

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 Réseaux et Télécommunications

Présenté par

**Meriem BENRAHMOUNE**

&

**Mohamed MOUSSERATI**

---

# IMPLEMENTATION FPGA D'UNE TRANSMISSION SÉCURISÉE PAR LE SYSTÈME CHAOTIQUE UNIFIÉ

---

Proposé par **CHIKHI Mohamed Lazhar**

Année Universitaire 2016-2017

# Mémoire de Master

Mention Électronique  
Spécialité Réseaux et Télécommunications

Présenté par

**Meriem BENRAHMOUNE**  
&  
**Mohamed MOUSSERATI**

---

## **IMPLEMENTATION FPGA D'UNE TRANSMISSION SÉCURISÉE PAR LE SYSTÈME CHAOTIQUE UNIFIÉ**

---

Proposé par **CHIKHI Mohamed Lazhar**

Année Universitaire 2016-2017

## Remerciements

---

*Nous remercions ALLAH de nous avoir donné la santé et le courage afin de pouvoir réussir ce travail. Ce travail est l'aboutissement d'un long cheminement au cours duquel nous avons bénéficié d'encadrement, des encouragements et du soutien de plusieurs personnes, à qui nous tenons à dire profondément et sincèrement merci.*

*Nous exprimons notre grande gratitude à notre promoteur Mr. CHIKHI Mohamed Lazhar, d'avoir accepté de nous encadrer durant notre travail et pour ses précieux conseils et ses orientations. Nous avons eu le privilège de travailler parmi votre équipe et d'apprécier vos qualités et vos valeurs, votre sérieux et votre compétence.*

*Nous remercions très chaleureusement Mr. KAZED pour son soutien, ses précieux conseils tout au long de notre cursus universitaire.*

*Nous remercions également tous les membres du jury pour nous avoir honorés par leur présence et pour avoir accepté d'évaluer notre travail.*

*Nous voudrions aussi remercier toute l'équipe du laboratoire LABSET qui nous a apporté leur soutien tout au long de ce travail.*

*Nous tenons à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseignés et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.*

**MERCI.**

# *Dédicaces*

*À l'homme de ma vie, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, que dieu te garde dans son vaste paradis, à toi mon père.*

*À la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur ; maman que j'adore.*

*Aux personnes dont j'ai bien aimé la présence dans ce jour, à mon mari Touhami, à mes frères Mohamed et Achraf, mes sœurs Soumia et Ahlem, et mon neveu Adem, je dédie ce travail dont le grand plaisir leurs revient en premier lieu pour leurs conseils, aides, et encouragements.*

*À toute ma famille, et mes amis.*

*À mon binôme MOUSSERATI Mohamed et toute sa famille en particulier sa petite fille Meriem.*

*Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit une réussite, je vous dis merci.*

*Meriem.*

# *Dédicaces*

*À ma mère, à ma mère et à ma mère.*

*À mon père.*

*Aux deux personnes que j'aime le plus, pour leurs prières, soutiens, conseils et sacrifices, que dieu vous bénisse et garde une place pour vous au paradis INCHALLH.*

*À ma petite princesse Meriem Assil.*

*À ma femme, pour sa patience, son soutien et ses encouragements durant ce travail et dans la vie.*

*À mes chers frères et sœurs*

*À toute ma famille*

*À mes amis et mes collègues*

*À mon binôme Meriem et à toute sa famille*

*Et à tous ceux qui ont contribué de près ou de loin pour la réussite de ce projet, je vous dis MERCI.*

*Mohamed.*

---

**ملخص :** في هذا العمل قمنا بتعديل و تنفيذ تطبيق في دارة FPGA لنظام الارسال الفوضوي المقترح من طرف C.J. Cheng. يعتمد هذا النظام على التزامن التكيفي بين نظامين فوضويين مختلفين نوع Unifié مجهولي الإعدادات. تدرج الرسالة بطريقة الجمع مع حالة (état) من المرسل وترسل ضمن جميع الحالات الأخرى الى المستقبل، أين يتم إعادة تشكيلها بعد القيام بعملية المزامنة مع المرسل استنادا الى وحدة التحكم التكيفية. تم عرض نتائج المحاكاة والمحاكاة المشتركة (Co-Simulation) اعتمادا على البيئة Xilinx System Generator – Matlab Simulink، إضافة الى النتائج التجريبية لتطبيق المرسل الفوضوي في دارة FPGA.

**كلمات المفاتيح :** نظام فوضوي ؛ الجاذب ؛ تقريع ؛ تزامن ؛ محاكاة مشتركة ؛ تشفير ؛ تطبيق ؛ Ac'97 ؛ FPGA.

---

**Résumé :** Ce travail consiste en une amélioration et une implémentation FPGA du système de transmission chaotique proposé par C.J. Cheng basé sur une synchronisation adaptative de deux différents systèmes chaotiques Unifié, dont les paramètres sont inconnus. Le message est inséré par addition avec un état de l'émetteur et transmis avec l'ensemble des autres états vers le récepteur pour être reconstitué après synchronisation des états émetteur – récepteur à l'aide du contrôleur adaptatif. Les résultats de simulation et de Co-Simulation de l'ensemble du système sous environnement Xilinx System Generator - Matlab Simulink sont présentés ainsi que les résultats expérimentaux de l'implémentation FPGA de l'émetteur chaotique.

**Mots clés :** Système chaotique ; attracteur ; bifurcation ; exposant et stabilité de Lyapunov ; synchronisation ; cryptage ; Co-Simulation ; implémentation ; Ac'97 ; FPGA.

---

**Abstract :** In this work, an improvement and an FPGA implementation of the chaotic transmission system proposed by C.J. Cheng. This system is based on a chaotic adaptive synchronization between two different unified chaotic systems with uncertain parameters. The message is inserted by addition method with a state of the transmitter and transmitted with all the other states to the receiver to be reconstituted after synchronization of the receiver states with the emitter using the adaptive controller. The results of simulation and co-simulation of the whole system under Xilinx System Generator - Matlab Simulink environment are presented as well as the experimental results of the chaotic emitter FPGA implementation.

**Keywords :** Chaotic system ; attractor ; bifurcation ; Stability and Lyapunov exponent ; synchronization; encryption; Co-Simulation; Implementation; Ac'97; FPGA.

---

## Listes des acronymes et abréviations

AC'97	: Audio Codec 1997.
ASIC	: Application-Specific Integrated Circuit.
ASMBL	: Advanced Silicon Modular Block.
BRAM	: Block Runder Access Memory.
CAN	: Convertisseur Analogique Numérique.
CAO	: Conception Assistée par Ordinateur.
CLB	: Configurable Logic Block.
CNA	: Convertisseur Numérique Analogique.
CPLD	: Complex Programmable Logic Device.
CPU	: Central Unit Processing.
CSK	: Chaos Shift Keying.
DDR SDRAM	: Double Data Rate Synchronous Dynamic Random Access Memory.
DSP	: Digital Signal Processing.
DVI	: Digital Visual Interface.
EEPROM	: Electrically Erasable Programmable Read-Only Memory.
EPROM	: Erasable Programmable Read-Only Memory.
EQM	: Erreur Quadratique Moyenne.
FIFO	: First-In First-Out.
FPGA	: Field Programmable Gate Array.
HDL	: Hardware Description Language.
HP	: Haut-Parleur
IOB	: Input Output Block.
ISE	: Integrated Software Environment.
JHDL	: Just-another Hardware Description Language.
JTAG	: Joint Test Action Group.
LCD	: Liquid Crystal Display.
LED	: Light-Emitting Diode.
LUT	: Look-Up Table.

Matlab-GUI	: Matlab Guide User Interface.
PROM	: Programmable Read-Only Memory.
SDATA	: Serial data.
SNR	: Signal to Noise Ratio.
SPDIF	: Sony/Philips Digital Interface.
SPI	: Serial Peripheral Interface.
SRAM	: Static Random Access Memory.
UCF	: User Constraint File.
USB	: Universal Serial Bus.
VGA	: Video Graphics Array.
VHDL	: Very high speed integrated Hardware Description Language.
VLSI	: Very large Scale Integration.
$\mathbb{R}$	: L'ensemble des nombres réels.
$\mathbb{R}^n$	: L'espace vectoriel de dimension $n$ construit sur le corps des réels.
$\mathbb{R}^p$	: L'espace des paramètres.
$U \subseteq \mathbb{R}^n$	: L'espace d'états.
$x$	: Le vecteur d'états du système.
$u$	: Le vecteur d'entrée du système.
$y$	: Le vecteur de sortie du système.
$f(\cdot)$	: La fonction d'état.
$\frac{dx}{dt} = \dot{x}$	: La dérivée de la variable $x$ par rapport au temps.
$\xi$	: La différence entre l'état $x$ et le point fixe.
$x^*$	: Le point d'équilibre.
$A$	: La matrice jacobienne de $f(x)$ .
$\lambda_i$	: Valeurs propres de la matrice jacobienne ou exposant de Lyapunov.
$V(x)$	: Fonction de Lyapunov.
$\mu$	: Paramètre de bifurcation.
$\Omega$	: L'attracteur.
$J_f(x_p)$	: La matrice Jacobienne de $f$ au point $x_p$ .

$m(t)$	: Le message utile.
$s(t)$	: Le message chiffré transmis.
$m_r(t)$	: Le message récupéré.
$z(t)$	: Vecteur d'états du récepteur.
$e(t)$	: Erreur de synchronisation.
$r(t)$	: Signal reçu.
$\alpha$	: Paramètre du système chaotique Unifié de l'émetteur.
$\beta$	: Paramètre du système chaotique Unifié du récepteur.
$\hat{\alpha}$	: Paramètre estimé du système chaotique Unifié de l'émetteur.
$\hat{\beta}$	: Paramètre estimé du système chaotique Unifié du récepteur.
$T_{S_1}$	: Temps de synchronisation.
$T_{S_2}$	: Temps de resynchronisation.
$T_{c_1}$	: Temps du début d'émission-réception.
$T_{c_2}$	: Période d'émission-réception.
$dt$	: Le pas de temps de l'intégration.
$x_n$	: La variable d'état $x$ à l'instant $t$ .
$f$	: Le facteur de réduction d'amplitude.

# Table des matières

INTRODUCTION GÉNÉRALE .....	1
Chapitre 1 GÉNÉRALITÉS SUR LES SYSTÈMES DYNAMIQUES CHAOTIQUES.....	4
1.1 Introduction .....	4
1.2 Les systèmes dynamiques.....	4
1.2.1 Définition d'un système dynamique.....	4
1.2.2 Systèmes dynamiques autonome et non autonome.....	5
1.3 Le chaos .....	5
1.3.1 Définition du chaos .....	5
1.3.2 Historique sur le chaos.....	6
1.3.3 Caractéristiques du chaos.....	7
1.3.4 Route vers le chaos .....	9
1.3.5 Domaines d'application du chaos.....	10
1.4 Espace des phases.....	11
1.5 Point fixe ou point d'équilibre .....	11
1.5.1 Stabilité du point fixe au sens de Lyapunov .....	11
1.5.2 Méthode indirecte de Lyapunov.....	13
1.5.3 Méthode directe de Lyapunov.....	14
1.6 Bifurcation.....	15
1.6.1 Bifurcation Nœud-col.....	16
1.6.2 Bifurcation Trans-critique .....	17
1.6.3 Bifurcation Fourche (pitchfork).....	18
1.6.4 Bifurcation de Hopf.....	19
1.6.5 Bifurcation flip ou doublement de période .....	19
1.7 Attracteurs et Bassin d'attraction.....	20
1.7.1 Attracteurs réguliers .....	20
1.7.2 Attracteurs étranges .....	22
1.8 Section de Poincaré .....	22
1.9 Exposants de Lyapunov.....	24
1.9.1 Calcul des exposants de Lyapunov.....	24
1.9.2 Comportement des systèmes dynamiques en fonction des exposants de Lyapunov .....	26
1.10 Exemples de systèmes chaotiques .....	26

1.10.1	Système de Lorenz .....	26
1.10.2	Système de Rössler .....	29
1.11	Conclusion.....	31
Chapitre 2	ÉTUDE DU SYSTÈME CHAOTIQUE UNIFIÉ .....	32
2.1	Introduction .....	32
2.2	Le système chaotique de Chen .....	33
2.2.1	Présentation du système chaotique de Chen .....	33
2.2.2	Résultats de simulation du système chaotique de Chen .....	34
2.3	Le système chaotique de Lü.....	36
2.3.1	Présentation du système chaotique de Lü .....	36
2.3.2	Résultats de simulation du système chaotique de Lü .....	36
2.4	Le système chaotique Unifié.....	39
2.4.1	Présentation du système chaotique Unifié.....	39
2.4.2	Caractéristiques et avantages du système chaotique Unifié.....	40
2.4.3	Propriétés du système chaotique Unifié.....	41
2.4.4	Résultats de simulation du système chaotique Unifié.....	43
2.5	Conclusion.....	46
Chapitre 3	TRANSMISSION SÉCURISÉE PAR SYNCHRONISATION CHAOTIQUE ADAPTATIVE....	47
3.1	Introduction .....	47
3.2	Concept et classes de la synchronisation chaotique .....	48
3.2.1	Synchronisation par couplage unidirectionnel .....	49
3.2.2	Synchronisation par couplage bidirectionnel .....	50
3.3	Méthodes de synchronisation chaotique .....	51
3.3.1	Synchronisation identique ou par décomposition du système .....	51
3.3.2	Synchronisation généralisée .....	52
3.3.3	Synchronisation retardée.....	53
3.3.4	Synchronisation projective .....	53
3.3.5	Synchronisation impulsive .....	53
3.3.6	Synchronisation par boucle fermée.....	54
3.3.7	Synchronisation adaptative .....	54
3.4	Méthodes de transmission chaotique .....	55
3.4.1	Méthode par addition.....	55
3.4.2	Méthode par modulation paramétrique .....	56
3.4.3	Méthode par commutation chaotique .....	57

3.4.4	Méthode mixte ou combinée .....	58
3.4.5	Méthode par transmission à deux voies.....	58
3.5	Transmission sécurisée par synchronisation chaotique adaptative.....	60
3.5.1	Synchronisation adaptative de systèmes chaotiques Unifiés avec paramètres inconnus.....	60
3.5.2	Méthode de transmission utilisée .....	64
3.5.3	Implémentation du système de transmission sous Matlab Simulink.....	66
3.5.4	Simulation et résultats .....	71
3.5.5	Récapitulatif sur le système de transmission chaotique proposé.....	80
3.5.6	Effet du bruit sur la transmission.....	80
3.6	Conclusion.....	82
Chapitre 4	IMPLÉMENTATION FPGA DE LA TRANSMISSION SÉCURISÉE PAR CHAOS.....	83
4.1	Introduction .....	83
4.2	Présentation des circuit FPGA .....	84
4.2.1	Architectures des FPGA.....	84
4.2.2	Technologies des FPGA .....	85
4.2.3	Domaines d'application des FPGA .....	86
4.3	Processus d'implémentation .....	87
4.3.1	Flot de conception sur FPGA.....	87
4.3.2	Présentation du logiciel Xilinx ISE .....	88
4.3.3	System Generator et Co-Simulation .....	89
4.4	Réalisation expérimentale .....	91
4.4.1	Plate-forme de développement Xilinx ML501.....	93
4.4.2	Codec Ac'97 – AD1981B.....	95
4.5	Implémentation de l'émetteur chaotique sur FPGA .....	97
4.6	Implémentation de la transmission chaotique sur FPGA.....	101
4.7	Conclusion.....	106
	CONCLUSION GÉNÉRALE .....	107
	ANNEXE .....	109
	BIBLIOGRAPHIE .....	116

## Liste des figures

Figure 1.1 : État chaotique $x_1$ du système de Rössler .....	8
Figure 1.2 : Illustration de l'aspect de sensibilité aux conditions initiales .....	9
Figure 1.3 : Diagramme de bifurcation Noeud-Col.....	16
Figure 1.4 : Diagramme de bifurcation trans-critique .....	17
Figure 1.5 : Diagramme de bifurcation fourche (a) Sur critique - (b) Sous critique .....	18
Figure 1.6 : Bifurcation de Hopf.....	19
Figure 1.7 : Diagramme de bifurcation par doublement de période .....	20
Figure 1.8 : Les différents types d'attracteurs réguliers.....	21
Figure 1.9 : Exemples d'attracteurs étranges .....	22
Figure 1.10 : Section de Poincaré .....	23
Figure 1.11 : Réponse temporelle du système chaotique de Lorenz .....	27
Figure 1.12 : Évolution des exposants de Lyapunov du système chaotique de Lorenz .....	27
Figure 1.13 : Représentation du système chaotique de Lorenz dans l'espace des phases .....	28
Figure 1.14 : Diagramme de bifurcation pour le système chaotique de Lorenz .....	28
Figure 1.15 : Modèle Matlab Simulink du système chaotique de Lorenz .....	28
Figure 1.16 : Réponse temporelle du système chaotique de Rössler .....	29
Figure 1.17 : Évolution des exposants de Lyapunov du système chaotique de Rössler .....	29
Figure 1.18 : Représentation du système chaotique de Rössler dans l'espace des phases.....	30
Figure 1.19 : Diagramme de bifurcation pour le système chaotique de Rössler .....	30
Figure 1.20 : Modèle Matlab Simulink du système chaotique de Rössler .....	31
Figure 2.1 : Principe général d'un système de communications.....	32
Figure 2.2 : Réponse temporelle du système chaotique de Chen.....	34
Figure 2.3 : Représentation du système chaotique de Chen dans l'espace des phases ...	34
Figure 2.4 : Diagramme de bifurcation pour le système chaotique de Chen.....	35
Figure 2.5 : Évolution des exposants de Lyapunov du système chaotique de Chen.....	35

Figure 2.6 : Modèle Matlab Simulink du système chaotique de Chen .....	35
Figure 2.7 : Variable d'état $x_1$ en fonction du paramètre $c$ du système chaotique de Lü.....	37
Figure 2.8 : Évolution du système chaotique de Lü dans l'espace des phases .....	37
Figure 2.9 : Diagramme de bifurcation pour le système chaotique de Lü. ....	38
Figure 2.10 : Modèle Matlab Simulink du système chaotique de Lü .....	38
Figure 2.11 : Évolution du maximum des exposants de Lyapunov du système Unifié en fonction du paramètre $\alpha$ .....	41
Figure 2.12 : Variable d'état $x_1$ en fonction du paramètre $\alpha$ du système chaotique Unifié ...	43
Figure 2.13 : Diagramme de bifurcation pour le système chaotique Unifié .....	44
Figure 2.14 : Évolution du système chaotique Unifié dans l'espace des phases .....	44
Figure 2.15 : Modèle Matlab Simulink du système chaotique Unifié .....	45
Figure 3.1 : Couplage unidirectionnel.....	50
Figure 3.2 : Couplage bidirectionnel.....	50
Figure 3.3 : Synchronisation par décomposition du système chaotique .....	52
Figure 3.4 : Synchronisation impulsive .....	54
Figure 3.5 : Synchronisation par boucle fermée.....	54
Figure 3.6 : Synchronisation adaptative .....	55
Figure 3.7 : Méthode de cryptage chaotique par addition .....	55
Figure 3.8 : Méthode de cryptage chaotique par modulation paramétrique.....	56
Figure 3.9 : Méthode de cryptage chaotique par commutation.....	57
Figure 3.10 : Méthode de cryptage chaotique mixte.....	58
Figure 3.11 : Méthode de cryptage chaotique par transmission à deux voies .....	59
Figure 3.12 : Schéma de la synchronisation adaptative proposée.....	60
Figure 3.13 : Schéma de la transmission chaotique proposé.....	64
Figure 3.14 : Implémentation de l'intégrateur d'Euler.....	66
Figure 3.15 : Schéma global de l'implémentation de la synchronisation adaptative sous Matlab Simulink.....	67

Figure 3.16 : Implémentation du système chaotique Unifié de l'émetteur (système maître) sous Matlab Simulink.....	67
Figure 3.17 : Implémentation du système chaotique Unifié du récepteur (système esclave) sous Matlab Simulink.....	68
Figure 3.18 : Implémentation du contrôleur adaptatif sous Matlab Simulink.....	68
Figure 3.19 : Implémentation des estimateurs de paramètres $\alpha$ et $\beta$ des modèles de Cheng et proposé sous Matlab Simulink .....	69
Figure 3.20 : Schéma global de l'implémentation du système de transmission chaotique sous Matlab Simulink.....	69
Figure 3.21 : Implémentation des différents blocs d'acquisition sous Matlab Simulink ....	70
Figure 3.22 : Erreurs de synchronisation pour le modèle de Cheng et le modèle proposé.....	72
Figure 3.23 : Représentation des états de l'émetteur et du récepteur chaotiques du modèle proposé pour $dt = 10^{-4}$ .....	73
Figure 3.24 : Les paramètres estimés $\alpha$ et $\beta$ du modèle proposé pour $dt = 10^{-4}$ .....	73
Figure 3.25 : Résultats de transmission d'une sinusoïde à travers le système de transmission chaotique implémenté sous Matlab Simulink.....	74
Figure 3.26 : Résultats de transmission des images "CameraMan.jpg" et "Lenna.jpg" à travers le système de transmission chaotique implémenté sous Matlab Simulink.....	75
Figure 3.27 : Influence du paramètre $T_{c_2}$ sur la qualité de la transmission .....	76
Figure 3.28 : Les signaux utile et reconstitué résultants de la transmission de l'image "CameraMan.jpg" (agrandissement sur les colonnes [86 – 89] ).....	77
Figure 3.29 : Schéma global du système de transmission chaotique proposé avec correcteur d'erreur.....	78
Figure 3.30 : Influence du paramètre $T_{c_2}$ sur la qualité de la transmission en utilisant le correcteur d'erreur .....	79
Figure 3.31 : Schéma global du système de transmission chaotique proposé avec un Bruit Blanc Gaussien Additif associé au canal de transmission sous Matlab Simulink .....	80
Figure 3.32 : Effet d'un Bruit Blanc Gaussien Additif sur la qualité de transmission chaotique de la sinusoïde et de l'image "Lenna.jpg" en fonction du SNR en dB .....	81

Figure 4.1 : Architecture moderne des FPGA Xilinx.....	84
Figure 4.2 : Structure d'une cellule logique d'un FPGA XILINX.....	85
Figure 4.3 : Processus de programmation d'un FPGA .....	88
Figure 4.4 : Interface Project Navigator ISE 14.7 .....	89
Figure 4.5 : Flot de conception de Xilinx System Generator .....	91
Figure 4.6 : Architecture de l'implémentation de la transmission chaotique.....	92
Figure 4.7 : Réalisation expérimentale de l'implémentation .....	92
Figure 4.8 : La plateforme de développement XILINX ML501.....	94
Figure 4.9 : Diagramme de fonctionnement du codec stéréo Ac'97 – AD1981B.....	95
Figure 4.10 : La liaison AC entre le codec stéréo Ac'97 et son contrôleur.....	96
Figure 4.11 : Trames audio de la liaison AC entre le codec stéréo Ac'97 et son contrôleur....	96
Figure 4.12 : Implémentation du contrôleur du codec stéréo Ac'97-AD1981B.....	97
Figure 4.13 : Implémentation du bloc intégrateur .....	98
Figure 4.14 : Implémentation de l'émetteur chaotique.....	98
Figure 4.15 : Implémentation de l'émetteur chaotique avec le codec Ac'97 .....	99
Figure 4.16 : Signaux $x_1(t)$ et $x_3(t)$ (a) simulés - (b) Co-Simulés - (c) expérimentaux.....	99
Figure 4.17 : Plan de phase XZ (a) simulé - (b) Co-Simulé - (c) expérimental.....	100
Figure 4.18 : Comparaison entre : (a) porteuse chaotique $x_3(t)$ et signal transmis $s(t)$ - (b) message à transmettre $m(t)$ et signal transmis $s(t)$ .....	100
Figure 4.19 : Schéma global de l'implémentation de la transmission chaotique sous l'environnement System Generator - Matlab Simulink.....	101
Figure 4.20 : Implémentation du système chaotique Unifié esclave sous l'environnement System Generator - Matlab Simulink.....	102
Figure 4.21 : Implémentation du contrôleur chaotique adaptatif sous l'environnement System Generator - Matlab Simulink.....	102
Figure 4.22 : Implémentation du correcteur d'erreur sous l'environnement System Generator - Matlab Simulink .....	102
Figure 4.23 : Implémentation de l'estimateur du paramètre $\alpha$ sous l'environnement System Generator - Matlab Simulink .....	103

Figure 4.24 : Implémentation de l'estimateur du paramètre $\beta$ sous l'environnement System Generator - Matlab Simulink .....	103
Figure 4.25 : Schéma de la Co-Simulation matérielle sous l'environnement System Generator - Matlab Simulink .....	103
Figure 4.26 : Signaux $x_1(t)$ et $z_1(t)$ récupérés par (a) simulation - (b) Co-Simulation ....	104
Figure 4.27 : Signaux $x_2(t)$ et $z_2(t)$ récupérés par (a) simulation - (b) Co-Simulation ....	104
Figure 4.28 : Signaux $x_3(t)$ et $z_3(t)$ récupérés par (a) simulation - (b) Co-Simulation ....	105
Figure 4.29 : Erreurs de synchronisation $e_i = z_i - x_i$ ( $i = 1,2,3$ ) récupérés par (a) simulation - (b) Co-Simulation .....	105
Figure 4.30 : Message reconstitué (a) original - (b) simulation - (c) Co-Simulation.....	105

## Liste des tableaux

Tableau 1. 1 : Correspondance attracteur – section de Poincaré.....	24
---	----

# INTRODUCTION GÉNÉRALE

---

## Introduction

Le monde d'aujourd'hui présente des avancées technologiques spectaculaires dans tous les domaines constituant ainsi un système d'information mondial interconnecté via des réseaux publics (Internet, communication par satellite DVB, vidéo conférences, communications sans fil, messagerie électronique, ...) qui sont vulnérables et incertains, d'où le besoin et la nécessité d'un mécanisme de sécurisation (cryptage) des échanges.

À cet effet, plusieurs travaux de recherche dans ce domaine ont été réalisés et d'autres sont en cours de développement par des chercheurs académiques, dont le résultat est la conception d'algorithmes et des techniques de cryptages qui sont basées sur la théorie des nombres et en particulier sur la décomposition d'un entier en éléments simples. Parmi ces algorithmes les plus répondus, nous citons : DES (Data Encryption Standard) et RSA (dont le nom est formé des initiales de ses inventeurs : R. Rivest, A. Shamir et L. Adleman). En effet, bien que la structure de ces algorithmes soit bien connue par tout le monde, leur fiabilité est assurée grâce à la limitation des calculateurs en terme de puissance de calcul. Néanmoins, avec la révolution de l'informatique, ces algorithmes s'avèrent peu sécurisés, d'où la nécessité de trouver une autre alternative de cryptage.

Depuis quelques années, la théorie du chaos découverte en 1963 par Mr Edward N. Lorenz [1, 2], a intéressé plusieurs chercheurs dans le domaine de la cryptographie. En effet, grâce aux propriétés des systèmes chaotiques (aspect non linéaire et aléatoire, spectre continu et extrême sensibilité aux conditions initiales), ils constituent de bons outils pour la sécurisation des échanges de données. En 1990, Pecora et Carroll [3, 4] ont réussi à reproduire de manière exacte un signal électrique en synchronisant deux signaux chaotiques. Cette découverte de la synchronisation des signaux chaotiques a permis d'utiliser le chaos comme moyen de modulation de l'information. Depuis, plusieurs

schémas sont proposés afin d'appliquer les systèmes chaotiques dans le domaine de la cryptographie.

## **Objectif du projet**

Ce travail consiste à une implémentation FPGA d'un système de transmission chaotique, basé sur une synchronisation adaptative pour deux différents systèmes chaotiques Unifié, dont les paramètres sont inconnus. Ce système a été inspiré du système de transmission chaotique proposé par C.J. Cheng [5]. L'émetteur chaotique est construit à base d'un système chaotique Unifié tandis que le récepteur est constitué d'un système chaotique Unifié différent de l'émetteur piloté par un contrôleur adaptatif, permettant ainsi de reconstruire d'une manière le plus fidèle les états de l'émetteur et de récupérer le message transmis. Les différentes étapes de conception de ce système sont développées en détail ainsi que son implémentation sur circuit FPGA.

Pour l'implémentation de ce système sur circuit FPGA, nous avons utilisé l'outil System Generator de Xilinx sous l'environnement Matlab Simulink. Une Co-Simulation matérielle – Matlab Simulink s'est effectuée, afin de nous permettre de comparer les signaux obtenus sous simulation et ceux provenant de la carte FPGA et de valider ainsi le bon fonctionnement du système de communication.

## **Structure du manuscrit**

Après une introduction générale sur notre travail, le mémoire se compose en quatre chapitres comme suit :

**Chapitre 1 :** Ce chapitre présente des généralités et des définitions relatives aux systèmes chaotiques ainsi que leurs propriétés fondamentales et leurs différents domaines d'application. Il présente aussi les résultats de simulation de quelques exemples canoniques des systèmes chaotiques.

**Chapitre 2 :** Ce chapitre sera consacré à l'étude du système chaotique Unifié constituant notre émetteur, ses caractéristiques et ses propriétés ainsi que l'analyse des systèmes chaotiques qui le constituent (Lorenz, Chen et Lü). Les résultats de simulation de ces systèmes sont également présentés.

**Chapitre 3 :** Dans ce chapitre, nous présentons dans un premier temps, les différentes méthodes de synchronisation chaotique ainsi que les systèmes de transmission chaotiques (cryptage). Par la suite, nous décrivons en détail les différentes étapes de conception de notre système de transmission chaotique inspiré du système proposé par Cheng [5], tout en montrant notre contribution dans l'amélioration du système via des explications et des analyses des résultats de simulation de la synchronisation chaotique ainsi que de la transmission chaotique. Une étude sur l'influence d'un Bruit Blanc Gaussien Additif sur le comportement de notre système est également développée.

**Chapitre 4 :** Ce chapitre présente un aperçu sur la technologie FPGA avec laquelle nous allons implémenter notre système de transmission chaotique, tout en utilisant l'outil System Generator sous l'environnement Matlab Simulink qui nous permettra par la suite de générer le projet ISE adéquat pour notre implémentation. Afin de valider le bon fonctionnement de notre système, une comparaison entre les résultats de la simulation et ceux de la Co-Simulation matérielle – Matlab Simulink sera effectuée.

Ce mémoire sera finalisé par une conclusion générale reprenant les principaux points abordés dans ce manuscrit et en exposant certaines perspectives d'approfondissement, d'élargissement et d'amélioration de notre travail.

## 1.1 Introduction

Depuis l'immense découverte de Mr Edward N. Lorenz en 1963 [1] avec son fameux papillon [2] et la naissance de la théorie du Chaos avec, les recherches dans ce domaine ont connus une croissance sans cesse. À nos jours l'application de la théorie du chaos a envahi la plupart des sciences, c'est le cas de la physique, la biologie (la démographie, les épidémies, etc.), la chimie et l'économie, etc. Notre étude se focalise sur l'application du chaos dans le domaine des transmissions sécurisées de l'information. À cet effet, ce chapitre sera consacré à l'étude des systèmes dynamiques chaotiques.

Après un rappel et des généralités sur les systèmes dynamiques, la théorie du chaos sera introduite avec un bref historique, tout en étalant cette étude sur les propriétés fondamentales des systèmes chaotiques : espace des phases, différents points d'équilibres et leurs stabilités en passant par les méthodes de stabilité de Lyapunov, bifurcation, section de Poincaré, les attracteurs, les exposants de Lyapunov et leurs influences sur le comportement chaotique des systèmes dynamiques. Nous terminerons ce chapitre par une présentation des résultats de simulation de quelques exemples canoniques des systèmes chaotiques.

## 1.2 Les systèmes dynamiques

### 1.2.1 Définition d'un système dynamique

Un système dynamique est décrit par une fonction mathématique qui présente deux types de variables : dynamiques et statiques ; les variables dynamiques sont les quantités fondamentales qui changent avec le temps ; les variables statiques encore appelées paramètres du système sont fixes [6]. Cette fonction est définie par l'équation différentielle suivante :

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

où :  $x \in U \subseteq \mathbb{R}^n$  est un vecteur de dimension  $n$  représentant l'état du système et  $u \in U \subseteq \mathbb{R}^p$  représente le vecteur paramètre du système.  $\mathbb{R}^n$  est appelé l'espace des phases et  $\mathbb{R}^p$  l'espace des paramètres. Cette équation est généralement associée à une autre équation désignant le vecteur de sortie :

$$y = h(t, x, u) \quad (1.2)$$

### 1.2.2 Systèmes dynamiques autonome et non autonome

Soit le système dynamique décrit par l'équation (1.1). Ce système est dit autonome lorsqu'il ne dépend pas explicitement du temps. Dans le cas contraire, il est dit non autonome. L'écriture du système dynamique autonome se réduit à l'équation suivante :

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

Un système autonome est indépendant du temps initial, alors qu'un système non autonome ne l'est pas. Dans un système autonome, tout instant peut être considéré comme instant initial, et tout état  $x(t)$  du système peut être considéré comme un état initial [7].

## 1.3 Le chaos

### 1.3.1 Définition du chaos

Dans le langage usuel, le mot chaos renvoie à la notion de désordre total. Alors que dans le domaine des sciences, il en est tout autre ; il se rattache plutôt à une notion d'imprévisibilité, d'impossibilité de prévoir l'évolution d'un système à long terme du fait que l'état final dépend de manière si sensible de l'état initial [8].

On dit alors qu'un système dynamique est chaotique, lorsque son évolution dans le temps est très sensible aux conditions initiales. Ainsi, deux trajectoires générées à partir de conditions initiales très voisines, vont diverger très rapidement l'une par rapport à l'autre [9]. Ces conditions initiales ne sont pas déterminées ou modélisées par des systèmes

d'équations linéaires, ni par les lois de la mécanique classique ; pourtant, ils ne sont pas nécessairement aléatoires, relevant du seul calcul des probabilités.

### **1.3.2 Historique sur le chaos**

Le 17<sup>ième</sup> siècle a été marqué par des inventions scientifiques révolutionnaires de l'humanité, consistant essentiellement à l'invention de la mécanique classique, introduite par Newton. En introduisant ses lois et son calcul différentiel, Newton a pu déterminer la trajectoire d'un corps à partir de l'étude de son mouvement. Sur ce lancé, il était très facile de prévoir les différents états d'un système donné tel que le passage des comètes, les heures des marées, etc. À cet effet, les mathématiciens et philosophes de cette époque pensaient que l'Univers était réglé comme une horloge, tout mouvement, que ce soit celui d'une planète ou d'une molécule, étant régi par les mêmes lois était donc prédictible. Ce n'était qu'une question de calculs. Le hasard n'existait pas, et tout mouvement semblant relever du hasard n'était qu'un mouvement que l'on n'était pas encore capable de calculer. Le monde étant régi par des lois, l'avenir devenait prévisible.

Afin d'établir le « modèle » d'un phénomène naturel, il y avait trois étapes majeures à suivre : observer le phénomène, le traduire en équations et enfin résoudre ces équations. C'est au niveau de la troisième étape, celle de la résolution des équations, que les difficultés sont apparues. Certaines de ces équations étaient difficiles à résoudre et d'autres ne l'étaient pas du tout. C'était le cas de plusieurs phénomènes dans divers domaines ; on cite par exemple [10] le problème des trois corps célestes, les prévisions météorologiques, etc.

Edward N. Lorenz (mathématicien Américain) quant à lui, en 1961, s'était branché dans le domaine des prévisions météorologiques. Il a ainsi obtenu un système très complexe en équations différentielles, difficile à résoudre ou bien impossible à résoudre à leur époque. Alors il a choisi de comprendre le phénomène et pas le résoudre, pour cela il a simplifié vers un système ultra simplifié (équation (1.21)), avec seulement trois équations différentielles à trois variables. Après, Lorenz a eu un coup de génie, il a utilisé un ordinateur pour la résolution de son système. C'est par pur 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.

En 1972 Lorenz a donné une conférence devant le grand public du congrès de l'Association américaine pour l'avancement des sciences, intitulée : « Prédicibilité : le battement des ailes d'un papillon au Brésil peut-il déclencher une tornade au Texas ? » [2, 10]. Cette conférence a rendu Lorenz célèbre avec sa théorie et son fameux papillon, comme il a démontré que l'avenir n'est pas prédictible, vu la sensibilité de quelques systèmes aux conditions initiales, ces derniers ne pourront jamais être exactes. Le terme "Chaos " de cette théorie a été proposé par le mathématicien Yorke, deux ans plus tard, en 1975.

Edward Lorenz est reconnu comme le père ou l'inventeur de la théorie du Chaos. 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 19<sup>ième</sup> 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 célestes. Aussi, on trouve les travaux du mathématicien russe Alexandre Lyapunov sur la stabilité du mouvement, où il a 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. Ces travaux étaient très précieux pour étudier certains aspects de la théorie du chaos [11].

Les travaux des prédécesseurs de Lorenz ont donc été très importants pour la compréhension du chaos, 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.

### **1.3.3 Caractéristiques du chaos**

Malgré l'aspect aléatoire des phénomènes chaotiques (figure 1.1), ils obéissent au contraire à des lois déterministes, parfois assez simple dans leur représentation mathématique. Ces phénomènes se caractérisent par des propriétés génériques fondamentales, en plus de l'aspect aléatoire. On peut ainsi citer [9, 11, 12] :

### a. 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, comme le montre la figure 1.1 :

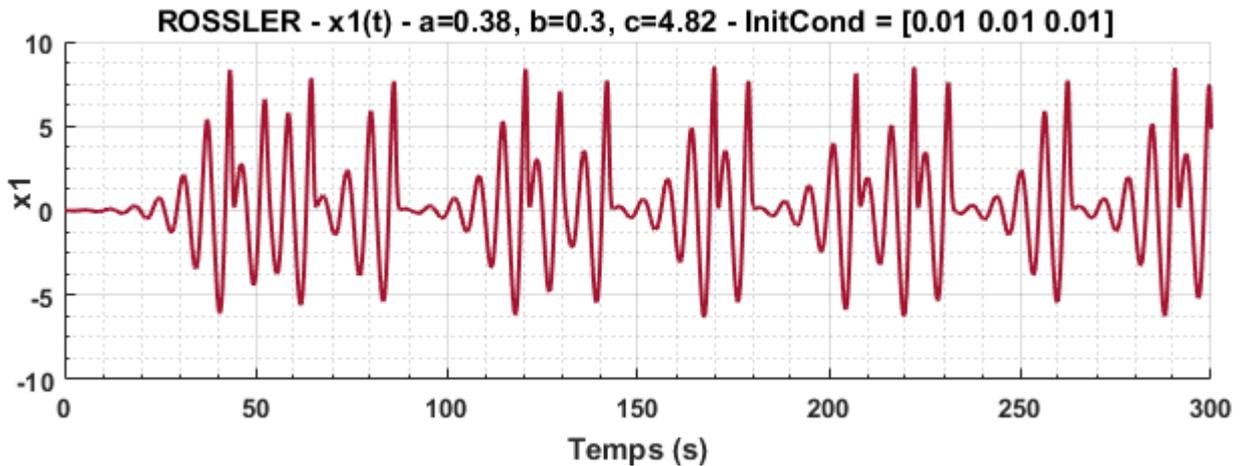


Figure 1.1 : État chaotique  $x_1(t)$  du système de Rössler

### b. Sensibilité aux conditions initiales

Pour un système dynamique chaotique, la moindre erreur ou imprécision sur l'état initial  $x_0$  va changer radicalement l'état du système à long terme (figure 1.2), et de ce fait il est impossible de prédire exactement l'état du système à long terme. C'est l'une des propriétés essentielles des systèmes chaotiques.

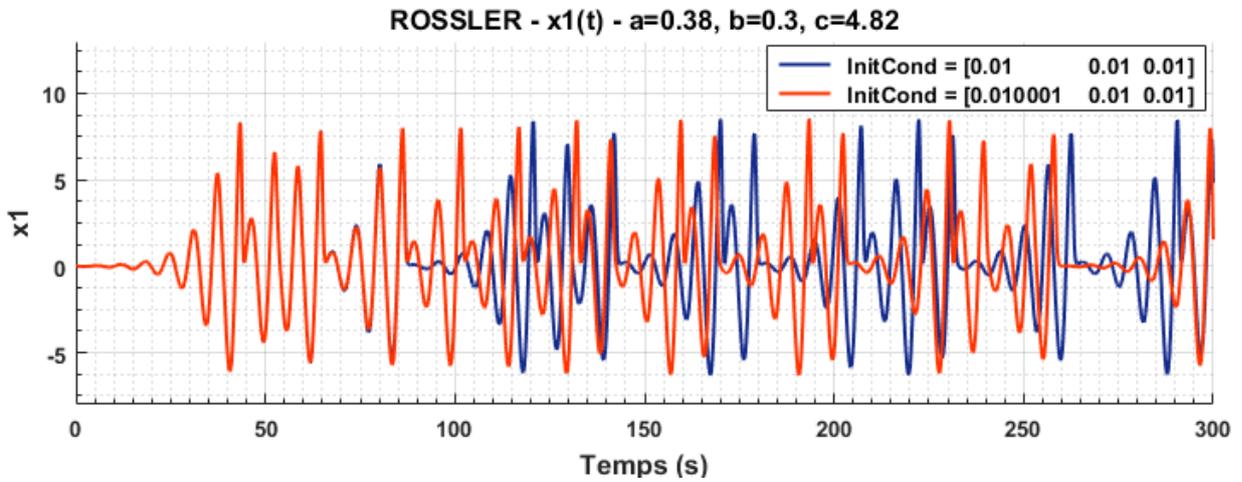
D'un point de vue mathématique, on dit que  $f$  montre une dépendance sensible aux conditions initiales lorsque :

$$\exists \delta > 0, \forall x \in D, \forall \varepsilon > 0, \exists (y, p) \in D: \begin{cases} \|x - y\| < \varepsilon \\ \|f^p(x) - f^p(y)\| > \delta \end{cases} \quad (1.4)$$

### c. L'attracteur étrange

Dans l'espace des phases (section 1.4), le chaos donne lieu à des trajectoires, appelées attracteur chaotique ou attracteur étrange. Géométriquement, un tel attracteur peut être décrit comme le résultat d'une opération d'étirement et de repliement d'un cycle de l'espace des phases [10], répétée un nombre infini de fois. La longueur de l'attracteur est

infinie bien qu'elle soit contenue dans un espace fini. Cette notion sera bien développée dans la section 1.7.2.



**Figure 1.2 :** Illustration de l'aspect de sensibilité aux conditions initiales (Exemple : système chaotique de Rössler)

#### **d. Le spectre de puissance**

Une façon simple de caractériser le chaos consiste à calculer le spectre de Fourier de l'évolution temporelle d'une des variables du système. L'existence de spectres larges parfois confondu avec le bruit, est une caractéristique essentielle des mouvements chaotiques d'un système.

### **1.3.4 Route vers le chaos**

Un système dynamique chaotique est un système qui dépend d'un ou de plusieurs paramètres dit de contrôle, agissant sur les caractéristiques de la fonction de transition. Selon la valeur de ce paramètre, les mêmes conditions initiales mènent à des trajectoires correspondant à des régimes dynamiques qualitativement différents. La modification continue du paramètre de contrôle (les résultats de cette variation de paramètre de contrôle s'appellent bifurcation que l'on va développer dans la section 1.6) conduit à une complexification progressive du régime dynamique développé par le système.

Il existe plusieurs scénarios qui décrivent le passage du point fixe (section 1.5) au chaos ; on peut citer trois types de transitions vers le chaos [9] :

### ***a. L'intermittence vers le chaos***

Un mouvement périodique stable est entrecoupé par des bouffées de turbulence. Lorsqu'on augmente le paramètre de contrôle, les bouffées de turbulence deviennent de plus en plus fréquentes, et finalement, la turbulence domine.

### ***b. Le dédoublement de période***

C'est une succession de bifurcation de fourches (section 1.6.3). À mesure que la contrainte augmente, la période d'un système forcé est multipliée par deux, puis par quatre, puis par huit, etc. ; ces doublements de période sont de plus en plus rapprochés ; lorsque la période est infinie, le système devient chaotique. La turbulence dans les fluides peut apparaître suivant ce scénario.

### ***c. La quasi-périodicité***

Elle intervient quand un deuxième système perturbe un système initialement périodique. Si le rapport des périodes des deux systèmes en présence n'est pas rationnel, alors le système est quasi-périodique et peut évoluer vers le chaos.

## **1.3.5 Domaines d'application du chaos**

Comme on a déjà mentionné que le chaos a envahi la plupart des sciences, il a trouvé aussi son application dans divers domaines, parmi lesquels on peut citer [11] :

### ***a. Ordinateurs***

Commutation des paquets dans des réseaux informatiques. Cryptage. Contrôle du chaos dans les systèmes robotiques.

### ***b. Communications***

Compression et stockage d'images. Conception et management des réseaux d'ordinateurs.

### ***c. Médecine et Biologie***

Cardiologie : analyse du rythme du cœur, prédiction et contrôle d'activité irrégulière du cœur.

Neurologie : analyse de l'activité cérébrale sur un électroencéphalogramme (EEG).

#### **d. Management et finance**

Prévisions économiques, analyse financière, et prévision du marché.

### **1.4 Espace des phases**

Dès que la dimension  $n$  du système dépasse l'unité, il devient assez difficile de représenter "mentalement" comment le système évolue. L'outil de base pour y palier est l'espace de phase.

Dans un système dynamique de dimension  $n$ , l'espace formé par les composantes du vecteur d'état  $x_1, x_2, x_3, \dots, x_n$  est appelé espace des phases ou espace d'états. Ainsi l'évolution du système au cours du temps se traduit par un déplacement du point représentatif dans l'espace des phases, tout en parcourant un chemin appelé "trajectoire".

### **1.5 Point fixe ou point d'équilibre**

On appelle point fixe (point stationnaire ou point d'équilibre ou point critique) d'un système dynamique représenté par l'équation (1.1), tout point  $x^*$  de l'espace de phase satisfaisant l'équation suivante :

$$f(x^*) = 0 \quad (1.5)$$

Par un changement de variables  $\xi = x - x^*$ , on peut ramener le point fixe  $x^*$  à l'origine. Les points fixes jouent un rôle très important dans les applications car ils permettent de caractériser les trajectoires voisines.

#### **1.5.1 Stabilité du point fixe au sens de Lyapunov [7, 13]**

Soit le système dynamique autonome :  $\dot{x} = f(x)$ , où  $f: D \longrightarrow \mathbb{R}_n$  est une projection localement Lipschitz de  $D \subset \mathbb{R}^n$  dans  $\mathbb{R}^n$ . On suppose que le point  $x^*$  est un point d'équilibre de ce système :  $f(x^*) = 0$ .

On dit que le point  $x^*$  est :

**a. Stable (au sens de Lyapunov)**

Si  $\forall \varepsilon > 0, \exists \alpha > 0$  tel que :

$$\|x(0) - x^*\| < \alpha \Rightarrow \|x(t) - x^*\| < \varepsilon$$

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.

