

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Mention Électronique

Spécialité Microélectronique Microsystème et Conception Circuits

présenté par

Kacimi El Hassani Nour El Houda

Etude et implémentation FPGA d'un système chaotique Auto commuté

Proposé par : Mr Chikhi Mohamed Lazhar

Année Universitaire 2016-2017

D é d i c a e e

*A la mémoire de mon frère « Alaeddine » paix à son âme, qui
m'a toujours encouragé et motivé dans mes études. Qu'ALLAH
l'accueille en son vaste paradis.*

Puisse Dieu, le tout puissant, l'avoir en sa sainte miséricorde.

Remerciements

*Je tiens à présenter ma profonde reconnaissance et mes remerciements à mon encadreur Monsieur «**Lazhar Mohamed CHIKHI**» pour l'aide compétente qu'il m'a apporté, pour sa patience et ses encouragements. Son aide m'a été très précieuse pour structurer ce travail et pour améliorer la qualité des différentes sections.*

Je remercie toutes les personnes qui m'ont apporté leurs aides, leurs soutiens et leurs accompagnements dans la réalisation de ce mémoire : elles se reconnaîtront...

*A mon collègue et mon ami le futur ingénieur informatique «**Adel Bounssairi**» qui me vient toujours en aide lorsque j'en ai besoin.*

*A la fin, je souhaite exprimer mon GRAND MERCI à «**mes parents** », Mes sœurs «**Wafaa**», «**Hanine**», «**Fadoua**» pour leur affection, leur compréhension et leur patience.*

A tous ceux qui ont une relation de près ou de loin avec la réalisation du présent mémoire.

A ma famille ...

A mes amis ...

ملخص:

يتمثل هذا العمل في توليد إشارة فوضوية يقدمها نظام التحويل التلقائي الفوضوي. بعد وصف السمات الرئيسية للأنظمة الديناميكية الفوضوية، تم تقديم تحليل مفصل لنظامنا التحويلي التلقائي الفوضوي. وقد أجريت دراسة محاكاة لتحليل سلوك هذا النظام في MATLAB-Simulink. أكملنا عملنا بالجزء التجريبي، القائم على تطبيق FPGA على بطاقة Xilinx Virtex-5 ML501 التي تُتيح عرض إشارات مختلفة في الوقت الحقيقي، ومقارنتها مع نتائج المحاكاة. وأظهرت النتائج التجريبية اتفاق جيد مع النتائج التي تم الحصول عليها في المحاكاة.

كلمات المفاتيح : أنظمة الديناميكية الفوضوية، نظام التحويل التلقائي الفوضوي، بطاقة FPGA.

Résumé : Ce travail consiste à générer un signal chaotique fourni par un système chaotique Auto-commuté. Après une description des principales caractéristiques des systèmes dynamiques chaotique, l'analyse détaillée de notre système chaotique est présentée. Une étude en simulation a été effectuée sous Matlab-Simulink pour analyser le comportement du système. Nous avons finalisé notre travail par une partie expérimentale, basée sur une implémentation FPGA sur carte Xilinx Virtex-5 ML501 permettant la visualisation des différents signaux en temps réel et pour pouvoir les comparer avec les résultats de simulation. Les résultats expérimentaux ont montré une bonne concordance avec les résultats obtenus lors de la simulation.

Mots clés : système dynamique, système chaotique Auto-commuté, circuit FPGA.

Abstract:

This work consists in generating a chaotic signal provided by a chaotic system Auto-switched. After a description of the main characteristics of dynamic chaotic systems, the detailed analysis of our Auto-switched chaotic system is presented. A simulation study was carried out under MATLAB-Simulink to analyze the behavior of the system. We finalized our work with an experimental part, based on an FPGA implementation on card Xilinx Virtex-5 ML501 allowing the visualization of the various signals in real time and to be able to compare them with simulation results. The experimental results showed a good agreement with the results obtained during the simulation.

Keywords: Dynamic system, Auto-switched chaotic system, FPGA card.

Introduction générale

Les concepts du chaos et sa phénoménologie traversent aujourd'hui une large variété de champs de recherche. Le chaos est conventionnellement défini par un comportement lié à l'instabilité et la non-linéarité dans les systèmes dynamiques déterministes non-intégrables. La relation entre l'instabilité et le chaos est que le système manifeste une très haute sensibilité aux changements de conditions initiales.

C'était en 1963, quand le météorologue Edward Lorenz [1] a décrit un modèle mathématique simple composé de trois équations différentielles non linéaires très sensible aux conditions initiales. Lorenz a appelé ces systèmes présentant la dépendance sensible sur les conditions initiales comme avoir de «l'effet papillon» : ce nom unique est venu de la proposition qu'un papillon qui bat ses ailes à Hong Kong peut affecter le cours d'une tornade au Texas [2].

La théorie du chaos étudie donc en détail comment une petite incertitude sur l'état initial d'une évolution temporelle déterministe peut donner lieu à une incertitude des prédictions qui croît rapidement avec le temps. Le comportement chaotique peut être étudié grâce à l'analyse par des modèles mathématiques chaotiques, ou par des techniques analytiques de récurrence et des applications de Poincaré. Le chaos est utilisé dans de très nombreux domaines : en météorologie, sociologie, physique, informatique, ingénierie, économie, biologie... etc.

Dans le cadre de notre projet, nous proposons d'étudier et d'analyser un système chaotique Auto-commuté, et d'en déterminer toutes ses propriétés en vue de son implémentation sur circuit FPGA. Différents outils de développement seront utilisés (Matlab-Simulink, System-Generator de Xilinx, ISE de Xilinx) afin d'effectuer les différentes simulations pour tester la validation de l'implémentation sur la carte FPGA Virtex-5 ML501 de Xilinx.

Pour cela, nous avons organisé notre mémoire de la manière suivante :

- Le premier chapitre présente un rappel sur les systèmes dynamiques en général et chaotiques en particulier. Il énoncera quelques concepts sur la théorie du chaos, les principales caractéristiques des systèmes dynamiques chaotiques et leurs méthodes d'analyse.
- Le second chapitre consiste à faire l'étude et l'analyse du comportement du système chaotique Auto-commuté à l'aide des outils de simulation.
- Dans le troisième chapitre, on va présenter un aperçu sur la technologie FPGA, dans laquelle on implémentera notre système chaotique Auto-commuté en utilisant l'outil System Generator sous MTLAB (Simulink), qui grâce à la bibliothèque XILINX, servira à générer le programme en langage VHDL.
- Le quatrième et dernier chapitre est consacré à la partie expérimentale de l'implémentation du système chaotique Auto-commuté sur la carte FPGA Virtex-5 ML501 de Xilinx, et l'utilisation des différents outils de développement permettant d'effectuer les différentes simulations, en vue d'une comparaison des différents signaux obtenus lors de la simulation et des signaux relevés au niveau de la carte FPGA.

On termine par une conclusion générale et quelques perspectives d'avenir.

Table des matières

Introduction générale	1
Chapitre 1 Introduction aux systèmes dynamiques chaotiques	
1.1 Introduction	3
1.2 Les systèmes dynamiques.....	4
1.2.1 Représentations mathématiques des systèmes dynamiques.....	4
1.3 La théorie du chaos.....	5
1.3.1 Définition du chaos.....	5
1.3.2 Applications du chaos.....	5
1.3.3 Domaine d'application	6
1.3.4 Caractéristiques du chaos	6
1.4 Espace des phases.....	8
1.5 Point fixe :	9
1.5.1 Stabilité des points fixes	9
1.6 Exposants de Lyapunov	11
1.6.1 Comportement du système en fonction des exposants de.....	12
Lyapunov	12
1.7 Section de Poincaré.....	13
1.8 Bifurcation.....	14
1.8.1 Diagramme de bifurcation.....	14
1.9 Exemples de systèmes chaotiques	15
1.9.1 Système de Lorenz.....	16
1.9.2 Système de Rössler.....	18
1.10 Conclusion.....	20
Chapitre 2 Etude du système chaotique Auto-commuté	
2.1 Introduction	21

2.2	Description du système.....	21
2.3	Analyse du système Auto-commuté.....	23
2.3.1	Les propriétés de base du système Auto-commuté.....	23
2.3.2	Equilibre et stabilité.....	24
2.3.3	Évolution du système Auto-commuté en fonction du temps.....	25
2.3.4	Sensibilité aux conditions initiales.....	28
2.3.5	Plan de phase.....	28
2.3.6	Attracteur étrange.....	30
2.3.7	Exposants de Lyapunov.....	30
2.3.8	Section de Poincaré.....	33
2.3.9	Bifurcation.....	34
2.4	Conclusion.....	37
Chapitre 3 Implémentation FPGA du système chaotique Auto-commuté		
3.1	Introduction.....	38
3.2	Présentation des circuits FPGAs.....	38
3.2.1	Architecture des FPGAs.....	39
3.2.2	Les outils de développement des circuits FPGAs.....	40
3.3	Présentation des différents outils de travail.....	40
3.3.1	Présentation du logiciel ISE.....	40
3.3.2	Système Generator.....	41
3.3.3	Présentation du logiciel de simulation ModelSim.....	42
3.3.4	Plate- forme de développement Virtex-5 (ML501).....	43
3.4	Processus d'implémentation.....	44
3.5	Implémentation FPGA du système chaotique Auto-commuté.....	45
3.5.1	Simulation sous System Generator- Simulink.....	45
3.5.2	Compilation du System Generator.....	47
3.5.3	Implémentation sur FPGA.....	48

3.6	Conclusion.....	49
Chapitre 4 Interface codec AC97 et résultats expérimentaux		
4.1	Introduction	50
4.2	Le codec AC97	50
4.3	Principe de fonctionnement du CODEC AC97	51
4.4	Expérimentation et résultats	53
4.4.1	Visualisation des signaux.....	55
4.5	Conclusion.....	59
Conclusion générale.....		
		60

Liste des figures

Figure 1.1 : Sensibilité aux conditions initiales d'un système chaotique.....	7
Figure 1.2 : Evolution dans le temps d'un système chaotique, comparé à une sinusoïde.	8
Figure 1.3 : Exemple d'un espace des phases.	8
Figure 1.4 : Exemple des exposants de Lyapunov d'un système chaotique.	12
Figure 1.5 : Principe de section de Poincaré.	13
Figure 1.6 : Exemple de diagramme de bifurcation.	15
Figure 1.7 : Attracteur étrange de Lorenz.	16
Figure 1.8 : Sensibilité aux conditions initiales du système de Lorenz.	16
Figure 1.9 : Les plans de phases du système de Lorenz.	17
Figure 1.10 : Diagramme de bifurcation du système de Lorenz.	17
Figure 1.11 : Attracteur étrange de Rössler.	18
Figure 1.12 : Sensibilité aux conditions initiales du système de Rössler.	19
Figure 1.13 : Les plans de phase du système de Rössler.....	19
Figure 1.14 : Diagramme de bifurcation du système de Rössler.	20
Figure 2.1 . Attracteur chaotique du sous-système (2.2).....	22
Figure 2.2 . Attracteur chaotique du sous-système (2.3).....	23
Figure 2.3 . Représentation du système chaotique Auto-commuté sous MATLAB-Simulink.....	26
Figure 2.4 . L'état x en fonction du temps.....	26
Figure 2.5 . L'état y en fonction du temps.....	27
Figure 2.6 . L'état z en fonction du temps.....	27
Figure 2.7 . Les états x, y et z en fonction du temps.....	27
Figure 2.8 . La propriété de sensibilité aux conditions initiales sur l'état x	28
Figure 2.9 . Plan de phase x en fonction de y	28

Figure 2.10. Plan de phase x en fonction de z	29
Figure 2.11. Plan de phase y en fonction de z	29
Figure 2.12. Attracteur étrange du système Auto-commuté en fonction de x, y et z ..	30
Figure 2.13. L'interface de l'outil MATDS.....	31
Figure 2.14. Création du système (2.2) sur MATDS	31
Figure 2.15. Les exposants de Lyapunov du sous-système (2.2).....	32
Figure 2.16. Les exposants de Lyapunov du sous-système (2.3).....	32
Figure 2.17. La section de Poincaré en 2D des deux sous –systèmes (2.2) et (2.3).....	33
Figure 2.18. La section de Poincaré en 3D des deux sous –systèmes (2.2) et (2.3).....	34
Figure 2.19. Section de Poincaré sans trajectoires des sous-systèmes (2.2) et (2.3) ...	34
Figure 2.20. Le diagramme de bifurcation du système Auto-commuté.	35
Figure 2.21. Les états x, y, z en fonction du temps lorsque $h = 9$	35
Figure 2.22. Le plan de phase x en fonction de y pour $h = 9$	36
Figure 2.23. Le plan de phase x en fonction de z pour $h = 9$	36
Figure 2.24. Le plan de phase y en fonction de z pour $h = 9$	36
Figure 3.1. Architecture générique d'un circuit FPGA.....	39
Figure 3.2. Structure de bloc logique configurable (CLB).....	39
Figure 3.3. L'interface Project Navigator de l'ISE 14.2	41
Figure 3.4. La bibliothèque Xilinx de Simulink.....	42
Figure 3.5. Le bloc System Generator	42
Figure 3.6. Interface graphique du logiciel ModelSim de Mentor Graphics.....	42
Figure 3.7. Plateforme de développement Virtex-5-ML501	43
Figure 3.8. La simulation du système chaotique Auto-commuté sous System Generator	45
Figure 3.9. Le bloc de l'intégrateur sous Système Generator	45
Figure 3.10. Plan de phase xy	46

Figure 3.11. Plan de phase xz	46
Figure 3.12. Plan de phase yz	46
Figure 3.13. Attracteur étrange xyz	46
Figure 3.14. Les courbes des états x, y et z en fonction du temps.....	47
Figure 3.15. Fenêtre des paramètres System Generator	47
Figure 3.16. Représentation des signaux du système chaotique Auto-commuté sous Modelsim	48
Figure 3.17. Fenêtre de synthétisation	49
Figure 4.1. Circuit AD1981B Codec audio.....	51
Figure 4.2. Connexions du contrôleur au CODEC AC97.	52
Figure 4.3. Codec AC97 sous System Generator.	53
Figure 4.4. Interface permettant la programmation du FPGA.	54
Figure 4.5. Implémentation du système chaotique Auto-commuté sur la carte FPGA.	54
Figure 4.6. Les états x, y de en fonction du temps.	55
Figure 4.7. Les états x, z de en fonction du temps.....	55
Figure 4.8. Les états y, z de en fonction du temps.	56
Figure 4.9. Plan de phase xy	56
Figure 4.10. Plan de phase xz	56
Figure 4.11. Plan de phase yz	56
Figure 4.12. Plan de phase xy sur l'oscilloscope numérique et analogique.....	57
Figure 4.13. Plan de phase xz sur l'oscilloscope numérique et analogique.....	57
Figure 4.14. Plan de phase yz sur l'oscilloscope numérique et analogique.....	58
Figure 4.15. Ressources consommées par l'implémentation.	58

Listes des acronymes et abréviations

R^n : Ensemble des nombres réels entiers.

R^p : Ensemble des nombres réels positifs .

x_0 : L'état initial de la variable x .

x_k : L'état de x au temps $t = kT$.

x_{k+1} : L'état de x au temps $t = (k + 1)T$.

Λ_i : Exposant de Lyapunov d'ordre i .

\bar{x} : Point fixe.

FPGA: Field Programmable Gate Array.

ISE: Integrated Software Environment.

VHDL: Very High Description Language.

VLSI: Very Large Scale Integration.

CLB: Configurable Logic Bloc.

IOB: Input Output Bloc.

LUT: Look Up Table.

CPLD: Complex Programmable Logic Device.

DSP : Digital Signal Processor.

AC97 : Audio Codec 97.

HDL : Hardware Description Language.

ASIC : Application Specific Integrated Circuit.

AD : Analogic Digital.

DA : Digital Analogic.

NETLIST: les interconnexions entre les composants qui interviennent avant le routage.

Chapitre 1 Les systèmes dynamiques chaotiques

1.1 Introduction

Depuis longtemps, le chaos était synonyme de désordre et de confusion. Il s'opposait à l'ordre et devait être évité. La science était caractérisée par le déterminisme, la prévisibilité et la réversibilité. L'existence du chaos a été formalisée d'un point de vue théorique et observée dans de nombreuses expériences, les mouvements chaotiques ne sont ni périodiques ni permanents, ce qui complique leur analyse et incite à les qualifier de turbulent ou aléatoires... Cependant, ces mouvements sont très communs et peuvent apparaître même dans les systèmes simples.

Des chercheurs ont alors commencé à s'intéresser à ce comportement, Poincaré fut l'un des premiers à entrevoir la théorie du chaos. Il découvrit la notion de sensibilité aux conditions initiales à travers le problème de l'interaction de trois corps célestes [3].

Cette partie est consacrée aux définitions des différents outils mathématiques qui nous servent à mieux comprendre le comportement chaotique, tels que : les ensembles attracteurs, les méthodes permettant la détermination du type de stabilité de leur état d'équilibre, la section de Poincaré, la théorie de bifurcation, les exposants de Lyapunov.

Le but de ce chapitre est de présenter les conditions nécessaires pour qu'un système ait un comportement chaotique.

1.2 Les systèmes dynamiques

Un système dynamique est la représentation mathématique d'un phénomène observé (naturel ou artificiel) en termes d'équations différentielles décrivant son évolution temporelle. En mathématiques, en physique, en ingénierie un système dynamique est un système «classique» qui évolue au cours du temps de façon à la fois :

- causale (c.à.d. que son avenir ne dépend que des phénomènes du passé ou du présent).
- déterministe (c.à.d. qu'à une "condition initiale" donnée à l'instant présent va correspondre à chaque instant ultérieur à "un et un seul état futur" possible) [4].

L'évolution déterministe du système dynamique peut alors se modéliser de deux façons distinctes :

- une évolution continue dans le temps représentée par une équation différentielle non linéaire ordinaire.
- une évolution discrète dans le temps, l'étude théorique de ces modèles discrets est fondamentale, car elle permet de mettre en évidence des résultats importants qui se généralisent souvent aux évolutions dynamique continues. Elle est représentée par le modèle général des équations au différences finies.

1.2.1 Représentations mathématiques des systèmes dynamiques

Un système dynamique décrit par une fonction mathématique présente deux types de variables : dynamiques et statiques : les variables dynamiques sont les quantités fondamentales qui changent avec le temps, les variables statiques, encore appelés paramètres du système, sont fixes.

A temps continu, le système dynamique non linéaire est défini par une équation différentielle [5] :

$$\frac{dx}{dt} = \dot{x} = f(x, t, p) \quad (1.1)$$

Où : $x \in U \subseteq R^n$ est un vecteur de dimension n représentant l'état du système et $p \in V \subseteq R^p$ représente l'entrée du système (paramètre). R^n est appelé l'espace des phases et R^p l'espace des paramètres.

A temps discret, le système dynamique est représenté par une application itérative [4] :

$$x_{k+1} = f(x_k; p); x_k \in R^n, \text{ et } p \in R^p; k = 1, 2, 3 \dots \quad (1.2)$$

Un système dynamique non linéaire est dit autonome lorsqu'il ne dépend pas explicitement du temps. Un système autonome est décrit par les équations suivantes :

$$\begin{cases} \dot{x} = f(x, y) \\ \dot{y} = g(x, y) \end{cases} \quad (1.3)$$

On peut toujours transformer un système autonome en un système non autonome par un changement de variables.

1.3 La théorie du chaos

1.3.1 Définition du chaos

Le chaos tel que le scientifique le comprend ne signifie pas l'absence d'ordre ; on dit qu'un système a un comportement chaotique quand la sortie du système est extrêmement sensible aux conditions initiales du fait que l'état final dépend de manière si sensible de l'état initial. C'est un système non linéaire qui fait preuve de comportements imprévisibles à long terme, qui peut même sembler aléatoire, alors qu'il s'agit de système parfaitement déterministe.

On appelle donc un système dynamique chaotique, un système qui dépend de plusieurs paramètres et caractérisé par une extrême sensibilité aux conditions initiales.

Remarque : il faut au moins trois variables d'état pour générer le chaos pour un système continu.

1.3.2 Applications du chaos

- **Contrôle** : La première application du chaos est le contrôle du comportement irrégulier dans les circuits et les systèmes.

- **Synchronisation** : Communication sécurisée, cryptage, radio.

- **Traitement d'information** : Traitement d'information Codage, décodage et stockage d'information dans des systèmes chaotiques, tel que les éléments de mémoires et les circuits, Reconnaissance de forme.
- **Prédiction à court terme** : Les maladies contagieuses, température, économie [6].

1.3.3 Domaine d'application

Les domaines d'applications du chaos sont très nombreux, on peut citer principalement les domaines suivants [6] :

- **Ingénierie** : Contrôle de vibration, stabilisation des circuits, réactions chimiques, turbines, étages de puissance, lasers, combustion, et beaucoup plus.
- **Ordinateurs** : Commutation des paquets dans des réseaux informatiques, cryptage, et contrôle du chaos dans les systèmes robotiques.
- **Communications** : Compression et stockage d'image, conception et management des réseaux d'ordinateurs.
- **Médecine et biologie** : Cardiologie, analyse du rythme du cœur (EEG), prédiction et contrôle d'activité irrégulière du cœur.
- **Management et finance** : Prévisions économiques, analyse financière, et prévision du marché.

1.3.4 Caractéristiques du chaos

Caractériser le chaos est un sujet vaste, et les tests les plus utiles pour ce type de comportement sont les suivants [7] :

- **La non-linéarité :**

Un système chaotique est décrit par un ensemble d'équations dynamiques non linéaires et déterministes. Bien que ces équations définissent complètement son évolution, il est imprédictible à long terme.

- **Le déterministe :**

La notion de déterminisme signifie la capacité de « prédire » le futur d'un phénomène à partir d'un événement passé ou présent.

L'évolution irrégulière du comportement d'un système chaotique est due aux non linéarités.

- **L'attracteur étrange :**

C'est la forme géométrique complexe qui caractérise l'évolution des systèmes dynamiques chaotiques.

- **Sensibilité aux conditions initiales :**

La sensibilité aux conditions initiales est l'une des propriétés essentielles du chaos, la moindre perturbation du système peut conduire à des états extrêmement différents. Un événement insignifiant n'a donc pas toujours des conséquences insignifiantes.

Cette propriété a été observée pour la première fois par E. Lorenz sur son modèle météorologique. Elle est connue sous le nom populaire d'effet papillon [1].

La figure 1.1 illustre cette sensibilité aux conditions initiales d'un système chaotique.

