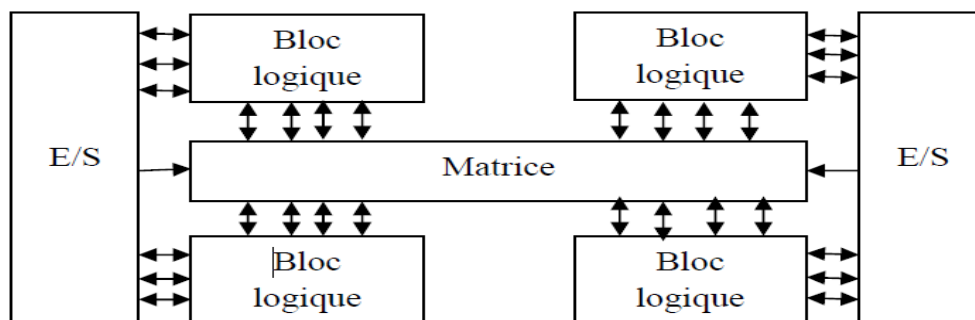


**Figure (VI.4) : La structure de base d'une PLDs.**

**2-CPLD ou EPLD (Erasable Programmable Logic Device)**: circuits logiques reprogrammables, qui sont des circuits effaçables par rayons ultraviolets, ils peuvent être reprogrammés.

- **ISP (In System Programmable)**: Circuits que l'on peut programmer même lorsqu'il est en place sur l'application.

Les CPLD sont des circuits composés de plusieurs PALs élémentaires reliés entre eux par une zone d'interconnexion (matrice d'interconnexion) [28] [29]. Sa physionomie est généralement très structurée. Un certain nombre de macro-cellules de base sont regroupées pour former des blocs logiques. Grâce à leurs structures, ils peuvent atteindre des vitesses de fonctionnement élevées (plusieurs centaines de MHz) [31] [32]. Ces circuits ont une capacité en nombre de portes et en possibilités de configuration très supérieure à celle de PALs.



*Figure (VI.5) : Physionomie d'un CPLD.*

**3-FPGA (Field Programmable Gate Array)**: Ces circuits sont une évolution des CPLD.

### **IV.5. Présentation de FPGAs [6] :**

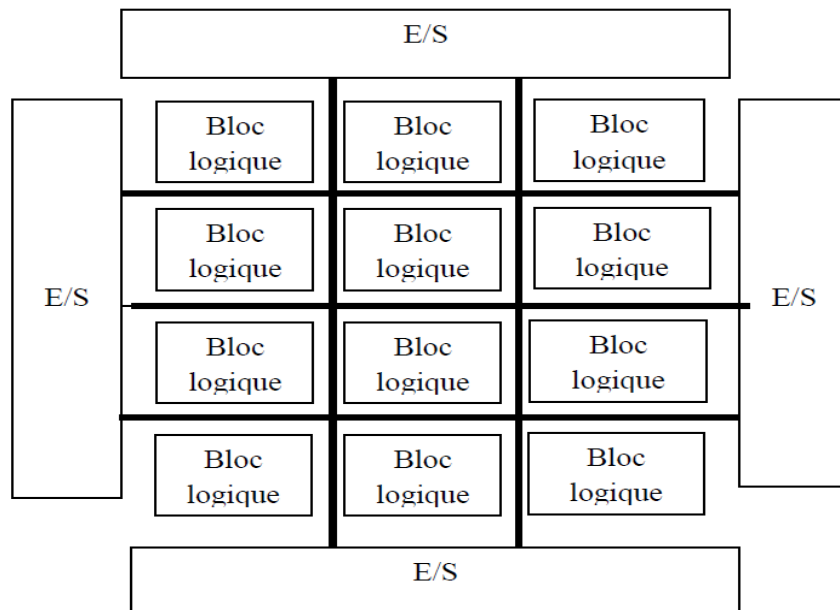
Les circuits FPGAs sont des composants entièrement reconfigurables ce qui permet de les reprogrammer à volonté afin d'accélérer notablement certaines phases de calculs. Ils se composent d'une matrice de blocs logiques programmables entourée de blocs d'entrées/sorties et l'ensemble est relié par un réseau d'interconnexion programmable. Ces circuits comparés aux CPLDs utilisent des modules logiques beaucoup plus réduits, mais beaucoup plus nombreux.

Les interconnexions de ces modules ne sont pas centralisées comme les CPLDs. La physionomie du réseau de routage est vue comme une multitude de segments métalliques (lignes) pouvant être reliés entre eux ou connectés en entrée ou en sortie des blocs logiques. On distingue plusieurs types de lignes définies par leur longueur relative, on trouve :

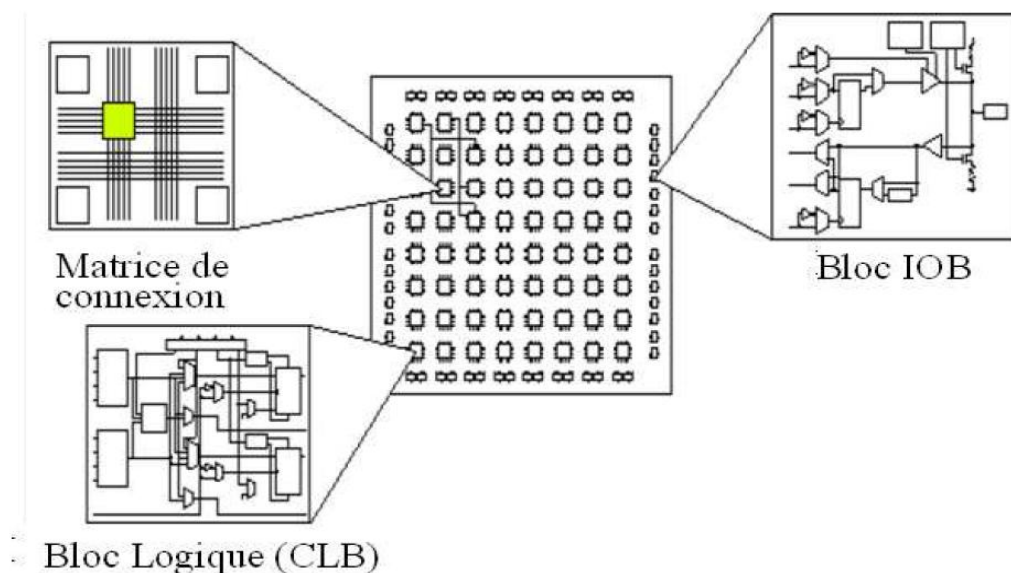
- Les interconnexions à usage général qui sont composées de segments verticaux et horizontaux qui entourent chaque **bloc logique configurable** CLB et qui peuvent être reliés entre eux par une matrice de commutation.
- Les lignes directes fournissant des chemins entre les CLB adjacents et entre les CLB et les cellules d'entrée-sorties.
- Les lignes longues qui sont des lignes verticales et horizontales qui n'utilisent pas de matrice de commutation. Elles parcourent toutes les zones d'interconnexion. Elles sont utilisées pour véhiculer les signaux qui doivent parcourir de long trajet. Ces lignes conviennent pour véhiculer les signaux d'horloge.

## Chapitre IV: Présentation des FPGAs

L'ensemble de points de connexion est appelé PIP (abréviation anglaise de Programmable Interconnect Points) et chaque point de connexion peut être réalisé selon deux techniques définissant deux classes de FPGAs (FPGAs à SRAM et FPGAs à ANTI-FUSIBLE).



*Figure (IV.6) : physionomie d'un FPGA.*



*Figure (IV.7) : Structure générale d'une FPGA.*

### IV.6. Principaux fondateurs de FPGA:

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

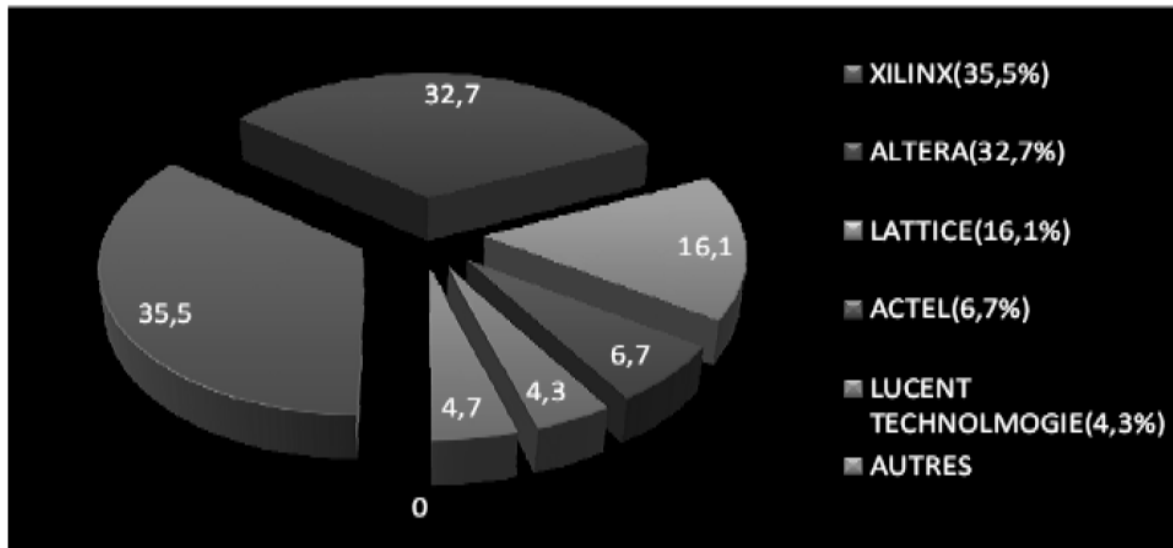


Figure (VI.8) : Statistique de marché occupée par les vendeurs des FPGAs.

### IV.7. Classification des FPGAs [26] :

Selon la technique de réalisation des PIP (point de connexion), on distingue trois technologies d'FPGA :

- 1-Technologie de programmation par RAM.
- 2-Technologie de programmation par EEPROM ou FLASH.
- 3-Technologie de programmation par ANTI-FUSIBLE.

➤ **Technologie à base de RAM (XILINIX et ALTERA) :**

Cette technologie permet d'avoir une reconfiguration rapide des FPGAs, les points de connexions sont des ensembles de transistores commandés.

L'inconvénient majeur de cette technologie c'est qu'elle nécessite beaucoup de place et il est nécessaire de sauvegarder le design du FPGA dans un autre mémoire flash.

➤ **Technologie à base d'EEPROM ou FLASH (LATTICE et ACTEL) :**

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

### ➤ *Technologie à base d'ANTI-FUSIBLE (ACTEL) :*

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

### *IV.8. Critères du choix des FPGAs [2]:*

Les FPGAs sont développés récemment grâce aux progrès de la technologie VLSI. L'application de ce type de circuits est une évolution des systèmes ouvrant des perspectives de traitement numérique inaccessibles auparavant. La fin des années 80 a vu l'apparition des premiers circuits FPGAs qui sont des circuits intégrés que l'on peut configurer en un temps relativement court pour réaliser n'importe quelle fonction logique « câblée » à bas coût par une programmation de ses cellules logiques et ses interconnexions avec une restriction de ne pas épuiser les ressources du FPGA. Typiquement, un circuit FPGA haute densité peut contenir jusqu'à plusieurs millions d'éléments programmables.

Pour réussir une application à base d'FPGA et afin d'obtenir un système plus performant, consommant un minimum de puissance, il est nécessaire de respecter un certain nombre de règles comme :

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

### *IV.II.7. Différents domaines d'applications des FPGAs :*

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

Parmi ces applications nous citons :

- 1-Informatique : périphériques spécialisés.
- 2-Machinerie industrielle : Contrôleur pour machines.
- 3-Télécommunication : Traitement d'image, filtrage.

## Chapitre IV: Présentation des FPGAs

3-Instrumentation : Équipement médical, Prototypage.

5-Transport : Contrôle d'avions et métros.

7-Militaire : Radar, communication protégée, la détection au la surveillance.

### IV.9. Les familles architecturales des FPGAs [17] :

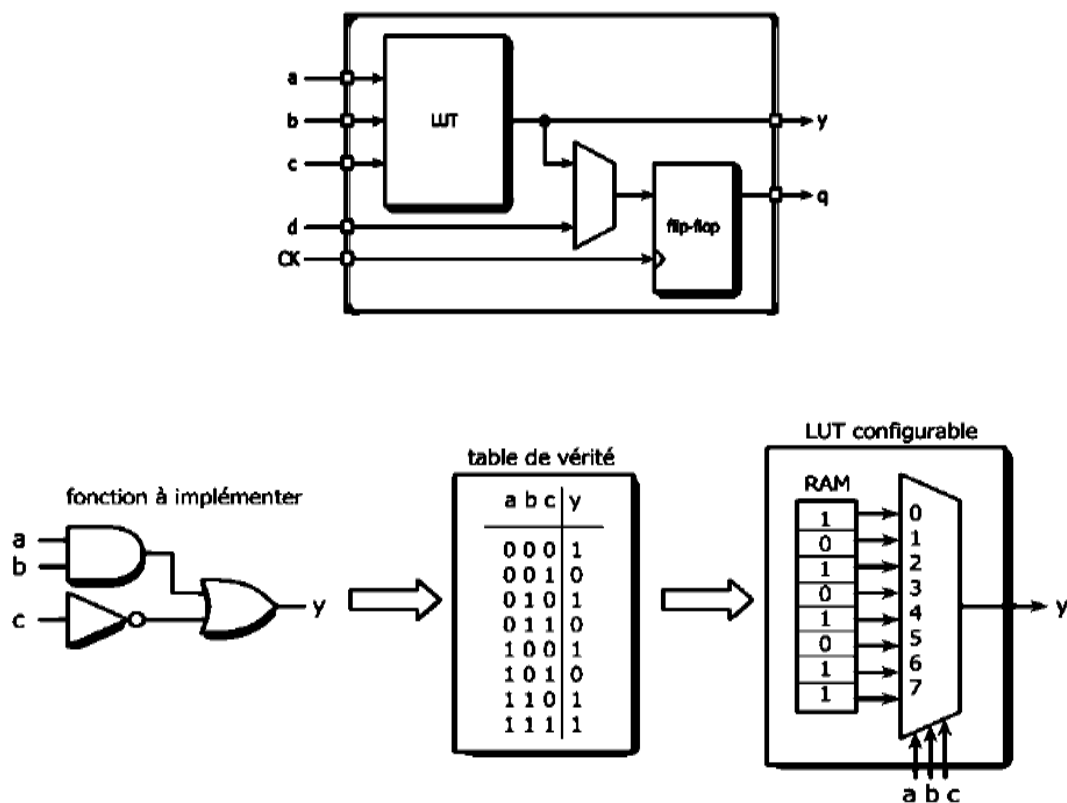
Les familles des FPGAs peuvent se regrouper en deux groupes :

- *Les Circuits FPGAs à base de « LUT » (Look Up Tables).*
- *Les circuits FPGAs à base de Multiplexeurs « MUX ».*

#### 1) Les Circuits FPGAs à base de « LUT » (Look Up Tables) :

Les LUT (Look Up Tables) ressemblent aux tables de vérité des fonctions logiques et réalisables par des mémoires de type SRAM. Aujourd'hui la structure la plus utilisée est basée sur ce type (Look Up Tables) d'FPGA. Des possibilités offertes par les circuits programmables FPGA à SRAM permettent par ailleurs de mettre en œuvre le concept de prototypage (ou maquette) pour la vérification fonctionnelle de systèmes sur puce pour certaines applications.


La fonction de LUT est de stocker la table de vérité de la fonction combinatoire à implémenter comme le montre la figure suivante :

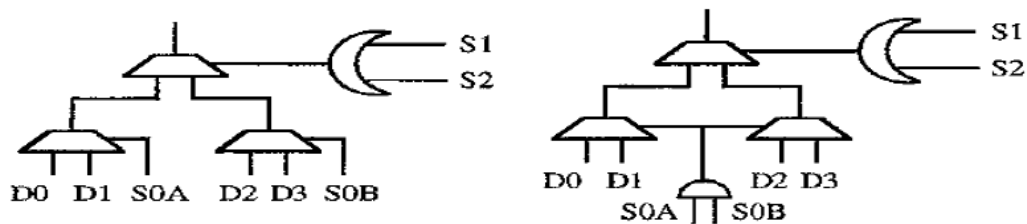


Figure(VI.9) : Exemple d'implémentation sur LUT.

### 2) Les circuits FPGAs à base de Multiplexeurs « MUX »:

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

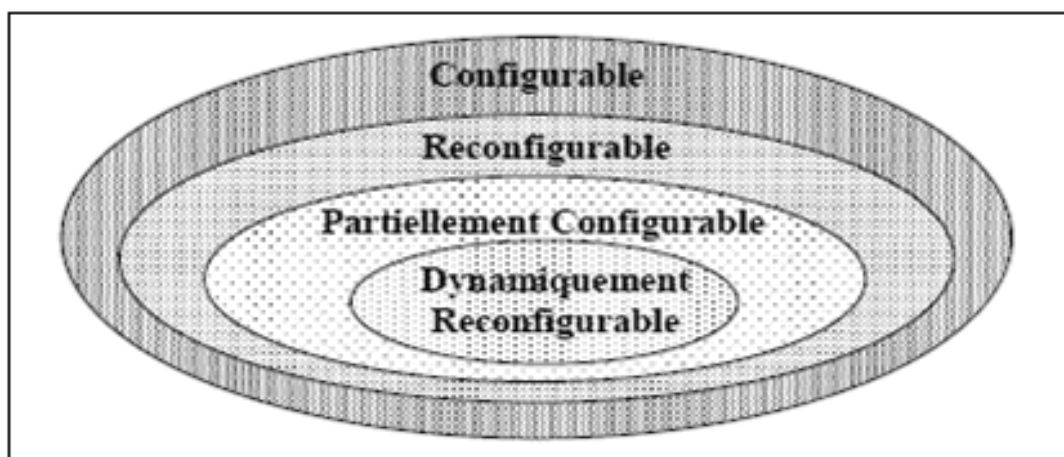
<i>Equation logique</i>	<i>Symbole</i>
$F = (a \text{ AND } b) \text{ OR } (\bar{a} \text{ AND } b)$	



*Figure (VI.10) : Exemple d'implémentation sur des multiplexeurs.*

### VI.10. Programmation et configuration des circuits FPGAs [20] :

La configuration est le processus qui permet de charger des données spécifiques à une conception dans un ou plusieurs FPGAs pour définir l'opération fonctionnelle des blocs internes ainsi que leur interconnexion. Et leurs temps de configuration dépendent du mode de configuration sélectionnée. Dans la littérature on peut distinguer 4 modes de configuration et programmation des circuits FPGAs comme le montre **la figure (VI.8)**.



*Figure (VI.11) : Classification des circuits FPGAs selon leurs configurations.*

#### ➤ **Circuit configurable :**

Un circuit configurable est un circuit programmé et chargé par différentes données, où les interconnexions d'un FPGA sont programmées afin de donner un fonctionnement spécifique à un tel circuit.

### ➤ **Circuit reconfigurable :**

C'est le même principe de configuration, sauf que cette fois on reconfigure le circuit FPGA une deuxième fois pour l'utiliser dans une autre fonction comme on peut garder la dernière configuration du circuit. On peut même effacer cette configuration et on reconfigure le circuit une nouvelle fois.

### ➤ **Circuit partiellement reconfigurable :**

Un dispositif ou un circuit est défini comme partiellement reconfigurable (dans la littérature on trouve aussi la terminologie Run Time Reconfiguration RTR globale) s'il est possible de le reconfigurer sélectivement, tandis que l'état de repos du reste du dispositif est inactif, mais il conserve son information configurée. Encore, il ne semble pas y avoir n'importe quel dispositif sur le marché qui soit partiellement reconfigurable, mais non aussi dynamiquement reconfigurable, La reconfiguration partielle permet de rendre un FPGA effectif, multiple fonctions, et change de fonctions pendant le fonctionnement du système.

### ➤ **Circuit dynamiquement reconfigurable :**

Un circuit FPGA est reconfigurable dynamiquement (dans la littérature on trouve la terminologie Run Time Reconfiguration RTR locale) s'il peut être partiellement reconfiguré durant son fonctionnement, c.-à-d. une partie du circuit correspondant à certaines fonctions logique et leur interconnexions peut être changée sans affecter le fonctionnement de la logique restante. On peut aussi parler de reconfiguration dynamique dans le cas où plusieurs circuits FPGAs sont connectés entre eux et il s'agit de reconfigurer un seul composant FPGA tout en maintenant les autres circuits en fonctionnement.