Si l'on suppose que le point d'équilibre est ramené à l'origine par un changement de coordonnées  $\xi = x - x^*$ , la stabilité du point d'équilibre  $x^* = 0$  sera étudiée de la façon suivante :

Si  $\forall \varepsilon > 0, \exists \alpha > 0$  tels que :

$$\|x(0)\| < \alpha \Rightarrow \|x(t)\| < \varepsilon$$

**N.B :** Le point d'équilibre  $x^*$  est dit "instable" s'il n'est pas stable au sens de Lyapunov.

**b. Asymptotiquement stable**

Si  $\forall \alpha > 0$  alors :

$$\|x(0) - x^*\| < \alpha \Rightarrow \lim_{t \rightarrow \infty} x(t) = x^*$$

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  $x^*$ , tende vers  $x^*$  lorsque  $t \rightarrow \infty$ .

**c. Exponentiellement stable**

S'il est asymptotiquement stable et  $\exists \varepsilon > 0$  et  $\lambda > 0$  tel que :

$$\|x(0) - x^*\| < \alpha \Rightarrow \|x(t) - x^*\| < \varepsilon \|x(0) - x^*\| e^{-\lambda t}, \forall t \geq 0$$

$\lambda$  : est appelé taux de convergence.

**d. Globalement stable**

Si la condition de stabilité asymptotique (resp. Exponentielle) est vérifiée dans tout  $\mathbb{R}^n$  (c'est-à-dire  $\forall \varepsilon$ ), le point d'équilibre est globalement asymptotiquement (resp. Exponentiellement) stable.

Les définitions précédentes sont locales car elles ne concernent que les orbites voisines d'un point d'équilibre. Pour cela, nous présentons les deux méthodes de Lyapunov permettant d'étudier la stabilité d'un système dynamique.

### 1.5.2 Méthode indirecte de Lyapunov

Cette première méthode appelée aussi stabilité du système linéarisé, est similaire à l'étude classique pour la méthode dite du plan de phase [13]. Elle consiste à qualifier la stabilité d'un point d'équilibre d'un système non linéaire à travers les propriétés du système linéarisé autour de ce point d'équilibre.

Par un changement de coordonnées, le point fixe de l'équation (1.1) se ramène à l'origine ( $f(0) = 0$ ) et le développement de  $f$  en série de Taylor autour du point  $x = 0$  donne :

$$f(x) = Df(0)x + \frac{1}{2!}D^2f(0)(x, x) + \frac{1}{3!}D^3f(0)(x, x, x) + \dots \quad (1.6)$$

où  $Df(0)$  est la matrice Jacobienne de  $f(x)$  autour du point d'équilibre  $x^* = 0$ .

Pour étudier la stabilité autour d'un point d'équilibre, la méthode indirecte de Lyapunov consiste à étudier le système linéaire [7, 9] :

$$\dot{x} = Ax \quad (1.7)$$

avec :

$$A = Df(0) = \left( \begin{array}{ccc} \frac{df_1}{dx_1} & \dots & \frac{df_1}{dx_n} \\ \vdots & \ddots & \vdots \\ \frac{df_n}{dx_1} & \dots & \frac{df_n}{dx_n} \end{array} \right)_{x=0} \quad (1.8)$$

$A$  est la matrice jacobienne de  $f(x)$  et son déterminant est le jacobien.

Dans le cas où la matrice jacobienne  $Df(0)$  possède  $n$  valeurs propres  $\lambda_i, i = 1, 2, \dots, n$ . distinctes, la solution de l'équation (1.7) est :

$$x(t) = \sum_{i=1}^n c_i a^{(i)} e^{\lambda_i t} \quad (1.9)$$

où  $a^{(i)}$  est le vecteur propre correspondant à la valeur propre  $\lambda_i$  et les  $c_i, i = 1, 2, \dots, n$ , sont des constantes (déterminées par les conditions initiales).

Selon les valeurs propres  $\lambda_i, i = 1, 2, \dots, n$ , on peut déduire sur la stabilité du point d'équilibre  $x^*$  ce qui suit [9] :

- a-** Si toutes les valeurs propres  $\lambda_i$  ont leur partie réelle négative, le point fixe est asymptotiquement stable.
- b-** 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).
- c-** Si l'une des valeurs propres a sa partie réelle positive, le point fixe est instable.
- d-** Si  $Df(0)$  n'a pas de valeur propre 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 selle.
- f-** Si toutes les valeurs propres de  $Df(0)$  sont réelles et de même signe, le point fixe est un nœud. Un nœud stable est un puits. Un nœud instable est une source.

### 1.5.3 Méthode directe de Lyapunov

L'idée de base de cette méthode s'appuie sur le principe de dissipation de l'énergie. Si pour un système mécanique ou électrique, l'énergie totale se dissipe continûment alors le système en question va s'arrêter [13] (ce que signifie que le système tend vers un point d'équilibre). De ce fait, l'idée de Lyapunov consiste à examiner la variation d'une fonction scalaire pour étudier la stabilité d'un système donné. La méthode directe de Lyapunov permet d'étudier la stabilité d'un système sans avoir besoin de calculer les solutions de celui-ci.

Soit le système suivant :

$$\begin{cases} \dot{x} = f(x) \\ x \in \mathbb{R}^n, f(x^*) = 0 \end{cases} \quad (1.10)$$

S'il existe une fonction scalaire  $V: U \rightarrow \mathbb{R}$ , continue sur un voisinage de  $U$  de  $x^*$  et différentiable telle que :

- 1-  $V(x^*) = 0$  et  $V(x) > 0$  et  $x \neq 0$
- 2-  $\dot{V}(x) = \sum_{j=1}^n \frac{\partial V}{\partial x_j} \dot{x}_j = \sum_{j=1}^n \frac{\partial V}{\partial x_j} f_j(x) \leq 0, \forall x \in U$

Alors  $x^*$  est un point d'équilibre stable pour le système (1.10).

- 3- Si de plus la fonction  $V$  est telle que :

$$\dot{V}(x) < 0, \forall x \in U \setminus \{x^*\}$$

Alors  $x^*$  est un point asymptotiquement stable.

La difficulté majeure dans cette méthode est de trouver une fonction de Lyapunov  $V(x)$  adéquate, car il n'existe pas de moyen systématique pour en trouver une. Cependant, dans les systèmes mécaniques et électriques, l'énergie est souvent un bon candidat.

## 1.6 Bifurcation

Soit un système dynamique, dépendant d'un paramètre  $\mu$ , de la forme :

$$\dot{x} = f(x(t), \mu) \tag{1.11}$$

La génération d'un système chaotique n'est pas immédiate. En effet, le système n'évolue pas d'un état inexistant à un état chaotique sans passer par des transitions. Considérons un système dynamique de la forme (1.11) dépendant d'un paramètre de contrôle  $\mu$ . En variant ce paramètre, le système peut passer d'un état stationnaire à un état périodique, puis au-delà d'un certain seuil, suivre un scénario de transition et devenir chaotique [14]. Ce phénomène s'appelle bifurcation.

La bifurcation est en effet associée au changement topologique de la trajectoire d'un système dynamique lorsqu'un ou plusieurs de ses paramètres varient. Le paramètre responsable de ce changement s'appelle "paramètre de bifurcation", et la valeur de ce paramètre pour laquelle on peut observer une bifurcation s'appelle "point de bifurcation". La représentation d'une propriété caractéristique d'une solution, en fonction du paramètre de bifurcation constitue un "diagramme de bifurcation" [7].

Il est possible de tracer un diagramme de bifurcation pour tout système non linéaire. Ainsi, en faisant varier un des paramètres du système, on peut étudier son comportement. Il existe plusieurs types de bifurcations [9] que l'on va décrire par la suite.

### 1.6.1 Bifurcation Nœud-col

C'est la bifurcation la plus simple. Lorsque le paramètre  $\mu$  franchit le 0, un point d'équilibre stable (nœud) et un point d'équilibre instable (col) apparaissent simultanément ; elle est souvent représentée par l'équation suivante :

$$\frac{dx}{dt} = \mu - x^2 \quad (1.12)$$

qui s'appelle équation générique de la bifurcation nœud-col. On a alors :

$$f(x, \mu) = \mu - x^2$$

- ✓ Si  $\mu < 0$ , l'équation  $f(x, \mu) = 0$  n'admet pas de solution : il n'y a donc pas de points fixes.
- ✓ Si  $\mu > 0$ , on a :  $\mu - x^2 = 0 \Rightarrow x = \pm\sqrt{\mu}$ . Par conséquent l'équation (1.12) admet deux points fixes (figure 1.3), dont on déterminera leurs stabilités.

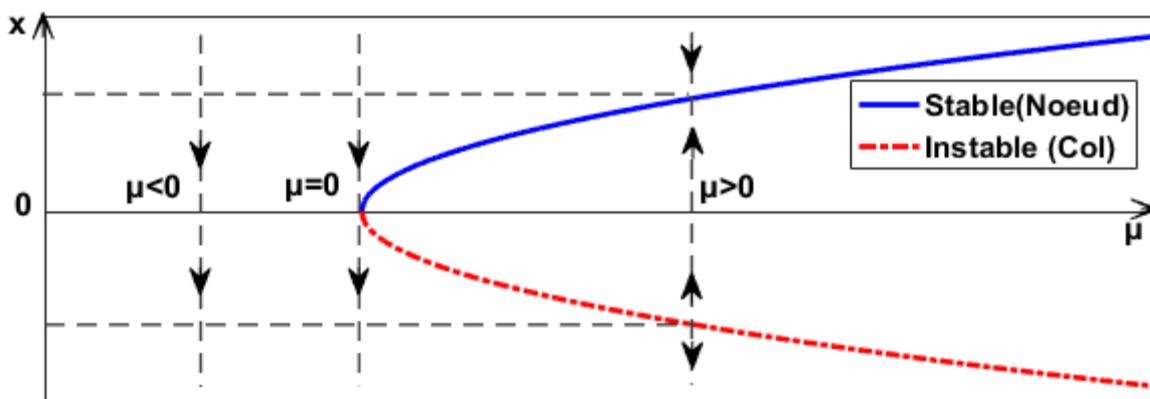
$$\frac{df(x, \mu)}{dx} = -2x \Rightarrow \left. \frac{df(x, \mu)}{dx} \right|_{x=\sqrt{\mu}} = -2\sqrt{\mu} < 0 \Rightarrow$$

*Le point fixe  $x = \sqrt{\mu}$  est stable.*

$$\frac{df(x, \mu)}{dx} = -2x \Rightarrow \left. \frac{df(x, \mu)}{dx} \right|_{x=-\sqrt{\mu}} = 2\sqrt{\mu} > 0 \Rightarrow$$

*Le point fixe  $x = -\sqrt{\mu}$  est instable.*

- ✓ Si  $\mu = 0$ , le seul point fixe est  $x = 0$ .



**Figure 1.3** : Diagramme de bifurcation Nœud-Col

## 1.6.2 Bifurcation Trans-critique

Elle est caractérisée par un échange de stabilité entre les points fixes (les points stables deviennent instables et vice versa) lorsque le paramètre  $\mu$  franchit le 0. Elle est souvent représentée par l'équation suivante :

$$\frac{dx}{dt} = \mu x - x^2 \quad (1.13)$$

qui s'appelle équation générique de la bifurcation trans-critique. On a alors :

$$f(x, \mu) = \mu x - x^2 = 0 \Rightarrow \begin{cases} x = 0 \\ \text{ou} \\ x = \mu \end{cases}$$

On a donc deux points fixes (figure 1.4), dont on déterminera leurs stabilités.

$$\left. \frac{df(x, \mu)}{dx} \right|_{x=0} = \mu \quad \text{et} \quad \left. \frac{df(x, \mu)}{dx} \right|_{x=\mu} = -\mu$$

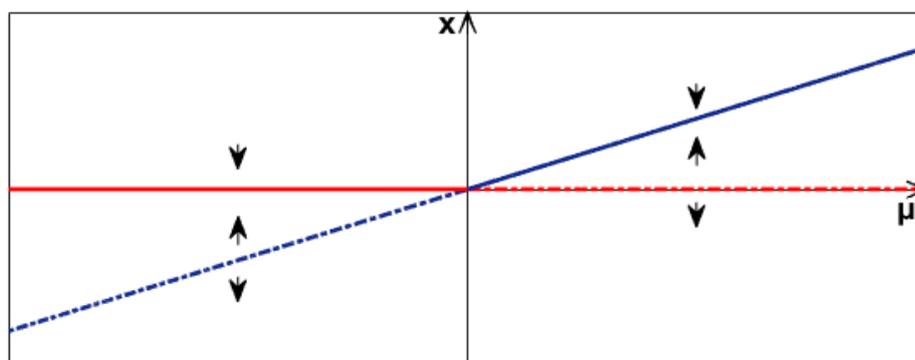
- ✓ Si  $\mu < 0$ , le point fixe  $x = 0$  est stable, tandis que le point fixe  $x = \mu$  est instable.
- ✓ Si  $\mu > 0$ , le point fixe  $x = 0$  est instable, tandis que le point fixe  $x = \mu$  est stable.

On remarque un échange de stabilité en  $\mu = 0$ .

- ✓ Si  $\mu = 0$ , le seul point fixe est  $x = 0$ . En intégrant l'équation (1.13), on obtient :

$$x(t) = \frac{1}{t + \frac{1}{x_0}}$$

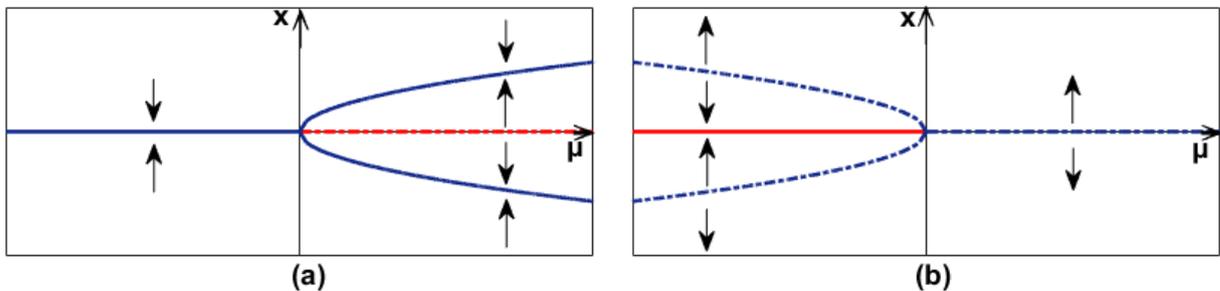
D'où le point  $x = 0$  est semi-stable (stable si  $x_0 > 0$  et instable si  $x_0 < 0$  ).



**Figure 1.4 :** Diagramme de bifurcation trans-critique

### 1.6.3 Bifurcation Fourche (pitchfork)

Au point de la bifurcation fourche (figure 1.5), la stabilité du point fixe change au profit de la naissance d'une paire de points fixes.



**Figure 1.5 :** Diagramme de bifurcation fourche (a) Sur critique - (b) Sous critique

L'équation générique d'une bifurcation fourche (sur-critique) est :

$$\frac{dx}{dt} = \mu x - x^3 \quad (1.14)$$

L'équation générique d'une bifurcation fourche (sous-critique) est :

$$\frac{dx}{dt} = \mu x + x^3 \quad (1.15)$$

En considérant le cas d'une bifurcation sur-critique, on a

$$f(x, \mu) = \mu x - x^3 = 0 \Rightarrow \begin{cases} x = 0 \\ \text{ou} \\ x = \pm\sqrt{\mu} \end{cases}$$

✓ Si  $\mu < 0$ , on a un seul point fixe  $x = 0$ .

✓ Si  $\mu > 0$ , on a trois points fixes :  $\begin{cases} x = 0 \\ x = \sqrt{\mu} \\ x = -\sqrt{\mu} \end{cases}$

Déterminons la stabilité de ces point fixes.

$$\left. \frac{df(x, \mu)}{dx} \right|_{x=0} = \mu \quad \text{et} \quad \left. \frac{df(x, \mu)}{dx} \right|_{x=\pm\sqrt{\mu}} = -2\mu$$

✓ Si  $\mu < 0$ , le point fixe  $x = 0$  est stable.

- ✓ Si  $\mu > 0$ , le point fixe  $x = 0$  est instable, tandis les deux points fixes  $x = \pm\sqrt{\mu}$  sont stables.

### 1.6.4 Bifurcation de Hopf

La bifurcation de Hopf représentée sur la figure 1.6 apparait lorsque le paramètre de contrôle  $\mu$  prend une valeur critique  $\mu_0$  pour laquelle la matrice jacobienne du système possède une paire de valeurs propres complexes conjuguées qui traversent l'axe imaginaire et le type de stabilité de l'équilibre existant change avec l'apparition d'un cycle limite.

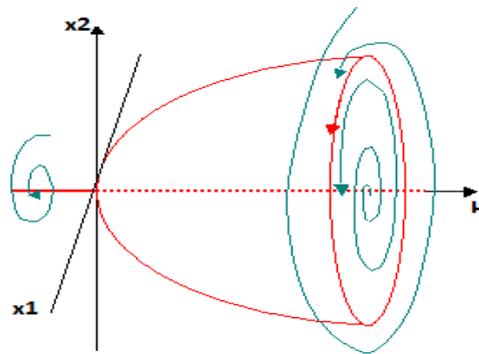
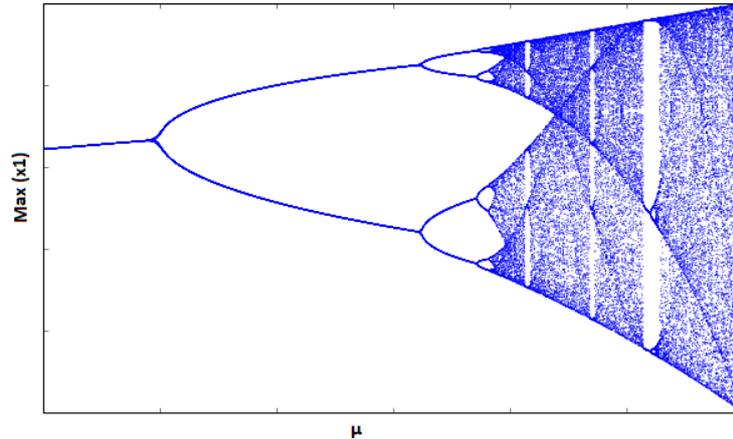


Figure 1.6 : Bifurcation de Hopf

### 1.6.5 Bifurcation flip ou doublement de période

Considérons un système dynamique de la forme (1.11), qui possède un point fixe stable pour  $\mu < \mu_0$ . Si on augmente  $\mu$  au-delà de la valeur  $\mu_0$ , le point d'équilibre se déstabilise et une bifurcation se produit et donne lieu à un cycle d'ordre 2 stable. Puis, si  $\mu$  continue d'augmenter, le cycle d'ordre 2 se déstabilise et chacun des deux points du cycle bifurque à son tour. Cette nouvelle bifurcation donne naissance à un cycle d'ordre 4 stable. Si  $\mu$  augmente toujours, des bifurcations continuent d'apparaître en doublant la période du cycle à chaque fois, d'où le nom de cette bifurcation, conduisant ainsi à une suite infinie de bifurcations et éventuellement au chaos. Un exemple illustratif est donné dans la figure 1.7.



**Figure 1.7 :** Diagramme de bifurcation par doublement de période

## 1.7 Attracteurs et Bassin d'attraction

La région de l'espace de phases vers laquelle convergent les trajectoires d'un système dynamique dissipatif s'appelle "attracteur". Elle est de forme géométrique particulière caractérisant l'évolution du système en régime permanent. Mathématiquement, l'ensemble  $\Omega$  est dit attracteur si [9] :

- Pour tout voisinage  $U$  de  $\Omega$ , il existe un voisinage  $V$  de  $\Omega$  tel que toute solution  $x(x_0, t) = \phi_t(x_0)$  restera dans  $U$  si  $x_0 \in V$ .
- $\cap \phi_t(V) = \Omega, t \geq 0$ .
- Il existe une orbite dense dans  $\Omega$ .

Lorsque  $\Omega$  est un attracteur, l'ensemble  $W = \cup \phi_t(V)$ , est appelé bassin d'attraction de  $\Omega$ . C'est l'ensemble des points dont les trajectoires asymptotiques convergent vers  $\Omega$ .

Il existe deux types d'attracteurs : les attracteurs réguliers et les attracteurs étranges ou chaotiques.

### 1.7.1 Attracteurs réguliers

Les attracteurs réguliers caractérisent l'évolution des systèmes non chaotiques, dont on peut citer [9] :

**a. Le point fixe**

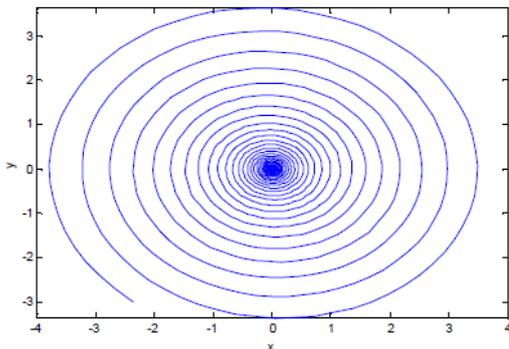
C'est un point de l'espace de phase vers lequel tendent les trajectoires, c'est donc une solution stationnaire constante.

**b. Le cycle limite périodique**

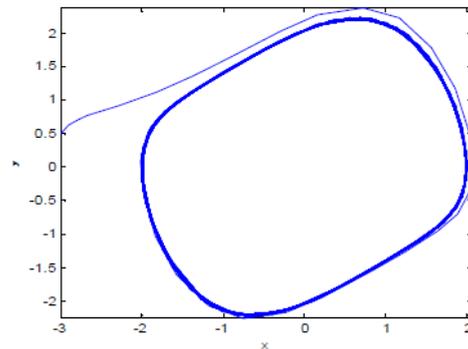
C'est une trajectoire fermée dans l'espace des phases vers laquelle tendent les trajectoires. C'est donc une solution périodique du système.

**c. Le cycle limite pseudo-périodique**

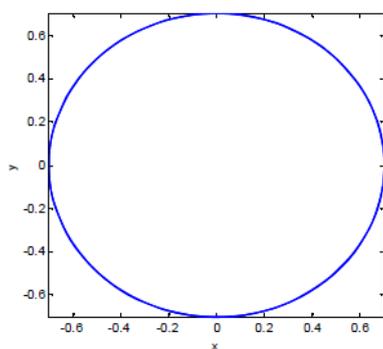
Cet attracteur est un cas particulier du précédent, il représente les mouvements résultants dus au moins à de deux oscillations indépendantes, dont le rapport des périodes est irrationnel. La trajectoire de phase ne se referme pas sur elle-même, mais s'enroule sur une variété de dimension (le cas d'un tore dans une dimension de 2). La figure 1.8 illustre les différents attracteurs réguliers.



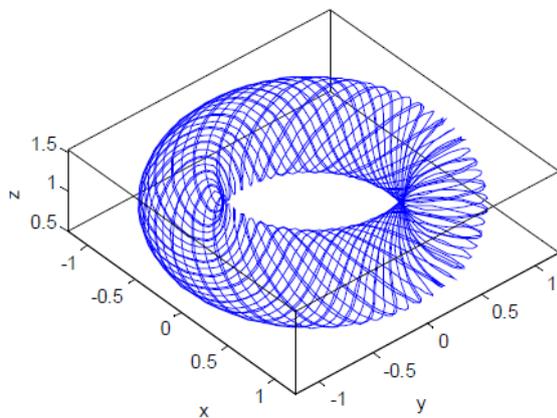
(a) Point fixe



(b) Cycle limite



(c) Cercle



(d) Tore(quasi-périodique)

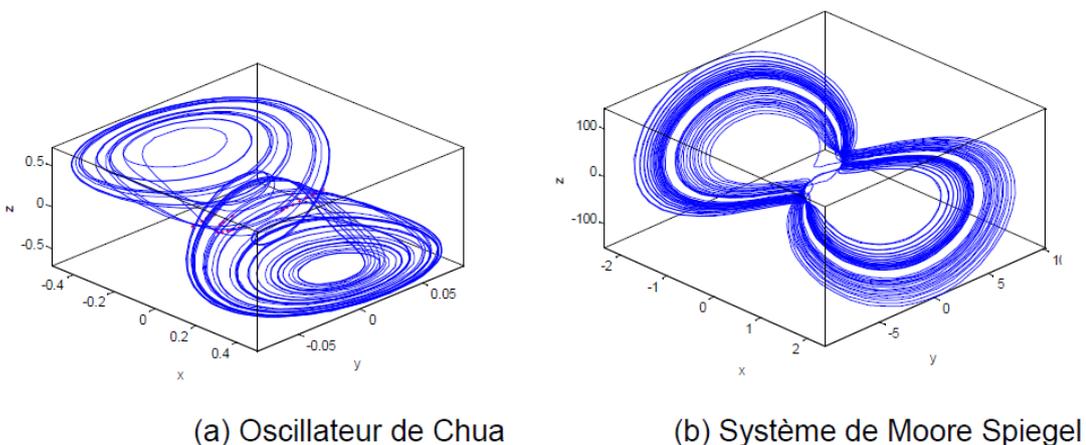
**Figure 1.8 : Les différents types d'attracteurs réguliers**

### 1.7.2 Attracteurs étranges

Ils ont des formes géométriques bien plus complexes que les autres, caractérisant l'évolution des systèmes chaotiques. En régime permanent, tous les points de l'espace de phase appartenant au bassin d'attraction donnent des trajectoires qui tendent à former l'attracteur étrange. La figure 1.9 montre des exemples sur les attracteurs étranges.

Les attracteurs étranges sont caractérisés par [12]:

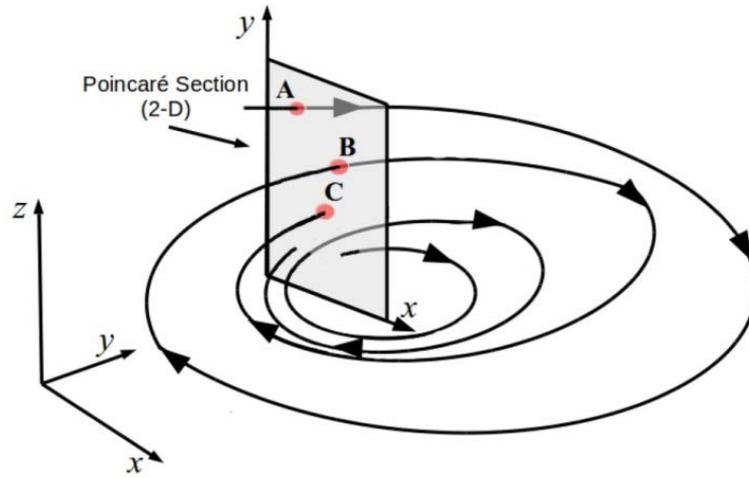
- ✓ Dans l'espace des phases, l'attracteur est de volume nul.
- ✓ La dimension  $d$  de l'attracteur est fractale (non-entière) avec  $0 < d < n$ , où  $n$  est la dimension de l'espace des phases.
- ✓ Sensibilité aux conditions initiales : deux trajectoires de l'attracteur initialement voisines finissent par s'écartier exponentiellement l'une de l'autre.



**Figure 1.9 :** Exemples d'attracteurs étranges

## 1.8 Section de Poincaré

Henri Poincaré a apporté une contribution très utile pour l'étude des systèmes chaotiques. Parmi ces contributions, on peut citer la section de Poincaré. 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 en dimension trois par exemple. On passe alors d'un système dynamique à temps continu à un système dynamique à temps discret. Les mathématiciens ont bien démontré que les propriétés du système sont conservées après la réalisation d'une section de Poincaré judicieusement choisie [11].



**Figure 1.10** : Section de Poincaré

Pour réaliser la section de Poincaré, il faut tout d'abord ramener le système dynamique de dimension  $n$  à temps continu, à un système de dimension  $n - 1$  à temps discret [9]. Pour cela, on définit une hyper surface  $\Sigma_p$  de dimension  $n - 1$  appelé section de Poincaré. On considère l'ensemble des points  $p_1, p_2, p_3, \dots$  correspondant aux intersections successives de la trajectoire  $\varphi_t(x_0)$  avec l'hyper surface  $\Sigma_p$ . L'application du premier retour  $T$  est alors définie comme étant l'application qu'est à un point  $p_i$  de  $\Sigma_p$  fait correspondre le point  $p_{i+1}$ , prochaine intersection de la trajectoire  $\varphi_t(x_0)$  avec  $\Sigma_p$ . Ainsi le système dynamique initial de dimension  $n$  à temps continu, est transformé en un système de dimension  $n - 1$  à temps discret :

$$p_{k+1} = T(p_k), k = 1, 2, 3, \dots \quad (1.16)$$

Le numéro de l'intersection remplace le temps. Un point de la section  $\Sigma_p$  définit une orbite et une seule. Une orbite du système original est alors transformée en une suite d'intersections  $p_k, k = 1, 2, 3, \dots$  (figure 1.10) caractérisant le système dynamique et dont les propriétés sont résumées dans le Tableau 1. 1.

L'étude de la stabilité d'un orbite periodique peut se faire à travers l'analyse de l'application du premier retour  $T$ .

Type d'attracteur dans l'espace de phase	Application de Poincaré
Cycle limite	1 Point
Cycle limite avec $p$ maximas par période	$p$ Points
Attracteur quasi-periodique	Courbe fermée
Attracteur étrange	Courbe(s) ouverte(s)

**Tableau 1. 1 :** Correspondance attracteur – section de Poincaré

## 1.9 Exposants de Lyapunov

Les systèmes dynamiques chaotiques sont caractérisés par leurs sensibilités aux conditions initiales, un phénomène qui se traduit par une divergence exponentielle au cours du temps entre deux orbites issues de conditions initiales voisines. Afin de comprendre et analyser ce phénomène, il est nécessaire d'utiliser un outil permettant de mesurer cette divergence et le degré de sensibilité d'un système dynamique. À cet effet, Alexandre Lyapunov a introduit un outil très puissant permettant la mesure quantitative de cette divergence exponentielle et qui porte son nom, à savoir "**Exposants de Lyapunov**".

### 1.9.1 Calcul des exposants de Lyapunov [7, 9]

Considérons le système dynamique autonome à temps continu défini par :

$$\begin{cases} \dot{x} = f(x); & f: \mathbb{R}^n \rightarrow \mathbb{R}^n \\ x(0) = x_0 \end{cases} \quad (1.17)$$

où  $\varphi(x_0, t)$  est une trajectoire solution de ce système de condition initiale  $x_0$ , et  $x_p$  est un point de cette trajectoire à  $t = t_p$  dans l'espace des phases.

Un champ de vecteurs ou champ vectoriel est une fonction qui associe un vecteur à chaque point d'un espace euclidien ou plus généralement d'une variété différentielle. Les champs de vecteurs sont souvent utilisés en physique, pour modéliser par exemple la vitesse et la direction d'un fluide en mouvement dans l'espace, ou la valeur et la direction d'une force, comme la force magnétique ou gravitationnelle, qui évoluent point par point.

Le calcul des exposants de Lyapunov consiste dans un premier temps à linéariser le vecteur champ au voisinage d'un point de la trajectoire considérée. Soient  $\varphi(x_0, t)$  cette trajectoire et  $x_p$  un point de la trajectoire ( $x_p = \varphi(x_p, t_p)$ ).

En considérant une petite perturbation  $\delta x_p(t)$  appliquée au voisinage  $x_p$  et en développant en série de Taylor du premier ordre le vecteur champ  $f(x_p)$ , le système linéarisé autour de  $x_p$  s'écrit :

$$\frac{d\delta x_p}{dt} = J_f(x_p)\delta x_p \quad (1.18)$$

où  $J_f(x_p)$  est la matrice Jacobienne de  $f$  au point  $x_p$ .

Il s'agit ensuite d'intégrer chacune des composantes  $x_k(t)$ ,  $k = 1, 2, \dots, n$  de la trajectoire  $\varphi(x_0, t)$  à partir de l'équation (1.17). Chacune de ces composantes  $x_k(t)$  est intégrée et introduite dans l'équation (1.18). La dernière opération consiste à intégrer le système (1.18) lui-même. Au final, nous obtenons une matrice  $\Phi_t(x_p)$  de dimension  $n \times n$  appelée matrice de la solution fondamentale.

Toute perturbation  $\delta x_p(t)$  à  $t = t_p$ , au voisinage  $\delta x_p(t)$  d'un point  $x_p$ , de la trajectoire pourra s'écrire sous la forme :

$$\delta x_p(t) = \Phi(\delta x_p(t)) \quad (1.19)$$

Soient  $\mu_k(t)$ ,  $k = 1, 2, \dots, n$  les valeurs propres de cette matrice. L'exposant de Lyapunov  $\lambda_k$  du  $k^{\text{ième}}$  ordre est lié à la valeur propre  $\mu_k$  par la formule suivante :

$$\lambda_k = \lim_{t \rightarrow \infty} \frac{1}{n} \ln |\mu_k(t)| \quad (1.20)$$

L'exposant de Lyapunov  $\lambda_k$  existe dans la mesure où la limite existe.

**N.B :** Si  $\lambda_k > 0$ , la distance entre les trajectoires augmente exponentiellement provoquant le régime chaotique. Si  $\lambda_k < 0$ , la distance entre les trajectoires converge vers le zéro conduisant le système vers un régime régulier.

## 1.9.2 Comportement des systèmes dynamiques en fonction des exposants de Lyapunov

En étudiant les exposants de Lyapunov d'un système dynamique, on peut définir le type d'attracteur (comportement asymptotique) généré par le système :

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

## 1.10 Exemples de systèmes chaotiques

Dans cette partie, on présente deux exemples les plus célèbres des systèmes chaotiques, ceux de Lorenz et de Rössler, tout en illustrant les variables d'états en fonction du temps, les plans des phases, exposants de Lyapunov et le diagramme de bifurcation.

Les résultats de la simulation présentés dans cette partie, sont obtenus à partir d'un code que nous avons développé sous Matlab (ANNEXE 01), relativement à la réponse temporelle des systèmes, plans des phases et les diagrammes de bifurcations. Pour les exposant de Lyapunov, on a utilisé une Matlab-GUI (Guide User Interface) téléchargé depuis internet : "Matds". Pour la méthode d'intégration, on a utilisé la fonction de Matlab **ODE45** (Méthode de Runge Kutta 4 à pas variable).

Nous avons également implémenté ces dits systèmes sous Matlab Simulink.

### 1.10.1 Système de Lorenz

Ce système a été introduit par Edward N. Lorenz [1, 2] en 1963 dans le cadre de l'étude des prévisions météorologiques : c'est le premier et le plus connu des systèmes

dynamiques chaotiques ; il constitue ainsi la référence du chaos. Il est défini par les équations suivantes :

$$\begin{cases} \dot{x}_1 = \sigma(x_2 - x_1) \\ \dot{x}_2 = rx_1 - x_2 - x_1x_3 \\ \dot{x}_3 = -bx_3 + x_1x_3 \end{cases} \quad (1.21)$$

où  $\sigma, r$  et  $b$  sont les paramètres du système, ils représentent dans la pratique des coefficients météorologiques [10]. Le système (1.21) présente un comportement chaotique pour les valeurs de ces paramètres suivantes [1] :  $\sigma = 10$ ,  $r = 28$  et  $b = \frac{8}{3}$ .

Dans ce qui suit, on présente les différentes caractéristiques du système de Lorenz obtenues par simulation sous Matlab, dont les conditions initiales du système sont les suivantes :  $x_1(0) = 0$ ,  $x_2(0) = 1$ ,  $x_3(0) = 0$ .

#### a. Réponse temporelle

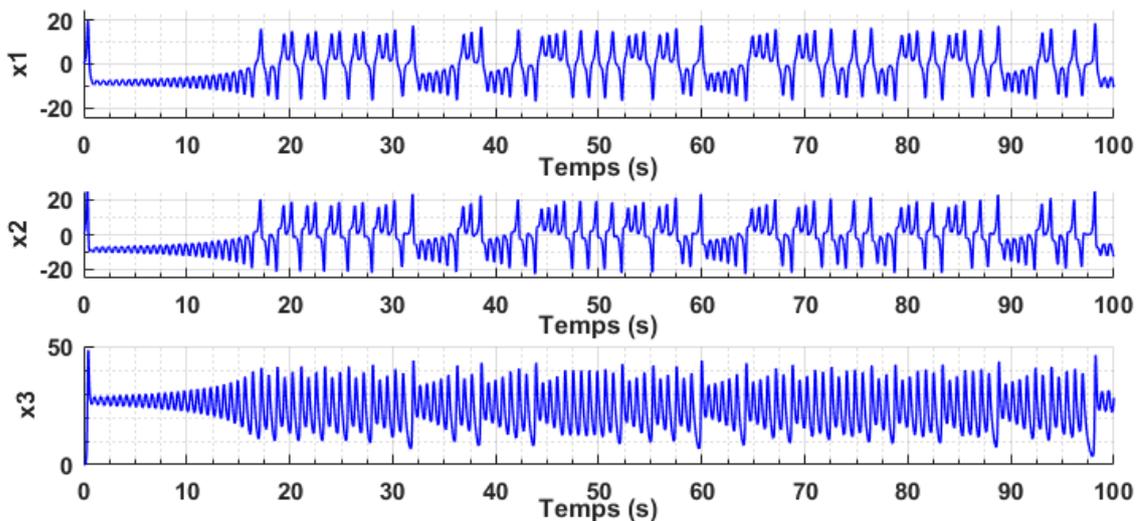


Figure 1.11 : Réponse temporelle du système chaotique de Lorenz

#### b. Exposants de Lyapunov

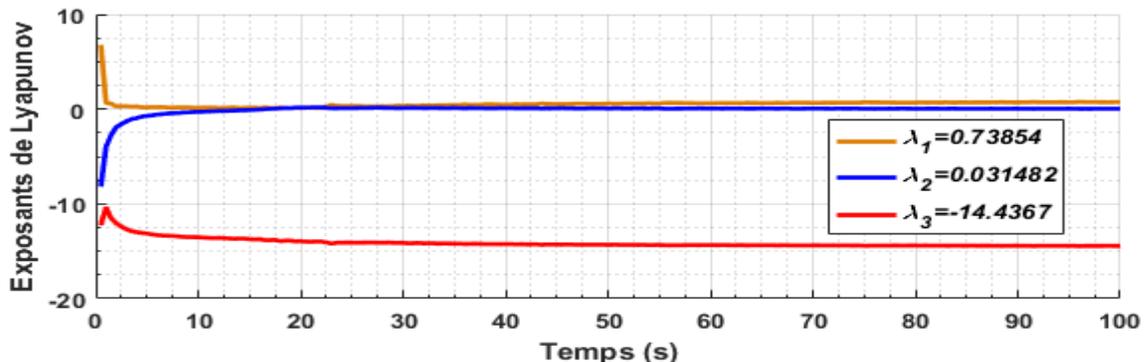
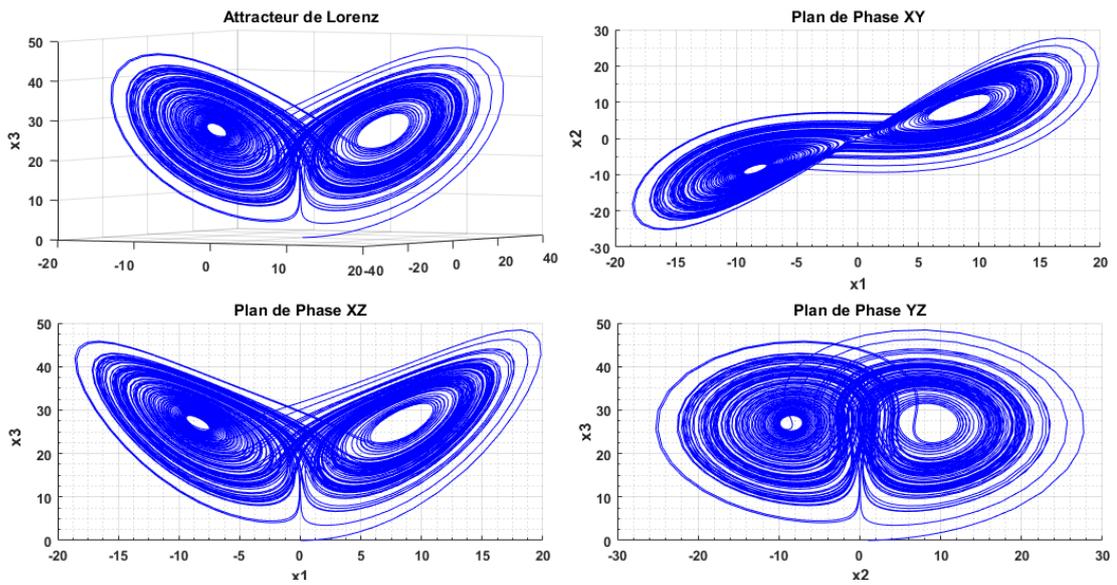


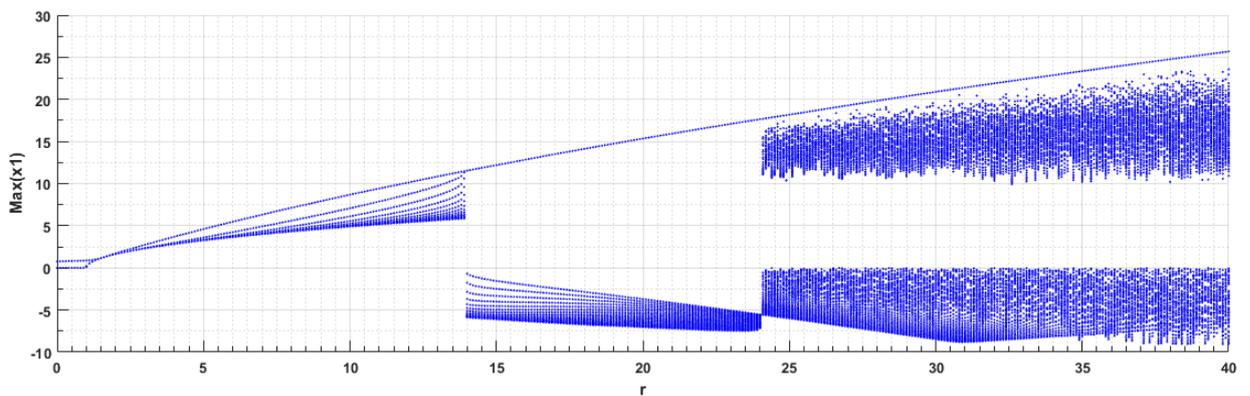
Figure 1.12 : Évolution des exposants de Lyapunov du système chaotique de Lorenz

**c. Plans des phases**



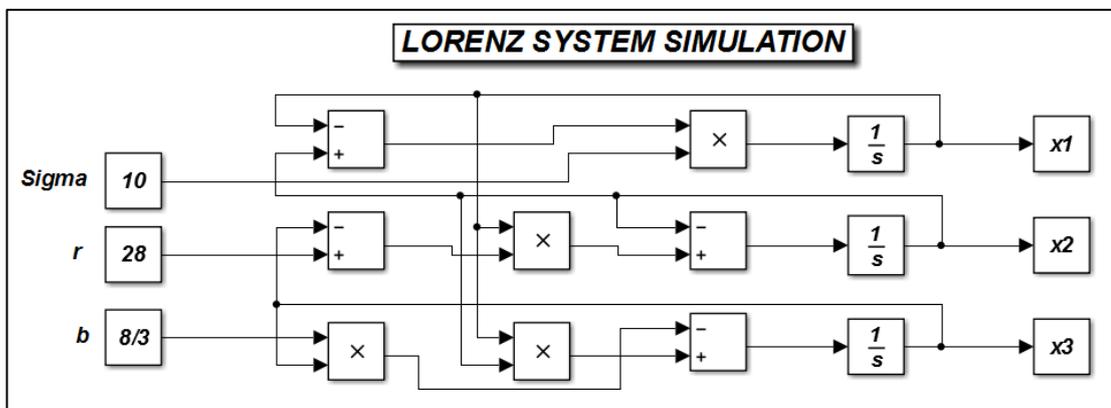
**Figure 1.13 :** Représentation du système chaotique de Lorenz dans l'espace des phases

**d. Diagramme de bifurcation**



**Figure 1.14 :** Diagramme de bifurcation pour le système chaotique de Lorenz. avec  $r = [0-40]$  et un pas égale à 0.1

**e. Modèle Matlab Simulink**



**Figure 1.15 :** Modèle Matlab Simulink du système chaotique de Lorenz

### 1.10.2 Système de Rössler

Ce système a été proposé en 1970, par l'Allemand Otto Rössler, dans le cadre de l'étude de l'écoulement des fluides. Il est défini par les équations suivantes :

$$\begin{cases} \dot{x}_1 = -(x_1 + x_3) \\ \dot{x}_2 = x_1 + ax_2 \\ \dot{x}_3 = b + x_3(x_1 - c) \end{cases} \quad (1.22)$$

Le système (1.22) présente un comportement chaotique pour les valeurs de ces paramètres suivantes [15] :  $a = 0.38, b = 0.3$  et  $c = 4.820$ . La partie suivante représente les résultats de simulation du système de Rössler sous Matlab. On a pris comme conditions initiales du système, les valeurs suivantes :  $x_1(0) = 0.01, x_2(0) = 0.01, x_3(0) = 0.01$ .