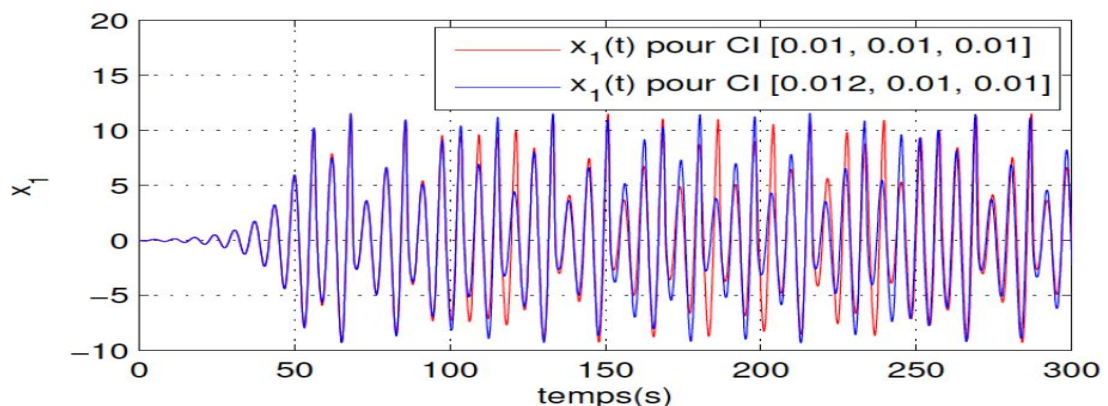


Figure 1.1 : Sensibilité aux conditions initiales d'un système chaotique.

- **L'aspect aléatoire :**

Tous les états d'un système chaotique présentent des aspects aléatoires. La figure 1.2 représente une comparaison entre un système chaotique et une sinusoïde.

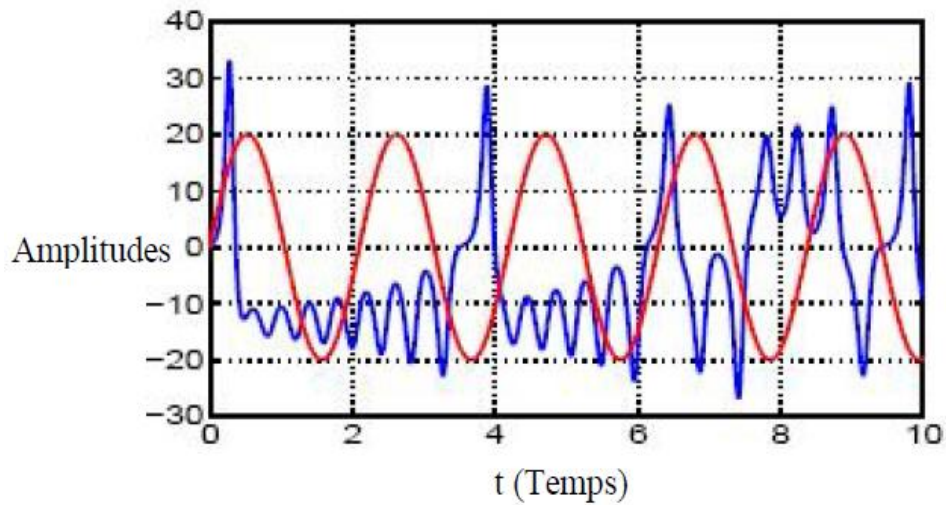


Figure 1.2 : Evolution dans le temps d'un système chaotique, comparé à une sinusoïde.

1.4 Espace des phases

Un système dynamique est caractérisé par un certain nombre de variables d'état, qui ont la propriété de définir complètement l'état du système à un instant donné. Dans un système dynamique de dimension n , l'espace x_1, x_2, \dots, x_n est appelé espace de phases ou espaces d'états [5] [7]. L'évolution par rapport au temps du système se traduit alors par un tracé des trajectoires graphiques ou chaque point définit un état dans l'espace, la lecture de ces trajectoires permet de distinguer un comportement chaotique d'un comportement purement aléatoire à l'aide des conditions initiales.

La figure 1.3 donne une représentation sur un espace des phases à 3 dimensions.

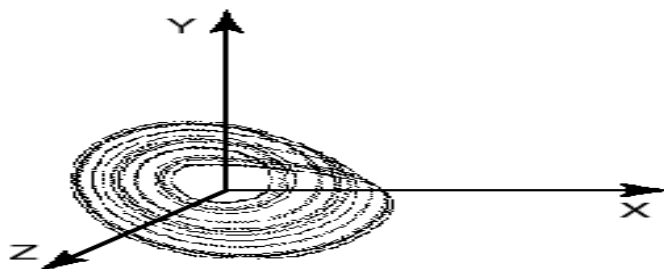


Figure 1.3 : Exemple d'un espace des phases.

1.5 Point fixe :

Appelé aussi point d'équilibre, de repos, point critique, singulier ou bien stationnaire, c'est le point \bar{x} de l'espace des phases obtenu en annulant le second membre de la fonction dynamique f du système (1.1) :

$$f(\bar{x})=0 \quad (1.6)$$

Par le changement de variables $\xi=x-\bar{x}$, on peut ramener le point \bar{x} à l'origine.

On s'intéresse souvent à l'évolution locale de la trajectoire au point fixe. Il s'agit de prévoir l'évolution (asymptotique) de la trajectoire autour du point fixe [7].

1.5.1 Stabilité des points fixes

On désigne par stable tout système qui reprend son état d'équilibre après une perturbation ; la stabilité d'un point stationnaire garantit une stabilité locale (condition nécessaire mais pas suffisante). Le système dynamique non linéaire est stable si tous ses états stationnaires sont stables. Pour déterminer la stabilité d'un système dynamique non linéaire, on étudie la stabilité du point fixe.

- **Stabilité au sens du Lyapunov :**

Le point d'équilibre \bar{x} est dit stable au sens de Lyapunov si $\forall \varepsilon > 0, \exists \alpha > 0$, tel que :

$$\|x(0) - \bar{x}\| < \alpha \rightarrow \|x(t) - \bar{x}\| < \varepsilon, \forall t \geq 0 \quad (1.7)$$

Où $\|\cdot\|$ désigne la norme dans R^n .

Autrement dit, le point d'équilibre est stable si toutes les solutions issues des points proches du point d'équilibre restent proches de celui-ci.

Le point \bar{x} est instable s'il n'est pas stable au sens de Lyapunov.

Si de plus, il existe δ avec $\delta < 0$ tel que :

$$\|x(0) - \bar{x}\| < \delta \rightarrow \lim_{t \rightarrow \infty} x(t) = \bar{x} \quad (1.8)$$

On dit alors que \bar{x} est asymptotiquement stable.

La stabilité asymptotique signifie qu'on peut déterminer un voisinage du point d'équilibre tel que n'importe quelle trajectoire, issue d'un point $x(0)$ appartenant à un voisinage de \bar{x} tende vers \bar{x} lorsque $t \rightarrow \infty$ [5].

- **Stabilité du système linéarisé (valeurs propres) :**

La matrice $Df(x)$ est la matrice jacobéenne de $f(x)$ au point d'équilibre $x = 0$:

$$Df(x) \equiv \frac{\partial f_i(x)}{\partial x_j} \quad (1.9)$$

Cette méthode consiste à étudier le système linéaire (1.10), pour étudier la stabilité autour d'un point.

$$\dot{x} = Df(0) * x \quad (1.10)$$

Avec :

$$Df(0) = \begin{pmatrix} \frac{df_1}{dx_1} & \dots & \frac{df_1}{dx_n} \\ \vdots & \ddots & \vdots \\ \frac{df_n}{dx_1} & \dots & \frac{df_n}{dx_n} \end{pmatrix}_{x=0} \quad (1.11)$$

Dans le cas où la matrice $Df(0)$ possède n valeurs propres $\lambda_i, i = 1, \dots, n$ distinctes, la solution est :

$$x = \sum_{i=1}^n c_i a^{(i)} \exp(\lambda_i t) \quad (1.12)$$

Où $a^{(i)}$ est le vecteur propre correspondant à la valeur propre λ_i et les $c_i, i = 1, 2, \dots, n$ sont des constantes (déterminées par les conditions initiales). On en déduit que :

- Si toutes les valeurs propres λ_i ont leur partie réelle négative, le point fixe est asymptotiquement stable.
- Si une ou plusieurs valeurs propres sont des imaginaires pures, les autres valeurs propres ayant leur partie réelle négative, le point fixe est un centre ou un point elliptique (stable mais pas asymptotiquement stable).
- Si une des valeurs propres à sa partite réelle positive le point fixe est instable.

- d) Si $Df(0)$ n'a pas de valeur nulle ou purement imaginaire, le point fixe est un point hyperbolique. Dans le cas contraire, il est non-hyperbolique.
- e) S'il existe i et j tels que $\Re\lambda_i < 0$ et $\Re\lambda_j > 0$, le point fixe est un point de selle.
- f) Si toutes les valeurs propres de $Df(0)$ sont réelles et de même signe, le point fixe est un nœud [5].

1.6 Exposants de Lyapunov

Le 12 octobre 1892, Lyapunov soutient à l'Université de Moscou une thèse de doctorat intitulée : Le problème général de la stabilité du mouvement. Il y introduit l'idée de mesurer la divergence possible entre deux orbites issues de conditions initiales voisines [8].

L'exposant de Lyapunov sert à mesurer le degré de stabilité d'un système. Un système sensible à de très petites variations de la condition initiale aura un exposant positif (système chaotique). En revanche, l'exposant est négatif si le système n'est pas sensible à des petites variations des conditions initiales, les trajectoires se rapprochent et on perd donc l'information sur les conditions initiales.

Un système de dimension n possède n exposants de Lyapunov qui mesurent le taux de divergence suivant un des axes de l'espace de phase.

Un exposant de Lyapunov positif (respectivement négatif) selon une direction indique, qu'une divergence entre deux trajectoires voisines augmente (respectivement diminue) exponentiellement avec le temps [9].

La figure 1.4 représente la dynamique des exposants de Lyapunov d'un système en fonction du temps.

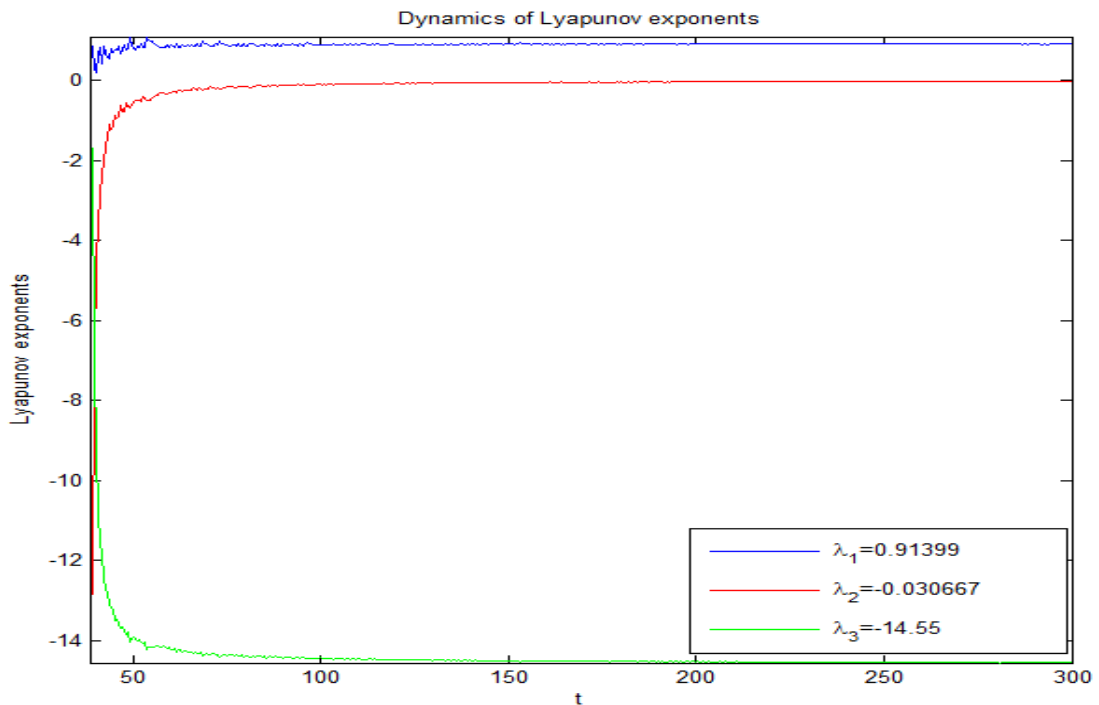


Figure 1.4 : Exemple des exposants de Lyapunov d'un système chaotique.

1.6.1 Comportement du système en fonction des exposants de Lyapunov

En étudiant les exposants de Lyapunov d'un système non linéaire, on peut définir le type d'attracteur (comportement asymptotique) généré par le système (sous l'hypothèse que les trajectoires évoluent dans une région bornée) :

- $\lambda_n \leq \dots \leq \lambda_1 < 0$: des exposants de Lyapunov négatifs montrent l'existence d'un point fixe.
- $\lambda_1 = 0, \lambda_n \leq \dots \leq \lambda_2 < 0$: l'attracteur est une orbite fermée.
- $\lambda_1 = \lambda_2 = 0, \lambda_n \leq \dots \leq \lambda_3 < 0$: l'attracteur est quasi-périodique (2 fréquences).
- $\lambda_1 = \dots = \lambda_k = 0, \lambda_n \leq \dots \leq \lambda_{k+1} < 0$: l'attracteur est quasi-périodique (k fréquences).
- $\lambda_1 > 0, \sum_i \lambda_i < 0$: l'attracteur est chaotique.
- $\lambda_1 > \dots > \lambda_k > 0, \sum_i \lambda_i < 0$: l'attracteur est hyper-chaotique [5].

1.7 Section de Poincaré

L'application de Poincaré est un outil mathématique simple permettant de transformer un système dynamique continu en un système dynamique discret.

Réaliser une section de Poincaré revient à couper la trajectoire dans l'espace des phases afin d'étudier les intersections de cette trajectoire avec un plan. Cette transformation se fait par une réduction d'une unité de l'ordre du système. Les mathématiciens ont démontré que les propriétés du système sont conservées après la réalisation d'une section de Poincaré intelligemment choisie [10].

L'ensemble des points d'intersection situé sur la surface, représente la section de Poincaré. Un exemple de section de Poincaré est représenté sur la figure 1.5.

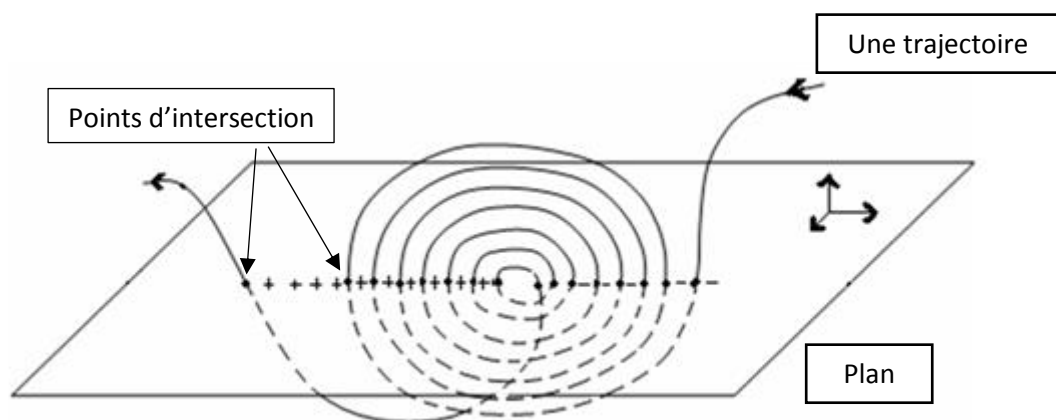


Figure 1.5 : Principe de section de Poincaré.

Le graphique de la section de Poincaré peut être :

- **Un point unique** : le système est périodique.
- **Un ensemble discret de points** : le système est également périodique.
- **Une courbe fermée** ; le système est quasi-périodique.
- **Un nuage de points** : le système est chaotique.

1.8 Bifurcation

La théorie de bifurcation est l'étude mathématique des changements qualitatifs ou topologiques de la structure d'un système dynamique. Une bifurcation survient lorsqu'une variation quantitative d'un paramètre du système engendre un changement qualitatif des propriétés d'un système tel que la stabilité, le nombre de points d'équilibre ou la nature des régimes permanents. Les valeurs des paramètres au moment du changement sont appelées valeurs de bifurcation. Dans les systèmes dynamiques, un diagramme de bifurcation montre les comportements possibles d'un système, à long terme, en fonction des paramètres de bifurcation [3].

La bifurcation veut dire division d'une branche principale en au moins deux branches.

1.8.1 Diagramme de bifurcation

Le diagramme de bifurcation est un outil mathématique qui permet de repérer les valeurs du paramètre qui induisent des bifurcations. C'est un diagramme qui porte les valeurs du paramètre en abscisse et des valeurs particulières d'une variable d'état. En fixant les valeurs des autres paramètres du système, on obtient le diagramme de bifurcation par rapport au paramètre considéré accompagné d'un changement de stabilité.

Il existe plusieurs types de bifurcations, parmi lesquelles on peut citer :

- 1- Bifurcation Nœud-col (ou pli).
- 2- Bifurcation Trans-critique.
- 3- Bifurcation fourche (sous-critique ou sur-critique).
- 4- Bifurcation de Hopf (sous-critique ou sur-critique).

Les trois premiers types de bifurcations correspondent à des bifurcations statiques où le point de bifurcation sépare des branches de points fixes. Les bifurcations de Hopf sont des bifurcations dynamiques où le point critique délimite dans l'espace de contrôle d'état des branches de points fixes et un cycle limite. La figure 1.6 illustre le tracé de bifurcation pour un système de chaotique [10].

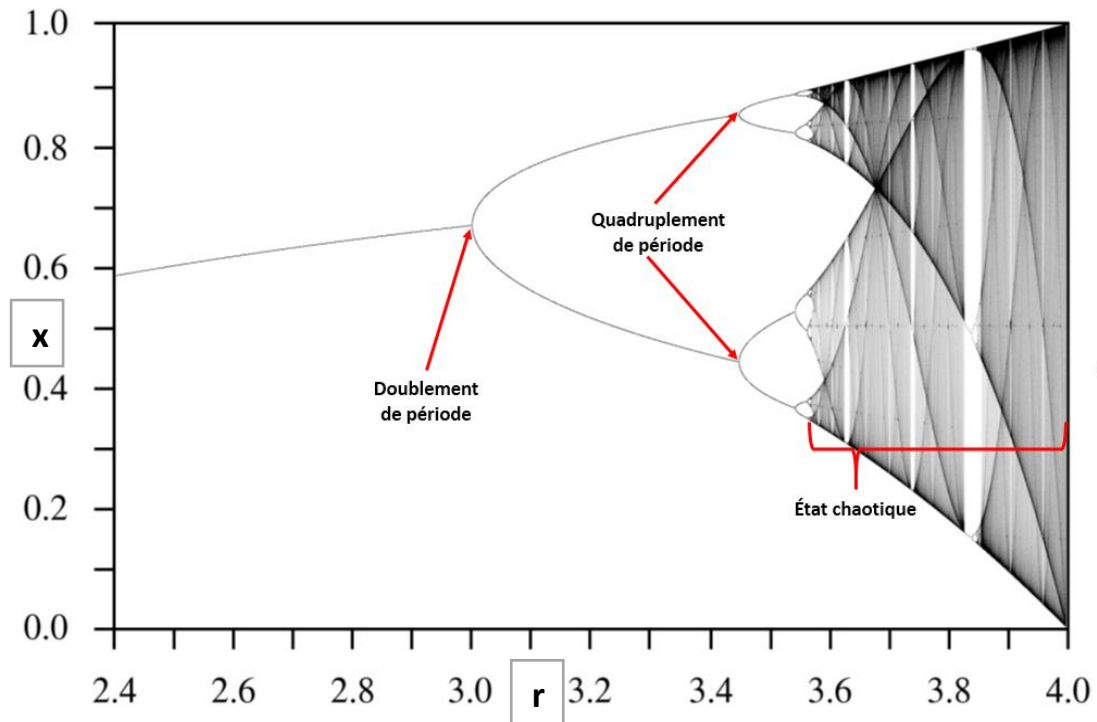


Figure 1.6 : Exemple de diagramme de bifurcation.

- Pour $2.4 < r < 3$: le système possède un point fixe stable (un cycle d'ordre 1).
- Pour $3 < r < 3.45$: le point fixe se déstabilise et un cycle d'ordre 2 apparaît (doublement de période).
- Pour $3.45 < r < 3.55$: le cycle d'ordre 2 se déstabilise et un cycle d'ordre 4 apparaît et ainsi de suite.
- Pour $r > 3.6$: l'apparition du chaos.

1.9 Exemples de systèmes chaotiques

Dans cette section, nous présentons quelques exemples de systèmes chaotiques les plus célèbres.

1.9.1 Système de Lorenz

Le système de Lorenz fut introduit par Edward Lorenz en 1963 [1]. Il s'agit d'un système dynamique non linéaire de dimension 3, obtenu à partir des équations de transfert de la chaleur dans un liquide. Le système de Lorenz est défini par :

$$\begin{cases} \frac{dx}{dt} = a(y - x) \\ \frac{dy}{dt} = bx - y - xz \\ \frac{dz}{dt} = xy - cz \end{cases} \quad (1.13)$$

Le système de Lorenz montre un comportement chaotique et génère un attracteur étrange pour $a = 10, b = 28, c = 8/3$.

Les figures qui suivent, représentent les propriétés du système chaotique de Lorenz.

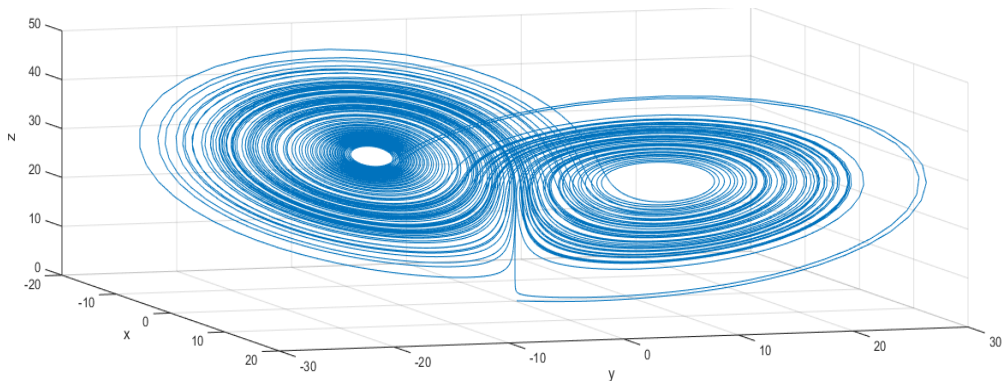


Figure 1.7 : Attracteur étrange de Lorenz.

