الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Projet de Fin d'Études

présenté par

DJENOURI Mohamed Amine

&

CHIKHI Mohamed Hichem

Pour l'obtention du diplôme de Master en Électronique

Spécialité : Télécommunications et réseaux.

Thème

Communication sécurisée par chaos : Etude et implémentation sur carte FPGA

Proposé par : Mr. CHIKHI Mohamed Lazhar

Année Universitaire 2013-2014

Remerciements

Nous tenons à remercier tout d'abord DIEU qui nous a donné durant toutes ces années la santé, le courage et la patience pour en arriver là.

Nous remercions très chaleureusement notre promoteur Mr.CHIKHI Mohamed Lazhar pour avoir dirigé nos travaux. Merci pour vos échanges scientifiques, vos conseils et votre rigueur.

Nous tenons à exprimer notre profonde gratitude à tous les enseignants de la spécialité réseaux et télécommunications et le Professeur M. BENSEBTI le responsable de la spécialité en particulier.

Nous voudrions aussi remercier Mr. FERDJOUNI et Mr. BOUNEKHLA et toute l'équipe du laboratoire qui nous a apporté leur soutien tout au long de ce travail de thèse.

Nous remercions également tous les membres du jury pour nous avoir honoré par leur présence et pour avoir accepté d'évaluer ce travail de mémoire.

Nous tenons aussi à remercier nos parents respectifs, nos frères et sœurs sans oublier nos amis.

Enfin, nous remercions tous ceux qui ont participé de près ou de loin à l'achèvement de ce travail.

Dédicace :

Je dédie ce modeste travail à mes très chers parents pour l'éducation qu'ils m'ont prodigué ainsi que tous leurs sacrifices.Vous avez toujours tout fait pour me préserver une part de bonheur et de réconfort. Que Dieu me vienne en aide pour vous prouver ma sincère gratitude, à ma très chère sœur : « Mimi » que j'aime de tout mon cœur que dieu vous protège tous.

A toute la famille qui m'ont soutenu durant les moments difficiles. A mes oncles et surtout Mohamed Lazhar, artisan de ma réussite et sa petite famille. A mes très chères tantes paternelles et maternelles. A tous mes cousins et cousines merci du fond du cœur.

A mon binôme « Mohamed Amine » avec qui j'ai passé de très bons moments.

Au groupe Master 2 R&T et tous les étudiants avec qui j'ai parcouru mon cursus universitaire et a tous mes amis.

CHIKHI Mohamed Hichem

Dédicace

C'est avec un énorme plaisir, un cœur ouvert et une immense joie que je dédie ce modeste travail à :

- Mes très chers et magnifiques parents qui m'ont bien soutenu et aidé tout au long de mon parcours jusqu'à en arriver jusqu'ici.
- Ma sœur « Meriem » et son mari « Farid » sans oublier leurs petit garçon « Youcef » ;
- Mon frère « Khaled » et sa femme « Nesrine » ainsi que leurs bébé «Nihal» ;
- Toute la famille, grands et petits ;
- « Lily », une personne très chère pour moi et qui m'a vraiment soutenu dans les moments difficiles et qui a été tous le temps présente pour me remonter le moral d'une manière ou d'une autre ainsi qu'à sa cousine « Rym» qui m'a tout autant aidé;
- Mon binôme «Hichem » avec qui j'ai collaboré pour mener à bien ce modeste travail qui n'a pas été des plus simples ;
- Mon très cher ami « Aboubakr » que je considère plus qu'un frère ;
- Tous les étudiants du Master 2 R&T, en particulier mes très chers amisIshak, Ayoub, Abdou, Tina,Randa, Celia, Sofia… et toutes les personnes qui me connaissent.
- Toutes celles et ceux qui ont contribué de près ou de loin à l'accomplissement de ce modeste travail.

DJENOURI Mohamed Amine

ملخص:

يتمثل هذا العمل في استعمال اشارة فوضوية لنظام Sprott لتشفير معلومة، تمّ دراسة نظام الإرسال مع ادراج الرسالة، ثمّ تمكّنا من مزامنة جهازي نظام الإرسال و الاستقبال و ذلك بفضل طريقة تزامن حلقة مغلقة، بعد ذلك تمّ دراسة نظام الإستقبال مع اخذ بعض البيانات يُظهر الجُزء التجريبي من عملنا.

تمّت محاكاة نظام Sprott باستخدام برنامج MATLAB (simulink)، استعملنا بطاقة FPGA كأداة مبرمجة لمشاهدة البيانات في الوقت الحقيقي.

كلمات المفاتيح: دليل Lyapunov، تشعيب ، بطاقة FPGA نظام Sprott اشارة فوضوية، المزامنة بحلقة مغلقة.

Résumé :

Ce travail consiste à utiliser un signal chaotique de Sprott pour crypter un signal informatif. Pour cela, dans un premier temps, on a étudié l'émetteur chaotique avec l'insertion du message. Ensuite on a synchronisé les deux systèmes émetteur-récepteur grâce à la méthode de synchronisation par boucle fermée.

Nous avons achevé notre travail par une simulation du système de Sprott à l'aide du logiciel MATLAB (Simulink), suivie d'une réalisation pratique qui a consisté en une implémentation de notre système de transmission de Sprott sur carte FPGA. La visualisation des signaux simulés et expérimentaux nous a permis de constater une convergence entre les deux types de signaux.

Mots clés: Signal chaotique ; bifurcation ; exposant de Lyapunov ; synchronisation, boucle fermée ; système de Sprott ; carte FPGA.

Abstract :

This work is to use a chaotic signal Sprott to encrypt an informative signal. For this, at first, the transmitter has been studied with the insertion of the message. Then the two systems transmitter-receiver are synchronized with synchronization method closed loop. We completed our work by Sprott simulation system using MATLAB (Simulink) software followed by a practical implementation, which consisted of an implementation of our transmission system Sprott on FPGA card. The visualization of simulated and experimental signals allowed to see a convergence between the two types of signals.

Keywords: Chaotic signal; bifurcation; Lyapunov exponent; synchronization, closed loop; Sprott system; MATLAB(Simulink); FPGA card.

Nomenclature

CAN : Convertisseur Analogique Numérique.

CLB: Configurable Logic Block.

CNA: Convertisseur Numérique Analogique.

DES : Data Encryption Standard.

$D\mathcal{F}(x)$: Matrice du système.

DSP : Digital Signal Processor.

EEPROM: Electrically Erasable Programmable Read-Only Memory

EPROM: Erasable Programmable Read-Only Memory.

FPGA: Field Programmable Gate Array.

HDL: Hardware Description Langage.

IOB: Input output block.

ISE: Integrated Software Environment.

m(t) : Message informatif.

M_k : $k^{ième}$ point d'intersection de la trajectoire avec le plan de coupe.

Q_1 à Q7 : Paramètres du système de Sprott.

r : Paramètre de bifurcation.

r(t) : Message récupéré

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

$\mathbb{R}_n$: Espace vectoriel de dimension n construit dans le corps des réels.

s(t) : Signal crypté

SRAM: Static Random Access Memory.

VHDL: Very High Density Logic

VLSI : Very Large Scale Integration.

x(t) : Signal chaotique.

$\dot{x} = \frac{dx}{dt}$: Dérivée de la variable x par rapport au temps.

x_0 : L'état initial.

x_k : L'état x au temps t=k.

$\bar{x}$: Point fixe.

ξ : La différence entre l'état x et le point fixe.

θ : L'orbite périodique.

ω : L'angle de l'orbite.

λ_i : Valeurs propres de la matrice jacobienne ou exposant de Lyapunov.

Liste des Figures

Liste des tableaux

Table des matières

INTRODUCTION GENERALE

Depuis l'antiquité, l'homme n'a pas cessé de chercher les différents moyens pour transmettre un message à son correspondant et pouvoir ainsi communiquer avec lui en toute sécurité. Il a fourni à travers des époques successives, des efforts autant physiques qu'intellectuels pour pouvoir trouver une technique de communication efficace et appropriée.

En effet, les modes de télécommunications sont en évolution continue avec la recherche permanente de meilleurs débits, de facilité d'utilisation, de mobilité améliorée et surtout d'une confidentialité élevée.

Depuis des siècles, la cryptographie a été une histoire de conflit qui oppose deux camps, un qui cherche à cacher une information et l'autre qui essaie de trouver ce qu'on lui cache. Ainsi à chaque fois que le premier trouve un moyen de chiffrer ses messages le second essaie et avec le temps et les moyens dont il dispose, réussit à trouver la méthode ou l'astuce pour le décrypter. La cryptographie ancienne utilisait différents outils pour dissimuler une information ou un texte secret. Certains remplaçaient des mots par des nombres, d'autres mélangeaient, décalaient ou permutaient les lettres, comme dans la substitution alphabétique inverse, pour rendre la lecture du message difficile.

La cryptographie actuelle cherche à transformer de façon mathématique et algorithmique un message clair pour obtenir un autre chiffré et qui à première vue semble aléatoire. Plus l'inve rsion de la transformation est difficile, plus la sécurité est élevée et vice-versa. On cherche alors un phénomène d'apparence aléatoire mais qui est déterministe à l'origine pour le masquage d'information.

Il existe plusieurs systèmes présentant ce comportement, ils sont dits chaotiques, ils sont régis par des lois déterministes, dépendent d'un ou de plusieurs paramètres et leur évolution dans le temps est imprévisible. L'étude de tels systèmes est liée à la théorie du chaos qui a connu un grand essor à partir de 1960 grâce aux travaux de plusieurs chercheurs notamment ceux de Lorenz.

La cryptographie chaotique est ainsi née par inclusion du chaos dans les télécommunications et systèmes de transmission. L'idée consiste à noyer un message dans un signal chaotique pour faire face aux éventuelles tentatives de piratage.

La transmission chaotique est un mode de communication à clé secrète. La connaissance de cette clé est nécessaire du côté de l'émetteur du message ainsi que du récepteur pour le chiffrement et le déchiffrement du message. On doit alors disposer au niveau du récepteur, d'un signal chaotique identique à la porteuse pour pouvoir récupérer le message masqué.

Ce travail de mémoire consiste à réaliser un système de transmission sécurisée à base du chaos. Il repose d'une part sur la synchronisation chaotique et d'autre part sur le masquage de l'information secrète. Ce système se compose de deux oscillateurs chaotiques liés par un canal de transmission publique. Un message sera crypté puis envoyé à partir de l'oscillateur émetteur. L'objectif est de récupérer ce signal utile en utilisant une synchronisation chaotique de l'oscillateur émetteur.

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

- Le premier chapitre présente un rappel sur les systèmes dynamiques en général et chaotiques en particulier. Il énoncera quelques concepts sur la théorie du chaos.
- Le second chapitre consiste à étudier l'émetteur qui est composé de l'oscillateur chaotique de Sprott et du circuit d'insertion du message à crypter.
- Dans le troisième chapitre, on mettra en évidence d'une part, les différents types de la synchronisation et d'autre part la méthode qu'on a choisie pour récupérer notre signal crypté émis.
- Le quatrième chapitre présente un aperçu sur la technologie FPGA et avec laquelle on implémentera notre système de transmission chaotique en utilisant l'outil System Generator sous MATLAB (Simulink), qui grâce à la bibliothèque XILINX, servira à générer le programme en langage VHDL.

Chapitre 1: GENERALITES SUR LES SYSTEMES DYNAMIQUES CHAOTIQUES

1.1. Introduction :

Il y a quatre siècles, Newton et d'autres scientifiques ont introduit l'idée du déterminisme dans la représentation mathématique du monde réel. Nous nous intéresserons dans ce chapitre aux systèmes dynamiques chaotiques en nous attardant sur les espaces de phase, les attracteurs étranges et les scénarios de transition vers le chaos (bifurcations), lesquels nous permettront de mieux comprendre la nature du chaos. Notre étude se focalise sur l'application des signaux chaotiques dans les systèmes de transmission, et on montrera que certaines propriétés des systèmes chaotiques sont très intéressantes pour crypter les données par exemple. Donc le but de ce chapitre est de présenter quelques rappels indispensables et nécessaires à la compréhension de ce mémoire.

1.2. Les systèmes dynamiques :

1.2.1. Définition d'un système dynamique :

C'est un système d'équation différentielle de la forme :

$$\frac{dx}{dt} \equiv \dot{x} \tag{1.1}$$

Un système dynamique décrit par une fonction mathématique présente deux types de variables dynamiques et statiques : les variables dynamiques sont les quantités fondamentales qui changent avec le temps ; les variables statiques encore appelées paramètres du système sont fixes [1].

1.2.2. Système dynamique à temps discret :

Un système discret est représenté par l'équation d'état suivante :

$$x_{k+1} = f(x_k, v) \tag{1.2}$$

Où $f : \mathbb{R}^n \longmapsto \mathbb{R}^n$ est une fonction au moins continue ou continue par morceaux qui définit la dynamique du système discret. De la même manière si nous associons à cette dynamique un état initial $x_0$nous pourrons avoir une solution unique de$f$ où v est le vecteur de paramètre [1].

1.2.3. Système dynamique à temps continu :

Un système à temps continu est décrit par un système d'équations différentielles :

$$\frac{dx}{dt} \equiv \dot{x} = f(x_t, t, v) \tag{1.3}$$

Où f est de classe $C^1 : \mathbb{R}^n \longmapsto \mathbb{R}^n$ définit la dynamique du système continu. Nous pouvons associer une solution unique du système définie à l'aide de l'équation (1.2). L'évolution des ensembles d'états successifs du système à chaque instant t, représente la trajectoire [1].

1.2.4. Systèmes autonomes et non autonomes :

Soit le système dynamique suivant :

$$\dot{x} = \frac{dx}{dt} = f(x, t) \tag{1.4}$$

Lorsque le champ de vecteurs f ne dépend pas explicitement du temps, on dit que le système dynamique est autonome. Dans le cas contraire il est non autonome.

Dans un système autonome, la trajectoire ne dépend pas du temps initial t, alors que dans un système non autonome elle dépend de t [2].

1.3. Le chaos :

1.3.1. Définition du chaos :

Le chaos tel que le scientifique le comprend ne signifie pas l'absence d'ordre; il se rattache plutôt à une notion d'imprévisibilité, d'impossibilité de prévoir une évolution à long terme du fait que l'état final dépend de manière si sensible de l'état initial. On appelle donc un système dynamique chaotique, un système qui dépend de plusieurs paramètres et caractérisé par une extrême sensibilité aux conditions initiales. Ils ne sont pas déterminés ou modélisés 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.

Les définitions et propriétés suivantes permettent de comprendre qualitativement les points marquants des systèmes chaotiques [3]. :

1.3.1.1. La non-linéarité :

Un système chaotique est un système dynamique non linéaire. Un système linéaire ne peut pas être chaotique.

1.3.1.2. Le déterminisme :

La notion de déterminisme signifie la capacité de « prédire » le futur d'un phénomène à partir d'un évènement passé ou présent. L'évolution irrégulière du comportement d'un système chaotique est due aux non linéarités Dans les phénomènes aléatoires, il est absolument impossible de prévoir la trajectoire d'une quelconque particule. À l'opposé, un système chaotique a des règles fondamentales déterministes et non probabilistes.

1.3.1.3. L'aspect aléatoire :

Tous les états d'un système chaotique présentent des aspects aléatoires.La figure 1.1 représente l'état chaotique x_1 du système de Rössler :

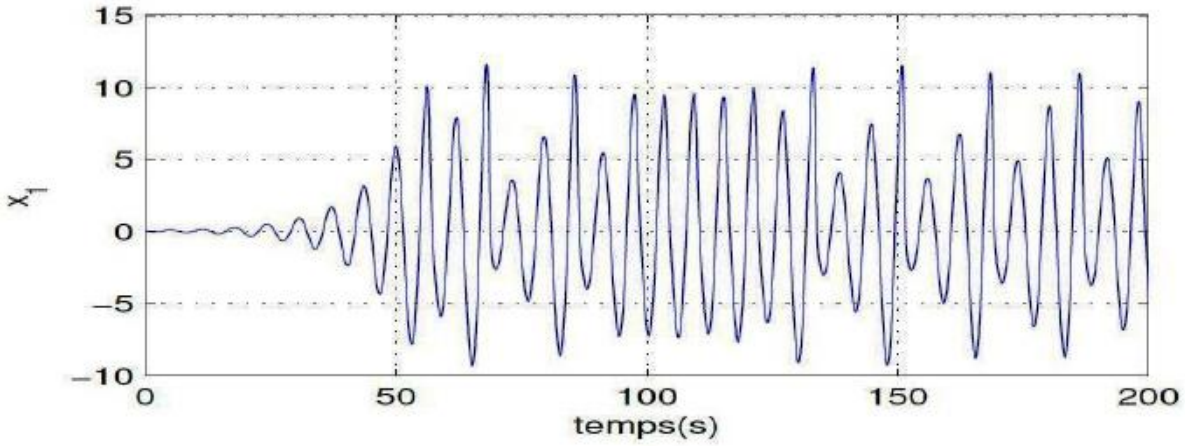


Figure 1.1 : Etat chaotique x_1 du système de Rössler [3].