### ***IV.11. Avantages et Inconvénients des FPGAs :***

Les avantages et les inconvénients des FPGAs sont multiples. On trouve :

Avantages	Inconvénients
1- Technologie facile à maîtriser. 2- Temps de développement réduit. 3- Reprogrammable. 4- Coût peu élevé. 5- Flexibilité et possibilité de réduire fortement les délais de développement et de commercialisation.	1- Performances non optimisées. 2- Temps de réponse long par rapport aux ASIC.

### ***Remarque :***

Dans notre travail nous utilisons la carte **FPGA ALTERA DE2**.

### IV.12. Architecteur de la carte FPGA DE2 [9] :

La figure (VI.12) : représente l'architecture de la carte FPGA.

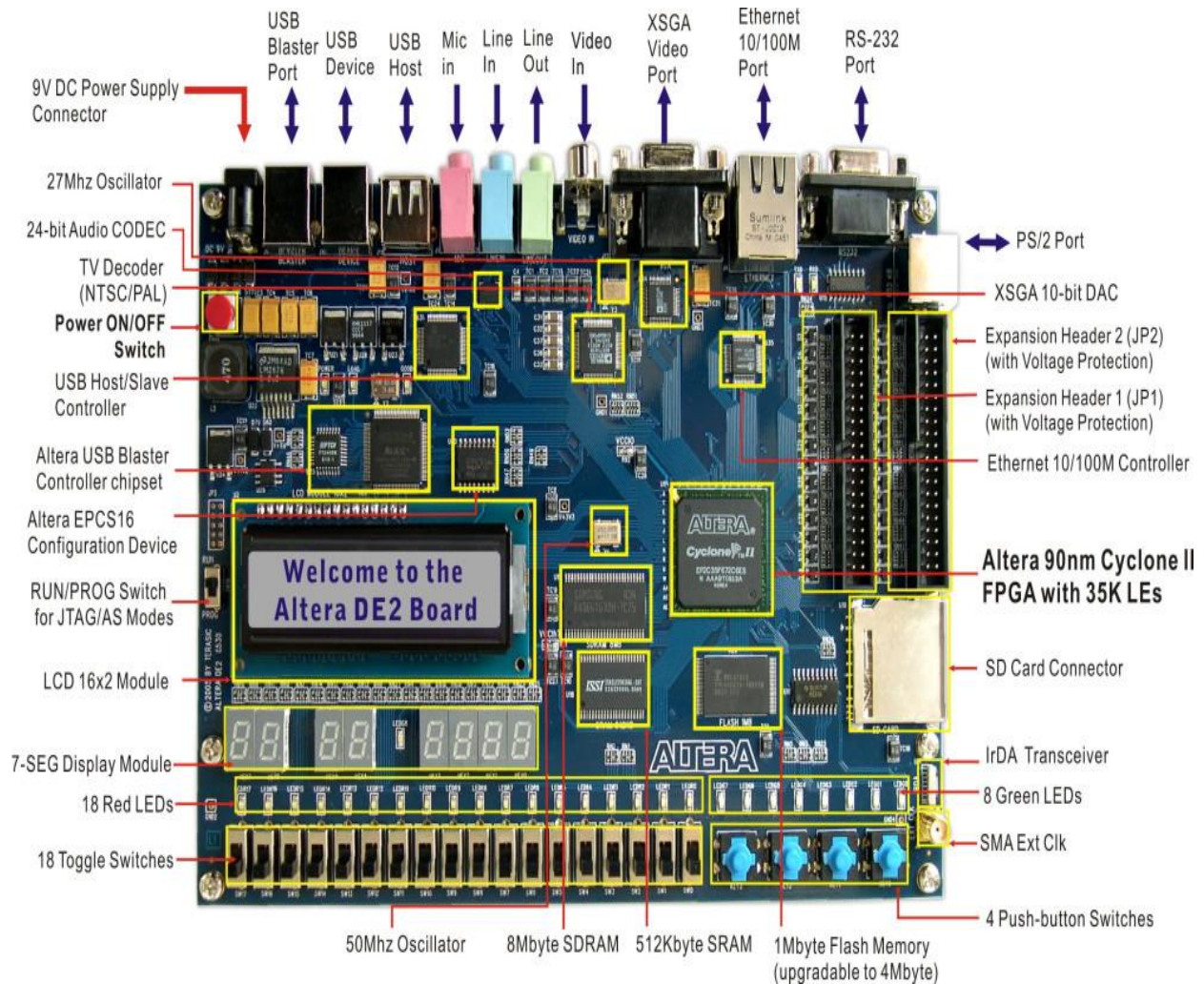


Figure (VI.12) : La carte DE2.

Cette carte dispose des éléments suivants :

- Un FPGA Altera Cyclone II 2C35
- Un circuit de mémoire Flash série Altera EPCS16
- Une interface USB-Blaster pour la programmation et le contrôle des API utilisateur, les modes de programmation JTAG et Active Serial (AS) sont supportés
- 512-Kbyte de SRAM
- 8-Mbyte de SDRAM
- 4-Mbyte de mémoire Flash (1 Mbyte sur certaines cartes)

- Un connecteur SD Card
- 4 boutons poussoirs
- 18 interrupteurs
- 18 LEDs utilisateur rouges
- 9 LEDs utilisateur vertes
- Un oscillateur 50-MHz et un oscillateur 27-MHz comme sources d'horloges
- Un CODEC 24-bit CD-quality avec jacks entrée ligne, sortie ligne et microphone-in.
- Un CNA VGA (10-bit high-speed triple DACs) avec connecteur VGA-out
- Un décodeur TV (NTSC/PAL) avec connecteur TV-in
- Un contrôleur 10/100 Ethernet avec connecteur
- Un contrôleur USB Hôte/Esclave avec connecteurs USB type A et type B
- Un E/R RS-232 avec connecteur 9-broches
- Un connecteur PS/2 souris/clavier
- Un E/R IrDA
- 2 connecteurs d'extension 40-broches avec diodes de protection.

### ***IV.12. Conclusion:***

Cette partie est portée sur les généralités sur les circuits logiques programmables où on s'intéresse plus particulièrement au FPGA avec ses propres caractéristiques.

# Chapitre V

# Chapitre V : Réalisation

## V.1.Introduction :

La conception théorique de notre transmission étudiée au chapitre précédent nous offre la possibilité de la réaliser en pratique à l'aide d'une carte électronique FPGA.

Ce chapitre est consacré à la réalisation de notre transmission, la première partie est consacrée à la présentation générale de logiciel DSP-Builder, en suite nous discrétisons le système de Qi sous la forme numérique. Aussi nous allons simuler notre système sur un logiciel de simulation, enfin nous passerons à la visualisation sur la carte électronique (FPGA).

## V.2. Présentation de DSP-Builder :

La bibliothèque DSP-Builder pour Simulink est une bibliothèque ajoutée de Simulink qui fournit les blocs spéciaux pour les systèmes de contrôle DSP-Builder. Ces blocs permettent la programmation facile des systèmes DSP-Builder en employant MATLAB/Simulink

Le DSP Builder d'Altera raccourcit des cycles de conception de traitement numérique du signal (DSP) et aussi intègre le développement d'algorithmes, la simulation, et les possibilités de vérification les outils de MathWorks MATLAB et de Simulink de conception au niveau système avec le logiciel d'Altera QuartusII.

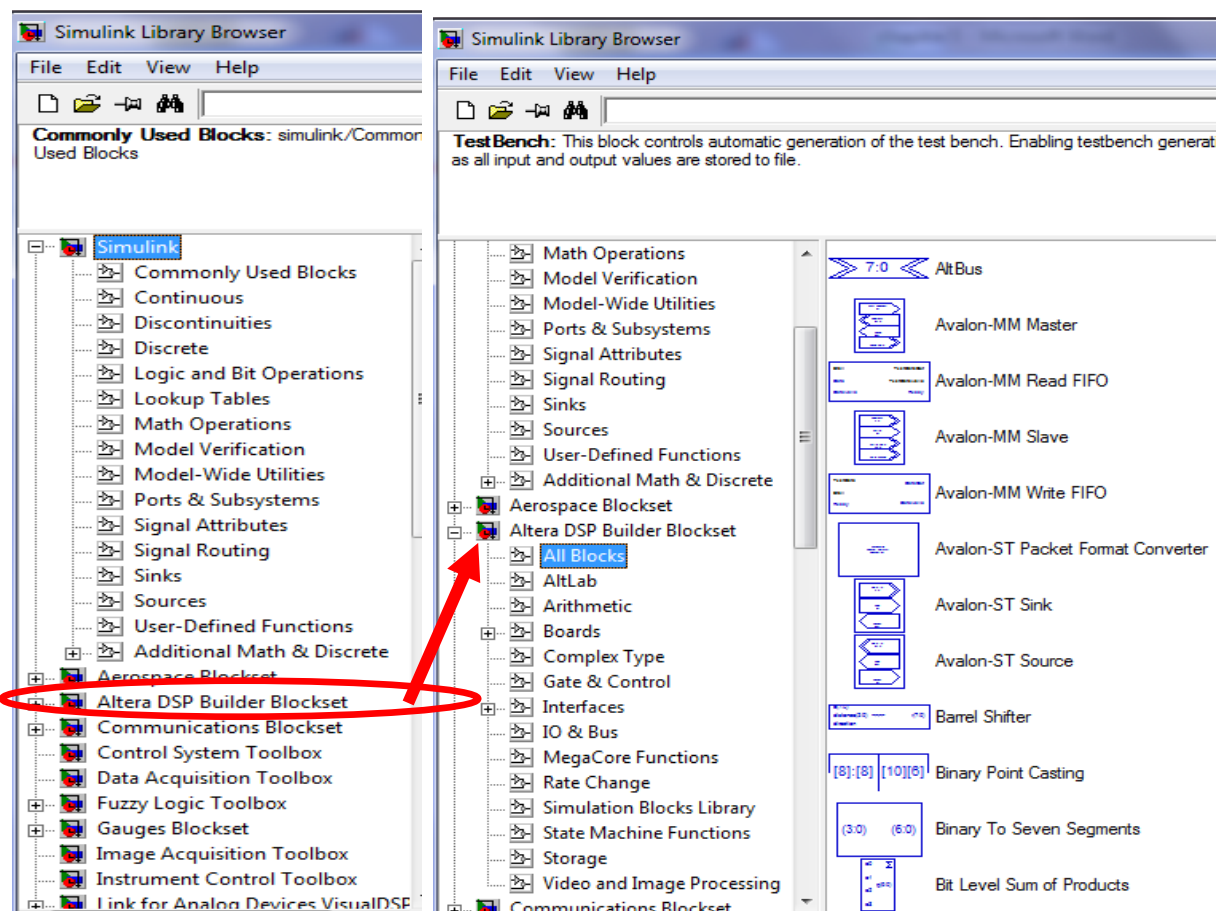





Figure (V.1) : Présentation de la bibliothèque DSP-Builder.

*Pour créer un modèle sous DSP-Builder, nous allons exécuter les étapes suivantes :*

- 1- Lancer le logiciel **Matlab**.
- 2- Sélectionner **Fil > New > Model**, pour créer un nouveau modèle.
- 3- Sélectionner **Fil > save** , pour sauvegarder le nouveau modèle (*exemple : chaos.mdl*)
- 4- Cliquer sur l'icône **start**  **du Matlab**.
- 5- **Cliquer sur Altera DSP Builder Blockset** ou bien Vous pouvez ouvrir **Simulink** par l'utilisation de l'icône  , puis cliquer sur **Altera DSP Builder Blockset**.
- 6- Sélectionner les blocks.
- 7- On bascule alors sur le modèle créé (*chaos.mdl*).
- 8- Lancer la simulation : cliquer sur l'icône **start** .

### V.3. Simulation sous DSP-Builder :

#### V.2.1. Émetteur :

Avant d'arriver à la simulation du circuit, tout d'abord, nous discrétisons le système de QI (équation III.1), pour mettre le système continu sous une forme numérique et pour cela nous avons l'équation suivante :

$$\frac{dx_i}{dt} = f(x_1, x_2, x_3, \dots, x_N) = \lim_{\tau \rightarrow 0} \frac{x_{i(n+1)} - x_{i(n)}}{\tau} \quad (IV.1)$$

Alors :

$$x_{i(n+1)} = x_{i(n)} + \tau(f(x_1, x_2, x_3, \dots, x_N)) \quad (IV.2)$$

Comme dans notre cas  $i=1, 2, 3$  et  $4$  alors l'expression (IV.2) devient :

$$\left\{ \begin{array}{l} x_{1(n+1)} = x_{1(n)} + \tau(a(x_{2(n)} - x_{1(n)}) + x_{2(n)}x_{3(n)}) \\ x_{2(n+1)} = x_{2(n)} + \tau(b(x_{2(n)} + x_{1(n)}) + x_{1(n)}x_{3(n)}) \\ x_{3(n+1)} = x_{3(n)} + \tau(-cx_{3(n)} - ex_{4(n)} + x_{1(n)}x_{2(n)}) \\ x_{4(n+1)} = x_{4(n)} + \tau(-dx_{4(n)} + fx_{3(n)} + x_{1(n)}x_{2(n)}) \end{array} \right. \quad (IV.3)$$

Donc, notre Émetteur est aura pour équation :

$$\left\{ \begin{array}{l} x_{1m(n+1)} = x_{1m(n)} + \tau(50(x_{2m(n)} - x_{1m(n)}) + x_{2m(n)}x_{3m(n)}) \\ x_{2m(n+1)} = x_{2m(n)} + \tau(20(x_{2m(n)} + x_{1m(n)}) + x_{1m(n)}x_{3m(n)}) \\ x_{3m(n+1)} = x_{3m(n)} + \tau(-13x_{3m(n)} - 33x_{4m(n)} + x_{1m(n)}x_{2m(n)}) \\ x_{4m(n+1)} = x_{4m(n)} + \tau(-8x_{4m(n)} + 30x_{3m(n)} + x_{1m(n)}x_{2m(n)}) \end{array} \right. \quad (IV.4)$$

## Chapitre V : Réalisation

On insère l'équation (IV.4) dans DSP-Builder, on observe le comportement de chaque étage de la simulation, la simulation se fera pour les valeurs suivantes :

$a=50$ ,  $b=20$ ,  $c=13$ ,  $d=8$ ,  $e=33$ ,  $f=30$ .

a) *Schéma de simulation sous DSP-Builder :*

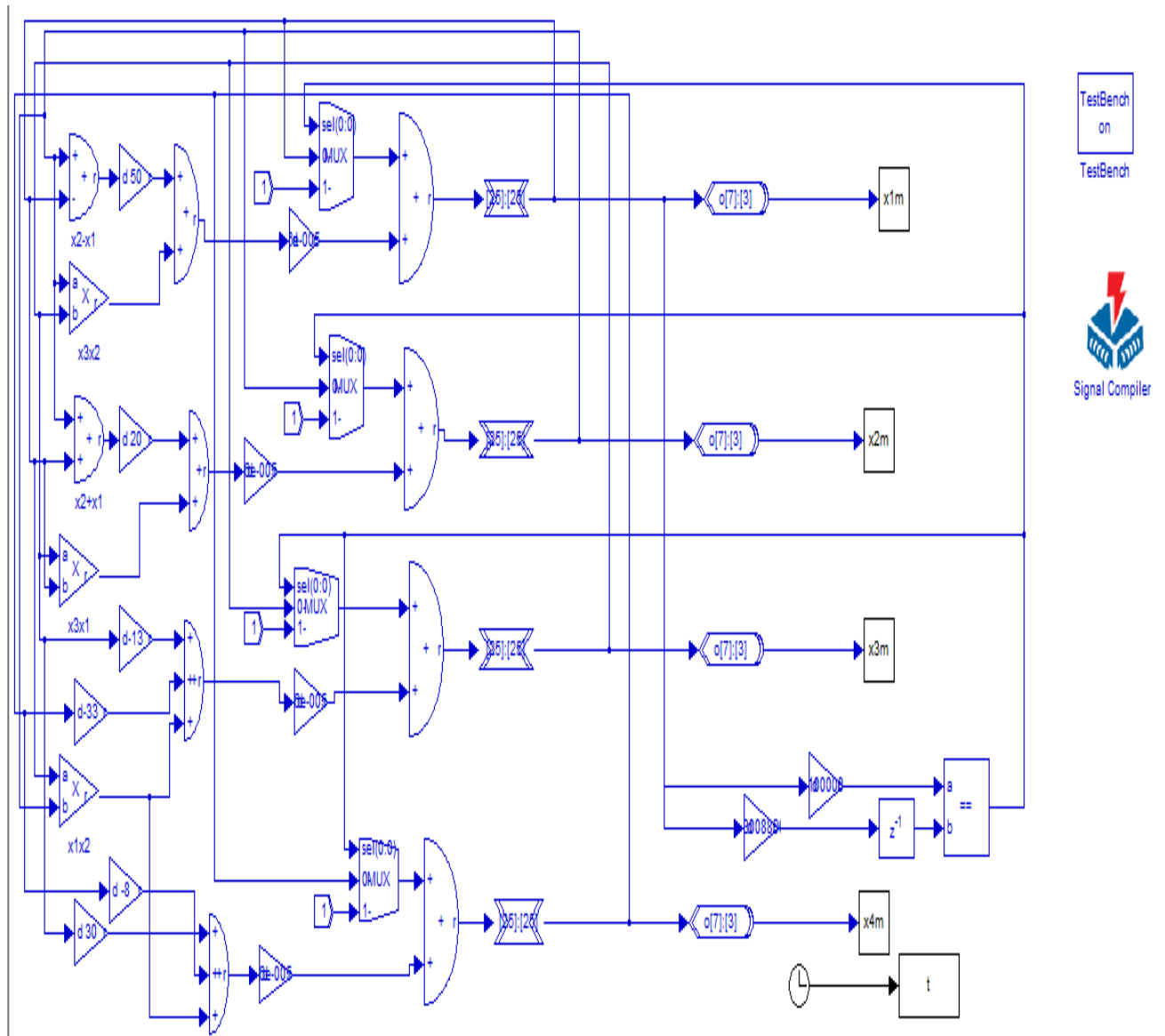
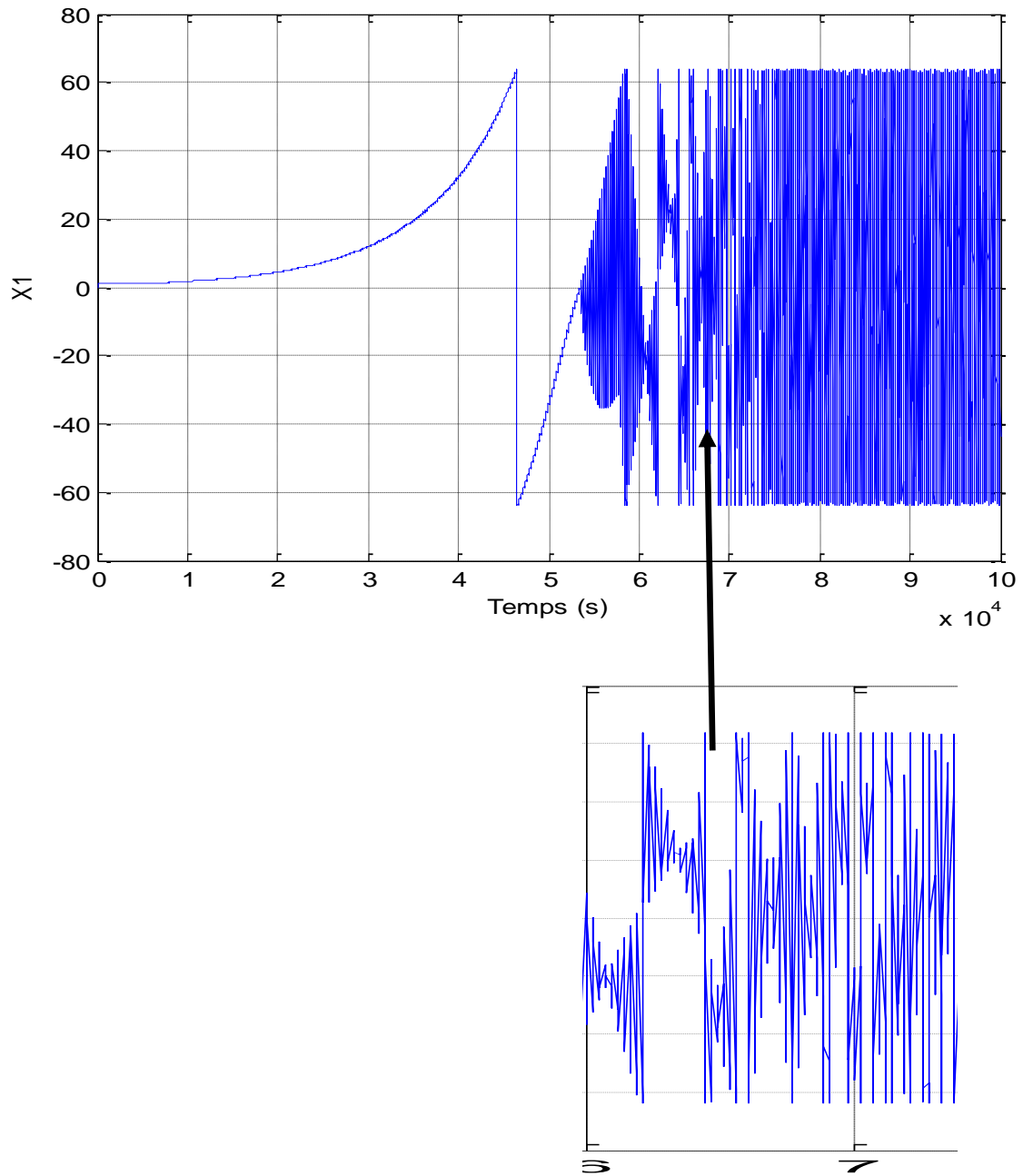
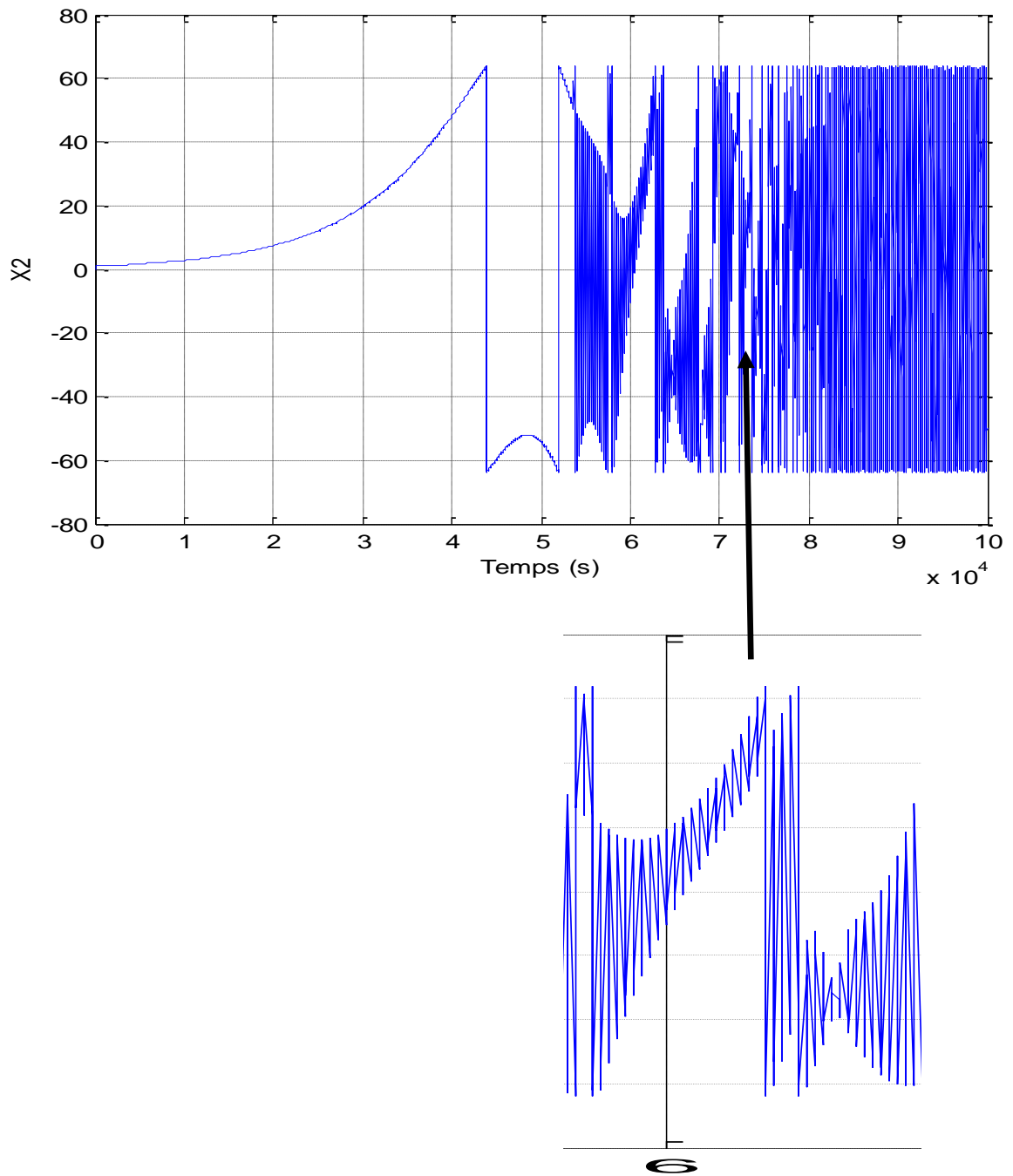