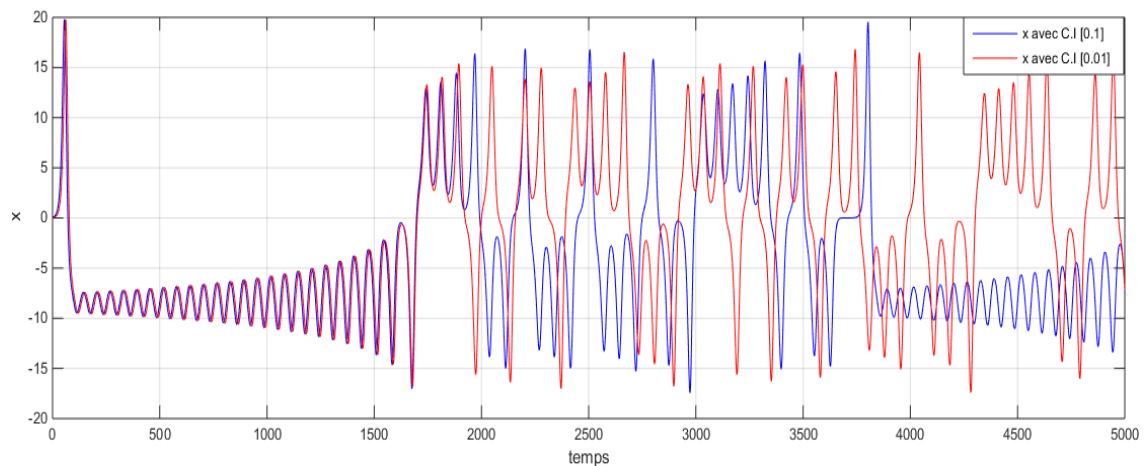


Figure 1.8 : Sensibilité aux conditions initiales du système de Lorenz.

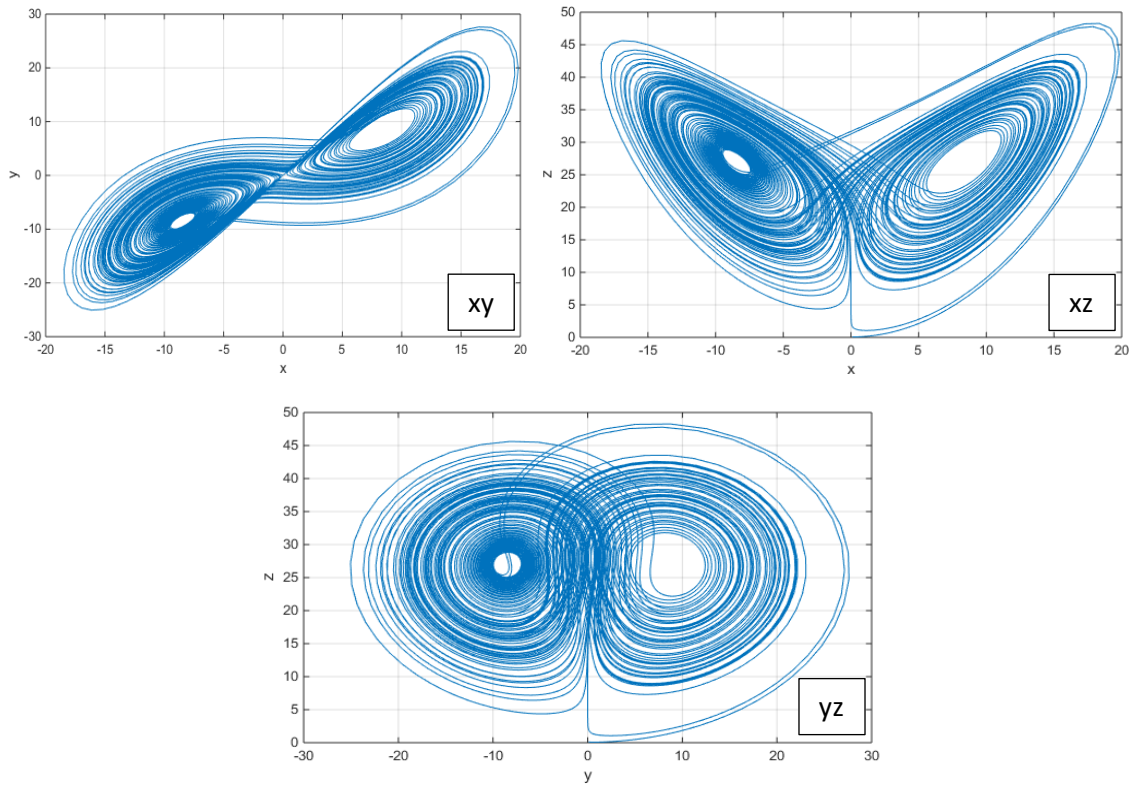


Figure 1.9 : Les plans de phases du système de Lorenz.

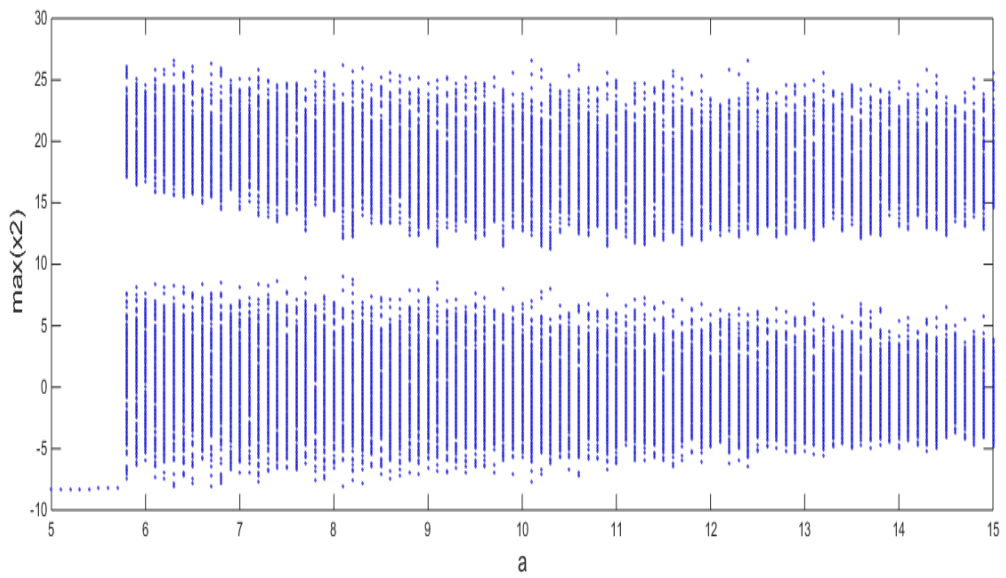


Figure 1.10 : Diagramme de bifurcation du système de Lorenz.

1.9.2 Système de Rössler

Proposé par l'Allemand Otto Rössler en 1974, le système de Rössler est lié à l'étude de l'écoulement des fluides. Il résulte d'un effort de simplification pour étudier plus facilement la "chute" d'une trajectoire dans un bassin d'attraction [11].

Les équations de ce système sont les suivantes :

$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + xz - cz \end{cases} \quad (1.14)$$

On prendra : $a = 0.1$, $b = 14$ et $c = 0.1$.

Les figures suivantes représentent les principales propriétés du système chaotique de Rössler.

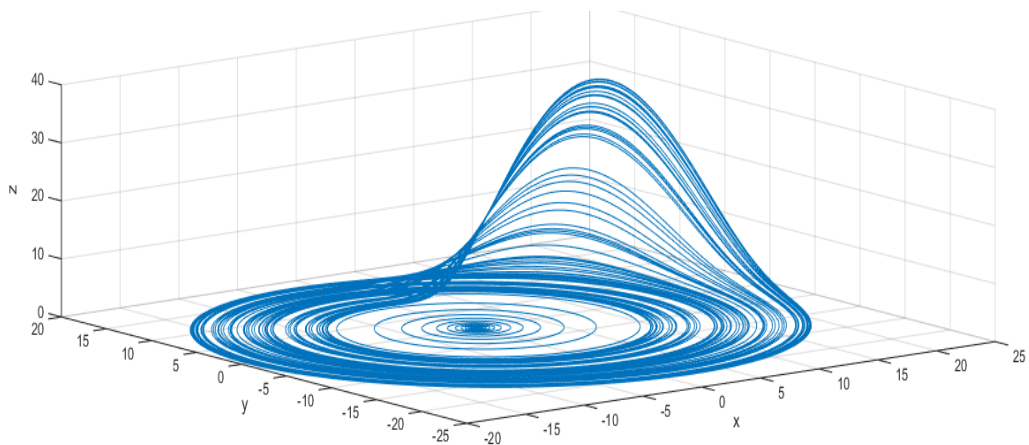


Figure 1.11 : Attracteur étrange de Rössler.

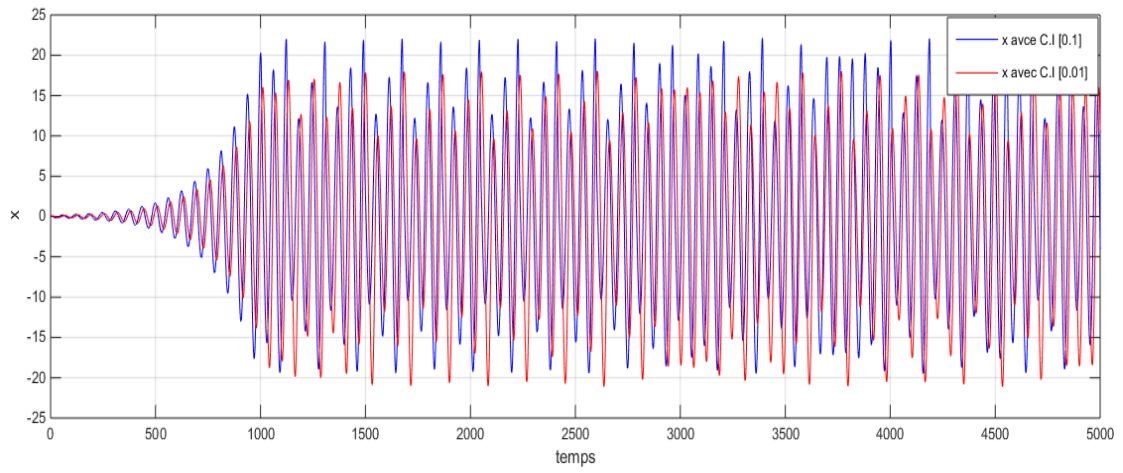


Figure 1.12 : Sensibilité aux conditions initiales du système de Rössler.

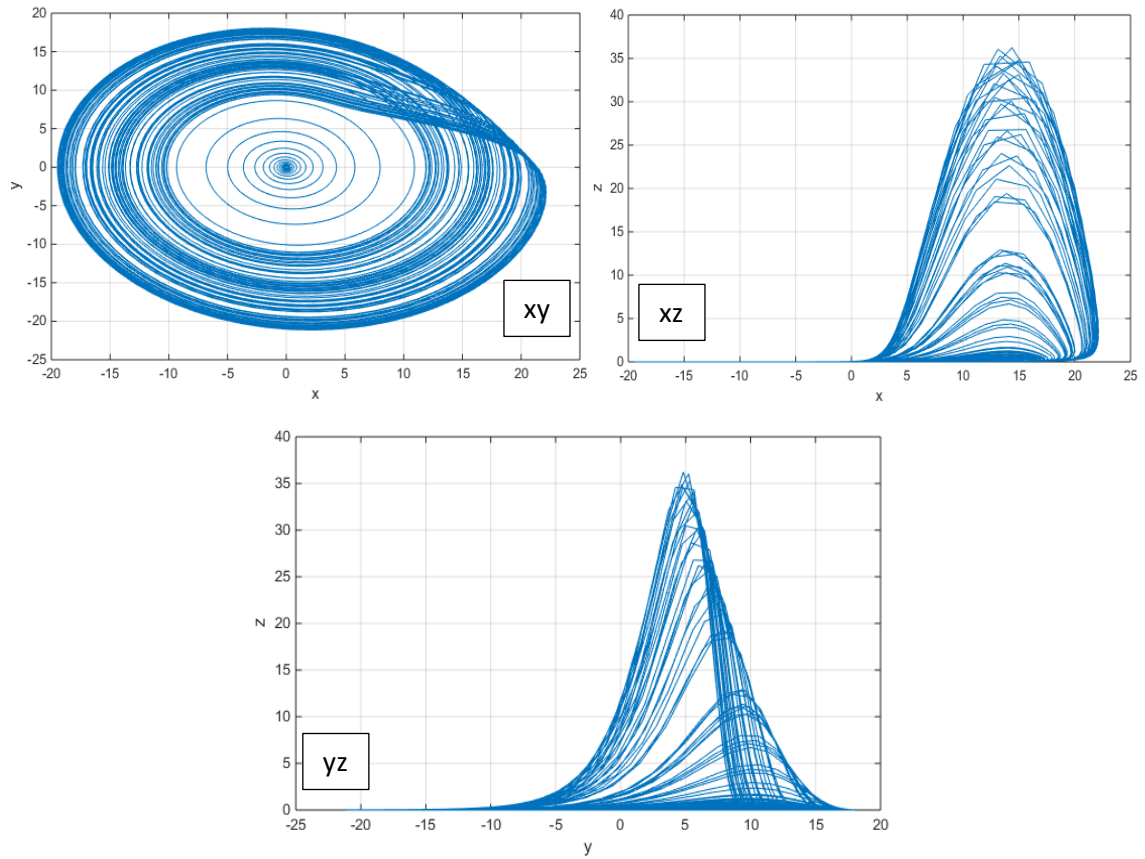


Figure 1.13 : Les plans de phase du système de Rössler.

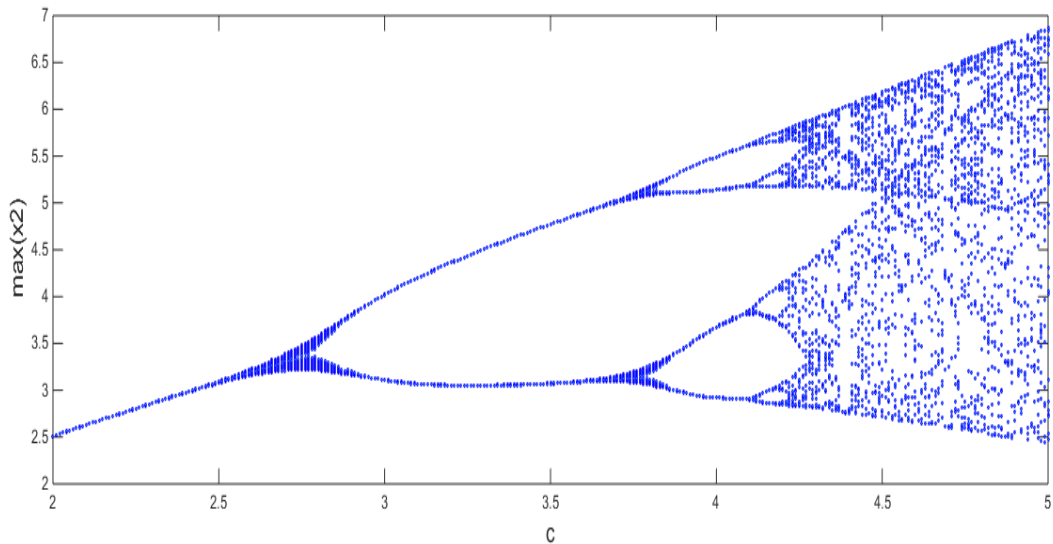


Figure 1.14 : Diagramme de bifurcation du système de Rössler.

1.10 Conclusion

Dans ce chapitre, nous avons présenté les principales caractéristiques des systèmes dynamiques chaotiques.

Nous avons mis en relief certaines propriétés à savoir : les points d'équilibre, le diagramme de Bifurcation, le plan de phase, la section de Poincaré, les exposants de Lyapunov et l'attracteur étrange.

Ces notions nous seront très utiles pour l'analyse du système chaotique auto-commuté dont le développement sera détaillé dans le prochain chapitre.

Chapitre 2 Etude du système chaotique

Auto-commuté

2.1 Introduction

En 1963, Lorenz a découvert un système chaotique autonome tridimensionnel, qui a généré le bien connu attracteur chaotique Lorenz. Ce système a été largement étudié dans le domaine de la théorie chaotique et des systèmes dynamiques. Cependant, le système chaotique commuté a des comportements dynamiques plus complexes que le système chaotique ordinaire. Il a été étudié avec un grand intérêt en raison de ses applications théoriques et pratiques dans les domaines technologiques, telles que la communication sécurisée, les lasers, les circuits non linéaires, les réseaux neurones, le contrôle et la synchronisation. Il existe deux méthodes principales pour générer le système commuté. L'une est par manœuvre manuelle, l'autre est la commutation automatique. Ce dernier est présente de meilleurs avantages.

Ce chapitre propose l'étude des différentes caractéristiques d'un système chaotique à commutation automatique (système Auto-commuté) et son analyse détaillée à l'aide des outils de simulation MATLAB- SIMULINK et MATDS.

2.2 Description du système

Le système Auto-commuté est un système chaotique qui se compose de deux sous systèmes et peut changer ses comportements automatiquement grâce à sa constitution. Considérons le système tridimensionnel non-linéaire suivant [12] :

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = bx + cy - xz \\ \dot{z} = f^2(x) - hz \end{cases} \quad (2.1)$$

Où $f(x) = \begin{cases} x, & x \geq 0 \\ y, & x < 0 \end{cases}$. On obtient alors un système chaotique à commutation automatique.

Pour la simulation numérique, les paramètres du système sont comme suit :

$$a = 20, b = 14, c = 10.6 \text{ et } h = 2.8$$

Lorsque la variable d'état x du système Auto-commuté (2.1) satisfait la condition $x \geq 0$, la fonction $f(x)$ est égale x , donc notre système fonctionne sur le sous-système (2.2).

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = bx + cy - xz \\ \dot{z} = x^2 - hz \end{cases} \quad (2.2)$$

Lorsque $x < 0$, le système s'exécute sur un autre sous-système (2.3).

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = bx + cy - xz \\ \dot{z} = y^2 - hz \end{cases} \quad (2.3)$$

Lorsque $a = 20, b = 14, c = 10.6$ et $h = 2.8$, les sous-systèmes (2.2) et (2.3) ont tous les deux des attracteurs chaotiques et leurs plans de phases sont représentés sur les figures (2.1) et (2.2).

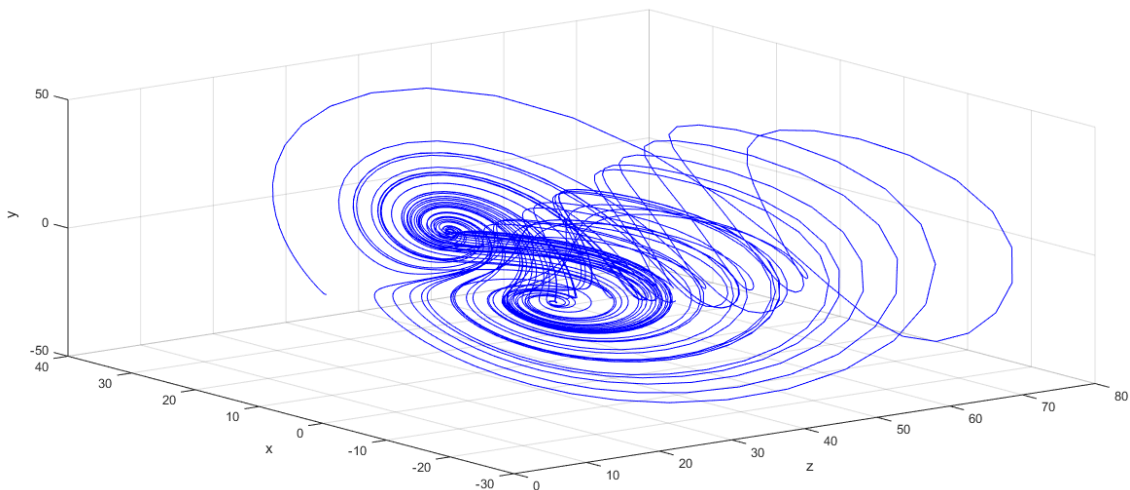


Figure 2.1. Attracteur chaotique du sous-système (2.2).

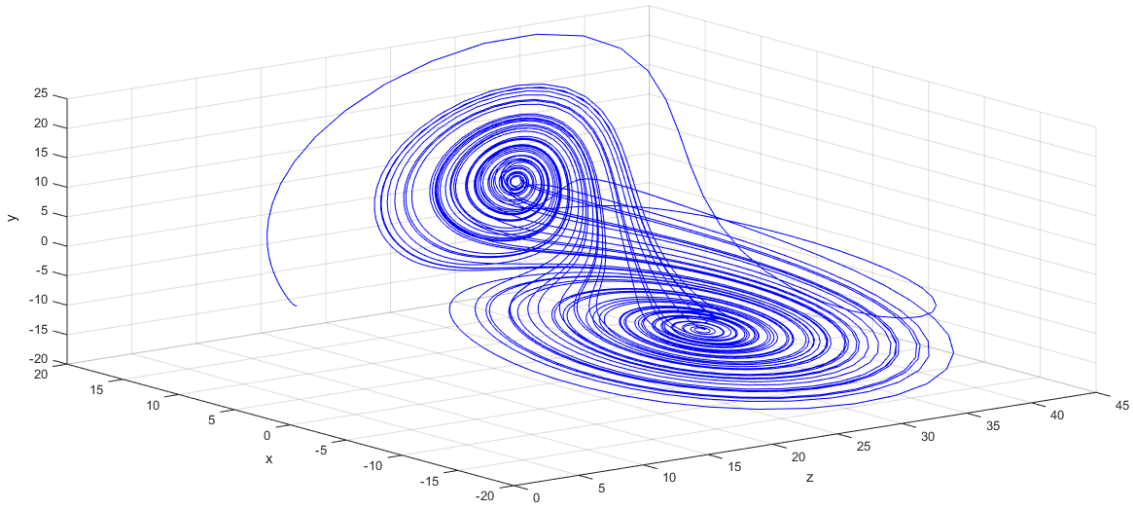


Figure 2.2. Attracteur chaotique du sous-système (2.3).

2.3 Analyse du système Auto-commuté

2.3.1 Les propriétés de base du système Auto-commuté

- **La symétrie** : Les sous-systèmes (2.2) et (2.3) sont symétriques par rapport à l'axe z pour son invariance sous la transformée de coordonnées $(x, y, z) \rightarrow (-x, -y, z)$, mais le système chaotique Auto-commuté (2.1) n'est pas symétrique autour de l'axe z . La propriété de symétrie du système chaotique Auto-commuté est modifiée par la fonction $f(x)$.
- **La dissipation** : Pour les systèmes (2.1), (2.2) et (2.3) nous avons :

$$\Delta V = \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z} = -a + c - h \quad (2.4)$$

Tous les trois systèmes sont dissipatifs si et seulement si $-a + c - h > 0$. Selon cette condition ; les trois systèmes convergent exponentiellement.

$$\frac{dv}{dt} = e^{-(a-c+h)t} \quad (2.5)$$

C'est-à-dire que l'élément de volume à l'instant t est $V_0 e^{-i\omega t}$, avec l'élément de volume initial V_0 , toutes les orbites de ces systèmes sont limitées à un sous-ensemble spécifique de volume nul quand $t \rightarrow \infty$.

Alors, l'existence d'attracteur est prouvée.

2.3.2 Equilibre et stabilité

Pour déterminer les points fixes du système, on doit résoudre le système d'équation algébrique suivant :

$$\frac{dx}{dt} = f(x) = 0 \quad (2.6)$$

C'est-à-dire :

$$\begin{cases} \dot{x} = a(y - x) & = 0 \\ \dot{y} = bx + cy - xz & = 0 \\ \dot{z} = f^2(x) - hz & = 0 \end{cases} \quad (2.7)$$

On obtient une solution : $x = y = z = 0 \rightarrow D_0(0,0,0)$, ainsi que deux autres :

$$D_1(-\sqrt{h(b+c)}, -\sqrt{h(b+c)}, b+c) \quad \text{et} \quad D_2(\sqrt{h(b+c)}, \sqrt{h(b+c)}, b+c).$$

Une fois les points fixes déterminés, on calcule la matrice Jacobienne du système pour ces points fixes.