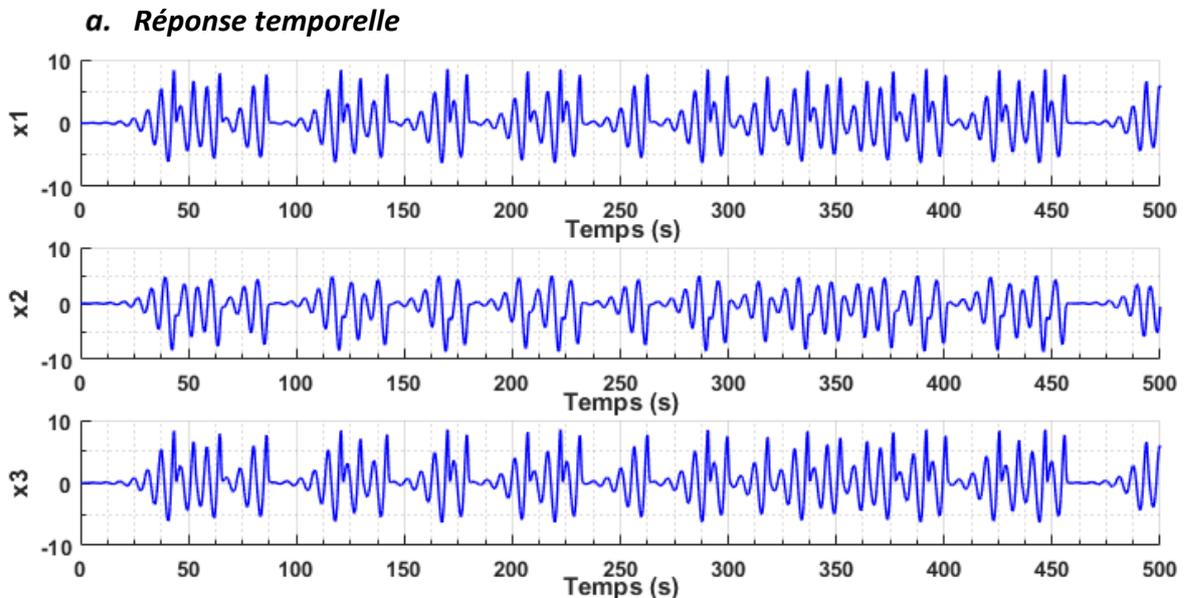


Figure 1.16 : Réponse temporelle du système chaotique de Rössler

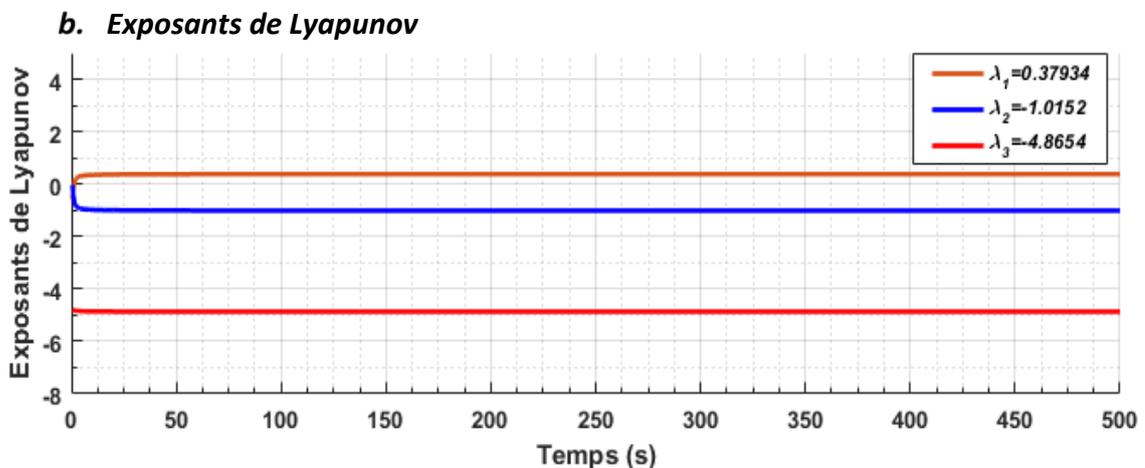
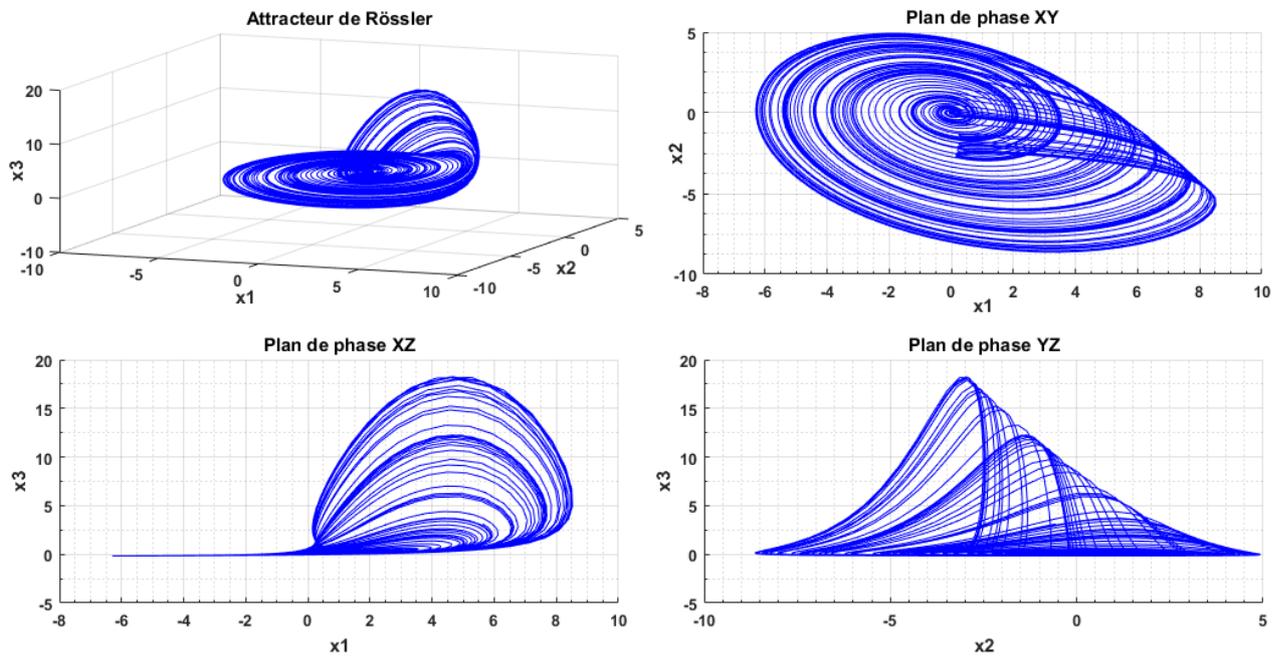


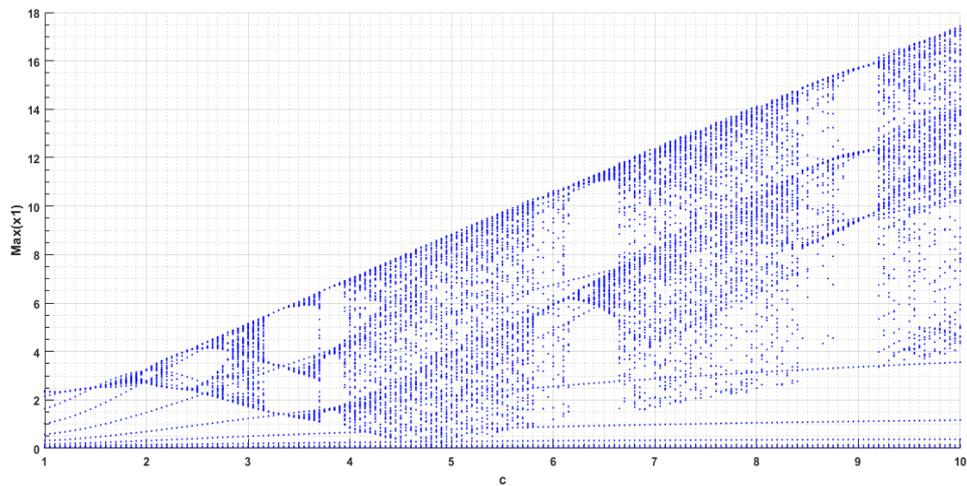
Figure 1.17 : Évolution des exposants de Lyapunov du système chaotique de Rössler

**c. Plans des phases**



**Figure 1.18 :** Représentation du système chaotique de Rössler dans l'espace des phases

**d. Diagramme de bifurcation**



**Figure 1.19 :** Diagramme de bifurcation pour le système chaotique de Rössler. avec  $c \in [1-10]$  et un pas égale à 0.05

### e. Modèle Matlab Simulink

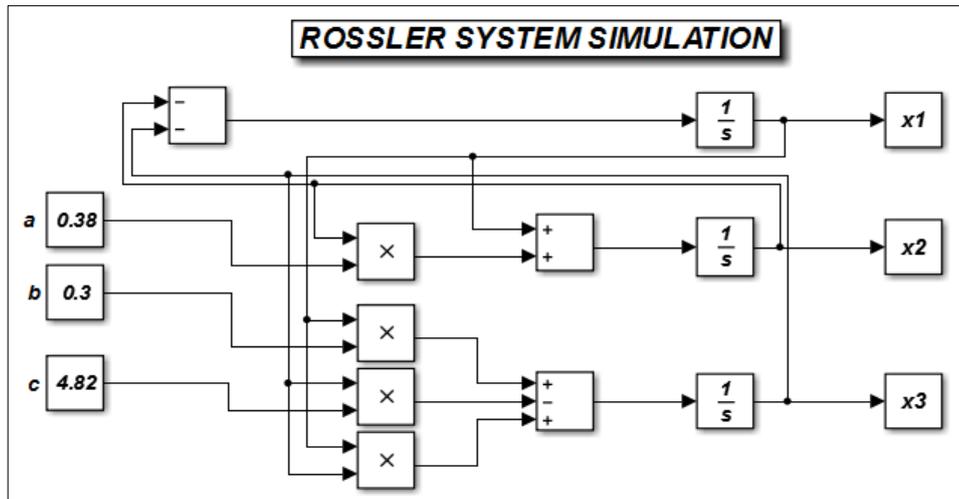


Figure 1.20 : Modèle Matlab Simulink du système chaotique de Rössler

## 1.11 Conclusion

Dans ce chapitre on a présenté quelques notions sur la théorie des systèmes dynamiques et du chaos, tout en mettant l'accent sur les principales caractéristiques d'un système chaotique, à savoir : Plans des phases, les points fixes et leurs stabilités, la bifurcation et les différents scénarios évoluant vers le chaos, les attracteurs, etc. On a montré par la suite, l'intérêt du calcul des exposants de Lyapunov. Les résultats de simulation sous Matlab des deux systèmes les plus célèbres Lorenz et Rössler sont présentés en fin du chapitre.

Tous ces aspects vont être exploités dans les chapitres suivants, lors de l'étude du système chaotique Unifié qui sera utilisé pour masquer notre signal informatif, ainsi que dans la construction du contrôleur pilotant le système chaotique esclave au niveau du récepteur afin qu'il puisse se synchroniser avec celui de l'émetteur pour la récupération du message.

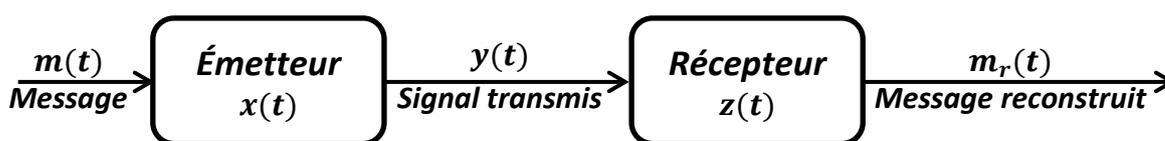
## Chapitre 2 ÉTUDE DU SYSTÈME CHAOTIQUE UNIFIÉ

---

### 2.1 Introduction

En 1990, Pecora et Carroll [3, 4] ont mis en œuvre l'idée de synchroniser deux systèmes chaotiques identiques, dont ils ont pris comme exemples les systèmes chaotiques de Lorenz et Rössler. Durant les deux dernières décennies, la synchronisation chaotique était l'objet de plusieurs recherches dus à son application potentielle dans divers domaines [16] : transmissions sécurisées de l'information, systèmes biologiques et réactions chimiques, etc. Notre travail consiste à la réalisation d'une transmission sécurisée par systèmes chaotiques.

La conception d'un système de communication chaotique peut être décomposée en trois étapes [9] à savoir le choix de l'émetteur, le choix du récepteur et la mise au point de la synchronisation de la transmission chaotique pour la récupération du message crypté (figure 2.1).



**Figure 2.1** : Principe général d'un système de communications

Les systèmes chaotiques constituent le noyau de ce système de communication et le choix du ou des système(s) chaotique(s) à utiliser au niveau de l'émetteur, joue un rôle très important afin d'augmenter le niveau de sécurité des communications. Actuellement, il existe une large variété d'approches de synchronisation : identique, généralisée, retardée, adaptative, etc. (cet aspect sera bien détaillé dans le chapitre suivant). Nous nous intéressons à la synchronisation adaptative qui est conçue généralement pour synchroniser

différents systèmes chaotiques avec des paramètres inconnus. À cet effet, notre choix s'est porté sur le système chaotique unifié [17], qui englobe dans un seul système d'équation, trois systèmes chaotiques : Lorenz, Chen et Lü. Ledit système sera l'objet de ce chapitre.

Dans ce chapitre, nous commençons par une présentation des différents systèmes chaotiques constituant le système chaotique unifié : Chen et Lü sachant que le système chaotique de Lorenz a été présenté dans le chapitre précédent, puis nous introduisons le système chaotique unifié avec ses différentes caractéristiques et propriétés. Ce chapitre se termine par des résultats de simulation des dits systèmes chaotiques, tout en illustrant les caractéristiques d'un système chaotique : réponse temporelle, diagramme de bifurcation, exposants de Lyapunov et plans des phases.

Les résultats de la simulation présentés dans ce chapitre, sont obtenus à partir d'un code que nous avons développé sous Matlab (ANNEXE 01), relativement à la réponse temporelle des systèmes, plans des phases et les diagrammes de bifurcations. Pour les exposant de Lyapunov, on a utilisé une Matlab-GUI (Guide User Interface) téléchargé depuis internet : "Matds". Pour la méthode d'intégration, on a utilisé la fonction de Matlab ODE45 (Méthode de Runge Kutta 4 à pas variable).

Nous avons également implémenté ces dits systèmes sous Matlab Simulink.

## 2.2 Le système chaotique de Chen

### 2.2.1 Présentation du système chaotique de Chen

En 1999, Chen et Ueta [18, 19] ont introduit un nouveau système chaotique, dénommé par la suite : système chaotique de Chen, qui est similaire mais topologiquement pas équivalent aux systèmes chaotiques de Lorenz et Rössler. Le système est ainsi décrit par les équations suivantes :

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1) \\ \dot{x}_2 = (c - a)x_1 - x_1x_3 + cx_2 \\ \dot{x}_3 = x_1x_2 - bx_3 \end{cases} \quad (2.1)$$

où  $a, b, c$  sont des paramètres réels strictement positifs.

Le système (2.1) présente un comportement chaotique pour les valeurs de ces paramètres suivantes :  $a = 35, b = 3$  et  $c = 28$ .

### 2.2.2 Résultats de simulation du système chaotique de Chen

La partie suivante présente les résultats de simulation des différentes caractéristiques du système de Chen sous Matlab. On a pris comme conditions initiales du système, les valeurs suivantes :  $x_1(0) = 0, x_2(0) = 1, x_3(0) = 0$ .

#### a. Réponse temporelle

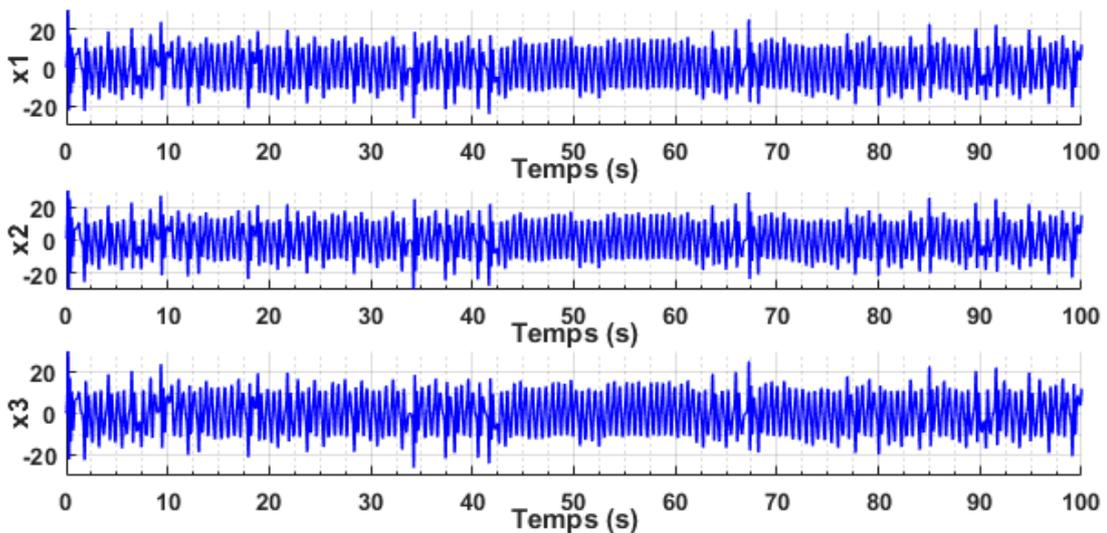


Figure 2.2 : Réponse temporelle du système chaotique de Chen

#### b. Plans des phases

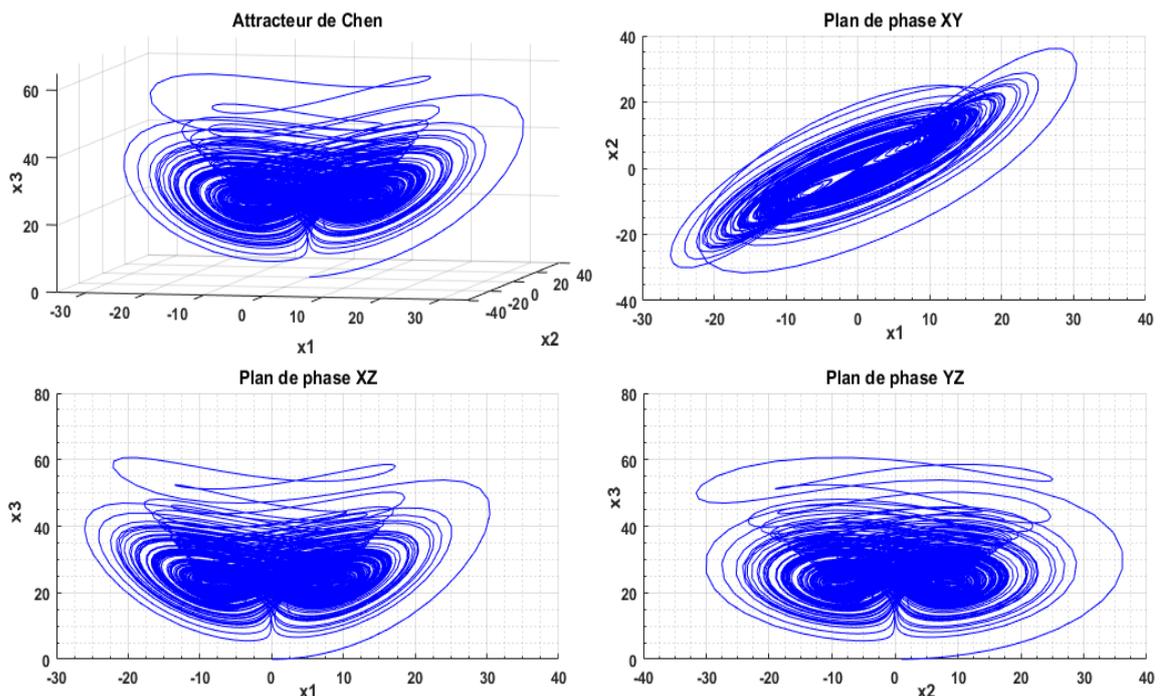
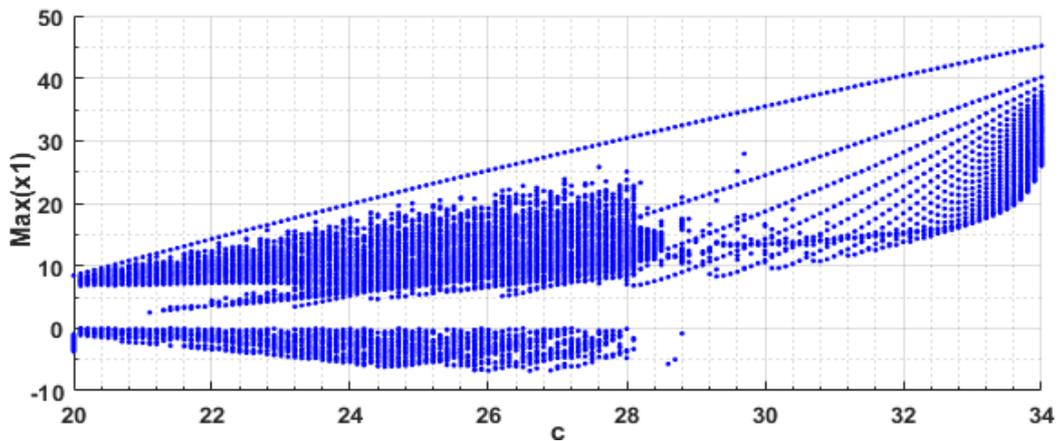


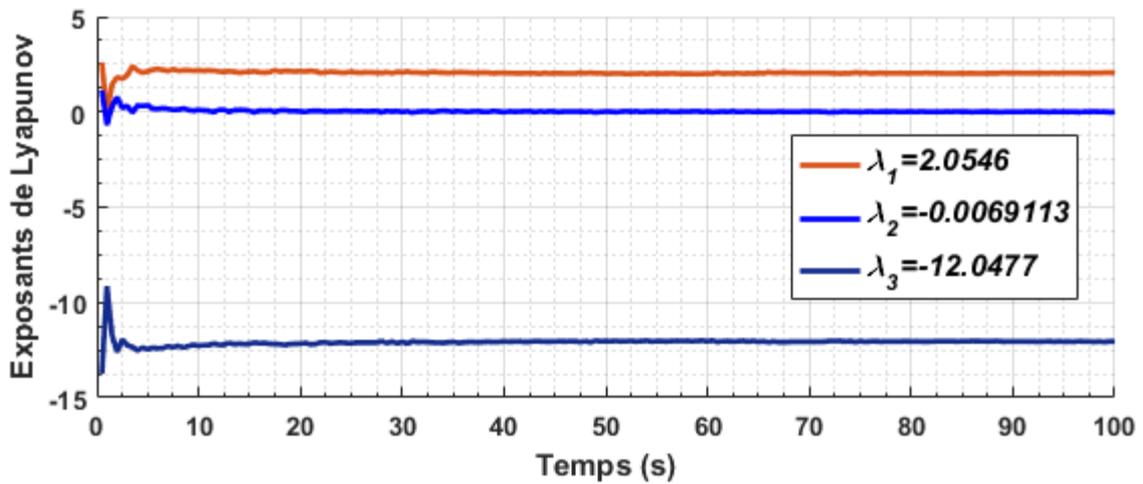
Figure 2.3 : Représentation du système chaotique de Chen dans l'espace des phases

**c. Diagramme de bifurcation**



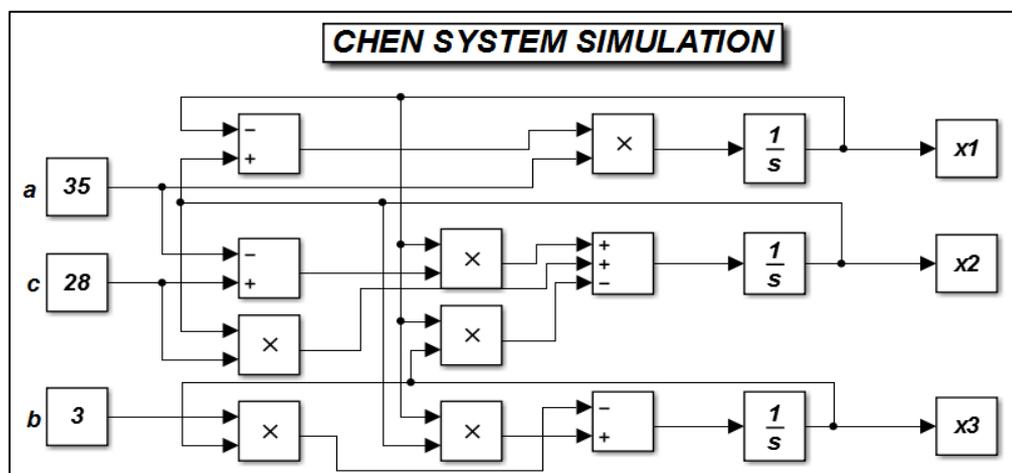
**Figure 2.4 :** Diagramme de bifurcation pour le système chaotique de Chen. avec  $c = [20-34]$  et un pas égale à 0.1

**d. Exposants de Lyapunov**



**Figure 2.5 :** Évolution des exposants de Lyapunov du système chaotique de Chen

**e. Modèle Matlab Simulink**



**Figure 2.6 :** Modèle Matlab Simulink du système chaotique de Chen

## 2.3 Le système chaotique de Lü

### 2.3.1 Présentation du système chaotique de Lü

En 1996, Vanecek et Celikovský [20] ont classifié une famille canonique du système chaotique généralisé de Lorenz par une condition sur sa partie linéaire  $A = [a_{ij}]$  tels que  $a_{12}a_{21} > 0$ . Cependant le système chaotique de Chen [18] qui est similaire au système chaotique de Lorenz satisfait la condition  $a_{12}a_{21} < 0$ , appartenant ainsi à une autre famille canonique des systèmes chaotiques [21]. La question qui se pose : existe-t-il un système chaotique entre le système chaotique de Lorenz et le système chaotique de Chen qui satisfait la condition  $a_{12}a_{21} = 0$  ? En réponse à cette question, en 2002, Lü et Chen [22] ont introduit un nouveau système chaotique, appelé par la suite : système chaotique de Lü, qui représente la transition entre les deux systèmes chaotiques de Lorenz et Chen, et satisfaisant la condition  $a_{12}a_{21} = 0$ . Il est décrit par les équations suivantes :

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1) \\ \dot{x}_2 = -x_1x_3 + cx_2 \\ \dot{x}_3 = x_1x_2 - bx_3 \end{cases} \quad (2.2)$$

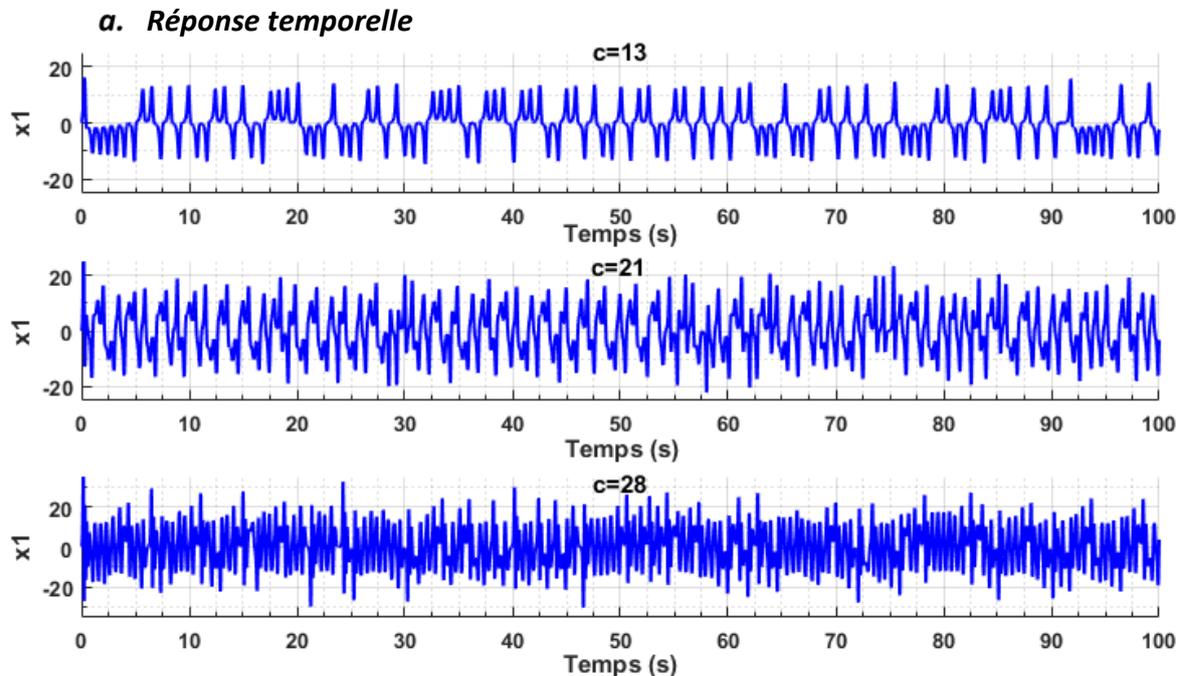
où  $a, b$  et  $c$  sont les paramètres du système, dont les paramètres  $a = 36$  et  $b = 3$  sont fixes, le paramètre  $c$  étant variable. Selon la valeur du paramètre  $c$ , la forme de l'attracteur générée par le système (2.2) peut être similaire à :

- ✓ Attracteur de Lorenz, si  $12.7 < c < 17$ .
- ✓ Forme transitoire, si  $18 < c < 22$ .
- ✓ Attracteur de Chen, si  $23 < c < 28.5$ .

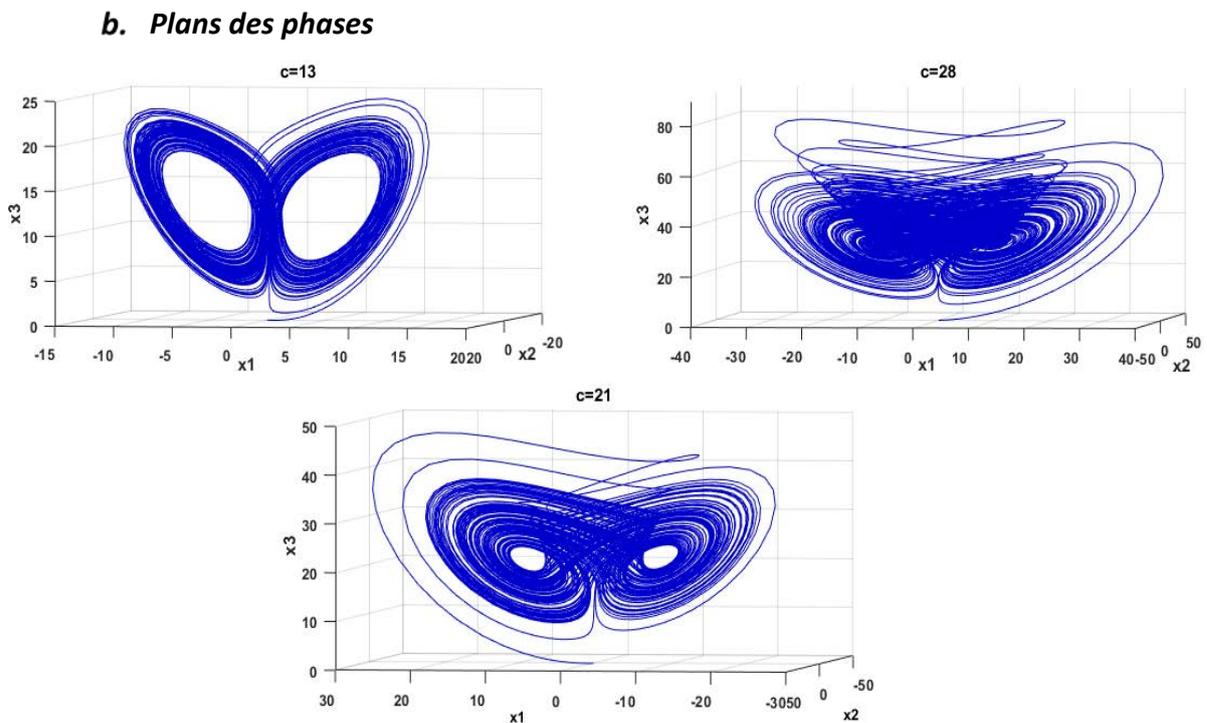
### 2.3.2 Résultats de simulation du système chaotique de Lü

La partie suivante présente les résultats de simulation des différentes caractéristiques du système chaotique de Lü sous Matlab, tout en montrant la ressemblance entre le système chaotique de Lü et les systèmes chaotiques de Lorenz et Chen et la transition entre eux. Pour cela, on a pris trois valeurs de  $c$  :  $\mathbf{c = 13}$  pour la génération de l'attracteur de Lorenz,  $\mathbf{c = 28}$  pour la génération de l'attracteur de Chen et  $\mathbf{c = 21}$  pour voir le comportement transitoire du système chaotique de Lü entre eux.

Pour pouvoir faire une comparaison entre les résultats de ce système et les systèmes chaotiques de Lorenz et Chen, on ne représente que la composante  $x_1$  pour chaque cas relativement à la réponse temporelle et seulement l'attracteur du système relativement à l'espace des états. On a pris comme conditions initiales du système, les valeurs suivantes :  $x_1(0) = 0$  ,  $x_2(0) = 1$  ,  $x_3(0) = 0$  .



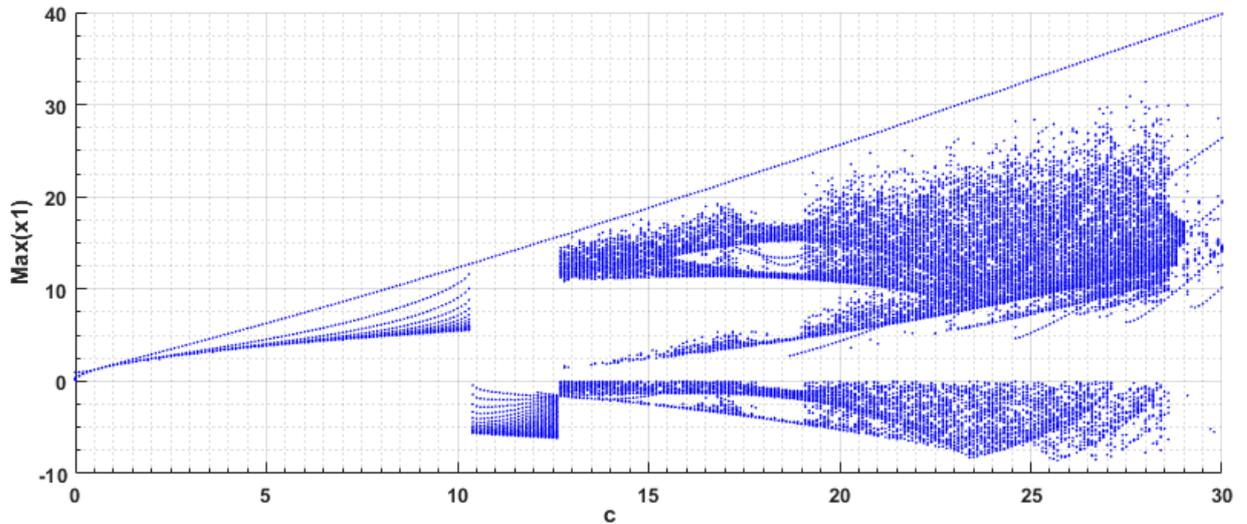
**Figure 2.7 :** Variable d'état  $x_1$  en fonction du paramètre  $c$  du système chaotique de Lü



**Figure 2.8 :** Évolution du système chaotique de Lü dans l'espace des phases selon le paramètre  $c=13, 21$  et  $28$

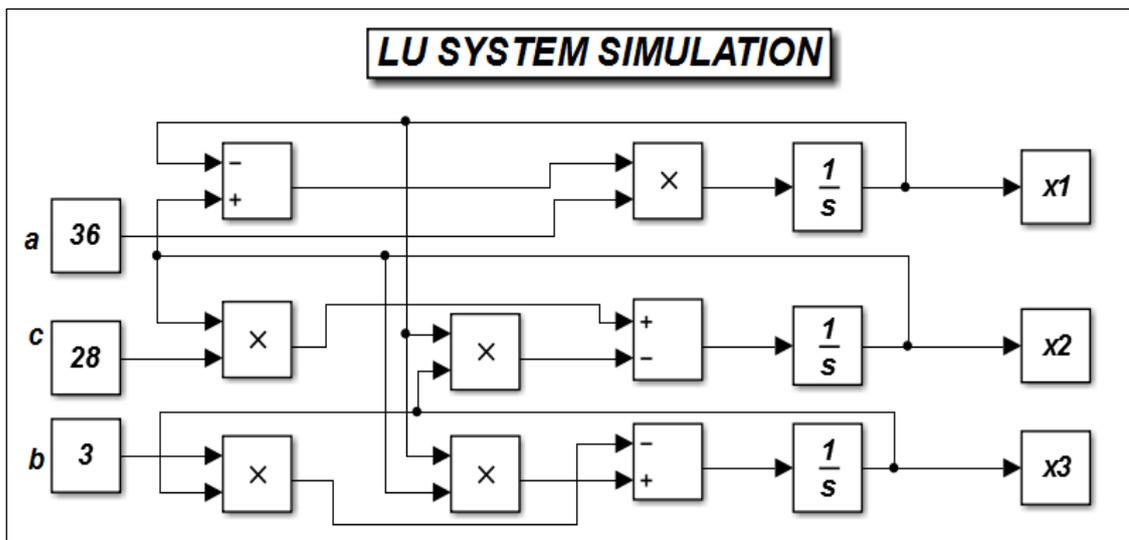
En comparant la figure 2.7 avec les figures 1.11 et 2.2, et la figure 2.8 avec les figures 1.13 et 2.3, il est bien clair que le système chaotique de Lü assure la transition entre la famille des systèmes généralisés de Lorenz et celle de Chen.

**c. Diagramme de bifurcation**



**Figure 2.9 :** Diagramme de bifurcation pour le système chaotique de Lü. avec  $c = [0-30]$  et un pas égale à 0.1

**d. Modèle Matlab Simulink**



**Figure 2.10 :** Modèle Matlab Simulink du système chaotique de Lü

## 2.4 Le système chaotique Unifié

### 2.4.1 Présentation du système chaotique Unifié

Suite aux travaux de Chen et autres [18, 19, 21, 22] présentés dans la section précédente, on se retrouve avec trois systèmes chaotiques qui sont similaires et complémentaires entre eux, mais qui sont topologiquement pas équivalents. Ces systèmes ont des propriétés communes [17], à savoir : même symétrie, caractère dissipatif, stabilité des points d'équilibres et bifurcation similaire. La question qui se pose maintenant est : existe-t-il un unique système chaotique qui peut unifier lesdits systèmes ?

Lü et al. [17] ont introduit un système chaotique nommé système chaotique Unifié qui répond à cette problématique. Il est décrit par les équations suivantes :

$$\begin{cases} \dot{x}_1 = (25\alpha + 10)(x_2 - x_1) \\ \dot{x}_2 = (28 - 35\alpha)x_1 - x_1x_3 + (29\alpha - 1)x_2 \\ \dot{x}_3 = x_1x_2 - \frac{\alpha + 8}{3}x_3 \end{cases} \quad (2.3)$$

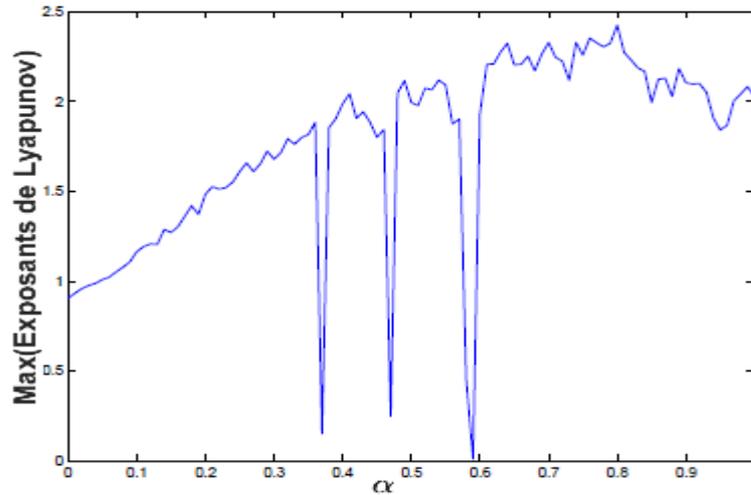
où  $\alpha$  est le paramètre du système, tels que  $\alpha \in [0,1]$ .

Selon Vanecek et Celikovský [20], la partie linéaire du système (2.3) est une matrice constante  $A = [a_{ij}]$ , ayant une valeur critique  $a_{12}a_{21}$  qui dépend du paramètre du système  $\alpha$ . Selon l'augmentation de ce paramètre de 0 à 1, le système chaotique Unifié évolue de l'attracteur de Lorenz vers l'attracteur de Chen comme suit [17] :

- ✓ Si  $0 \leq \alpha < 0.8 \Rightarrow a_{12}a_{21} < 0$  alors le système (2.3) fait partie de la famille du système généralisé de Lorenz.
- ✓ Si  $\alpha = 0.8 \Rightarrow a_{12}a_{21} = 0$  alors le système (2.3) fait partie de la famille du système généralisé de Lü.
- ✓ Si  $0.8 < \alpha < 1 \Rightarrow a_{12}a_{21} > 0$  alors le système (2.3) fait partie de la famille du système généralisé de Chen.

## 2.4.2 Caractéristiques et avantages du système chaotique Unifié

- ✓ Le système (2.3) est très simple et a un seul paramètre de contrôle  $\alpha$ .
- ✓ Le système (2.3) est chaotique quand  $\alpha \in [0,1]$ . La figure 2.11 illustre le comportement chaotique de ce système, en représentant le maximum des exposants de Lyapunov en fonction du paramètre  $\alpha$  [17], qui est positif.
- ✓ Le système (2.3) connecte les systèmes dynamiques de Lorenz et Chen, tout en assurant la transition spectrale complète entre eux.
- ✓ Le contrôle du paramètre  $\alpha$  dans le système (2.3), montre l'évolution du comportement dynamique du système (2.3), de l'attracteur de Lorenz vers l'attracteur de Chen.
- ✓ La figure 2.11 montre l'existence de trois fenêtres périodiques ( $W_1 = [0.369, 0.371]$ ,  $W_2 = [0.468, 0.470]$ ,  $W_3 = [0.575, 0.597]$ ) dont le maximum des exposants de Lyapunov se rapproche vers zéro [17]. Ces fenêtres divisent l'intervalle entier en quatre sous intervalles comme suit :
  - $I_1 = [0, 0.36]$  : la structure topologique du système (2.3) est similaire à l'attracteur du système de Lorenz original.
  - $I_2 = [0.38, 0.46]$  : la structure topologique du système (2.3) est une variante de l'attracteur du système de Lorenz original.
  - $I_3 = [0.48, 0.57]$  : la structure topologique du système (2.3) est une variante de l'attracteur du système de Chen original.
  - $I_4 = [0.6, 1]$  : la structure topologique du système (2.3) est similaire à l'attracteur du système de Chen original.
- ✓ Les fenêtres périodiques de système (2.3) révèlent la vraie raison pour laquelle les attracteurs de Lorenz et Chen sont similaires, mais différents topologiquement.



**Figure 2.11** : Évolution du maximum des exposants de Lyapunov du système Unifié en fonction du paramètre  $\alpha$

### 2.4.3 Propriétés du système chaotique Unifié

Le système (2.3) partage plusieurs propriétés importantes avec les systèmes chaotiques de Lorenz et Chen, à savoir :

#### a. Symétrie et invariance

Les deux systèmes chaotiques de Lorenz et Chen présentent une symétrie par rapport à l'axe  $x_3$  sous la transformation  $(x_1, x_2, x_3) \rightarrow (-x_1, -x_2, x_3)$ . Sous la même transformation il est simple de vérifier l'invariance du système (2.3), c'est-à-dire la réflexion autour de l'axe  $x_3$  ; ainsi, cette symétrie persiste pour toutes les valeurs du paramètre  $\alpha \in [0,1]$ . Il est clair aussi que l'axe  $x_3$  lui-même est une orbite, c'est-à-dire que si  $x_1 = x_2 = 0$  à l'instant  $t = t_0$ , alors il restera  $x_1 = x_2 = 0$  pour tous  $t \geq t_0$ . En outre, la trajectoire sur l'axe  $x_3$  tend vers l'origine lorsque  $t \rightarrow \infty$ , car pour une telle trajectoire, on a :  $\frac{dx_1}{dt} = \frac{dx_2}{dt} = 0$  et  $\frac{dx_3}{dt} = -\frac{\alpha+8}{3}x_3$ . Donc le système (2.3) partage la symétrie et l'invariance avec les systèmes chaotiques de Lorenz et Chen pour toutes les valeurs du paramètre  $\alpha \in [0,1]$ .

#### b. Dissipation et existence des attracteurs

Pour montrer le caractère dissipatif du système (2.3), Lü et al. [17] ont utilisé sa divergence, qui est donné par l'équation suivante :

$$\nabla V = \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z} = -\frac{41 - 11\alpha}{3} \quad (2.4)$$

Il est clair que pour toutes les valeurs du paramètre  $\alpha \in [0,1]$ , on a :

$$41 - 11\alpha > 0 \Rightarrow -\frac{41 - 11\alpha}{3} < 0$$

Donc la divergence du système (2.3)  $\nabla V < 0$ , alors le système (2.3) est dissipatif avec une vitesse de contraction exponentielle égale à :  $\frac{dV}{dt} = e^{-\frac{(41-11\alpha)}{3}}$ , c'est-à-dire qu'un élément de volume  $V_0$  est contracté par le flux dans un élément de volume  $V_0 e^{-\frac{(41-11\alpha)}{3}}$  au temps  $t$ . Cela signifie que chaque volume contenant la trajectoire du système se rétrécit vers zéro lorsque  $t \rightarrow \infty$  à une vitesse exponentielle égale à :  $-\frac{(41-11\alpha)}{3}$ , d'où le caractère dissipatif du système (2.3) et par conséquent l'existence des attracteurs.

Ainsi le système (2.3) présente un comportement dissipatif comme les systèmes de Lorenz et de Chen, et ce pour toutes les valeurs du paramètre  $\alpha \in [0,1]$ .

### c. Points d'équilibres et stabilité

Pour trouver les points d'équilibre du système (2.3), on doit résoudre le système d'équations :  $\dot{x}_1 = \dot{x}_2 = \dot{x}_3 = 0$ , ce qui donne :

$$\begin{cases} (25\alpha + 10)(x_2 - x_1) = 0 \\ (28 - 35\alpha)x_1 - x_1x_3 + (29\alpha - 1)x_2 = 0 \\ x_1x_2 - \frac{\alpha + 8}{3}x_3 = 0 \end{cases} \quad (2.5)$$

Par un simple calcul, on peut trouver que le système (2.5) possède trois solutions correspondant aux points d'équilibres du système (2.3) :

$$\begin{aligned} & S_0(0,0,0) \\ & S_+(\sqrt{(8+\alpha)(9-2\alpha)}, \sqrt{(8+\alpha)(9-2\alpha)}, 27-6\alpha) \\ & S_-(\sqrt{(8+\alpha)(9-2\alpha)}, -\sqrt{(8+\alpha)(9-2\alpha)}, 27-6\alpha) \end{aligned}$$

où les deux point d'équilibres  $S_+$  et  $S_-$  sont symétriques par rapport à l'axe  $x_3$ .

Pour l'étude de stabilité de ces points d'équilibres, la démonstration et les résultats des travaux effectués par Lü et al. [17] sont les suivants :

- ✓ Le point d'équilibre  $S_0(0,0,0)$  est un point selle dans l'espace des phases.
- ✓ Les deux points d'équilibres  $S_+$  et  $S_-$  sont instables pour toutes les valeurs du paramètre  $\alpha \in [0,1]$ .

#### 2.4.4 Résultats de simulation du système chaotique Unifié

Dans cette partie, nous présentons les résultats de simulation des différentes caractéristiques du système chaotique Unifié sous Matlab, tout en montrant la ressemblance entre ce système et des systèmes chaotiques de Lorenz, Lü et Chen et la transition entre eux, ainsi un exemple de chaque intervalle cité dans la section 2.4.2 sera présenté. Pour cela, on a pris cinq valeurs de  $\alpha$  :  $\alpha = 0$  pour la génération de l'attracteur de Lorenz,  $\alpha = 0.4$  pour la génération d'une variante de l'attracteur de Lorenz,  $\alpha = 0.5$  pour la génération d'une variante de l'attracteur de Chen,  $\alpha = 0.8$  pour la génération de l'attracteur de Lü,  $\alpha = 1$  pour la génération de l'attracteur de Chen.

Pour pouvoir faire une comparaison entre les résultats de ce système et les systèmes chaotiques de Lorenz, Lü et Chen, on ne représente que la composante  $x_1$  pour chaque cas relativement à la réponse temporelle et seulement l'attracteur du système relativement à l'espace des états. On a pris comme conditions initiales du système, les valeurs suivantes :  $x_1(0) = 0$  ,  $x_2(0) = 1$  ,  $x_3(0) = 0$  .