Figure (V.2) : le schéma de simulation de l'émetteur.

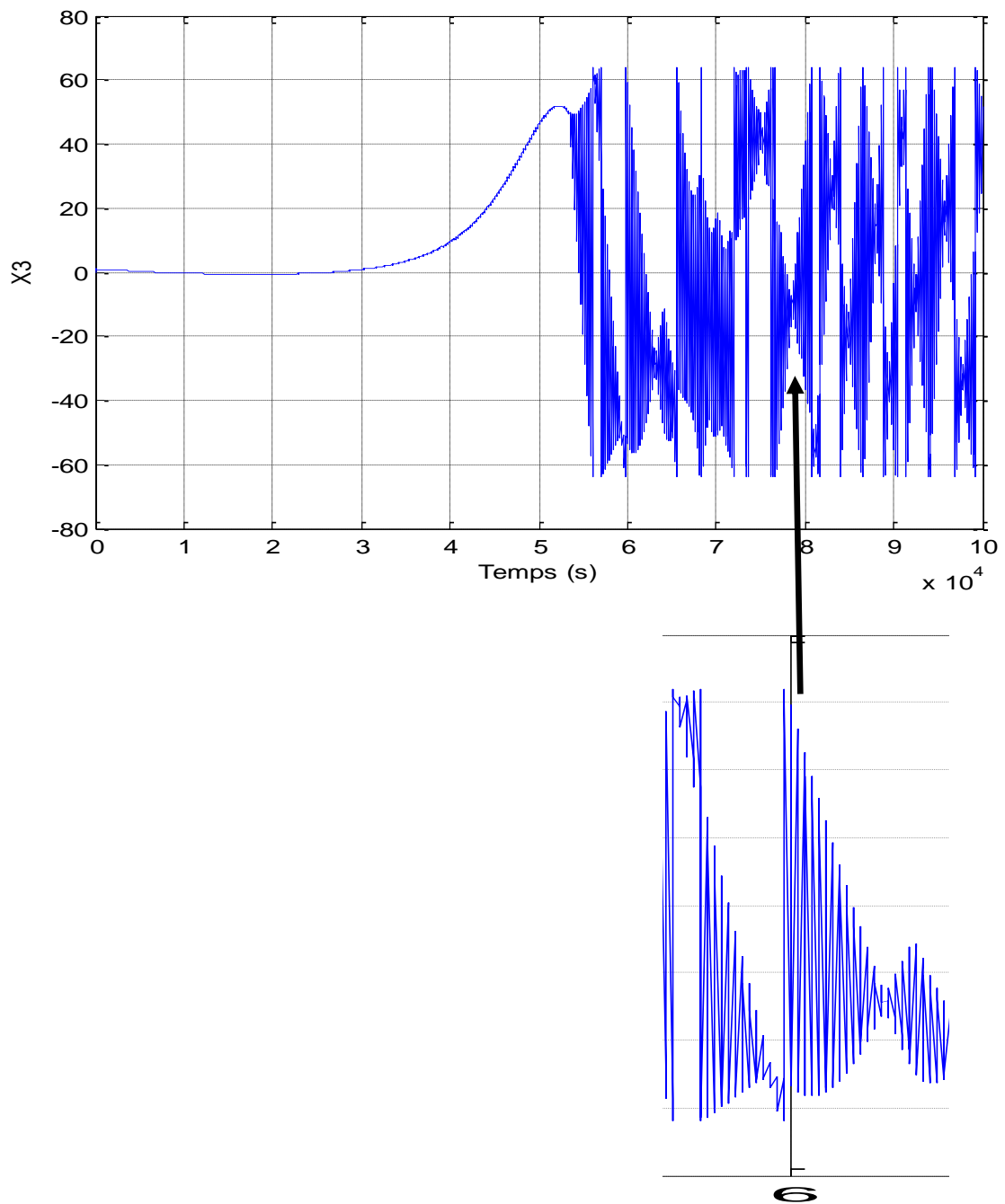
*b) Visualisation des états :*



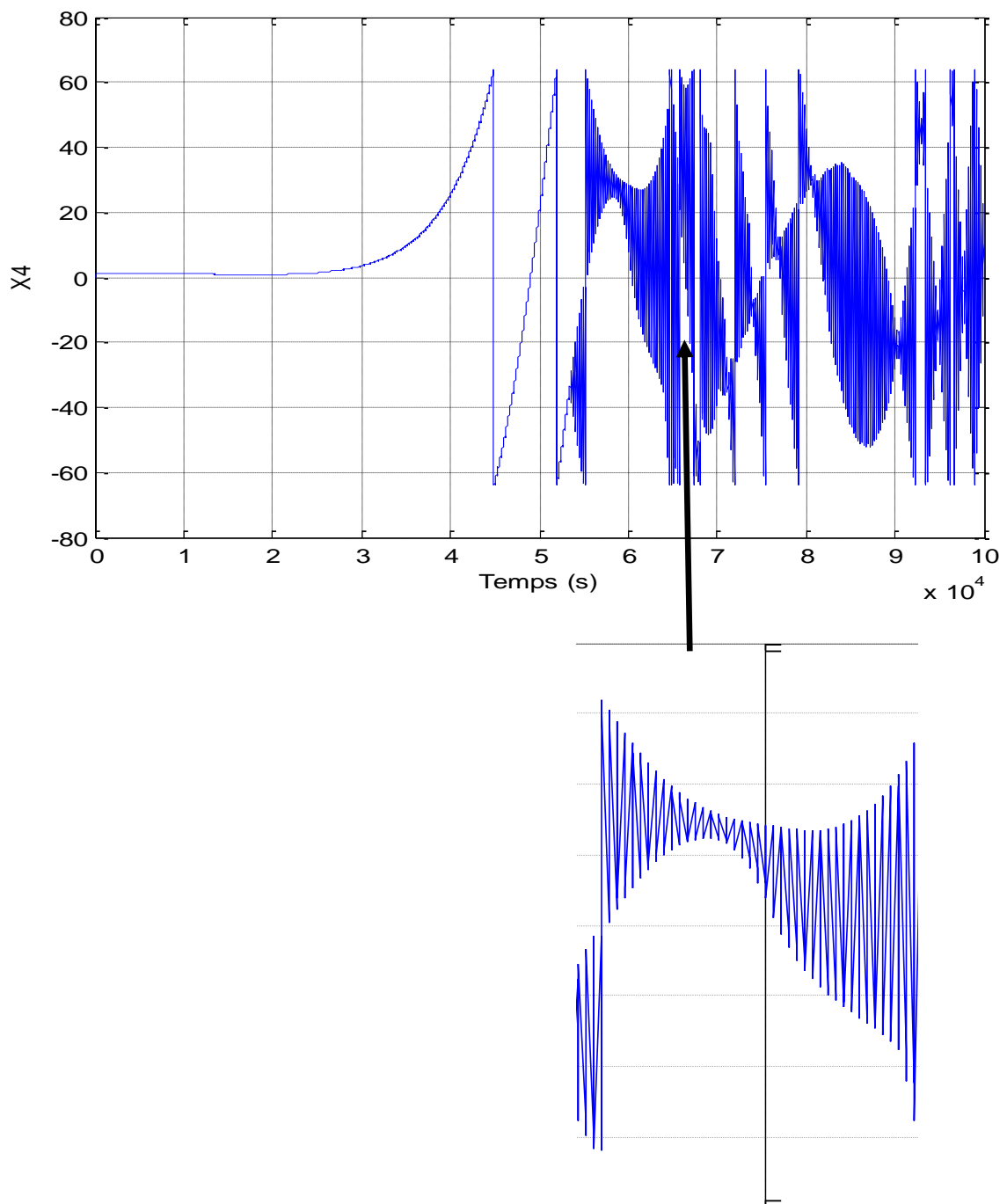
*Figure (V.3): Graphe de l'état  $x_{1m}(t)$  du l'émetteur.*



*Figure (V.4) : Graphe de l'état  $x_{2m}(t)$  du l'émetteur.*



*Figure (V.5) : Graphe d'état  $x_{3m}(t)$  du l'émetteur.*



*Figure (V.6) : Graphe d'état  $x_{4m}(t)$  du l'émetteur.*

❖ *Observation :*

D'après les quatre dernières figures nous remarquons bien la présence d'oscillations irrégulières et aperiodiques alors les signaux  $x_{1m}(t)$ ,  $x_{2m}(t)$ ,  $x_{3m}(t)$  et  $x_{4m}(t)$  sont des signaux chaotiques.

## c) Visualisation des attracteurs :

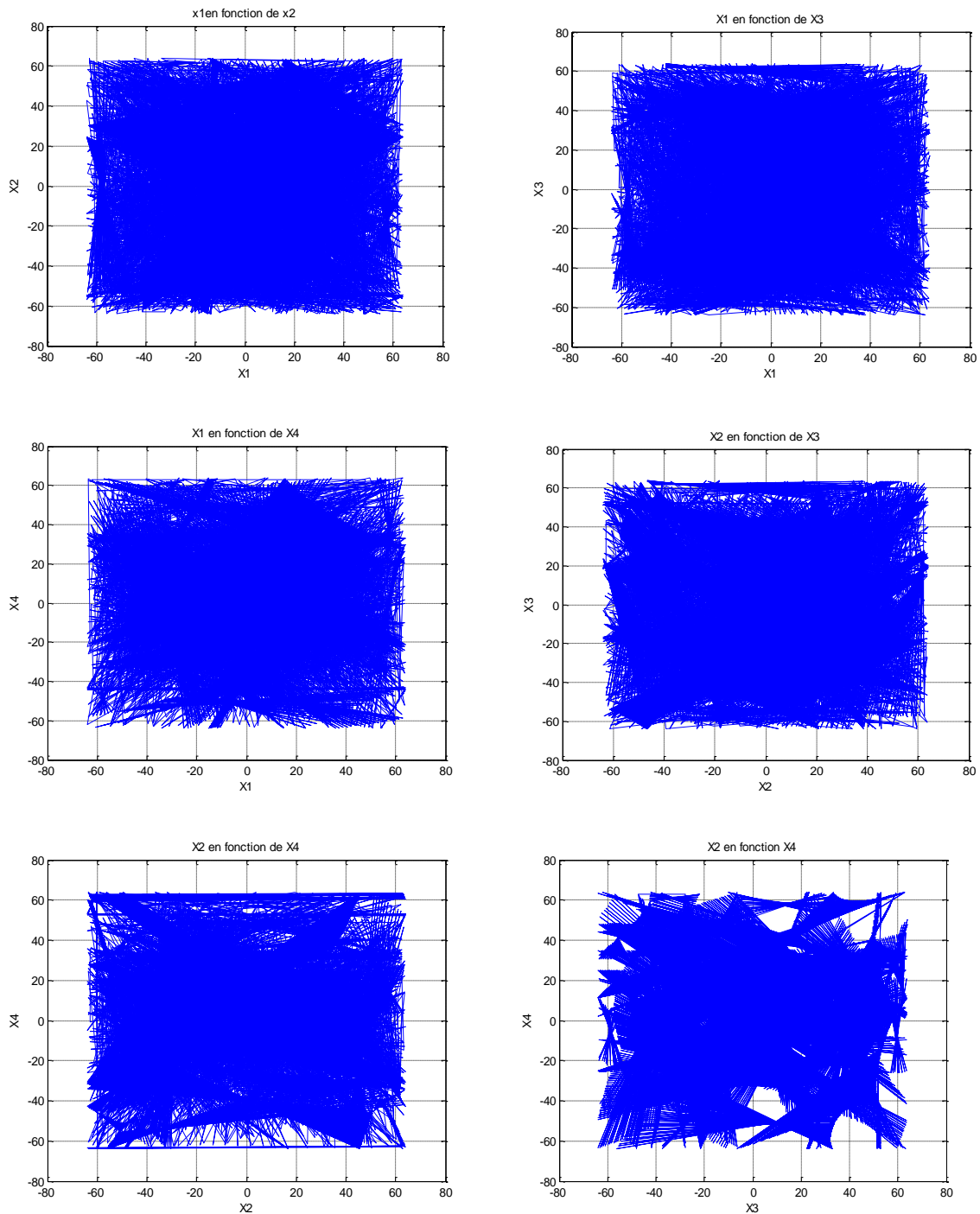
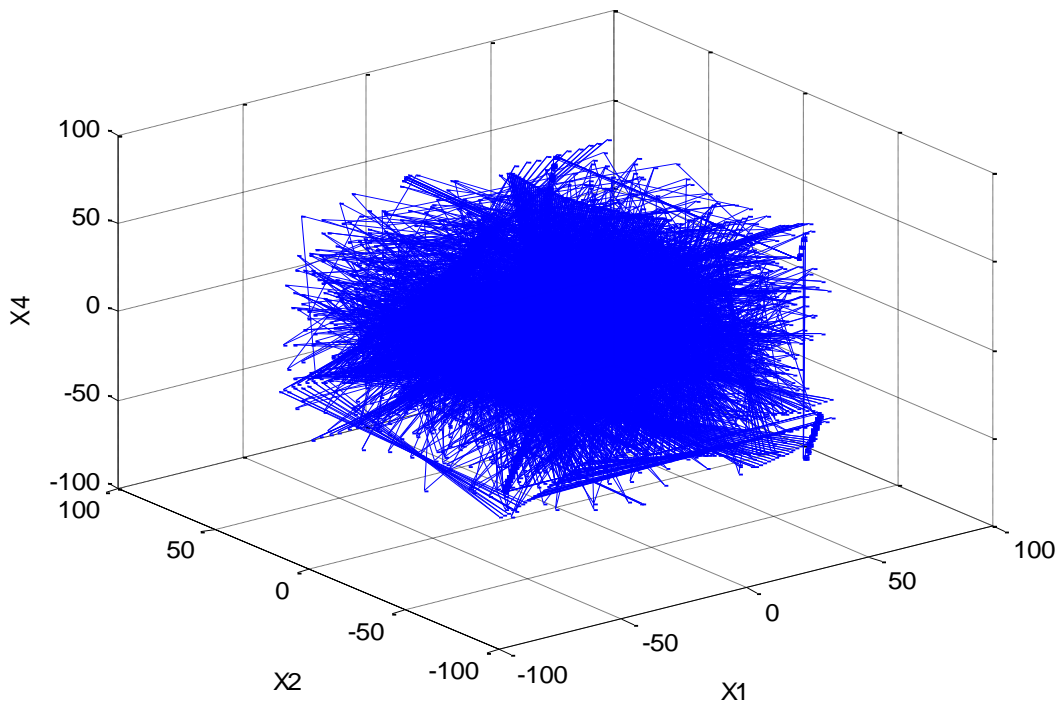
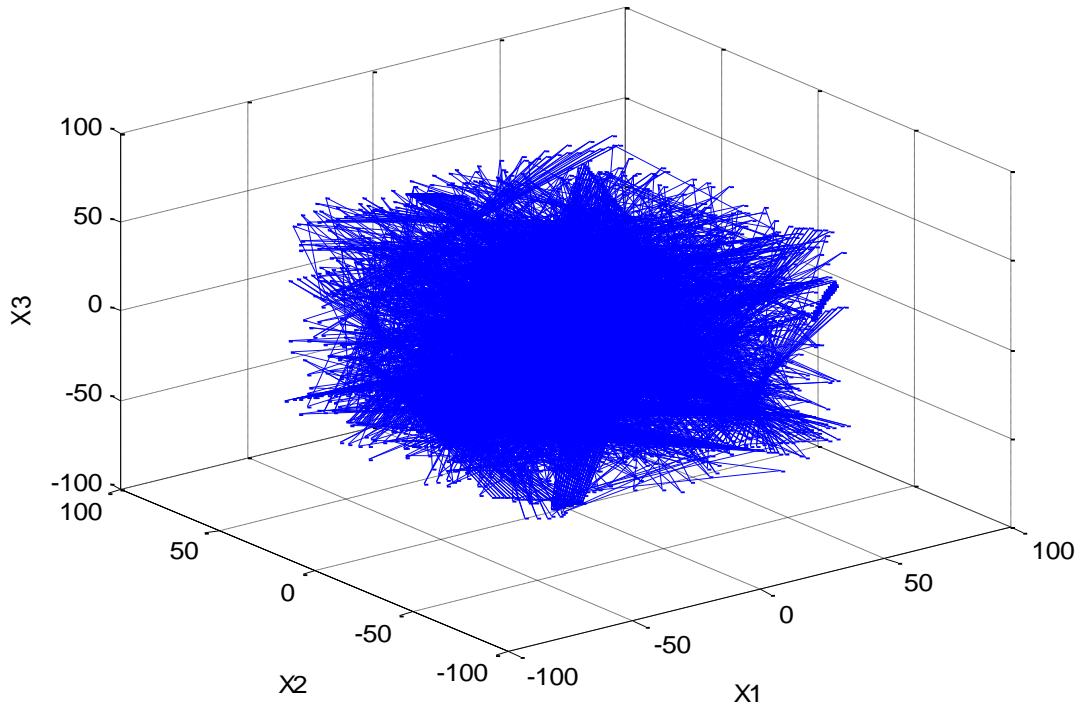
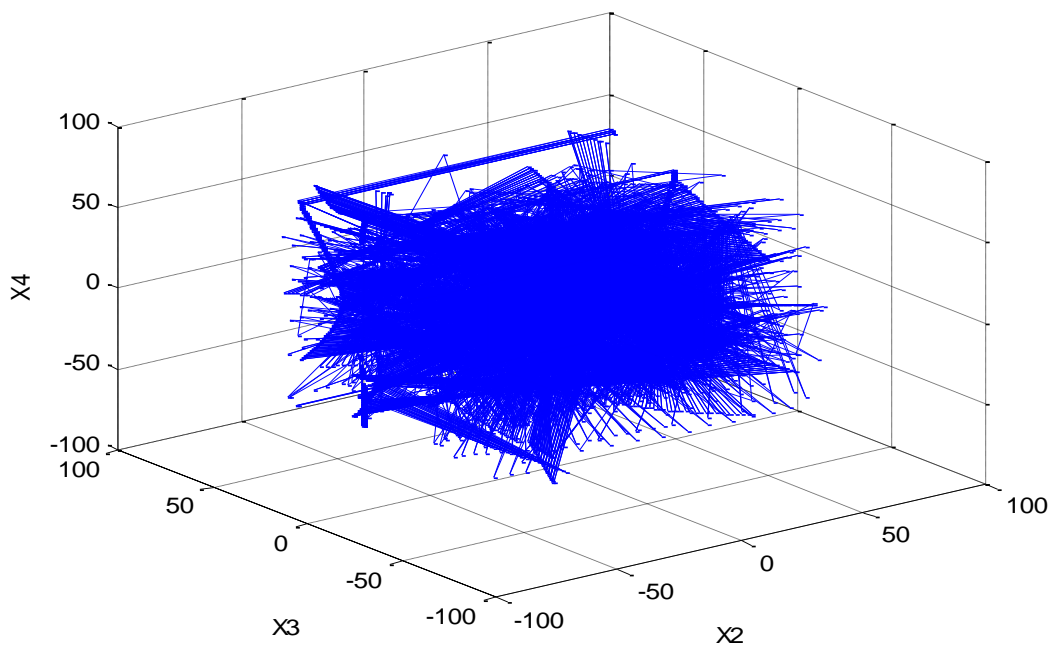
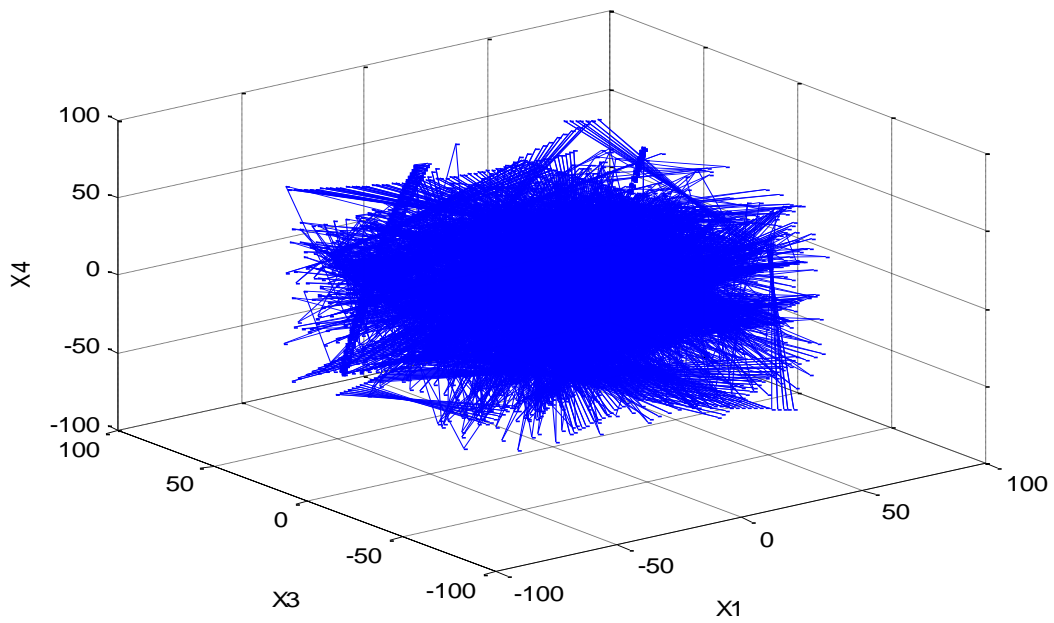