Pour le système chaotique Auto-commuté, la matrice Jacobienne au point d'équilibre $D_0(0,0,0)$ est donnée par :

$$J(0) = \begin{bmatrix} -a & a & 0 \\ b & c & 0 \\ 0 & 0 & -h \end{bmatrix} \rightarrow J(0) = \begin{bmatrix} -20 & 20 & 0 \\ 14 & 10.6 & 0 \\ 0 & 0 & -2.8 \end{bmatrix} \quad (2.8)$$

$$a = 20, b = 14, c = 10.6, h = 2.8$$

Les valeurs propres sont données par les solutions de l'équation (2.9) :

$$\lambda I - J(0) = 0 \quad (2.9)$$

D'où λI est donnée par :

$$\lambda I = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \quad (2.10)$$

Avec : λ valeur propre de la matrice, et I la matrice identité.

Son équation caractéristique est donnée par : $d = \det(\lambda I - J(0)) = 0$, soit :

$$\lambda I - J(0) = \begin{bmatrix} \lambda + a & 0 & 0 \\ b & \lambda - c & 0 \\ 0 & 0 & \lambda + h \end{bmatrix} \quad (2.11)$$

Le polynôme caractéristique de la matrice :

$$d = (\lambda + a)[(\lambda - c)(\lambda + h)] - a[b(\lambda + h)] = 0 \quad (2.12)$$

Les valeurs propres correspondantes sont trois :

$$\lambda_1 = -27.4, \quad \lambda_2 = 18, \quad \lambda_3 = -2.8$$

- Nous avons obtenus trois valeurs purement réelles d'où l'une est positive λ_2 , et les deux autres sont négatives λ_1 et λ_3 , par conséquent notre point fixe $D_0(0,0,0)$ est instable.

Aux points D_1 et D_2 , nous obtenons respectivement :

$$D_1: \quad \lambda_1 = -18.7, \quad \lambda_2 = 3.24 + j11.7, \quad \lambda_3 = 3.24 - j11.7$$

$$D_2: \quad \lambda_1 = -14.3, \quad \lambda_2 = 1.03 + j13.9, \quad \lambda_3 = 1.03 - j13.9$$

- D'après les valeurs propres obtenues des deux points D_1 et D_2 , la valeur propre λ_1 est un nombre réel négatif, λ_2 et λ_3 sont des nombres complexes conjugués avec des parties réelles positives ; ce qui signifie que les deux points d'équilibre sont instables.

Les résultats obtenus montrent que les trois points d'équilibre du système sont tous des nœuds de focalisation de selle.

2.3.3 Évolution du système Auto-commuté en fonction du temps

En utilisant MATLAB-Simulink, nous avons réalisé un circuit qui décrit notre système chaotique Auto-commuté afin de visualiser les différents signaux issus du système, tels que : les états x, y, z en fonction du temps, les plans de phase, et le tracé de l'attracteur chaotique en 3D, à partir des équations du système (2.1).

La figure 3.2 représente le système chaotique Auto-commuté sous MATLAB-Simulink.

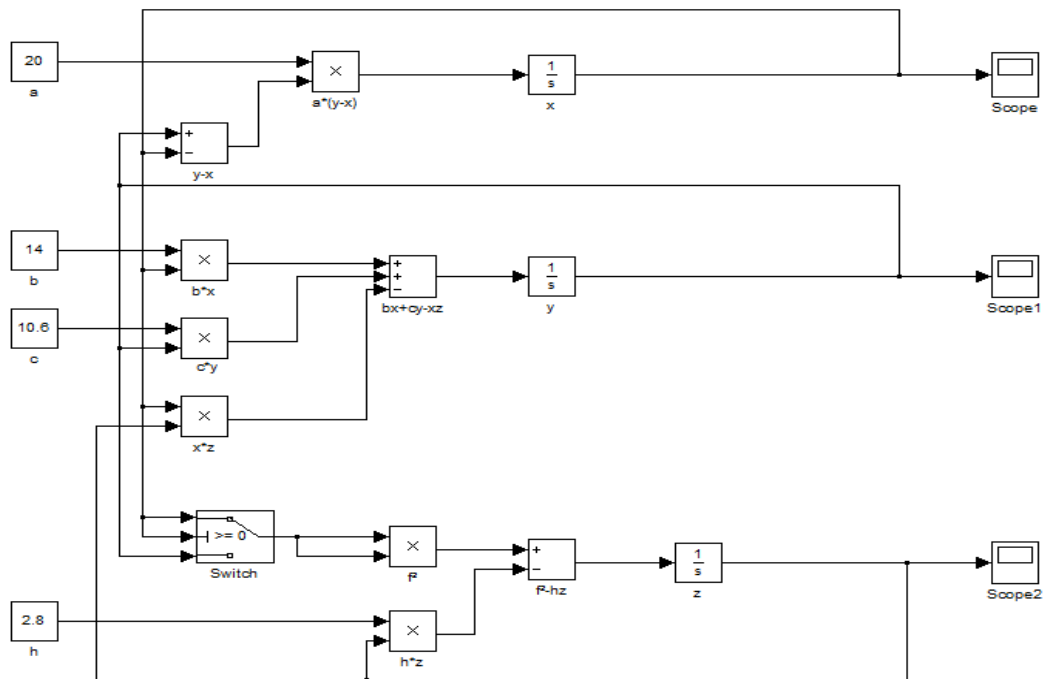


Figure 2.3. Représentation du système chaotique Auto-commuté sous MATLAB-Simulink.

Les figures suivantes représentent l'évolution des états x , y et z en fonction du temps.

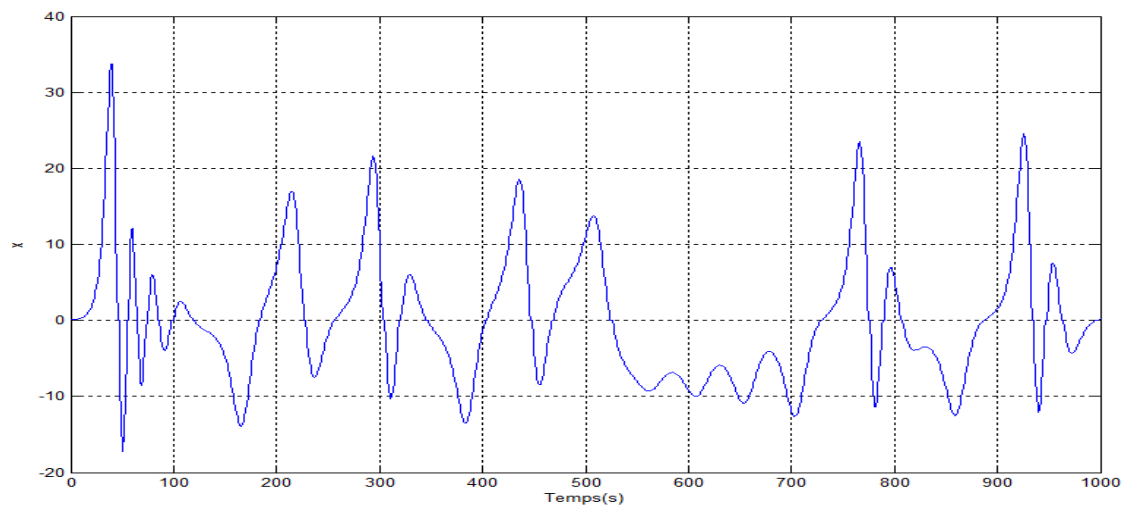


Figure 2.4. L'état x en fonction du temps.

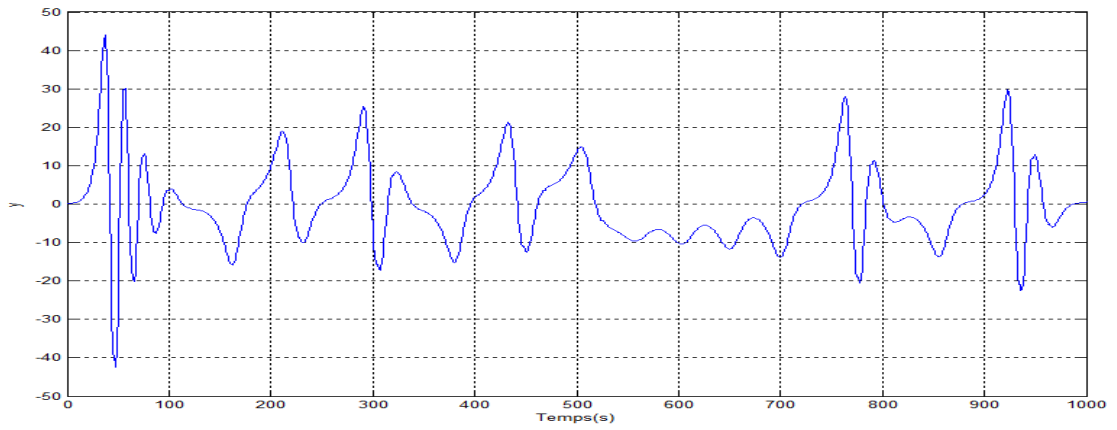


Figure 2.5. L'état y en fonction du temps.

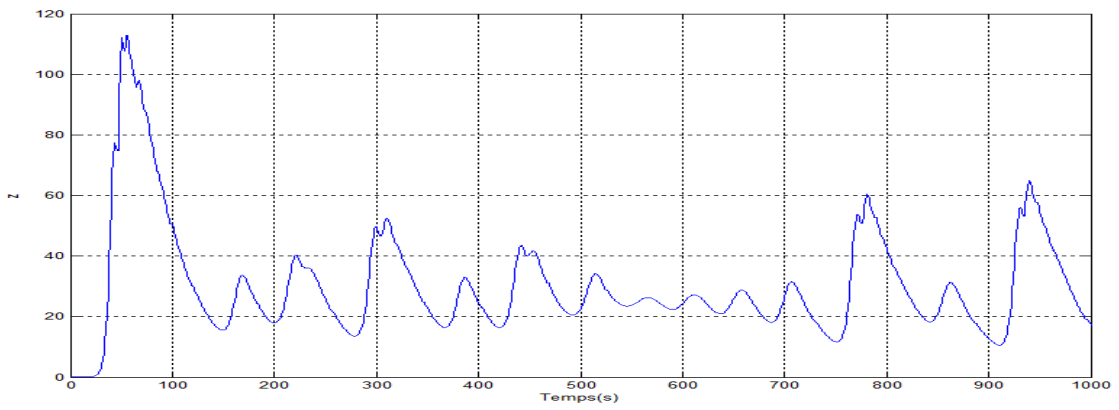


Figure 2.6. L'état z en fonction du temps.

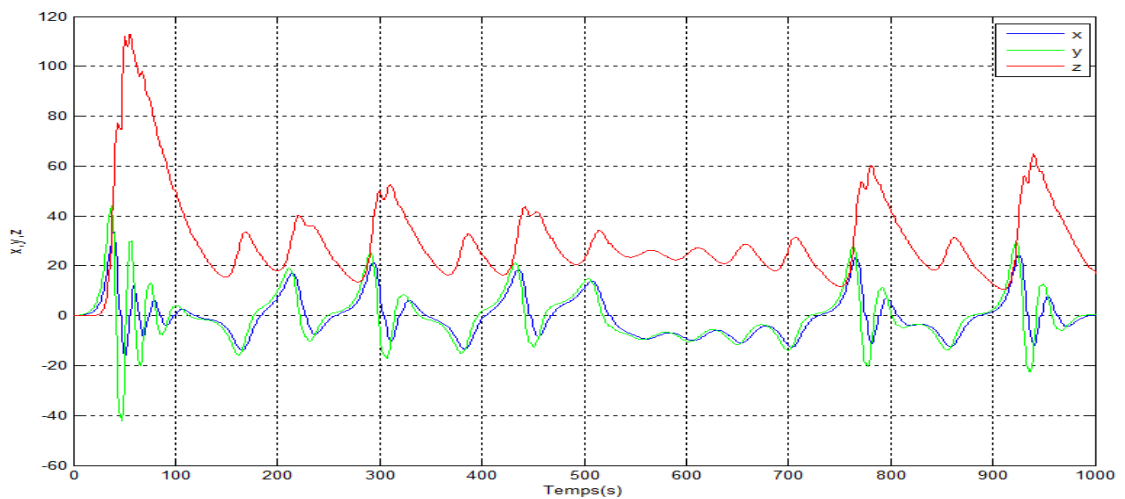


Figure 2.7. Les états x , y et z en fonction du temps.

2.3.4 Sensibilité aux conditions initiales

Afin d'illustrer la propriété de sensibilités aux conditions initiales, on va faire une simulation avec deux différentes conditions initiales très proches.

La figure 2.8 représente l'état x avec les deux conditions initiales [0.1] et [0.01].

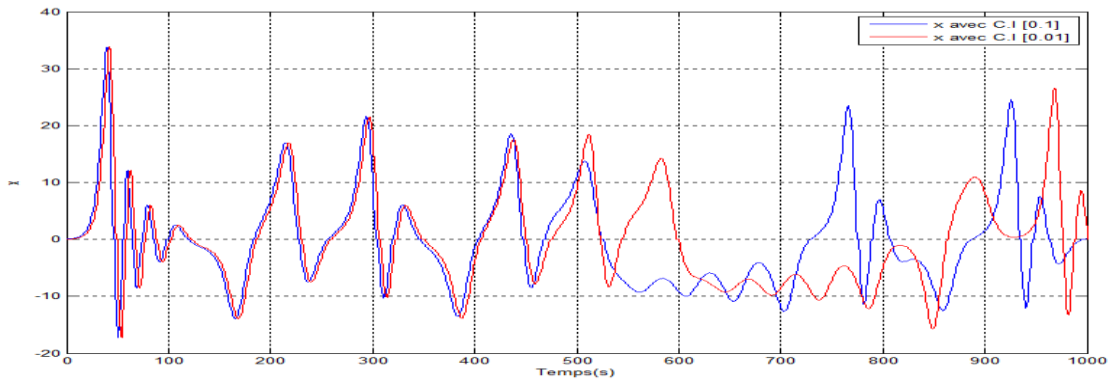


Figure 2.8. La propriété de sensibilité aux conditions initiales sur l'état x .

On remarque ainsi l'extrême sensibilité aux conditions initiales du système.

2.3.5 Plan de phase

Le système chaotique Auto-commuté (2.1) peut changer son comportement aléatoirement entre les deux sous-systèmes (2.2) et (2.3) lorsque x est varié. Les plans de phases du système sont représentés sur les figures 2.9 à 2.11 sans lesquels les parties gauches expriment les orbites de sous-système (2.2), et les parties droites expriment les orbites de sous-système (2.3).

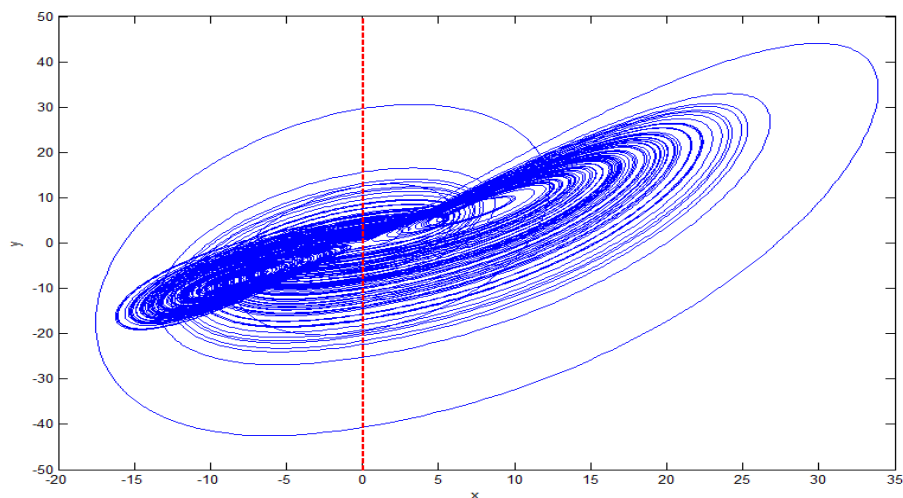


Figure 2.9. Plan de phase x en fonction de y .

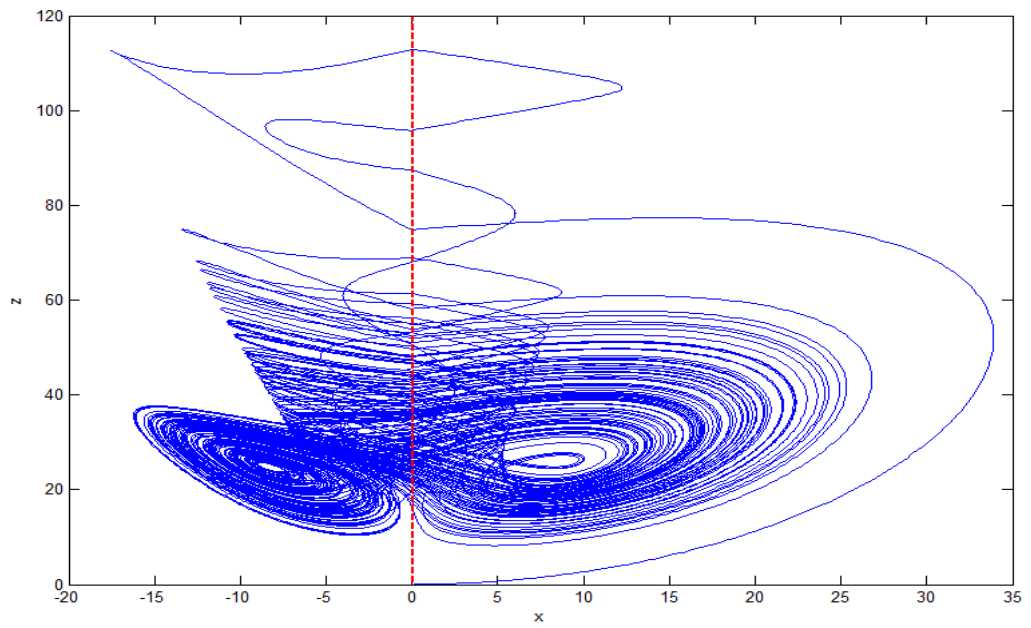


Figure 2.10. Plan de phase x en fonction de z .

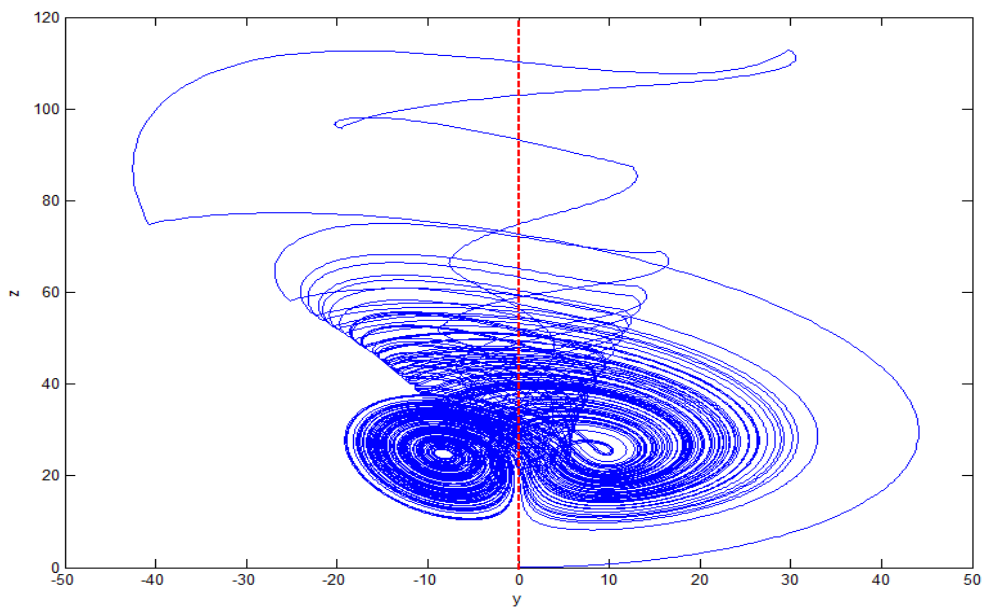


Figure 2.11. Plan de phase y en fonction de z .

2.3.6 Attracteur étrange

La figure 2.12 représente l'attracteur chaotique du système Auto-commuté.

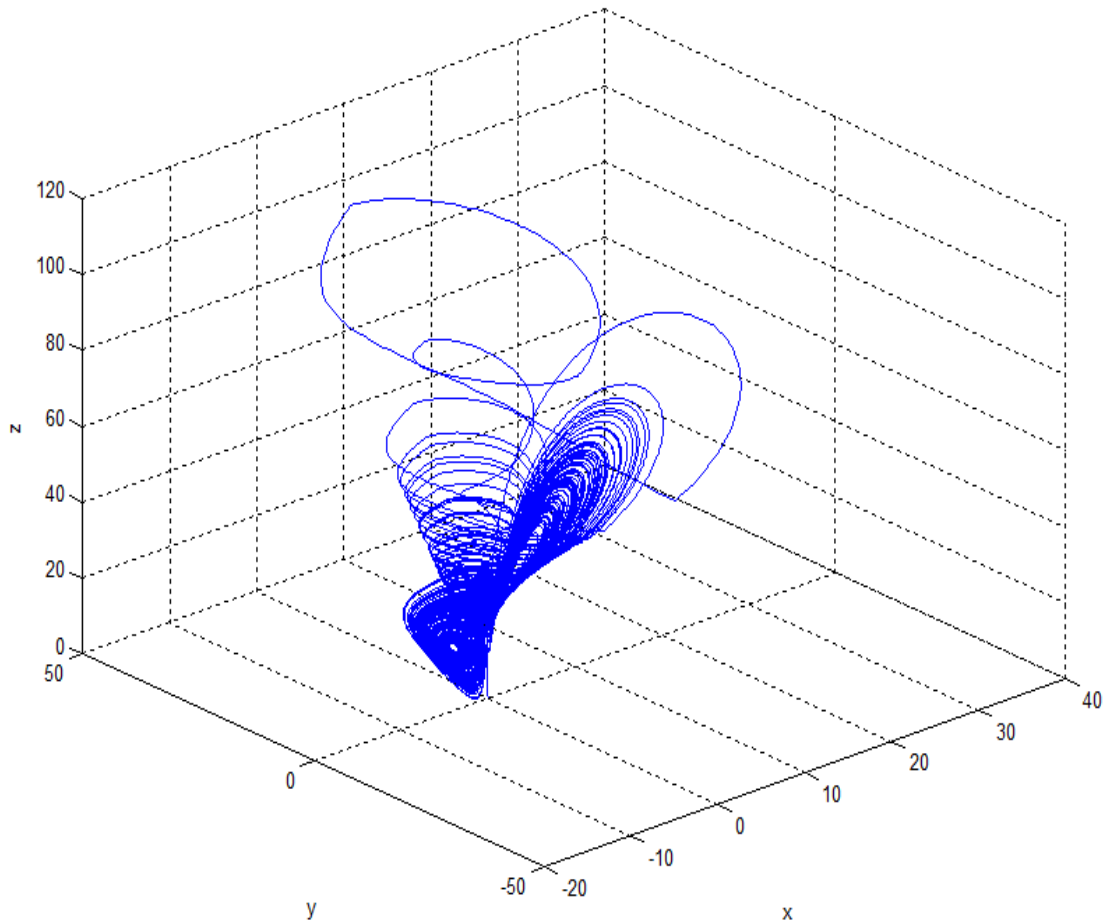


Figure 2.12. Attracteur étrange du système Auto-commuté en fonction de x , y et z .