##### a. Réponse temporelle

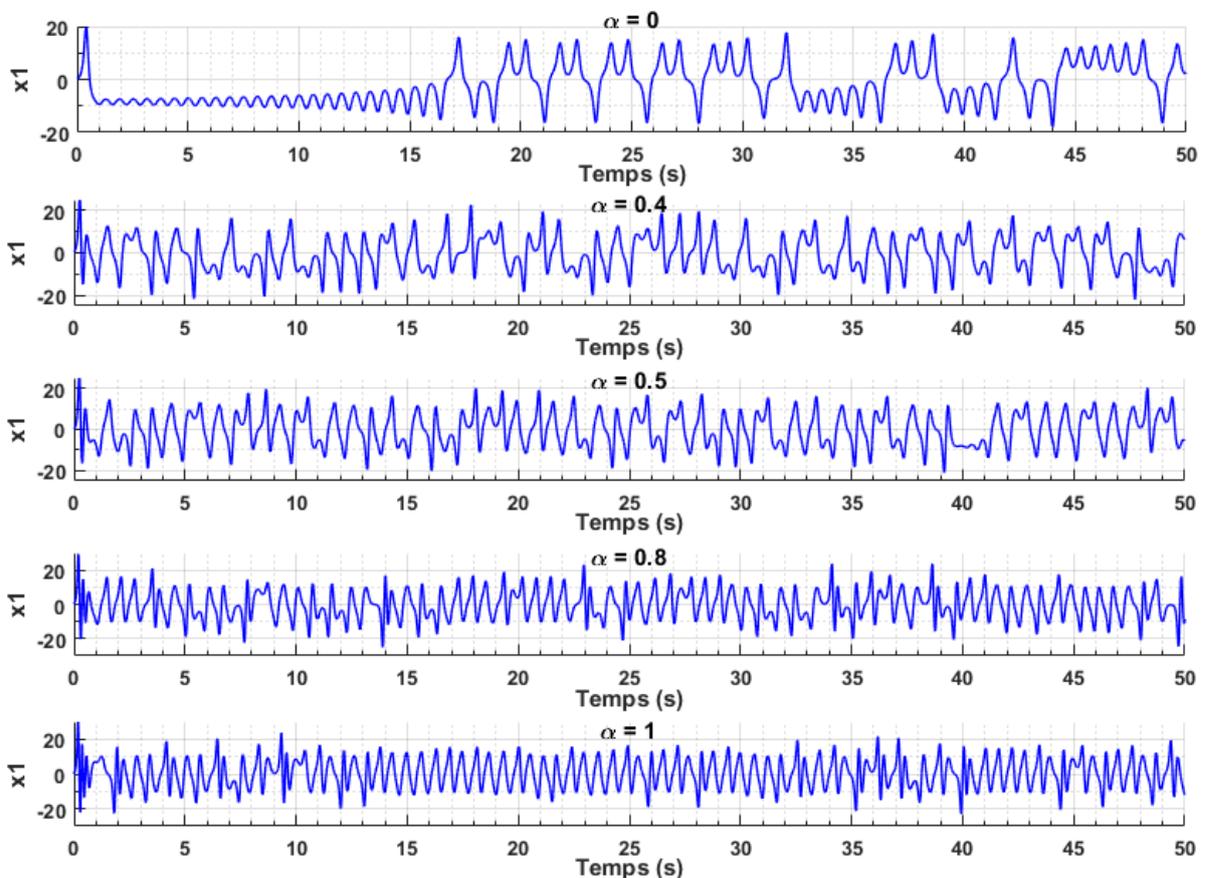
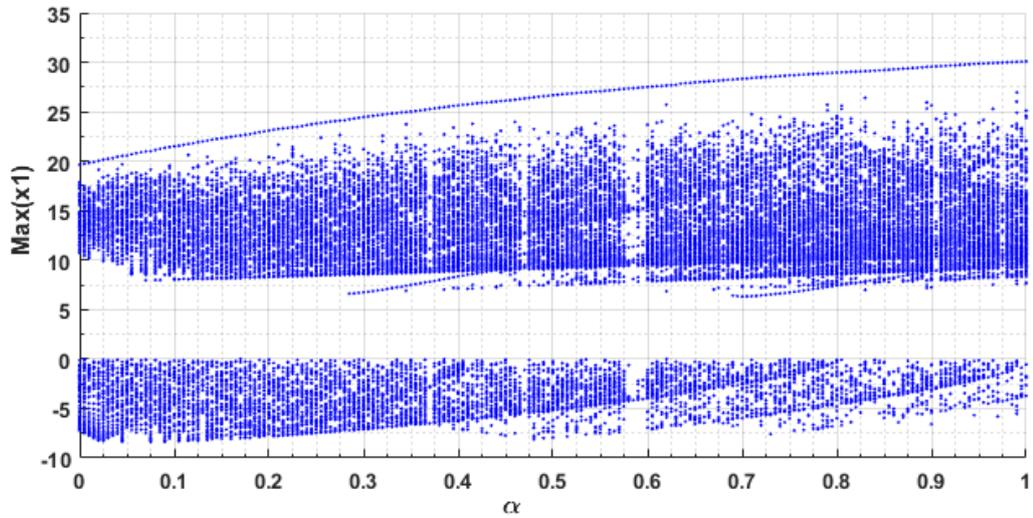


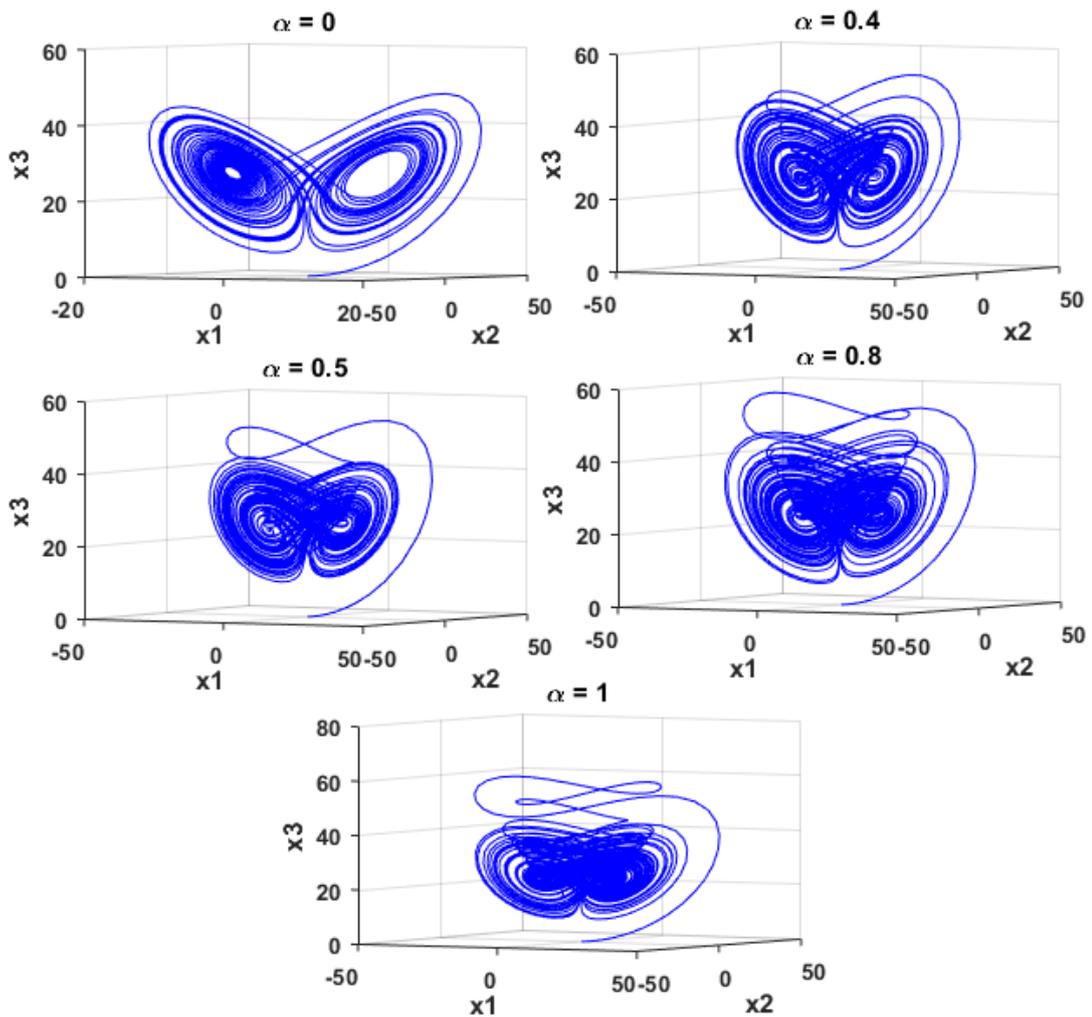
Figure 2.12 : Variable d'état  $x_1$  en fonction du paramètre  $\alpha$  du système chaotique Unifié

**b. Diagramme de bifurcation**



**Figure 2.13 :** Diagramme de bifurcation pour le système chaotique Unifié avec  $\alpha = [0 - 1]$  et un pas égale à 0.005

**c. Plans des phases**

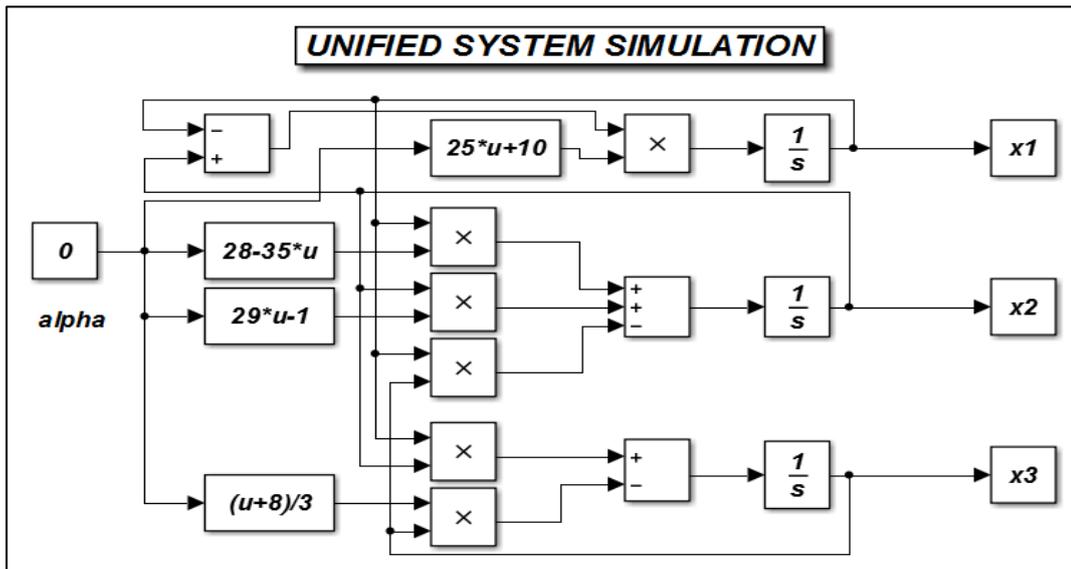


**Figure 2.14 :** Évolution du système chaotique Unifié dans l'espace des phases selon le paramètre  $\alpha = 0, 0.4, 0.5, 0.8$  et  $1$

En comparant la figure 2.12 ( $\alpha = 0$ ) avec la figure 1.11 (variable d'état  $x_1$ ) et la figure 2.14 ( $\alpha = 0$ ) avec la figure 1.13, on constate bien que le système chaotique Unifié représente bien l'attracteur de Lorenz pour  $\alpha = 0$ . De même, la comparaison de la figure 2.12 ( $\alpha = 1$ ) avec la figure 2.2 (variable d'état  $x_1$ ) et la figure 2.14 ( $\alpha = 1$ ) avec la figure 2.3, confirme la représentation fidèle du système chaotique Unifié de l'attracteur de Chen pour  $\alpha = 1$ . Ainsi pour  $\alpha = 0.8$ , les figures du système chaotique Unifié figure 2.12 ( $\alpha = 0.8$ ) et figure 2.14 ( $\alpha = 0.8$ ) sont similaires à celles de l'attracteur de Lü : figure 2.7 ( $c = 21$ ) et figure 2.8 ( $c = 21$ ) respectivement.

La figure 2.14 pour  $\alpha = 0.4$  et  $\alpha = 0.5$  montre bien l'évolution du système chaotique Unifié de l'attracteur de Lorenz vers l'attracteur de Chen, tout en augmentant le paramètre  $\alpha$  de 0 vers 1.

**d. Modèle Matlab Simulink**



**Figure 2.15 :** Modèle Matlab Simulink du système chaotique Unifié

## 2.5 Conclusion

Dans ce chapitre, on a dans un premier temps décrit les systèmes chaotiques de Chen et de Lü, avec une présentation des résultats de la simulation des dits systèmes. Ensuite on a introduit le système chaotique Unifié qui rassemble les trois systèmes chaotiques de Lorenz, Chen et Lü. Les résultats de simulation de ce système sont présentés en fin du chapitre, tout en illustrant les avantages et les caractéristiques du système chaotique Unifié, ainsi que sa fidèle représentation des systèmes chaotiques de Lorenz, Chen et Lü.

Dans le chapitre suivant, le système chaotique unifié sera intégré dans le système de communication par synchronisation adaptative, afin d'être utilisé pour masquer notre signal informatif.

### 3.1 Introduction

Comme nous avons vu dans les chapitres précédents, les systèmes chaotiques sont des systèmes déterministes, à comportement complexe, qui présentent des propriétés très intéressantes pour cacher un message confidentiel, à savoir : forte sensibilité aux conditions initiales et aux variations paramétriques, non linéarité et caractère aléatoire, large spectre de fréquences similaire à celui d'un bruit, etc. Toutes ces propriétés ont motivé les chercheurs à voir comment peut-on exploiter les systèmes chaotiques dans les transmissions sécurisées de l'information ?

L'idée était de noyer le signal informatif dans le signal chaotique généré par le système chaotique de l'émetteur, et le restituer à la réception à l'aide du signal chaotique du récepteur dupliqué de celui de l'émetteur. Afin de réaliser cette tâche, on a besoin au premier lieu, de synchroniser le système chaotique du récepteur avec celui de l'émetteur, chose qui paraît à l'époque : énigmatique, ambiguë et même impossible pour certains [7]. Contrairement au cas d'oscillations périodiques où il n'y a pas d'instabilité intrinsèque [8], la synchronisation des systèmes chaotiques présente plus de contraintes, surtout qu'un système chaotique n'est pas asymptotiquement stable [9]. Donc l'application du chaos dans les transmissions sécurisées de l'information repose essentiellement sur la synchronisation des systèmes chaotiques.

La synchronisation des systèmes chaotiques n'était possible qu'après les travaux de Pecora et Carroll [3, 4] en 1990, où ils ont démontré que deux systèmes chaotiques pourraient se synchroniser sous certaines conditions : si l'un d'eux est piloté par au moins une composante de l'autre. Depuis les années 90, de nombreux ouvrages ont été publiés au sujet de la synchronisation chaotique [23, 24, 25]. C'est donc, à partir de 1992 que les

ingénieurs ont pu réaliser des systèmes de communications à base des systèmes chaotiques.

Dans ce chapitre, nous allons définir dans un premier temps : le concept de la synchronisation chaotique, les principales méthodes de synchronisation, ainsi que les méthodes de transmission chaotiques les plus utilisées. Par la suite nous présentons le système de communication chaotique que nous allons implémenter. Il se base principalement sur les travaux de Cheng [5] et consiste à la mise en œuvre d'une synchronisation adaptative de deux systèmes chaotiques Unifiés avec différents paramètres qui sont inconnues pour le contrôleur, tout en utilisant un masquage chaotique pour la confidentialité de l'information. Nous terminerons ce chapitre par la présentation de l'implémentation du dit système sous Matlab Simulink, ainsi les résultats de la simulation tout en montrant notre contribution dans l'amélioration des processus de synchronisation et de transmission de notre système.

### **3.2 Concept et classes de la synchronisation chaotique**

Le phénomène de synchronisation peut être décrit comme étant un processus d'ajustement des rythmes des événements répétitifs par l'intermédiaire des faibles interactions. Ce phénomène a été observé pour la première fois par Huygens, en 1673, en étudiant un système de deux pendules couplées. Depuis le constat de Huygens, la synchronisation des systèmes dynamiques a trouvé ses applications en théorie et en pratique, dont plusieurs types de synchronisation ont été distingués [26] : l'auto-synchronisation qui se manifeste par les interactions internes entre les systèmes considérés et la synchronisation commandée qui nécessite une intervention externe pour forcer deux ou plusieurs systèmes dynamiques à se synchroniser. La synchronisation maître-esclave appartient à la catégorie de la synchronisation commandée, pour laquelle on dispose d'un système dominant (le système maître) qui impose son rythme à un second système (le système esclave).

Pendant les deux dernières décennies, la configuration maître-esclave a été appliquée avec succès, dans les systèmes de communication sécurisée par synchronisation chaotique, où l'émetteur chaotique (système maître) génère un signal chaotique incluant le message chiffré qui sera transmis dans le canal de communication vers le récepteur chaotique

(système esclave). Ce dernier a pour rôle de se synchroniser avec l'émetteur et de restaurer ainsi le message chiffré. La synchronisation chaotique consiste donc, à trouver un moyen pour lequel le récepteur réplique le plus fidèlement le signal chaotique de l'émetteur.

Dans les méthodes traditionnelles, la synchronisation est en général basée sur l'utilisation des circuits identiques. Supposons deux systèmes chaotiques identiques oscillant de façon totalement indépendante. Si par un moyen quelconque, on leur permet d'échanger de l'énergie, action que l'on nomme "couplage", les deux systèmes finiront par céder la place à un comportement commun : ils se synchronisent. Il est possible de coupler les systèmes chaotiques dans un sens (couplage unidirectionnel) ou dans les deux sens (couplage bidirectionnel).

À cet effet, la synchronisation chaotique peut s'écrire comme suit [9] :

Considérons les deux systèmes suivants :

$$\begin{cases} \dot{x} = f_1(x) \\ \dot{z} = f_2(z) \end{cases} \quad (3.1)$$

avec  $x, z \in \mathbb{R}$ ,  $f_1$  et  $f_2$  sont des fonctions non linéaires définies de  $\mathbb{R}^n \rightarrow \mathbb{R}^n$ .

Les deux systèmes sont dits synchronisés si :

$$\lim_{t \rightarrow \infty} \|z(t) - x(t)\| = 0 \quad (3.2)$$

où :  $z(t) - x(t)$  représente l'erreur de synchronisation.

### 3.2.1 Synchronisation par couplage unidirectionnel

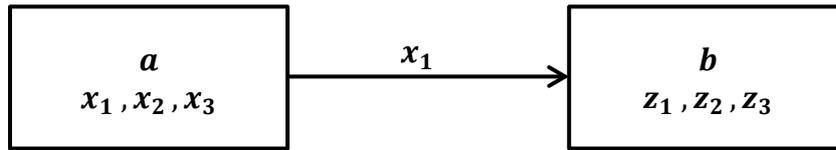
Dans ce cas de couplage, l'énergie est transférée d'un système à l'autre, à l'aide d'un élément de couplage fonctionnant dans un seul sens comme par exemple un amplificateur suiveur. Pour cela une partie du signal d'erreur est injecté dans le système esclave (figure 3.1). Mathématiquement parlant, on peut considérer les deux systèmes suivant [9]:

$$\begin{cases} \dot{x} = f_1(x) \\ \dot{z} = f_2(z) + \alpha(x - z) \end{cases} \quad (3.3)$$

où :  $x, z \in \mathbb{R}$  et  $\alpha = \text{diag}[\alpha_1, \alpha_2, \dots, \alpha_n]^T$ .

Le problème de synchronisation consiste alors à trouver  $\alpha$  qui vérifie l'équation suivante :

$$\lim_{t \rightarrow \infty} \|z(t) - x(t)\| = 0$$

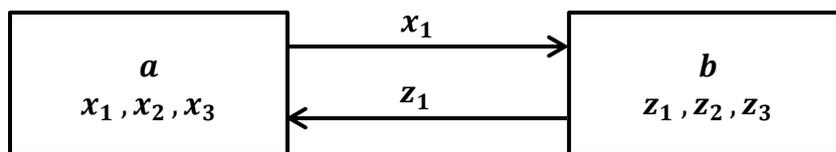


**Figure 3.1 : Couplage unidirectionnel**

Ce type de synchronisation a été appliqué au circuit de Chua [27] et a été également étudié par Koracev et al. dans [28]. Les conditions de convergence de cette approche sont analysées dans [29].

En plus du couplage simple (exemple d’amplificateur suiveur), d'autres méthodes ont été proposées pour la synchronisation des systèmes chaotiques, on peut citer : la méthode par décomposition du système, la synchronisation impulsive, la synchronisation par des méthodes itératives ou la synchronisation par la boucle fermée, etc. 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 [9].

### 3.2.2 Synchronisation par couplage bidirectionnel



**Figure 3.2 : Couplage bidirectionnel**

Contrairement au couplage unidirectionnel, l’élément de couplage dans ce type de couplage permet l’échange de l’énergie dans les deux sens comme par exemple une simple résistance. Pour cela une partie du signal d'erreur est injecté dans les deux systèmes (figure 3.2). Mathématiquement parlant, on peut considérer les deux systèmes suivant [9]:

$$\begin{cases} \dot{x} = f_1(x) + \lambda(z - x) \\ \dot{z} = f_2(z) + \mu(x - z) \end{cases} \quad (3.4)$$

où :  $x, z \in \mathbb{R}$ ,  $\lambda = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_n]^T$  et  $\mu = \text{diag}[\mu_1, \mu_2, \dots, \mu_n]^T$ .

Le problème de synchronisation consiste alors à trouver  $\lambda$  et  $\mu$  qui vérifient l'équation suivante :

$$\lim_{t \rightarrow \infty} \|z(t) - x(t)\| = 0$$

### 3.3 Méthodes de synchronisation chaotique

Plusieurs méthodes de synchronisation ont été proposées dans la littérature. Dans ce qui suit, nous citerons quelques approches en expliquant leurs principes et avantages [8, 11, 30].

#### 3.3.1 Synchronisation identique ou par décomposition du système

Cette méthode introduite par Pecora et Carroll [3, 4] suggèrent qu'il est possible de construire un ensemble de systèmes dynamiques chaotiques tels que leurs signaux communs soient synchronisés. Ils considèrent un système  $S$  de dimension  $n$  de la forme :

$$\dot{x} = f(x(t)) \quad (3.5)$$

où:  $x = [x_1, x_2 \dots \dots, x_n]$  est le vecteur d'état du système  $S$ .

L'idée consiste à décomposer le système d'origine  $S$  en deux sous-systèmes  $S_1$  et  $S_2$  de telle sorte que les variables dynamiques du système d'origine soient réparties de part et d'autre de chacun des sous-systèmes, comme le montre les équations suivantes :

$$\begin{aligned} S_1: \dot{y} &= f_1(y, z) \\ S_2: \dot{z} &= f_2(y, z) \end{aligned} \quad (3.6)$$

avec  $y = [x_1, x_2 \dots \dots, x_m]^T$ ,  $z = [x_{m+1}, x_{m+2} \dots \dots, x_n]^T$  et  $f(t) = \begin{pmatrix} f_1(t) \\ f_2(t) \end{pmatrix}$

Le système  $S$  s'écrit alors sous la forme :

$$\dot{x} = f(x(t)) = \begin{pmatrix} f_1(y, z) \\ f_2(y, z) \end{pmatrix} \quad (3.7)$$

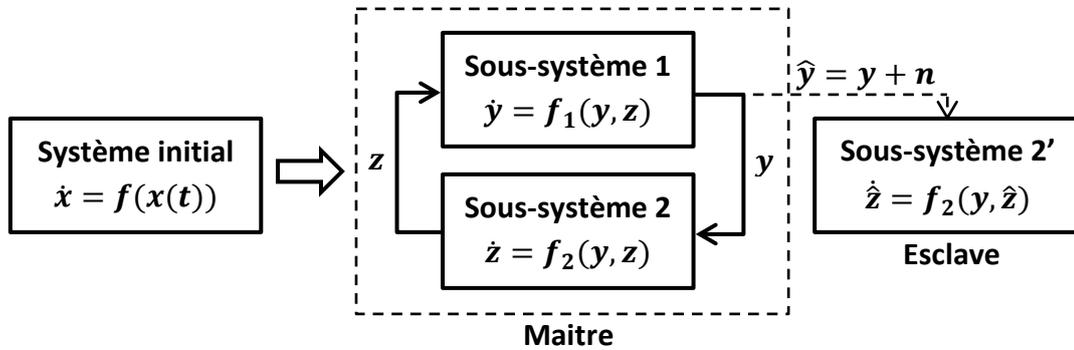
Le système (3.7) est appelé pilote. On considère un système de réponse  $S'_2$  comme étant une copie du sous-système  $S_2$ , de dynamique  $f_2(t)$  et de vecteur d'état  $\hat{z}$ . Le sous-système  $S'_2$  sera piloté par l'état  $y(t)$  du sous-système  $S_1$  et il est défini par l'équation suivante :

$$S'_2: \dot{\hat{z}} = f_2(y, \hat{z}) \quad (3.8)$$

Pecora et Carroll ont démontré que le système  $S'_2$  est candidat pour se synchroniser avec le système initial à la condition nécessaire et suffisante qu'il soit stable. Ceci est équivalent à ce que les exposants de Lyapunov soient négatifs. Le schéma de cette méthode est représenté sur la figure 3.3.

Le système (3.8) se synchronise avec le système (3.7) lorsque :

$$\lim_{t \rightarrow \infty} \|\hat{z}(t) - z(t)\| = 0 \quad (3.9)$$



**Figure 3.3 :** Synchronisation par décomposition du système chaotique

### 3.3.2 Synchronisation généralisée

Cette méthode est une généralisation du concept de synchronisation identique. Les deux systèmes se synchronisent au sens généralisé, s'il existe une transformation  $M$  telle que :

$$\lim_{t \rightarrow \infty} \|z(t) - M(x(t))\| = 0 \quad (3.10)$$

où :  $x(t)$  est le vecteur d'état de l'émetteur et  $z(t)$  est le vecteur d'état du récepteur. Les conditions initiales ne sont pas tenues en compte dans ce cas.

Si  $M$  est inversible, alors  $M^{-1}(z)$  fournit une estimation de l'état  $x$  ; dans le cas contraire, il serait impossible de fournir une estimation de l'état  $x$ . Ceci présente alors un inconvénient majeur pour les techniques de communication utilisant l'état de l'émetteur pour décrypter le message transmis.

### 3.3.3 Synchronisation retardée

Dans la synchronisation retardée, l'état du système esclave converge vers un état décalé dans le temps du système maître :

$$\lim_{t \rightarrow \infty} \|z(t) - x(t - \tau)\| = 0 \quad (3.11)$$

où  $x(t)$  est le vecteur d'état de l'émetteur,  $z(t)$  est le vecteur d'état du récepteur et  $\tau$  est un retard positif.

### 3.3.4 Synchronisation projective

Dans cette méthode, l'état du récepteur converge vers un multiple de l'état décalé dans le temps de l'émetteur :

$$\lim_{t \rightarrow \infty} \|z(t) - \alpha x(t - \tau)\| = 0 \quad (3.12)$$

où  $\alpha$  est le facteur d'échelle,  $x(t)$  est le vecteur d'état de l'émetteur,  $z(t)$  est le vecteur d'état du récepteur et  $\tau$  est un retard positif.

Cette approche est utilisée pour des systèmes partiellement linéaires et permet de synchroniser à un facteur près les états qui ne peuvent être synchronisés.

### 3.3.5 Synchronisation impulsive

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, la synchronisation impulsive a été proposée (figure 3.4).

Le contrôle impulsif d'un système signifie qu'à des moments choisis, les états du système changent soudainement. Dans ce schéma de synchronisation, on considère un système maître de la forme générale suivante :

$$\dot{x}(t) = f(x(t)) \quad (3.13)$$

On définit un signal impulsif qui consiste en une suite d'instantanés discrets auxquels un signal  $y(t) = Cx(t)$  est envoyé par le système maître au système esclave, dont les variables d'état subissent un saut et un changement d'état.

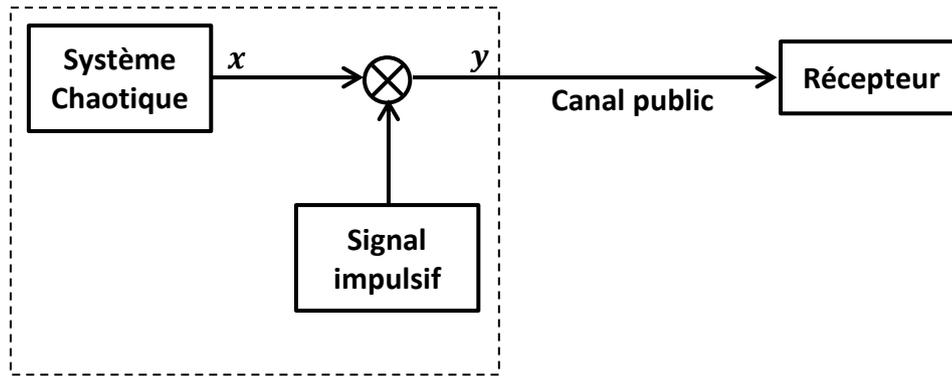


Figure 3.4 : Synchronisation impulsive

### 3.3.6 Synchronisation par boucle fermée

La synchronisation des systèmes chaotiques par les méthodes en boucle ouverte implique une sensibilité aux variations paramétriques. Pour y remédier, de nouvelles techniques basées sur un bouclage par contre-réaction ont été proposées.

L'idée est d'appliquer une correction au système en fonction de l'erreur entre le signal transmis par l'émetteur et le signal régénéré par le récepteur. Cette erreur est ainsi injectée en contre-réaction (figure 3.5) d'où l'appellation de l'approche.

Cette technique permet également la synchronisation entre des paires différentes de systèmes chaotiques.

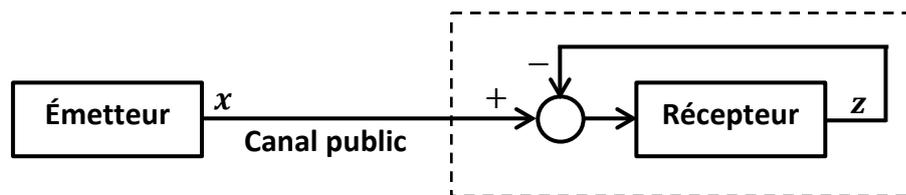


Figure 3.5 : Synchronisation par boucle fermée

### 3.3.7 Synchronisation adaptative

La synchronisation adaptative a pour objectif de synchroniser des systèmes chaotiques identiques et non identiques à état affine dépendant de paramètres inconnus. Elle consiste à construire un contrôleur adaptatif à la réception, qui est un algorithme récursif permettant l'estimation des paramètres inconnus des systèmes chaotiques à synchroniser, tout en contrôlant le système chaotique de réception à suivre le plus fidèlement l'état de

l'émetteur. Ce contrôle se fait par la minimisation de l'erreur entre les états des deux systèmes chaotiques (figure 3.6) selon l'équation suivante :

$$\lim_{t \rightarrow \infty} \|x(t) - z(t)\| = 0 \quad (3.14)$$

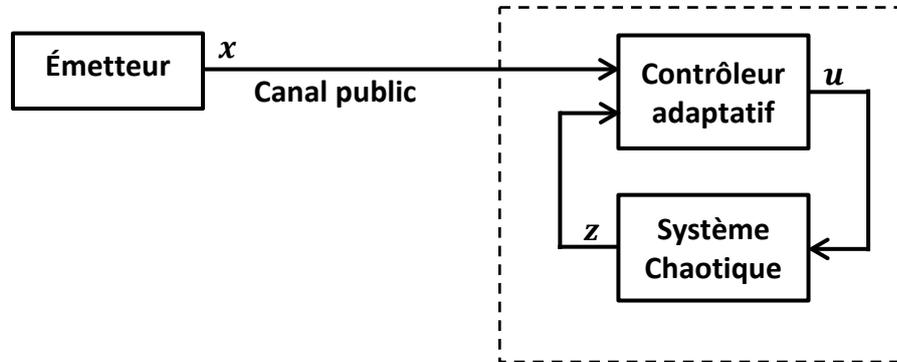


Figure 3.6 : Synchronisation adaptative

### 3.4 Méthodes de transmission chaotique

Il existe plusieurs techniques qui peuvent servir comme moyen de masquage de l'information dans le chaos, nous décrivons ici quelques-uns [9, 11, 31] :

#### 3.4.1 Méthode par addition

Le cryptage par addition, appelé aussi le masquage chaotique, est la technique de la plus simple et la plus élémentaire des techniques de transmission à base du chaos. Le principe de cette méthode est très simple : il suffit d'ajouter le message d'information  $m(t)$  de nature numérique ou analogique avec une porteuse chaotique  $x(t)$  généré par l'émetteur chaotique. Le message chiffré  $s(t)$  ainsi obtenu (figure 3.7) est transmis à travers le canal de transmission vers le récepteur. Au niveau du récepteur et après synchronisation avec l'émetteur par l'une des méthodes de synchronisation précédemment citées, grâce au signal reçu, on récupère le message transmis  $m_r(t)$  par une simple soustraction.

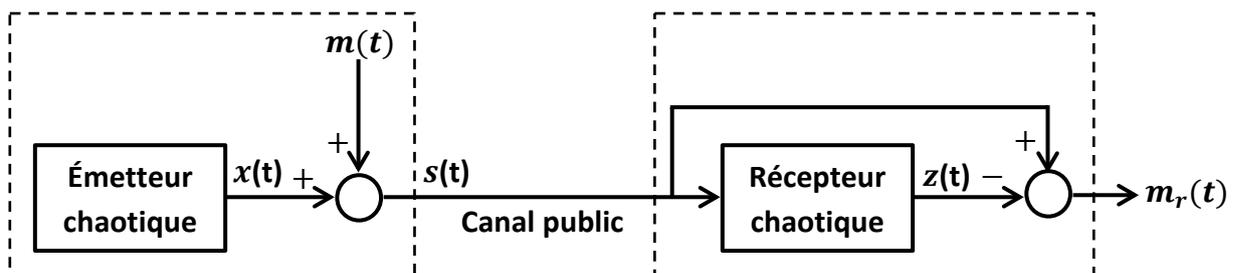


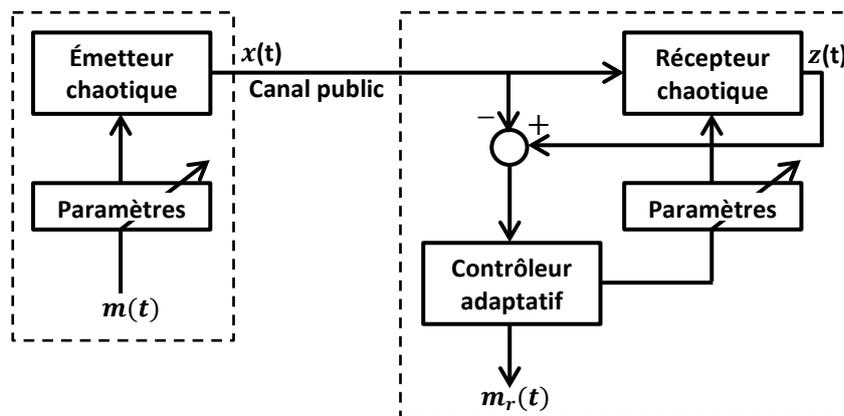
Figure 3.7 : Méthode de cryptage chaotique par addition

Vu la nature aléatoire de la porteuse chaotique, un intrus qui a capté le signal transmis (porteuse chaotique + message utile), ne soupçonnera pas qu'un message est transmis. Alors, il ne cherchera pas à le décrypter.

Le principal avantage de cette méthode réside dans la simplicité du cryptage, ainsi elle peut être utilisée pour transmettre des messages continus ou discrets. Dans les deux cas, il est impératif que l'amplitude du message original soit significativement plus petite que celle de la porteuse chaotique, d'une part pour ne pas perturber l'établissement de la synchronisation au niveau du récepteur, et d'autre part pour garantir le secret de la transmission. Dans tous les cas, à cause de la présence du message, la synchronisation ne peut être parfaite. En outre, la fréquence du message doit être comprise dans le spectre du signal chaotique. Un autre problème qui se pose naturellement concerne la présence d'un bruit additif au niveau du canal de transmission. Dans ce cas, il faut que l'amplitude du message soit plus grande que celle du bruit. Il y a donc un compromis à trouver entre la sécurité de la transmission, et la robustesse au bruit.

### 3.4.2 Méthode par modulation paramétrique

Cette technique utilise le message contenant l'information pour moduler un paramètre ou plusieurs de l'émetteur chaotique. Un contrôleur adaptatif est chargé de maintenir la synchronisation au niveau du récepteur, tout en suivant les changements des paramètres modulés. Le schéma correspondant de cette technique est présenté dans la figure 3.8.



**Figure 3.8 :** Méthode de cryptage chaotique par modulation paramétrique

Au niveau de l'émetteur, le fait de moduler un (ou plusieurs) paramètre(s) impose à la trajectoire de changer continûment d'attracteur, et de ce fait, le signal transmis est plus

complexe qu'un signal chaotique normal. Cependant, la façon d'injecter le message (c'est-à-dire : la fonction de modulation des paramètres) ne doit pas supprimer le caractère chaotique du signal envoyé au récepteur.

Il est important de souligner que cette technique exploite pleinement les qualités des systèmes chaotiques. Elle n'a pas d'équivalent parmi les systèmes de communication classique.

### 3.4.3 Méthode par commutation chaotique

Cette technique (appelée aussi Chaos Shift Keying : CSK) est réservée aux messages prenant un nombre fini de valeurs. L'émetteur est constitué de deux ou plusieurs systèmes chaotiques, selon le nombre de valeurs que le message à transmettre peut prendre. Pour plus de simplicité, prenant le cas des messages binaire :  $m(t) = \{0,1\}$ . Dans ce cas, l'émetteur est constitué de deux systèmes chaotiques  $S_0$  et  $S_1$  : ces deux systèmes peuvent avoir le même modèle dynamique, avec des paramètres différents, ou avoir deux modèles dynamiques totalement différents.

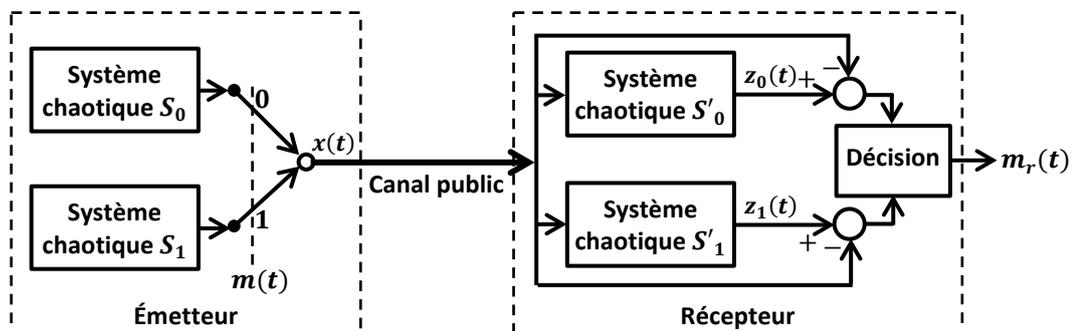


Figure 3.9 : Méthode de cryptage chaotique par commutation

Le principe de fonctionnement de cette technique est illustré dans la figure 3.9 : selon la valeur du message  $m(t)$  à l'instant  $t$ , l'émetteur est soit le système  $S_0$ , soit le système  $S_1$ . Le récepteur est constitué de deux systèmes chaotiques identiques à ceux de l'émetteur  $S'_0$  et  $S'_1$ , plus un bloc de décision. Si  $m(t) = 0$ , alors le système  $S'_0$  peut se synchroniser et le système  $S'_1$  ne peut pas ; ainsi l'erreur de synchronisation  $e_0(t) = z_0(t) - x(t)$  tend vers 0, tandis que l'erreur  $e_1(t) = z_1(t) - x(t)$  sera d'amplitude non nulle. Le processus est symétrique lorsque  $m(t) = 1$ .

La détermination de la valeur exacte du message transmis  $m_r(t)$ , se fait par : l'évaluation de l'erreur de synchronisation au niveau des deux copies comme précédemment, ou bien par une corrélation entre le signal  $x(t)$  reçu et les signaux  $z_0(t)$  et  $z_1(t)$  [9].

Cette technique présente une forte robustesse contre le bruit par rapport aux autres méthodes. En revanche, le taux de transmission est assez bas, ainsi elle est très sensible aux attaques par détection de rupture.

### 3.4.4 Méthode mixte ou combinée

Afin de faire face aux problèmes de sécurité des méthodes précédentes, la technique de cryptage chaotique mixte combine les principes de la cryptographie standard et la synchronisation chaotique. Le message  $m(t)$  contenant l'information est cryptée grâce à une clé  $c(t)$  générée par l'émetteur chaotique.

Le message crypté est alors injecté dans la dynamique du système chaotique, pour la rendre plus complexe. Ensuite, un signal  $y(t)$  fonction des variables d'état de l'émetteur est transmis au récepteur, qui établit une synchronisation avec l'émetteur. La clé est alors reconstruite par le récepteur, qui peut finalement décoder le message. Le principe général de cette méthode est illustré par la figure 3.10.

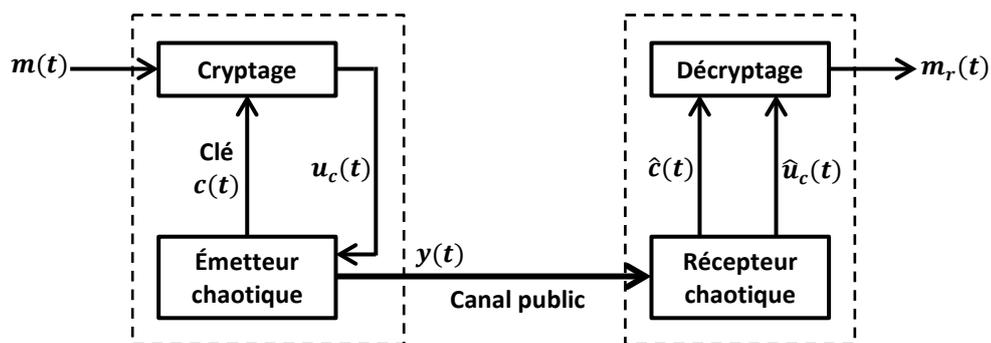
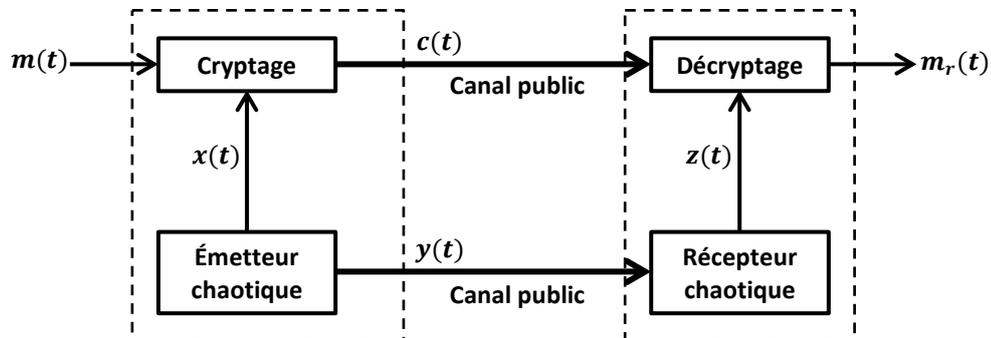


Figure 3.10 : Méthode de cryptage chaotique mixte

### 3.4.5 Méthode par transmission à deux voies

L'idée de base consiste à séparer les tâches de synchronisation et de cryptage en utilisant deux voies de communication. L'émetteur chaotique génère un signal chaotique  $y(t)$  transmis dans un premier canal de communication vers le récepteur qui doit se synchroniser avec le système maître (figure 3.11). L'émetteur génère également un autre

signal chaotique  $x(t)$  utilisé pour une fonction de cryptage qui produit le signal du texte chiffré  $c(t)$  transmis dans un deuxième canal de transmission.



**Figure 3.11 :** Méthode de cryptage chaotique par transmission à deux voies

Grâce à cette indépendance entre les tâches de synchronisation et de cryptage, il n’y a pas de contrainte à imposer sur l’amplitude du signal d’information puisque, ce dernier n’agit ni sur la dynamique de l’émetteur chaotique, ni sur le signal transmis  $y(t)$  contrairement aux autres techniques. De plus, le signal d’information n’a aucune influence sur l’opération de synchronisation qui s’établit d’une façon idéale, ce qui permet de garantir une meilleure qualité du signal récupéré au niveau du récepteur.

D’un autre côté, cette méthode garantit un meilleur niveau de sécurité par rapport aux autres techniques, puisque la séparation entre les opérations de cryptage et de synchronisation permet de concevoir une fonction de cryptage de plus en plus complexe sans détériorer l’aspect chaotique de l’émetteur ou de perdre la synchronisation entre les systèmes maître et esclave. Cependant, cette technique présente de mauvaises performances en présence du bruit de transmission puisque l’effet du bruit est doublé en agissant à la fois sur le signal transmis  $y(t)$  dans la première voie et également sur le message chiffré  $c(t)$  présent dans la deuxième voie de transmission [26].

### 3.5 Transmission sécurisée par synchronisation chaotique adaptative

Dans cette partie, nous allons étudier le système de transmission chaotique que nous avons réalisé, et présenter ainsi les résultats de l'implémentation dudit système sous Matlab Simulink. L'étude de la synchronisation adaptative sera introduite en premier lieu, puis nous présentons la méthode de transmission utilisée pour injecter le message à transmettre. Par la suite nous verrons l'implémentation des différents modules du système de transmission sous Matlab Simulink, ainsi que les résultats de la simulation. À la fin de cette partie, nous présentons l'influence du bruit sur la transmission en fonction du SNR.

#### 3.5.1 Synchronisation adaptative de systèmes chaotiques Unifiés avec paramètres inconnus

Dans cette section, nous allons décrire la méthode de synchronisation chaotique utilisée dans le système de transmission chaotique réalisé (figure 3.12), qui consiste à une synchronisation adaptative entre différents systèmes chaotiques Unifiés (vu dans la section 2.4) avec paramètres inconnus, en se basant sur les travaux de Cheng [5] et Yassen [32, 33]. Cheng a inspiré des travaux de Yassen [33] pour construire son contrôleur adaptatif pour synchroniser deux différents systèmes chaotiques Unifiés, tout en considérant que les paramètres desdits systèmes comme inconnus pour le contrôleur ; par la suite, il a proposé une estimation des paramètres illustrées dans les équations (3.23) et (3.24).

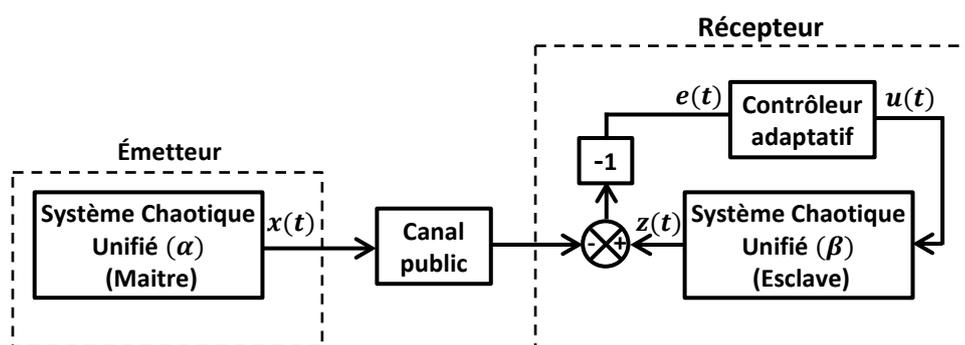


Figure 3.12 : Schéma de la synchronisation adaptative utilisée

Dans notre travail, nous nous sommes inspirés dudit contrôleur de Cheng, tout en ajoutant un terme en fonction des erreurs quadratiques aux équations des estimations des paramètres des systèmes chaotiques Unifiés illustrées dans les équations (3.27) et (3.28).