Figure (V.7) : Les attracteurs en deux dimensions.

*d) Visualisation en trois dimensions :*





*Figure (V.8) : Les attracteurs en trois dimensions.*

❖ **Observation :**

Les figures ci-dessus démontrent l'aspect hyper chaotique du système de QI.

### V. 3.2. Récepteur :

Notre récepteur aura pour équations :

$$\begin{cases} \dot{x}_{1s} = a(x_{2s} - x_{1s}) + x_{2m}x_{3m} \\ \dot{x}_{2s} = b(x_{1s} + x_{2s}) - x_{1m}x_{3m} - k(x_{2s} - x_{2m}) \\ \dot{x}_{3s} = -cx_{3s} - ex_{4s} + x_{1m}x_{2m} \\ \dot{x}_{4s} = -dx_{4s} + fx_{3s} + x_{1m}x_{2m} \end{cases} \quad (\text{V.5})$$

On insère les équations (V.5) du récepteur dans DSP-Builder, et on additionne un message signal sinusoïdal) dans l'émetteur dans l'un de ces étages et on observe le comportement de chaque étage du récepteur.

Une simple opération de soustraction entre  $x_{4m}$  et  $x_{4s}$  permet de récupérer le message (le message sera ajouté au niveau de  $x_{4m}$ ).

La simulation se fera pour  $a=50$ ,  $b=20$ ,  $c=13$ ,  $d=8$ ,  $e=33$ ,  $f=30$  et  $k=100$ .

Donc l'équation (V.5) devient :

$$\begin{cases} \dot{x}_{1s} = 50(x_{2s} - x_{1s}) + x_{2m}x_{3m} \\ \dot{x}_{2s} = 20(x_{1s} + x_{2s}) - x_{1m}x_{3m} - 100(x_{2s} - x_{2m}) \\ \dot{x}_{3s} = -13x_{3s} - 33x_{4s} + x_{1m}x_{2m} \\ \dot{x}_{4s} = -8x_{4s} + 30x_{3s} + x_{1m}x_{2m} \end{cases} \quad (\text{V.6})$$

#### a) Le schéma de Simulation :

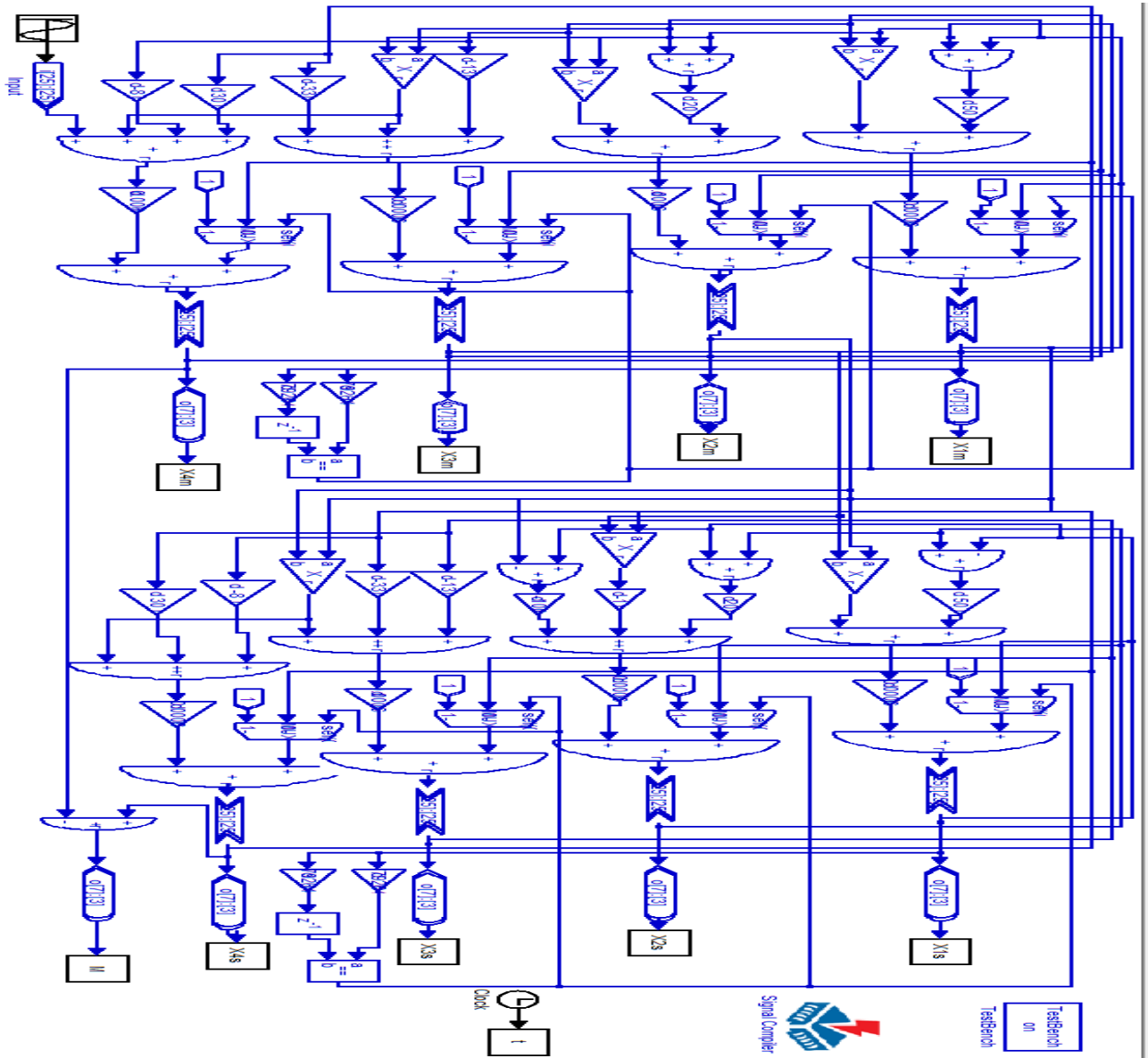
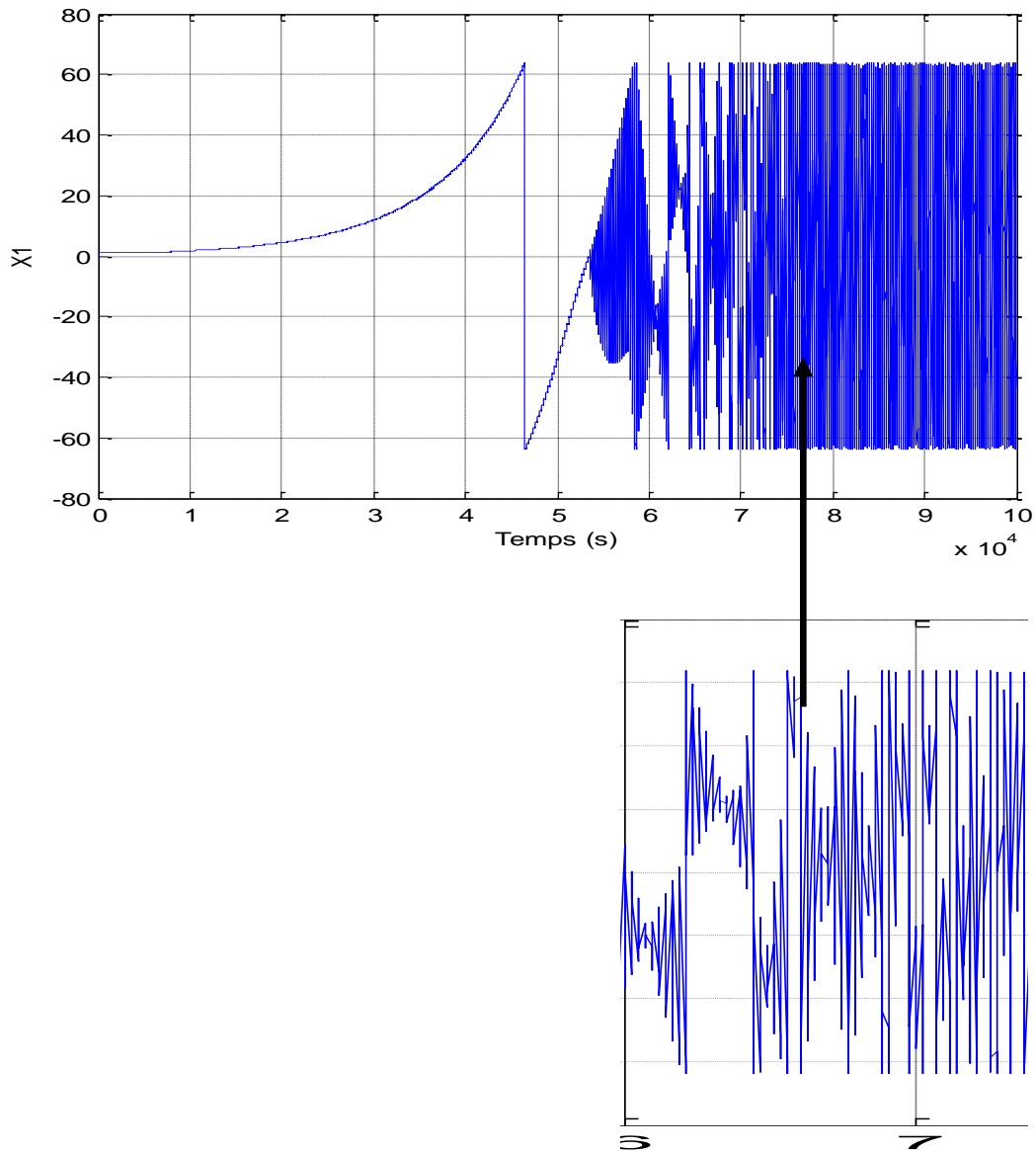
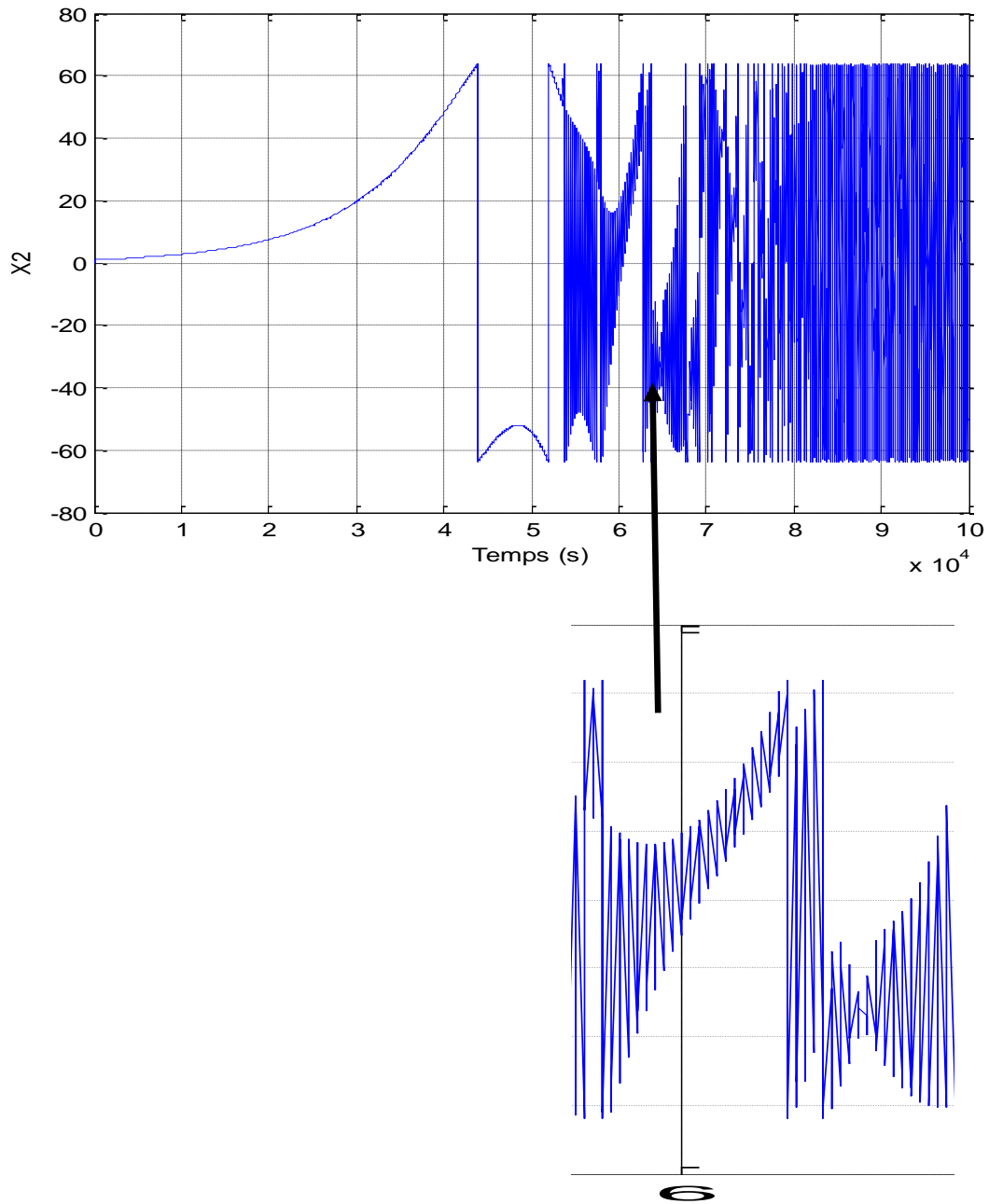


Figure (V.9) : Le schéma de simulation Émetteur-Récepteur.