2.3.7 Exposants de Lyapunov

Pour étudier les systèmes dynamiques, nous avons utilisé l'outil MATDS qui fonctionne sous MATLAB. Tout d'abord on fait la présentation de l'outil MATDS :

- En lançant le MATDS, une fenêtre apparaît et est représentée dans la figure 2.13.

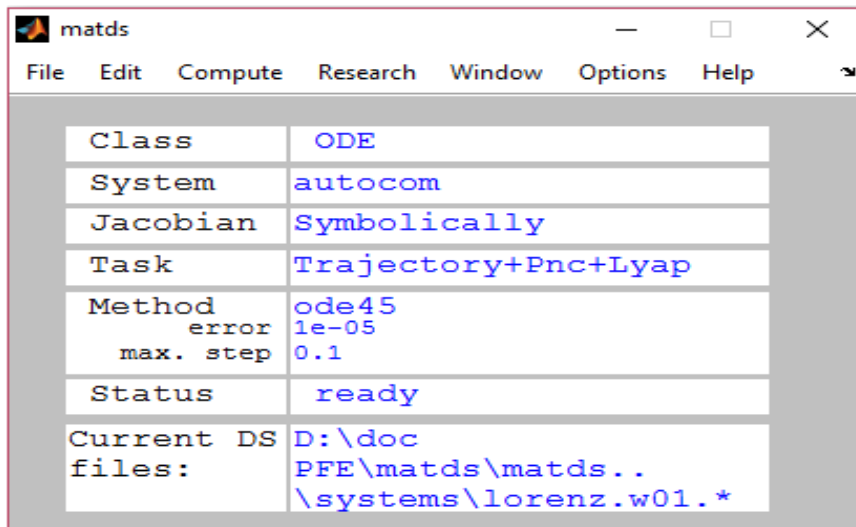


Figure 2.13. L'interface de l'outil MATDS.

- On va créer par la suite notre système chaotique Auto-commuté en introduisant les équations (2.2) (figure 2.14) en premier, puis les équations (2.3) avec les paramètres suivants : $a = 20, b = 14, c = 10.6$ et $h = 2.8$.

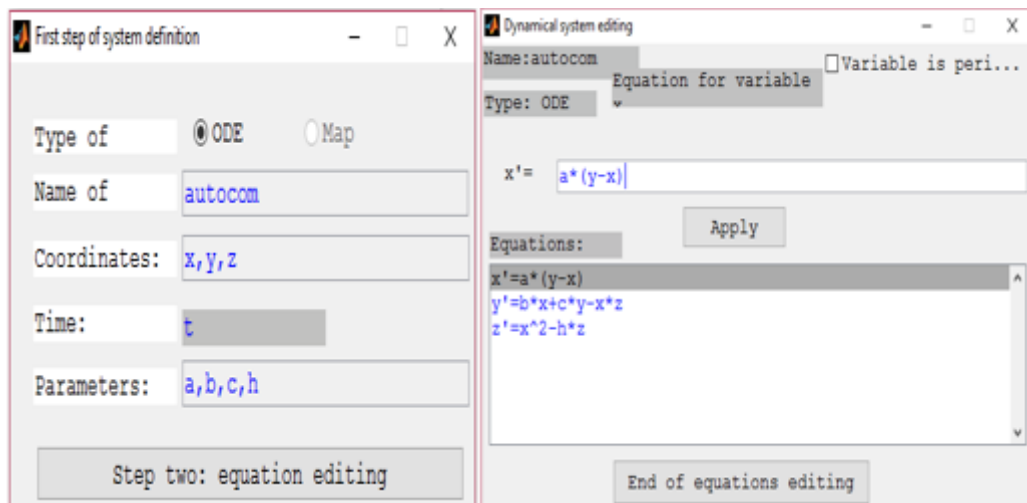


Figure 2.14. Création du système (2.2) sur MATDS.

Le MATDS nous permet de visualiser les exposants de Lyapunov ainsi que la section de Poincaré.

On sait que pour un attracteur étrange, il faut que la somme des exposants de Lyapunov soit négative et qu'au moins un de ses exposants soit positif. Pour notre système chaotique Auto-commuté, les exposants de Lyapunov dépendent principalement sur les deux sous-systèmes (2.2) et (2.3) qui sont représentés sur les deux figures suivantes :

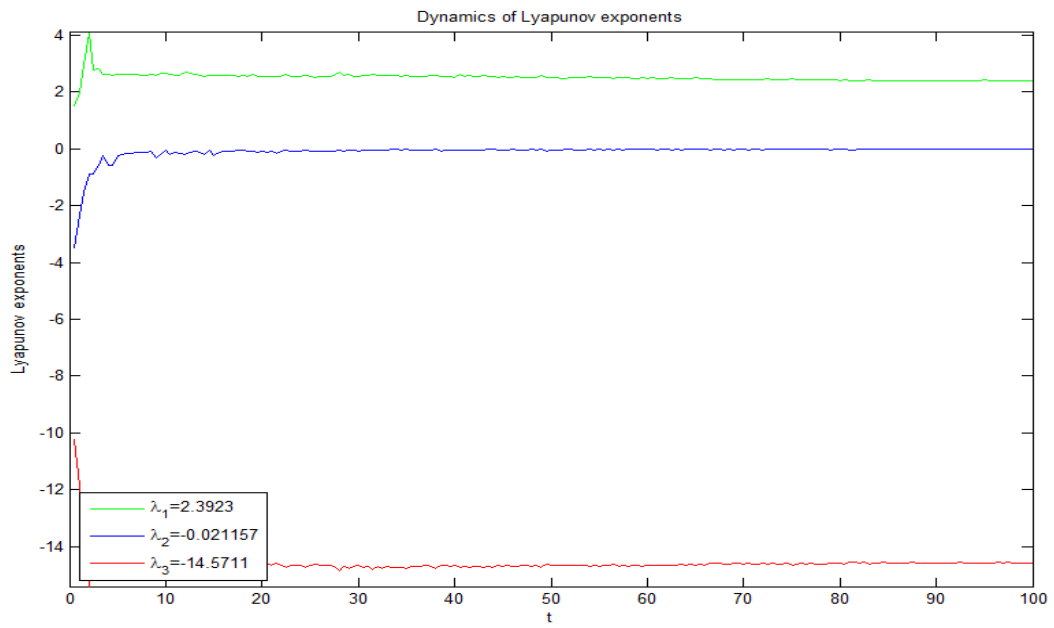


Figure 2.15. Les exposants de Lyapunov du sous-système (2.2).

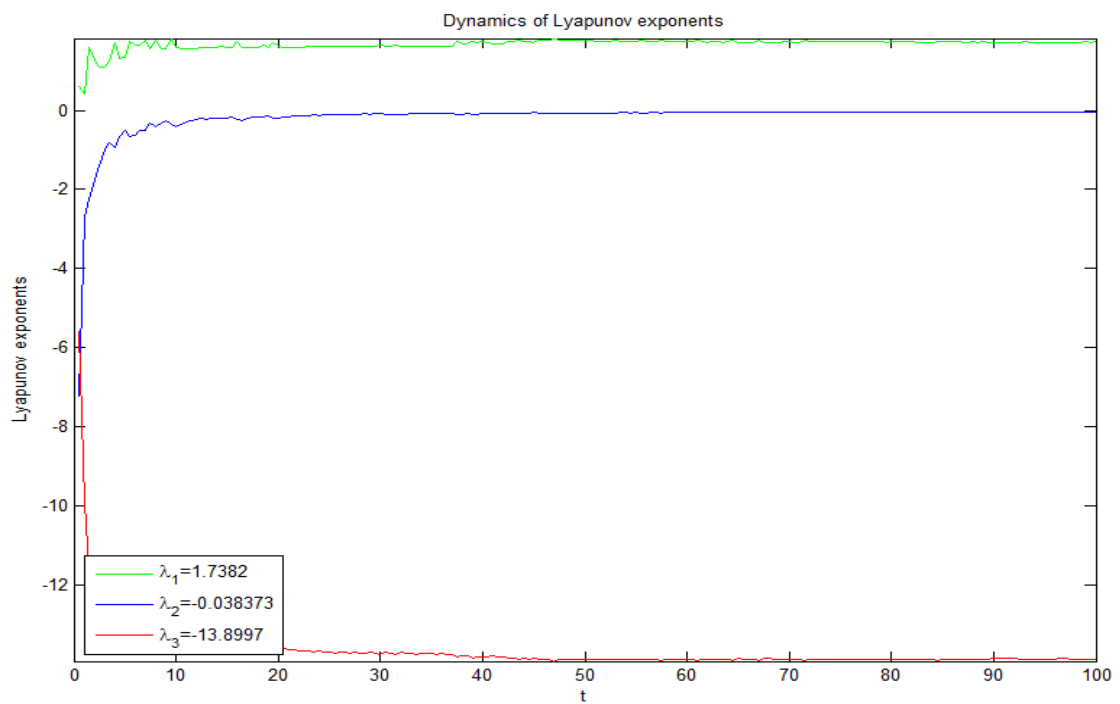


Figure 2.16. Les exposants de Lyapunov du sous-système (2.3).

Les deux sous-systèmes qui composent notre système Auto-commuté ont tous les deux trois équations différentielles, donc on obtient trois exposants de Lyapunov pour chaque sous-système tels que :

- Sous-système (2.2) : $\lambda_1 = 2.3923$, $\lambda_2 = -0.021157$, $\lambda_3 = -14.5711$.
- Sous-système (2.3) $\lambda_1 = 1.7382$, $\lambda_2 = -0.038373$, $\lambda_3 = -13.8997$.

Les résultats prouvent que les deux sous-systèmes sont chaotiques, ce qui veut dire que notre système Auto-commuté est bien chaotique vu qu'il bascule entres ces deux sous-systèmes de façon automatique.

2.3.8 Section de Poincaré

Pour la section de Poincaré, nous avons utilisé le même outil MATDS pour analyser le système Auto-commuté (2.1) en fonction des deux sous-systèmes (2.2) et (2.3).

On va tracer l'intersection du plan de phase y en fonction de x avec un plan $x = 0$, ainsi on obtient la section de Poincaré mentionné par des points rouges sur les figures suivantes :

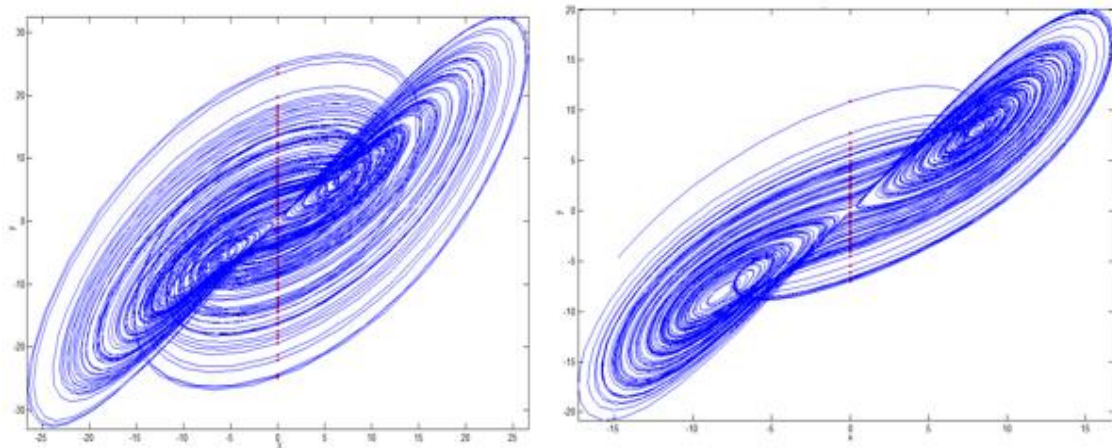


Figure 2.17. La section de Poincaré en 2D des deux sous –systèmes (2.2) et (2.3).

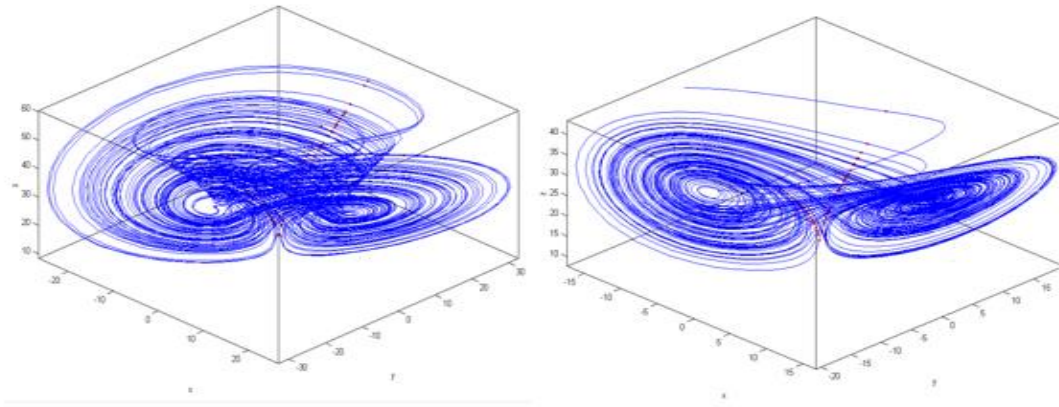


Figure 2.18. La section de Poincaré en 3D des deux sous –systèmes (2.2) et (2.3).

On supprime les trajectoires pour obtenir uniquement la section de Poincaré qui est représentée dans la figure suivante :

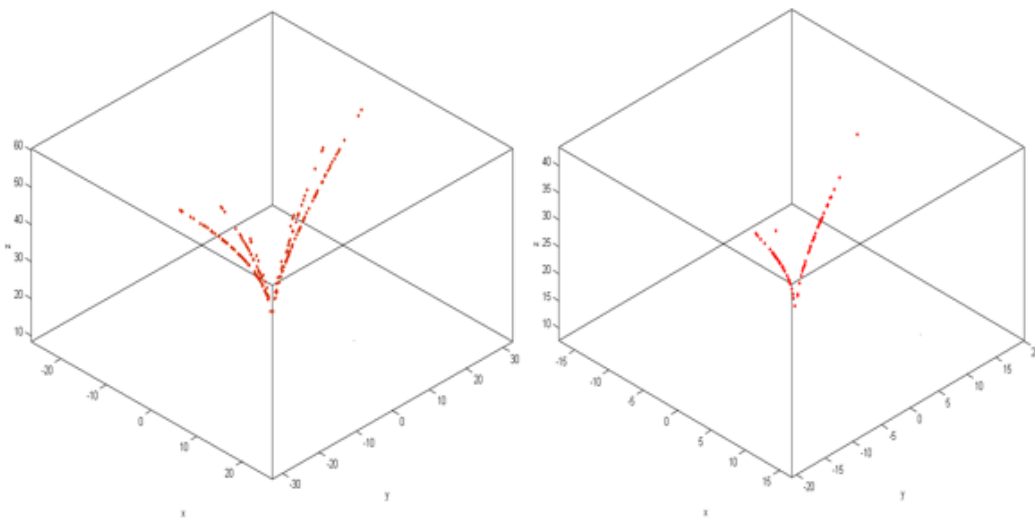


Figure 2.19. Section de Poincaré sans trajectoires des sous-systèmes (2.2) et (2.3).

2.3.9 Bifurcation

Afin d’observer l’effet des paramètres sur le système dynamique nous fixons les paramètres $a = 20, b = 14, c = 10.6$, et on fait varier h entre $[0 ; 9]$.

Pour tracer le diagramme de bifurcation, un programme sous MATLAB a été écrit et les résultats obtenus sont représentés sur la figure 2.20.

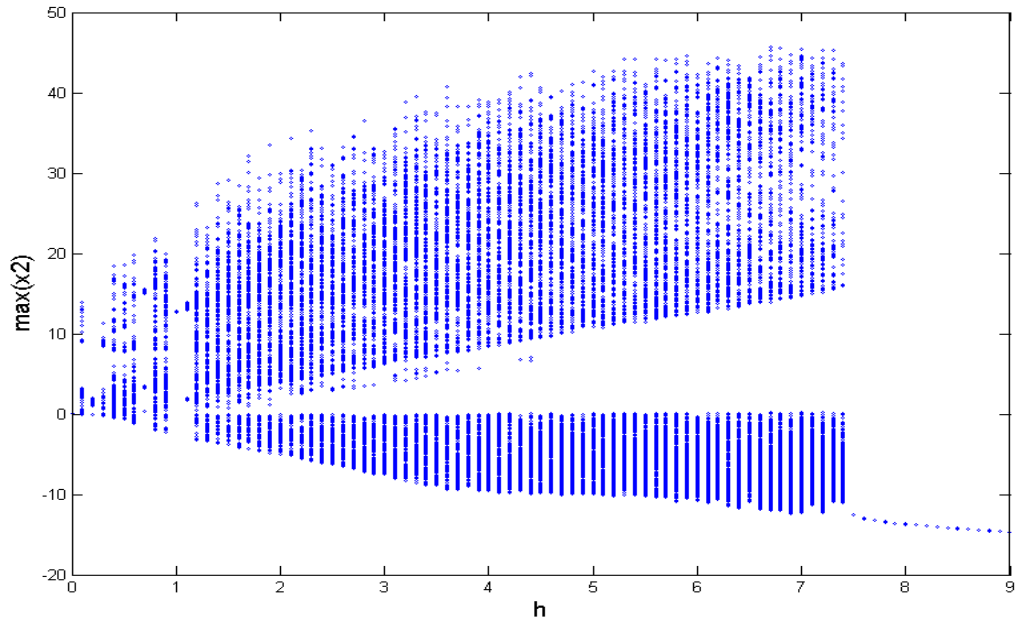


Figure 2.20. Le diagramme de bifurcation du système Auto-commuté.

On remarque que pour $h \in [0; 7.45]$, le système est chaotique, et pour $h \in [7.45 ; 9]$ le système présente un caractère périodique comme le montre les figures 2.21 à 2.24.

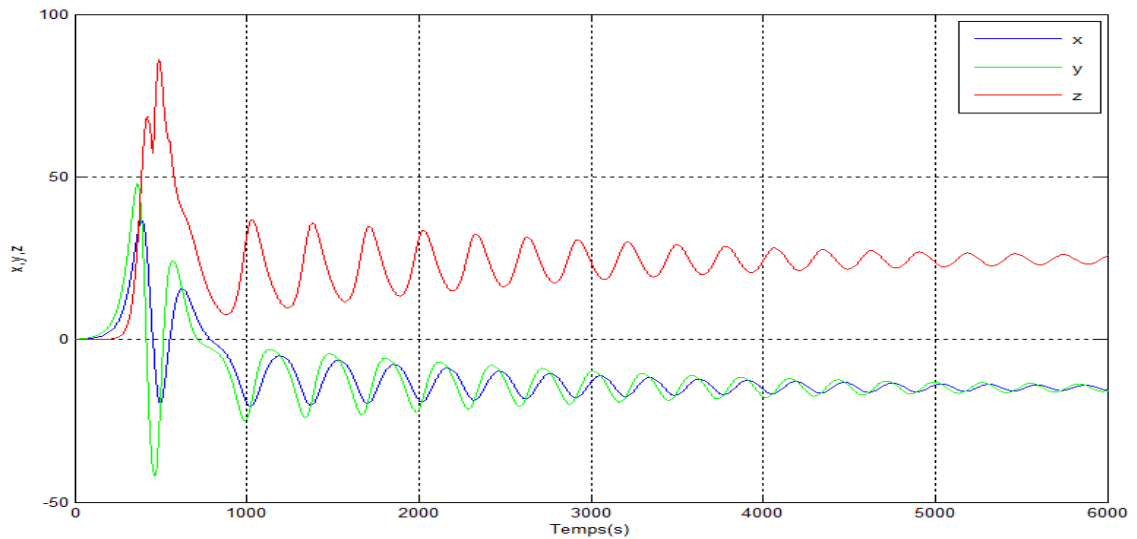


Figure 2.21. Les états x, y, z en fonction du temps lorsque $h = 9$.

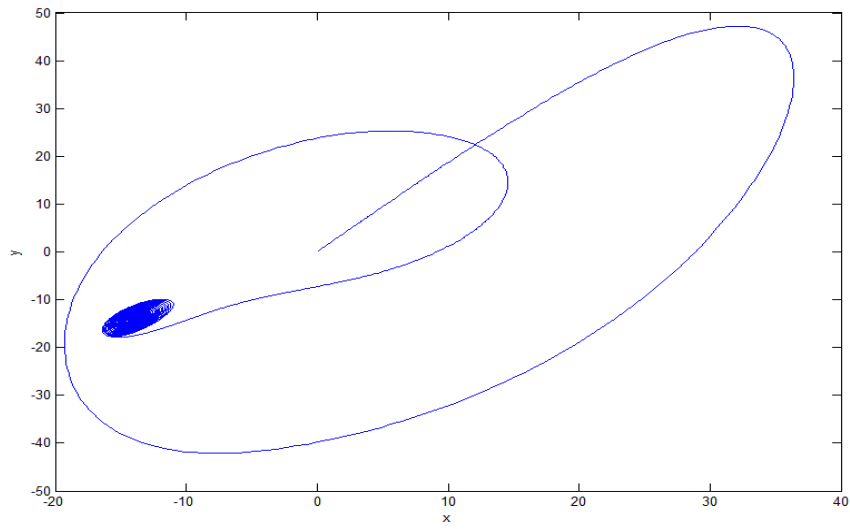


Figure 2.22. Le plan de phase x en fonction de y pour $h = 9$.

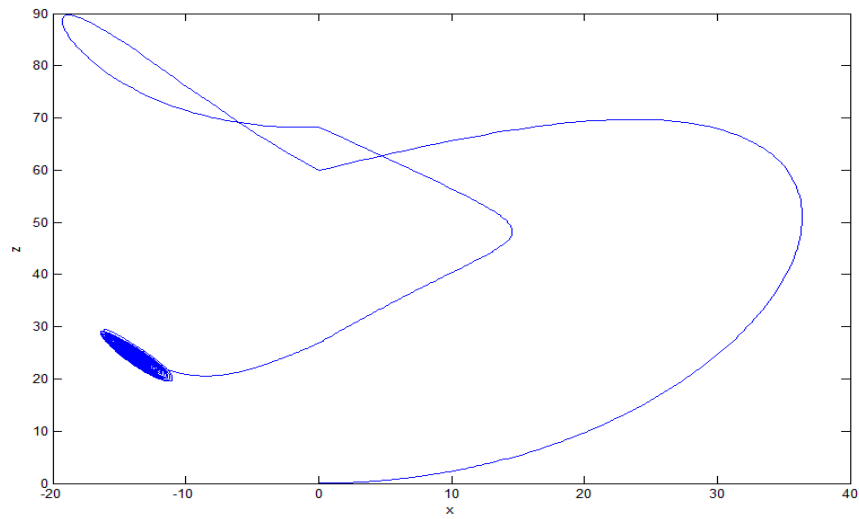


Figure 2.23. Le plan de phase x en fonction de z pour $h = 9$.