1.3.1.4. Sensibilité aux conditions initiales :

Certains phénomènes dynamiques non linéaires sont si sensibles aux conditions initialesque, même s'ils sont régis par des lois rigoureuses et parfaitement déterministes, les prédictions exactes sont impossibles.

Il est clair que la moindre erreur ou imprécision sur la condition initiale interdit de décider à tout temps quelle sera la trajectoire effectivement suivie et de faire une prédiction sur l'évolution à long terme du système.

Une des propriétés essentielles du chaos est donc bien cette sensibilité aux conditions initiales que l'on peut caractériser en mesurant des taux de divergence des trajectoires. Ceci est illustré par la figure1.2.

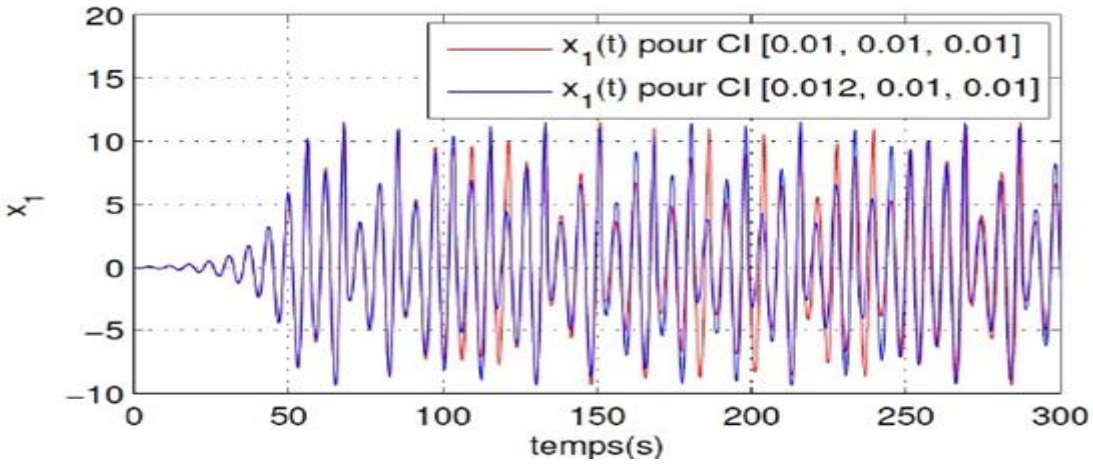


Figure 1.2 : Illustration de la propriété de sensibilité aux conditions initiales sur l'état x_1 [3]

1.3.2. L'histoire du chaos déterministe :

En 1963 le météorologue Edward Lorenz expérimentait une méthode lui permettant de prévoir les phénomènes météorologiques. 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. Les systèmes répondant à cette propriété seront à partir de 1975 dénommés : systèmes chaotiques. C'est donc au cours des années soixante-dix que la théorie du chaos a pris son essor.

Cependant, les travaux de certains scientifiques menés bien avant cette découverte vont être très utiles à la compréhension de la dynamique chaotique. En effet, vers la fin du XIXe siècle le mathématicien, physicien et philosophe français Henri Poincaré avait déjà mis en évidence le phénomène de sensibilité aux conditions initiales lors de l'étude astronomique du problème des trois corps. On trouve dans le Calcul des Probabilités d'Henri Poincaré l'affirmation suivante : «Une cause très petite, qui nous échappe, détermine un effet considérable que nous ne pouvons pas ne pas voir, et alors nous disons que cet effet est dû au hasard. Si nous connaissions exactement les lois de la nature et la situation de l'univers à l'instant initial, nous pourrions prédire exactement la situation de ce même univers à un instant ultérieur. Mais, alors même que les lois naturelles n'auraient plus de secret pour nous, nous ne pourrions connaître la situation qu'approximativement. Si cela nous permet de prévoir la situation ultérieure avec la

même approximation, c'est tout ce qu'il nous faut, nous disons que le phénomène a été prévu, qu'il est régi par des lois; mais il n'en est pas toujours ainsi, il peut arriver que de petites différences dans les conditions initiales en engendrent de très grandes dans les phénomènes finaux ; une petite erreur sur les premières produirait une erreur énorme sur les derniers. La prédiction devient alors impossible et nous avons le phénomène fortuit.

Cette citation définit parfaitement le chaos en tant que sensibilité aux conditions initiales mais aussi le déterminismequi réside dans le fait que si une condition initiale est parfaitement déterminée alors l'évolutiondu système l'est aussi. Le déterminisme traduit l'unicité de la solution pour l'équation différentielle d'unsystème donné, c'est le théorème de Cauchy.

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

Les travaux de Lyapunov, d'abord tombés dans l'oubli, seront plus tard très précieux pour étudier certains aspects de la théorie du chaos.

La figure 1.3 illustre un exemple d'une suite à comportement chaotique où $un{+}1 = 3{,}82\ un(1{-}un)$, pour deux conditions initiales différentes de 10^{-2}, l'erreur augmente de façon exponentielle durant les huit premières itérations.

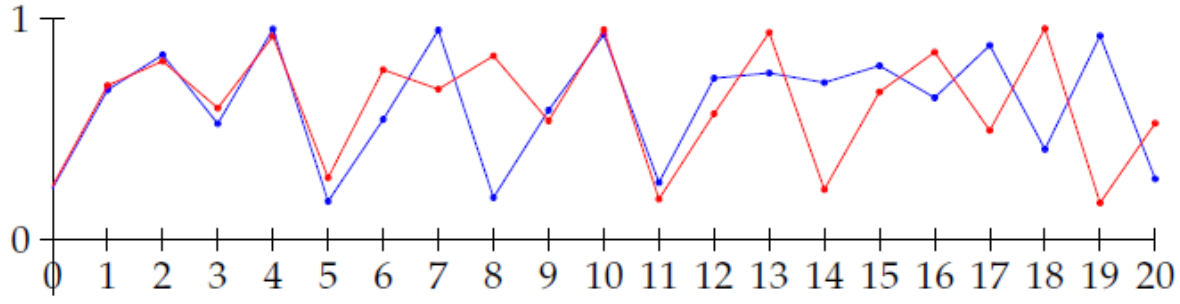


Figure 1.3 : Exemple de suite à comportement chaotique [4].

Les travaux des prédécesseurs de Lorenz ont donc été très importants pour la compréhension du chaos déterministe, mais il faut souligner que ce qui va permettre aux scientifiques une compréhension plus accrue des systèmes chaotiques, c'est l'ordinateur. En effet, les équations différentielles régissant un système chaotique sont

nécessairement non linéaires et sans ordinateur, leur résolution est en général impossible [4].

1.3.3. Domaine d'application du chaos:

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

1.3.3.1. Ingénierie :

Contrôle de vibrations, stabilisation des circuits, réactions chimiques, turbines, étages de puissance, lasers, combustion, et beaucoup plus.

1.3.3.2. Ordinateurs :

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

1.3.3.3. Communications :

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

1.3.3.4. Médecine et biologie :

Cardiologie, analyse du rythme du cœur (EEG), prédiction et contrôle d'activité irrégulière du cœur.

1.3.3.5. Management et finance :

Prévisions économiques, analyse financière, et prévision du marché.

1.4. Plan de phase :

Le portrait de phase d'un système dynamique est une représentation graphique de plusieurs trajectoires représentatives dans l'espace des phases. Étant donné un système dynamique, $\dot{x} = f(x,t)$, sans résoudre les équations, on peut toujours, à un instant t donné, représenter graphiquement (à l'aide de flèches) le champ des $\dot{x}$ (x est la coordonnée du champ des vitesses). La lecture de cette représentation graphique sera très utile pour se faire une idée du comportement du système. Il s'agit d'un espace de dimensions 2 ou 3 dans lequel chaque coordonnée est une variable d'état du système considéré Il permet de distinguer un comportement chaotique d'un comportement purement aléatoire à l'aide des conditions initiales.

La figure 1.4 représente un exemple d'un plan de phase.

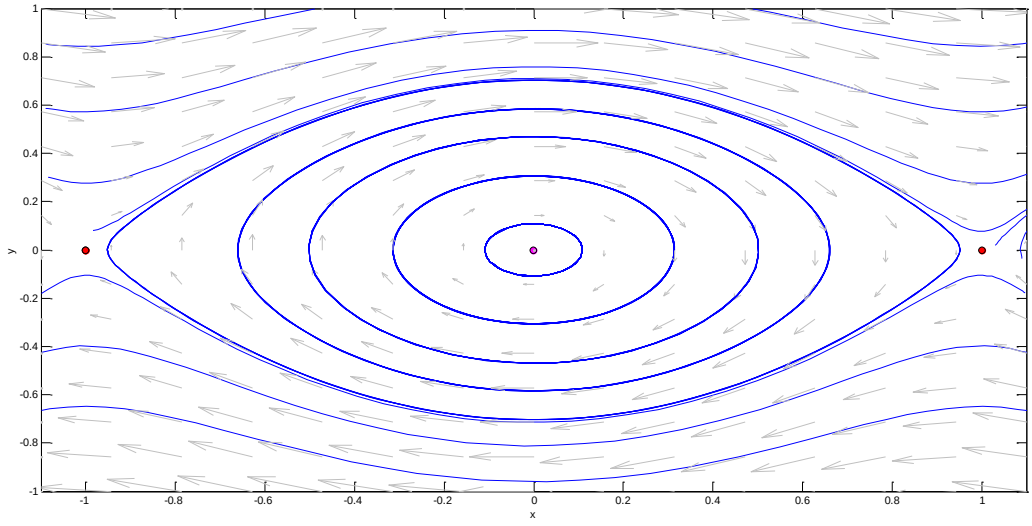


Figure 1.4 : Exemple d'un plan de phase.

1.5. Les points fixes :

On appelle points fixe appelé aussi, point stationnaire, point d'équilibre ou point critique, le point $\bar{x}$ de l'espace des phases obtenu en annulant le second membre de la fonction dynamique F :

$$F(\bar{x}) = 0 \tag{1.5}$$

Par le changement de variables $\xi = x - \bar{x}$, on peut ramener le point $\bar{x}$ à l'origine [2].

1.5.1. Stabilité des points fixes :

Un point fixe $\bar{x} \in R^n$ est stable si :

$$\forall \varepsilon > 0,\ \exists \delta > 0\ telque\ \|x(0) - \bar{x}\| < \delta \ \rightarrow\ \|x(t) - \bar{x}\| < \varepsilon \tag{1.6}$$

où $\|\ \ \|$ désigne la norme dans R^n.

Si de plus, il existe δ_0 avec $0 < \delta_0 < \delta$ tel que :

$$\|x(0) - \bar{x}\| < \delta_0 \ \rightarrow\ \lim_{t \to \infty} x(t) = \bar{x} \tag{1.7}$$

$\bar{x}$ est asymptotiquement stable.

1.5.2. Stabilité du système linéarisé (valeurs propres) :

La matrice :

$$D\mathcal{F}(x) \equiv \frac{\partial \mathcal{F}_i(x)}{\partial x_j} \tag{1.8}$$

est la matrice jacobéenne de $\mathcal{F}$(x). Son déterminant est le jacobien.

Pour x petit, le comportement du système au voisinage de 0 est celui du système linéarisé :

$$\dot{x} = D\mathcal{F}(0)x \tag{1.9}$$

Dans le cas où la matrice :$D\mathcal{F}(0)$ possède n valeurs propres λ_i, i=1,..., n distinctes, la solution est :

$$x = \sum_{i=1}^{n} c_i a^{(i)} exp\lambda_i t \tag{1.10}$$

où a^i est le vecteur propre correspondant à la valeur propre λ_i et les c_i, i =1,2,..., n sont des constantes (déterminées par les condition initiales).On en déduit que :

a) Si toutes les valeurs propres λ_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 une des valeurs propres à sa partite réelle positive le point fixe est instable.
d) Si $D\mathcal{F}(0)$ n'a pas de valeur nulle ou purement imaginaire, le point fixe est un point hyperbolique. Dans le cas contraire, il est non-hyperbolique.
e) S'il existe $ietj$ tels que $\Re\lambda_i$<0 et $\Re\lambda_i$>0, le point fixe et point seul.
f) Si toutes les valeurs propres de $D\mathcal{F}(0)$ sont réelles et de même signe, le point fixe est un nœud.

1.6. Bifurcation :

Dans un système dynamique, une bifurcation est un doublement de période quadruplement, etc., qui accompagne le début du chaos. Il représente l'apparition soudaine d'une solution qualitativement différente pour un système non linéaire suite à une modification d'un paramètre. Il existe plusieurs types de bifurcation [5] :

1.6.1. Bifurcation nœud–col :

C'est la bifurcation associée à l'équation du premier ordre

$$\dot{x} = r + x^2 \tag{1.11}$$

avec r le paramètre de contrôle.

Quand r<0, on aura deux points fixes, un point stable et l'autre point instable. A chaque fois que r tend vers 0, la parabole se déplace vers le haut et les deux points fixes se déplacent l'un vers l'autre jusqu'à ce qui ils se rejoignent en un point fixe demi stable. Ce dernier disparaît dès que r> 0.

La figure 1.5 illustre la bifurcation nœud-col.

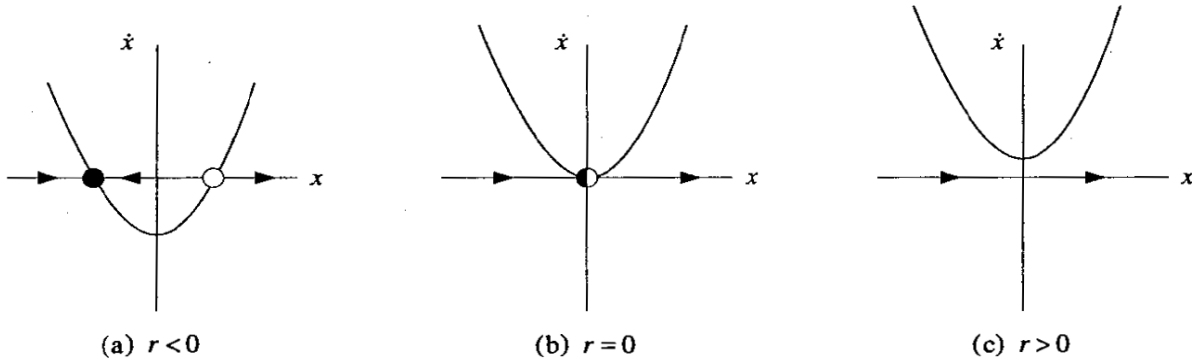


Figure 1.5 : Bifurcation nœud –col.

1.6.2. Bifurcation Trans-critique :

C'est la bifurcation associée à l'équation du premier ordre :

$$\dot{x} = rx - x^2 \tag{1.12}$$

Il y a deux points fixes $\bar{x}$=0 et $\bar{x}$=r.

- Pour r <0 , le point fixe $\bar{x}$=0 est donc stable tandis que le point fixe $\bar{x}$=r est instable.
- Pour r=0 , il y a qu'un seul point fixe demi stable$\bar{x}$=0 (même raisonnement que celui de la bifurcation noud-col). Il y a donc échange de stabilité en r=0.
- Pour r>0 , le point fixe $\bar{x}$=0 est donc instable tandis que le point fixe $\bar{x}$=r est stable

La figure 1.6 illustre la bifurcation trans-critique.

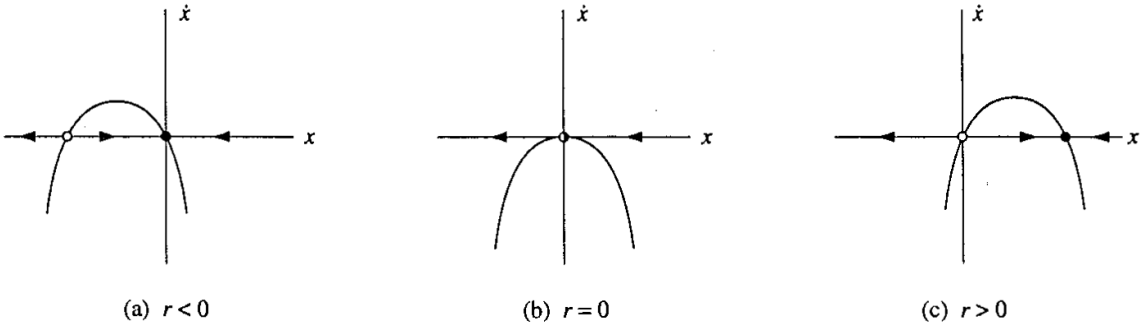


Figure 1.6 : Bifurcation Trans-critique.