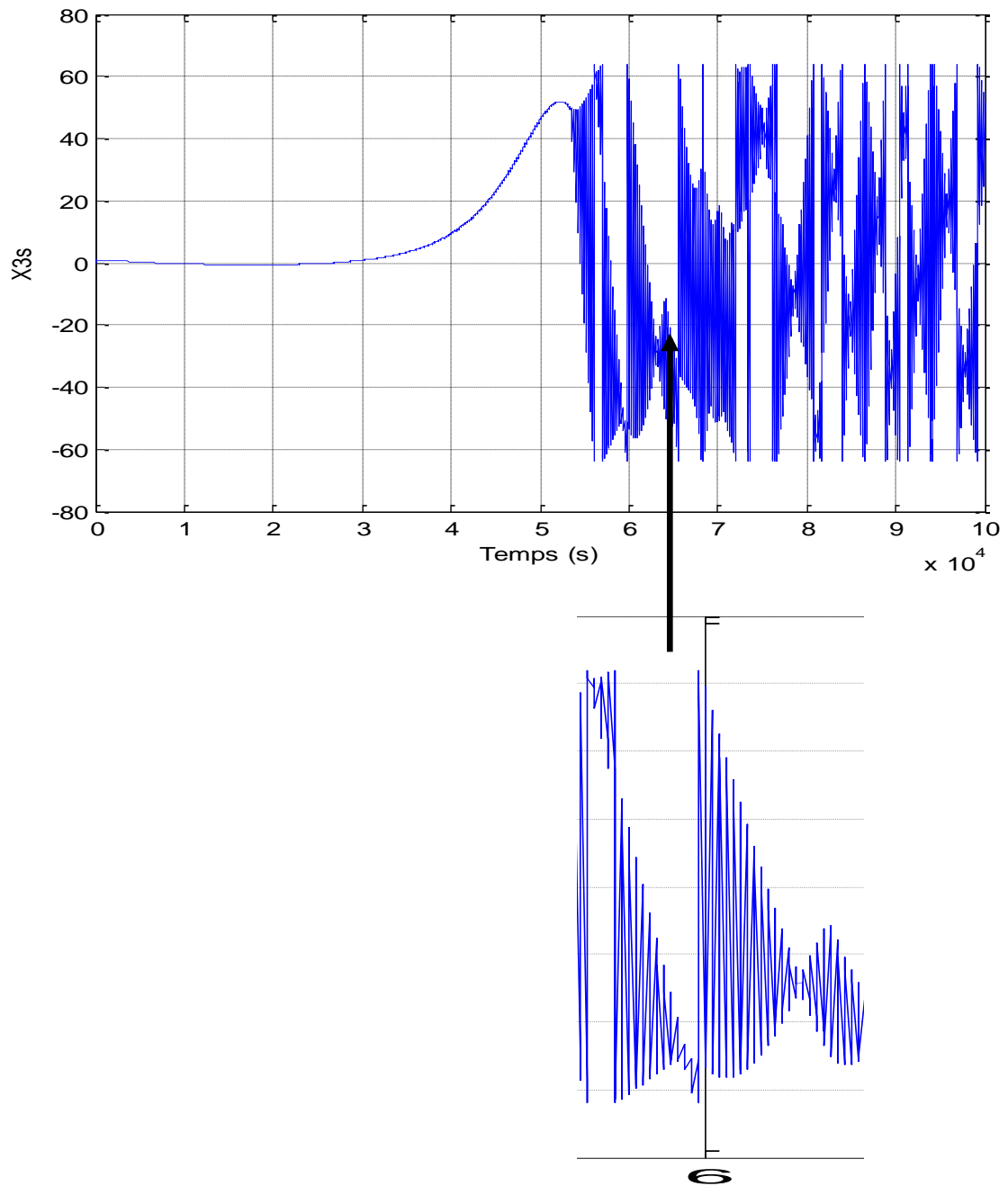
b) *Visualisation des états :*



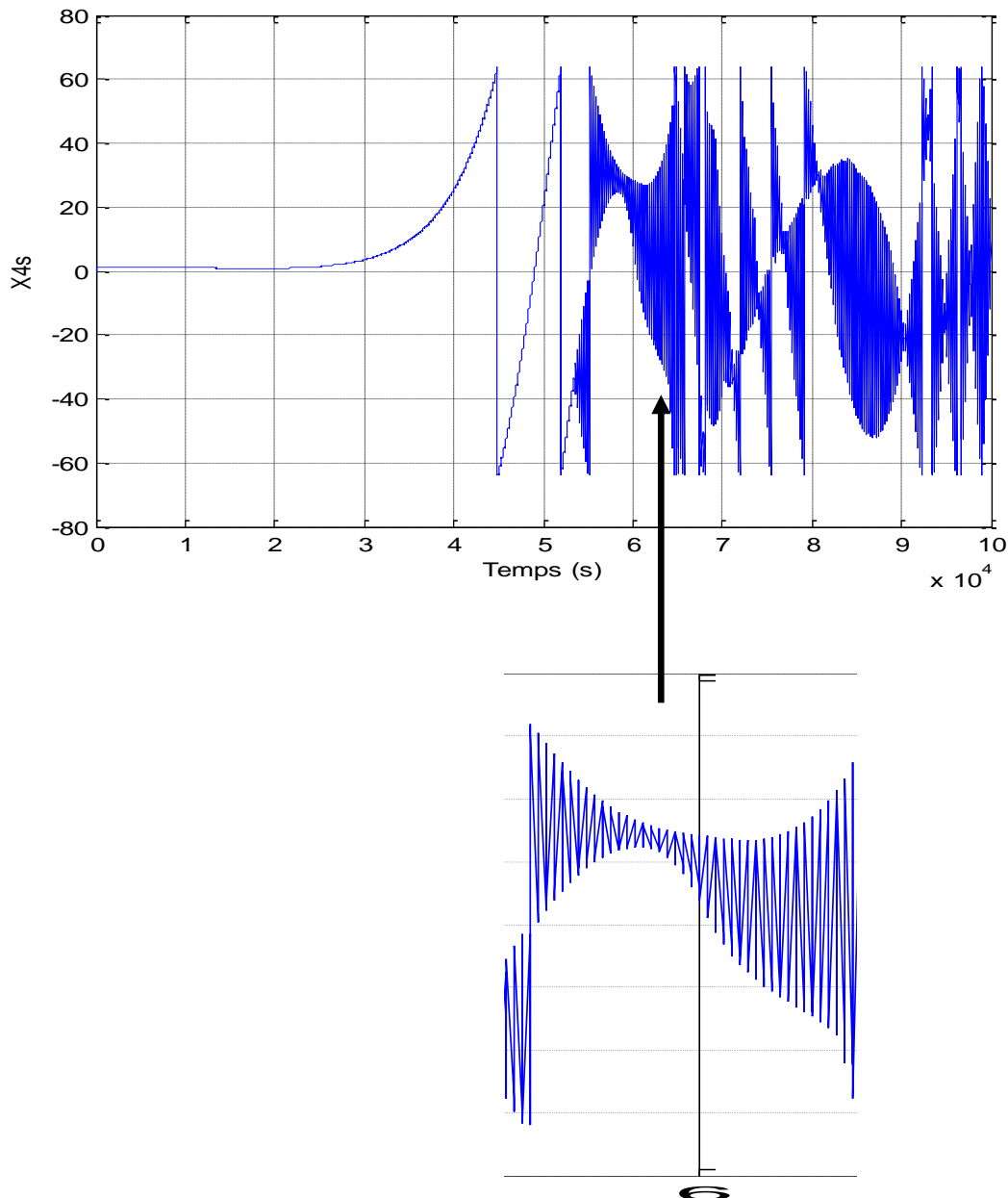
*Figure (V.10) : le graphe d'état  $x_{1s}(t)$ .*



*Figure (V.11) : le graphe d'état  $x_{2s}(t)$ .*



*Figure (V.12) : le graphe d'état  $x_{3s}(t)$ .*



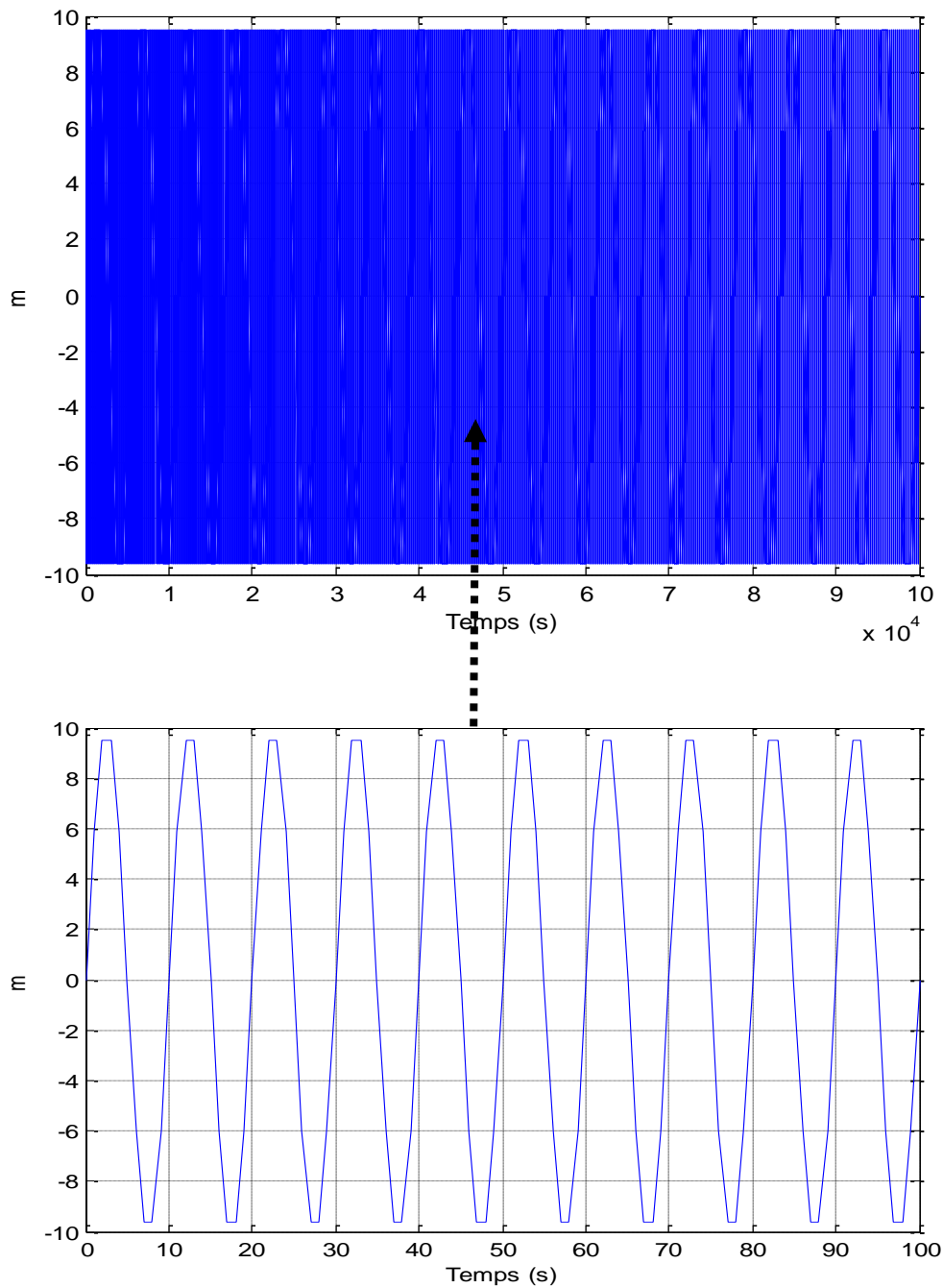
*Figure (V.13) : le graphe d'état  $x_{4s}(t)$ .*

❖ **Observation :**

Nous remarquons la présence d'oscillations irrégulières et apériodiques dans ces figures, donc les signaux  $x_{1s}(t)$ ,  $x_{2s}(t)$ ,  $x_{3s}(t)$  et  $x_{4s}(t)$  sont des signaux chaotiques, et aussi nous remarquons bien que les signaux du récepteur sont identiques aux signaux de l'émetteur ce qui démontre la synchronisation des deux systèmes et alors le message envoyé au niveau de l'émetteur sera récupéré facilement avec une simple opération de soustraction.

**c) Récupération du message :**

Pour récupérer le message, nous faisons une soustraction entre les états  $x_{4m}(t)$  et  $x_{4s}(t)$ .



*Figure (V.14) : Comparaison entre le message et la sortie de récepteur.*

❖ **Observation :**

La figure (V.14) montre bien que le message envoyé a été bien récupéré, ce qui prouve l'efficacité de l'approche.

### V.4 .Implémentation sur la carte FPGA :

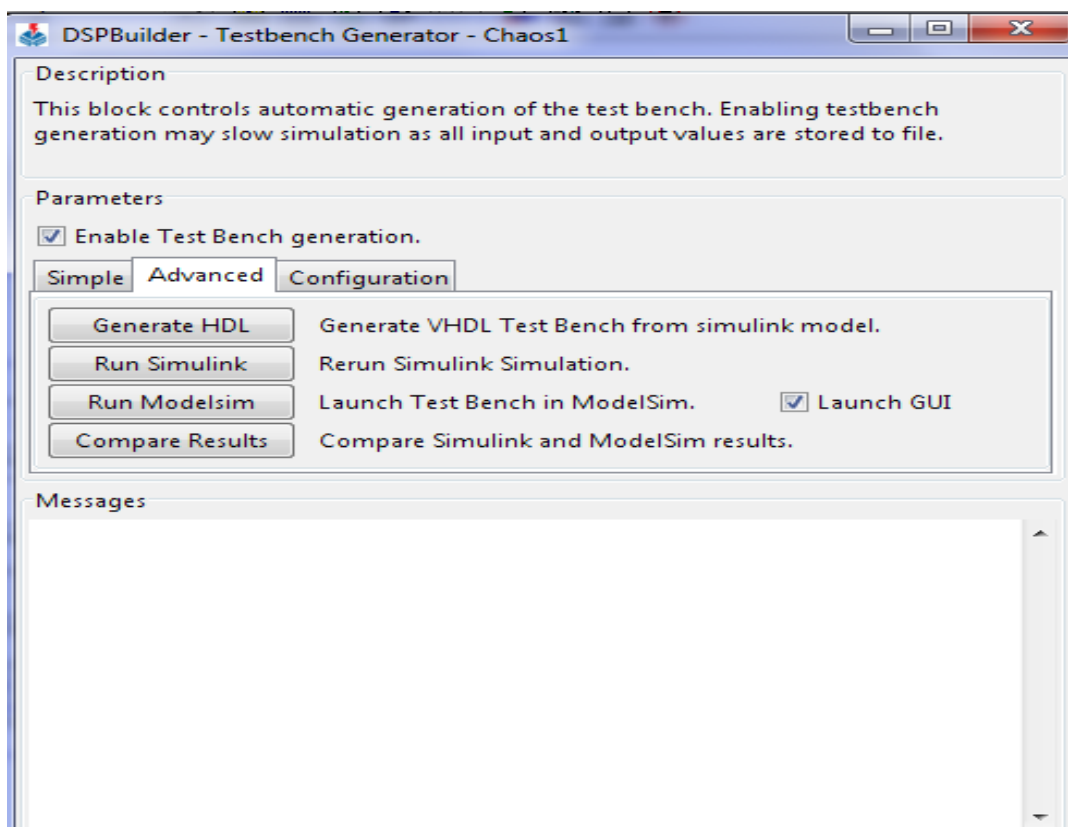
Avant d'implémenter les signaux sur la carte électronique FPGA, tout d'abord, nous générons le code **VHDL**.

Pour générer le code VHDL, tout d'abord on va prendre le bloc **Test Bench** dans la fenêtre **Atera DSP Builder BlockSet**



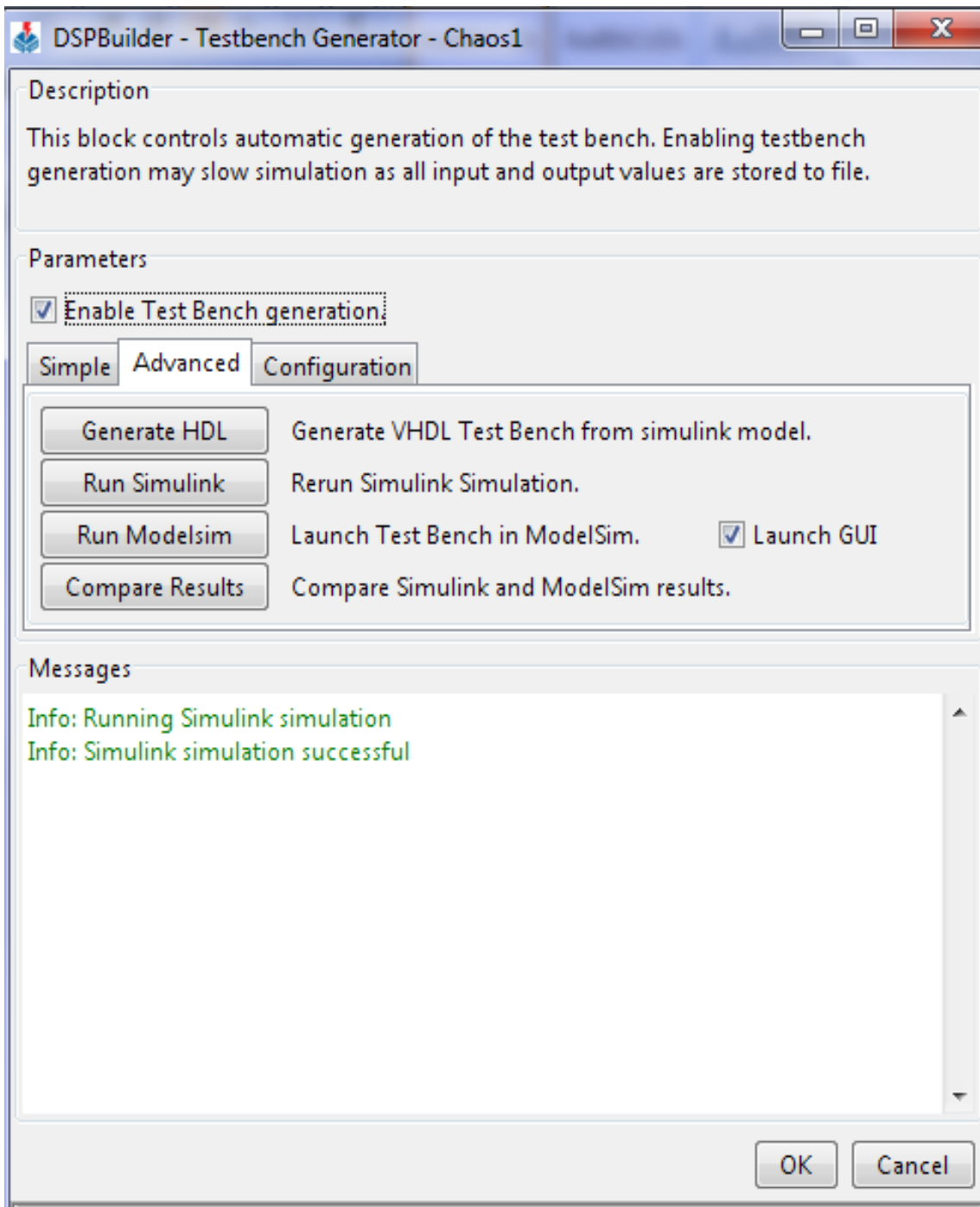
*Figure (V.15) : Le block TestBench.*

Puis on sélectionne sur **Enable Test Bench** pour que ce block soit en marche



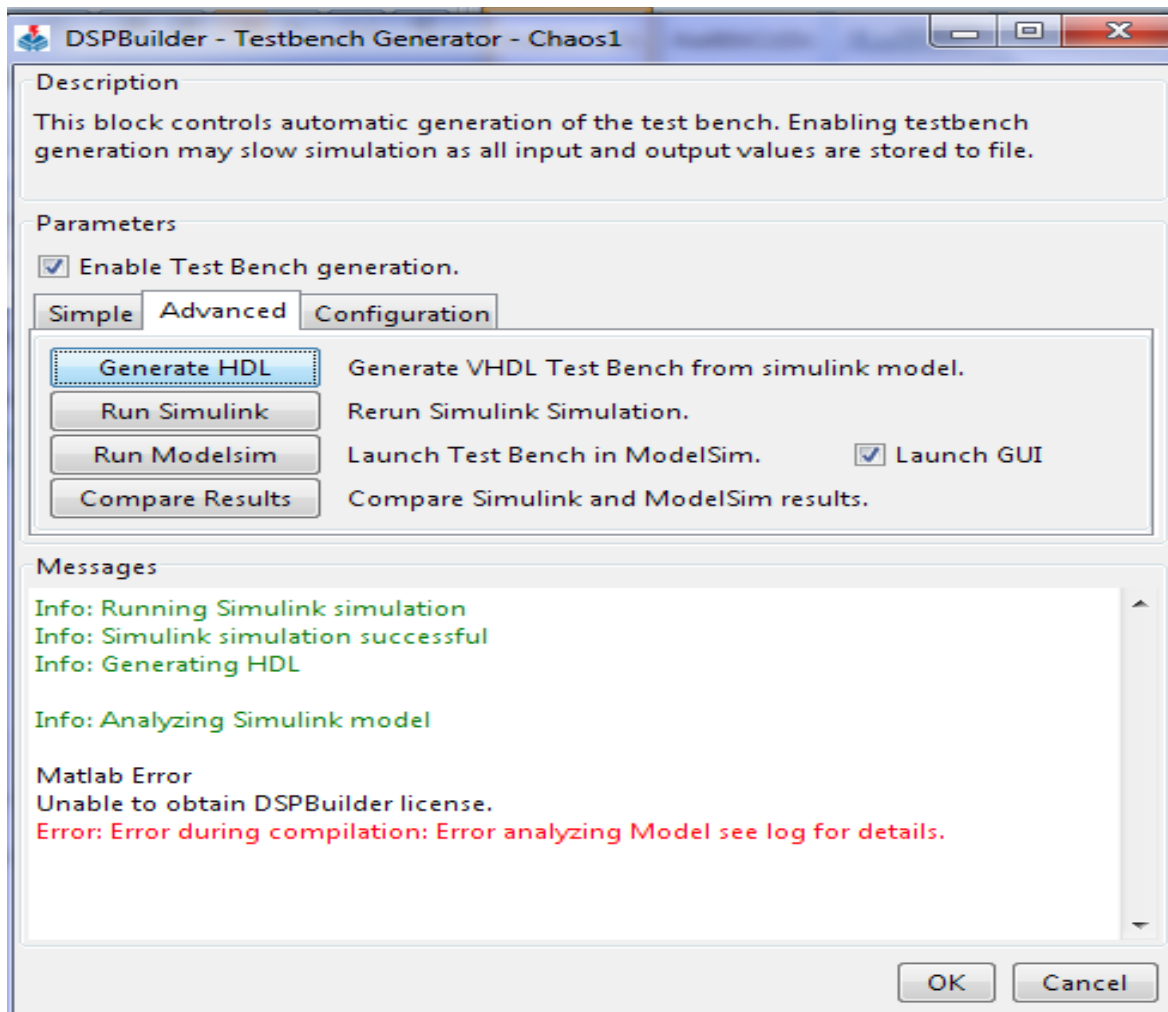
*Figure (V.16) : La configuration du bloc TestBench.*

Ensuit on clique sur *Advanced*



*Figure (V.17) : Simulation du programme sur le bloc TestBench.*

Enfin, on clique sur *Generate HDL* pour générer le code *VHDL*



*Figure (V.18) : Généré le code VHDL.*

Mais malheureusement, on n'a pas la licence de ce logiciel donc on ne peut pas générer le code VHDL et alors on ne peut pas réaliser notre application sur la carte FPGA.