Dans ce qui suit, nous allons présenter les différentes étapes pour construire ce contrôleur adaptatif.

### a. Contrôleur adaptatif

L'émetteur est un système chaotique Unifié de paramètre  $\alpha \in [0,1]$  et de vecteur d'états est  $x(t) = (x_1, x_2, x_3)^T$ . Il est défini par l'équation suivante :

$$\begin{cases} \dot{x}_1 = (25\alpha + 10)(x_2 - x_1) \\ \dot{x}_2 = (28 - 35\alpha)x_1 - x_1x_3 + (29\alpha - 1)x_2 \\ \dot{x}_3 = x_1x_2 - \frac{\alpha + 8}{3}x_3 \end{cases} \quad (3.15)$$

Le récepteur est un système chaotique Unifié de paramètre  $\beta \in [0,1]$  et de vecteur d'états est  $z(t) = (z_1, z_2, z_3)^T$  qui est piloté par l'entrée de contrôle  $u(t) = (u_1, u_2, u_3)^T$ . Il est défini par l'équation suivante :

$$\begin{cases} \dot{z}_1 = (25\beta + 10)(z_2 - z_1) + u_1 \\ \dot{z}_2 = (28 - 35\beta)z_1 - z_1z_3 + (29\beta - 1)z_2 + u_2 \\ \dot{z}_3 = z_1z_2 - \frac{\beta + 8}{3}z_3 + u_3 \end{cases} \quad (3.16)$$

**N.B :**  $u(t) = (u_1, u_2, u_3)^T$  représente le contrôleur adaptatif qu'on doit construire.

Pour qu'on puisse avoir la synchronisation entre les deux systèmes chaotiques, il faut que l'erreur entre les variables d'états de l'émetteur et du récepteur  $e_i = z_i - x_i$  ( $i = 1, 2, 3$ ) tend vers 0 lorsque  $t \rightarrow \infty$ . Ainsi, l'erreur dynamique entre les systèmes (3.15) et (3.16) défini par  $\dot{e}_i = \dot{z}_i - \dot{x}_i$  ( $i = 1, 2, 3$ ), peut s'écrire comme suit :

$$\begin{cases} \dot{e}_1 = 10(e_2 - e_1) - 25\alpha(x_2 - x_1) + 25\beta(z_2 - z_1) + u_1 \\ \dot{e}_2 = 28e_1 - e_2 + 35\alpha x_1 - 35\beta z_1 - 29\alpha x_2 + 29\beta z_2 + x_1x_3 - z_1z_3 + u_2 \\ \dot{e}_3 = -\frac{8}{3}e_3 + \frac{\alpha}{3}x_3 - \frac{\beta}{3}z_3 - x_1x_2 + z_1z_2 + u_3 \end{cases} \quad (3.17)$$

Supposant que les paramètres soient parfaitement connus et d'après la méthode du contrôle actif [32, 33] qui exige que le contrôleur soit la partie non linéaire de l'erreur dynamique (équation (3.17)), le contrôleur adaptatif  $u(t)$  peut-être choisi comme suit :

$$\begin{cases} u_1 = -10e_2 + 25\alpha(x_2 - x_1) - 25\beta(z_2 - z_1) \\ u_2 = -28e_1 - 35\alpha x_1 + 35\beta z_1 + 29\alpha x_2 - 29\beta z_2 - x_1x_3 + z_1z_3 \\ u_3 = -\frac{\alpha}{3}x_3 + \frac{\beta}{3}z_3 + x_1x_2 - z_1z_2 \end{cases} \quad (3.18)$$

En remplaçant l'équation (3.18) dans l'équation (3.17), on obtient l'équation suivante :

$$\begin{cases} \dot{e}_1 = -10e_1 \\ \dot{e}_2 = -e_2 \\ \dot{e}_3 = -\frac{8}{3}e_3 \end{cases} \quad (3.19)$$

Le système d'équations (3.19) contient trois valeurs propres négatives  $(-10, -1, -8/3)$ , cela indique que l'erreur  $e(t)$  converge vers 0 lorsque  $t \rightarrow \infty$ , alors la synchronisation entre les deux systèmes chaotiques est atteinte.

Le contrôleur adaptatif défini par l'équation (3.18) est basé sur des paramètres qui sont supposés être parfaitement connus. Alors, dans notre cas, le contrôleur doit permettre la synchronisation entre deux systèmes chaotiques unifiés avec paramètres inconnus. Pour cela, nous allons considérer que les paramètres  $\alpha$  et  $\beta$  dans l'équation de l'erreur dynamique (équation (3.17)) comme inconnus ; ainsi, nous allons les remplacer par leurs valeurs estimées  $\hat{\alpha}$  et  $\hat{\beta}$  dans les équations (3.17) et (3.18). Le nouveau contrôleur adaptatif s'écrit comme suit :

$$\begin{cases} u_1 = -10e_2 + 25\hat{\alpha}(x_2 - x_1) - 25\hat{\beta}(z_2 - z_1) \\ u_2 = -28e_1 - 35\hat{\alpha}x_1 + 35\hat{\beta}z_1 + 29\hat{\alpha}x_2 - 29\hat{\beta}z_2 - x_1x_3 + z_1z_3 \\ u_3 = -\frac{\hat{\alpha}}{3}x_3 + \frac{\hat{\beta}}{3}z_3 + x_1x_2 - z_1z_2 \end{cases} \quad (3.20)$$

En remplaçant l'équation (3.20) dans l'équation (3.17), l'équation de l'erreur dynamique entre le récepteur et l'émetteur s'écrit comme suit :

$$\begin{cases} \dot{e}_1 = -10e_1 - 25(\alpha - \hat{\alpha})(x_2 - x_1) + 25(\beta - \hat{\beta})(z_2 - z_1) \\ \dot{e}_2 = -e_2 + 35(\alpha - \hat{\alpha})x_1 - 35(\beta - \hat{\beta})z_1 - 29(\alpha - \hat{\alpha})x_2 + 29(\beta - \hat{\beta})z_2 \\ \dot{e}_3 = -\frac{8}{3}e_3 + \frac{(\alpha - \hat{\alpha})}{3}x_3 - \frac{(\beta - \hat{\beta})}{3}z_3 \end{cases} \quad (3.21)$$

### ***b. Estimation des paramètres proposée par Cheng [5]***

Le contrôleur adaptatif est maintenant construit et donné par l'équation (3.20) ; il reste maintenant à déterminer les formules des paramètres estimés  $\hat{\alpha}$  et  $\hat{\beta}$ , tout en assurant que l'erreur  $e(t)$  défini par le système (3.21) converge vers 0 lorsque  $t \rightarrow \infty$ . Pour cela, nous allons utiliser la méthode directe de Lyapunov (vu dans la section 1.5.3). Nous allons procéder comme suit :

Considérant la fonction de Lyapunov  $V$  donnée par l'équation suivante :

$$V = \frac{1}{2}e^T e + \frac{1}{2}\tilde{\alpha}^2 + \frac{1}{2}\tilde{\beta}^2 \quad (3.22)$$

où  $\tilde{\alpha} = \alpha - \hat{\alpha}$  et  $\tilde{\beta} = \beta - \hat{\beta}$ .

Selon la méthode de stabilité directe de Lyapunov, le système (3.21) est dit stable au sens de Lyapunov si  $\dot{V} \leq 0$ . Calculant la dérivé  $\dot{V}$  de la fonction de Lyapunov donnée par l'équation (3.22) :

$$\begin{aligned} \dot{V} &= e^T \dot{e} + \tilde{\alpha}(-\dot{\hat{\alpha}}) + \tilde{\beta}(-\dot{\hat{\beta}}) \\ \dot{V} &= -10e_1^2 - e_2^2 - \frac{8}{3}e_3^2 - \tilde{\alpha} \left[ 25(x_2 - x_1)e_1 - 35x_1e_2 + 29x_2e_2 - \frac{1}{3}x_3e_3 + \dot{\hat{\alpha}} \right] \\ &\quad - \tilde{\beta} \left[ -25(z_2 - z_1)e_1 + 35z_1e_2 - 29z_2e_2 + \frac{1}{3}z_3e_3 + \dot{\hat{\beta}} \right] \end{aligned} \quad (3.23)$$

Si les quantités entre crochets sont nulle, on a  $\dot{V} \leq 0$ . Ainsi, l'erreur  $e(t)$  converge vers 0 lorsque  $t \rightarrow \infty$ , alors les paramètres estimés  $\hat{\alpha}$  et  $\hat{\beta}$  peuvent s'écrire comme suit [5] :

$$\begin{cases} \dot{\hat{\alpha}} = -25(x_2 - x_1)e_1 + (35x_1 - 29x_2)e_2 + \frac{1}{3}x_3e_3 \\ \dot{\hat{\beta}} = 25(z_2 - z_1)e_1 - (35z_1 - 29z_2)e_2 - \frac{1}{3}z_3e_3 \end{cases} \quad (3.24)$$

$$\quad (3.25)$$

### c. Estimation des paramètres proposée dans ce travail

En remplaçant les paramètres estimés de la section précédente (équations (3.24) et (3.25)) dans la dérivé de la Fonction de Lyapunov (équation (3.23)), on aura l'équation suivante :

$$\dot{V} = -(10e_1^2 + e_2^2 + \frac{8}{3}e_3^2) \quad (3.26)$$

L'équation (3.26) montre que la convergence de l'erreur  $e(t)$  est en fonction des états quadratiques de l'erreur, pondérées par des coefficients (10,1,8/3). Notre idée est de faire de sorte de rapprocher cette convergence à tendre vers un multiple du carré de la norme de l'erreur  $e(t)$  (c'est-à-dire :  $\dot{V} \cong -(e_1^2 + e_2^2 + e_3^2) \times Cste$ ). Pour cela nous avons ajouté un terme aux paramètres estimés  $\hat{\alpha}$  et  $\hat{\beta}$  comme le montre les équations suivantes :

$$\begin{cases} \dot{\hat{\alpha}} = -25(x_2 - x_1)e_1 + (35x_1 - 29x_2)e_2 + \frac{1}{3}x_3e_3 + (1 - \hat{\alpha})\left(\frac{9}{2}e_2^2 + \frac{11}{3}e_3^2\right) & (3.27) \\ \dot{\hat{\beta}} = 25(z_2 - z_1)e_1 - (35z_1 - 29z_2)e_2 - \frac{1}{3}z_3e_3 + (1 - \hat{\beta})\left(\frac{9}{2}e_2^2 + \frac{11}{3}e_3^2\right) & (3.28) \end{cases}$$

### 3.5.2 Méthode de transmission utilisée

Dans cette section, nous présentons la méthode d'insertion du message informatif  $m(t)$  dans le système de transmission qu'on a utilisé, ainsi que la méthode de récupération de ce message à la réception.

La méthode de cryptage utilisée dans le système de transmission chaotique est le masquage chaotique additif. La figure 3.13 présente le schéma global de la transmission chaotique utilisée.

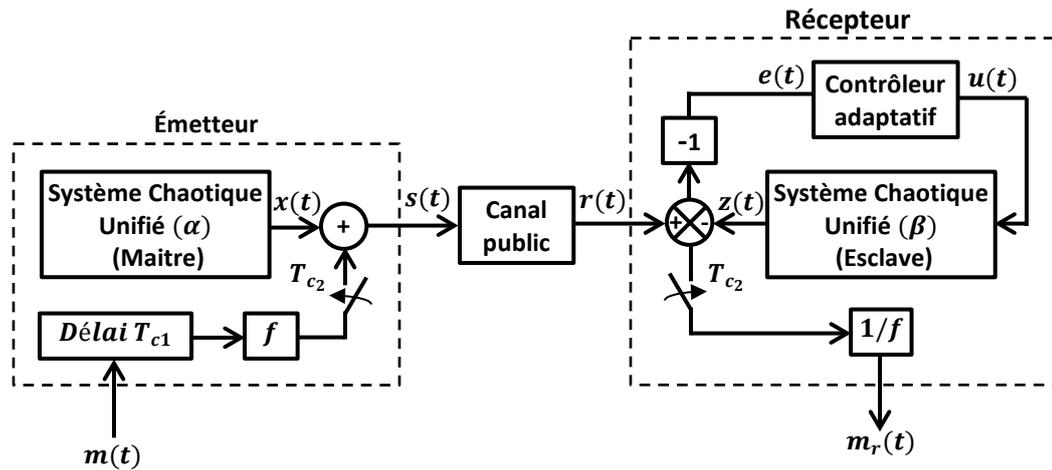


Figure 3.13 : Schéma de la transmission chaotique proposée

Initialement, le système se synchronise après un temps  $T_{S_1}$  ; à cet effet, l'émetteur commence la collection des échantillons du message  $m(t)$  à transmettre après un temps  $T_{c_1} > T_{S_1}$ . Chaque transmission d'un échantillon conduira le système vers une désynchronisation, d'où la nécessité d'une resynchronisation du système. Sachant que le système nécessite un temps  $T_{S_2}$  pour se resynchroniser et pour ne pas perdre totalement la synchronisation en transmettant en continu, nous allons transmettre un échantillon chaque période  $T_{c_2} > T_{S_2}$ .

En utilisant le masquage chaotique additif, on aura deux contraintes à respecter sur l'amplitude du signal informatif à transmettre, qui doit être significativement plus petite que l'amplitude du signal chaotique (porteuse) et en même temps plus grande que

l'amplitude du bruit du canal de transmission. Pour cela nous allons multiplier le signal informatif à transmettre par un facteur de réduction  $f$  qui doit être choisi judicieusement pour respecter les contraintes du masquage chaotique additif d'un côté et pour ne pas trop perturber la synchronisation du système d'un autre côté. Dans notre système, nous avons choisi le facteur  $f$  de telle façon à normaliser le signal à transmettre ( $\max(f \times m(t)) \leq 1$ ).

À ce niveau, le signal résultant (message + porteuse chaotique) à transmettre peut s'écrire sous la forme suivante :

$$s(t) = f \times m(t) + x(t) \quad (3.29)$$

Pour reconstituer le message  $m_r(t)$  à la réception, l'opération inverse de celle de l'émission est appliquée. La partie responsable de la reconstitution du message informatif au niveau du récepteur chaotique et après un retard  $T_{c_1}$  par rapport au début de l'émission, collecte à chaque période  $T_{c_2}$  un échantillon du signal total reçu. Par la suite, le message  $m_r(t)$  sera reconstitué comme suit :

À la réception l'équation de réception du message informatif peut s'écrire comme suit :

$$m_r(t) = (r(t) - z(t)) \times (1/f) \quad (3.30)$$

Avec  $r(t) = s(t) + n(t)$ , où  $n(t)$  est le bruit du canal de transmission.

Pour un faible bruit de transmission on a :  $r(t) \cong s(t)$ . L'équation (3.30) devient :

$$m_r(t) \cong (s(t) - z(t)) \times (1/f) \quad (3.31)$$

Après un retard  $T_{c_1}$  du début de processus de synchronisation et à chaque instant  $t = T_{c_1} + i \times T_{c_2} - dt$  (avec  $i = 1, 2, \dots$ ) dont la synchronisation est déjà établie, on a :  $e(t)$  tend vers 0  $\Rightarrow x(t) \cong z(t)$ . Aux instants  $t = T_{c_1} + i \times T_{c_2}$  dont on émet les échantillons du message  $m(t)$ , on aura toujours  $x(t) \cong z(t)$ . Alors, en remplaçant de l'équation (3.29) dans l'équation (3.31), on aura :

$$\begin{aligned} m_r(t) &\cong (f \times m(t) + x(t) - z(t)) \times \left(\frac{1}{f}\right) \\ &\cong (f \times m(t)) \times \left(\frac{1}{f}\right) \\ &\cong m(t) \end{aligned}$$

Le fait d'utilisé dans la transmission, en plus des systèmes chaotiques, le facteur  $f$ , le retard de transmission  $T_{c_1}$  et la période d'émission-réception  $T_{c_2}$ , ne fait qu'augmenter la sécurité de l'information. En effet, l'utilisation d'un facteur  $f$  adéquat va masquer notre message  $m(t)$  au point à qu'il passera pour un bruit s'il y aura une capture du signal. En plus, si on loupe à la réception un seul pas de temps ( $dt$ ) pour le retard  $T_{c_1}$  ou pour la période  $T_{c_2}$ , alors on perd totalement notre message. Pour cela, la structure chaotique de notre système, le facteur  $f$ , le retard de transmission  $T_{c_1}$  et la période d'émission-réception  $T_{c_2}$  constituent la clé de chiffrement de notre système de transmission chaotique réalisé.

### 3.5.3 Implémentation du système de transmission sous Matlab Simulink

Cette section consiste à présenter les schémas des différents modules utilisés dans l'implémentation du système de transmission chaotique utilisé sous Matlab Simulink.

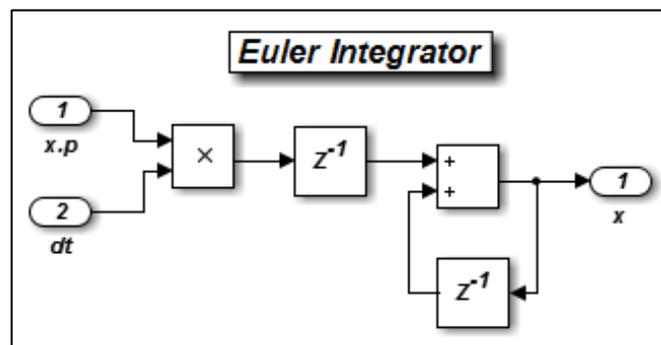
#### a. Intégrateur d'Euler

Pour préparer la route à l'implémentation FPGA du système de transmission chaotique qu'on va décrire dans le chapitre suivant, qui nécessite l'implémentation d'une méthode d'intégration discrète, nous n'avons pas utilisé les blocs intégrateurs de Matlab Simulink pour réaliser les différentes intégrations. Cependant, nous avons choisi d'implémenter la méthode d'intégration numérique d'Euler, qui est définit pour une équation différentielle de type  $\frac{dx}{dt} = f(x, t)$  comme suit [34] :

$$x_{n+1} = x_n + dt \times f(x_n, n) \quad (3.32)$$

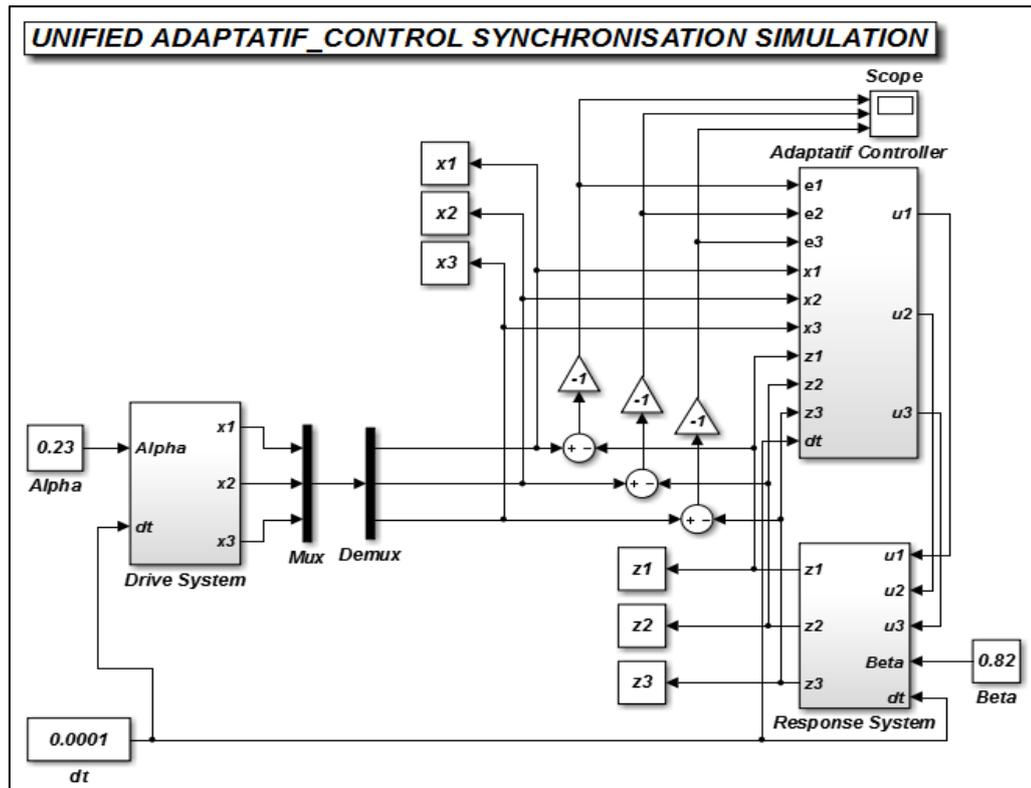
Sachant que :  $x_n$  représente la fonction  $x$  à l'instant  $t$ ,  $x_{n+1}$  représente la fonction  $x$  à l'instant  $t + dt$  et  $dt$  représente le pas d'intégration.

La figure 3.14 illustre l'implémentation de l'équation (3.32) sous Matlab Simulink.



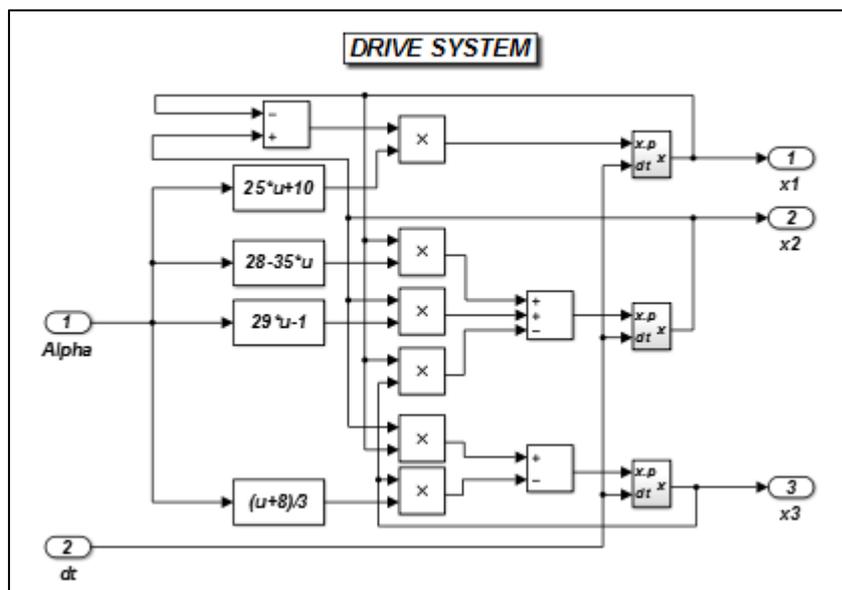
**Figure 3.14** : Implémentation de l'intégrateur d'Euler sous Matlab Simulink

**b. Synchronisation adaptative**



**Figure 3.15 :** Schéma global de l'implémentation de la synchronisation adaptative sous Matlab Simulink

**c. Émetteur chaotique (Système maître)**



**Figure 3.16 :** Implémentation du système chaotique Unifié de l'émetteur (système maître) sous Matlab Simulink

d. Récepteur chaotique (Système esclave)

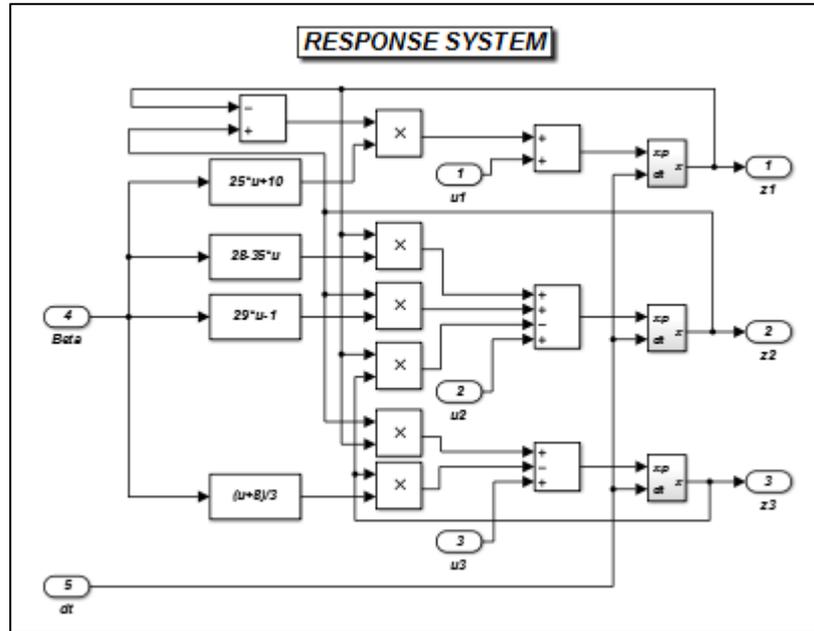


Figure 3.17 : Implémenté du système chaotique Unifié du récepteur (système esclave) sous Matlab Simulink

e. Contrôleur adaptatif

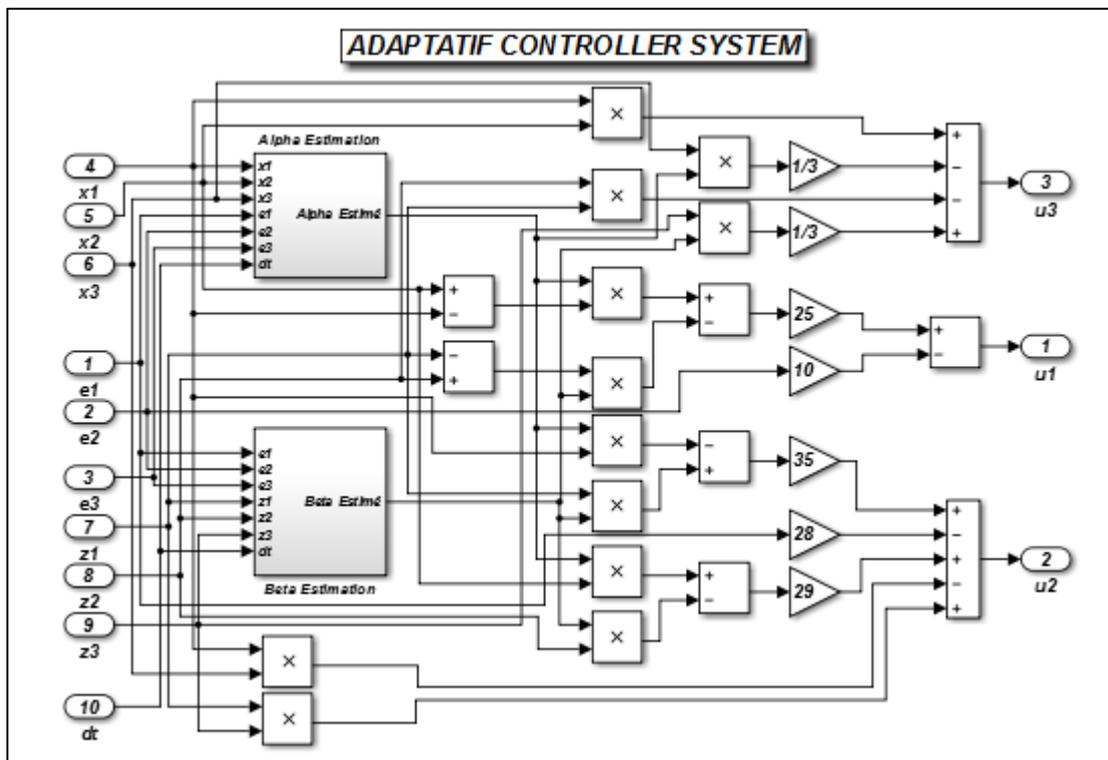
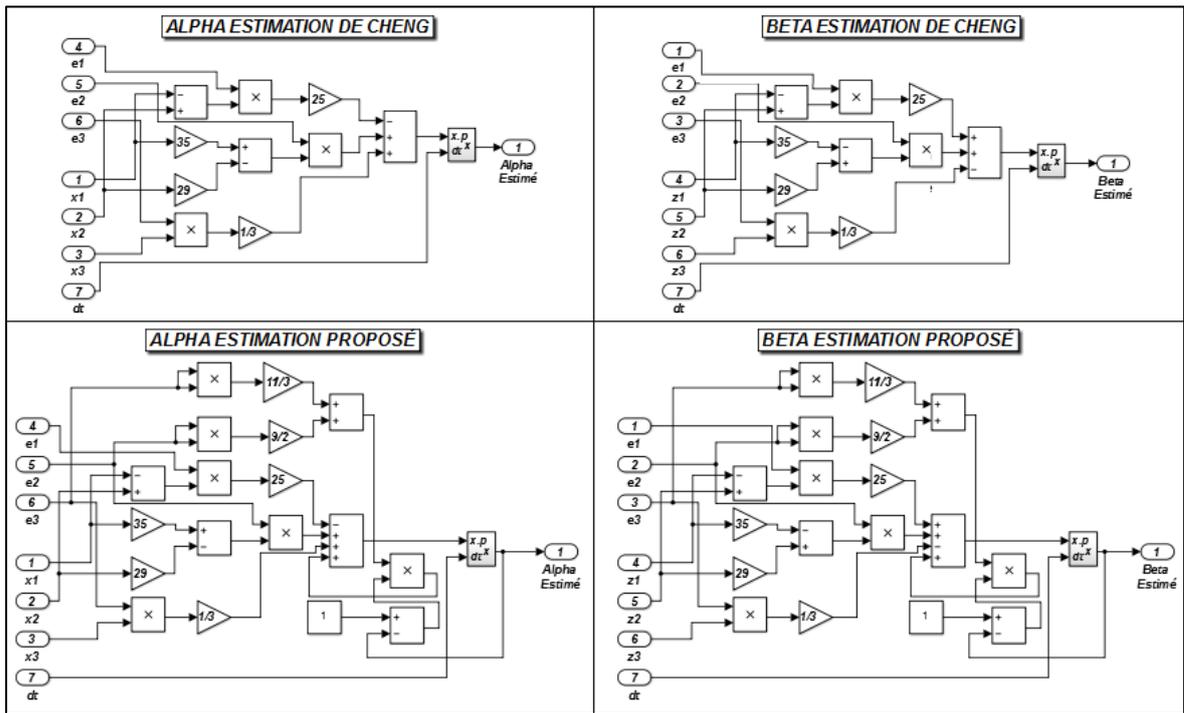


Figure 3.18 : Implémenté du contrôleur adaptatif sous Matlab Simulink

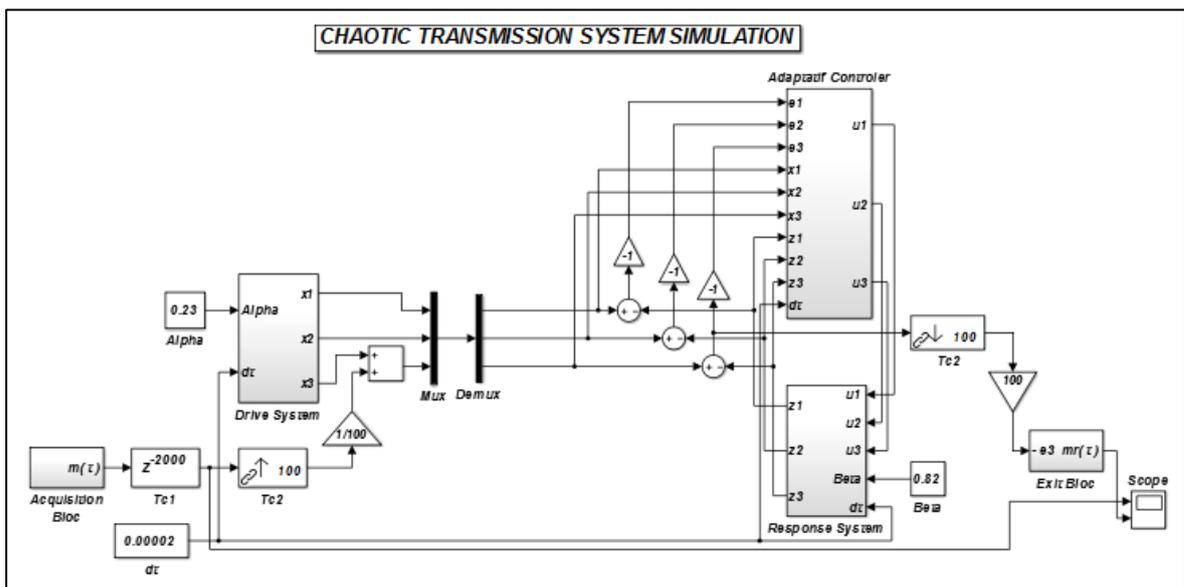
**f. Estimation des paramètres  $\hat{\alpha}$  et  $\hat{\beta}$**



**Figure 3.19 :** Implémentation des estimateurs de paramètres  $\hat{\alpha}$  et  $\hat{\beta}$  des modèles de Cheng et proposé sous Matlab Simulink

**g. Système de transmission**

Nous avons pris comme porteuse du signal informatif la variable d'état  $x_3$  du système chaotique Unifié de l'émetteur. La figure 3.20 illustre le principe général de l'implémentation du système de transmission chaotique implémenté sous Matlab Simulink.



**Figure 3.20 :** Schéma global de l'implémentation du système de transmission chaotique sous Matlab Simulink

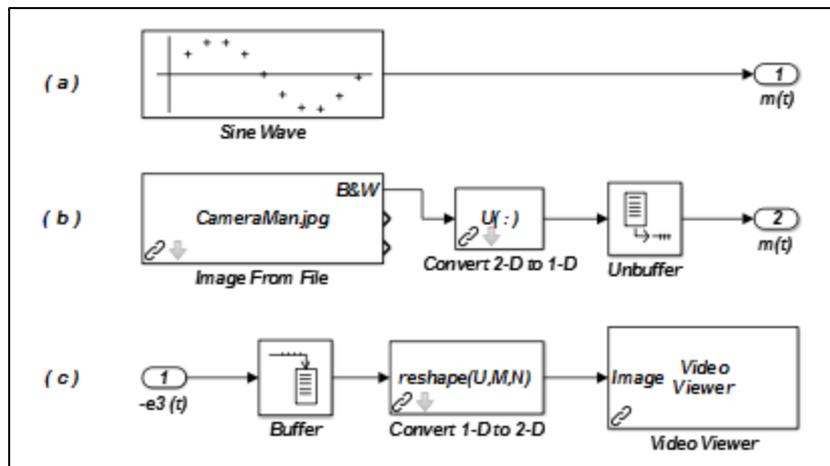
L'injection du message dans l'émetteur du système de transmission chaotique se fait après un retard  $T_{c1}$ .

Pour permettre à l'émetteur chaotique de collecter un échantillon chaque période  $T_{c2}$  (vu dans la section 3.5.2), nous avons utilisé le bloc sur-échantillonneur (UPSample) de Matlab Simulink ; par contre, à la réception nous avons utilisé le bloc sous-échantillonneur (DownSample) de Matlab Simulink.

#### ***h. Blocs d'acquisitions et de sorties de la sinusoïde et de l'image***

Afin de valider le bon fonctionnement de notre système de transmission chaotique, nous avons procédé à l'émission de deux types de signaux : une sinusoïde et une image de 256 niveaux de gris. Pour la transmission de la sinusoïde, nous avons simplement utilisé le bloc "Sine Wave" du Matlab Simulink. Cependant pour la transmission de l'image, nous avons effectué un prétraitement de cette image qui consiste après l'acquisition de l'image, à convertir la matrice de l'image de dimension 2 ( $n \times m$ ) vers un seul vecteur de dimension 1 ( $n \times m$ ) qui est sérialisé pour être finalement injecter dans le système de transmission chaotique.

À la réception, la récupération du signal sinusoïdal se fait directement en branchant un oscilloscope à la sortie du récepteur chaotique. Pour récupérer l'image à la sortie du récepteur chaotique, nous avons effectué l'opération inverse de celle de l'émission. La figure 3.21 montre l'implémentation des blocs d'acquisitions de l'image et de la sinusoïde au niveau de l'émetteur, ainsi l'implémentation du bloc de récupération de l'image au niveau du récepteur chaotique.



**Figure 3.21 :** Implémentation des différents blocs d'acquisition sous Matlab Simulink  
 (a) Acquisition de sinusoïde - (b) Acquisition d'image - (c) Reconstitution d'image

### 3.5.4 Simulation et résultats

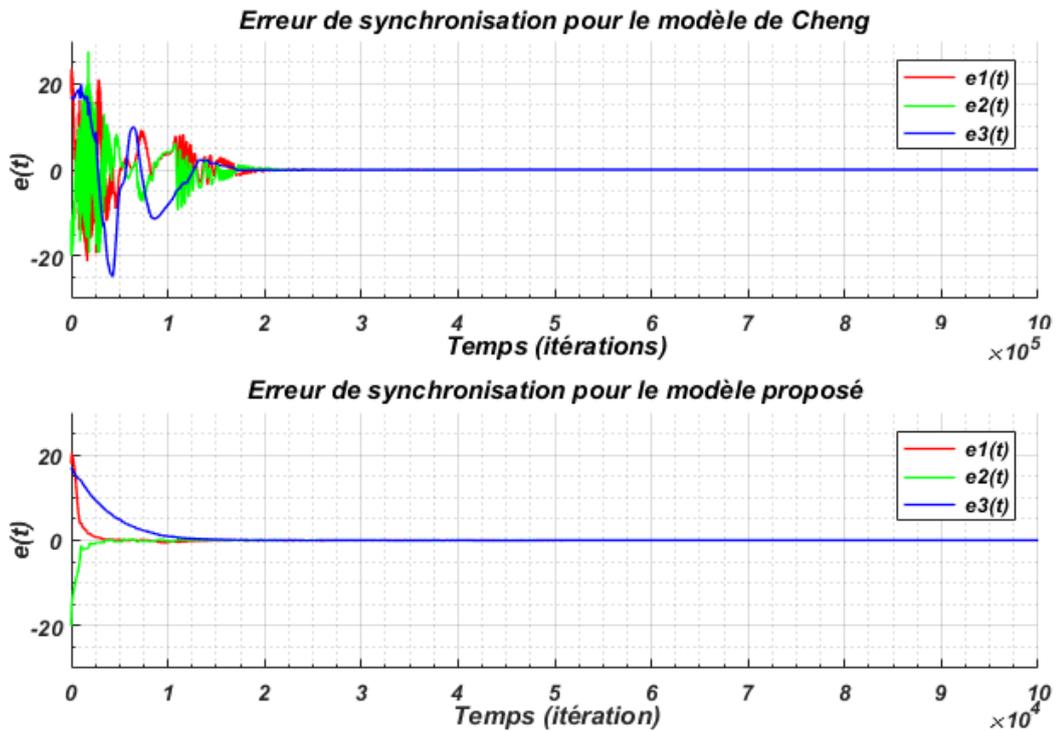
Cette partie consiste à présenter les différents résultats de la simulation de l'implémentation du système de transmission chaotique réalisé sous Matlab Simulink, avec une amélioration de ce système en se basant sur une analyse détaillée de ces résultats. Nous commençons par présenter l'erreur de synchronisation  $e(t)$  entre les états du récepteur chaotique et de l'émetteur chaotique des deux modèles précédemment cités : le modèle de Cheng et le modèle proposé. Une comparaison entre lesdits modèles est faite. Par la suite et uniquement pour le modèle proposé, nous présentons l'évolution temporelle des états du récepteur chaotique et de l'émetteur chaotique et des paramètres estimés  $\hat{\alpha}$  et  $\hat{\beta}$ .

Pour la transmission chaotique, nous présentons les résultats de transmission d'un signal sinusoïdal et de quelques images. Par la suite, une analyse sur l'influence du paramètre de transmission  $T_{c_2}$  sur la qualité de transmission, tout en présentant une technique pratique basée sur cette analyse, utilisée pour améliorer la qualité de transmission tout en diminuant au maximum le paramètre  $T_{c_2}$ .

Les résultats présentés dans cette partie sont obtenus pour les paramètres suivants :  $\alpha = 0.82$  ,  $\beta = 0.23$  ,  $X_0 = (-15, 10, 3)$  ,  $Z_0 = (3, -10, 20)$  ,  $\hat{\alpha}_0 = \hat{\beta}_0 = \mathbf{0}$ . Ces paramètres sont choisis de manière à simuler les extrêmes conditions du système de transmission (émetteur appartient aux systèmes généralisés de Chen ; récepteur appartient aux systèmes généralisés de Lorenz ; conditions initiales espacées). Il est à noter que le système de transmission chaotique proposé présente un bon fonctionnement pour n'importe quelles valeurs  $\alpha$  ,  $\beta$  ,  $X_0$  et  $Z_0$ .

#### ***a. Erreur de synchronisation (Comparaison modèle de Cheng-modèle proposé)***

Afin de faire une comparaison entre le modèle de Cheng et le modèle proposé, nous avons décidé à chercher pour chaque modèle, le pas d'intégration maximal  $dt_{max}$  qui nous conduira vers la synchronisation. Pour cela et pour les paramètres cités, nous avons effectué plusieurs simulations de 10 s, tout en variant le pas d'intégration  $dt$ . Le résultat était que la synchronisation pour le modèle de Cheng s'établie pour un pas d'intégration  $dt \leq 10^{-5}$ . Tandis que pour le modèle proposé, la synchronisation s'établie pour un pas d'intégration  $dt \leq 10^{-4}$ .



**Figure 3.22 :** Erreurs de synchronisation pour le modèle de Cheng et le modèle proposé

La figure 3.22 illustre l’erreur de synchronisation de chaque modèle en fonction de nombre d’itérations nécessaires pour avoir un temps de simulation de 10 s. Elle montre que les deux modèles atteignent la synchronisation, sauf que la dynamique du modèle proposé durant la synchronisation est plus stable que pour le modèle de Cheng (c’est-à-dire : il y a moins de fluctuations). Ainsi, le modèle proposé atteint la synchronisation en  $2 \cdot 10^4$  itérations, tandis que le modèle de Cheng l’atteint en  $2 \cdot 10^5$  itérations ; ce gain en nombre d’itération (gain en calcul) sera très utile dans une réalisation pratique à savoir : une implémentation FPGA, comme il peut diminuer considérablement les temps  $T_{c_1}$  et  $T_{c_2}$ , et par conséquent augmenter la fréquence (le débit) de transmission.

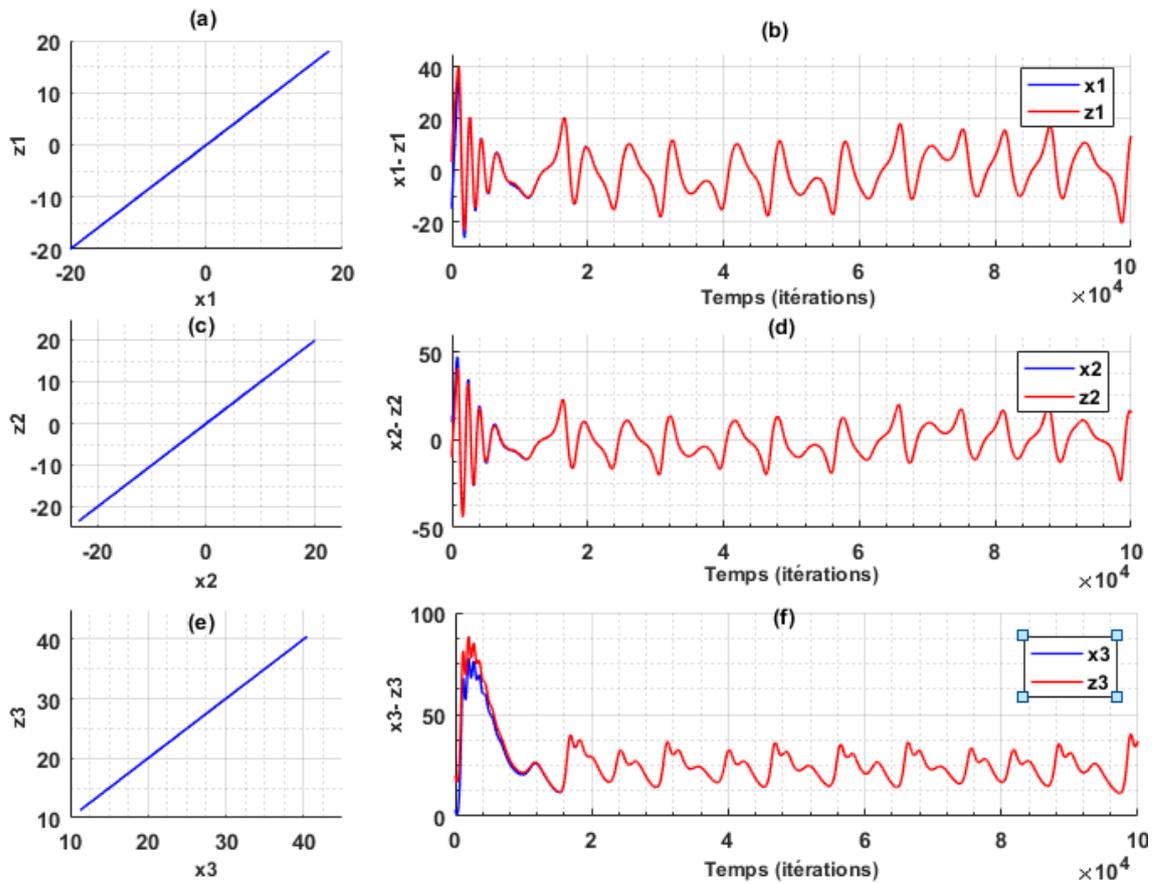
Pour le même pas d’intégration ( $dt = 10^{-5}$ ) et pour les mêmes conditions citées, nous avons calculé l’erreur quadratique moyenne (exemple donné pour  $e_3(t)$ ) après l’établissement de la synchronisation. Le résultat était que le modèle proposé ( $EQM(e_3) = 2.78 \times 10^{-7}$ ) converge mieux vers 0 que le modèle de Cheng ( $EQM(e_3) = 4.6 \times 10^{-3}$ ).

De ce qui précède, nous pouvons conclure que les termes ajoutés (équations (3.27) et (3.28)) aux équations des paramètres estimés proposé par Cheng [5] (équations (3.23)

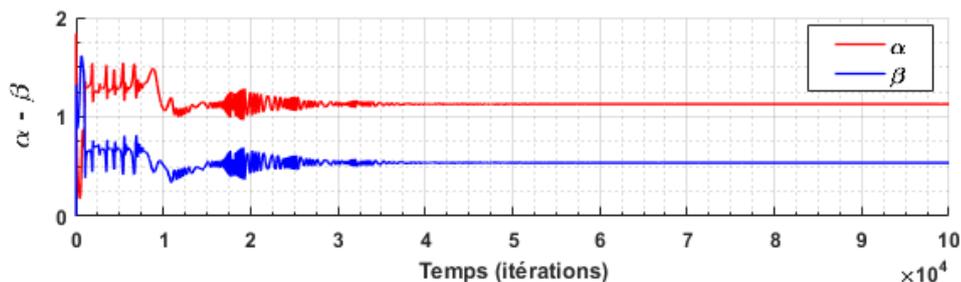
et (3.24)), ont amélioré la qualité du processus de synchronisation de notre système chaotique.