1.6.3. Bifurcation fourche :

C'est la bifurcation associée à l'équation du premier ordre :

$$\dot{x} = rx - x^3 \tag{1.13}$$

Il y a trois points fixe : $\bar{x} = \pm\sqrt{r}$ et $\bar{x}$=0.

- Pour r<0, il y a un seul point fixe stable $\bar{x}$=0.
- Pour r=0, il y a qu'un seul point fixe demi stable$\bar{x}$=0 (même raisonnement que celui de la bifurcation noeud-col), il y a donc ralentissement critique en r=0.
- Pour r>0, il y a trois points fixes. Les deux points fixes $\bar{x} = \pm\sqrt{r}$ sont stables tandis que le point fixe $\bar{x} = 0$ est instable.

La figure 1.7 illustre la bifurcation fourche

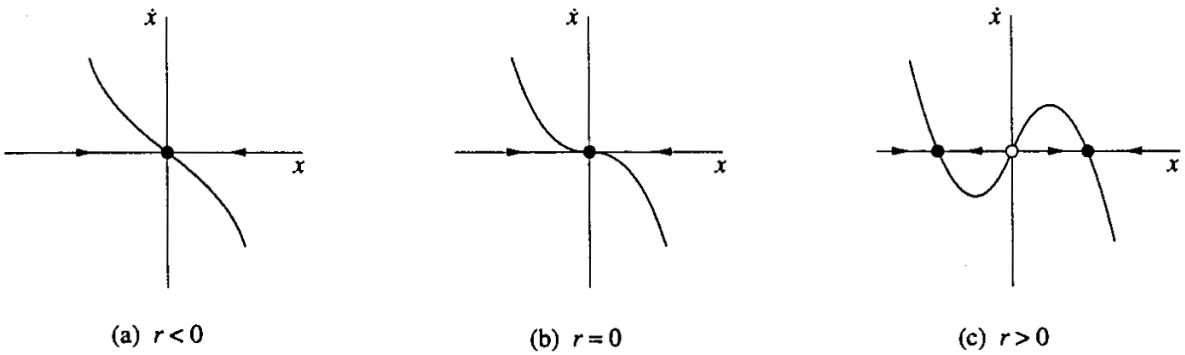


Figure 1.7 : Bifurcation fourche.

1.6.4. Bifurcation de Hopf :

C'est la bifurcation associée à l'équation dans le plan complexe :

$$z'(t) = \mathcal{F}\big(z(t)\big) = (\mu + i\omega)z(t) - |z|^2 z(t) \tag{1.14}$$

Pour étudier cette équation, on écrit la variable z sous la forme $z(t) = x(t)e^{i\theta(t)}$.

L'équation s'exprime sous forme d'un système :

$$x' = rx - x^3 \tag{1.15}$$

$$\theta' = \omega \tag{1.16}$$

La première équation n'est autre qu'une bifurcation fourche de paramètre de contrôle μ.
La figure 1.8 représente le diagramme de bifurcation de Hopf

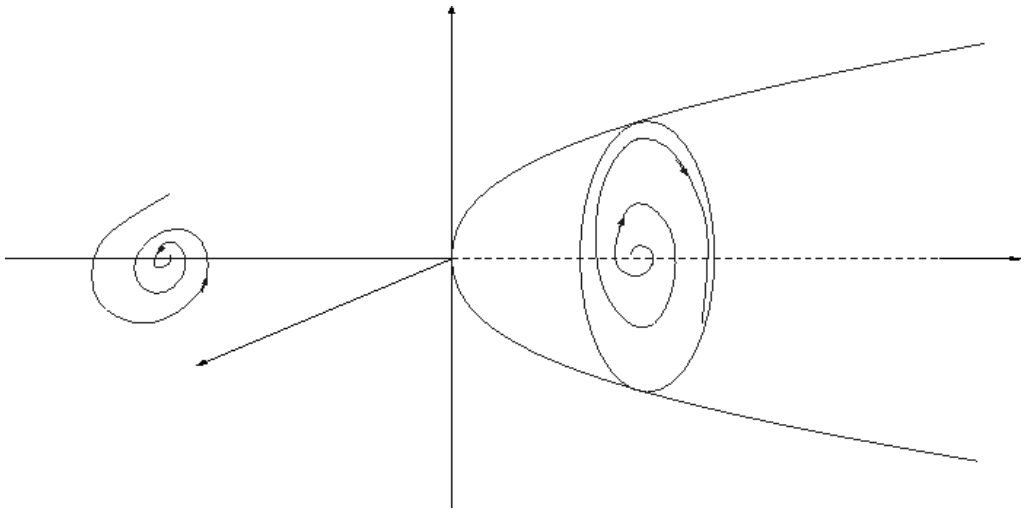


Figure1.8 : Diagramme de bifurcation de Hopf.

Nous partons d'un système où le paramètre $r$est négatif. Le système possède un point d'équilibre stable qui correspond ici à un point puits : les trajectoires s'enroulent en spirale vers l'origine. Lorsque r=0, ce point d'équilibre perd sa stabilité, puis lorsque r>0, il se forme alors une trajectoire périodique stable ou cycle limite.

La bifurcation de Hopf corresponde à une instabilité oscillatoire.

1.7. Section de Poincaré :

Henri Poincaré a apporté une contribution très utile pour l'étude des systèmes chaotiques. Parmi ses contributions on trouve les sections de Poincaré. Faire une section de Poincaré revient à couper la trajectoire dans l'espace des phases. Afin d'étudier les intersections de cette trajectoire, on passe alors d'un système dynamique à temps continu à un système dynamique à temps discret. On remplace l'étude du système continue $\dot{x}$=f(x) dans $\mathcal{R}^n$par [6] :

- l'application ponctuelle T dans $\mathcal{R}^2$ définie ainsi$M_{\mathcal{K}+1}$=T ($M_{\mathcal{K}}$) où $M_{\mathcal{K}}$ est le k^{ième} point d'intersction de la trajectoire avec le plan de coupe.
- l'application g dite de premier retour définie ainsi $x_{\mathcal{K}+1} = g(x_{\mathcal{K}})$ où $x_{\mathcal{K}}$ est l'abscisse du k-nième point d'intersection de la trajectoire avec les plans de coupe.

Les cas typique observés :

- la solution périodique dans $\mathcal{R}^n$(c'est un cycle limite) : la section de Poincaré est un point.
- la solution est quasi-périodique à deux fréquences f_1 et f_2. On distingue deux cas selon que le rapport $r=f_1/f_2$ soit rationnel ou pas :
  - si r n'est pas rationnel : la section de Poincaré est une courbe fermée.
  - si r est rationnel : la section de Poincaré se compose de quelque points.
- la solution est apériodique : la section de Poincaré est un nuage de points.

La figure 1.9 montre un exemple d'une section de Poincaré du système de Lorenz.

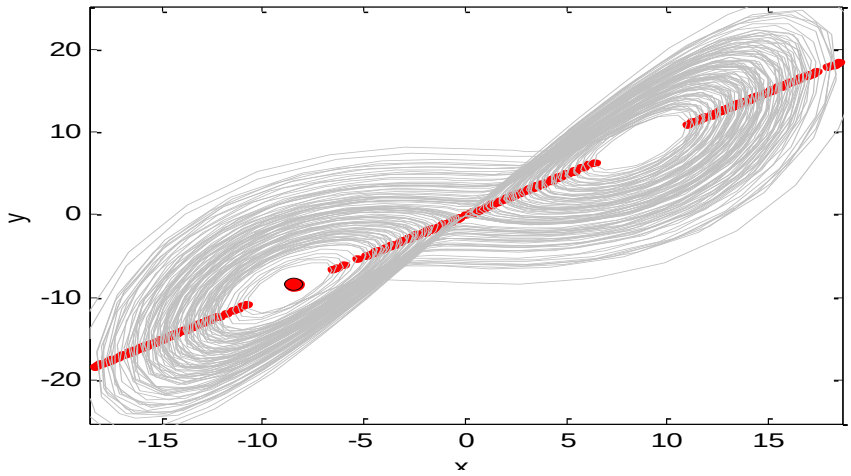


Figure 1.9 : Section de Poincaré du système de Lorenz.

1.8. Les exposants de Lyapunov :

Certains systèmes dynamiques sont très sensibles aux petites variations de leurs conditions initiales. Ces variations peuvent rapidement prendre d'énormes proportions. Le mathématicien russe Alexander Lyapunov s'est penché sur ce phénomène et a développé une quantité permettant de mesurer la vitesse à laquelle ces petites

variations peuvent s'amplifier. Cette quantité appelée "exposant de Lyapunov" mesure en fait le degré de sensibilité d'un système dynamique [1].

1.8.1. Les exposants de Lyapunov pour des attracteurs non chaotiques :

Pour un attracteur non chaotique, les exposants de Lyaponov sont tous négatifs ou nuls : $\lambda_{\mathcal{K}} \leq 0 \forall t, \mathcal{K}$ et leur somme est négative : $\sum_{\mathcal{K}=1}^{n} \lambda_{\mathcal{K}}$<0. Les attracteurs non chaotiques sont classés en quatre catégories :

- Point d'équilibre asymptotiquement stable : λ_{κ}< 0 pour κ=1,...,n.
- Cycle limite stable : $\lambda_1 1$ = 0 et λ_{κ}< 0 pour κ= 2,...,n.
- Tore d'ordre 2 asymptotiquement stable : λ_1 = 0, $\lambda_2 2$ = 0 et λ_{κ}< 0 pour κ=3,..., n.
- Tore d'ordre K asymptotiquement stable : $\lambda_1 1$ = ...λ_{κ}= 0, $\lambda_2 2$ = 0 et λ_{κ}k < 0 pour κ= k + 1,..., n.

1.8.2. Les exposants de Lyapunov pour un attracteur étrange (systèmes chaotiques) :

Une des particularités du chaos est son extrême sensibilité aux conditions initiales. Un attracteur étrange possèdera toujours au moins un exposant de Lyapunov positif avec la propriété $\sum_{\kappa=1}^{n} \lambda_{\kappa} < 0$. De plus, pour un attracteur étrange, un des exposants de Lypunov est toujours nul. Cela signifie que pour respecter la condition $\sum_{\kappa=1}^{n} \lambda_{\kappa} < 0$, un attracteur étrange doit avoir au minimum trois exposants de Lyapunov. Donc, un système continu dans le temps doit être au moins de dimension trois pour produire du chaos.

La figure 1.10 représente les exposants de Lyapunov de l'attracteur étrange de Lorenz :

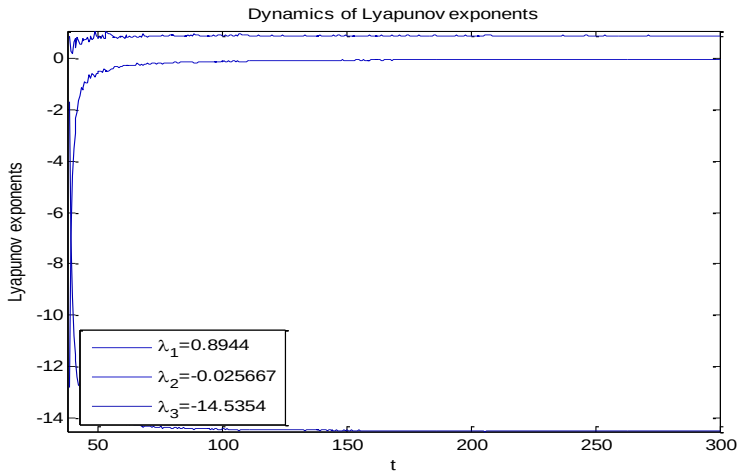


Figure 1.10 : La dynamique des exposants de Lyapunov.

1.9. Les attracteurs étranges :

Une des découvertes les plus spectaculaires des dernières années a été celle des attracteurs étranges. Ces objets géométriques issus de l'évolution de systèmes chaotiques. Dans le plan, ils sont formés d'une suite infinie de points x_0, x_1, x_2, ...$x_n$qui dépendent de la valeur initiale x_0. Au fur et à mesure que le nombre de points augmente, une image se forme dans le plan et devient de plus en plus nette. Cette image n'est pas une courbe ni une surface, c'est en fait un objet intermédiaire constitué de points avec entre eux des espaces inoccupés. L'objet est qualifié d'étrange en raison de sa structure pointilliste et de sa nature fractale. Une valeur différente de x_0 conduit à une toute autre suite qui après une courte phase, dessine la même image. D'où qu'on parte, on se retrouve toujours sur l'attracteur, c'est le côté prévisible de l'évolution. Où se retrouve-t-on exactement sur l'attracteur ? Il est impossible de répondre à la question, c'est le côté imprévisible de l'évolution. À la suite de la découverte d'Edward Lorenz en 1963 de son fameux attracteur à l'allure d'un papillon, plusieurs recherches principalement en physique ont permis d'améliorer nos connaissances sur les attracteurs étranges [7].

1.9.1. Attracteur de Lorenz :

L'attracteur de Lorenz tient son nom du météorologue Edward Lorenz qui l'a étudié le premier. C'est une simplification à l'extrême d'équations régissant les mouvements atmosphériques. Lorenz les a étudiés afin de mettre en évidence sur un système simple la sensibilité aux conditions initiales qu'il avait observée.

Les équations de ce système sont les suivantes :

$$\begin{cases} \frac{dx}{dt} = a(y - x) \\ \frac{dy}{dt} = bx - y - xz \\ \frac{dz}{dt} = xy - cz \end{cases} \tag{1.17}$$

On prendra : *a= 10*, *b = 28* et *c = 8/3.*

La figure 1.11 représente l'attracteur étrange de Lorenz :

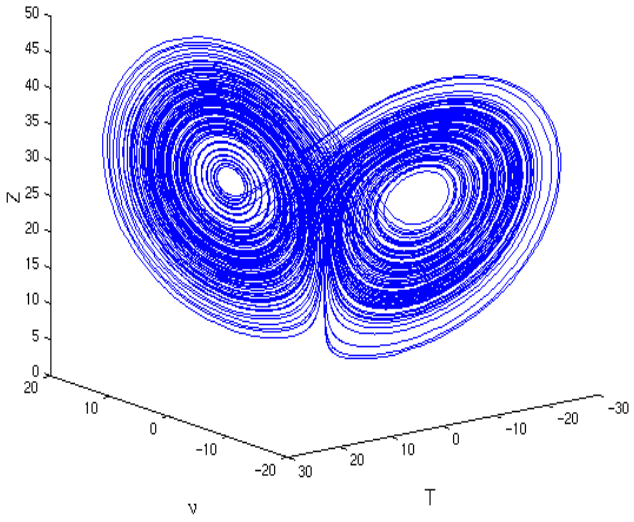


Figure 1.11 : Attracteur étrange de Lorenz.

1.9.2. Attracteur de Rössler:

Proposé par l'Allemand Otto Rössleren1974, l'attracteur de Rössler ne provient pas de l'étude d'un système physique, du moins pas directement. Il résulte d'un effort de simplification pour étudier plus facilement la "chute" d'une trajectoire dans un bassin d'attraction.

Les équations de ce système sont les suivantes :

$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + xz - cz \end{cases} \quad (1.18)$$

On prendra : $a = 0.398$, $b = 2$ et $c = 4$.

La figure 1.12 représente l'attracteur étrange de Rössler :

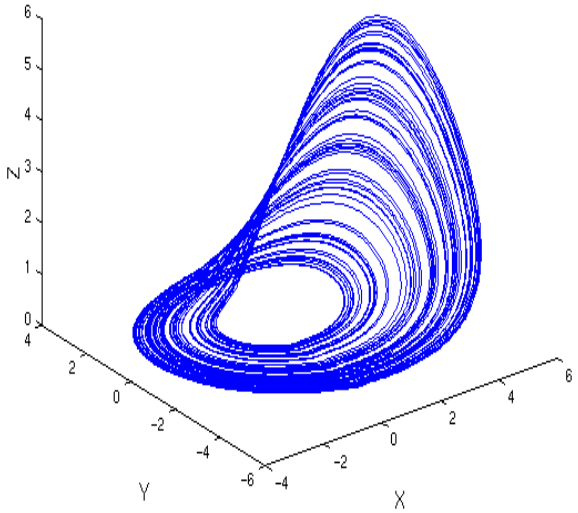


Figure 1.12 : Attracteur étrange de Rössle.