#### **V.4. Conclusion :**

Dans ce chapitre nous avons discrétiser notre oscillateur de QI sous la forme numérique, nous avons aussi simulé notre système (émetteur et récepteur) sur le DSP Builder, pour la suite nous avons donné une idée sur les différentes étapes à suivre pour réaliser notre système sur une carte FPGA, mais malheureusement on n'a pas réussi dans la pratique à cause du manque de moyens, il ne manque qu'une licence pour notre logiciel (DSP-Builder).

# Conclusion Générale

## Conclusion Générale

---

Le travail développé dans ce mémoire a fait l'objet d'une étude et une réalisation d'une transmission sécurisée par le chaos. Ce qui nous a permis de toucher à un domaine difficile qui est le chaos et la transmission sécurisée, où nous avons présenté tous les points essentiels concernant le chaos, tel que leurs définitions, et leurs caractéristiques.

Le premier chapitre est consacré à l'étude des systèmes chaotique par leurs caractéristiques et leurs comportements.

Le deuxième chapitre expose la synchronisation leurs déférentes méthodes ainsi le chaos dans la transmission sécurisée.

Dans le troisième chapitre, nous avons élaboré les équations différentielles régissant le système de transmission, nous avons utilisé comme émetteur le système hyperchaotique de QI. La conception du récepteur s'est faite par un circuit de QI identique à l'émetteur que nous avons synchronisé à l'aide d'observateur à retour d'état. Nous avons ensuite inséré un message dans l'émetteur en l'additionnant à un de ses états. Nous avons simulé le système obtenu sur Matlab et nous avons récupéré le message envoyé au niveau de l'émetteur.

Le quatrième chapitre présente les circuits logiques programmables avec ses différentes classes et en particulier les circuits FPGA, leur architecture et leur principe de fonctionnement.

La réalisation pratique exposée dans le cinquième chapitre s'est fait par la carte FPGA. Nous avons pu obtenir les mêmes résultats obtenus sous en analogique dans le chapitre trois, mais on à pas pu visualiser ces résultats sur la carte FPGA à cause des moyennes (il ne manque la licence de logiciel DSP-Builder).

Nos résultats prouvent que la transmission sécurisée par le chaos, étudiée avant par simulation uniquement, fonctionne en pratique et ouvrent une possibilité de développement de ces méthodes dans le futur dans le domaine de la communication.

# Annexe A

## Annexe A : quelque définition

### A.1. Définition (un système non linéaire) :

Un système physique est dit non linéaire, si la relation entre les grandeurs d'entrées et les grandeurs de sorties est un système d'équations différentielles avec des coefficients non constants.

En général, la représentation d'état de tels systèmes est formulée par un ensemble d'équations différentielles non linéaires du premier ordre de type :

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) & \text{Équation d'état.} \\ \dot{y}(t) = g(x(t), u(t)) & \text{Équation de sortie.} \end{cases}$$

Avec

$x(t), u(t)$  et  $y(t)$  Représentent respectivement le vecteur d'état, le vecteur d'entrée, et le vecteur de sortie à l'instant  $t$ ,  $f$  est la fonction vectorielle de dimension  $n$ .

### A.2. Définition (un système dynamique) :

Mathématiquement, un système dynamique est décrit par un problème où seules sont données les valeurs de départ des variables d'état. Il peut avoir une composante de temps "**discrète**" ou "**continu**".

On appelle système dynamique en temps continu, tout système décrit par un système d'équations différentielles alors qu'en temps discret on parle d'équations aux différences finies. Donc, un système dynamique en temps continu est décrit par un système d'équations différentielles, temps discret on parle d'un système d'équations aux différences finies, alors les systèmes dynamiques sont classés en deux catégories :

- Système dynamique à temps discret.
- Système dynamique à temps continu.

**1-En temps continu :**

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) & x(t_0) = x_0 \\ y(t) = h(x(t), u(t), t) \end{cases} \quad (A.1)$$

Où

$x(t) = [x_1(t) \ x_2(t) \ x_3(t) \ \dots \ x_n(t)]^T$  est le vecteur d'état du système,  $x_1(t) \ x_2(t) \ \dots \ x_n(t)$  sont les états du système et  $n$  est la dimension du système (le degré de liberté).

$u$  est l'entrée du système et  $t$  la variable temporelle ( $u \in \mathbb{R}^p$ ).

$f$  est une fonction non linéaire désignant un champ de vecteurs, et  $h$  est une fonction éventuellement non linéaire qui désigne le vecteur de sortie.

## Annexe A : quelque définition

➤ Remarque :

Si le système (A.1) ne dépend pas de l'entrée, alors on aura :

$$\begin{cases} \dot{x}(t) = f(x(t), t) \\ x(t_0) = x_0 \end{cases} \quad (A.2)$$

Le système 2 est un système dynamique, car à partir de n'importe quelle condition initiale  $x_0$  on peut obtenir l'état futur  $x(t)$  pour  $t > 0$ .

**2- temps discret :**

Comme il a été déjà précisé, le système dynamique est dans ce cas représenté par des équations aux différences finies, avec le modèle général suivant :

$$\begin{cases} x(k+1) = G(x(k), u(k), k) \\ y(k+1) = h(x(k), u(k), k) \\ x(k_0) = k_0 \end{cases} \quad (A.3)$$

Avec:

$G: \mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^p$  désigne la dynamique du système.

**A.3. Définition (un système déterministe) :**

On dit d'un système qu'il est déterministe si en connaissant l'état du système à un instant donné, on est, on mesure de prévoir son état à instant ultérieur.

**A.4. Définition (degré de liberté) :**

Est le nombre de variables qui caractérise un espace d'état c'est-à-dire il représente l'ordre qui égale à la dimension de l'espace d'état (il caractérise l'espace d'état).

**A.5. Définition (espace des phases) :**

Dans un système dynamique de dimension  $n$ , l'espace  $x_1, x_2, \dots, x_n$  est appelé espace des phases ou espace d'état. Ainsi, le chemin parcouru par le système est appelé « trajectoire »,  $x_1, x_2, \dots, x_n$  sont les états du système.

**A.6. Définition (un Système autonome) :**

Un système dynamique non linéaire est dit autonome lorsqu'il ne dépend pas explicitement du temps, un système autonome est donné ci-dessous :

$$\dot{x} = f(x, u) \quad (A.4)$$

ou  $x \in \mathbb{R}^n$

## Annexe A : quelque définition

---

Un système autonome est indépendant du temps initial et tout instant peut être considéré comme instant initial, et tout état  $x(t)$  du système peut être considéré comme un état initial.

### A.7. Définition (une Trajectoire et flot) :

Si on observe l'ensemble des différents états successifs de l'espace d'état, on peut remarquer l'émergence d'une trajectoire dans cet espace. Cette trajectoire est également appelée orbite du système, il est à noter que si les variables d'état prennent des valeurs réelles, l'orbite d'un système dynamique à temps continu sera une courbe alors que l'orbite d'un système dynamique discret sera représentée par une série de points.

Donc toute solution  $\psi(t, x)$  du système autonome considérée comme un ensemble de trajectoires avec différentes conditions initiales est appelée *flot*.

### A.8. Définition (un Attracteurs) :

Si l'on observe l'ensemble des différents états successifs de l'espace d'état, on peut observer l'émergence d'une **trajectoire** dans cet espace. Cette trajectoire est également appelée **orbite** du système. Il est à noter que si les variables d'état prennent des valeurs réelles, l'orbite d'un système dynamique à temps continu sera une courbe alors que l'orbite d'un système dynamique discret sera représentée par une série de points.

**L'attracteur est une limite vers laquelle semblent converger les orbites du système.** On peut définir un attracteur comme un ensemble compact de l'espace d'état vers lequel toutes les trajectoires environnantes, convergent c'est-à-dire que l'attracteur décrit en fait une situation de régime telle qu'elle peut apparaître après disparition des **phénomènes transitoires**. Le bassin d'attraction est alors l'ensemble des points initiaux dont les trajectoires convergent vers l'attracteur. On peut avoir plusieurs attracteurs dans un même espace des phases. Il existe deux types d'attracteurs :

- les attracteurs réguliers (le point fixe, le cycle limite, le tore)
- les attracteurs étranges (chaotique)

Le type des attracteurs réponde du comportement du système.

### A.9. Définition (Les systèmes dynamiques non linéaires) :

À partir d'un état initial  $x_0$  et après un régime transitoire la trajectoire d'un système dynamique non linéaire atteint une région limitée de l'espace des phases ce comportement asymptotique obtenu  $t \rightarrow \infty$  est une des caractéristiques les plus importantes des systèmes dynamiques, si dans le cas d'un système linéaire la solution asymptotique est indépendante de la condition initiale en présence de non-linéarités, il existe une plus grande variété de régimes permanents parmi lesquelles on trouve par ordre de complexité : point d'équilibre, solutions périodiques, solution quasi périodique et chaos

Il faut préciser que cette fois le comportement développé par un système dynamique particulier est fortement dépendant de la condition initiale choisie. Ces solutions correspondent à différents types d'attracteurs dans l'espace des phases.

### A.10. Définition (Point d'équilibre) :

Dans ce cas, la solution asymptotique est représentée par un point, sa valeur étant déterminée en fonction de la condition initiale choisie. Ainsi, pour des conditions initiales différentes on peut retrouver plusieurs points d'équilibres, il existe plusieurs types des points

## Annexe A : quelque définition

---

d'équilibre (nœud, col, foyer), de même ces points peuvent être stables ou instables suivant que les trajectoires voisines convergent ou divergent entre elles.

Dans l'espace des phases l'attracteur correspondant à une solution d'équilibre est un point fixe, c'est la catégorie la plus simple d'attracteurs parce qu'elle n'évolue plus dans le temps.

Donc un point d'équilibre ou un point fixe est un point qui vérifie :

$$\text{En temps continu : } f(x^*) = 0 \tag{A.5}$$

$$\text{En temps discret : } f(x^*) = x^*$$

### ***A.11. Définition (Régime périodique) :***

Le régime asymptotique permanent périodique correspond à une trajectoire dont les répliques d'une portion élémentaire sont espacées à des intervalles  $nT$ ,  $n \in \mathbb{N}^+$ ,  $T$  désignant la période

#### ➤ ***Cycle limite :***

Un système non linéaire peut être un siège d'oscillations, caractérisé par leurs amplitudes et leurs fréquences, indépendantes de la condition initiale  $x_0$ , et sans excitation extérieure ce siège est appelé cycle limite. Un cycle limite est une trajectoire de phase fermée à un nombre fini de fréquences dans l'espace d'état.

### ***A.12. Définition (régime quasi périodique) :***

Corresponds à une somme de solutions périodiques dont le rapport des périodes est un nombre irrationnel. Un régime quasi périodique peut être représenté dans l'espace d'état par un tore.

### ***A.13. Définition (régime chaotique) :***

Le régime chaotique est par définition tout régime permanent qui n'appartient à aucune des classes présentées antérieurement. Une telle solution a une trajectoire asymptotique bornée avec une extrême sensibilité aux conditions initiales. Ainsi deux trajectoires générées à partir de CI (conditions initiales) très proches, vont diverger très vite l'une par rapport à l'autre. Cette sensibilité par rapport aux CI traduit aussi le comportement en apparence stochastique des générateurs chaotiques, de telle sorte qu'une prévision à long terme du comportement du système est impossible.

# Annexe B

L'environnement de développement permettant de programmer les composants reconfigurables d'Altera se nomme Quartus. Il permet de réaliser l'intégralité des étapes permettant de configurer un composant programmable, alors ce logiciel permet la programmation des circuits programmables ALTERA.

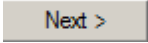
### **B.1- Créer un projet :**

#### **B.1.1- Lancer le logiciel Quartus :**

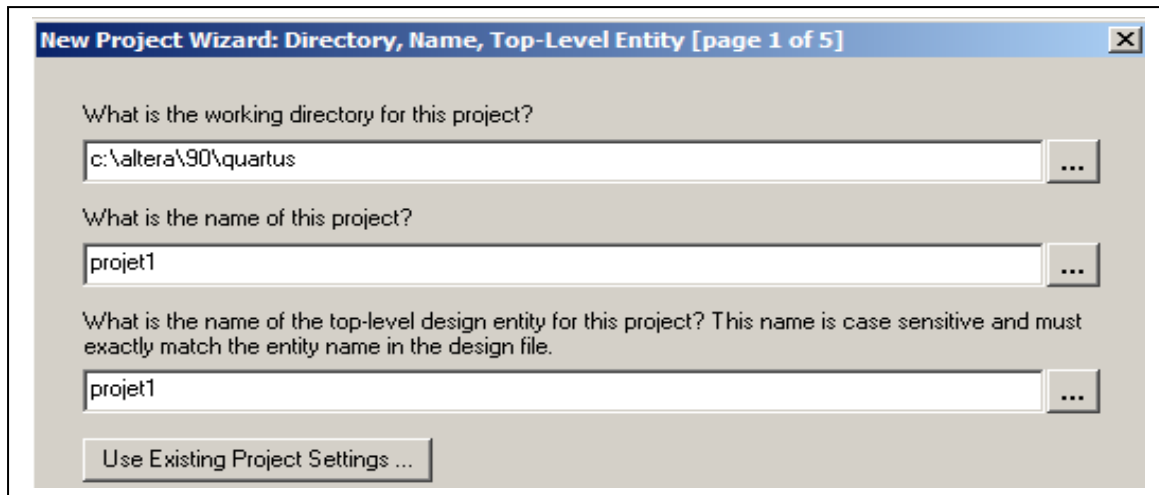
Lancez le logiciel Quartus II.



#### **B.1.2- Créer un projet sous Quartus avec l'assistant de création :**

- 1- **Quartus** fonctionnant par projet il est nécessaire de commencer par créer un projet à chaque début de conception.
- 2- Entrer ensuite dans le Project **Wizard** le nom choisi pour votre projet :
- 3- Cliquer 2 fois sur 

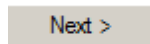
## Annexe B



4- Choisir le FPGA présent sur votre carte en sélectionnant :

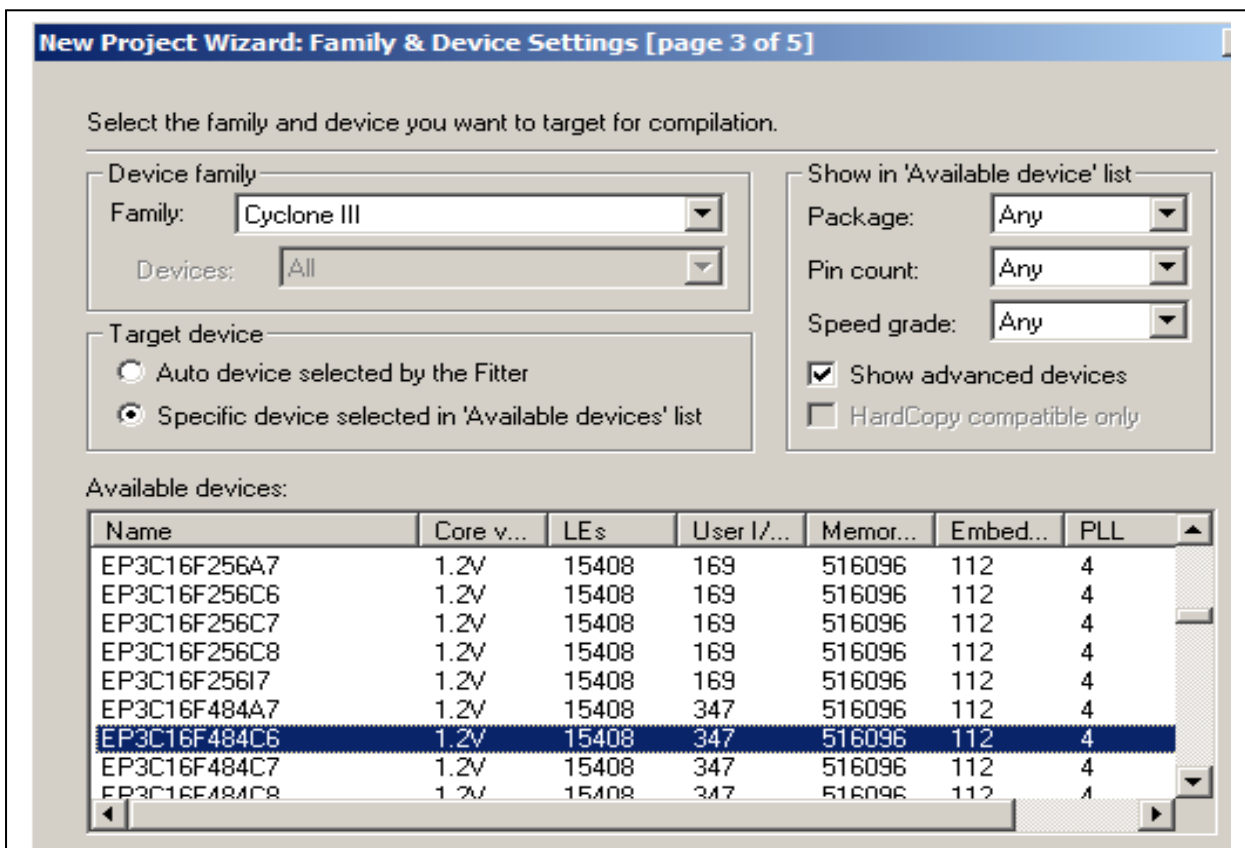
- **Add files**

- Next>

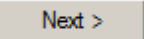
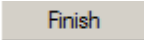


- **Device Settings**

- La famille de FPGA : *Cyclone III*
  - Le modèle : *EP3C16F84C6*



## Annexe B

- 5- Cliquer 2 fois sur  puis sur 
- 6- **EDA Tools Settings :**

**Design Entry:** Design Compiler / VHDL

**Simulation:** ModelSim-Altera / VHDL

**Timing:** Custom / VHDL

Specify the other EDA tools used with the Quartus II software to develop your project.