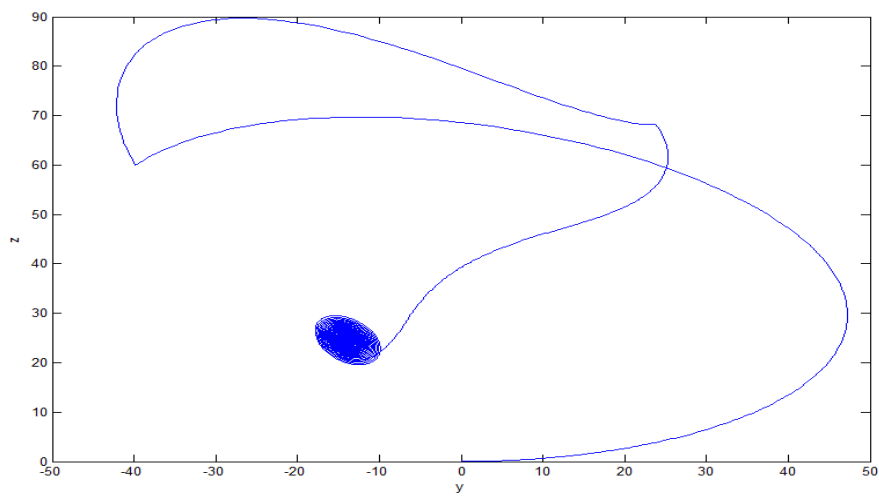


Figure 2.24. Le plan de phase y en fonction de z pour $h = 9$.

2.4 Conclusion

Dans ce chapitre, nous avons présenté les principales caractéristiques et propriétés du système chaotique Auto-commuté : point fixe, stabilité, exposant de Lyapunov, section de Poincaré, diagramme de bifurcation, etc...

Dans le chapitre suivant, l'implémentation FPGA du système chaotique Auto-commué sera décrite et le flot de conception sous environnement ISE de Xilinx sera développé.

Chapitre 3 Implémentation FPGA du système chaotique auto-commuté

3.1 Introduction

Dans les années 1970 et 1980, les systèmes électroniques ont été créés en regroupant des composants standards tels que des microprocesseurs et des puces de mémoires avec des composants logiques numériques. Au début, les FPGAs étaient considérés comme des puces logiques utilisées pour brancher les composants ensemble afin d'avoir des systèmes complexes. L'augmentation de la complexité due à l'évolution de la technologie a entraîné un écart croissant dans la portée offerte par la technologie FPGA actuelle.

Nous décrivons dans ce chapitre la technologie FPGA, puis nous expliquons par la suite d'une manière explicite la mise en œuvre logicielle et matérielle pour l'implémentation du système chaotique Auto-commuté sur FPGA à travers la carte Virtex-5-ML501 de Xilinx.

3.2 Présentation des circuits FPGAs

Les FPGAs (Field Programmable Gate Arrays ou "réseaux logiques programmables") sont des composants VLSI (Very Large Scale Integration) entièrement reconfigurables ce qui permet de les reprogrammer à volonté afin d'accélérer notablement certaines phases de calculs. L'avantage de ce genre de circuit est sa grande souplesse qui permet de les réutiliser à volonté dans des algorithmes différents en un temps très court. Le progrès de ces technologies permet de faire des composants toujours plus rapides et à plus haute intégration, ce qui permet de programmer des applications importantes [13].

3.2.1 Architecture des FPGAs

Les circuits FPGAs sont constitués d'une matrice de blocs logiques configurables (CLB) entourés de blocs d'entrée sortie configurables IOB. L'ensemble est relié par un réseau d'interconnexions programmables.

La figure 3.1 présente l'architecture générique d'un circuit FPGA.

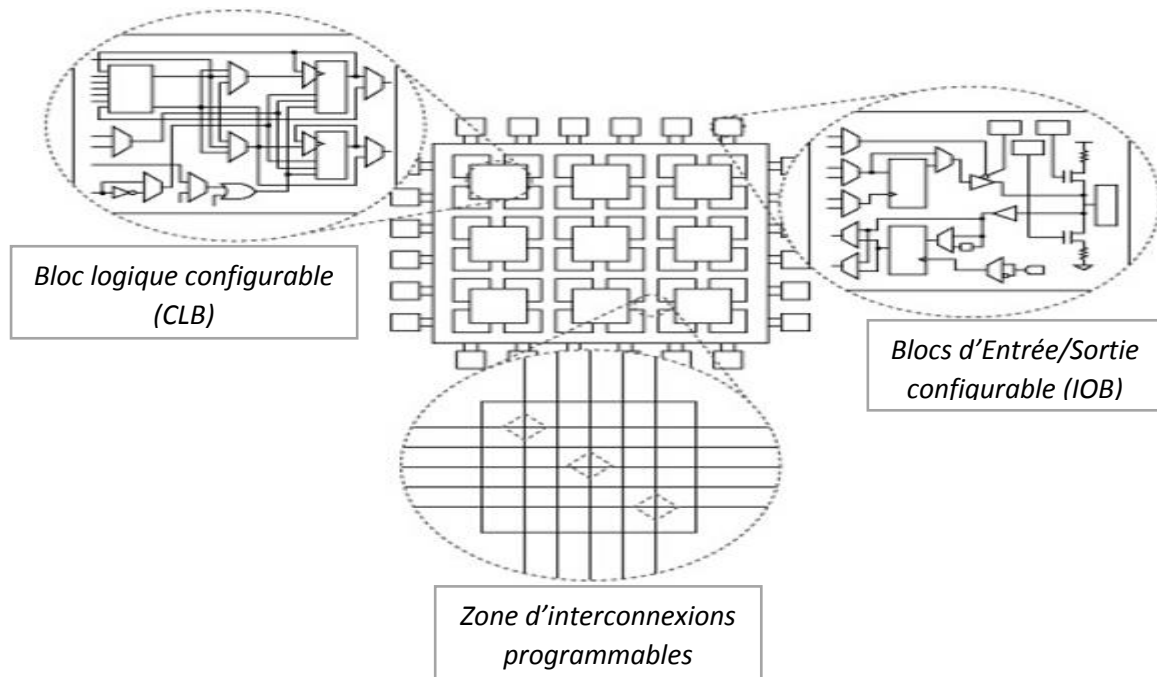


Figure 3.1. Architecture générique d'un circuit FPGA.

Les blocs logiques configurables (CLB) sont l'unité logique de base d'un FPGA. Appelés aussi cellules logiques, les CLBs sont composés de deux composants de base : Flip-Flop et table d'allocation LUT qui permet la réalisation de toute fonction logique (Look-Up-Table) (Figure 3.2) [14].

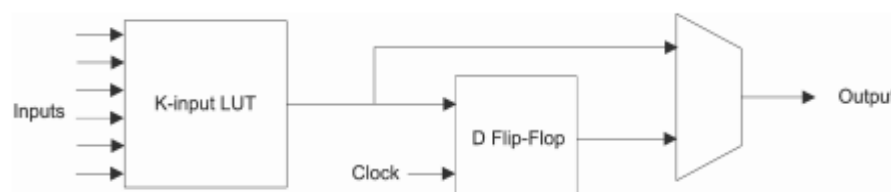


Figure 3.2. Structure de bloc logique configurable (CLB).

3.2.2 Les outils de développement des circuits FPGAs

Pour la conception sur FPGA, les deux langages de description les plus utilisés sont le Verilog et le VHDL. Tous deux font l'objet de standards IEEE afin d'uniformiser ces langages autant pour les concepteurs que les manufacturiers. Le langage utilisé dans le cadre de ce projet est le VHDL.

3.3 Présentation des différents outils logiciels

L'offre logicielle dans le domaine de conception des circuits numériques est très variée et l'un de ces environnements que nous allons exploiter au cours de ce travail est Xilinx ISE qui est un logiciel de création et de gestion de projet CAO pour circuits FPGA.

3.3.1 Présentation du logiciel ISE

Le logiciel Xilinx ISE (Integrated Software Environment) est un logiciel de description, de simulation et de programmation de circuits et systèmes numériques sur des composants programmables. C'est un logiciel qui possède dans son soft différents outils permettant la création des systèmes ou circuit numériques. D'une manière générale, le Xilinx ISE permet de réaliser toutes les étapes de conception et de programmation des circuits FPGA de Xilinx et même pour d'autres circuits programmables tel que les CPLDs, Spartan, Verilog... :

- La description de circuits numériques sous forme de schémas logiques, de machines à état finis ou en langages de description matérielle (VHDL, Verilog, ABEL).
- La compilation, la simulation comportementale.
- La synthèse, le placement routage et l'implémentation.
- La simulation temporelle et l'analyse de timing.
- La programmation sur les circuits programmables de Xilinx (CPLD et FPGA).

La figure 3.3 représente l'interface Project Navigator de l'ISE 14.2 permettant l'accès à toutes les ressources d'un projet ainsi qu'aux outils de l'implémentation.

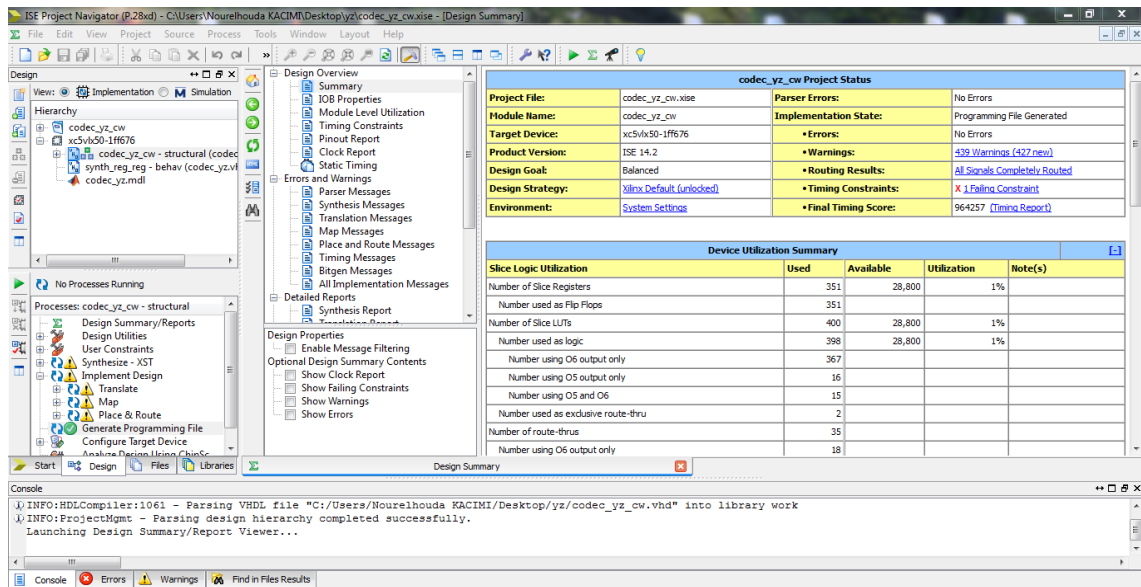


Figure 3.3. L'interface Project Navigator de l'ISE 14.2.

3.3.2 Système Generator

System Generator est un outil de conception DSP de Xilinx qui permet l'utilisation de Math Works basé sur des modèles Simulink pour la conception FPGA. Les concepteurs peuvent donc concevoir et simuler des systèmes en utilisant des blocs Xilinx dans la bibliothèque de modèles MATLAB(Simulink) (figure3.4).

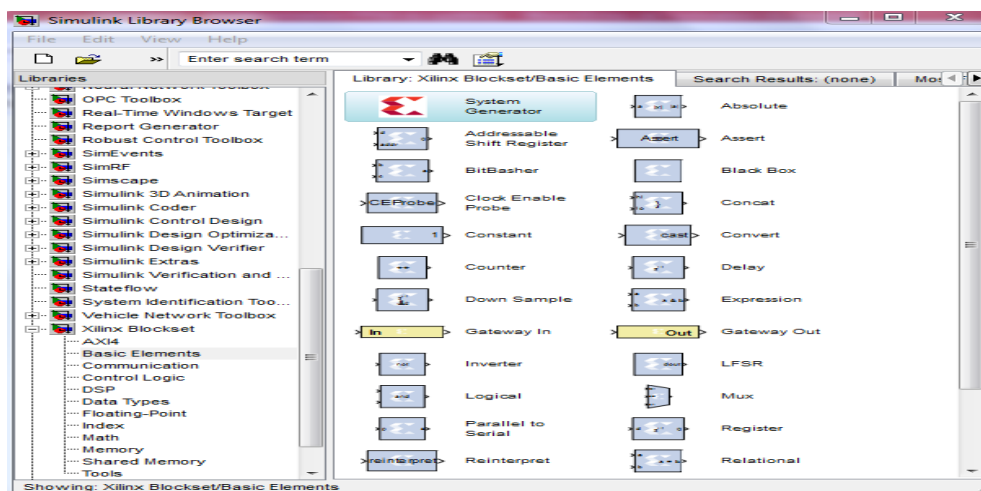


Figure 3.4. La bibliothèque Xilinx de Simulink.

Le bloc System Generator (figure 3.5) permet de contrôler les paramètres du système et la simulation, et est utilisé pour appeler le générateur de code VHDL dans notre cas. Tous les modèles Simulink contenant les blocs de Xilinx doivent contenir au moins un bloc System Generator. Une fois ce dernier ajouté à un modèle, il est possible de spécifier la manière dont la génération de code et de simulation doivent être manipulée.

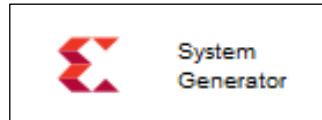


Figure 3.5. Le bloc System Generator.

3.3.3 Présentation du logiciel de simulation ModelSim

ModelSim est un logiciel de simulation pour les langages HDL tel que le VHDL, développé par Mentor Graphics. C'est un outil informatique permettant de compiler un code VHDL et en introduisant les TestBenchs, il permet de simuler le code et de le debugger en cas de dysfonctionnement.

La figure 3.6 illustre l'interface graphique de ModelSim. C'est une interface qui permet de visualiser les simulations dans une fenêtre appelée WAVE. Le programme permet aussi l'exportation des formes d'ondes pour visualisation indépendamment du programme ou sur un support papier.

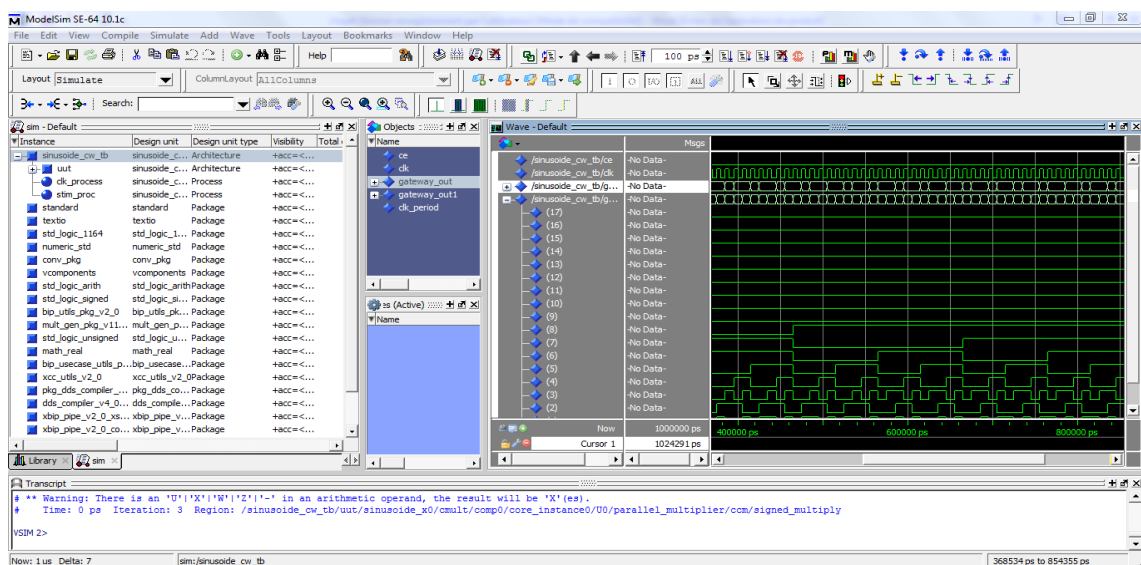


Figure 3.6. Interface graphique du logiciel ModelSim de Mentor Graphics.

3.3.4 Plate- forme de développement Virtex-5 (ML501) :

La plateforme de développement Virtex-5 (ML501) représentée sur la figure 3.7, est une plateforme de conception et de mise en œuvre de circuits numériques implémentés sur circuit FPGA.



Figure 3.7. Plateforme de développement Virtex-5-ML501.

La carte FPGA Virtex-5 a plusieurs caractéristiques et périphériques à savoir :

- Référence : Virtex-5-ML501 FPGA XC5VLX50-1FFG676.
- Puce de générateur d'horloge programmable du système.
- Horloge de fréquence 100 MHz.
- Commutateurs DIP à usage général, voyants et boutons poussoirs.
- Codec audio stéréo AC97 avec line- in, line-out, 50mW casque, entrée microphone, et SPDIF prises audio numérique.
- Port série RS-232.
- Ecran LCD 2 lignes x 16 caractères.
- Connecteur vidéo DVI (VGA pris en charge avec l'adaptateur fourni).
- Connecteurs de souris et claviers PS/2.
- Contrôleur de configuration système AC™ avec CompactFlash Type I CompactFlash connecteur.
- SRAM synchrone ZBT, 9 Mb sur le bus de données 32 bits avec quatre bits de parité

- Intel P30 StrataFlash[®] linéaires puces de mémoires flash (32Mo)
- Serial peripheral interface[™] (SPI) Flash (2Mo)
- Connecteur RJ45 Ethernet.
- Puce d'interface USB avec ports hôtes et périphériques [7].

3.4 Processus d'implémentation

La conception des architectures de commande s'effectue en utilisant les outils de conception assistée par ordinateur (CAO). La saisie est effectuée graphiquement ou via un langage de description matériel de haut niveau, nommé également langage HDL (Hardware Description Language) ; le langage HDL le plus couramment utilisé est le VHDL (Very High Speed Integrated Hardware Description Language). Il est standardisé et offre au concepteur différents niveaux de description, et surtout l'avantage d'être portable et compatible avec toutes les technologies FPGA précédemment introduites. Le synthétiseur des outils CAO génère dans un premier temps un Netlist qui décrit la connectivité de l'architecture. Après avoir vérifié la syntaxe, il génère aussi un schéma RTL et un schéma technologique qui décrivent le programme HDL. Puis l'outil de placement-routage place de façon optimale tous les composants et effectue le routage entre les différentes cellules logiques. Ces deux étapes permettent de générer un fichier de configuration à télécharger dans la mémoire de configuration du FPGA. Ce fichier est appelé Bit Stream et peut être directement chargé sur FPGA à partir d'un ordinateur [5]. L'objectif à ce stade est d'effectuer des simulations aboutissant à des résultats comparables à ceux obtenus sous Simulink.

3.5 Implémentation FPGA du système chaotique Auto-commuté

3.5.1 Simulation sous System Generator- Simulink

La simulation sous System Generator (figure 3.8) est faite de la même manière que sous Simulink de MATLAB.

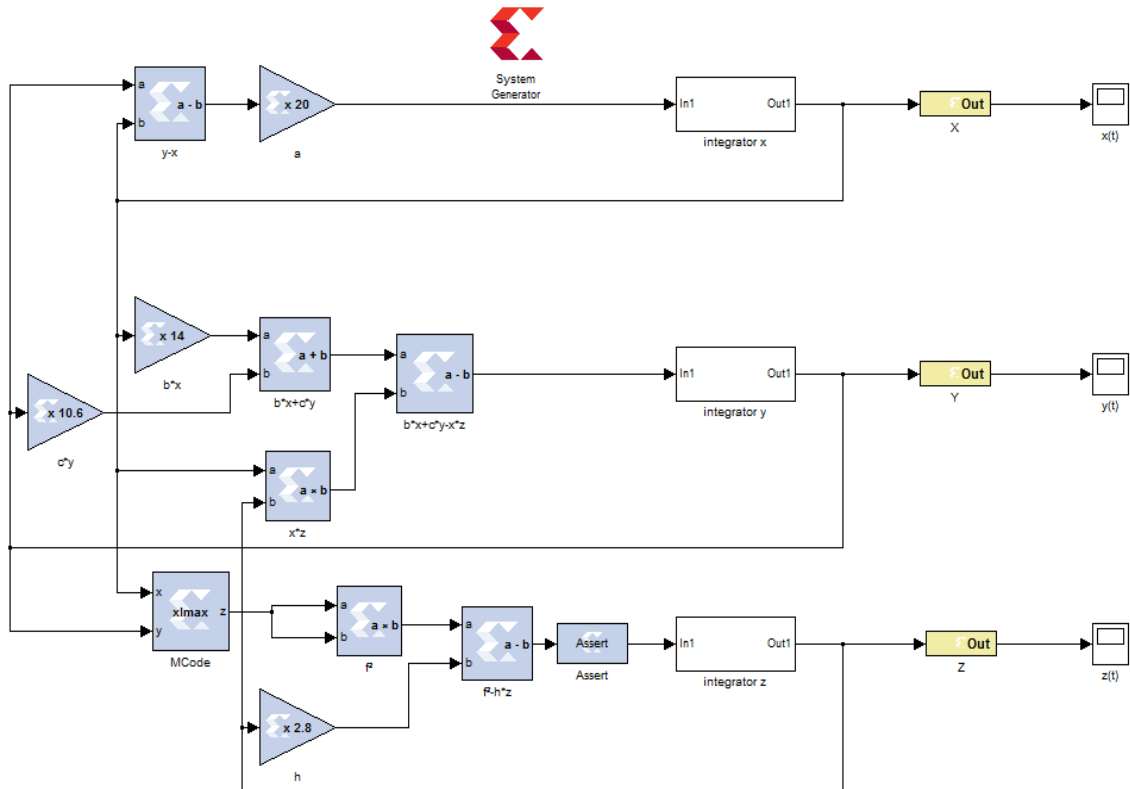


Figure 3.8. La simulation du système chaotique Auto-commuté sous System Generator.

Les intégrateurs discrets du système chaotique Auto-commuté ont tous la même structure suivante :

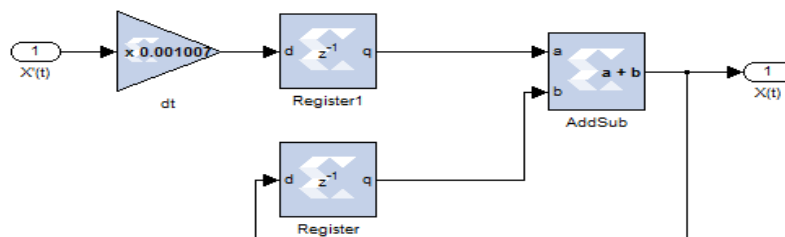


Figure 3.9. Le bloc de l'intégrateur sous Système Generator.