**b. Évolution temporelle des variables d'états et des paramètres estimés**

La Figure 3.23 présente les réponses temporelles des états  $x_i$  et  $z_i$  ainsi leurs représentations dans l'espace de phases  $X_iZ_i$  après synchronisation avec  $dt = 10^{-4}$ . On peut bien constater que le récepteur chaotique suit le plus fidèlement l'émetteur chaotique. La Figure 3.24 représente les paramètres estimés  $\hat{\alpha}$  et  $\hat{\beta}$  du modèle proposé.



**Figure 3.23 :** Représentation des états de l'émetteur et du récepteur chaotiques du modèle proposé pour  $dt = 10^{-4}$ , avec : (a) - (c) - (e) sont les plans des phases  $X_iZ_i$  après synchronisation et (b) - (d) - (f) sont les réponses temporelles des états  $x_i, z_i$



**Figure 3.24 :** Les paramètres estimés  $\hat{\alpha}$  et  $\hat{\beta}$  du modèle proposé pour  $dt = 10^{-4}$

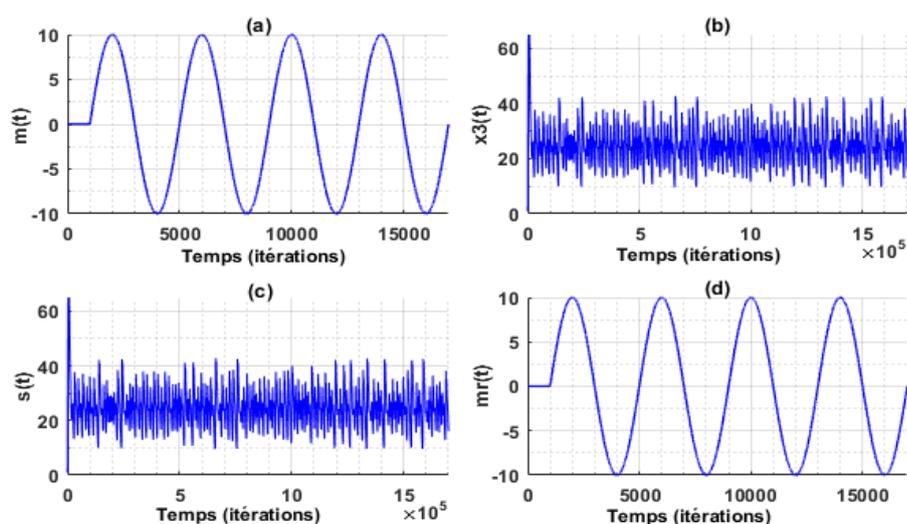
### c. Transmission chaotique

Les résultats de la simulation de cette section sont obtenus à partir des paramètres suivants :  $dt = 5 \cdot 10^{-5}$ ,  $T_{c_1} = 10^5$  échantillons,  $T_{c_2} = 100$  échantillons. Ces paramètres sont choisis après une série de tests, dont on a pris les valeurs optimales qui donnent un meilleur rapport qualité de transmission / temps de transmission.

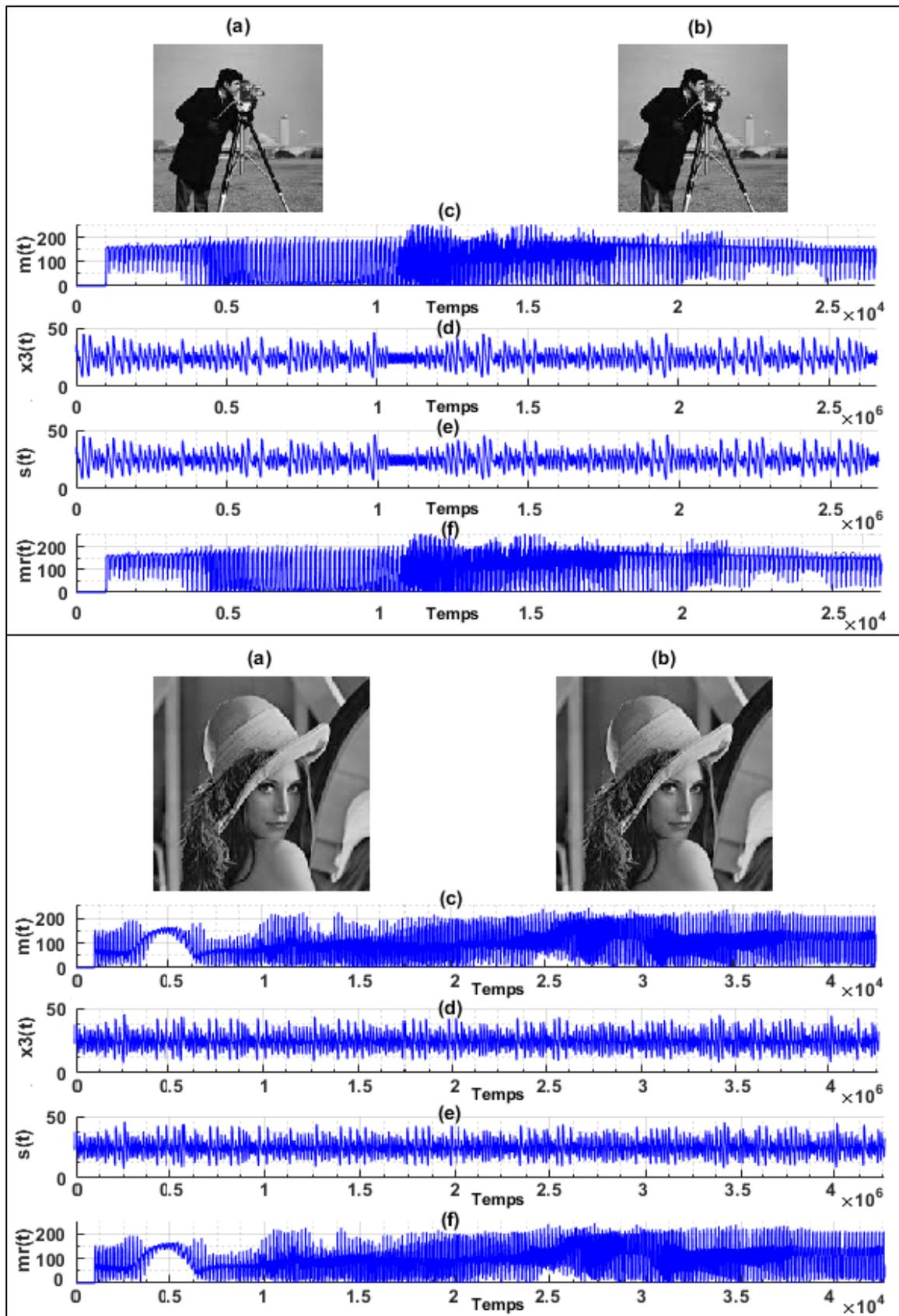
Pour la transmission de la sinusoïde, nous avons utilisé une sinusoïde (figure 3.25) d'amplitude égale à  $A = 10$  et de **4000 échantillons** par période :  $m(t) = 10 \sin(2\pi \left(\frac{f_h}{4000}\right) t)$ , avec  $f_h$  est la fréquence d'horloge du Matlab Simulink dans notre cas. Le facteur de réduction est choisi comme suit :  $f = 1/100$ .

Pour la transmission des images, nous avons transmis deux images de 256 niveaux de gris : l'image "CameraMan.jpg" (figure 3.26) de taille 160x160 pixels et l'image "Lenna.jpg" (figure 3.26) de taille 204x204 pixels, dont le facteur de réduction égale à :  $f = 1/256$ .

Les erreurs quadratiques moyennes calculées entre les signaux reconstitués et les signaux transmis des : sinusoïde, image "CamaraMan.jpg" et image "Lenna.jpg", sont respectivement :  $6.96 \times 10^{-4}$ , **0.4**, **0.25**. Les valeurs de ces erreurs montrent que le système de transmission chaotique implémenté sous Matlab Simulink fonctionne très bien pour lesdits paramètres.



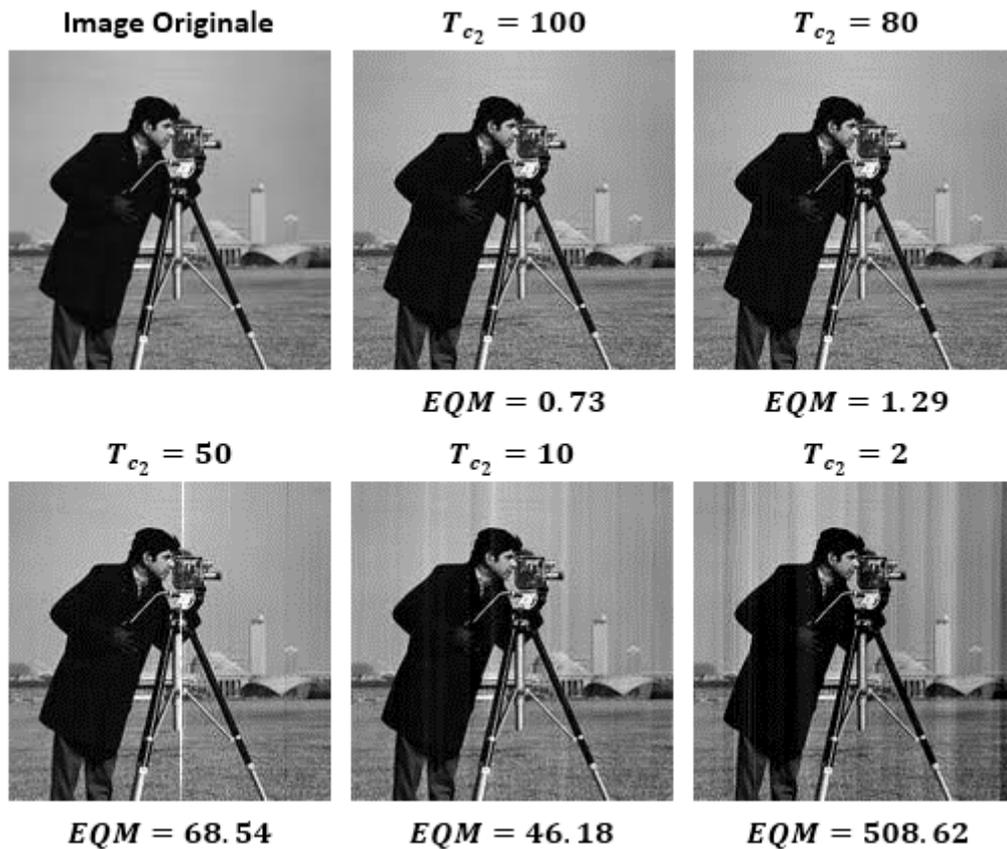
**Figure 3.25** : Résultats de transmission d'une sinusoïde à travers le système de transmission chaotique implémenté sous Matlab Simulink avec : (a) Signal utile, (b) Porteuse chaotique, (c) Signal crypté et (d) Signal reconstitué



**Figure 3.26 :** Résultats de transmission des images "CameraMan.jpg" et "Lenna.jpg" à travers le système de transmission chaotique implémenté sous Matlab Simulink avec :  
 (a) Image émise, (b) Image reconstituée, (c) Signal utile, (d) Porteuse chaotique, (e) Signal crypté et (f) Signal reconstitué

**d. Influence du paramètre  $T_{c_2}$  sur la qualité de transmission**

Dans cette section, nous allons illustrer l'influence du paramètre de temps  $T_{c_2}$  sur la qualité de transmission de notre système, en analysant un exemple de transmission de l'image "CameraMan.jpg". Par la suite nous introduisant une technique pratique que nous avons utilisé pour diminuer au maximum le temps  $T_{c_2}$ , tout en préservant une meilleure qualité de transmission. L'exemple suivant est donné pour un  $dt = 10^{-4}$  qui est choisi d'une manière à se situer dans le cas extrême ( $dt = dt_{max}$ ).

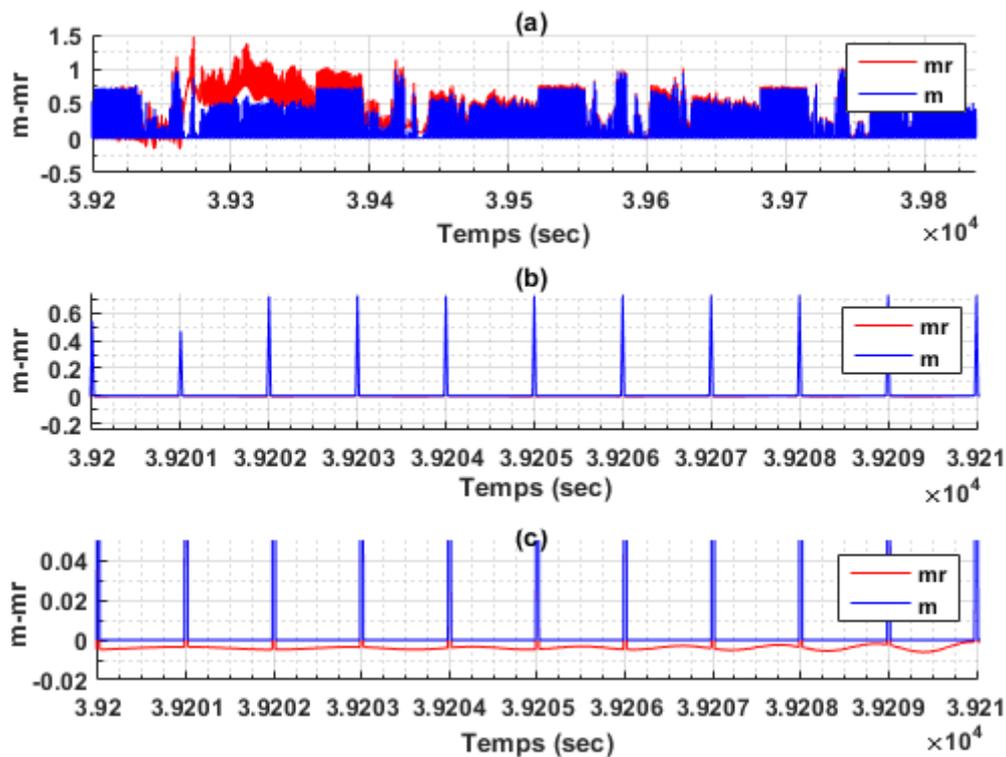


**Figure 3.27 :** Influence du paramètre  $T_{c_2}$  sur la qualité de la transmission  
Exemple donné sur la transmission de l'image "CameraMan.jpg"

La figure 3.27 illustre l'influence du paramètre  $T_{c_2}$  sur la qualité de transmission, en montrant les valeurs de l'erreur quadratique moyenne entre l'image reconstituée et l'image d'origine pour chaque valeur de  $T_{c_2}$ . En diminuant le paramètre  $T_{c_2}$ , on constate une dégradation dans la qualité de l'image reconstituée par rapport à l'image d'origine et par conséquent une dégradation dans la qualité de transmission, d'où la nécessité de choisir judicieusement le paramètre  $T_{c_2}$  adéquat pour assurer un meilleur rapport qualité de transmission / temps de transmission (gain en calcul). La question dont nous a eu à

l'esprit était : Est ce qu'il y a un moyen pour diminuer au maximum le paramètre  $T_{c_2}$  sans perturbé la qualité de transmission ? En réponse à cette question, nous allons analyser en détail les résultats de la figure 3.27 dont les résultats sont illustrés dans la figure 3.28.

La figure 3.28.a présente les signaux utile  $m(t)$  (prise après le sur-échantillonneur) et reconstitué  $m_r(t)$  (prise avant le sous-échantillonneur) correspondants à la bande blanche (colonnes [86 – 89]) représenté dans la figure 3.27 ( $T_{c_2} = 50$ ), dont les erreurs entre les pixels de cette image et l'image d'origine sont les plus grandes. Cependant, La figure 3.28.b présente un agrandissement (Zoom) d'une partie de la figure précédente suivant l'axe des X, alors que la figure 3.28.c présente un agrandissement suivant l'axe des Y de la figure 3.28.b.



**Figure 3.28 :** Les signaux utile et reconstitué résultants de la transmission de l'image "CameraMan.jpg" (agrandissement sur les colonnes [86 – 89] ) pour les paramètres :  $dt = 10^{-4}$  et  $T_{c_2} = 50$ , avec : (a) zoom sur tous les colonnes [86 – 89], (b) zoom sur une partie des colonnes [86 – 89] et (c) zoom suivant l'axe des Y de cette partie

En observant la figure 3.28.b, on remarque qu'il y a un décalage permanent entre le signal reconstitué (en rouge) et le signal utile (en bleu), qui est très faible et quasiment constant entre deux échantillons successifs des signaux utile et reconstitué. Ce décalage représente l'erreur de synchronisation ( $e_3(t)$  dans notre cas) qui n'est pas parfaitement

nulle (elle est de l'ordre  $10^{-3}$  dans cet exemple) due à l'influence du signal utile  $m(t)$  sur le processus de synchronisation. Cette erreur  $e_3(t)$  est parfois plus grande que le signal utile qui a subi une réduction par le facteur  $f$  ( $f = 1/256$  dans cet exemple, alors la valeur la plus petite de notre signal utile  $\cong 0.0039$  qui est de l'ordre de l'erreur), phénomène qui va s'ajouter à notre signal  $m(t)$  et s'amplifier lors de la reconstitution du signal  $m_r(t)$  due à l'amplification du signal par le facteur  $1/f$  (soit : 256 dans cet exemple). Ainsi, le signal reconstitué à la sortie peut s'écrire sous la forme :

$$m_r(t) \cong m(t) + (1/f) \times e_3(t) \quad (3.33)$$

Afin de faire face à ce phénomène, notre idée consiste à éliminer le terme en fonction de l'erreur (soit :  $(1/f) \times e_3(t)$ ) de l'équation (3.33). Vu que l'erreur est quasiment constante entre deux échantillons successifs du signal reconstitué (c'est-à-dire :  $e_3(t) \cong e_3(t - dt)$ ), nous avons ajouté à la sortie de notre système de transmission chaotique représenté dans la figure 3.20 et avant l'opération d'amplification, un correcteur d'erreur qui va soustraire le terme  $e_3(t - dt)$  de la sortie du système. L'équation (3.33) devient :

$$m_r(t) \cong m(t) + (1/f) \times (e_3(t) - e_3(t - dt)) \Rightarrow m_r(t) \cong m(t)$$

Le nouveau schéma de transmission avec correcteur d'erreur est illustré dans la figure 3.29.

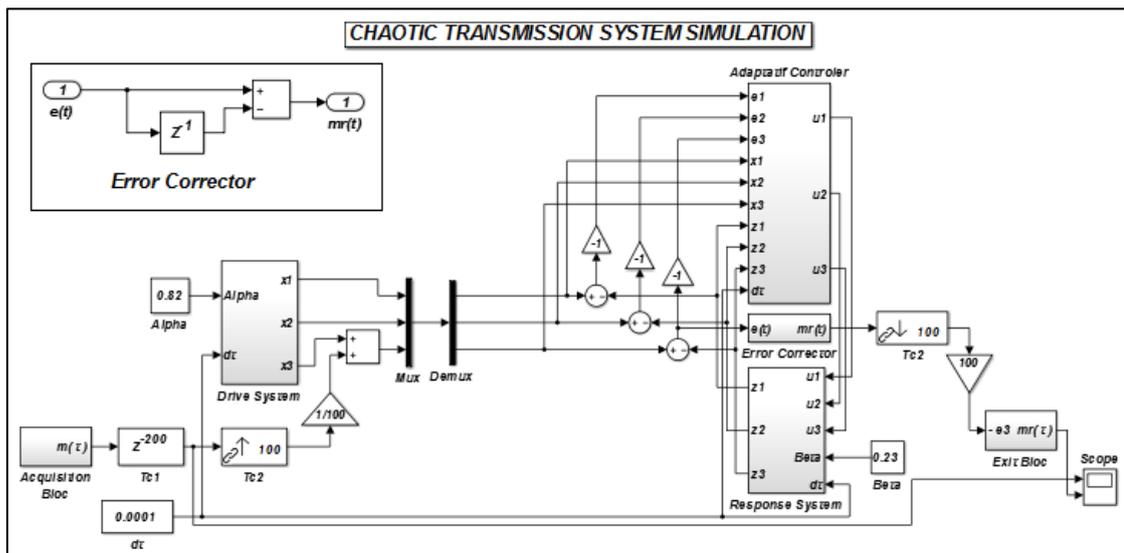
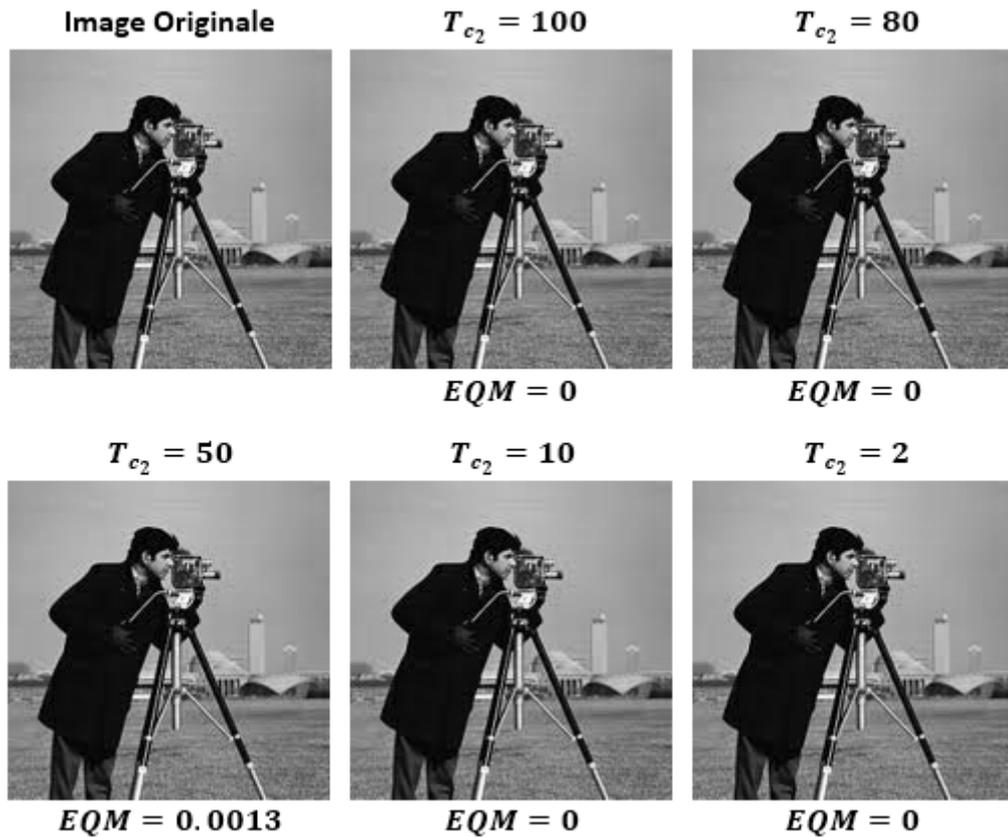


Figure 3.29 : Schéma global du système de transmission chaotique proposé avec correcteur d'erreur



**Figure 3.30** : Influence du paramètre  $T_{c_2}$  sur la qualité de la transmission en utilisant le correcteur d'erreur - exemple donné sur la transmission de l'image "CameraMan.jpg"

La figure 3.30 montre l'effet du correcteur d'erreur ajouté, sur la qualité de transmission en fonction du paramètre  $T_{c_2}$ . En analysant ces résultats, on remarque que l'erreur quadratique moyenne entre l'image reconstitué et l'image d'origine est quasiment nulle pour chaque cas de  $T_{c_2}$ . En comparant ces résultats avec ceux présentés dans la figure 3.27, nous pouvons conclure que le nouveau système de transmission chaotique proposé (figure 3.29) présente de meilleures performances que ceux présentés dans la figure 3.20.

L'ajout du correcteur d'erreur n'influe pas sur le processus de synchronisation, il force juste la sortie de notre système de transmission à reconstitué le plus fidèlement le signal informatif émis depuis l'émetteur, tout en diminuant au maximum le paramètre  $T_{c_2}$ . L'utilisation du correcteur d'erreur, nous a permis aussi d'utiliser un pas d'intégration assez grand (soit :  $dt = dt_{max} = 10^{-4}$  dans cet exemple) qui peut diminuer considérablement le paramètre de retard de transmission  $T_{c_1}$ , tout en préservant une meilleure qualité de transmission. En effet, le fait d'avoir des paramètres de transmission optimums ( $T_{c_1}, T_{c_2}$  et  $dt$ ), augmente considérablement le débit de transmission, tout en diminuant le

temps de calcul pour une éventuelle implémentation pratique (implémentation FPGA dans notre cas).

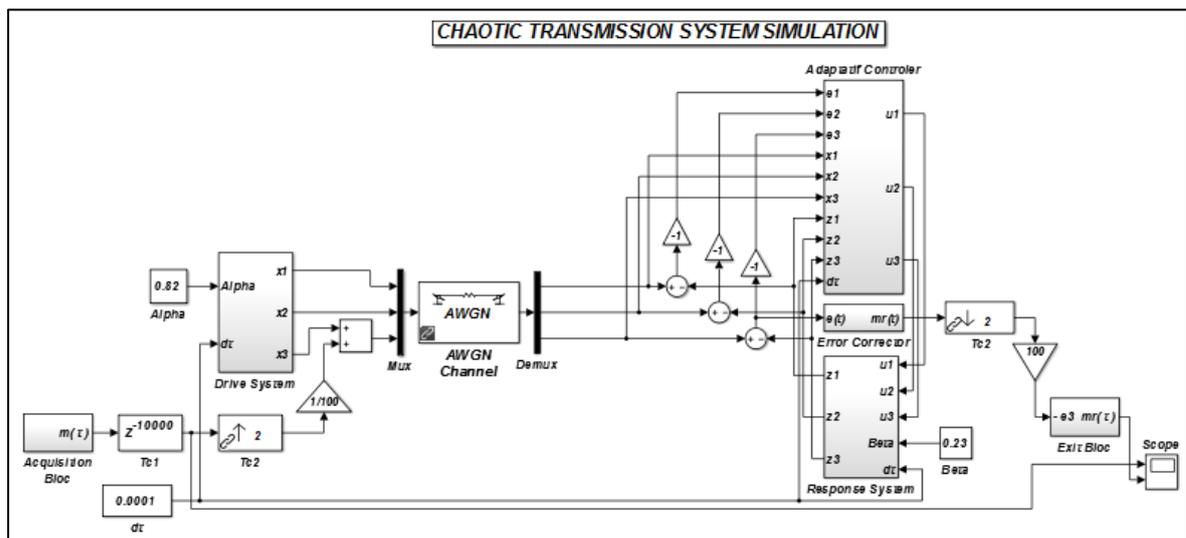
### 3.5.5 Récapitulatif sur le système de transmission chaotique proposé

Le système de transmission chaotique proposé dans ce travail est présenté dans la figure 3.29. Il est constitué essentiellement des blocs suivants :

- ✓ **Émetteur chaotique** : il est défini par le système d'équations (3.15) et représenté par la figure 3.16.
- ✓ **Récepteur chaotique** : il est défini par le système d'équations (3.16) et représenté par la figure 3.17.
- ✓ **Contrôleur adaptatif** : il est défini par le système d'équations (3.20) et représenté par la figure 3.18.
- ✓ **Estimateurs des paramètres  $\hat{\alpha}$  et  $\hat{\beta}$**  : ils sont définis par les équations (3.27) et (3.28) et représentés dans la figure 3.19.
- ✓ **Correcteur d'erreur** : il est représenté par la figure 3.29.
- ✓ **Blocs d'acquisitions et de sorties** : ils sont représentés par la figure 3.21.

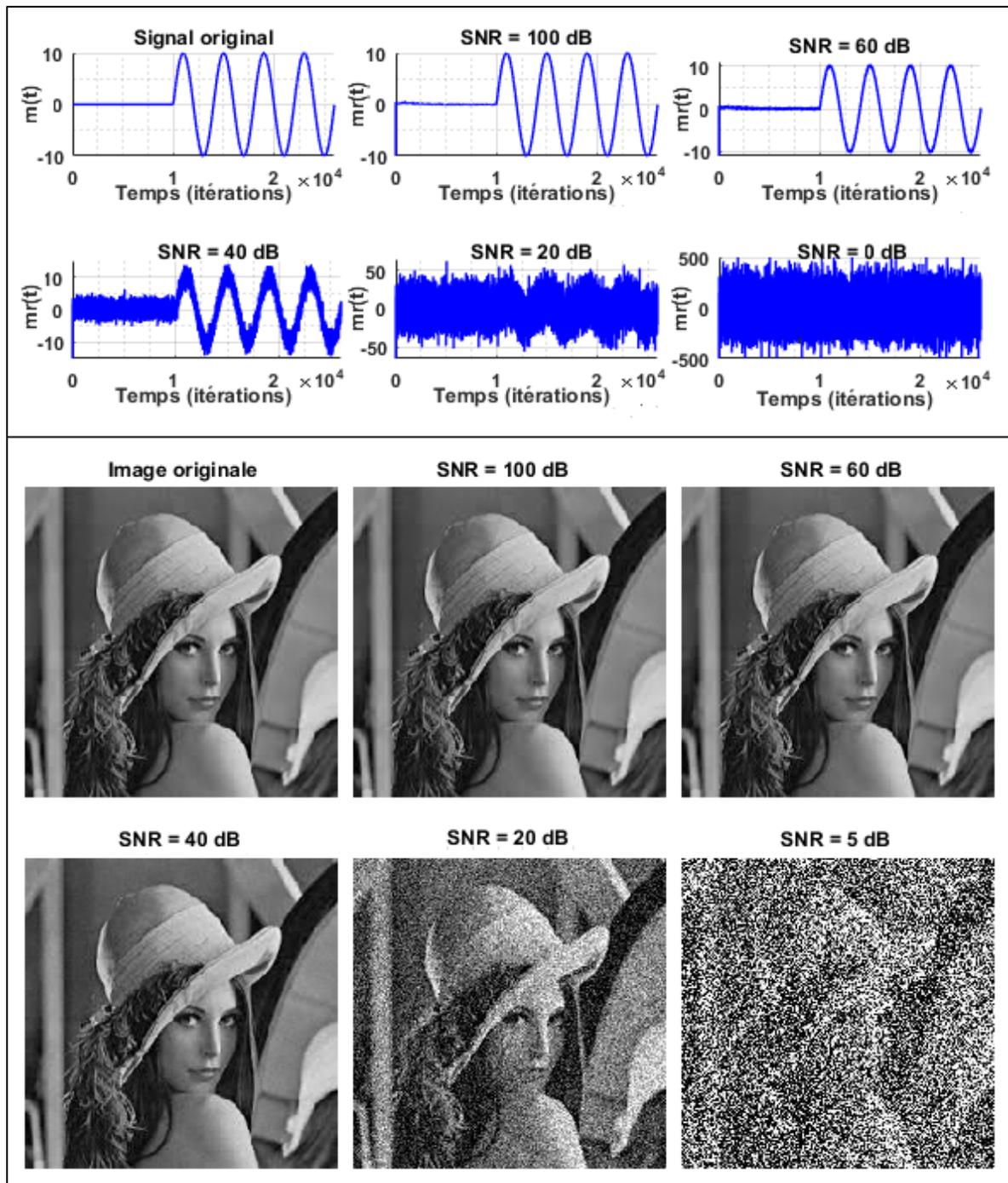
### 3.5.6 Effet du bruit sur la transmission

Cette section présente l'effet d'un bruit Blanc Gaussien Additif associé au canal de transmission sur les résultats de transmission de notre système chaotique (figure 3.31).



**Figure 3.31** : Schéma global du système de transmission chaotique proposé avec un Bruit Blanc Gaussien Additif associé au canal de transmission sous Matlab Simulink

Les résultats de la simulation associée à cette configuration sont illustrés dans la figure 3.32 pour différents SNR, dont les paramètres de transmission sont les suivants :  $dt = 10^{-4}$ ,  $T_{c_1} = 2 \times 10^4$  échantillons,  $T_{c_2} = 2$  échantillons,  $f = 1/100$  pour la sinusoïde et  $f = 1/256$  pour l'image.



**Figure 3.32** : Effet d'un Bruit Blanc Gaussien Additif sur la qualité de transmission chaotique de la sinusoïde et de l'image "Lenna.jpg" en fonction du SNR en dB

On remarque qu'un bruit élevé ne favorise pas une bonne qualité de transmission, ainsi la transmission en présence d'un canal bruité des images à travers du système de transmission chaotique proposé est plus robuste que la transmission d'une sinusoïde. Cela est due à ce que les images utilisées, ont un nombre fini de valeurs distinctes (256 valeurs avec un pas de 1) contrairement à la sinusoïde, présentant ainsi un nombre de valeurs assez élevé avec un pas qui tend vers 0, qui la rend plus susceptible au bruit.

### **3.6 Conclusion**

Dans ce chapitre et après l'introduction du concept de la synchronisation chaotique, les principales méthodes de synchronisation chaotique et les méthodes de transmission chaotique, nous avons illustré en détail le processus de conception du système de transmission chaotique proposé ainsi que son implémentation sur Matlab Simulink. Les résultats de la simulation ont prouvé que ledit système trouve une synchronisation fine en un temps fini, permettant par cela de recouvrir le plus fidèlement le message transmis depuis l'émetteur chaotique. La robustesse dudit système contre un Bruit Blanc Gaussien Additif est présentée en fin de chapitre.

Ces résultats nous seront d'un grand intérêt lors de l'implémentation de la transmission chaotique (émetteur et récepteur) sur circuit FPGA, qui sera étudiée dans le chapitre suivant.

### 4.1 Introduction

Les progrès dans la capacité d'intégration des circuits électroniques ont ouvert de nouvelles perspectives pour les systèmes embarqués dans de nombreux domaines industriels et grand public comme par exemple les télécommunications, les satellites, et l'imagerie médicale, etc. D'un côté, des processeurs spécifiques (CPU, microcontrôleurs, DSP) peuvent couramment effectuer des milliards d'opérations par seconde et d'un autre côté, des composants reprogrammables (CPLD, FPGA, ASIC) comporteront dans un avenir proche plusieurs milliards de portes logiques. Ces circuits permettent de réaliser des applications avec des performances en terme de vitesse de traitement sans cesse croissantes [35].

Dans ce contexte, le FPGA (Field Programmable Gate Array) avec ses grandes capacités d'intégration et de reconfiguration en font un composant clé pour développer rapidement des prototypes, d'où l'objet de ce chapitre : Implémentation FPGA du système de communication sécurisé par chaos, étudié dans le chapitre précédent.

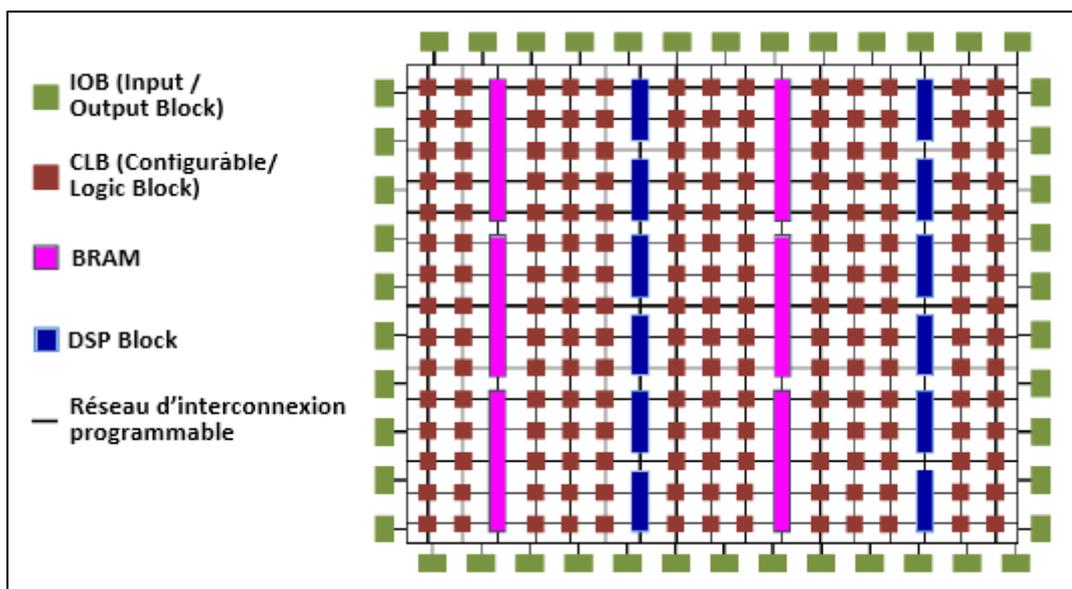
Dans ce chapitre, nous commençons par la présentation de la technologie FPGA ainsi du processus d'implémentation, tout en décrivant le flot de conception d'une architecture sur FPGA. Par la suite, nous introduisons les outils de Xilinx, nécessaires pour une telle implémentation, à savoir : le logiciel ISE et le System Generator qui permet, en plus de l'implémentation de la transmission chaotique sur FPGA, une simulation hardware (Co-Simulation) de l'architecture implémentée sous l'environnement Matlab Simulink. La description de la plateforme FPGA Xilinx ML501 ainsi du codec Ac'97-AD1981B, est

également présentée. Nous terminons ce chapitre par la présentation de l'ensemble des résultats expérimentaux et de leurs comparaisons aux résultats simulés.

## 4.2 Présentation des circuit FPGA

Les FPGA sont des composants logiques VLSI (Very Large Scale Integration) [9], de haute densité et reconfigurables qui permettent, après programmation, de réaliser des fonctions logiques, des calculs, et des générations de signaux, etc. [35]. 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 [11].

### 4.2.1 Architectures des FPGA



*Figure 4.1 : Architecture moderne des FPGA Xilinx*

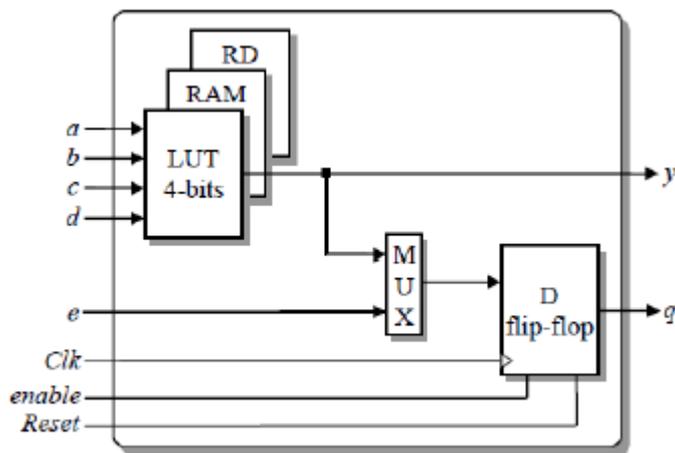
La structure régulière des FPGA (figure 4.1) permet de réaliser des éléments très performants pour les traitements de bas niveaux réguliers. Ils sont constitués principalement de trois parties [9] :

- ✓ Une matrice de blocs logiques configurables CLB (Configurable Logic Block).
- ✓ Des blocs d'entrée/sortie configurables IOB (Input Output Block).
- ✓ Un réseau d'interconnexions programmables.

Depuis quelques générations de FPGA, les fabricants ont ajouté aux ressources classiques, sur certains modèles, de nombreux éléments tels que [35] :

- ✓ Les blocs mémoire BRAM (Block Random Access Memory).
- ✓ Les multiplieurs.
- ✓ Les blocs processeur DSP (Digital Signal Processing).

La figure 4.2 présente la structure d'une cellule logique d'un bloc logique configurable CLB de la technologie Xilinx. Cette structure comporte une table LUT (Look-up Table) de 4 bits qui permet de réaliser n'importe quelle fonction combinatoire à quatre variables logiques. Cette LUT peut être aussi configurée comme étant une mémoire RAM (16×1) ou un registre à décalage de 16 bits. Elle comporte aussi un multiplexeur et une bascule D avec toutes ses entrées de contrôle (horloge, reset, enable).



**Figure 4.2 :** Structure d'une cellule logique d'un FPGA XILINX

#### 4.2.2 Technologies des FPGA

Il existe plusieurs fabricants de composants FPGA tels que [11, 36] : Xilinx, Altera, Microsemi (ex. Actel), Atmel, Cypress, Lattice, Minc, QuicLogic, Nallatech, SiliconBlue, Tabula Inc, TierLogic, etc. Ces constructeurs utilisent différentes technologies pour la fabrication des FPGA. Parmi ces technologies, celles qui assurent une reprogrammation des FPGA sont les plus intéressantes puisqu'elles permettent une grande flexibilité de conception. Parmi ces technologies, nous citons [9] :

- ✓ **La technologie EPROM** : cette technologie utilise des mémoires EPROM (Erasable Programmable Read-Only Memory). Son principal inconvénient est l'opération de reconfiguration qui nécessite l'utilisation d'une source d'ultraviolet.

- ✓ **La technologie EEPROM** : cette technologie utilise des mémoires EEPROM (Electrically Erasable Programmable Read-Only Memory). Par rapport à la technologie EPROM, elle présente l'avantage de pouvoir être reprogrammée électriquement.
- ✓ **La technologie Static Ram (SRAM)** : pour cette technologie, les connexions sont réalisées en rendant les transistors passants ce qui permet une reconfiguration rapide du circuit FPGA. Cependant, son principal inconvénient est la surface nécessaire pour la SRAM.
- ✓ **La technologie FLASH** : Cette technologie est limitée en nombre de reconfigurations et possède un temps de reconfiguration plus long par rapport à la technologie SRAM. L'avantage de cette technologie est qu'elle garde sa configuration même si l'alimentation est coupée. Par conséquent, un FPGA à base de technologie Flash déjà programmé est prêt à fonctionner dès sa mise sous tension.

### 4.2.3 Domaines d'application des FPGA

Les FPGA ont fait révolutionner certains domaines de contrôle numérique et de plus en plus utilisés pour intégrer des architectures numériques complexes. Ils sont devenus les plus populaires en matière d'implantation et de prototypage des circuits numériques après leur apparition sur le marché en 1984. La clé maîtresse de leurs réussites est l'aspect de programmation de ces derniers. Leurs utilisations actuelles couvrent les deux domaines : civil et militaire. Parmi ces applications nous citons [36] :

- ✓ **Informatique** : Périphériques spécialisés.
- ✓ **Machinerie industrielle** : Contrôleur pour machines.
- ✓ **Télécommunications** : Traitement d'images, Filtrage.
- ✓ **Instrumentation** : Équipement médical, Prototypage.
- ✓ **Transport** : Contrôle d'avions et métros.
- ✓ **Aérospatiale** : Satellites, Radar, Communication sécurisée, la détection ou la surveillance, etc.

## 4.3 Processus d'implémentation

Les FPGA sont des composants numériques reconfigurables dont on définit le fonctionnement interne par le biais d'un langage de description matériel de haut niveau, nommé également langage HDL (Hardware Description Language) [35]. Deux langages HDL sont les plus couramment utilisés, à savoir le VHDL (Very high speed integrated Hardware Description Language) et le Verilog. Ces deux langages sont standardisés et offrent au concepteur différents niveaux de description, et surtout l'avantage d'être portables et compatibles avec toutes les technologies FPGA précédemment introduites [9]. On définit ainsi une architecture de fonctionnement dans laquelle on peut utiliser toutes les ressources internes du composant, ainsi que les entrées/sorties, les communications avec d'autres composants, etc [35].

### 4.3.1 Flot de conception sur FPGA

La conception des architectures de fonctionnement des FPGA s'effectue en utilisant des outils de Conception Assistée par Ordinateur (CAO), dont le flot de conception est réalisé en plusieurs étapes (figure 4.3) :

#### *a. Spécifications algorithmiques*

Elles permettent de définir l'architecture par une démarche d'Adéquation Algorithme-Architecture [37].

#### *b. Conception de l'architecture*

Dans cette phase, on décrit l'architecture avec un langage HDL, tout en procédant à une simulation fonctionnelle par le biais d'un simulateur (par exemple : ModelSim) qui exécute le code HDL et confirme que des sorties correctes sont produites pour un ensemble donné d'entrées de test. Cette simulation est faite juste pour vérifier que la logique du code HDL est correcte [38].

#### *c. Synthèse*

Le résultat de cette phase est une représentation intermédiaire de la conception matérielle appelée Netlist, qui décrit la connectivité de l'architecture indépendamment du type de FPGA.

#### d. Placement et routage

Cette phase consiste à mapper d'une façon optimale les structures logiques décrites dans le Netlist sur les macrocellules réelles (CLB, BRAM, DSP, etc.), les interconnexions et les broches d'entrée et de sortie. Le résultat de ce processus est un fichier de configuration FPGA appelé Bitstream.

#### e. Chargement du fichier de configuration dans le FPGA

C'est la dernière phase dans le processus de l'implémentation. Elle consiste à charger le fichier Bitstream résultant de la phase précédente, sur le FPGA cible à partir d'un ordinateur hôte par le biais d'un outil de configuration.

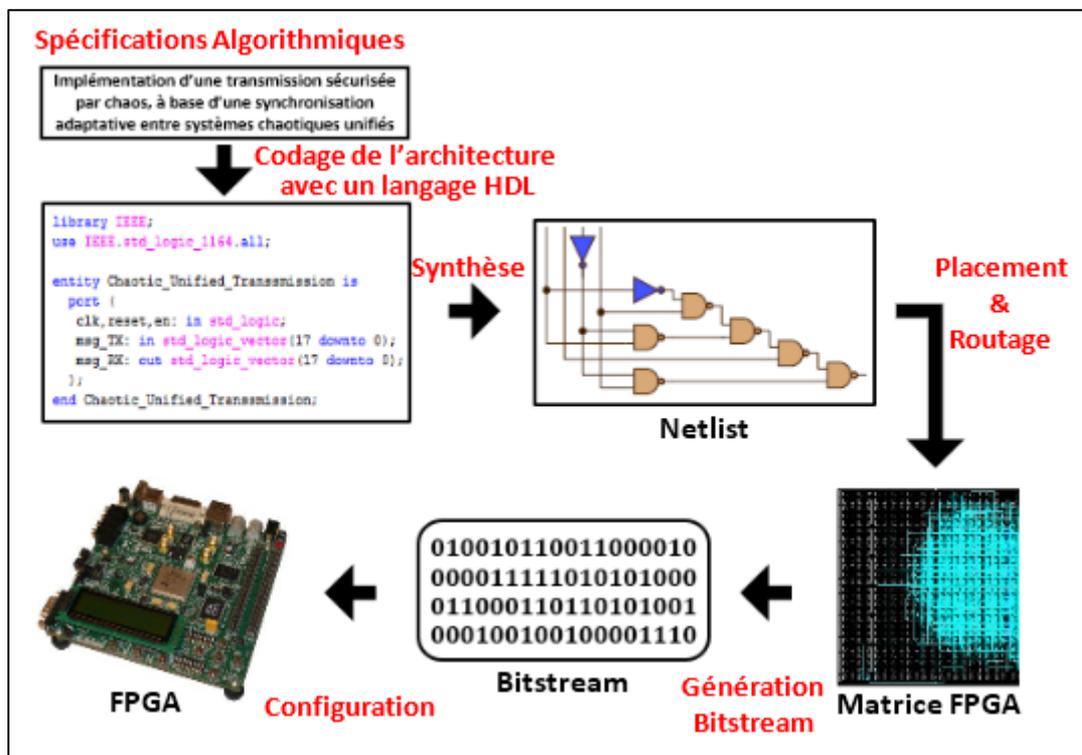


Figure 4.3 : Processus de programmation d'un FPGA

### 4.3.2 Présentation du logiciel Xilinx ISE

L'environnement Xilinx ISE (Integrated Software Environment) est un outil logiciel de programmation des produits Xilinx (CPLD, FPGA Spartan et Virtex, ...) ; permettant ainsi, la synthèse et l'analyse des architectures HDL, tout en passant à travers le flot de conception d'un système numérique. Pour se faire, il intègre différents outils à savoir :

- ✓ Un éditeur de textes, de schémas et de diagrammes d'états.