1.9.3. Attracteur de Chua :

Proposé par le professeur chinois LeonOngChua en1993, cet attracteur provient de l'étude de l'oscillateur chaotique de Chua .

Les équations de ce système sont les suivantes :

$$\begin{cases} \frac{dx}{dt} = \alpha * (y - x - c * x) \\ \frac{dy}{dt} = x - y + z \\ \frac{dz}{dt} = -\beta * y \end{cases} \quad (1.19)$$

On prendra :$\alpha = 10, c = -0.143,\ \beta = 16.$

La figure 1.13 représente l'attracteur étrange de Chua :

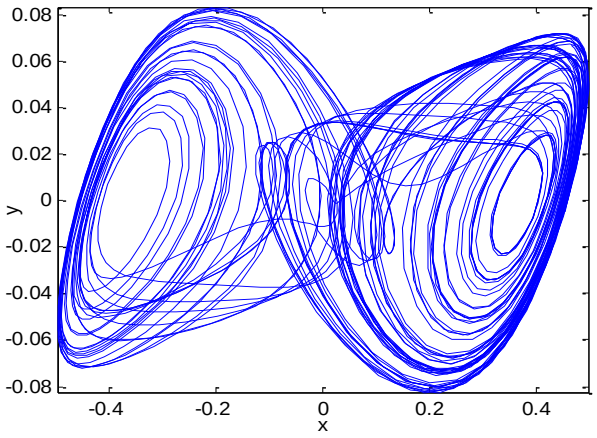


Figure 1.13 : Attracteur étrange de Chua.

1.10. Conclusion :

Dans ce chapitre nous avons donné un aperçu sur les systèmes dynamiques et la nécessité de présence de système non linéaire dans le régime chaotique. Nous avons aussi introduit les notions fondamentales qui caractérisent les systèmes chaotiques : la sensibilité aux conditions initiales, le plan de phase, stabilité des points fixe, etc…

Ensuite, nous avons évoqué le comportement des systèmes chaotiques grâce à la bifurcation en donnant quelques exemples types de bifurcation. Puis nous avons montré comment peut-on faire la différence entre un système chaotique par rapport aux autres systèmes grâce à la section de Poincaré et comment peut-on calculer le degré de sensibilité des systèmes grâce aux exposants de Lyaponov. Enfin, nous avons donné quelques exemples de systèmes chaotiques et leurs attracteurs étranges correspondants décrits par Lorenz, Rössler et Chua.

Chapitre 2 : ETUDE DE L'EMETTEUR CHAOTIQUE

2.1. Introduction :

La cryptographie actuelle, en particulier RSA (aux noms des inventeurs : Ronald Rivest, Adi Shamir et Leonard Adleman) et DES (Data Encryption Standard) souffre de la montée de la puissance de calculs des ordinateurs, d'autant plus que l'arrivée de l'ordinateur quantique pourrait bien sonner le glas de ces algorithmes. Pour remédier, et surtout trouver un remplaçant, deux pistes intéressent les chercheurs actuellement:

Le principe de la cryptographie chaotique est de noyer le message à transmettre dans un signal chaotique et de l'envoyer à un récepteur qui connait les caractéristiques du générateur de chaos, et qui pourra donc soustraire le chaos au signal reçu par synchronisation et ainsi en extraire le message. Dans ce chapitre nous allons aborder dans un premier temps l'étude de l'oscillateur chaotique de Sprott qui constitue l'élément essentiel de l'émetteur. L'étude du montage va nous permettre de tirer les équations du mouvement. Ensuite on va étudier les différentes caractéristiques de notre oscillateur de Sprott (diagramme de bifurcation, point fixe, sections de Poincaré, etc…). Des différentes méthodes d'insertion des messages proposées par la littérature [10] on choisira le type de message crypté le mieux adapté pour le fonctionnement de notre émetteur chaotique. Enfin à l'aide du logiciel Matlab (Simulink), les signaux suivants : le message à transmettre m(t), le signal chaotique de notre oscillateur x(t) et le signal crypté s(t) serons visualisés et commentés.

2.2. Étude de l'oscillateur chaotique de Sprott :

2.2.1. Etude du montage :

L'oscillateur chaotique de Sprott est représenté sur la figure 2.1.

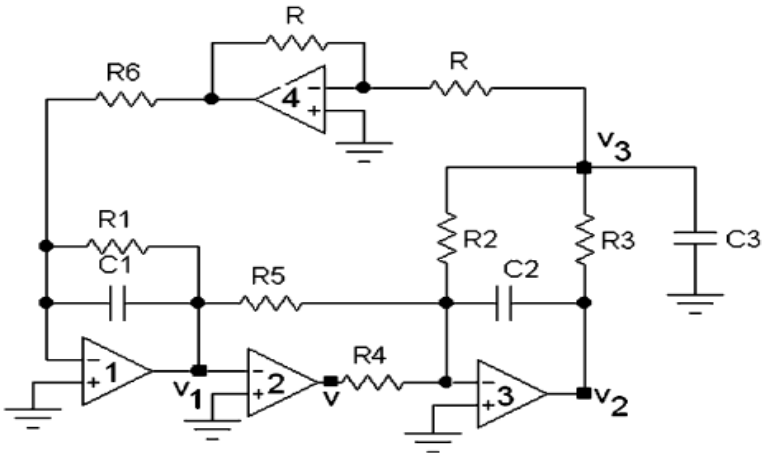


Figure 2.1 : L'oscillateur de Sprott [8].

2.2.2. Équations de l'oscillateur :

La figure 2.1 représentant notre oscillateur de Sprott comporte des résistances, condensateurs et des amplificateurs opérationnels. Seul le deuxième AOP agit comme un amplificateur non linéaire et sa tension de sortie v est donnée par l'équation suivante :

$$v = -\mathcal{V}_{cc} sign(v_1) \tag{2.1}$$

où $\mathcal{V}_{cc}$ est la tension de polarisation utilisée.

L'oscillateur est un système dynamique du troisième ordre. La dynamique de l'oscillateur est donnée par les trois équations différentielles suivantes :

$$\frac{dv_1}{dt'} = -\frac{1}{R_1 C_1} v_1 + \frac{1}{R_6 C_1} v_3 \tag{2.2}$$

$$\frac{dv_2}{dt'} = -\frac{1}{R_5 C_2} v_1 - \frac{1}{R_2 C_2} v_3 + \frac{1}{R_4 C_2} \mathcal{V}_{cc} sign(v_1) \tag{2.3}$$

$$\frac{dv_3}{dt'} = \frac{1}{R_3 C_3} v_2 - \left(\frac{1}{RC_3} + \frac{1}{R_2 C_3} + \frac{1}{R_3 C_3}\right) v_3 \tag{2.4}$$

La fonction sign est définie par:

$$sign\ (x)\begin{cases} = -1 & six < 0 \\ = 0 & six = 0 \\ = 1 & six > 0 \end{cases} \tag{2.5}$$

On procède ensuite les changements de variable suivants :

$$x = \frac{v_1}{\mathcal{V}_{cc}}, y = \frac{v_2}{\mathcal{V}_{cc}}, z = \frac{v_3}{\mathcal{V}_{cc}}, t = \omega_0 t', Q_1 = \frac{1}{R_1 C_1 \omega_0}, Q_2 = \frac{1}{R_6 C_1 \omega_0}, Q_3 = \frac{1}{R_5 C_2 \omega_0},$$

$$Q_4 = \frac{1}{R_2 C_2 \omega_0}, Q_5 = \frac{1}{R_4 C_2 \omega_0}, Q_6 = \frac{1}{R_3 C_3 \omega_0},$$

$$Q_7 = \left(\frac{1}{R_3 C_3 \omega_0} + \frac{1}{R_2 C_3 \omega_0} + \frac{1}{R C_3 \omega_0}\right), \omega_0 = \frac{1}{\sqrt{R_2 R_3 C_2 C_3}}.$$

Le changement de variable sert à rendre les grandeurs adimensionnelles.

On obtient le système d'équations différentielles suivant:

$$\dot{x} = -Q_1 x + Q_2 z \tag{2.6}$$

$$\dot{y} = Q_3 x - Q_4 z + Q_5 sign(x) \tag{2.7}$$

$$\dot{z} = Q_6 y - Q_7 z \tag{2.8}$$

Pour l'analyse et la simulation de notre oscillateur, on a choisi les paramètres suivants [10] :

$$Q_1 = 0.7\ ;\ Q_2 = 2.5\ ;\ Q_3 = 0.85\ ;\ Q_4 = 1\ ,;\ Q_5 = 3\ ;\ Q_6 = 3\ ;\ Q_7 = 1$$

2.2.3. Caractéristiques de l'oscillateur :

2.2.3.1. Diagramme de bifurcation :

Pour le tracé du diagramme de bifurcation, un programme MATLAB a été écrit et les résultats obtenus sont représentés sur la figure 2.2. Le paramètre variable utilisé estQ_3.

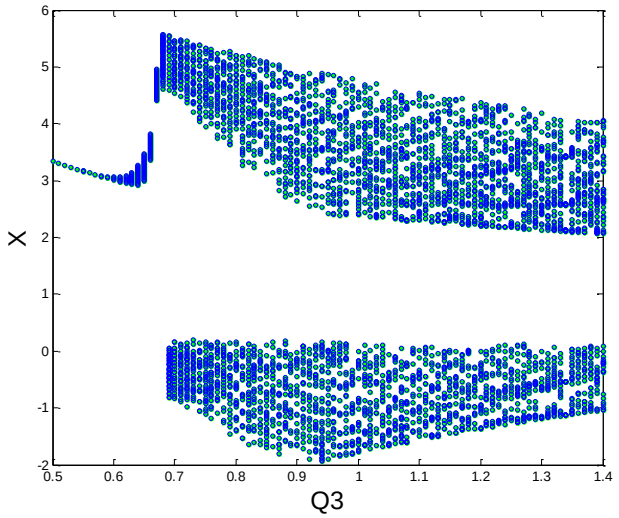


Figure 2.2 : Digramme de bifurcation de x en fonction de Q_3.

2.2.3.2. Points fixe :

On a notre système d'équation différentielle :

$$\dot{x} = -Q_1 x + Q_2 z \tag{2.9}$$

$$\dot{y} = -Q_3 x - Q_4 z + Q_5 sign(x) \tag{2.10}$$

$$\dot{z} = Q_6 y - Q_7 z \tag{2.11}$$

avec les valeurs suivantes :$Q_1 = 0.7$; $Q_2 = 2.5$; $Q_4 = 1$; $Q_5 = 3$; $Q_6 = 3$; $Q_7 = 1$

Et prenons $Q_3 = 0.85$ (cette valeur étant choisie à partir de la figure 2.2).

On aura trois cas pour étudier la stabilité des points fixes :

a) Cas $x < 0$:

On obtient le système d'équations différentielles suivant :

$$\dot{x} = -Q_1 x + Q_2 z \tag{2.12}$$

$$\dot{y} = (-Q_3 - Q_5)x - Q_4 z \tag{2.13}$$

$$\dot{z} = Q_6 y - Q_7 z \tag{2.14}$$

Soit sous forme matricielle :

$$D(0\ 0\ 0) = \begin{bmatrix} -Q_1 & 0 & Q_2 \\ -Q_3 - Q_5 & 0 & -Q_4 \\ 0 & Q_6 & -Q_7 \end{bmatrix} \tag{2.15}$$

et la matrice Jacobienne :

$$\lambda I = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \tag{2.16}$$

avec : □ la valeur propre de la matrice et I la matrice identité

L'équation caractéristique est donné par : $det(D(0\ 0\ 0) - \lambda I) = 0$, soit :

$$-\lambda^3 - 0.3\lambda^2 - 2.3\lambda - 30.975 = 0 \tag{2.17}$$

La résolution de l'équation caractéristique nous donne ses racines, qui sont les valeurs propres de la matrice $(0\ 0\ 0)$. On n'obtient qu'une seul valeur propre $\lambda = -3$.
On en déduit que le point d'équilibre $\lambda = -3$ est un point fixe stable.

b) Cas $x = 0$:

On obtient le système d'équations différentielles suivant :

$$\dot{x} = -Q_1 x + Q_2 z \tag{2.18}$$

$$\dot{y} = -Q_3 x - Q_4 z \tag{2.19}$$

$$\dot{z} = Q_6 y - Q_7 z \tag{2.20}$$

Soit sous forme matricielle :

$$D(0\ 0\ 0) = \begin{bmatrix} -Q_1 & 0 & Q_2 \\ -Q_3 & 0 & -Q_4 \\ 0 & Q_6 & -Q_7 \end{bmatrix} \tag{2.21}$$

et la matrice Jacobienne :

$$\lambda I = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \tag{2.22}$$

L'équation caractéristique : $det(D(0\ 0\ 0) - \lambda I) = 0$ devient

$$-\lambda^3 - 1.7\lambda^2 - 3.7\lambda - 8.5 = 0 \tag{2.23}$$

On n'obtient qu'une seul valeur propre $\lambda = -4$. On en déduit que le point d'équilibre $\lambda = -4$ est un point fixe stable.

b) Cas $x > 0$:

On obtient le système d'équations différentielles suivant :

$$\dot{x} = -Q_1 x + Q_2 z \tag{2.24}$$

$$\dot{y} = (-Q_3 + Q_{5)} x - Q_4 z \tag{2.25}$$

$$\dot{z} = Q_6 y - Q_7 z \tag{2.26}$$

soit :

$$D(0\ 0\ 0) = \begin{bmatrix} -Q_1 & 0 & Q_2 \\ -Q_3 + Q_5 & 0 & -Q_4 \\ 0 & Q_6 & -Q_7 \end{bmatrix} \tag{2.27}$$

L'équation caractéristique $(D(0\ 0\ 0) - \lambda I) = 0$ devient

$$-\lambda^3 - 1.7\lambda^2 - 7.575\lambda - 7.475 = 0 \tag{2.28}$$

On n'obtient qu'une seule valeur propre $\lambda = -1.054$.

On en déduit que le point d'équilibre $\lambda = -1.054$ est un point fixe stable.

2.2.3.3. Section de Poincaré :

Dans le premier chapitre nous avons définit la section de Poincaré qui permet de différencier un système chaotique d'un système périodique. Pour notre oscillateur de Sprott, la section de Poincaré avec le plan y=0 ainsi obtenue, est mentionnée par des points rouges.

La figure 2.3 représente la section de Poincaré sur l'attracteur étrange de Sprott.

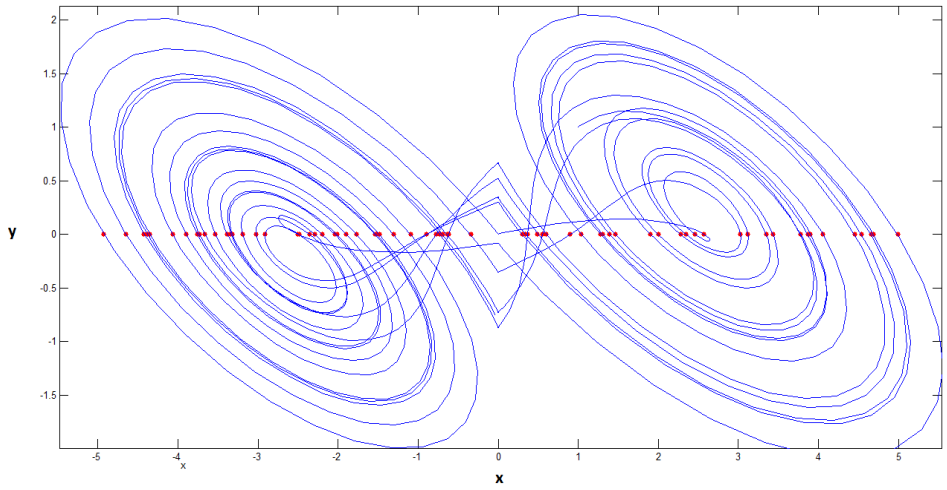


Figure 2.3 : Section de Poincaré de l'attracteur étrange de Sprott.

Nous avons utilisé l'outil MATDS travaillant sous MATLAB, permettant l'étude des systèmes dynamiques.