Le modèle de l'intégrateur est traduit en utilisant des blocs de Xilinx (figure 3.9) ; celui-ci utilise l'approximation d'Euler du premier ordre selon la formule [15] :

$$\frac{dx}{dt} = f'(x) = \frac{\Delta x}{\Delta t} = \dot{x} = \frac{x_{n+1} - x_n}{\Delta t} \quad (3.1)$$

$$x_{n+1} = \Delta t \cdot f'(x) + x_n \quad (3.2)$$

Où Δt : représente le pas d'intégration.

Les conditions initiales de l'intégrateur sont stockés dans le bloc Register dans le champ « Initial value ». Les résultats de simulation sous System Generator sont représentés dans les figures 3.10, 3.11, 3.12, 3.13 et 3.14.

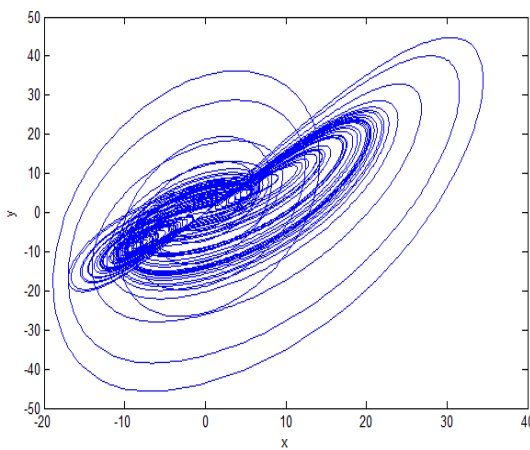


Figure 3.10. Plan de phase xy .

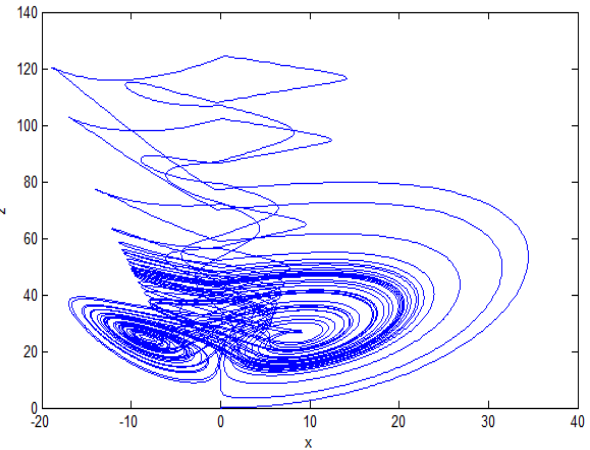


Figure 3.11. Plan de phase xz .

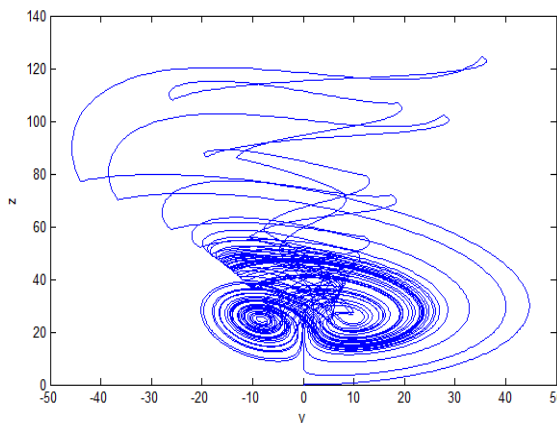


Figure 3.12. Plan de phase yz .

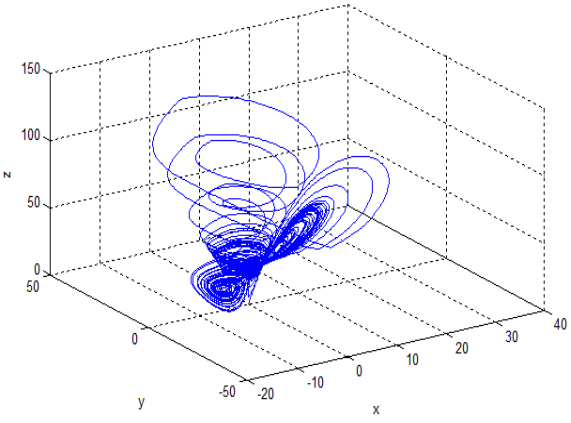


Figure 3.13. Attracteur étrange xyz .

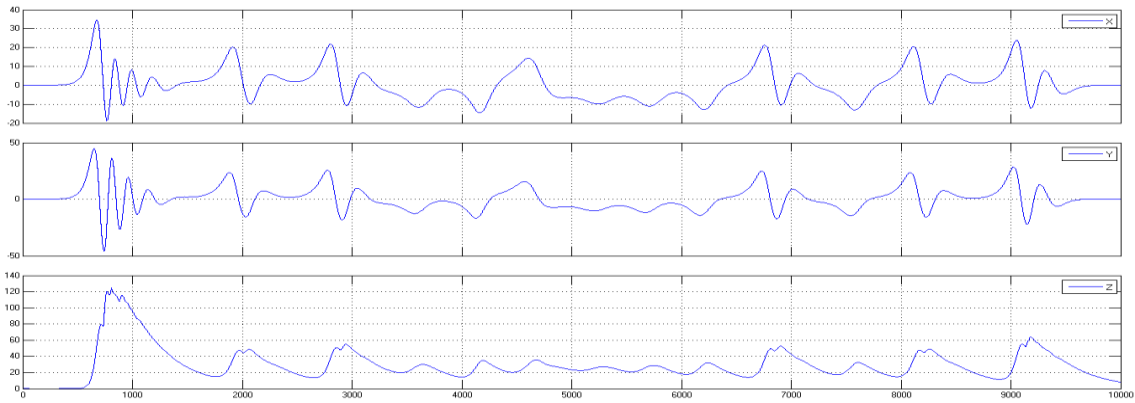



Figure 3.14. Les courbes des états x , y et z en fonction du temps.

3.5.2 Compilation du System Generator

Après la simulation sous MATLAB (System Generator), nous allons générer notre programme VHDL à partir du bloc System Generator  ; on obtient ainsi la fenêtre de la figure 3.15 dans laquelle on change les paramètres dont on a besoin, puis on génère le programme VHDL correspondant à notre système.

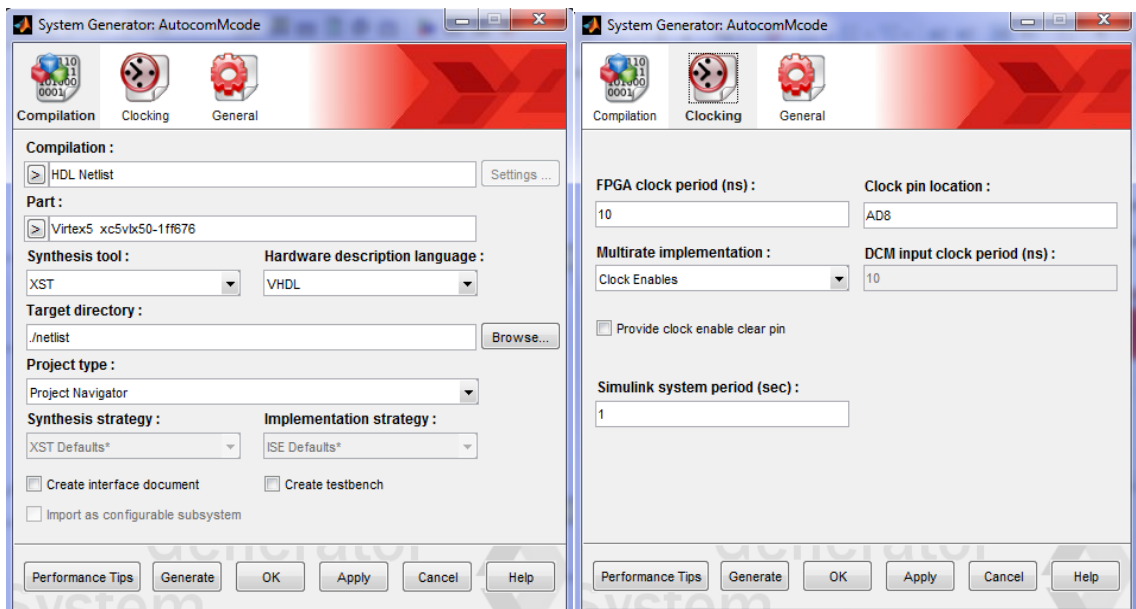


Figure 3.15. Fenêtre des paramètres System Generator.

Après la génération du programme HDL, on ouvre le programme ISE pour effectuer l'opération de simulation Modelsim représentée sur la figure 3.15.



Figure 3.16. Représentation des signaux du système chaotique Auto-commuté sous Modelsim.

3.5.3 Implémentation sur FPGA

A l'aide du contrôleur codec AC97 situé dans la carte FPGA Virtex-5-ML501, on ajoute le programme VHDL du codec AC97 correspondant à notre système chaotique Auto-commuté, puis on effectue une opération de synthèse comme le montre la figure 3.17 pour vérifier le bon fonctionnement du montage, et on génère par la suite « le Bitsteam » en cliquant sur « Generate Programming File »

Les signaux de sorties du système chaotique Auto-commuté vont subir donc une conversion numérique analogique.

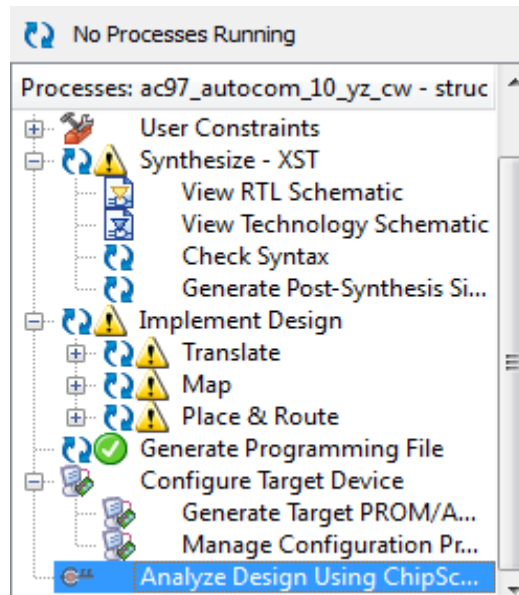


Figure 3.17. Fenêtre de synthétisation.

3.6 Conclusion

Dans ce chapitre, nous avons expliqué le processus d'implémentation sur la carte FPGA-Virtex5 du système chaotique Auto-commué à l'aide des différents outils de développement adaptés aux circuits FPGA.

L'observation et la comparaison entre les signaux obtenus par simulation sous Simulink et ModelSim et les signaux expérimentaux sera faite dans la partie expérimentale décrite dans le chapitre suivant.

Chapitre 4 Interface codec AC 97 et résultats expérimentaux

4.1 Introduction

La carte FPGA Virtex-5 ML501 ciblée par notre expérimentation est dotée d'un CODEC audio stéréo AC97.

Nous présentons dans ce chapitre un bref aperçu du protocole CODEC AC97 existant au niveau de la carte FPGA, qui va nous permettre de visualiser des différents signaux. On va comparer par la suite les signaux du système chaotique Auto-commuté obtenus par simulation sous ISE System Generator et ModelSim dans le chapitre précédent avec ceux obtenus dans la partie expérimentale par implémentation sur FPGA à travers la carte Virtex-5-ML501 de Xilinx.

4.2 Le codec AC97

L'AC97 (abréviation de Audio Codec'97, également MC'97 pour Modem Codec '97) est un standard de codec audio développé par les laboratoires d'architecture Intel en 1997. Le standard était utilisé dans les cartes mères, les modems et les cartes son. Il a été remplacé par la norme audio haute définition (HD), mais il est encore très utilisé dans les Soc (System on a chip) [16].

La carte FPGA contient un codec AC97 qui permet le traitement audio. Le circuit intégré AD1981B Codec audio, spécialisé dans la conversion de données prend en charge l'audio stéréo 20 bits avec jusqu'à 48 kHz échantillonnage. Le taux d'échantillonnage pour l'enregistrement et la lecture peut être différent.

La figure ci-dessous représente le circuit AD1981B Codec audio.

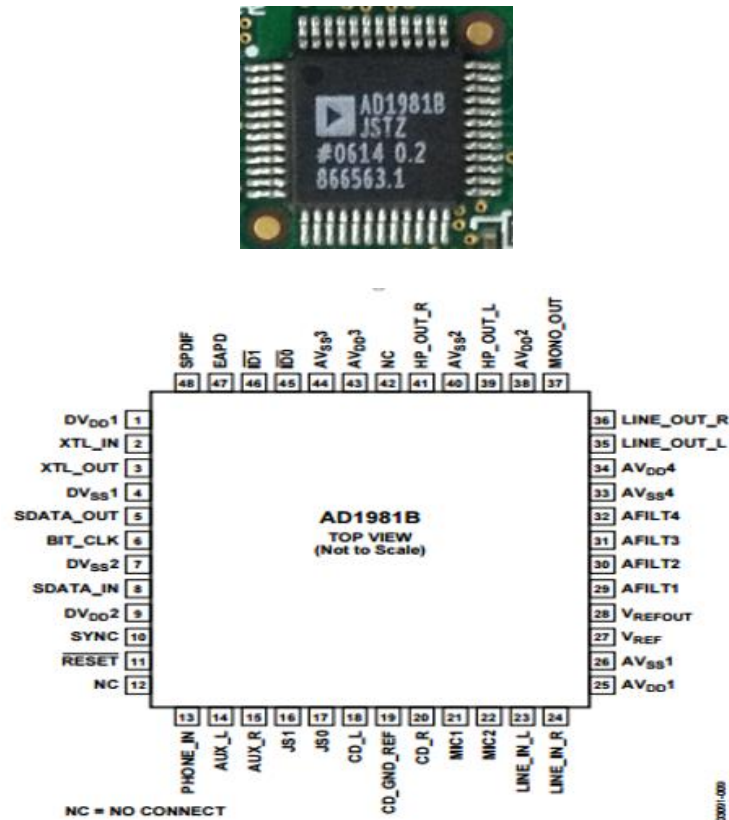


Figure 4.1. Circuit AD1981B Codec audio.

4.3 Principe de fonctionnement du CODEC AC97

Le CODEC AC97 intègre : la conversion AD et DA, le mixage des entrées-sorties analogiques pour l'audio. Il fonctionne toujours comme esclave par rapport à un contrôleur numérique AC'97, qui est généralement un contrôleur intégré dans les chipsets logiques de base.

Le CODEC AC97 peut effectuer une conversion numérique-analogique, une conversion analogique-numérique et un mélange d'entrée analogique. Il prend en charge différentes entrées / sorties audio analogiques et peut communiquer avec le contrôleur numérique via l'interface série numérique AC-Link constituée de 5 fils.

Toute communication de signaux audio ou signaux de contrôle et signaux d'état s'effectue à travers cette interconnexion série qui consiste en :

- Un signal d'horloge pour les signaux de données en série (BIT_CLK).
- Une rame de synchronisation pour codec (SYNC).
- Des données d'entrée du codec au contrôleur (SDATA_IN).
- Une sortie série (SDATA_OUT) du contrôleur au codec.
- et une remise à zéro du codec (RESET).

La configuration la plus simple et la plus utilisée est une configuration point à point de l'interface AC-Link entre le contrôleur et le CODEC comme illustré sur la figure 4.2 [17] [18].

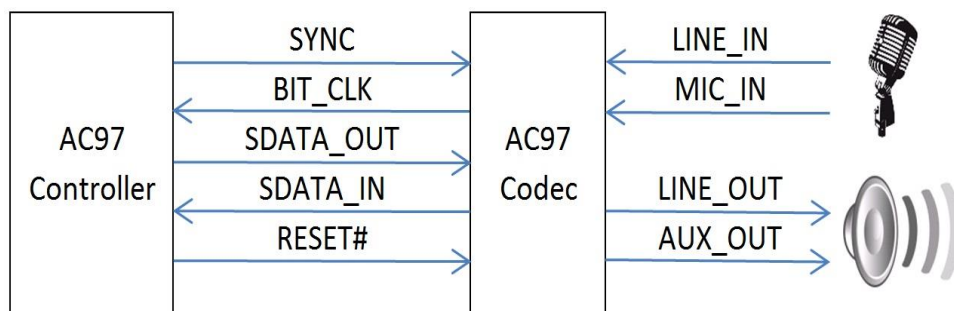


Figure 4.2. Connexions du contrôleur au CODEC AC97.

Les données circulent du CODEC vers le contrôleur et l'inverse (en totalité duplex).

- Le CODEC envoie le signal d'horloge à une fréquence fixe de 12,288 MHz (à savoir Bit_Clock) au contrôleur.
- Le signal de sortie de données sortant du contrôleur (SDATA_OUT) est en transition sur les bords croissants (les fronts montants) du Bit_Clock.
- Les données entrant dans le CODEC (SDATA_IN) sont échantillonnées sur les fronts descendants du Bit_Clock.
- Le signal de synchronisation (SYNC) est envoyé par le contrôleur au CODEC pour lui faire savoir qu'une nouvelle salve de données arrive, et doit avoir une fréquence fixe de 48 kHz.
- Le signal de réinitialisation (RESET) est donné par le contrôleur au CODEC ; il s'agit d'une réinitialisation à froid asynchrone, ce qui signifie que les registres CODEC sont initialisés sur leurs valeurs par défaut.

Des prises Jacks audio séparés sont prévus pour Microphone (MIC_IN), LINE_IN, LINE_OUT, et casque (AUX_OUT). Toutes les prises Jacks audio sont stéréo, sauf le microphone.

4.4 Expérimentation et résultats

La première phase de cette expérimentation consiste en l'élaboration du modèle du codec AC97 sous Simulink utilisant des blocs System Generator permettant la génération des Netlists HDL lors de l'implantation sur le FPGA. Il est représenté sur la figure 4.3.

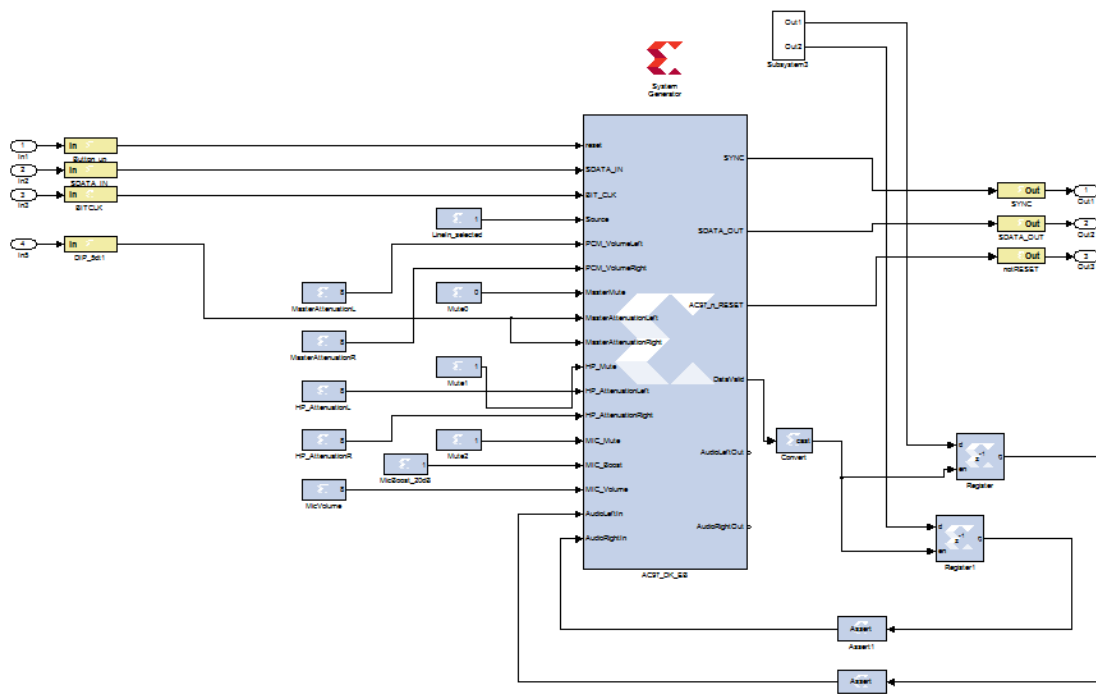


Figure 4.3. Codec AC97 sous System Generator.

La deuxième étape de notre travail avait pour but l'implémentation sur FPGA. Nous allons connecter la carte ML501 Virtex-5 par câble USB au PC. Ensuite on clique sur « configure Target Device ». Une fenêtre s'ouvre (figure 4.4), permettant d'implémenter le fichier BitStream sur la carte FPGA Virtex-5 : on clique sur bouton droit « initialize chain » puis sur « program ».

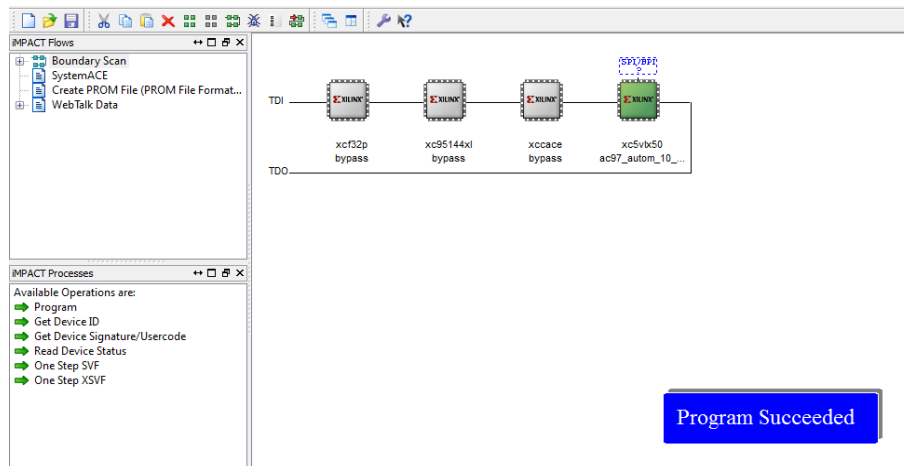


Figure 4.4. Interface permettant la programmation du FPGA.

La figure 4.5 représente la partie expérimentale de l'implémentation du système chaotique Auto-commuté sur la carte FPGA.