- ✓ Un outil pour la vérification et de compilation VHDL et Verilog.
- ✓ Un outil pour la gestion des contraintes temporelles.
- ✓ Un outil pour la synthèse.
- ✓ Un outil de simulation qui fonctionne en collaboration avec un simulateur externe (exemple : ModelSim).
- ✓ Un outil pour l'implémentation (Translation, Mappage, Placement et routage).
- ✓ Un outil pour la génération du fichier de configuration (Bitstream)
- ✓ Un outil pour le chargement du Bitstream sur un FPGA ou un CPLD (ISE iMPACT).
- ✓ Un outil pour analyser et déboguer les signaux réels récupérer depuis le FPGA ou le CPLD (ChipScoop Pro).

La figure 4.4 représente l'interface Project Navigator de ISE 14.7 permettant l'accès à toutes les ressources d'un projet ainsi qu'aux outils de l'implémentation.

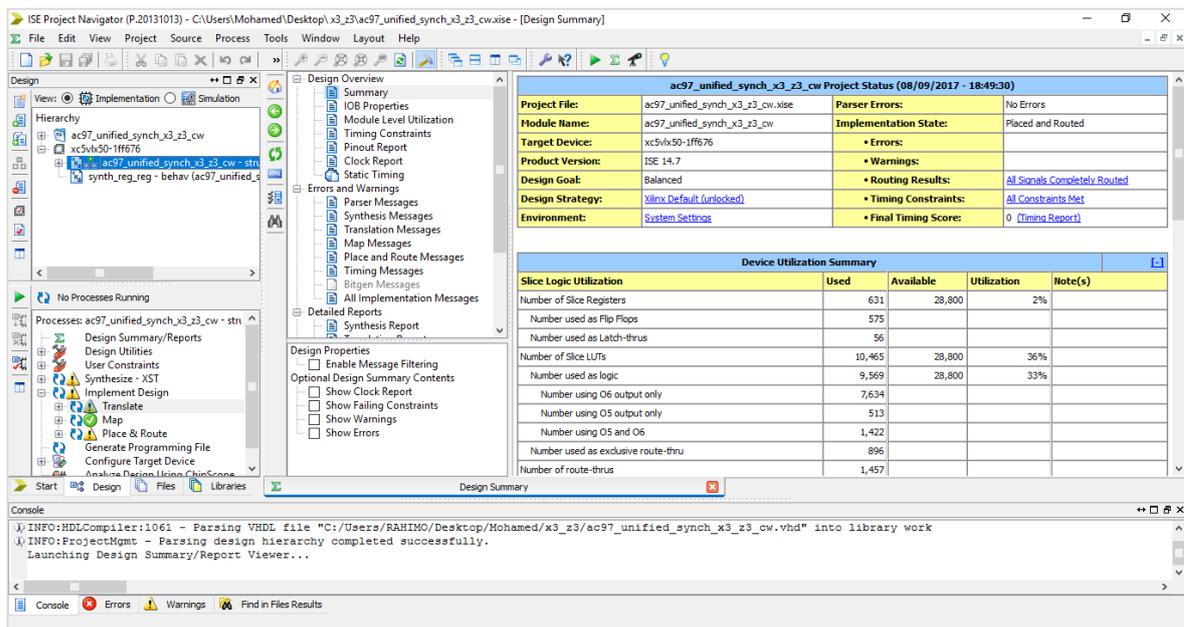


Figure 4.4 : Interface Project Navigator ISE 14.7

### 4.3.3 System Generator et Co-Simulation

L'utilisation d'un langage de description matérielle HDL comme le VHDL ou le Verilog pour configurer un FPGA, nécessite une bonne connaissance non seulement de l'algorithme mais aussi de l'architecture du FPGA et du compilateur utilisé. La plupart des

utilisateurs des FPGA ne sont pas familiarisés avec ces langages matériels, ce qui rend les FPGA d'une utilisation parfois délicate. Pour remédier à ce problème, de nouveaux langages de conception de circuits, basés sur le langage C, ont été proposés. On peut citer entre autres [35] les langages : System-C, Handel-C, JHDL, etc. Par ailleurs des outils de programmation graphique ont été aussi proposés, tels que LabVIEW et le System Generator de Xilinx.

Dans ce contexte et afin d'élaborer un modèle de la transmission chaotique en vue de son implémentation sur FPGA, tout en réduisant le temps de conception, nous avons utilisé l'outil graphique de Xilinx fourni avec l'ISE : System Generator, sous l'environnement Matlab Simulink. Cet outil est considéré comme l'interface entre Matlab Simulink et ISE Xilinx et nous permet de concevoir des applications sur des circuits FPGA. Parmi les principales tâches qu'on peut exécuter sous cet environnement, les suivantes :

- ✓ Conception des systèmes dans un environnement graphique (Matlab Simulink) indépendamment du type du FPGA et de faire des simulations aboutissant à des résultats comparables à ceux obtenus sous Matlab Simulink ainsi des analyses sur les contraintes de temps.
- ✓ Co-Simulation logicielle (Simulink) - matérielle (FPGA) par communication JTAG ou USB.
- ✓ Génération automatique du projet ISE incluant le code du langage HDL mappé aux algorithmes pré-optimisés de Xilinx, le paramétrage adéquat pour chaque bloc et le fichier des entrées/Sorties (UCF). Cette conception HDL peut ensuite être synthétisée pour la mise en œuvre dans différentes plateformes FPGA.

La Co-Simulation matérielle fournie par le System Generator, permet d'intégrer une conception en cours d'exécution dans un FPGA, directement dans une simulation Matlab Simulink par le biais d'une communication JTAG ou USB. En effet, le System Generator compile et crée automatiquement un fichier Bitstream et l'associe au bloc désiré pour une Co-Simulation [39]. Lorsque cette conception est simulée sous Matlab Simulink, les résultats pour la partie compilée sont calculés au niveau du circuit FPGA. Cela permet à la partie compilée d'être testée directement dans le circuit réel, ce qui va accélérer la simulation. Afin que le bloc de Co-Simulation matérielle se synchronise avec son circuit

FPGA, on trouve deux modes : le mode "Single-Step Clock", où le circuit FPGA est piloté par Matlab Simulink et le mode "Free-Running Clock", où le circuit FPGA est piloté par son propre horloge interne.

La figure 4.5 représente le flot de conception de l'outil System Generator et montre les relations entre les différents outils nécessaires pour une implémentation FPGA.



*Figure 4.5 : Flot de conception de Xilinx System Generator*

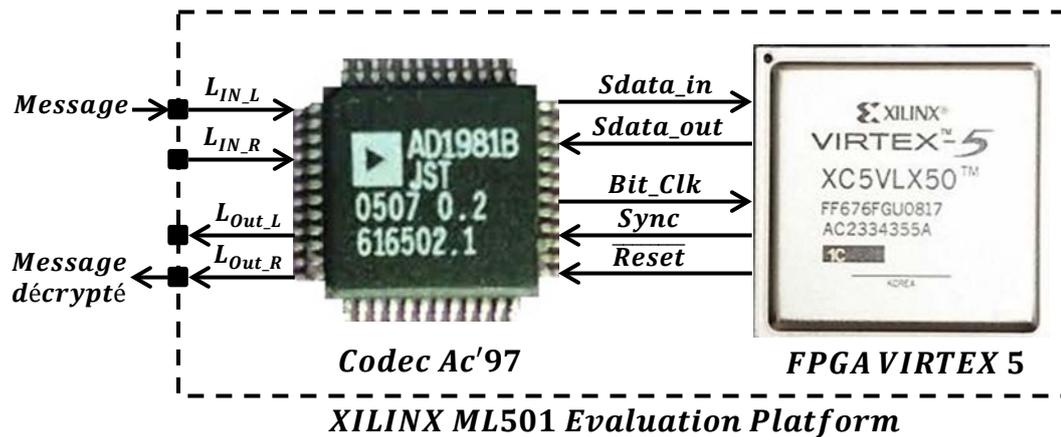
## 4.4 Réalisation expérimentale

La réalisation expérimentale s'est effectuée au niveau du laboratoire LABSET du département électronique de l'université de BLIDA. Ce dernier possède plusieurs plateformes de développement FPGA, à savoir : Xilinx SPARTAN 3<sup>E</sup>, Xilinx ML501, etc. Parmi ces cartes, notre choix est porté sur la carte Xilinx ML501 pour les raisons suivantes :

- ✓ La plateforme de développement Xilinx ML501 est construite autour du circuit FPGA VIRTEX 5 XC5VLX50 qui présente une haute densité d'intégration, alors elle est très riche en ressources logiques, chose qui va très bien avec la complexité de notre système de transmission chaotique.
- ✓ La plateforme de développement Xilinx ML501 contient dans son architecture un codec audio Ac'97 de type AD1981B. Afin de réduire le temps du processus d'implémentation, en éliminant des réalisations supplémentaires des étages de conversions analogique-numérique (CAN) à l'entrée du FPGA (acquisition du message à transmettre) et numérique-analogique (CNA) à la sortie du FPGA

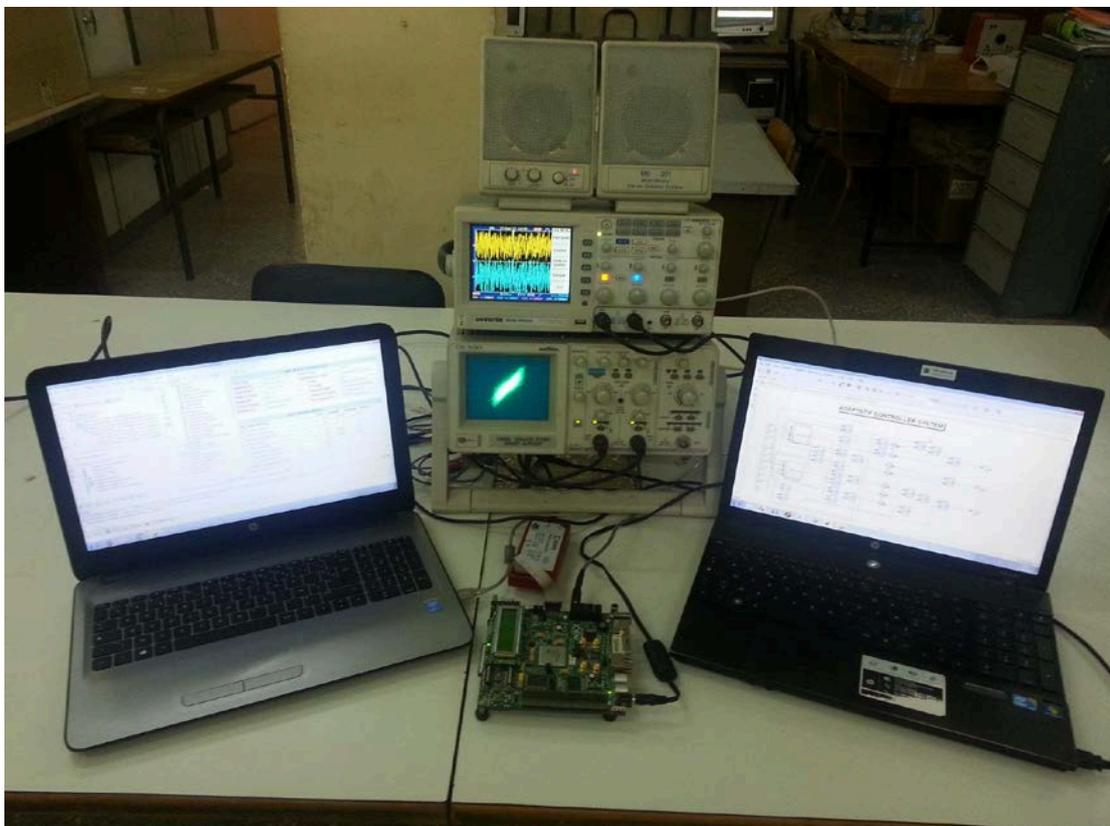
(visualisation du message reconstitué), nous avons utilisé dans notre implémentation le codec audio Ac'97 – AD1981B de la carte ML501 comme étant : un CAN à l'entrée du circuit FPGA et un CNA à la sortie du circuit FPGA.

À cet effet, le schéma synoptique de notre implémentation expérimentale de la transmission chaotique (émetteur-récepteur) est constitué de deux étages : le circuit FPGA XC5VLX50 et le codec audio Ac'97 – AD1981B (figure 4.6).



**Figure 4.6 :** Architecture de l'implémentation de la transmission chaotique

La photo de la figure 4.7 montre l'environnement de la réalisation expérimentale.



**Figure 4.7 :** Réalisation expérimentale de l'implémentation

#### 4.4.1 Plate-forme de développement Xilinx ML501 [40]

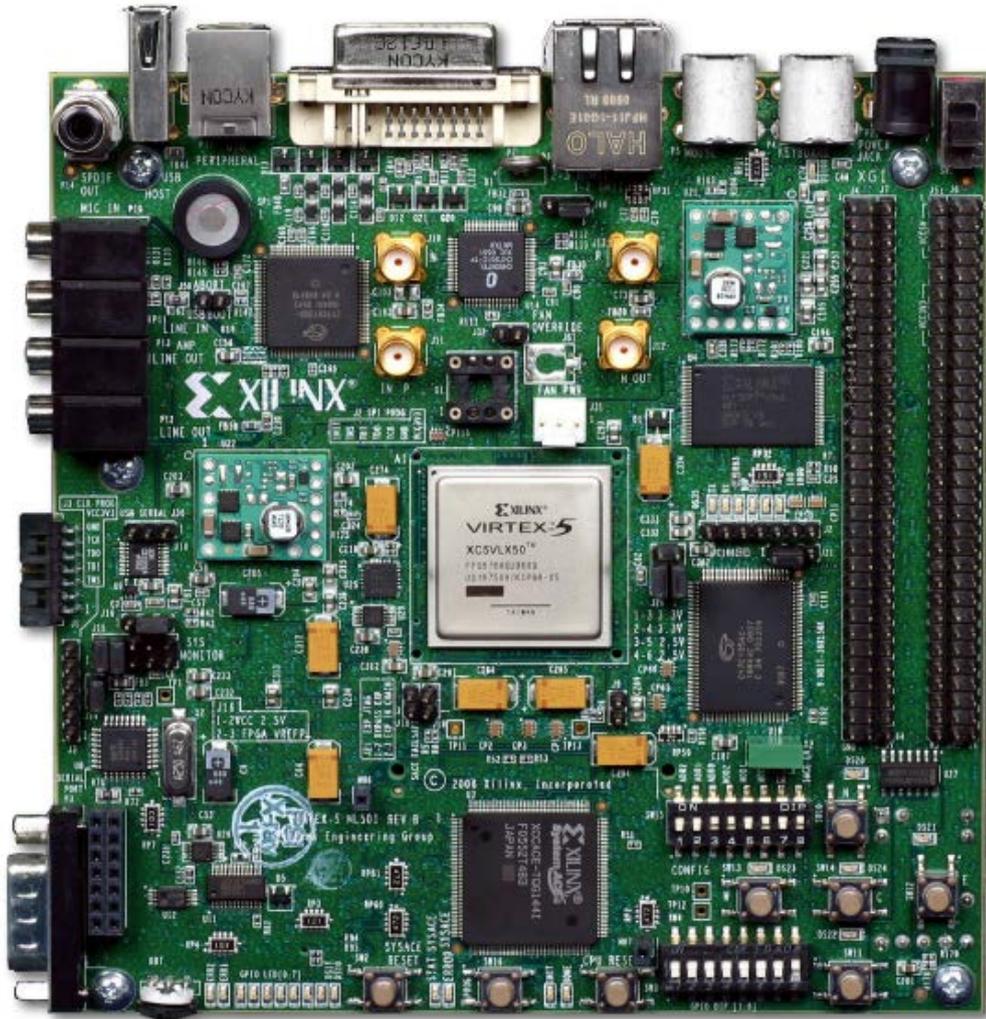
La carte Xilinx ML501 (figure 4.8) est une plateforme de conception et de mise on ouvre des circuits numériques sur un circuit FPGA de type Virtex5 XC5VLX50. Les caractéristiques de cette carte assurent une solution parfaite pour l'implémentation des applications sur FPGA surtout pour les applications qui ont besoin de ressources mémoires généreuses étendues et aussi de plus de flexibilité et d'efficacité.

##### *a. Description des éléments de la carte*

La plateforme ML501 dispose d'une mémoire externe (DDR2-SDRAM SODIMM) de capacité 256MB. En plus du circuit FPGA et de la DDR2, la carte ML501 est une plateforme riche en composants pour l'évaluation et le développement des multiples applications, ces composants permettent un accès facile et pratique aux ressources disponibles dans le circuit FPGA [41]. Les ports vidéo, audio et de communication ainsi que les ressources mémoires généreuses donnent la fonctionnalité et la flexibilité à cette plateforme et la rendent une plateforme de développement FPGA typique.

La plateforme ML501 dispose aussi de :

- ✓ **Mémoires** : Xilinx Platform Flash XCF32P ; 9Mb ZBT synchronous SRAM ; 32MB Intel P30 StrataFlash ; 2MB SPI Flash ; System ACE™ CompactFlash configuration controller avec un connecteur CompactFlash ; IIC-EEPROM de 8Kbits.
- ✓ **Connectivité** : Port JTAG utilisé avec un câble parallèle(III) ; câble parallèle(IV) ou bien plateforme de téléchargement USB ; Ethernet PHY 10/100/1000 ; Codec Stéréo Ac'97 avec connecteurs : line-in/microphone, line-out/headphone, et audio digital SPDIF ; Piezo Speaker ; Port PS/2 pour souri/clavier ; des E/S d'extension ; Port série type RS-232 ; Connecteur vidéo (DVI/VGA).
- ✓ **Composants divers** : Générateur de système d'horloge programmable ; CPLD type C95144XL de Xilinx ; support d'oscillateur d'horloge (3.3V) avec un oscillateur à 100MHz ; contrôleur de température et de tension ; boutons poussoirs à usage général ; afficheur LCD de 16 caractères sur 2 lignes ; entrée d'alimentation ; LED d'indication d'alimentation, d'initialisation et d'activation) ainsi que plusieurs switchers et ports d'E/S.



*Figure 4.8 : La plateforme de développement XILINX ML501*

### **b. Le circuit FPGA associé**

Le circuit FPGA XC5VLX50-1FFG676 est un circuit de la deuxième génération ASMBL (Advanced Silicon Modular Block) de Xilinx. En plus des caractéristiques hautes performances assurées par Xilinx dans des versions précédentes, les nouveaux circuits contiennent plusieurs blocs IP : des BRAM, des FIFOs, des DSPs, interfaces DCM, générateurs d'horloge, etc. La puce LX50 intègre également un bloc de gestion des horloges qui permet une répartition d'horloges parfaite, il dispose d'un maximum de 560 E/S. Lors de la synthèse des implémentations, il est possible de choisir de réaliser les mémoires et les multiplieurs par des LUTs ou bien par des BRAMs et des DSP48E. Ce circuit peut être configuré à l'aide de plusieurs dispositifs : câble JTAG, System ACE controller, Platform Flash PROM, Linear Flash memory et SPI Flash memory [41].

#### 4.4.2 Codec Ac'97 – AD1981B

La plateforme de développement Xilinx ML501 contient un codec stéréo Ac'97 de type AD1981B, dont le diagramme de fonctionnement est représenté dans la figure 4.9. Ce codec a été conçu spécifiquement pour fournir une haute qualité d'audio et peut être intégré dans divers systèmes. Il dispose de [42] : un CAN de résolution de 16 bits et un CNA de résolution de 20 bits, une sortie digitale SPDIF supportant des fréquences d'échantillonnages de 48kHz et 44.1kHz, une sortie stéréo analogique line-out, une sortie stéréo analogique amplifiée (HP), une sortie audio mono analogique, 3 entrées audio stéréo analogiques (line-in, AUX et CD), une entrée microphone doublé préamplifié et une entrée pour téléphone (phone\_in). Le contrôle des paramètres du codec (volume, mute, atténuation,...) s'effectue à base des registres adressables [42, 43].

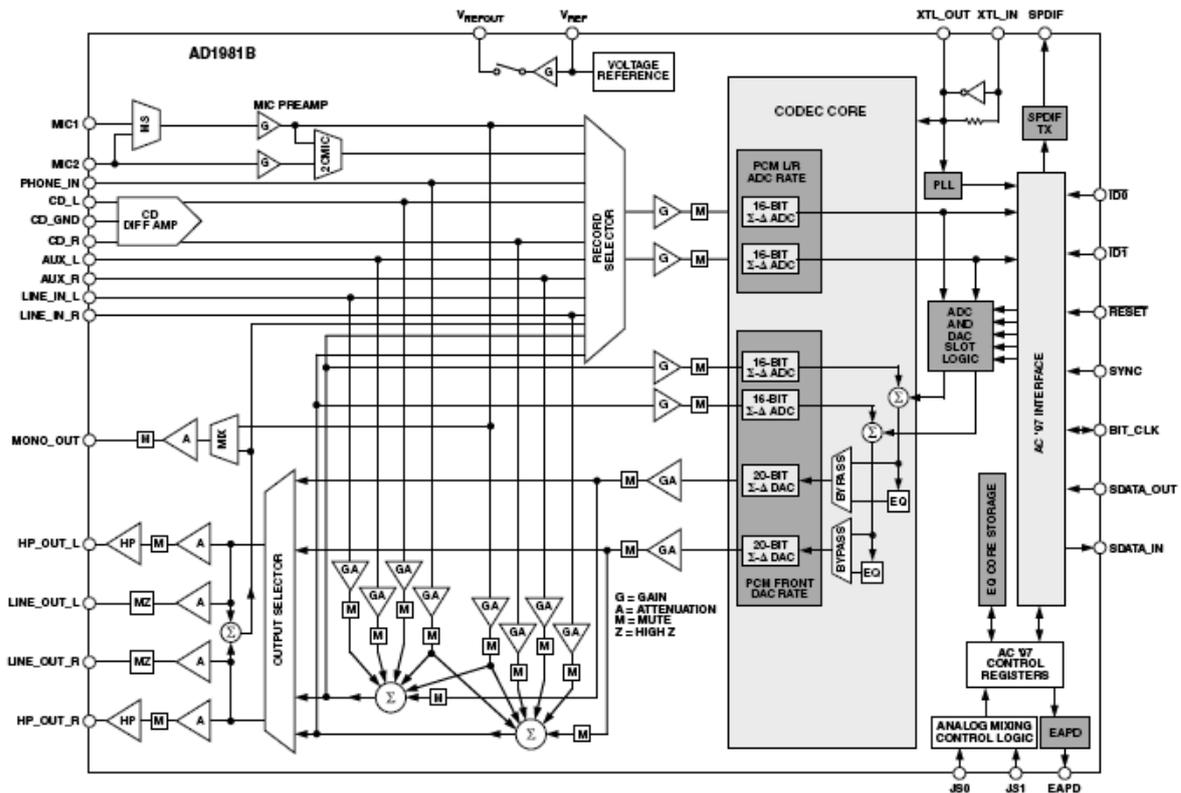
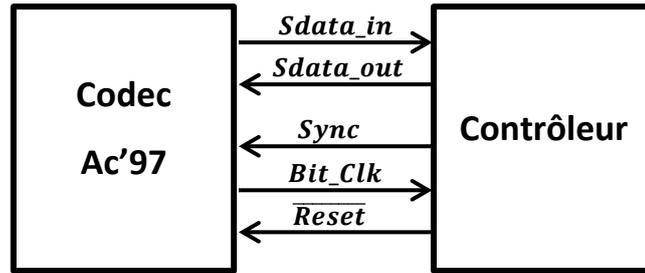


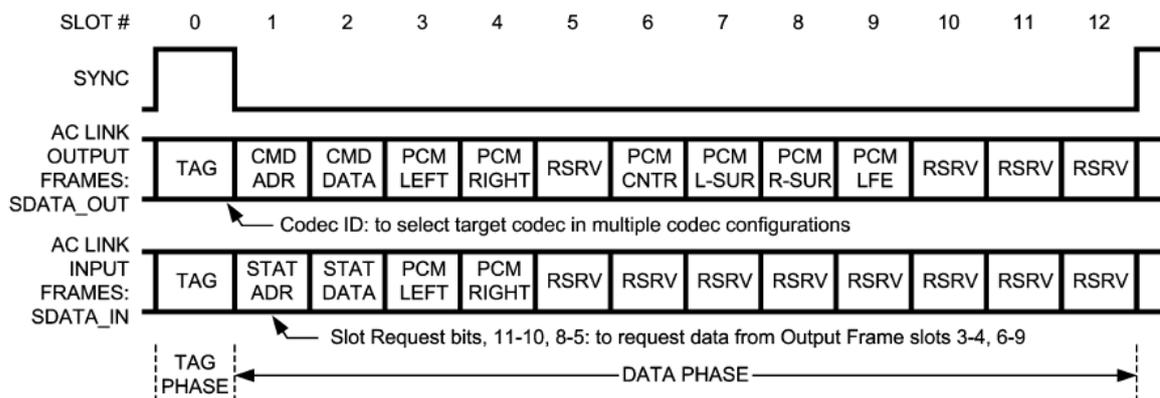
Figure 4.9 : Diagramme de fonctionnement du codec stéréo Ac'97 – AD1981B

L'implémentation du codec stéréo Ac'97 au sein d'un système nécessite un contrôleur du codec, formant ainsi une interface entre le codec et le cœur du système, appelée : AC link digital serial interface protocol. Ce contrôleur assure la communication des données audio et des commandes de registres du codec entre le cœur du système et le codec. La liaison AC comporte cinq signaux (figure 4.10) à savoir [42, 43, 44] :



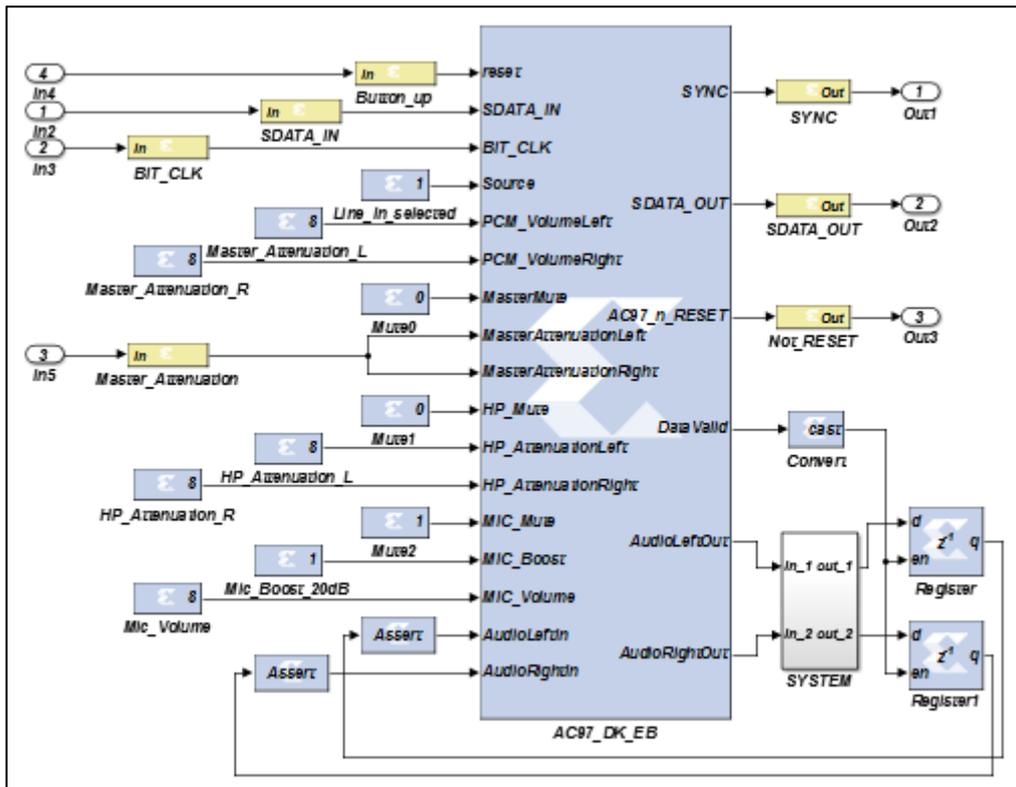
**Figure 4.10** : La liaison AC entre le codec stéréo Ac'97 et son contrôleur

- ✓ **SYNC** : signal de synchronisation entre le codec et le contrôle calibré pour la transmission d'une trame (1 trame = 256 bits).
- ✓ **BIT\_CLK** : l'horloge transmise par le codec vers le contrôleur calibré pour la transmission d'un bit et égale à : 12.288 MHz.
- ✓ **SDATA\_OUT** : la donnée sérialisée transmise depuis le contrôleur vers le codec (figure 4.11). Elle est constituée de trames de 256 bits, chaque trame comporte 13 slots (slot 0 = 16 bits et slot 1: 12 = 20 bits).
- ✓ **SDATA\_IN** : la donnée sérialisée transmise depuis le codec vers le contrôleur (figure 4.11). Elle est constituée de trames de 256 bits, chaque trame comporte 13 slots (slot 0 = 16 bits et slot 1: 12 = 20 bits).
- ✓ **RESET** : le signal reset transmis par le contrôleur.



**Figure 4.11** : Trames audio de la liaison AC entre le codec stéréo Ac'97 et son contrôleur

Dans notre cas, l'implémentation du contrôleur se fait à base d'un code VHDL que nous l'avons téléchargé depuis internet intégré dans System Generator (figure 4.12).



**Figure 4.12 :** Implémentation du contrôleur du codec stéréo Ac'97-AD1981B sous Matlab Simulink - System Generator

## 4.5 Implémentation de l'émetteur chaotique sur FPGA

Dans cette section, nous décrivons le processus de l'implémentation de notre émetteur chaotique sur la plateforme de développement Xilinx ML501 en utilisant l'outil de Xilinx System Generator sous l'environnement Matlab Simulink. Les résultats de simulation, de Co-Simulation matérielle et ceux relevés au niveau de la sortie du codec Ac'97 à l'aide d'un oscilloscope numérique seront présentés en fin de section.

Nous rappelons le système d'équations du système chaotique Unifié présenté dans le chapitre 2 (équation (2.3)) :

$$\begin{cases} \dot{x}_1 = (25\alpha + 10)(x_2 - x_1) \\ \dot{x}_2 = (28 - 35\alpha)x_1 - x_1x_3 + (29\alpha - 1)x_2 \\ \dot{x}_3 = x_1x_2 - \frac{\alpha + 8}{3}x_3 \end{cases}$$

où  $\alpha$  est le paramètre du système, tels que  $\alpha \in [0,1]$ .

L'intégrateur d'Euler n'existant pas dans la bibliothèque du System Generator, nous l'avons alors synthétisé à l'aide des blocs disponibles (Figure 4.13).

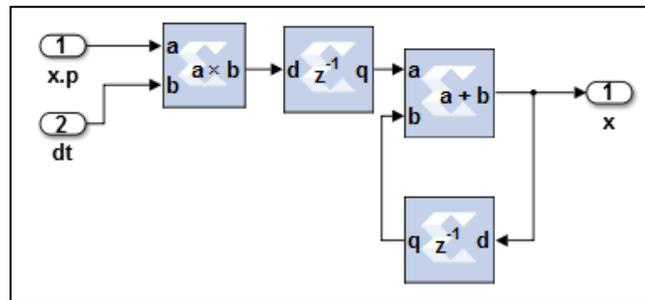


Figure 4.13 : Implémentation du bloc intégrateur

La figure 4.14 représente l'implémentation de l'émetteur chaotique, incluant le système chaotique Unifié et l'insertion du message à transmettre. Cette figure montre également le format de calcul utilisé pour chaque bloc qui est choisi de manière à assurer le bon fonctionnement du système tout en ayant un minimum de consommation en ressource FPGA.

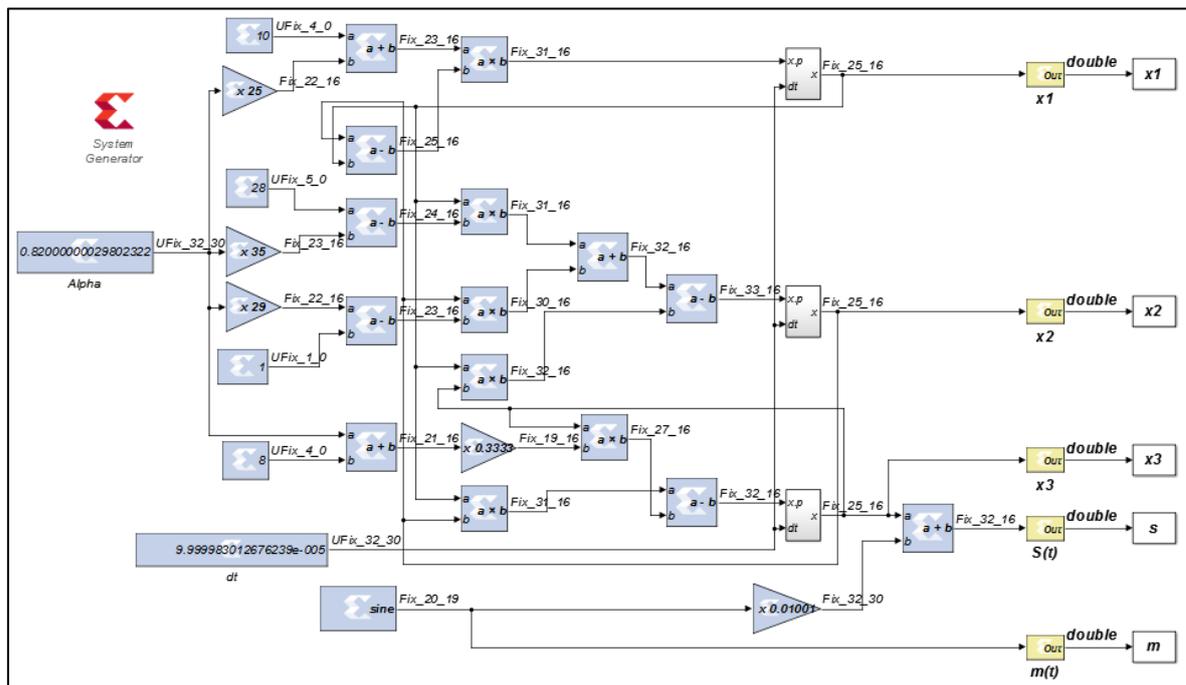


Figure 4.14 : Implémentation de l'émetteur chaotique

Pour le message à transmettre, nous avons utilisé un signal sinusoïdal généré par le bloc "DDS Compiler" qui est disponible dans la bibliothèque du system Generator.

La figure 4.15 représente l'implémentation de l'émetteur chaotique intégré avec le contrôleur du codec Ac'97.

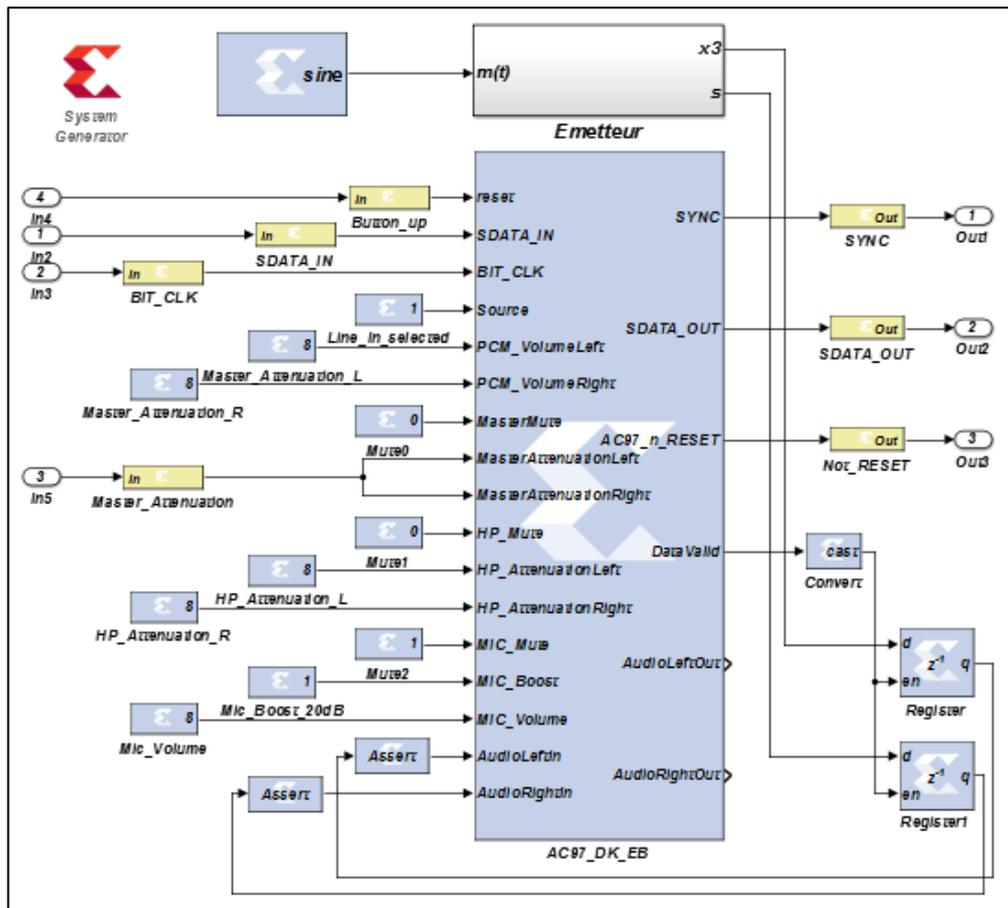


Figure 4.15 : Implémentation de l'émetteur chaotique avec le codec Ac'97

Les figures 4.16 et 4.17 représentent les signaux  $x_1(t)$  et  $x_3(t)$  : (a) simulés sous Matlab Simulink-System Generator, (b) récupérés par Co-Simulation matérielle, (c) relevés au niveau de la sortie du codec Ac'97 à l'aide d'un oscilloscope numérique. Ces résultats sont obtenus pour les paramètres suivants :  $\alpha = 0.82$ ,  $dt = 10^{-4}$  et  $X_0 = (-15, 10, 3)$ .

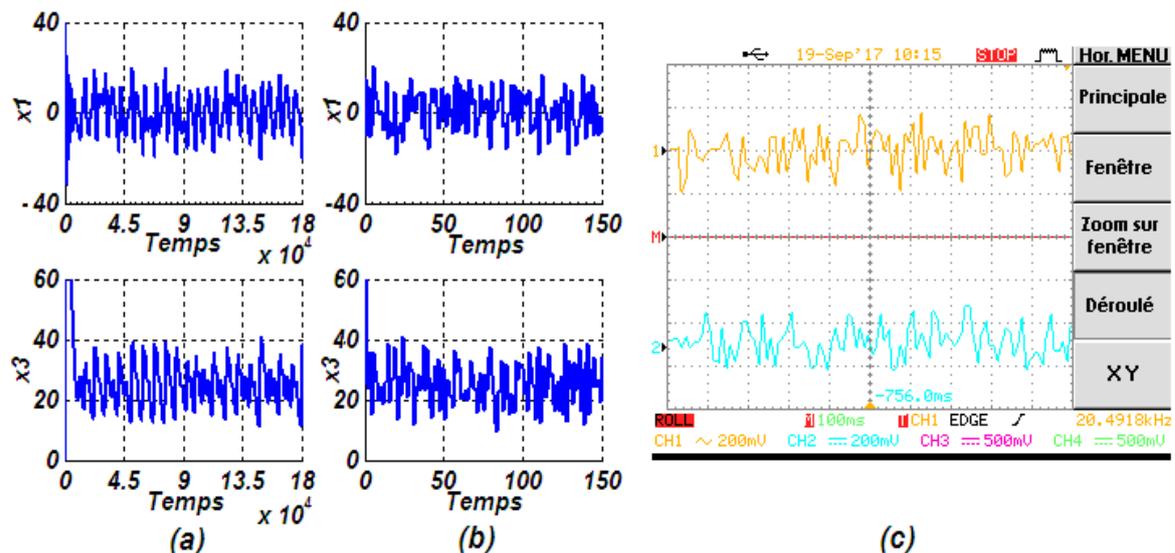
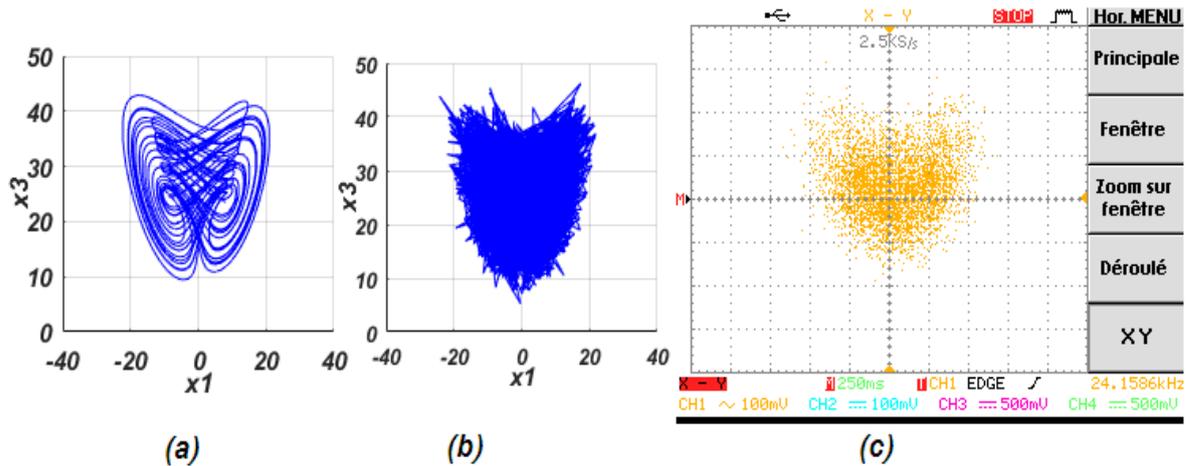
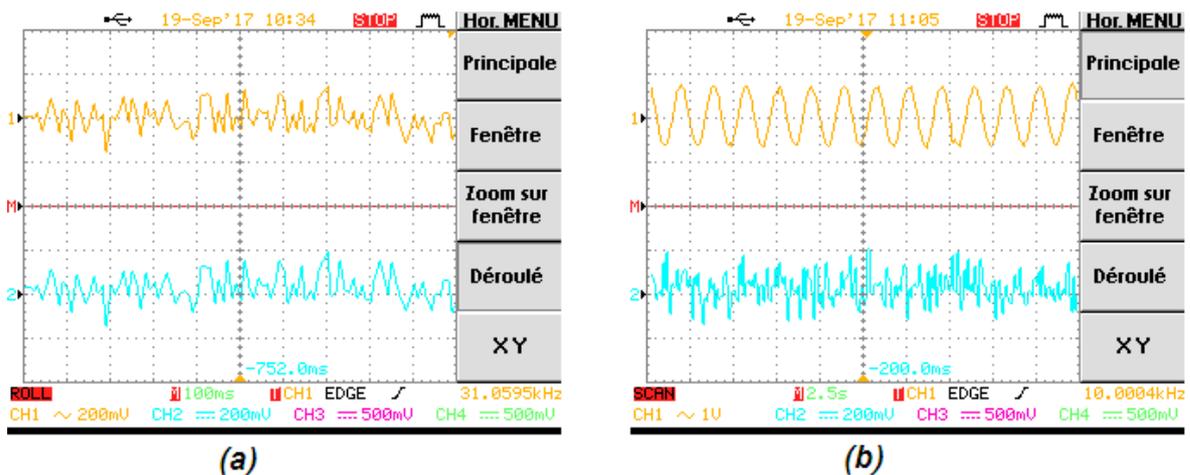


Figure 4.16 : Signaux  $x_1(t)$  et  $x_3(t)$  (a) simulés - (b) Co-Simulés - (c) expérimentaux



**Figure 4.17 :** Plan de phase XZ (a) simulé - (b) Co-Simulé - (c) expérimental

Ces figures montrent le bon fonctionnement de notre émetteur chaotique dont il est clair que l'insertion du message n'a pas modifié le comportement chaotique du système. La Figure 4.18 représente une comparaison entre : (a) la porteuse chaotique  $x_3(t)$  et le signal transmis  $s(t)$  et (b) le message à transmettre  $m(t)$  et le signal transmis  $s(t)$ . Cette figure montre bien que notre message est complètement noyé dans la porteuse chaotique dont on ne soupçonnera jamais l'existence d'un message utile à l'intérieur du signal  $s(t)$  ; il sera considéré comme un signal bruit.