2.2.3.4. Exposants de Lyapunov :

On sait que pour un attracteur étrange (chaotique), il faut que la somme des exposants de Lyapunov soit négative et qu'au moins un de ses exposant soit positif. Pour notre

oscillateur chaotique de Sprott, les exposants de Lyapunov sont représentés sur la figure 2.4 et ont été obtenu à l'aide de l'outil MATDS.

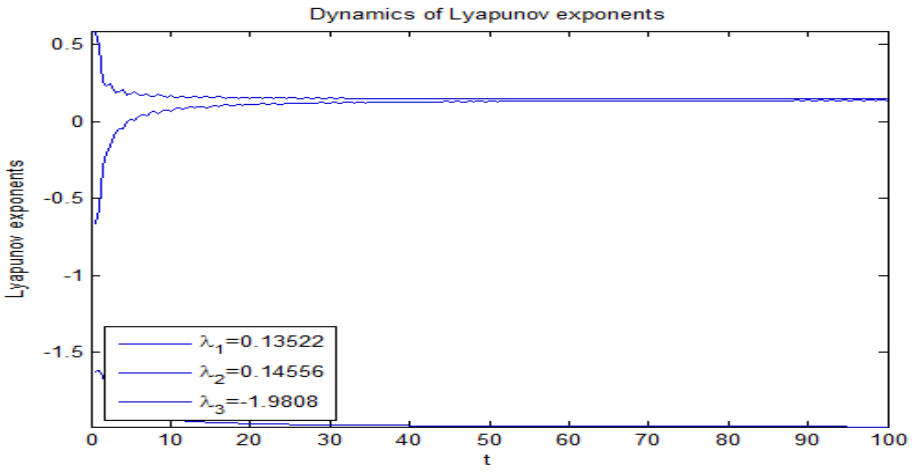


Figure 2.4 : Les exposants de Lyapunov de Sprott.

2.2.3.5. Aspect aléatoire :

Les figures 2.5 à 2.8 illustrent l'aspect aléatoire des états du système de Sprott :

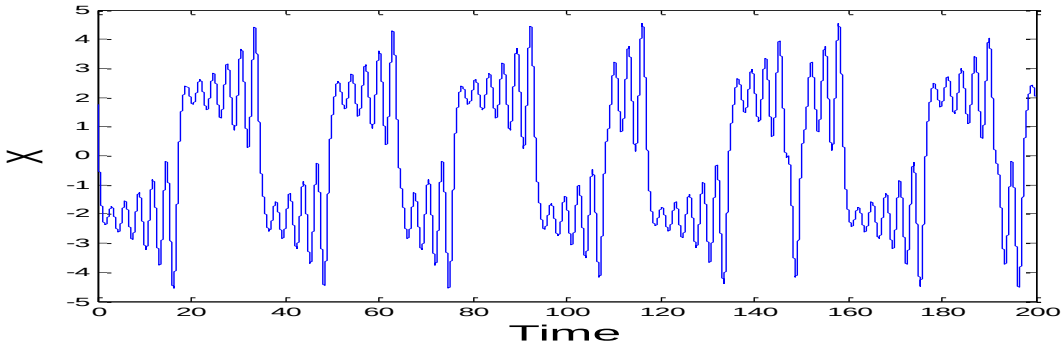


Figure 2.5 : Etats x du système de Sprott.

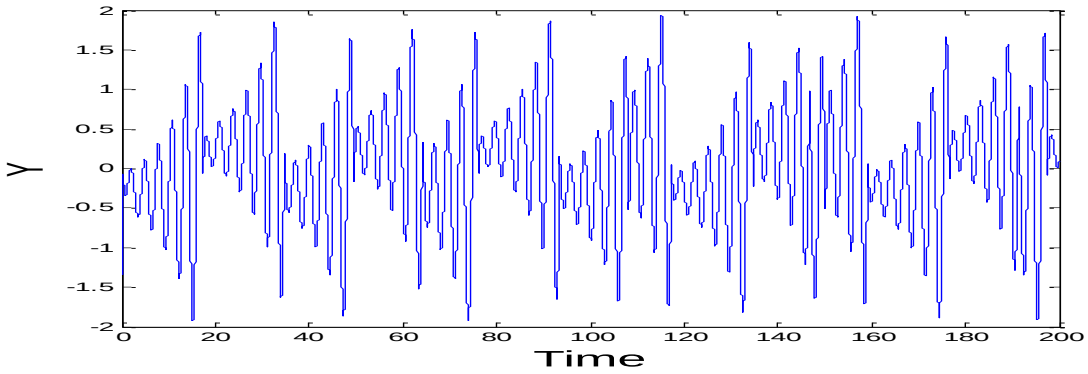


Figure 2.6 : Etats y du système de Sprott.

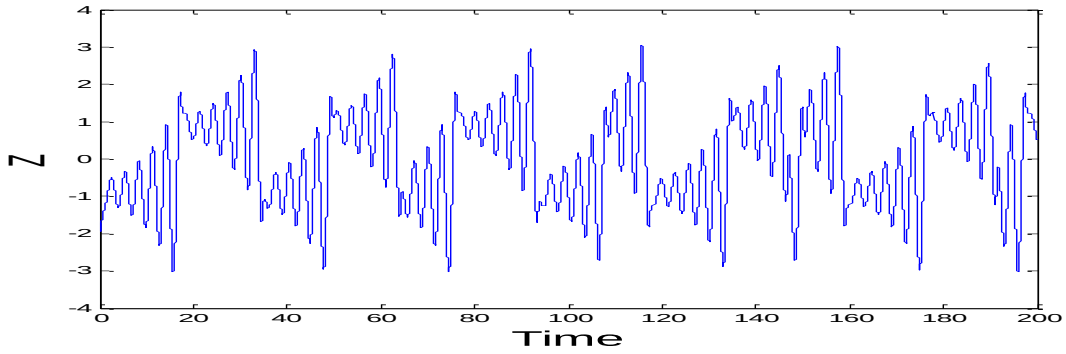


Figure 2.7 : Etats z du système de Sprott.

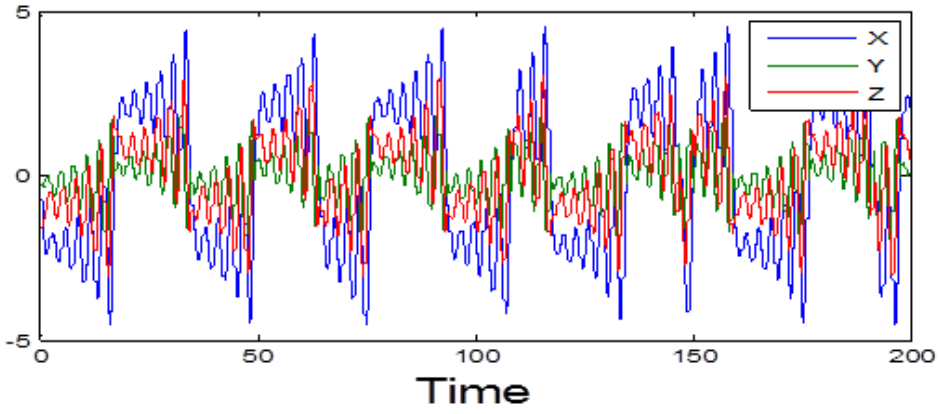


Figure 2.8 : Etats (x, y et z) du système de Sprott.

2.2.3.5. Plans de phase :

Les figures 2.9, 2.10 et 2.11 représentent les plans de phase obtenus à partir de MATLAB (Simulink) :

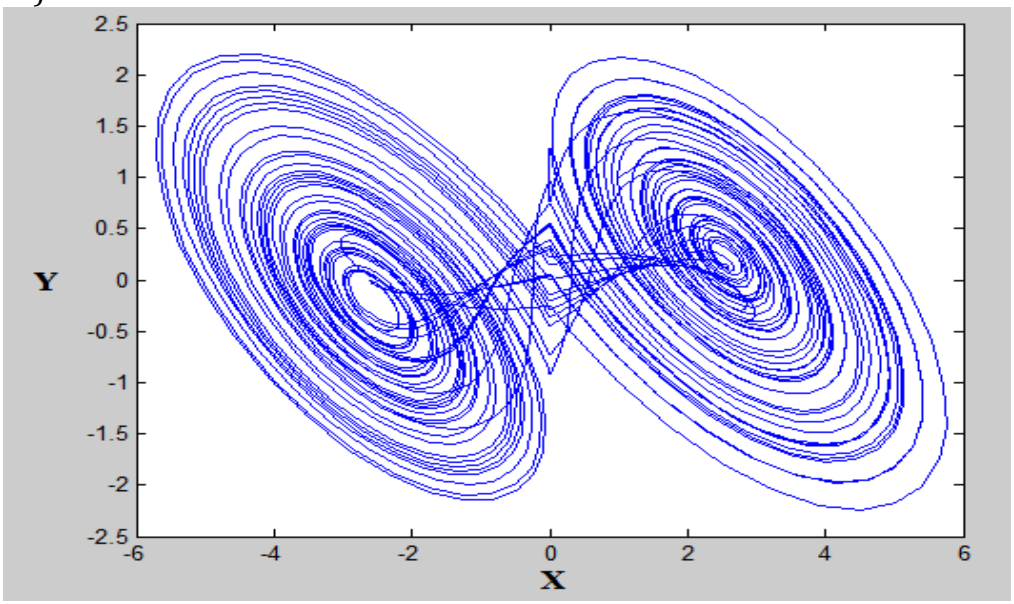


Figure 2.9 : Plan de phase (x,y) de l'oscillateur de Sprott.

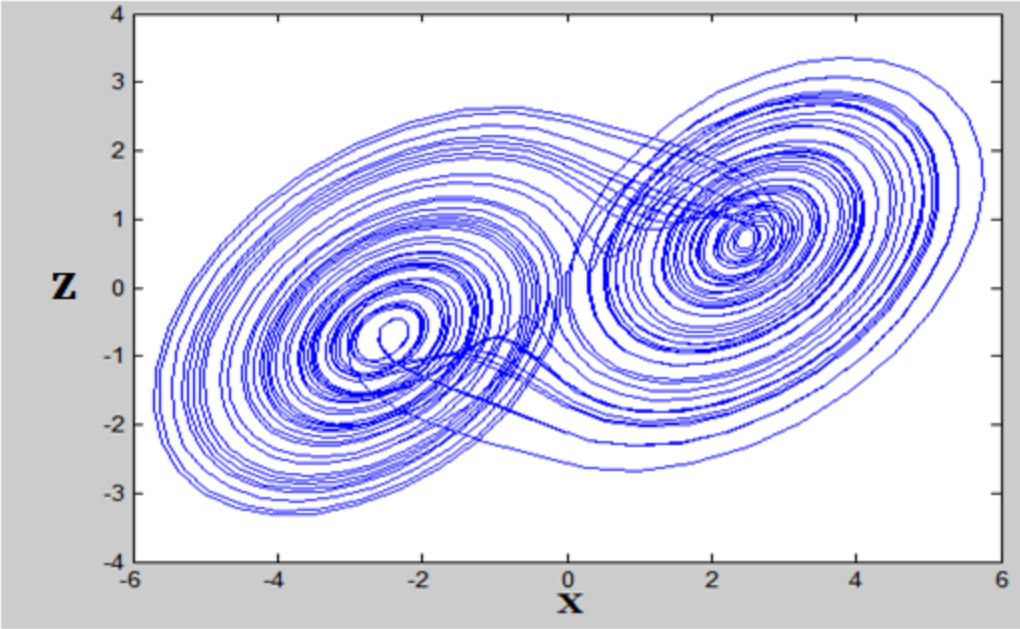


Figure 2.10 : Plan de phase (x,z) de l'oscillateur de Sprott.

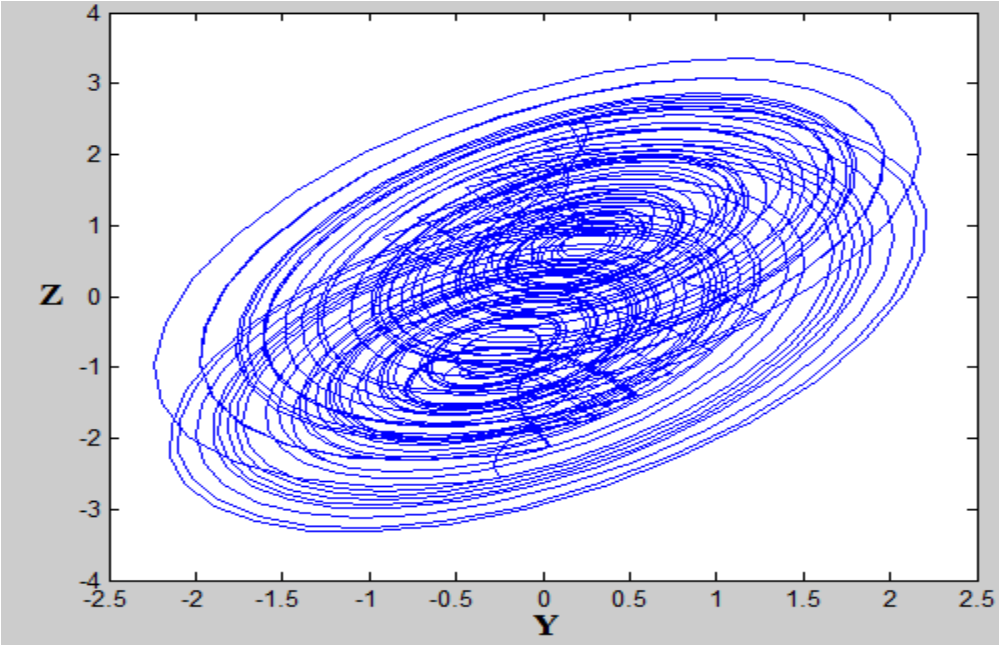


Figure 2.11 : Plan de phase (y,z) de l'oscillateur de Sprott.

2.2.3.6. Attracteur étrange :

La figure 2.12 représente l'attracteur étrange de l'oscillateur de Sprott obtenue à partir de MATLAB (Simulink) :

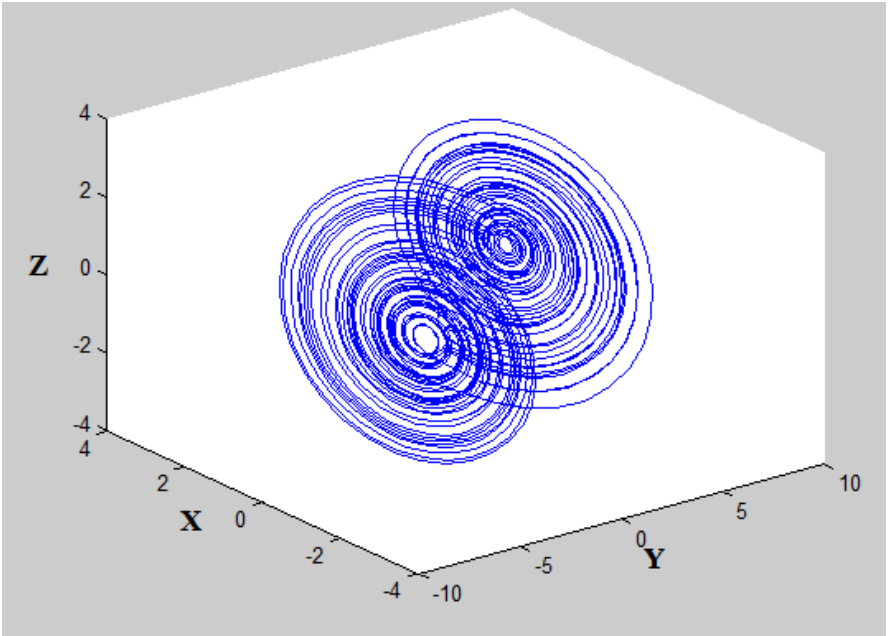


Figure 2.12 : Attracteur étrange de Sprott.

2.3. Méthodes d'insertion du message :

Un système de communication utilisant le chaos représente une application prometteuse de l'estimation d'état des systèmes non linaires.

A partir d'un message contenant l'information, l'émetteur génère un signal qui est transmis au récepteur par l'intermédiaire du canal.

Dans cette section, nous allons analyser quelques méthodes d'insertion du message dans un signal chaotique [3].

2.3.1. Insertion du message par addition :

Le principe de cette méthode est d'additionner directement notre signal informationnelle m(t) avec le signal de notre oscillateur chaotique de Sprott x(t) et de récupérer ensuite par synchronisation chaotique (voir figure 2.13). Le même oscillateur est utilisé à la fois au niveau de l'émetteur et au niveau du récepteur, avec la différence que le récepteur est contrôlé par le signal reçu de l'émetteur pour obtenir la synchronisation.