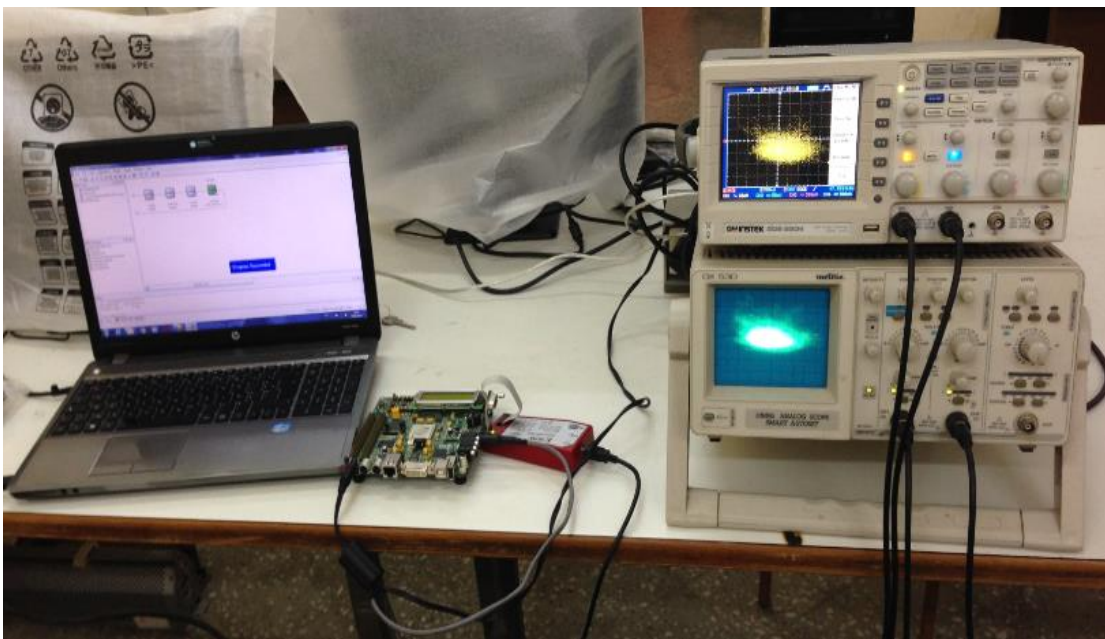


Figure 4.5. Implémentation du système chaotique Auto-commuté sur la carte FPGA.

4.4.1 Visualisation des signaux

Les figures 4.6 à 4.14 représentent les différents signaux obtenus par la simulation et les signaux expérimentaux obtenus à l'aide de l'oscilloscope numérique.

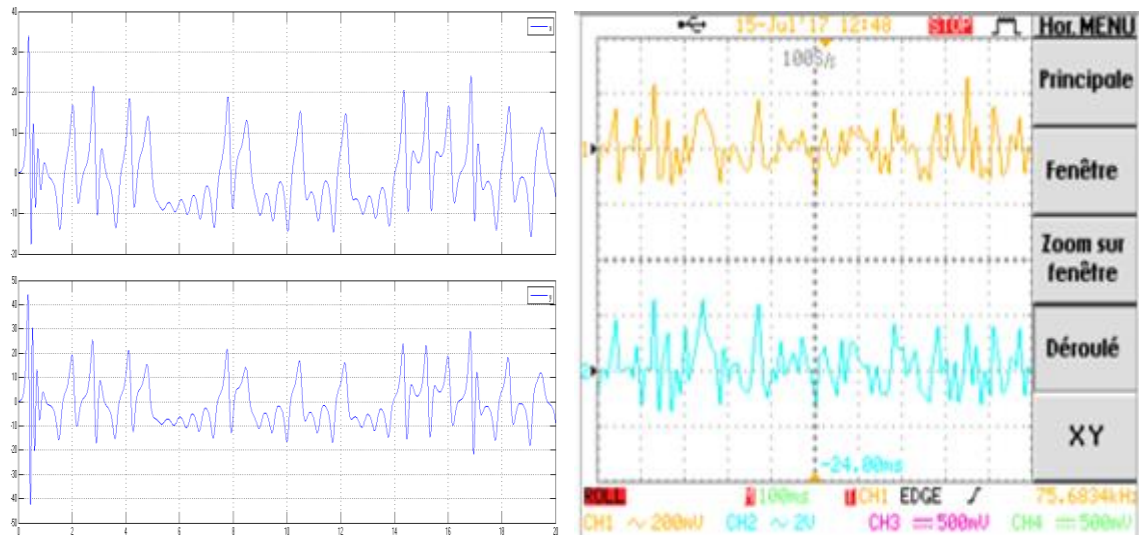


Figure 4.6. Les états x, y de en fonction du temps.

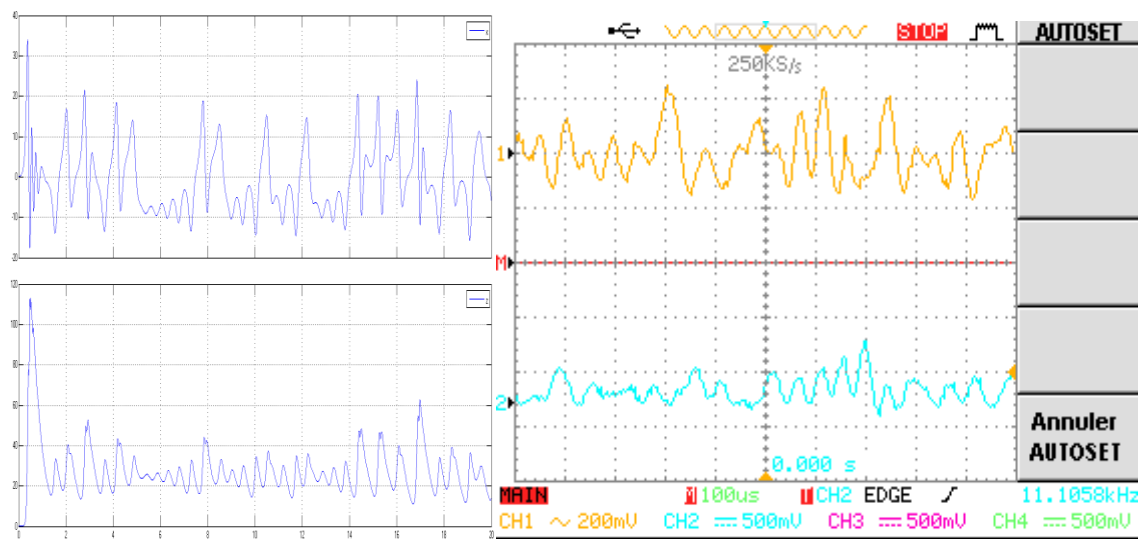


Figure 4.7. Les états x, z de en fonction du temps.

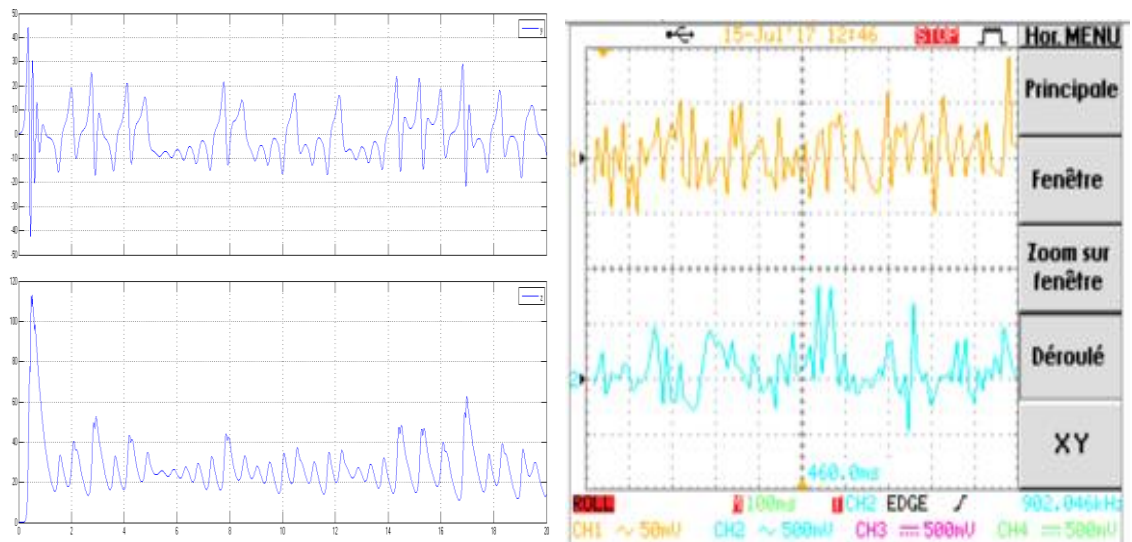


Figure 4.8. Les états y, z de en fonction du temps.

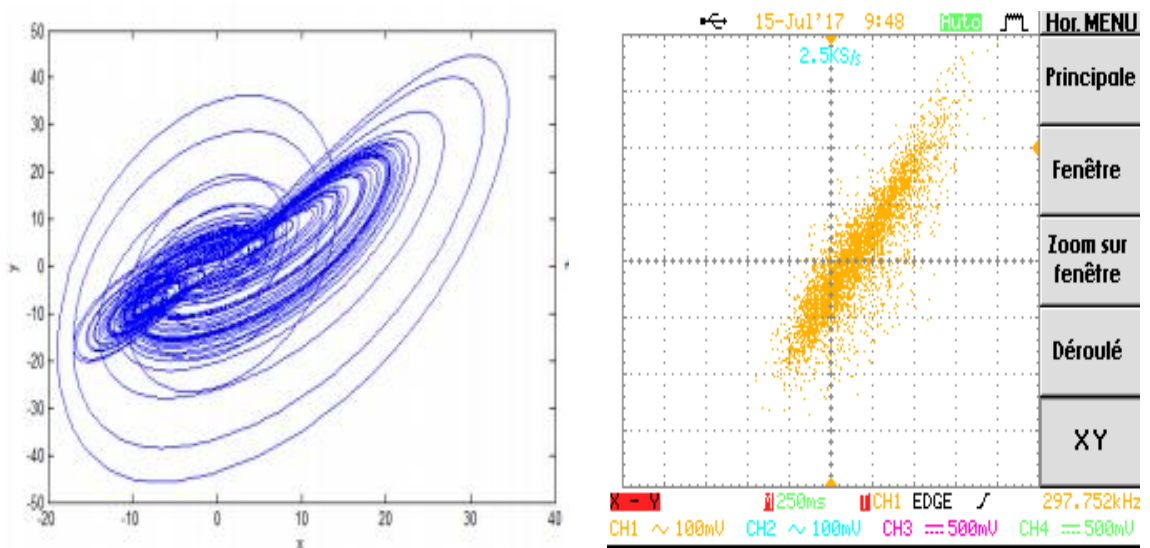


Figure 4.9. Plan de phase xy .

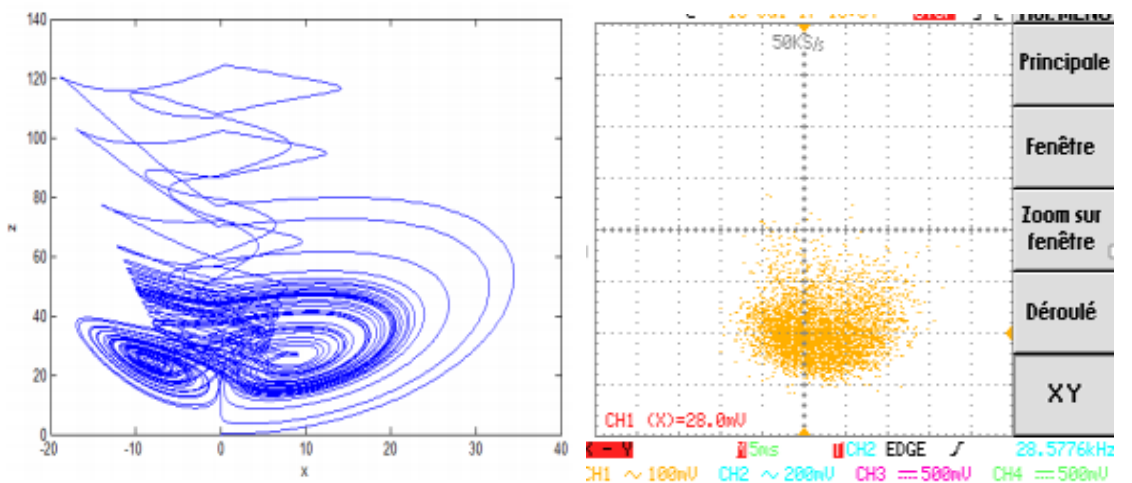


Figure 4.10. Plan de phase xz .

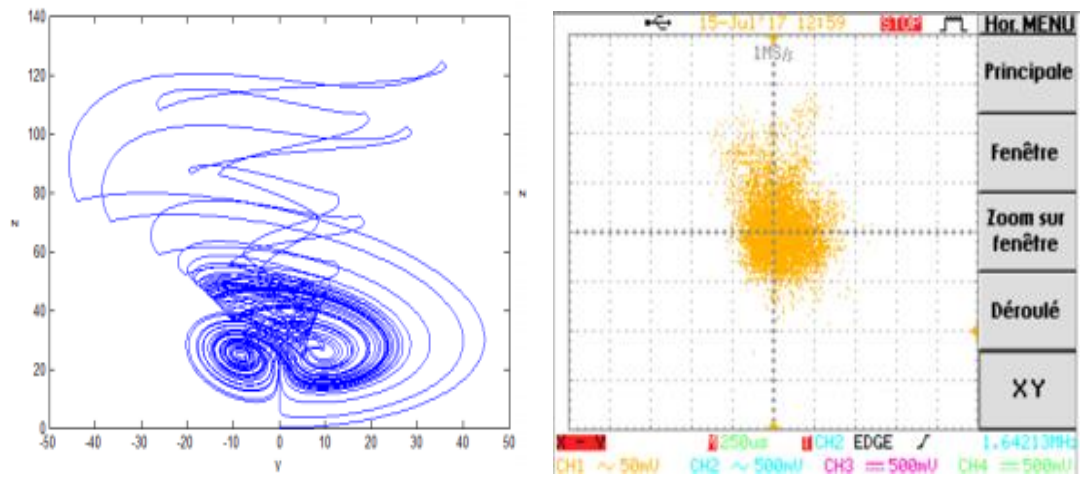


Figure 4.11. Plan de phase yz.

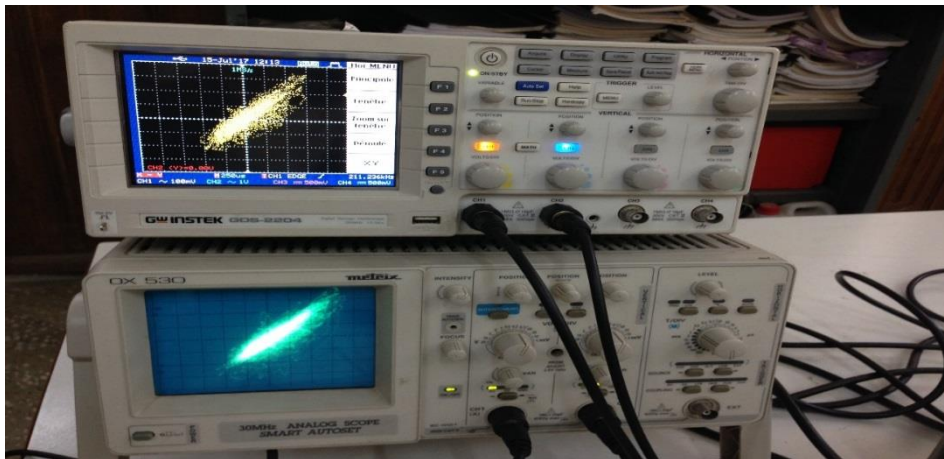


Figure 4.12. Plan de phase xy sur l'oscilloscope numérique et analogique.

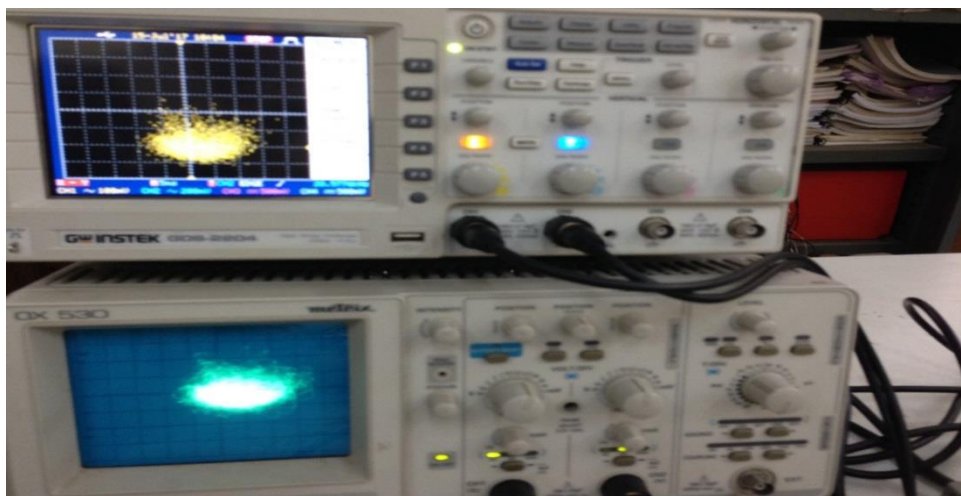


Figure 4.13. Plan de phase xz sur l'oscilloscope numérique et analogique.

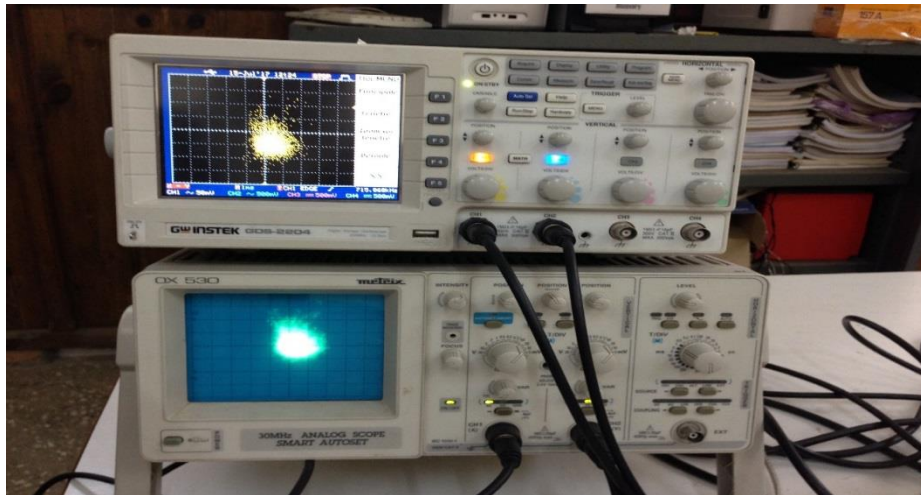


Figure 4.14. Plan de phase yz sur l'oscilloscope numérique et analogique.

Les signaux obtenus nous ont permis de valider les modèles proposés pour une implantation hardware sur la carte de développement Xilinx Virtex-5 ML-501.

Les figures 4.5 à 4.14 présentent les résultats en temps réel et montrent le bon fonctionnement du processus de l'implémentation. L'environnement ISE fournit un rapport donnant toutes les ressources utilisées lors de l'implémentation sur FPGA.

La figure 4.15 représente un tableau contenant toutes les ressources utilisées liées au design : il les comptabilise en nombre et en pourcentages.

ac97_autom_10_xz_cw Project Status			
Project File:	ac97_autom_10_xz_cw.xise	Parser Errors:	No Errors
Module Name:	ac97_autom_10_xz_cw	Implementation State:	Programming File Generated
Target Device:	xc5vlx50-1ff676	• Errors:	No Errors
Product Version:	ISE 14.2	• Warnings:	506 Warnings (155 new)
Design Goal:	Balanced	• Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	X 1 Failing Constraint
Environment:	System Settings	• Final Timing Score:	956115 (Timing Report)

Device Utilization Summary				
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	351	28,800	1%	
Number used as Flip Flops	351			
Number of Slice LUTs	396	28,800	1%	
Number used as logic	394	28,800	1%	
Number using O6 output only	363			
Number using O5 output only	16			
Number using O5 and O6	15			
Number used as exclusive route-thru	2			
Number of route-thrus	42			
Number using O6 output only	18			

Figure 4.15. Ressources consommées par l'implémentation.

4.5 Conclusion

Le présent travail avait pour but l'implantation sur FPGA d'un système chaotique Auto-commuté et l'utilisation de ressources matérielles disponibles sur la carte de développement ML-505 à base de Virtex-5 de Xilinx.

Les résultats obtenus ont montré la concordance entre les signaux obtenus par simulation dans le chapitre 3 et les signaux relevés au niveau de l'oscillateur numérique, validant ainsi le bon fonctionnement du processus de l'implémentation.

Conclusion générale

Dans ce mémoire, nous avons fait l'étude du système dynamique chaotique Auto-commuté, avec une implémentation FPGA sur carte Xilinx Virtex-5.

Après une étude détaillée des systèmes dynamiques chaotique, le système chaotique Auto-commuté a été analysé ainsi que son évolution vers le régime chaotique. Par la suite, en utilisant les différents outils de développement et en particulier System-Generator de Xilinx, l'implémentation FPGA du système chaotique Auto-commuté a été réalisée sur la carte Xilinx Virtex-5.

Le premier chapitre a été consacré à l'étude des systèmes dynamiques chaotiques. Leurs principales caractéristiques ont été décrites en mettant en évidence l'intérêt du calcul des points fixes ainsi que la bifurcation qui explique la transition vers le chaos. Nous avons également donné des exemples classiques des systèmes chaotiques utilisés par la communauté scientifique.

Dans le deuxième chapitre, nous avons étudié le comportement du système chaotique Auto-commuté. Ses principales propriétés ont été mises en évidence à l'aide de simulations. Nous avons expliqué le principe de fonctionnement du système chaotique Auto-commuté en étudiant les différents comportements de ce système en fonction des variations de variables d'états. Nous avons utilisé MATLAB Simulink pour visualiser les différents signaux, le plan de phase, les attracteurs étranges du système étudié et les exposants de Lyapounov.

Le chapitre 3 explique les étapes de l'implémentation FPGA de notre système étudié sur la carte FPGA avec les différents outils de développement. Nous constatons que la recherche dans ce domaine (implémentation d'un système chaotique sur des circuits numériques à architecture reconfigurable FPGA) est complexe, mais le travail sous System Generator de Xilinx facilite la tâche car il traduit directement le schéma fonctionnel de notre système sous Simulink en langage VHDL.

Dans le dernier chapitre, la partie expérimentale est décrite, par l'analyse du codec AC97 existant au niveau de la carte FPGA permettant la visualisation des différents signaux.

Nous avons finalisé notre travail avec une comparaison entre les résultats de simulation numérique et les résultats expérimentaux obtenus au niveau de l'implémentation FPGA. Les différents signaux visualisés obtenus sur l'oscilloscope numérique sont en parfaite concordance avec ceux obtenus par simulation sous Matlab Simulink et sous ModelSim.

La contribution principale de ce travail a été l'implémentation du système chaotique sur circuit FPGA et nous avons montré par simulation et par réalisation expérimentale que les résultats sont identiques. Nous avons constaté que ce domaine est en plein développement, avec une continuité de propositions et d'innovations de diverses origines.

Finalement, l'approche traitée au cours de ce travail peut être encore avantageusement amélioré et facilement étendue sur plusieurs domaines et sur d'autres types de systèmes chaotiques.