EDA tools:

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synthesis	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	ModelSim-Altera	VHDL	<input checked="" type="checkbox"/> Run gate-level simulation automatically after compilation
Timing Analysis	<None>	<None>	<input type="checkbox"/> Run this tool automatically after compilation
Formal Verification	<None>		
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

Cliquer sur **Next>**

**Finish** 

New Project Wizard: Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings:

Project directory:  
E:/ALTERA/descriptions/FL\_base/

Project name: FL\_base

Top-level design entity: FL\_base

Number of files added: 0

Number of user libraries added: 0

Device assignments:  
Family name: Cyclone II  
Device: EP2C35F672C6

EDA tools:  
Design entry/synthesis: <None>  
Simulation: <None>  
Timing analysis: <None>

Operating conditions:  
Core voltage: 1.2V  
Junction temperature range: 0-85 °C


< Back Next > Finish Annuler

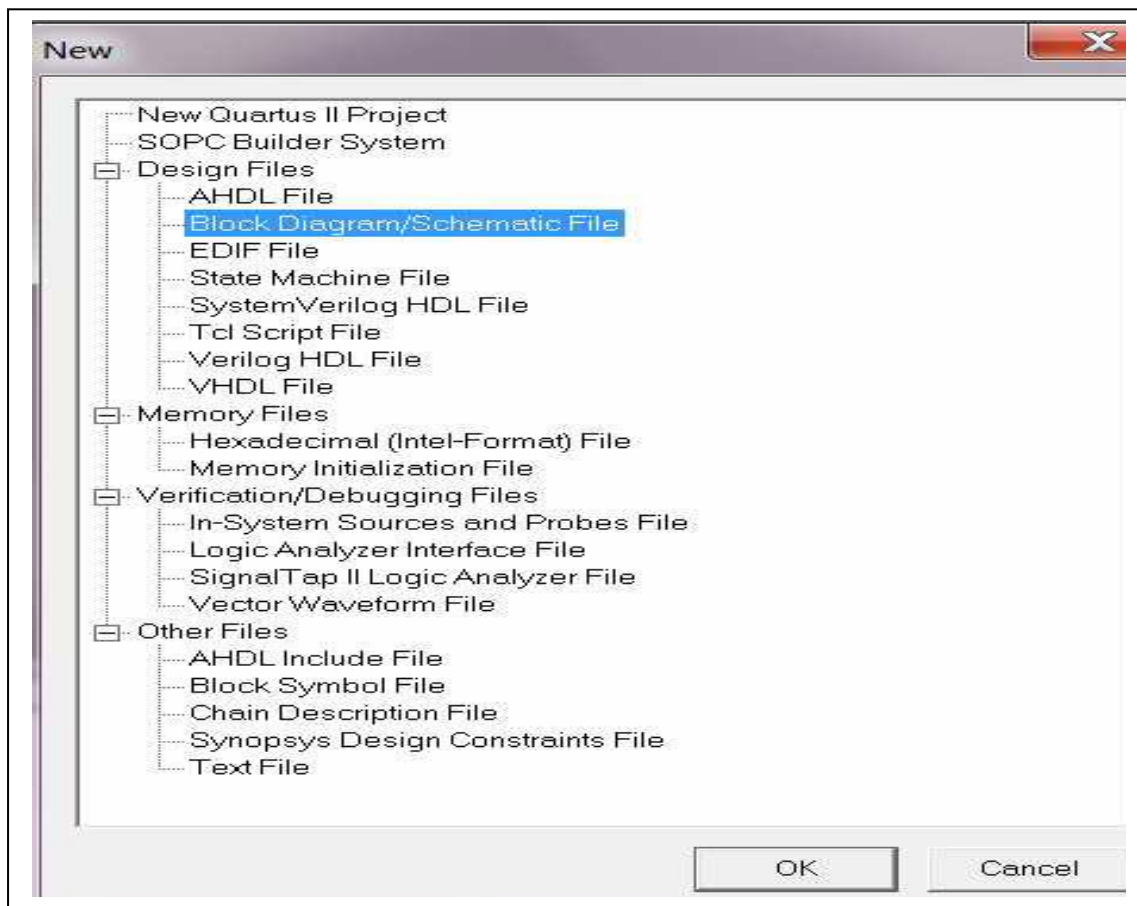
### B.2- Créer le ou les fichiers décrivant votre projet :


Un projet est généralement constitué de plusieurs fichiers décrivant chacun une ou plusieurs fonctions numériques. Parmi les possibilités proposées par Quartus pour créer des fichiers, nous allons en voir deux :

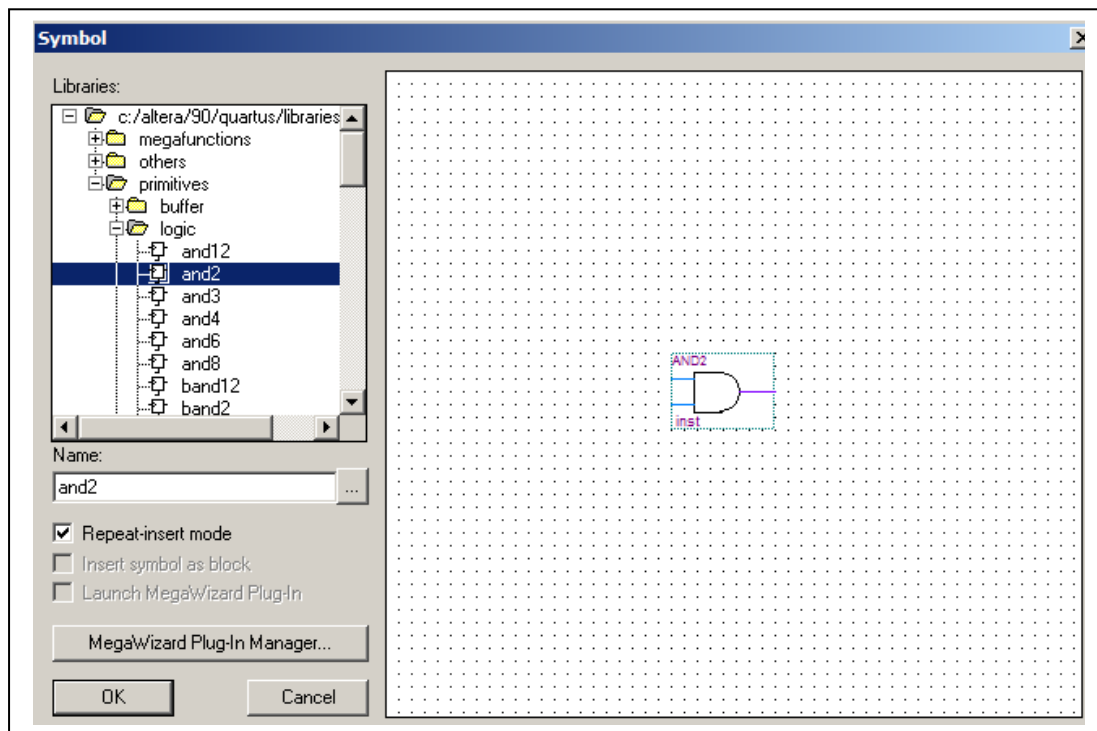
1. Création sous forme schématique : on parle alors de **Block Diagram/Schematic File** (extension **.bdf** des fichiers).
2. Création sous forme textuelle en utilisant un langage de programmation de circuits : Langage **VHDL** (extension **.vhd**) ou bien langage **Verilog** (extension **.v**, langage non enseigné à l'IOgs).

#### B.2.1- Création de fichier sous forme schématique :

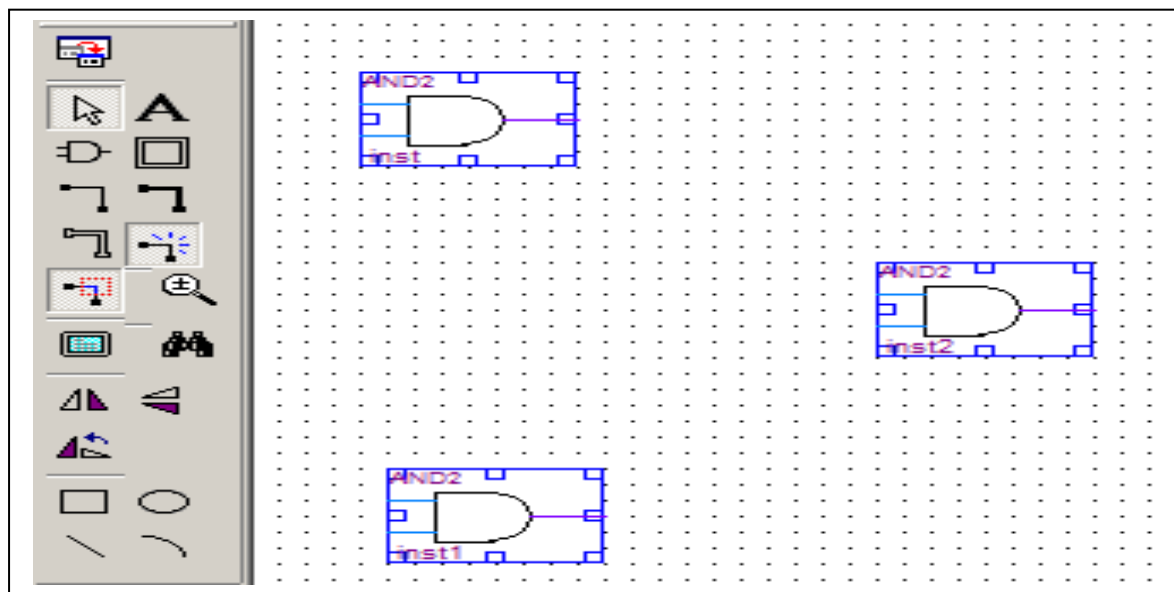
- 1- **File>New** (ou **Ctrl N**) ou icône  , puis sélectionner le type de fichier : **Block Diagram/Schematic file**.




- 2- Dans la fenêtre qui s'ouvre, sélectionner l'outil de sélection de composants  puis, dans la fenêtre **Symbol**, pour accéder par exemple, à une porte ET à 2 entrées, il faut descendre ensuite dans le menu de sélection de composants en choisissant :  
**C :/altera/ quartus/librairies/primitives/logic.**



- 3- Sélectionner le composant, puis cliquer sur **Ok**.
- 4- On bascule alors sur le schéma ; chaque clic souris dessine une nouvelle porte ; on sort de ce mode par la touche **Echap (ou Esc)**.




- 5- Pour tracer des connexions, on sélectionne l'outil  (**Orthogonal Node Tool**), et on relie les 2 terminaisons concernées entre elles.

## Annexe B

- 6- Il faut maintenant dessiner les entrées/sorties de la fonction logique (les in ou les out de la description VHDL). Pour cela sélectionner à nouveau l'outil puis, dans la fenêtre *Symbol*, choisir le chemin *C :/altera/quartus/librairies/primitives/pin* et choisir *input* pour *les entrées* ou *output* pour *les sorties*. Reprendre les mêmes opérations que pour les portes pour la fin de ce schéma.

### B.2.2- Création de fichier sous forme de fichier VHDL :

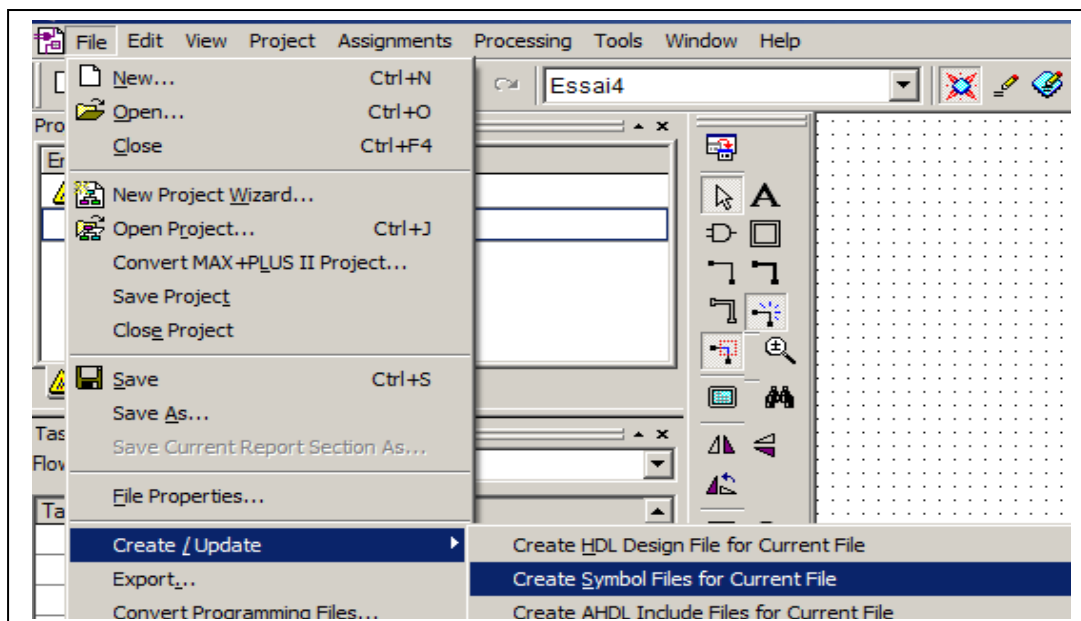
- 1- *File>New* (ou *Ctrl N*) ou icône  , puis sélectionner le type de fichier : *VHDL File*.
- 2- Tapez le code VHDL.
- 3- Pour associer les Entrées /sorties de votre code VHDL à des broches physiques du composant, le plus simple est de créer un schéma bloc associé à votre code VHDL.
- 4- Puis de relier les broches de ce schéma bloc aux broches physique.

### B.3- Schéma associé à une fonction :

*Quartus* propose une interface graphique, il est utile de pouvoir associer n'importe quelle fonction à un schéma ; cela s'applique indifféremment aux fonctions décrites sous forme de schéma, ou sous forme de code *VHDL*. On peut ensuite associer entre elles les nouvelles fonctions décrites sous forme de schémas, et ainsi de suite. On voit donc apparaître la notion de conception hiérarchique.

### B.3.1- Création de schéma associé à une fonction sous forme de fichier VHDL :


- 1- Ouvrir le fichier (*schéma ou code VHDL*).
- 2- Sélectionner *Create/Update > Create Symbol File for current File*, voir la figure ci-contre (nb : pour pouvoir créer un schéma associé à un fichier, le fichier doit être sauvegardé).
- 3- Sauvegarder le schéma (*fichier .bdf*) ainsi créé dans le dossier du projet en lui donnant un nom explicite différent du nom du fichier *VHDL* ainsi que de celui de l'entité (*entity*).



### **B.4- Compiler un projet :**

La compilation va générer la configuration du composant programmable qui permettra la réalisation concrète du projet. Pour cela :



Dans la barre de menu, choisir **Processing > Start Compilation** ou cliquer sur l'icône **Start Compilation**  , ou encore en tapant Ctrl L.

On peut ne pas lire le rapport de compilation dans la fenêtre de Messages (en bas de la fenêtre Quartus) lors des premières programmations. Si la compilation n'est pas couronnée de succès, il va falloir trouver et corriger les erreurs dans les fichiers sources. Sinon, on peut passer à la programmation du composant.

### **B.5- Simulation :**

La simulation est l'étape qui vous permet de valider un design avant de le programmer sur la carte. Il est en effet souvent bien plus facile de trouver une erreur sur des résultats de simulation plutôt que sur le circuit. De plus, programmer une configuration de circuit qui ne correspond pas à ce que vous deviez concevoir est une perte de temps. En dehors des cas les plus simples, il faut donc simuler après avoir compilé et avant de programmer. Pour simuler, vous allez devoir définir précisément le type de simulation voulu, la durée de cette simulation, les entrées et les sorties que vous voulez observer.

#### **Créer un fichier de vecteurs de test :**

- **File->New->Vector Waveform File**
- **OK**
- Enregistrer le fichier (*extension.vwf*)

#### **Durée de la simulation :**

Par défaut : **1 $\mu$ s**

#### **Pour la modifier :**

- **Edit->End Time...**

#### **Grille de simulation :**

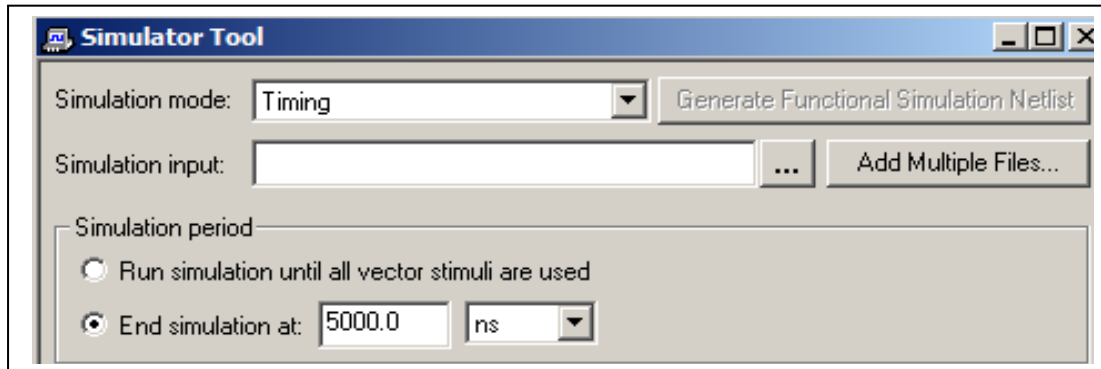
Par défaut : **10 ns**

#### **Pour la modifier :**

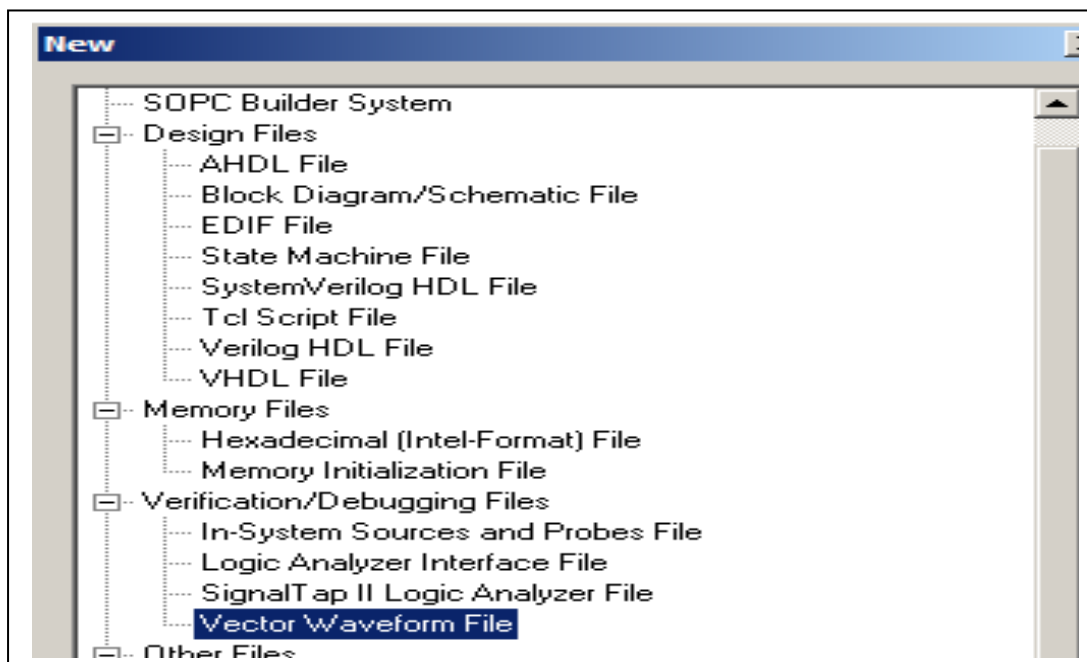
- **Edit->Grid Size**

### B.6- Configurer le simulateur :

- 1- Lancer le simulateur par le menu **Processing > Simulator Tool**.
- 2- Compléter les champs Simulation mode et End Simulation. Au niveau des modes de **simulation**, **Functionnal** correspond à une simulation avec des portes idéales, et **Timing** à une simulation en tenant compte des retards technologiques. On vous conseille de choisir ce dernier mode.

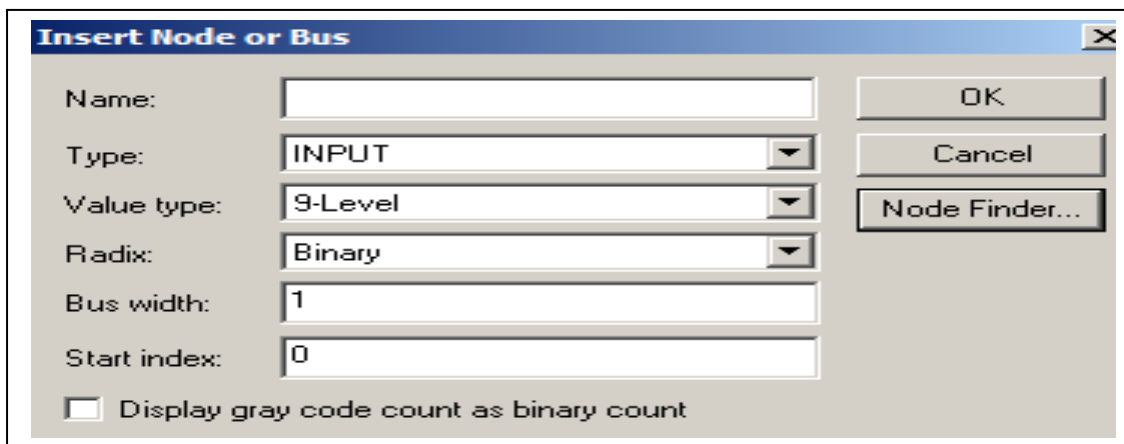
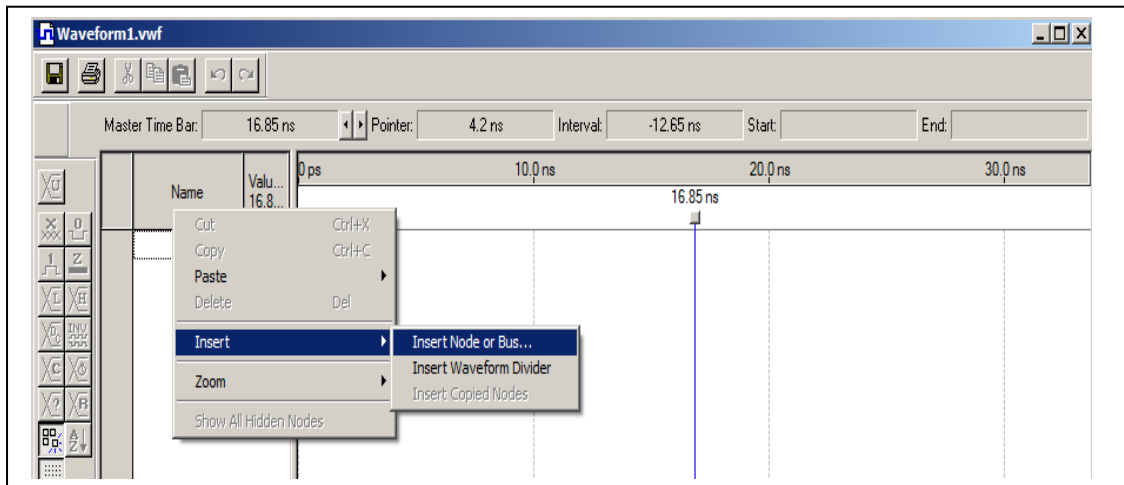


- 3- Il va falloir créer un fichier de vecteur de tests. On va pour cela, aller dans **File > New (ou Ctrl N)** et sélectionner **Vector Wave File**.