Au niveau du récepteur après synchronisation grâce au signal reçu, on récupère le message original par une simple soustraction.

Par conséquent, un intrus ne soupçonnera pas qu'un message est transmis, même s'il intercepte le signal S(t) (porteuse chaotique plus le message). Donc il ne cherchera pas à appliquer des techniques de décryptage.

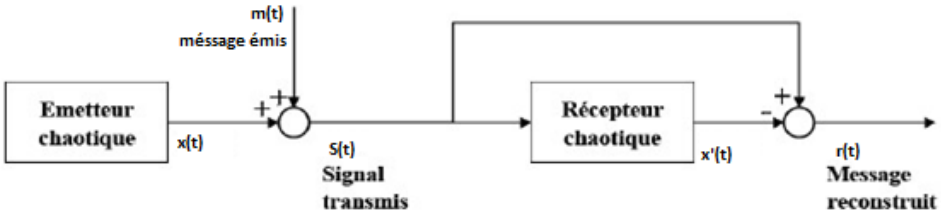


Figure 2.13 : Cryptage par addition [3].

Le principal avantage de cette méthode réside dans la simplicité du cryptage. On peut souligner que cette technique peut être appliquée à des messages continus ou discrets. L'inconvénient de cette méthode est qu'afin de garantir la synchronisation, le message doit être au moins de 20 à 30 dB inférieur à la sortie de l'émetteur. Toutefois, en présence d'un bruit de canal d'une puissance proche de celle du message, il devient difficile de détecter l'information. De plus, cette méthode reste sensible aux attaques extérieures.

2.3.2. Insertion du message par modulation paramétrique :

L'approche par modulation utilise le message contenant l'information pour moduler un ou plusieurs paramètres θ de l'émetteur chaotique. Un contrôleur adaptatif est chargé de maintenir la synchronisation au niveau du récepteur, tout en suivant le changement du paramètre modulé. Le schéma correspondant est présenté par la figure 2.14.

Au niveau de l'émetteur, le fait de moduler un ou plusieurs paramètres impose à la trajectoire un changement continu de l'attracteur et de ce fait, le signal transmis est plus complexe qu'un signal chaotique normal. Cependant, la façon d'injecter le message et donc la fonction démodulation des paramètres ne doivent pas supprimer le caractère chaotique du signal envoyé au récepteur.

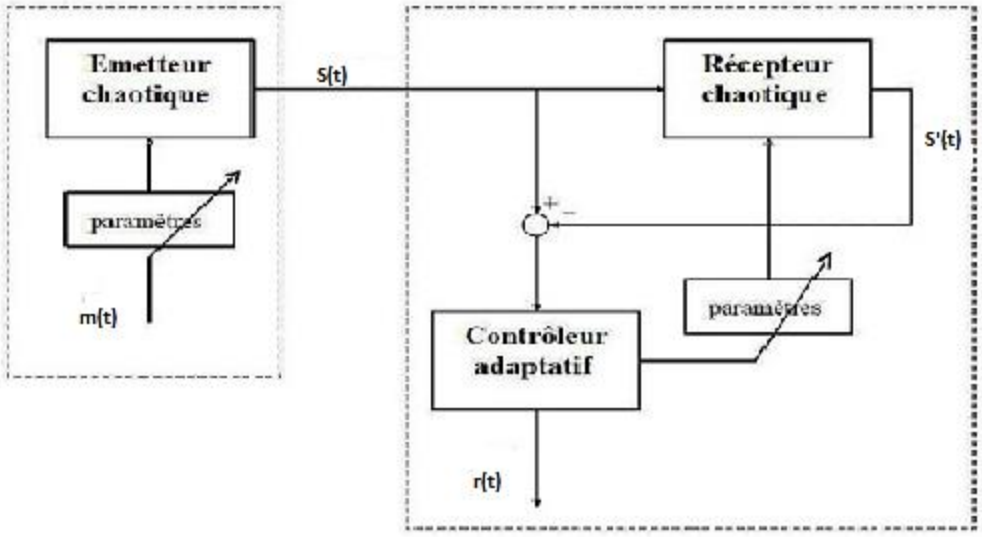


Figure 2.14 : Cryptage par modulation paramétrique [3].

2.3.3. Insertion du message par inclusion :

Cette technique de cryptage consiste à injecter le message dans la dynamique de l'émetteur.

La restauration de l'information se fait principalement par deux techniques, reposantes soit sur les observateurs à entrées inconnues soit sur l'inversion du système émetteur. Cette méthode présente beaucoup d'avantages et reste très utilisée en pratique.

2.3.4. Insertion du message mixte :

Afin de faire face aux problèmes de sécurité des méthodes précédentes, une nouvelle technique combinant les principes de la cryptographie standard et la synchronisation chaotique a été proposée. Le message u(t)contenant l'information est crypté 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 2.15.

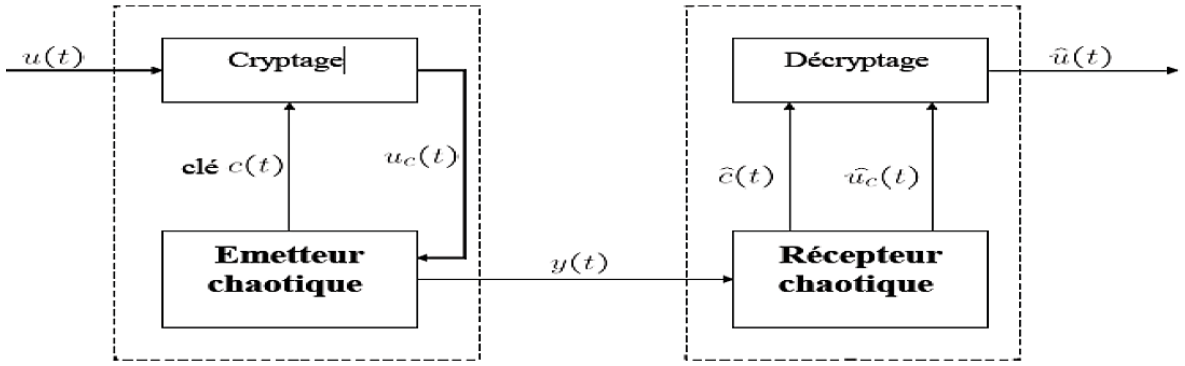


Figure 2.15 : Cryptage mixte [3].

2.4. Etude de l'émetteur chaotique [8] :

La figure 2.15 représente l'émetteur chaotique qui contient deux blocs :

- Bloc1 : l'oscillateur chaotique de Sprott.
- Bloc 2 : le circuit d'insertion du message (méthode par addition).

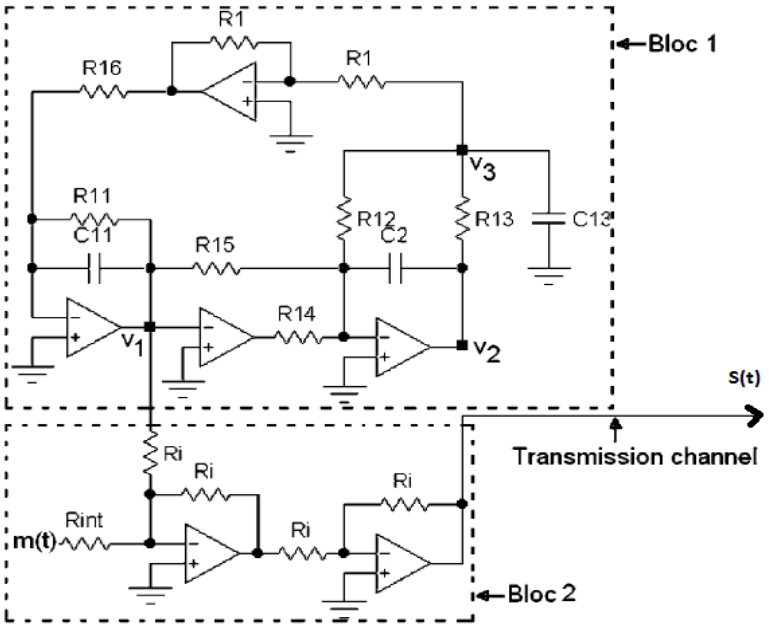


Figure 2.16 : Montage de l'émetteur chaotique.

2.4.1. Les équations de l'émetteur :

L'émetteur est décrit par les équations suivante où m(t) est le message que l'on veut crypter :

$$\dot{x} = -Q_1 x + Q_2 z \tag{2.29}$$

$$\dot{y} = -Q_3 x - Q_4 z + Q_5 sign(x) \tag{2.30}$$

$$\dot{z} = Q_6 y - Q_7 z \tag{2.31}$$

$$S(t) = m(t) + x(t) \tag{2.32}$$

La figure 2.17 représente le schéma de l'émetteur implémenté sous MATLAB (Simulink)

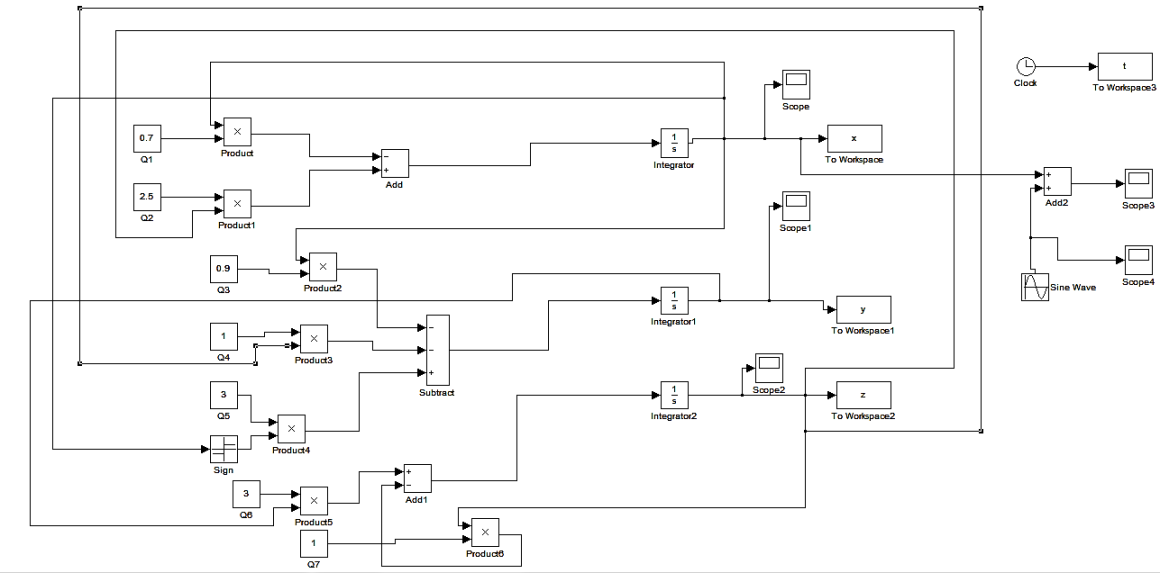


Figure 2.17: Implémentation de l'émetteur chaotique sous MATLAB (Simulink)

2.4.2. Visualisation des signaux :

A l'aide du logiciel MATLAB (Simulink) on visualise les signaux suivants :

- Le signal informatif que l'on veut crypter m(t) représenté en figure 2.18.
- Le signal chaotique x(t) représenté en figure 2.19.
- Le signal chaotique crypté et transmis dans le canal de transmission s(t) représenté en figure 2.20.

Dans la simulation, le signal informatif utilisé m(t) est un signal sinusoïdal

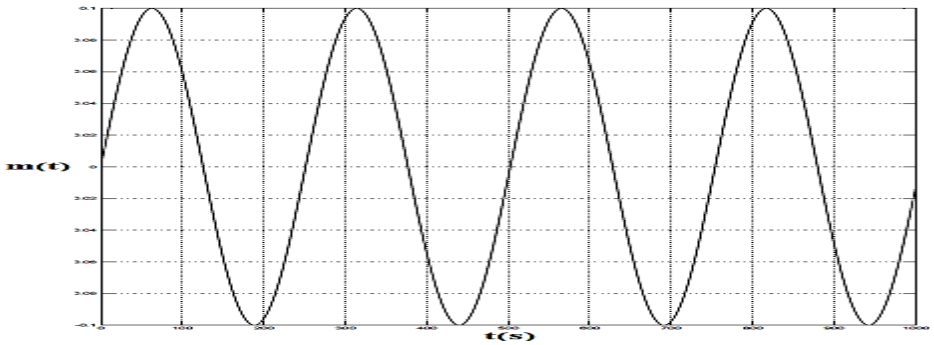


Figure 2.18 : Signal émis m(t).

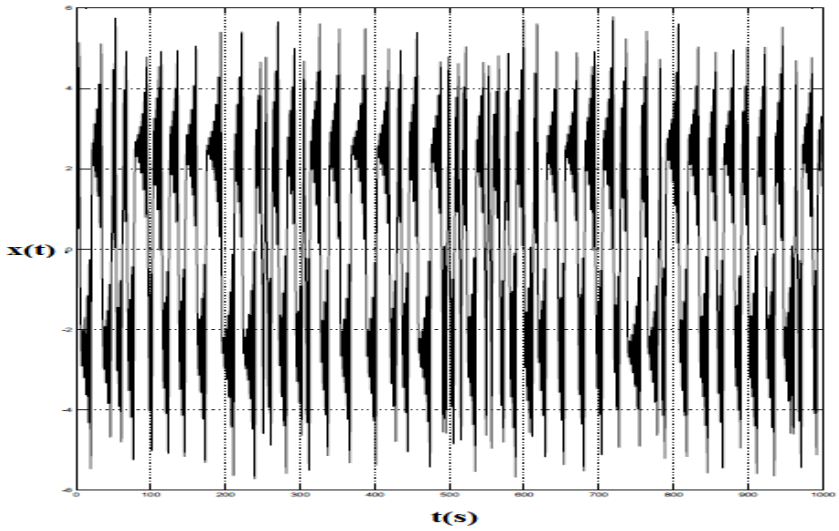


Figure 2.19 : Signal chaotique x(t).

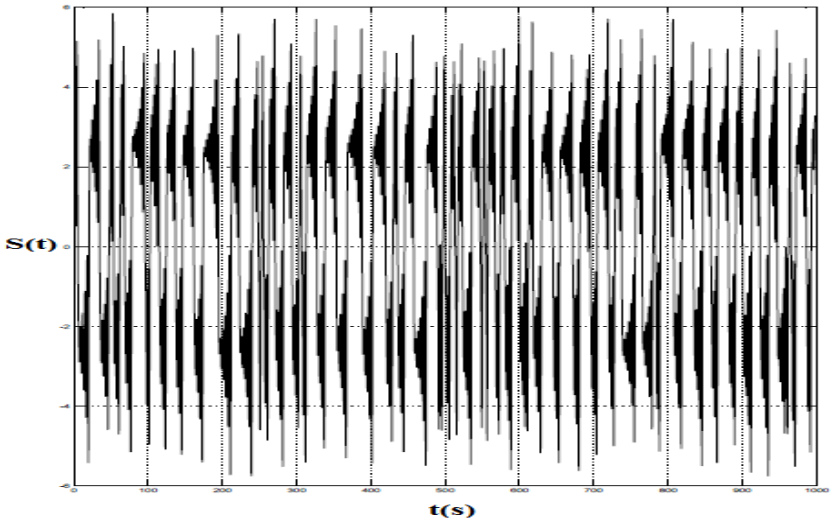


Figure 2.20 : Signal transmis s(t).

2.5. Conclusion :

Dans ce chapitre on a étudié les principales caractéristiques de l'oscillateur chaotique de Sprott qui constitue notre émetteur, en se basant sur les définitions présentées au chapitre précédent. Ensuite on a utilisé l'insertion du message par addition parmi les différentes méthodes d'insertion pour le cryptage. Enfin on a présenté l'ensemble des signaux mis en jeu au niveau de l'émetteur en mettant en relief, le caractère crypté du signal transmis.

Chapitre 3 : Synchronisation chaotique : Etude du récepteur

3.1. Introduction :

Dans les systèmes de communication, la synchronisation est une clé très importante pour une transmission réussie. La synchronisation classique employée dans les systèmes de télécommunication cherche à reproduire juste le signal périodique de la porteuse. Par contre, la synchronisation chaotique au niveau du récepteur cherche à dupliquer le signal chaotique envoyé de l'émetteur. Cela veut dire que deux signaux chaotiques seront dit synchronisés s'ils sont asymptotiquement identiques lorsque le temps « t » tend vers l'infini. Dans le premier chapitre nous avons déjà vu la sensibilité du chaos aux conditions initiales, il apparait alors que la synchronisation chaotique n'est pas si simple à établir et pose plus de contraintes qu'une synchronisation classique [9].