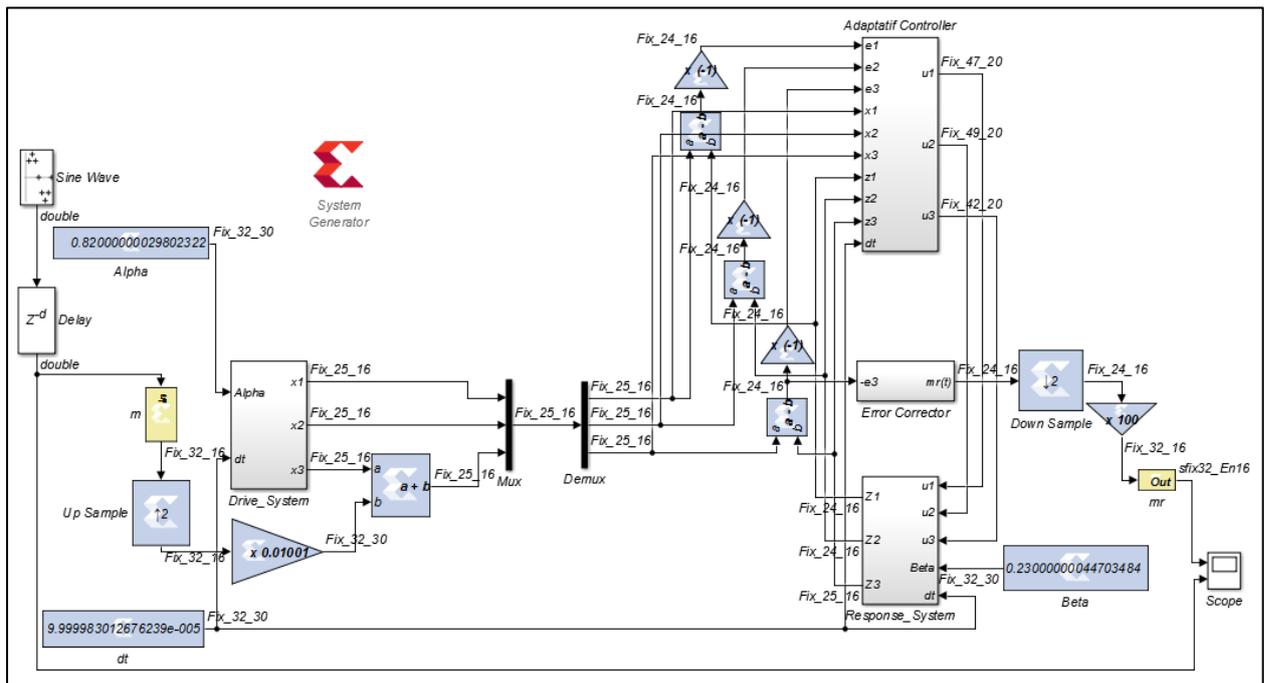


**Figure 4.18 :** Comparaison entre : (a) porteuse chaotique  $x_3(t)$  et signal transmis  $s(t)$  - (b) message à transmettre  $m(t)$  et signal transmis  $s(t)$

## 4.6 Implémentation de la transmission chaotique sur FPGA

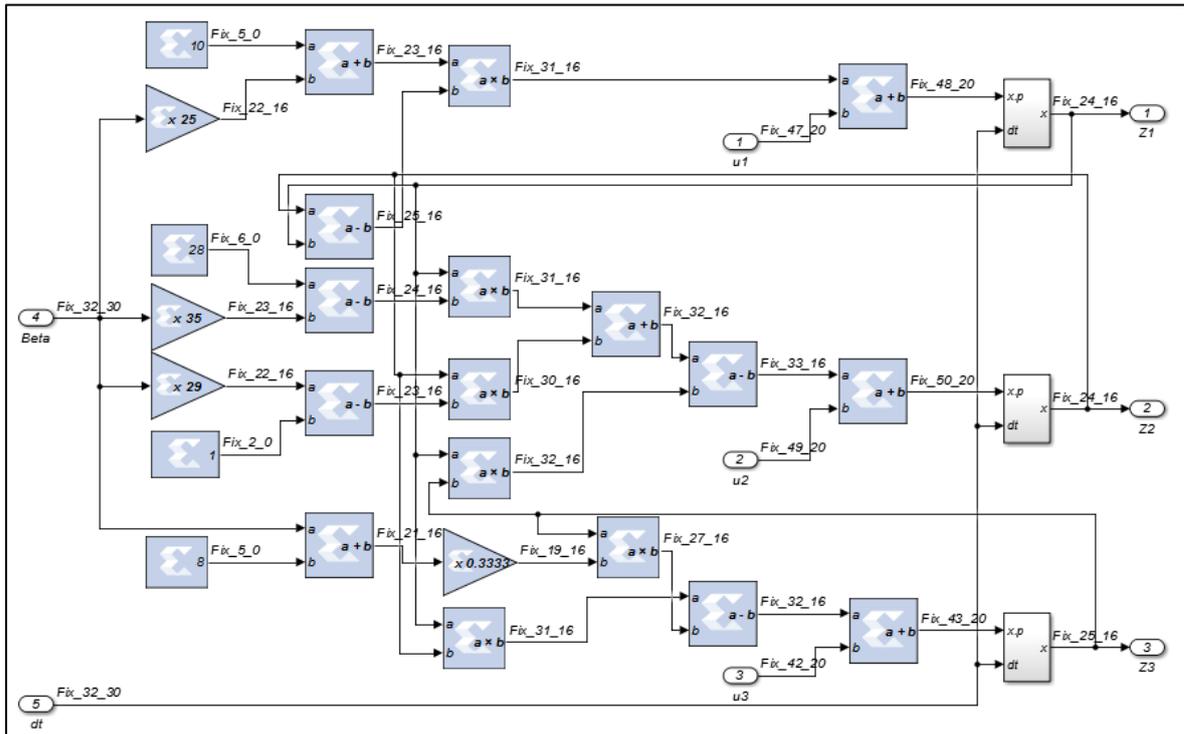
Dans cette section, nous présentons l'implémentation du système de transmission chaotique constitué de l'émetteur chaotique (le système chaotique Unifié maître incluant le message) et du récepteur chaotique (le système chaotique esclave, le contrôleur adaptatif et le correcteur d'erreur permettant la récupération du message) sur la plateforme de développement Xilinx ML501 en utilisant l'outil de Xilinx System Generator sous l'environnement Matlab Simulink. Les résultats de simulation et de Co-Simulation matérielle ainsi qu'une comparaison entre eux seront représentés en fin de section. Les résultats présentés dans cette partie sont obtenus pour les paramètres suivants :  $\alpha = 0.82$  ,  $\beta = 0.23$  ,  $dt = 10^{-4}$  ,  $X_0 = (-15, 10, 3)$  ,  $Z_0 = (3, -10, -20)$  ,  $T_{c_1} = 2 \times 10^4$  ,  $T_{c_2} = 2$  ,  $f = 1/100$  pour la sinusoïde et  $f = 1/256$  pour l'image.

La figure 4.19 représente le schéma global de l'implémentation du système de transmission chaotique sous l'environnement System Generator - Matlab Simulink.

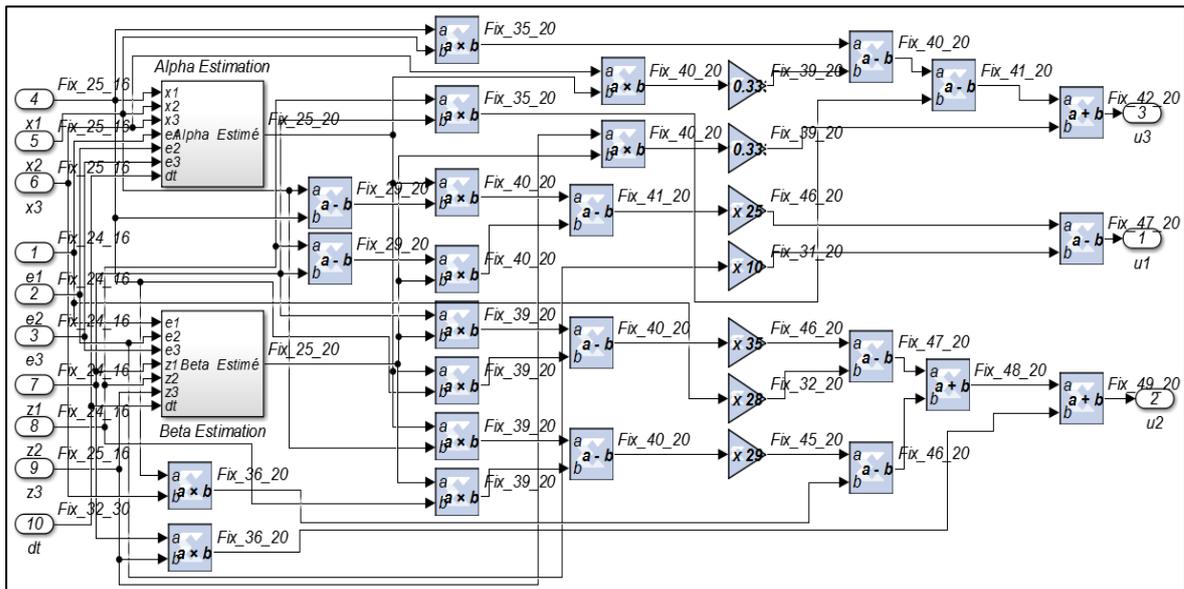


**Figure 4.19 :** Schéma global de l'implémentation de la transmission chaotique sous l'environnement System Generator - Matlab Simulink

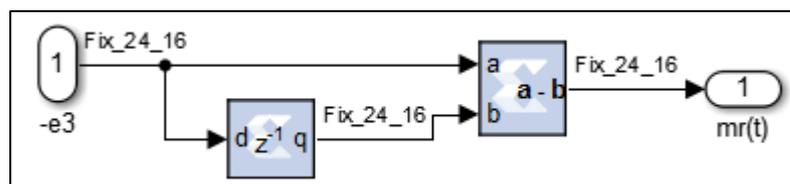
Les figures 4.20 à 4.24 représentent l'implémentation des différents blocs constituant le système de transmission chaotique qui sont respectivement : le système chaotique Unifié esclave, le contrôleur adaptatif, le correcteur d'erreur, les estimateurs de paramètres  $\hat{\alpha}$  et  $\hat{\beta}$ . La figure 4.25 représente le schéma de la Co-Simulation utilisé.



**Figure 4.20 :** Implémentation du système chaotique Unifié esclave sous l'environnement System Generator - Matlab Simulink



**Figure 4.21 :** Implémentation du contrôleur chaotique adaptatif sous l'environnement System Generator - Matlab Simulink



**Figure 4.22 :** Implémentation du correcteur d'erreur sous l'environnement System Generator - Matlab Simulink

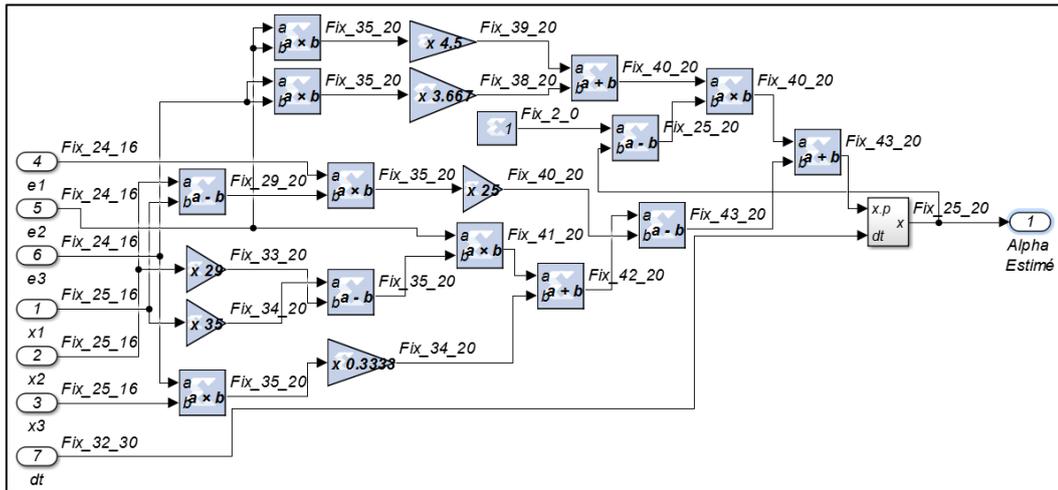


Figure 4.23 : Implémentation de l'estimateur du paramètre  $\hat{\alpha}$  sous l'environnement System Generator - Matlab Simulink

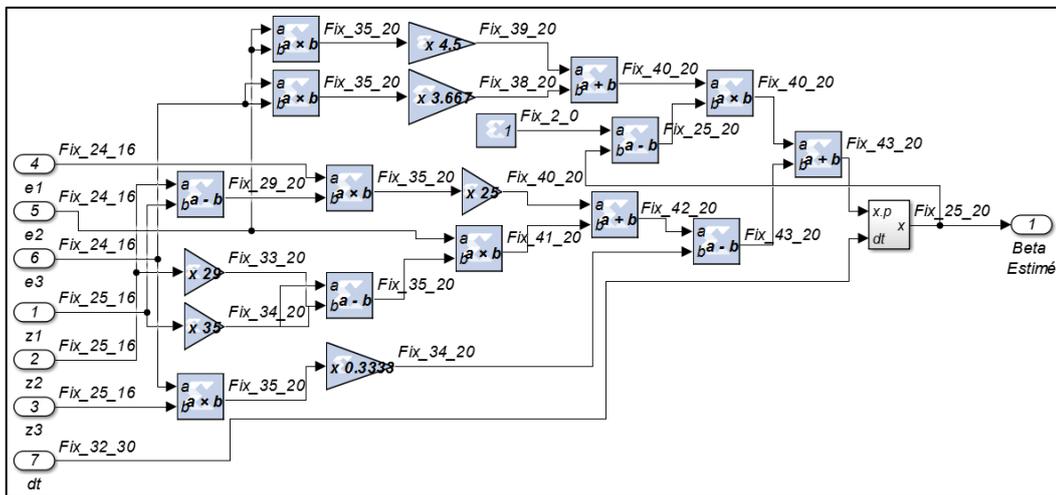


Figure 4.24 : Implémentation de l'estimateur du paramètre  $\hat{\beta}$  sous l'environnement System Generator - Matlab Simulink

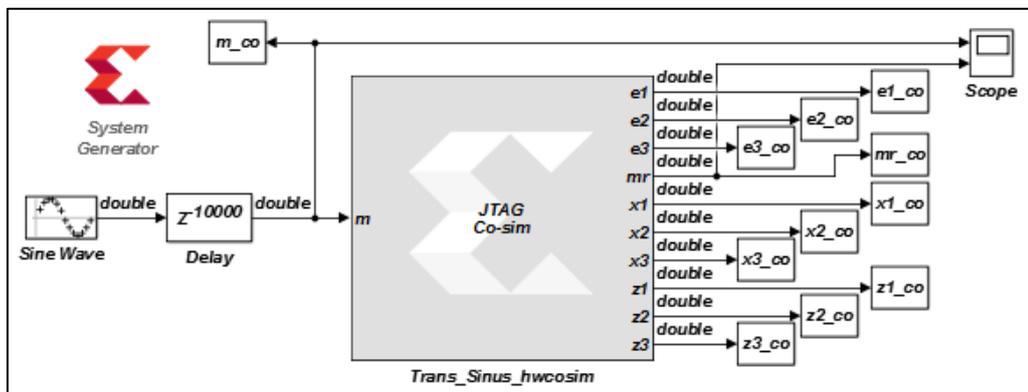
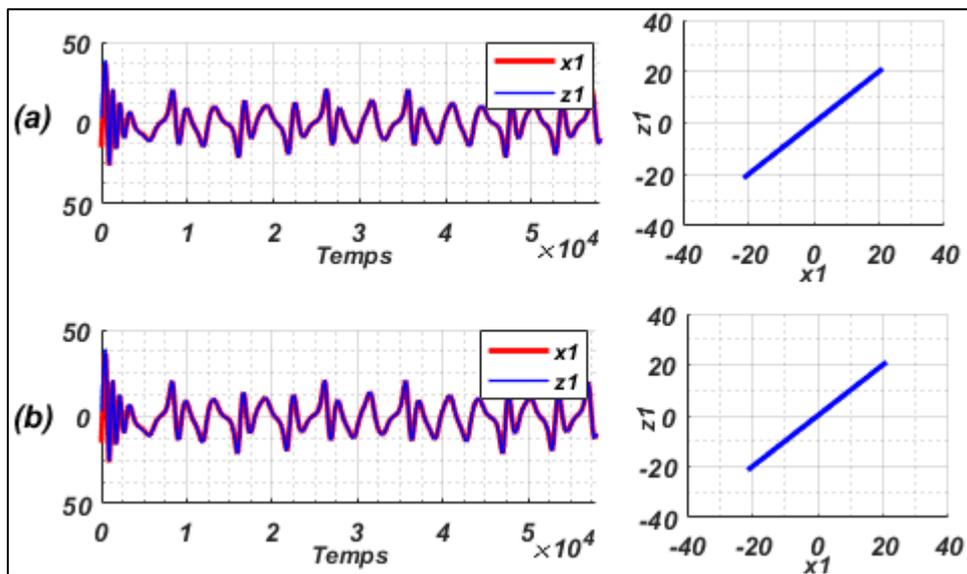


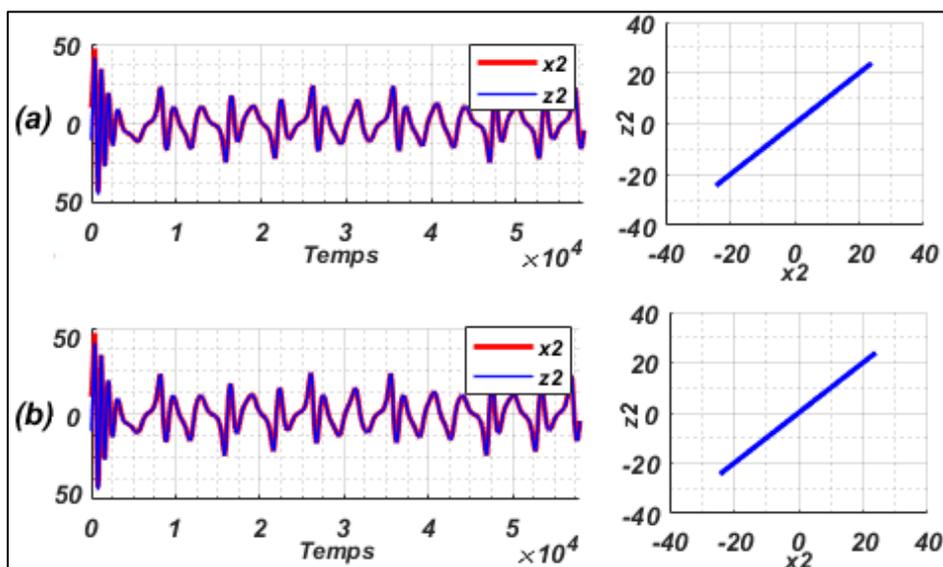
Figure 4.25 : Schéma de la Co-Simulation matérielle sous l'environnement System Generator - Matlab Simulink

Les figures 4.26 à 4.30 montrent les différents signaux simulés et récupérés par Co-Simulation depuis la carte FPGA. La visualisation de ces signaux nous permet de :

- ✓ Vérifier la synchronisation entre le récepteur et l'émetteur (figures 4.26 à 4.28).
- ✓ Évaluer l'erreur entre les signaux estimés au niveau du récepteur et ceux de l'émetteur (figure 4.29).
- ✓ Montrer la performance du système de transmission chaotique implémenté en comparant les signaux reconstitués au niveau du récepteur et ceux transmis par l'émetteur (figure 4.30).



**Figure 4.26 :** Signaux  $x_1$  et  $z_1$  récupérés par (a) simulation - (b) Co-Simulation



**Figure 4.27 :** Signaux  $x_2$  et  $z_2$  récupérés par (a) simulation - (b) Co-Simulation

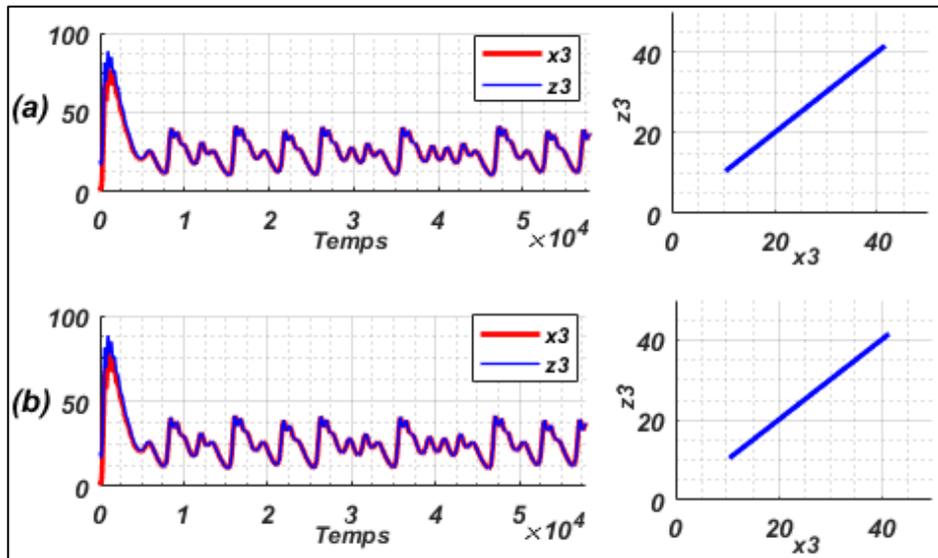


Figure 4.28 : Signaux  $x_3$  et  $z_3$  récupérés par (a) simulation - (b) Co-Simulation

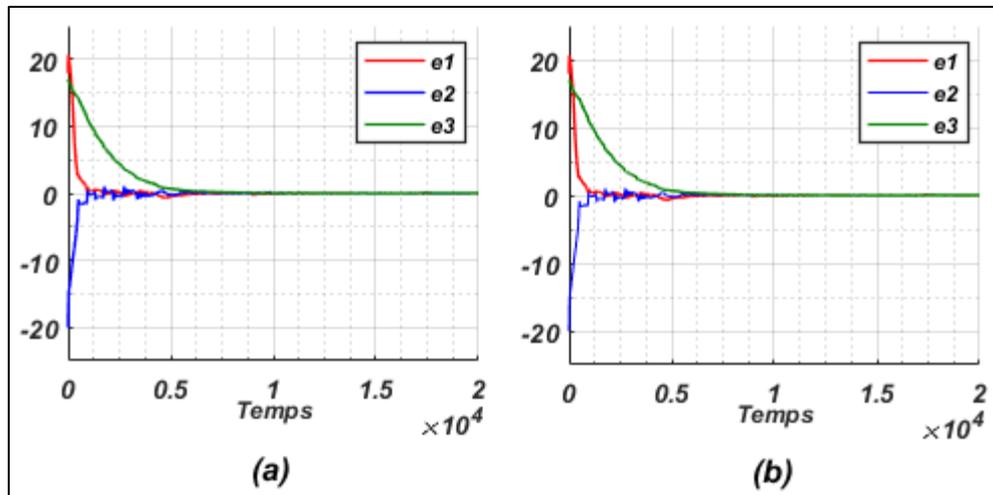


Figure 4.29 : Erreurs de synchronisation  $e_i = z_i - x_i$  ( $i = 1, 2, 3$ ) récupérés par (a) simulation - (b) Co-Simulation

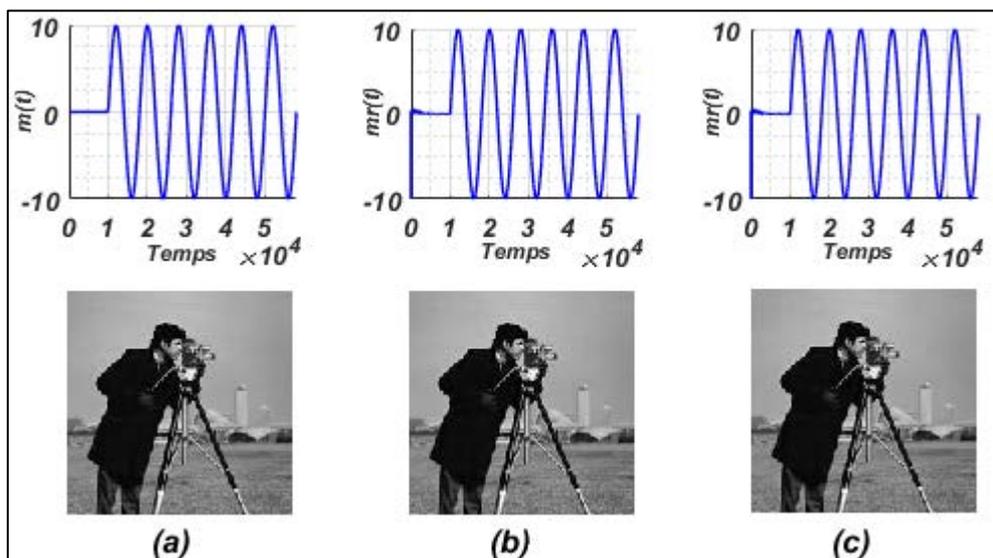


Figure 4.30 : Message reconstitué (a) original - (b) simulation - (c) Co-Simulation

## 4.7 Conclusion

Ce chapitre a été consacré à la description de l'implémentation sur circuit FPGA, du système de transmission chaotique proposé dans le chapitre précédent sous l'environnement Xilinx System Generator – Matlab Simulink. Après une brève présentation de la technologie FPGA et du flot de conception d'une implémentation FPGA et celui du System Generator, nous avons décrit la plateforme de développement utilisée Xilinx ML501 et ses différents composants. Par la suite nous avons présenté l'implémentation de l'émetteur chaotique sur la carte FPGA ; les signaux récupérés par simulation, Co-Simulation et ceux relevés à la sortie du codec Ac'97 à l'aide d'un oscilloscope numérique ont montré que le message est bien dissimulé dans le signal chaotique de l'émetteur, ainsi un intrus ne soupçonnera jamais l'existence d'un message utile dans le signal qui va être considéré comme un signal bruit. En fin de chapitre, l'implémentation de l'ensemble du système de transmission chaotique (émetteur-récepteur) est présentée. Les signaux récupérés par simulation et Co-Simulation ont montré une bonne synchronisation entre les états du récepteur chaotique et ceux de l'émetteur. Le message reconstitué à la sortie du récepteur présente une forte ressemblance à celui d'origine.

# CONCLUSION GÉNÉRALE

---

Dans ce travail, nous avons apporté des améliorations sur le système de transmission chaotique proposé par C.J. Cheng [5], basé sur une synchronisation adaptative entre deux différents systèmes chaotiques Unifiés dont les paramètres sont inconnus. En se basant sur la théorie de Lyapunov, Cheng a construit un contrôleur adaptatif dont l'objectif est de piloter le système chaotique Unifié du récepteur afin qu'il se synchronise avec l'émetteur ; cette synchronisation sera exploitée pour reconstruire le message transmis par l'émetteur. Notre contribution consiste à apporter des modifications au niveau dudit contrôleur avec un changement dans l'approche utilisée pour la récupération du message. Les résultats de la simulation sous Matlab Simulink du modèle proposé ont montré le bon fonctionnement du système.

Dans ce travail, nous avons également effectué une implémentation sur circuit FPGA (plateforme de développement Xilinx ML501) du système de transmission chaotique. L'implémentation de l'émetteur a montré via les signaux de simulation, de Co-Simulation matérielle et ceux récupérés depuis la sortie du codec Ac'97 de la carte FPGA, que le message a été bien crypté. La Co-Simulation matérielle de l'ensemble du système de transmission chaotique (émetteur-récepteur) a mis en évidence une forte synchronisation entre les états du récepteur chaotique et ceux de l'émetteur ainsi qu'une bonne qualité de réception.

Notre travail est présenté en quatre chapitres. Le premier chapitre a été consacré à l'étude des systèmes dynamiques, en particulier les systèmes dynamiques chaotiques et ses principales caractéristiques, tout en mettant en évidence l'intérêt du calcul des exposants de Lyapunov ainsi que la stabilité au sens de Lyapunov (directe et indirecte). Dans le deuxième chapitre, nous avons présenté le système chaotique Unifié, ses caractéristiques et ses propriétés ainsi que l'analyse des systèmes chaotiques qui le constituent (Lorenz, Chen et Lü). Les résultats de simulation de ces systèmes sont également présentés. Dans le troisième chapitre, après une présentation des différentes

méthodes de synchronisation ainsi que les systèmes de transmission chaotiques, nous avons décrit le processus de conception du système de transmission chaotique proposé. Les résultats de la simulation du système relative à la synchronisation, la transmission et la reconstitution du message ont présentés ainsi que l'influence d'un Bruit Blanc Gaussien Additif sur le système. Dans le dernier chapitre, une implémentation du système de transmission chaotique proposé a été réalisée sur circuit FPGA. Les résultats de simulation, de Co-Simulation expérimentaux ont été concluants.

Ce travail nous a été très bénéfique ; nous avons ainsi acquis énormément de connaissances, dans divers domaines : réflexion et étude de nouveaux systèmes, management et gestion du projet, recherche bibliographique, maîtrise de nouveaux outils et langages de programmation, techniques de rédaction, etc. Toutefois, malgré tous les efforts que nous avons fournis, le projet reste ouvert pour des enrichissements. Comme perspectives :

- ✓ Nous envisageons une étude d'amélioration de la robustesse du système de transmission chaotique proposé contre le bruit du canal de transmission, en intégrant l'estimation du bruit du canal dans les équations régissant le contrôleur adaptatif.
- ✓ L'ensemble du système de transmission présenté (émetteur-récepteur) est implémenté sur le même circuit FPGA ; nous recommandons de faire une implémentation séparée sur deux cartes FPGA indépendantes de l'émetteur et du récepteur.

### PROGRAMME DÉVELOPPÉ SOUS MATLAB SIMULANT LES SYSTÈMES CHAOTIQUES

Ce programme consiste à simuler les systèmes chaotiques (Lorenz, Rössler, Chen, Lü et Unifié) dans ces propriétés : réponse temporelle, attracteurs, plans de phases et diagrammes de bifurcations. Il se compose d'une procédure et trois fonctions :

- ✓ **Visualisation** : c'est le programme principal. En faisant appel aux autres fonctions, cette procédure permet l'initialisation des différents paramètres des systèmes chaotiques, la présentation des différents résultats à l'utilisateur et elle fournit à l'utilisateur via la fenêtre de commande Matlab la possibilité de choisir entre : intégration simple (réponse temporelle, attracteurs et plans de phases), bifurcation ou les deux d'un système chaotique sélectionné.
- ✓ **integration** : cette fonction permet la résolution du système d'équations différentielles du système chaotique en utilisant la méthode de Runge Kutta 4 à pas variable **ODE45**.
- ✓ **bifurcation** : cette fonction permet la détermination du diagramme de bifurcation du système chaotique.
- ✓ **SysModel** : cette fonction permet de définir le système d'équations différentielles du système chaotique. Elle est appelée par les fonctions **integration** et **bifurcation**.

## Procédure : visualisation

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ce programme permet de générer le diagramme de bifurcation & visualise%
%   les attracteurs étranges d'un système chaotique talque:           %
%       LORENZ & RÖSSLER & CHEN & Lü & UNIFIED SYSTEM.             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
clear all
clc
%%+++++++
%% choix du système chaotique à simulé
SysModel_ID =input('Entrer le Sytème chaotique à simulé: ...\n'...
                  ' 1 pour LORENZ\n'...
                  ' 2 pour RÖSSLER\n'...
                  ' 3 pour CHEN\n'...
                  ' 4 pour Lü\n'...
                  ' 5 pour UNIFIED SYSTEM\n'...
                  '...\n'...
                  ');
%%+++++++
%% paramétrage global
options = [];           % Option d'intégration (Matlab_ode45)

switch SysModel_ID

    case 1               % LORENZ SYSTEM
        %%
        fprintf('\n          ... LORENZ SYSTEM SIMULATION ...\n\n')
        %% Paramètres d'intégration
        TimeBigin = 0;   % Start TIME
        TimeEnd = 120;   % End TIME
        sigma = 10;
        r = 28;
        b = 8/3;
        InitCond = [0 1 0];
        %% Paramètres de bifurcation
        Parametres = [sigma r b];
        IndiceParam = 2;
        RangeParam = [0 40];
        StepParam = 0.1;

    case 2               % ROSSLER SYSTEM
        %%
        fprintf('\n          ... ROSSLER SYSTEM SIMULATION ...\n\n')
        %% Paramètres d'intégration
        TimeBigin = 0;   % Start TIME
        TimeEnd = 500;   % End TIME
        a = 0.38;
        b = 0.3;
        c = 4.82;
        InitCond = [0.01 0.01 0.01];
        %% Paramètres de bifurcation
        Parametres = [a b c];
        IndiceParam = 3;
        RangeParam = [1 40];
        StepParam = 0.1;
```

```

case 3                                % CHEN SYSTEM
%%
fprintf('\n                ... CHEN SYSTEM SIMULATION ... \n\n')
%% Paramètres d'intégration
TimeBigin = 0;                        % Start TIME
TimeEnd = 100;                        % End TIME
a = 35;
c = 28;
b = 3;
InitCond = [0 1 0];
%% Paramètres de bifurcation
Parametres = [a c b];
IndiceParam = 2;
RangeParam = [20 34];
StepParam = 0.1;

case 4                                % LU SYSTEM
%%
fprintf('\n                ... LU SYSTEM SIMULATION ... \n\n')
%% Paramètres d'intégration
TimeBigin = 0;                        % Start TIME
TimeEnd = 100;                        % End TIME
a = 36;
c = 28;
b = 3;
InitCond = [0 1 0];
%% Paramètres de bifurcation
Parametres = [a c b];
IndiceParam = 2;
RangeParam = [0 30];
StepParam = 0.1;

case 5                                % UNIFIED SYSTEM
%%
fprintf('\n                ... UNIFIED SYSTEM SIMULATION ... \n\n')
%% Paramètres d'intégration
TimeBigin = 0;                        % Start TIME
TimeEnd = 50;                          % End TIME
alpha = 0.8;
InitCond = [0 1 0];
%% Paramètres de bifurcation
Parametres = alpha;
IndiceParam = 1;
RangeParam = [0 1];
StepParam = 0.005;

otherwise
%%
fprintf(['\n\n Ce systeme n existe pas.'...
        'verifiez et recompilez ... \n']);
return
end

%+++++
%% choix visualisation des attracteurs ou du diagramme de bifurcation
%+++++
BufAttract = input(['\n\nvoulez vous visualiser quoi? ... \n'...
                   ' 1 pour Attracteurs étrange ... \n'...
                   ' 2 pour Diagramme de bifurcation ... \n'...
                   ' 0 pour les deux \n'...

```

```

        ' ...\n'...
        ']);
%%

switch BufAttract

case 0
    %%
    % visualisation des attracteurs étrange
    %+++++
    % Resolution des équations d'états
    [t,x] = integration(SysModel_ID,InitCond,TimeBigin,...
                        TimeEnd,options,Parametres);

    % Visualisation des résultats
    %% Réponse temporelle
        figure('name','Réponse temporelle','color','w');
        subplot(3,1,1);
        plot(t,x(:,1));
        title('x1(t)');
        subplot(3,1,2);
        plot(t,x(:,2));
        title('x2(t)');
        subplot(3,1,3);
        plot(t,x(:,3));
        title('x3(t)');

    %% Attracteur étrange
        figure('name','Attracteur et plan de Phase','color','w');
        subplot(2,2,1);
        plot3(x(:,1),x(:,2),x(:,3));
        title('Attracteur étrange');

    %% Plans de Phases
        subplot(2,2,2);
        plot(x(:,1),x(:,2));
        title('Plan de Phase XY');
        subplot(2,2,3);
        plot(x(:,1),x(:,3));
        title('Plan de Phase XZ');
        subplot(2,2,4);
        plot(x(:,2),x(:,3));
        title('Plan de Phase YZ');

    %+++++
    %%
    % visualisation du diagramme de bifurcation
    %+++++

    D = bifurcation(SysModel_ID,InitCond,TimeBigin,TimeEnd,...
                    options,Parametres,IndiceParam,RangeParam,StepParam);

    figure('name','Diagramme de bifurcation','color','w');
    plot(D(:,1),D(:,2),'ro','MarkerEdgeColor','b',...
         'MarkerFaceColor','g','MarkerSize',1.5);
    title('Diagramme de bifurcation');
    %+++++

```

```

case 1
%%
% visualisation des attracteurs étrange
%+++++
% Resolution des équations d'états
[t,x] = integration(SysModel_ID,InitCond,TimeBigin,...
                    TimeEnd,options,Parametres);
% Visualisation des résultats

%% Réponse temporelle
figure('name','Réponse temporelle','color','w');
subplot(3,1,1);
plot(t,x(:,1));
title('x1(t)');
subplot(3,1,2);
plot(t,x(:,2));
title('x2(t)');
subplot(3,1,3);
plot(t,x(:,3));
title('x3(t)');

%% Attracteur étrange
figure('name','Attracteur et plan de Phase','color','w');
subplot(2,2,1);
plot3(x(:,1),x(:,2),x(:,3));
title('Attracteur étrange');

%% Plans de Phases
subplot(2,2,2);
plot(x(:,1),x(:,2));
title('Plan de Phase XY');
subplot(2,2,3);
plot(x(:,1),x(:,3));
title('Plan de Phase XZ');
subplot(2,2,4);
plot(x(:,2),x(:,3));
title('Plan de Phase YZ');
%+++++

case 2
%%
% visualisation du diagramme de bifurcation
%+++++
D = bifurcation(SysModel_ID,InitCond,TimeBigin,TimeEnd,...
                options,Parametres,IndiceParam,RangeParam,StepParam);

figure('name','Diagramme de bifurcation','color','w');
plot(D(:,1),D(:,2),'ro','MarkerEdgeColor','b',...
     'MarkerFaceColor','g','MarkerSize',1.5);
title('Diagramme de bifurcation');

otherwise
%%
fprintf('\n\n mauvais choix. verifiez et recompilez ...\n');
return
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               FIN                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

## Fonction : integration

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                integration                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [t,x]=integration(SysModel_ID,InitCond,TimeBigin,TimeEnd,...
    options,Parametres)
    %% trouver la nouvelle condition initiale si TimeBigin > 0
    if TimeBigin > 0
        [t x] = ode45(@SysModel,[0 TimeBigin],InitCond,...
            options,SysModel_ID,Parametres);
        InitCond=x(end,:);
    End

    %% Integration

        [t x] = ode45(@SysModel,[TimeBigin TimeEnd],InitCond,...
            options,SysModel_ID,Parametres);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                FIN                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## Fonction : bifurcation

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                bifurcation                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function D = bifurcation(SysModel_ID,InitCond,TimeBigin,TimeEnd,...
    options,Parametres,IndiceParam,RangeParam,StepParam)
%%
    D=[]; % data (bifurcation diagram)

    for b= RangeParam(1):StepParam:RangeParam(2)
        Parametres(IndiceParam) = b;
        fprintf('Param=%g...\n',b);

        [t,x] = integration(SysModel_ID,InitCond,TimeBigin,...
            TimeEnd,options,Parametres);

        for i=2:length(x(:,1))-1
            if((x(i,1)>x(i-1,1))&&(x(i,1)>x(i+1,1)))
                D=[D; b x(i,1)];
            end
        end
    end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                FIN                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## Fonction : SysModel

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%                               SysModel                               %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
function y = SysModel(t,x,SysModel_ID,Parametres)  
  
    switch SysModel_ID  
        case 1          % LORENZ SYSTEM  
            y = [Parametres(1)*(x(2)-x(1));x(1)*(Parametres(2)-x(3))-...  
                x(2);-Parametres(3)*x(3)+x(1)*x(2)];  
        case 2          % ROSSLER SYSTEM  
            y = [-x(2)-x(3);x(1)+Parametres(1)*x(2);Parametres(2)*...  
                x(1)-Parametres(3)*x(3)+x(1)*x(3)];  
        case 3          % CHEN SYSTEM  
            y = [Parametres(1)*(x(2)-x(1));x(1)*(Parametres(2)-...  
                Parametres(1)+Parametres(2)*x(2)-x(1)*x(3);...  
                -Parametres(3)*x(3)+x(1)*x(2)];  
        case 4          % LU SYSTEM  
            y = [Parametres(1)*(x(2)-x(1));Parametres(2)*x(2)-x(1)*...  
                x(3);-Parametres(3)*x(3)+x(1)*x(2)];  
        case 5          % UNIFIED SYSTEM  
            y = [(25*Parametres(1)+10)*(x(2)-x(1));(28-35*...  
                Parametres(1))*x(1)-x(1)*x(3)+(29*Parametres(1)-1)*...  
                x(2) ; x(1)*x(2)-((Parametres(1)+8)/3)*x(3)];  
    end  
end  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%                               FIN                               %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## BIBLIOGRAPHIE

---

- [1] E. N. LORENZ, "Deterministic Nonperiodic Flow", Journal Of The Atmospheric Sciences, vol. 20, pp. 130-141, 1963.
- [2] E. N. LORENZ, "The essence of chaos", University of Washington Press: The Jessie and John Danz Lecture Series, 1993.
- [3] L.M. PECORA, T. L. CARROLL, "Synchronization in Chaotic Systems", Physical Review Letters, vol. 64, n° 8, pp. 821-824, 1990.
- [4] L.M. PECORA, T. L. CARROLL, "Driving systems with chaotic signals", Physical Review A, vol. 44, n° 4, pp. 2374-2383, 1991.
- [5] C.J CHENG, " Robust synchronization of uncertain unified chaotic systems subject to noise and its application to secure communication" , Applied Mathematics and Computation, vol. 219, pp. 2698-2712, 2012.
- [6] G. KADDOUM, "Contributions à l'amélioration des systèmes de communication multi-utilisateurs par chaos, Synchronisation et analyse des performances", Thèse de doctorat, Université de Toulouse, France, 2008.
- [7] M. L'HERNAULT-ZANGANEH, "Faisabilité d'un système d'émission-réception analogique pour les communications sécurisées par le chaos", Thèse de doctorat, Université Paris 6, France, 2007.
- [8] O. MEGHERBI, "Etude et Réalisation d'un système sécurisé à base de systèmes chaotiques", Thèse de magister, Université Mouloud MAMMERI TIZI-OUZOU, Algérie, 2013.
- [9] M.L. CHIKHI, "Application des systèmes dynamiques chaotiques en transmission de données", Thèse de magister, Université de Saad DAHLEB BLIDA, Algérie, 2012.
- [10] P. ETCHECOPAR, "Quelques éléments sur la théorie du chaos", Note de cours de Cégep de Rimouski, Québec.

- [11] M.A. DJENOURI, M.H. CHIKHI, "Communication sécurisée par chaos : Etude et implémentation sur carte FPGA", Mémoire de master, Université Saad DAHLEB Blida, Algérie, 2014.
- [12] M. TIDJANI, "Synchronisation des systèmes dynamiques chaotiques à dérivées fractionnaires", Thèse de magister, Université MENTOURI Constantine, Algérie.
- [13] V. MAHOUT, "Théorie de Lyapunov pour les systèmes autonomes" Note de cours INSA Toulouse, France.
- [14] L. GUESSAS, "Backstepping Backstepping adaptatif pour le contrôle, la poursuite et la synchronisation des systèmes dynamiques non linéaires chaotiques", Thèse de doctorat, Université Ferhat ABBAS Sétif, Algérie, 2012.
- [15] P. GASPARD, "Rossler systems", Encyclopedia of Nonlinear Science, New York, 2005, pp. 808-811.
- [16] J. Lu, X. Wu et J. Lü, "Synchronization of a unified chaotic system and the application in secure communication", Physics Letters A, vol. 305, pp. 365-370, 2002.
- [17] J. Lü, G. CHEN, D. CHENG et S. CELIKOVSKY, "Bridge The Gap Between The Lorenz System And The Chen System", International Journal of Bifurcation and Chaos, vol. 12, n° 12, pp. 2917-2926, 2002.
- [18] G. CHEN, T. UETA, "Yet Another Chaotic Attractor", International Journal of Bifurcation and Chaos, vol. 9, n° 7, pp. 1465-1466, 1999.
- [19] T. UETA, G. CHEN, "Bifurcation Analysis Of Chen's Equation", International Journal of Bifurcation and Chaos, vol. 10, n° 8, pp. 1917-1931, 2000.
- [20] A. VANECK, S. CELIKOVSKÝ, "Control Systems : From Linear Analysis to Synthesis of Chaos", Prentice-Hall, London, 1996.
- [21] S. CELIKOVSKÝ, G. CHEN, "On A Generalized Lorenz Canonical Form Of Chaotic Systems" International Journal of Bifurcation and Chaos, vol. 12, n° 8, pp. 1789-1812, 2002.
- [22] J. LÜ, G. CHEN "A New Chaotic Attractor Coined", International Journal of Bifurcation and Chaos, vol. 12, n° 3, pp. 659-661, 2002.
- [23] Lj. KOCAREV, K.S. HALLE, K. ECKERT, L.O. CHUA et U. PARLITZ, "Experimental Demonstration Of Secure Communications Via Chaotic Synchronisation", International Journal of Bifurcation and Chaos, vol. 2, n° 3, pp. 709-713, 1992.

- [24] U. PARLITZ, L.O. CHUA, Lj. KOCAREV, K.S. HALLE et A. SHANG, "Transmission Of Digital Signals By Chaotic Synchronization", International Journal of Bifurcation and Chaos, vol. 2, n° 4, pp. 973-977, 1992.
- [25] T. YANG, L.O. CHUA, "Secure Communication via Chaotic Parameter Modulation", IEEE Trans. on Circ. Sys. I, vol. 43, n° 9, pp. 817-819, 1996.
- [26] H. DIMASSI, "Synchronisation des systèmes chaotiques par observateurs et applications à la transmission d'informations", Thèse de doctorat, Université Paris Sud XI – Université Tunis El-Manar, France-Tunisie, 2013.
- [27] L.O. CHUA, Lj. KOCAREV, K. ECKERT et M. ITOH, "Experimental Chaos Synchronization In Chua's Circuit", International Journal of Bifurcation and Chaos, vol. 2, n° 3, pp. 705-708, 1992.
- [28] Lj. KOCAREV, A. SHANG et L.O. CHUA, "Transitions in dynamical regimes by driving: a unified method of control and synchronization of chaos", International Journal of Bifurcation and Chaos, vol. 3, n° 2, pp. 479-483, 1993.
- [29] G. MYKOLAITIS, A. TAMASEVICIUS et S. BUMELIENE, "Experimental demonstration of chaos from colpitts in VHF and UHF ranges", Electronics letters, vol. 40, n° 2, 2004.
- [30] H. HAMICHE, "Inversion à Gauche des Système Dynamiques Hybrides Chaotique: Application à la Transmission Sécurisée de Données", Thèse de doctorat, Université Mouloud MAMMERI TIZI-OUZOU, Algérie, 2011.
- [31] C. BENHABIB, "Etude D'un Systeme Chaotique Pour La Sécurisation Des Communications Optiques", Mémoire de master, Université Telemcen, Algérie, 2014.
- [32] M.T. YASSEN, "Adaptive control and synchronization of a modified Chua's circuit system", Applied Mathematics and Computation, vol. 135, n° 1, pp. 113-128, 2003.
- [33] M.T YASSEN, "Chaos synchronization between two different chaotic systems using active control", Chaos : Solution & Fractals, vol. 23, n° 1, pp. 131-140, 2005.
- [34] S. AMEN, P. BILOKON, A. BRINLEY CODD, M. FOFARIA et T. SHAH, "Numerical Solutions of Differential Equations, Londres" Imperial College, London, 2004.
- [35] D. SAPTONO, "Conception d'un outil de prototypage rapide sur le FPGA pour des applications de traitement d'images", UNIVERSITE DE BOURGOGNE , France, 2011.
- [36] A. GUETTAT, "Conception et Implémentation d'un Corrélateur Numérique sur FPGA", Thèse de magister, Université des Sciences et de la Technologie d'Oran Mohamed BOUDIAF, Algérie, 2012.

- [37] M. DARDAILLON, "Introduction aux FPGA", Notes de cours de INSA Lyon, France, 2011.
- [38] Institut Indien de Technologie de Kharagpur : Department of Electrical Engineering, "Design of Embedded Processors", Note de cours, Inde.
- [39] Xilinx Inc, "SYSTEM GENERATOR FOR DSP User Guide v 13.4", 2012.
- [40] K. MESSAOUDI, "Traitement des Signaux et Images en Temps Réel - Implantation de H.264 sur MPSoC", Thèse de doctorat, Université BADJI MOKHTAR ANNABA - Université de BORGOGNE, Algérie, 2012.
- [41] Xilinx Inc, "ML501 Evaluation Platform - User Guide V1.4", 2009.
- [42] ANALOG DEVICES, "AC'97 SoundMAX Codec - AD1981B Data Sheet Rev.C".
- [43] National Semiconductor, "LM4550 AC '97 Rev 2.1 Multi-Channel Audio Codec with Stereo Headphone Amplifier, Sample Rate Conversion and National 3D Sound", 2004.
- [44] J.R. SANCHEZ MAYEN, «Implementing an AC97Audio Controller IP,» Mémoire de Master à l'université de Gothenburg, Suède, 2011.