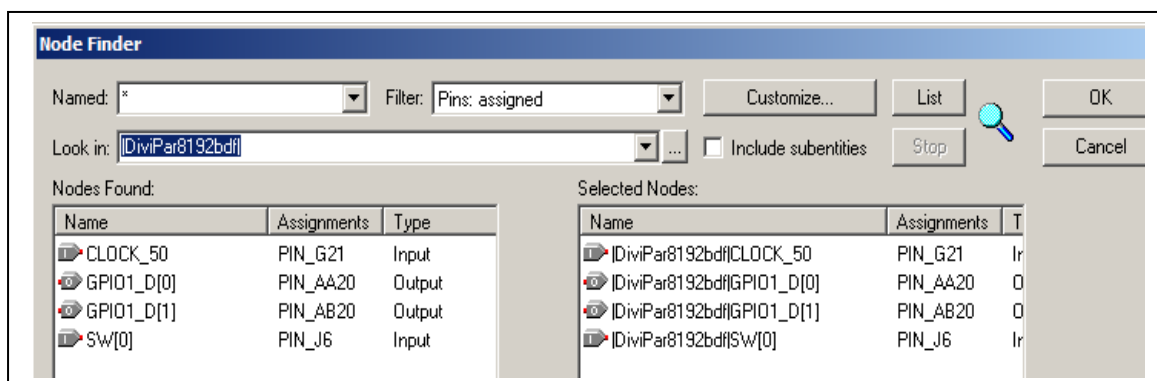


- 4- Dans la nouvelle fenêtre, clic droit sur **Name** pour ajouter des signaux, puis sélectionner **Insert > Insert Node or Bus** et sélectionner le menu **Node Finder**.

## Annexe B

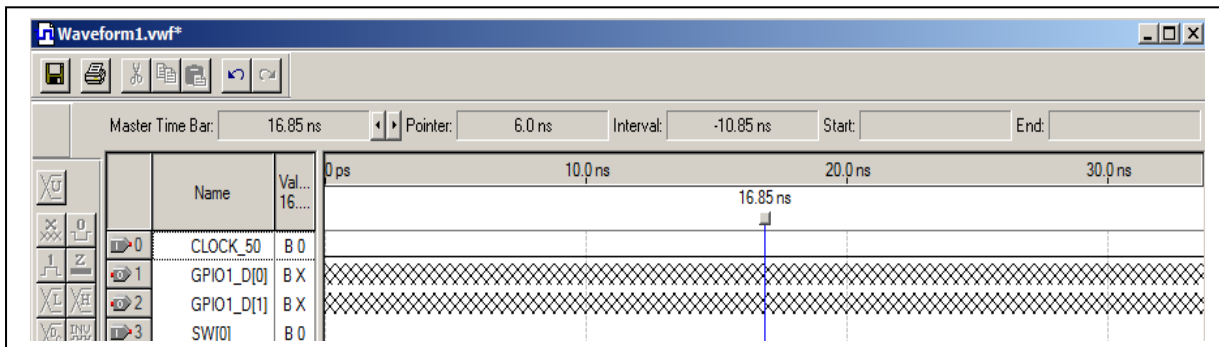


- 5- On ajoute les signaux qui nous intéressent en les faisant apparaître par **double-clic** (ou encore, **par sélection puis >**).

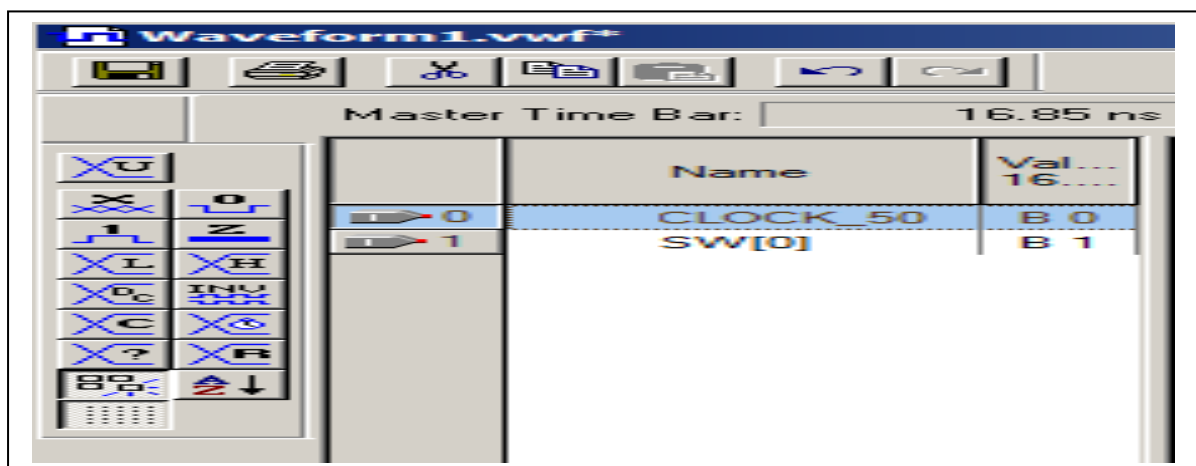


## Annexe B

L'interface du simulateur s'ouvre alors :



- 6- Il faut maintenant imposer une valeur pour chaque entrée. On sélectionne un signal, puis les icônes de la colonne de gauche passent en *bleu*, et on impose des valeurs, ou bien par *Clic Droit*, on choisit dans le menu *contextuel*.



### *Quelques remarques :*

- Pour imposer un 0 ou un 1 pendant juste une durée, sélectionner cette durée dans le panneau de droite puis cliquer sur l'icône associée dans le menu.
- Pour l'horloge, sélectionner le signal voulu puis cliquer sur l'icône de signal périodique.
- Quelques raccourcis clavier du simulateur :

**Ctrl W** : affiche tout l'intervalle de simulation.

**Ctrl Espace** : Zoom.

**Ctrl Shift Espace** : Dézoom.

## Annexe B

### Lancer la simulation

- Lancer la simulation par l'icône **Start** du **Simulator Tool**.
- Pour observer les résultats, cliquer sur **Report**.
- Pour modifier le fichier de vecteur de tests, cliquer sur les résultats, cliquer sur **Open**.



### B.8- Configurer le composant au Programmation :

Pour configurer le composant afin d'obtenir le fonctionnement décrit par le projet, il faut impérativement :

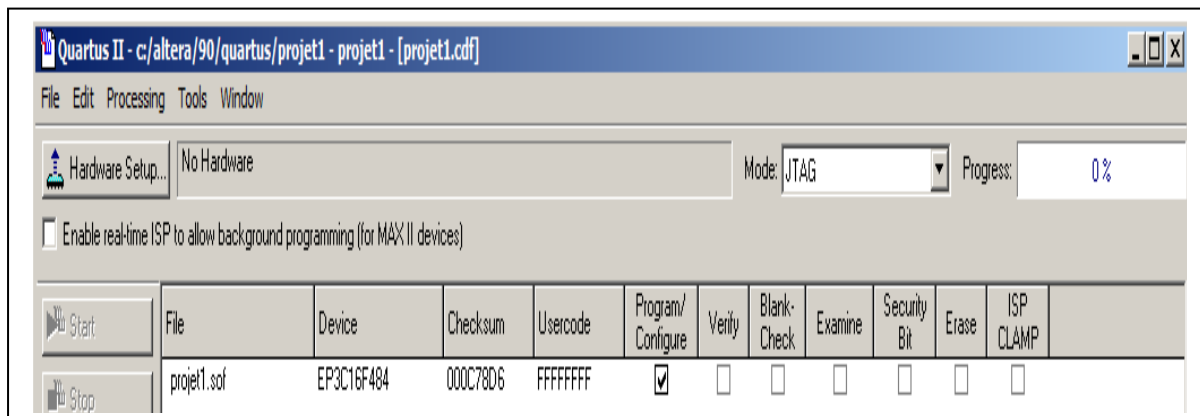
- que la maquette **DE2** soit reliée à l'ordinateur par câble **USB**.
- qu'elle soit sous tension (**bouton rouge**).
- que le commutateur **RUN / PROG** soit positionné sur **RUN**.

### Pour configurer le composant de la maquette DE2 :

Sélectionner l'icône de programmation ou bien Sélectionner **Tools > Programmer**.

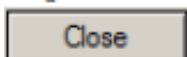


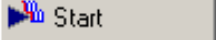
- 1- La fenêtre de "**programmation**" s'ouvre alors.



- 2- Cliquer sur **Hardware Setup**
- 3- Puis dans la nouvelle fenêtre sélectionner **USB-Blaster** à la place de **No Hardware** dans le menu **Currently Selected Hardware**.

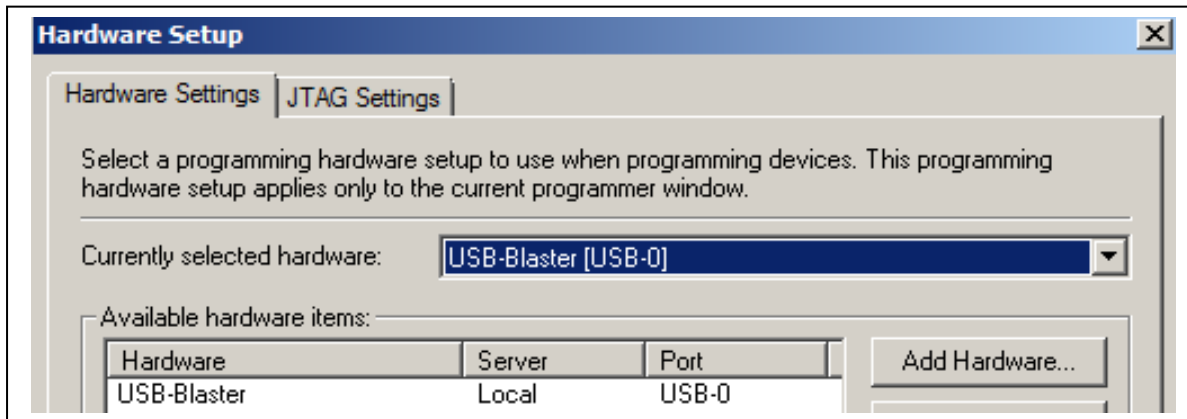
Fermer cette fenêtre



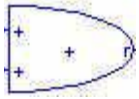
- 4- Enfin, cliquer sur  pour lancer la configuration du circuit de la maquette **DE2**.

## Annexe B

---



# Annex C



**Parallel Adder Subtractor** : Le bloc parallèle d'additionneur soustracteur prend n'importe quel type de données d'entrée. Si les largeurs d'entrée ne sont pas identiques, le signe de compilateur de signal prolonge les autobus de sorte qu'ils assortissent la plus grande largeur d'entrée.

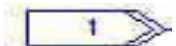


**Product** : Le bloc de produit soutient deux entrées scalaires (aucuns signaux multidimensionnels de Simulink). L'opérande a est multiplié par l'opérande b et le résultat produit sur r comme montré par l'équation suivante :

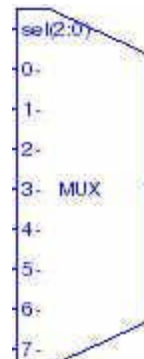
$$r = a \times b$$



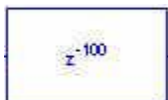
**Gain** : Le bloc de gain produit toujours en sortie. Vous devez écrire le gain comme une valeur numérique dans le domaine de paramètre de bloc de gain. Le facteur de gain doit être une grandeur scalaire.



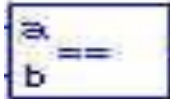
**Constant** : Le bloc constant spécifie un autobus constant.



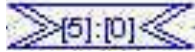
**Multiplexer** Le bloc de multiplexeur fonctionne comme multiplexeur binaire de n-to-1 d'autobus avec une commande choisie. La largeur de rendement du multiplexeur est égale à la largeur maximum des lignes de données d'entrée. Le bloc travaille sur n'importe quel type de données et le signe prolonge les entrées s'il y a une disparité de largeur de peu.



**Delay** : Le bloc de Delay retarde les données entrantes par une quantité spécifique par le nombre d'étapes de canalisation. Le bloc accepte n'importe quel type de données comme entrées

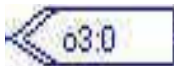


**Comparator** : Le bloc de comparator compare deux signaux de Simulink et renvoie un à bit unique. Le bloc comprend implicitement le type de données d'entrée (par exemple, binaire signée ou entier sans signe) et produit un résultat à bit unique.

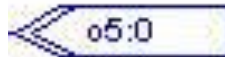


**AltBus** : Le bloc d'AltBus est employé pour modifier le format d'autobus d'un signal de DSP-Builder. Il peut seulement être employé comme nœud interne dans un système, pas comme entrée ou sortie.

Employer les paramètres de bloc pour spécifier la largeur désirée de peu de rendement. Si c'est plus large que l'entrée a mordu la largeur, l'autobus est signe prolongé à l'ajustement. S'il est plus petit que l'entrée a mordu la largeur, vous pouvez choisir à tronquez ou saturez le peu excessif.



**Output** : Le bloc output est employé pour définir la frontière de rendement d'une exécution de matériel et pour mouler le format partiel de binaire signée (des blocs de constructeur de DSP) aux signaux à point mobile de Simulink (blocs génériques d'alimentation de Simulink).



**Nonsynthesizable Output** : Le bloc Non-synthesizable Output est employé pour marquer un point de sortie d'un système non-synthesizable de DSP-Builder et devrait être employé en même temps que le bloc Non-synthesizable Input. Puisque DSP-Builder enregistre son propre type avec Simulink, ce bloc est exigé quand les blocs de DSP-Builder ne sont pas prévus pour être synthétisés. Ce bloc peut également être employé pour créer un résultat non-synthesizable d'un système synthesizable.

# Bibliographie

## Bibliographie

---

- [1] A.NKESTA, « Informatique industrielle, circuits logiques programmable, mémoires, PLD, CPLD et FPGAs » Edition Ellipses, 1998.
- [2] B.Farida, D.Sabrina. Système temps réel embarque pour commander un robot mobile. Mémoire de Fin d'Etudes, Automatique. Alger : Ecole Nationale Polytechnique (ENP), 2005 .
- [3] D.Houssey. Implémentation d'une commande MPPT floue sur FPGA. Mémoire de Fin d'Etudes, Electronique. Alger : Ecole Nationale Polytechnique (ENP), 2006.
- [4] D.Boutat, <<la notion d'observateur pour les systèmes non linéaires>>, ENSI de Bourges institut prisme, (2010).
- [5] D.Ruelle, <<Hasard et chaos>>, Odile Jacob sciences, (1991).
- [6] D.Houssey. Implémentation d'une commande MPPT floue sur FPGA. Mémoire de Fin d'Etudes, Electronique. Alger : Ecole Nationale Polytechnique (ENP), 2006.
- [7] Eduardo Sanchez, << Les circuits logiques programmables>>, Ecole Polytechnique Fédérale de Lausanne.
- [8] Eduardo Sanchez,<< Circuits reconfigurables: Les FPGAs>>.
- [9] FPGA HardCore Single Processor Implementation of RT Control Applications.
- [10] Gang ZHENG. << Formes Normales d'Observabilité Paramétrées par les Sorties : Applications au Cryptage par Synchronisation de Systèmes Chaotiques>>. Thèse de doctorat. L'université de Cergy-Pontoise(2006).
- [11] G Florin. << LES TECHNIQUES DE CRYPTOGRAPHIE>>.
- [12] G. Zheng, <<Formes normales d'observabilité paramétrées par la sorties : Applications cryptage par synchronisation de système chaotiques>>. Thèse de doctorat. Université de Cergy-Pontoise (2006).
- [13] Guoyuan Qi, Michael Antonie van Wyk, Barend Jacobus Van Wyk, Guanrong Chen, <<A new hyperchaotic system and its circuit implementation >>, Chos solutions & fractals,pages (254-2549),(2007).
- [14] H.Hamiche, « Inversion à gauche des systèmes dynamiques hybrides chaotiques. Application à la transmission sécurisée de donnée », Thèse doctorat, UMMTO, Algérie, (2011).
- [15] I.Ameur, << synchronisation, chaotification et hyperchaotification des systèmes non linéaires : Méthodes et Application>>, thèse doctorat, Université Mentouri de Constantine, Algérie, (2011).
- [16] J.M Ginoux, <<le chaos en quelques mots .... >>, Université du sud, la Garde Cedex, France.

## Bibliographie

---

- [17] M. Walid, M.I.Moctar. Implémentation d'un modulateur OFDM sur un circuit FPGA. Mémoire de Fin d'Etudes, Electronique. Alger : Ecole Nationale Polytechnique (ENP), 2007.
- [18] M.B.Luca, <<apport du chaos et des estimateurs d'états pour la transmission sécurisée de l'information>>, thèse doctorat. Université de Bretagne occidentale. (2008).
- [19] Mihai Bogdan Luca. << Apports du chaos et des estimateurs d'états pour la transmission sécurisée de l'information>>. Thèse de doctorat. Laboratoire d'Electronique et Systèmes de Télécommunications LEST - UMR CNRS 6165. Université de Bretagne Occidentale - Ecole Doctorale SMIS(2006).
- [20]M. AIB Abdelghani, <<Conception et implémentation d'un algorithme de vision artificielle sur FPGA>>, Mémoire de Fin d'Etudes, Automatique. Alger : Ecole Nationale Polytechnique (ENP), 2008 .
- [20] N.R.Galagher, « Système de communication haut débit sécurisé par chaos et intensité », Thèse doctorat, Université de Franche-Comté, (2006).
- [21] N.JULIEN. Circuits logiques programmables. *Université de Bretagne Sud. 1999.*
- [22] Paul Manneville, << Dynamique non linéaire et chaos>>, Laboratoire d'Hydrodynamique, Ecole Polytechnique.
- [23] Philippe Etchecopar et Cégep de Rimouski, << Quelques éléments sur la théorie du chaos>>.
- [24] P.Bérgé, Y.Pomeau et C.Vidal,<< L'ordre dans le chaos>>.Hermann, éditeur des sciences et des arts(1988).
- [25] Sundarapandian Vaidyanathan. <<Global chaos synchronization of hyperchaotic Qi and hyperchaotic JHA systems by avtive nonlinear>>.
- [26] T.L.Carrol and L.M.Pocora, Synchronisation in chaotic systems, Physicals review and letters, 1990.

## Bibliographie

---

Résumé : Dans ce travail, la transmission sécurisée de donnée à été étudié et appliquée sur une carte électronique de type FPGA. Ici, la transmission sécurisée à été faite a base d'un système émetteur et d'un système récepteur.

L'émetteur est composé d'un système hyperchaotique le système de Qi, dans lequel le message est inséré par la méthode additive. Le récepteur est un système identique à l'émetteur est un observateur à retour d'état, dans le quel le message à été récupérer avec une opération de soustraction.

Le but de ce travail, est de savoir les différentes méthodes de synchronisation et ses buts, ainsi les différentes méthodes de cryptage et aussi l'utilité des FPGAS.

Dans la partie pratique la réalisation à été faite à base de logiciel DSP-Builder de ALTERA. Et les résultats de simulation sont présentés pour illustrer les performances de la méthode proposée.

**Mots-Clés** : Chaos, Synchronisation, Observateurs, Cryptage, QI, FPGA, DSP-Builder.