3.2. Les classes de synchronisation :

Le concept de synchronisation repose sur le constat qu'un système chaotique est déterministe et possède un ou plusieurs exposants de Lyapunov positifs et qu'il est instable. Il est donc possible de construire une réplique identique à ce système et d'essayer de le synchroniser de façon que les deux signaux chaotiques issus des deux exemplaires soient identiques.

Il existe deux classes de synchronisation suivant la manière avec laquelle les deux systèmes chaotiques sont couplés : unidirectionnelle et bidirectionnelle [10].

3.2.1. Synchronisation unidirectionnelle :

Dans le cas d'une synchronisation unidirectionnelle, le couplage entre deux systèmes identiques a et b est réalisé à l'aide d'un élément qui fonctionne dans un seul sens, par exemple l'utilisation d'un circuit électrique suiveur. La figure 3.1 représente le couplage unidirectionnel.

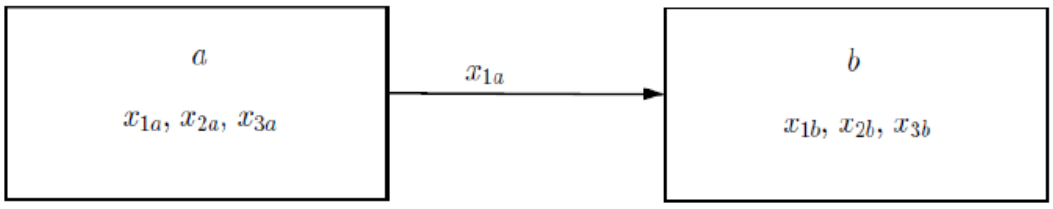


Figure 3.1 : Couplage unidirectionnel [10].

3.2.2. Synchronisation bidirectionnelle :

Dans le cas d'une synchronisation bidirectionnelle, le couplage entre deux systèmes identiques a et b (figure 3.2) est réalisé à l'aide d'un élément permettant l'échange d'énergie dans les deux sens, par exemple l'utilisation d'une simple résistance.

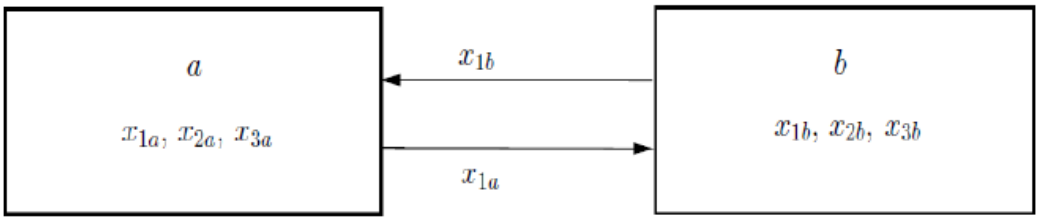


Figure 3.2 : Couplage bidirectionnel [10].

3.3. Méthodes de synchronisation :

Plusieurs méthodes de synchronisation ont été proposées dans la littérature. Dans ce qui suit nous citerons quelques approches en expliquant leurs principes et avantages [3].

3.3.1. Synchronisation par boucle fermé :

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 le premier système et le signal régénéré par l'autre. Cette erreur est ainsi injectée en contre-réaction d'où l'appellation de l'approche.

Cette technique permet également la synchronisation entre des paires différentes de systèmes chaotiques. La figure 3.3 indique un schéma simplifié de la synchronisation par boucle fermée.

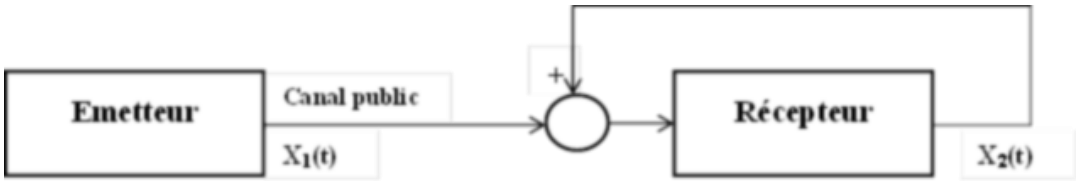


Figure 3.3 : Synchronisation par boucle fermée [3].

3.3.2. Synchronisation par répartition du système :

Pour illustrer la méthode de synchronisation par couplage entre deux systèmes chaotiques, nous avons choisi de présenter la synchronisation identique proposée par

Pecora et Carroll. L'avantage de cette approche est de représenter une solution simple et performante. L'objectif est qu'un système esclave peut reproduire le plus fidèlement possible l'état du système maître, après un régime transitoire. L'idée consiste à diviser le système d'origine en deux sous-systèmes de telle sorte que les variables dynamiques de départ soient réparties de part et d'autre de chacun des sous-systèmes. Il s'agit ensuite de reproduire les sous-systèmes à l'identique et de les mettre en cascade. Le signal issu du système de départ (système maître) sert à piloter (synchroniser) le premier des deux sous-systèmes dupliqués mis en cascade, qui lui-même permet de synchroniser le second sous-système dupliqué, Partant d'un système chaotique défini par la dynamique suivante :

$$\dot{x}(t) = f\big(x(t)\big) \tag{3.1}$$

où : $x = [x_1, x_2, \dots\dots, x_n]$ désigne le vecteur d'état.

On divise le système initial en deux sous-systèmes avec une réorganisation des variables d'état dans un ordre quelconque.

$$S_1 : \dot{x}^{\{1\}} = f^1\big(x^{\{1\}}, x^{\{2\}}\big) \tag{3.2}$$

$$S_2 : \dot{x}^{\{2\}} = f^2\big(x^{\{1\}}, x^{\{2\}}\big) \tag{3.3}$$

avec :

$$x^{\{1\}} = [x_1, \dots\dots, x_m]^T \tag{3.4}$$

$$x^{\{2\}} = [x_{m+1}, \dots\dots, x_n]^T \tag{3.5}$$

$$f(x) = \big[f^{\{1\}}(x)\ ; f^{\{2\}}(x)\big] \tag{3.6}$$

Soit un autre système $S_2^{'}$ de dynamique identique $f^{\{2\}}$ et un vecteur d'état $\hat{x}^{\{2\}}$:

$$S_2^{'} : \hat{x}^{\{2\}}\big(x^{\{1\}}, \hat{x}^{\{2\}}\big)$$

Pecora et Caroll 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. Une convergence parfaite des trajectoires est ainsi accomplie pour :

$$\lim_{n\to\infty} \big\| \hat{x}^{\{2\}}(t) - x^{\{2\}}(t) \big\|$$

La figure 3.4 représente le schéma synoptique de cette synchronisation maitre-esclave :

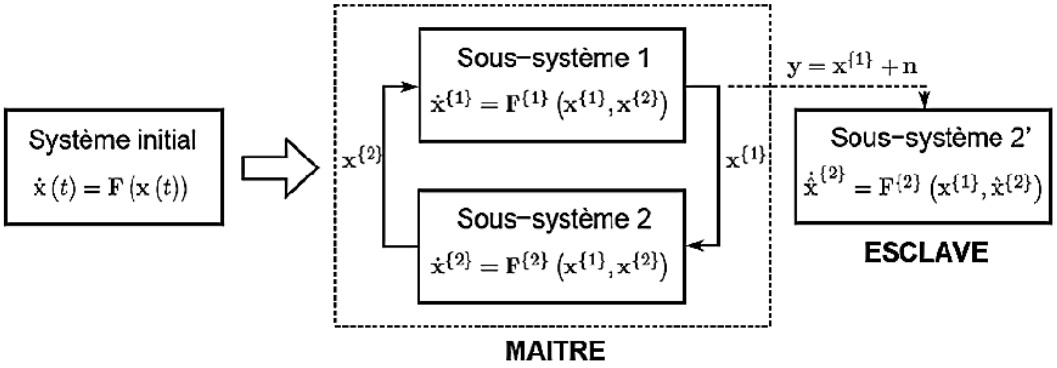


Figure 3.4 : Synchronisation maitre-esclave [3].

3.3.3. Synchronisation généraliste :

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

$$\lim_{n\to\infty} \|y(t) - M(x(t))\| = 0 \tag{3.7}$$

où : $x(t)$ l'état du système émetteur et $y(t)$ est l'état du système récepteur.

Les conditions initiales ne sont pas tenues en compte dans ce cas. Si M est inversible, alors $M^{-1}(y)$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.

Dans la synchronisation retardée, l'état du système esclave converge vers l'état décalé dans le temps du système maitre.

$$\lim_{n\to\infty} \|y(t) - x(t-\tau)\| = 0 \tag{3.8}$$

où$x(t)$ est l'état du système émetteur, $y(t)$est l'état du système récepteur et τ est un retard positif.

3.3.4. Synchronisation projective :

Dans cette méthode, l'état du système récepteur se synchronise avec un multiple de l'état du système émetteur. Soit α et τ tels que :

$$\lim_{n\to\infty} \|x'(t) - \alpha x(t-\tau)\| = 0 \tag{3.9}$$

où α est le facteur d'échelle, $x(t)$ est l'état du système émetteur, $x'(t)$ est l'état du système récepteur et τ 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.

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 :

$$\dot{x}(t) = f\big(x(t)\big) \tag{3.10}$$

On définit un signal impulsif qui consiste en une suite d'instants discrets auxquelles 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. La figure 3.5 représente le schéma synoptique de la synchronisation impulsive.

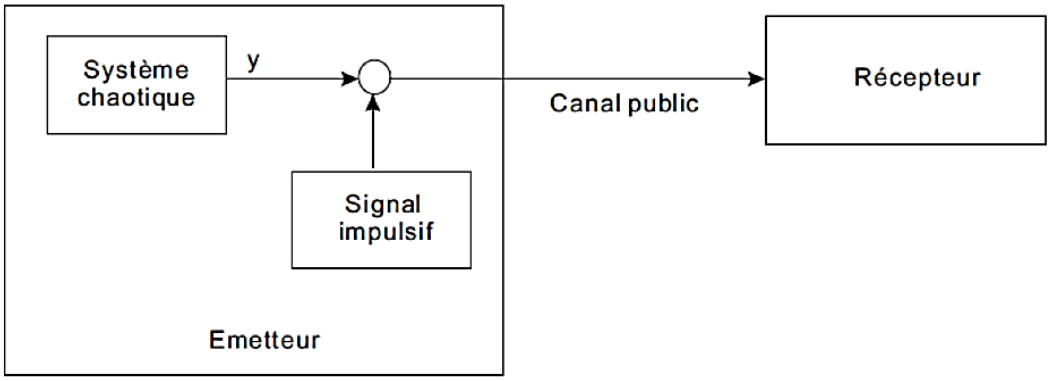


Figure 3.5 : Synchronisation impulsive [3].

3.4 Propriétés des systèmes chaotiques appliqués au cryptage d'une transmission de données :

Dans cette partie, des propriétés des systèmes de communication chaotiques seront étudiées et comparées aux propriétés des systèmes classiques [11].

3.4.1 Spectre à large bande :

Les systèmes chaotiques ont spécifiquement un spectre à large bande. Cette propriété est bénéfique pour les applications qui nécessitent une importante robustesse face aux interférences et une faible probabilité de détection.

Ces problèmes ont été pris en compte par les premiers systèmes de transmission en utilisant des spectres larges et des modulations par saut de fréquences. Cependant malgré le recours à ces moyens, la synchronisation entre l'émetteur et le récepteur reste une tâche qui n'est pas toujours triviale. En effet les schémas de transmission qui utilisent un saut de fréquence requièrent une nouvelle synchronisation à chaque changement de fréquence de la porteuse. Donc l'utilisation des systèmes chaotiques permet la transmission des signaux à large bandes, ainsi la synchronisation entre l'émetteur et le récepteur est plus simple.

3.4.2 Signal non périodique :

La périodicité, dans la communication sécurisée engendre des pics spectraux indésirables.

Par contre, un signal chaotique est non périodique et son évolution ne peut être prédite sur un long intervalle de temps. Par conséquent, il y a absence des pics spectraux. De plus il est plus difficile de développer un modèle de prévisions pour les dynamiques non périodiques.

3.5 Etude du récepteur chaotique :

3.5.1 Récepteur chaotique :

La figure 3.6 représente les différents blocs récepteurs chaotiques [8]:

- Bloc 1 : synchronisation (méthode par boucle fermée).
- Bloc 2 : oscillateur chaotique de Sprott.
- Bloc 3 : message informatif décrypté.

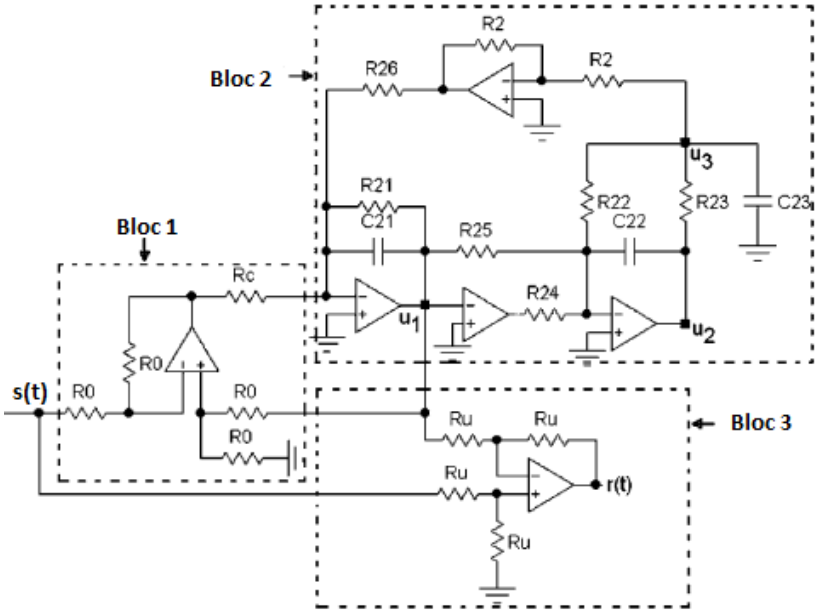


Figure 3.6 : Récepteur chaotique.

3.5.2 Analyse de la synchronisation chaotique :

Les équations de l'émetteur et du récepteur sont respectivement :

$$\begin{cases} \dot{x}_1 = -Q_{11}\dot{x}_1 + Q_{12}z_1 \\ \dot{y}_1 = -Q_{13}x_1 - Q_{14}z_1 + Q_{15}sign(x_1) \\ \dot{z}_1 = Q_{16}y_1 - Q_{17}z_1 \end{cases} \quad (3.11)$$

$$\begin{cases} \dot{x}_2 = -Q_{21}x_2 + \dot{Q}_{22}z_2 + K(x_1 - x_2) \\ \dot{y}_2 = -Q_{23}x_2 - Q_{24}z_2 + Q_{25}sign(x_2) \\ \dot{z}_2 = Q_{26}y_2 - Q_{27}z_2 \end{cases} \quad (3.12)$$

Les paramètres Q_{ij} sont définie dans l'oscillateur chaotique de Sprott $Q_j (1 \leq j \leq 7)$ avec $i = 1{,}2$ de sorte que $i = 1$ correspond à l'oscillateur chaotique émetteur(Maitre) et $i = 2$ l'oscillateur chaotique récepteur (esclave).

Les équations 3.11 et 3.12 peuvent s'écrire sous les formes réduites suivantes :

$$\dddot{x}_1 + (Q_7 + Q_1)\ddot{x}_1 + (Q_4Q_6 + Q_1Q_7)\dot{x}_1 + (Q_2Q_3 + Q_1Q_4)Q_6x_1 - Q_5Q_2Q_6sign(x_1) = 0 \quad (3.13)$$

$$\dddot{x}_2 + (Q_7 + Q_1)\ddot{x}_2 + (Q_4Q_6 + Q_1Q_7)\dot{x}_2 + (Q_2Q_3 + Q_1Q_4)Q_6x_2 - Q_5Q_2Q_6sign(x_2) - K[(\ddot{x}_1 - \ddot{x}_2) + Q_7(\dot{x}_1 - \dot{x}_2) + Q_4Q_6(x_1 - x_2)] = 0 \quad (3.14)$$

où $\dddot{x}$, $\ddot{x}$ *et* $\dot{x}$ représentent respectivement la troisième, deuxième et première dérivée par rapport au temps.

Dans notre étude, les conditions initiales et les paramètres ne sont pas exactement connus au niveau du récepteur. L'erreur de synchronisation entre le système maitre (émetteur) et le système esclave (récepteur) est donnée pour la variable x par :

$$e = x_1 - x_2 \tag{3.15}$$

On obtient alors l'équation suivante, en faisant intervenir l'erreur :

$$\dddot{e} + (Q_7 + Q_1 + K)\ddot{e} + (Q_4Q_6 + Q_1Q_7 + Q_7K)\dot{e} + Q_6(Q_2Q_3 + Q_1Q_4 + Q_4K)e - Q_5Q_2Q_6\big(sign(x_1) - sign(x_2)\big) = 0 \tag{3.16}$$

On définit le paramètre μ par :

$$\mu\begin{cases} = 1 & si\ x_1 > 0\ et\ x_2 < 0 \\ = 0 & si\ \ x_1\ et\ x_2 > 0 \\ = 1 & si\ x_1 < 0\ et\ x_2 > 0 \end{cases} \tag{3.17}$$

L'erreur de la synchronisation devient :

$$\dddot{e} + (Q_7 + Q_1 + K)\ddot{e} + (Q_4Q_6 + Q_1Q_7 + Q_7K)\dot{e} + Q_6[(Q_2Q_3 + Q_1Q_4 + Q_4K)e - 2\mu Q_5Q_2] = 0 \tag{3.18}$$

La solution de l'équation 3.18 est de la forme suivante :

$$e = Aexp(\lambda t) + e_0 \tag{3.19}$$

$$\text{avec} e_0 = \frac{2\mu\, Q_5Q_2}{Q_2Q_3 + Q_4(Q_1 + K)} \tag{3.20}$$

En remplacent (3.19) et (3.20) dans (3.18) on obtient l'équation suivante :

$$\lambda^3 + (Q_7 + Q_1 + K)\lambda^2 + (Q_4Q_6 + Q_1Q_7 + Q_7K)\lambda + Q_6(Q_2Q_3 + Q_1Q_4 + Q_4K) = 0 \tag{3.21}$$

On pose :

$$\alpha = (Q_7 + K)^2 - (Q_1 + K)Q_7 + Q_7^2 - 3Q_4Q_6 \tag{3.22}$$

$$\beta = (Q_1 + K)^3 - \frac{3}{2}Q_7(Q_1 + K)^2 - \frac{3}{2}(Q_7^2 - 6Q_4Q_6)(Q_1 + K) + Q_7^3 + \frac{9}{2}Q_6(3Q_2Q_3 - Q_4Q_7) \tag{3.23}$$

La solution complexe de l'équation 3.21 est donnée par :

$$\lambda = \frac{-2(Q_7 + Q_1 + K) + (1 \pm j\sqrt{3})\left(\sqrt[3]{\left(\sqrt{\beta^2 - \alpha^3} + \beta\right)}\right) - (1 \pm j\sqrt{3})\left(\sqrt[3]{\left(\sqrt{\beta^2 - \alpha^3} - \beta\right)}\right)}{6} \tag{3.24}$$

D'après l'équation (3.18), la synchronisation est stable quand $\lim_{t\to\infty} e(t) = 0$. Le domaine de la stabilité est analytiquement déterminé par la condition $\Re e(\lambda) < 0$, où $\Re e(\lambda)$ représente la partie réelle de λ.

On obtient l'inégalité suivante :

$$-2(Q_7 + Q_1 + K) + \sqrt[3]{\left(\sqrt{\beta^2 - \alpha^3} + \beta\right)} - \sqrt[3]{\left(\sqrt{\beta^2 - \alpha^3} - \beta\right)} < 0 \qquad (3.25)$$

On résolvantl'inégalité de l'équation (3.22), on trouve la valeur minimale K_{min} :

$$K_{min} = -\left(\frac{Q_7}{2} + Q_1\right) + \frac{1}{2Q_7}\sqrt{(Q_7^4 + 4Q_7Q_2Q_6Q_3 - 4Q_7^2Q_4Q_6)} \qquad (3.26)$$

La synchronisation aura lieu si $K > K_{min}$.

La figure 3.7 représente le schéma de transmission chaotique implémenté sous MATLAB (Simulink). On y distingue, le bloc émetteur, le bloc récepteur et le circuit permettant la synchronisation par boucle fermée.

avec pour l'oscillateur maitre les valeurs:

$Q_{11} = 0.7231$, $Q_{12} = 2.569$, $Q_{13} = 1.12$, $Q_{14} = 1.045$, $Q_{15} = 3.026$, $Q_{16} = 0{,}0875$ *et* $Q_{17} = 0{,}0875$

et pour l'oscillateur esclave les valeurs :

$Q_{21} = 0.723$, $Q_{22} = 2.566$, $Q_{23} = 1.15$, $Q_{24} = 1.043$, $Q_{25} = 3.024$, $Q_{26} = 3.0878$ *et* $Q_{27} = 1.017$

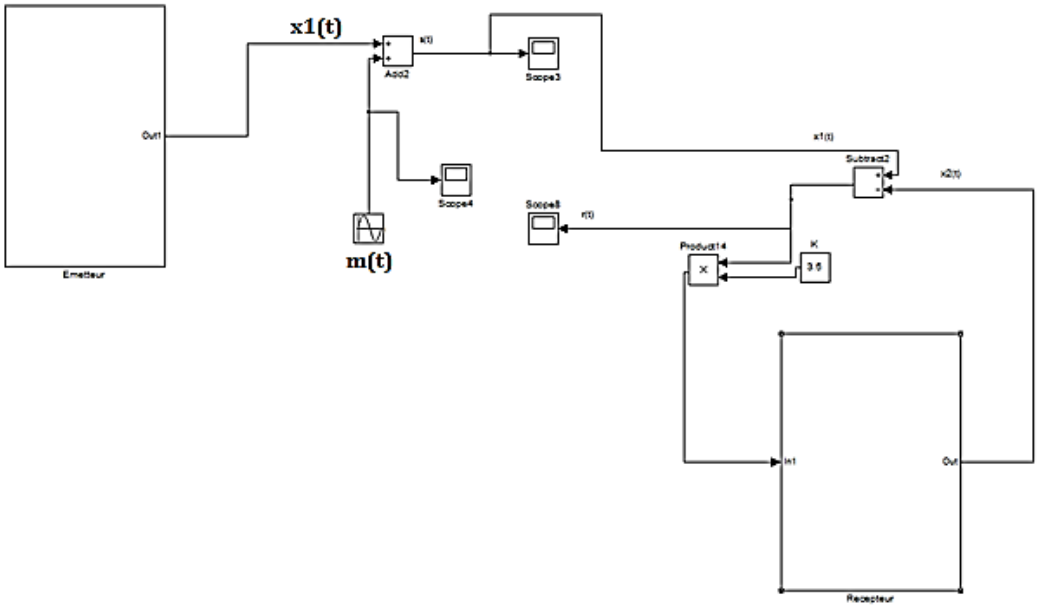


Figure 3.7 : Schéma complet MATLAB (Simulink).

Après la simulation du système « Emetteur-Récepteur » sous MATLAB (Simulink), on visualise les signaux suivants :

- Le signal émis m(t) représenté en figure 3.8 ;
- Le signal chaotique $x_1(t)$ représenté en figure 3.9 ;
- Le signal crypté $s(t)= x_1(t)+m(t)$ représenté en figure 3.10 ;
- Le signal informatif décrypté r(t) représenté en figure 3.11 ;
- La caractéristique $x_1(t)$ en fonction de $x_2(t)$ représentée en figure 3.12.

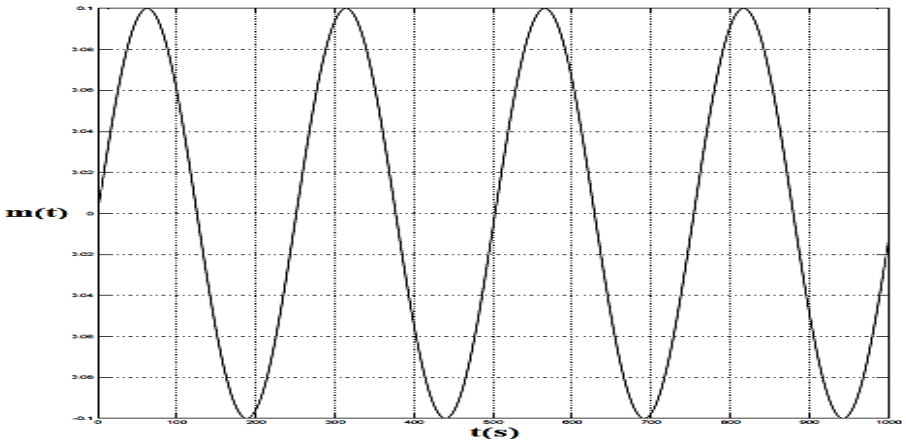


Figure 3.8 : Le signal informatif m(t).

Les figures suivantes représentent les différents signaux obtenus lors de la simulation sous MATLAB (Simulink) :

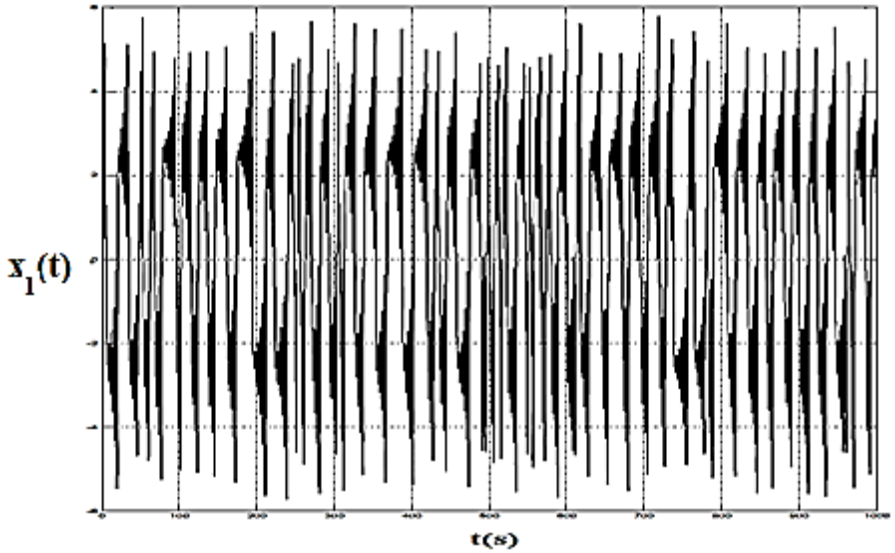


Figure 3.9 : Le signal chaotique $x_1(t)$.

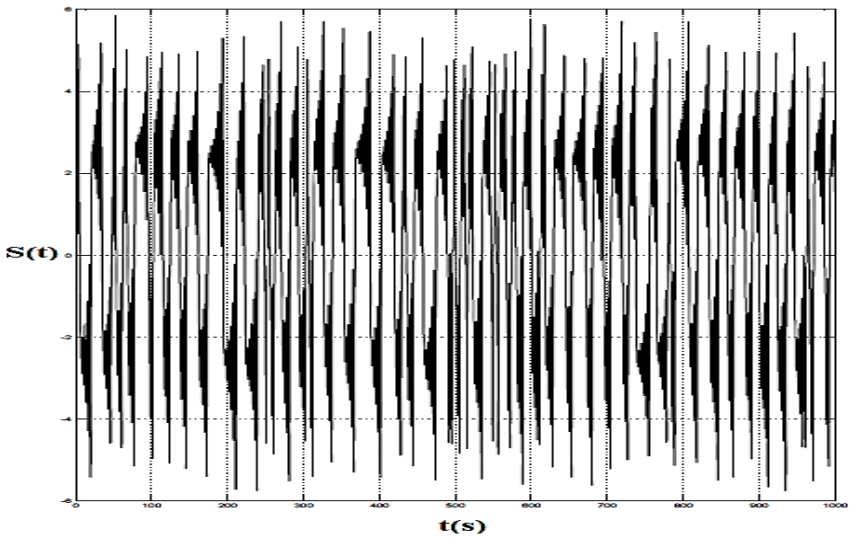


Figure 3.10 : Le signal crypté s(t).

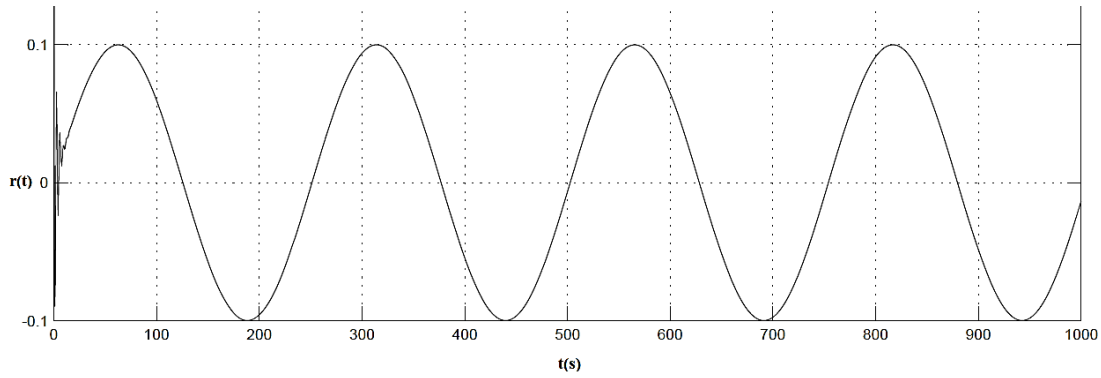


Figure 3.11 : Le signal décrypté r(t).

La figure 3.12 représente le signal x_2 en fonction de x_1. On remarque bien qu'on a obtenu une droite qui signifie que $x_1 = x_2$. Cela explique que pour K=3,5 notre système est synchronisé.

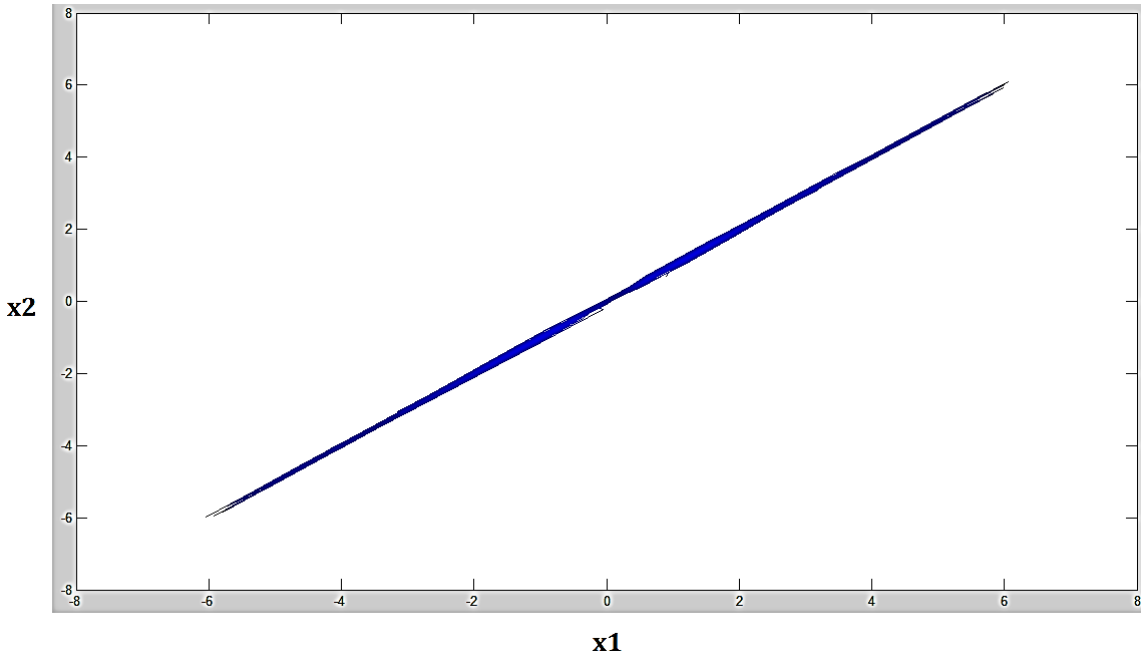


Figure 3.12 : Signal x_2 en fonction de x_1 avec K=3,5

La figure 3.13 représente l'erreur $e = x_1 - x_2$ obtenue sous MTLAB (Simulink) avec K=3,5

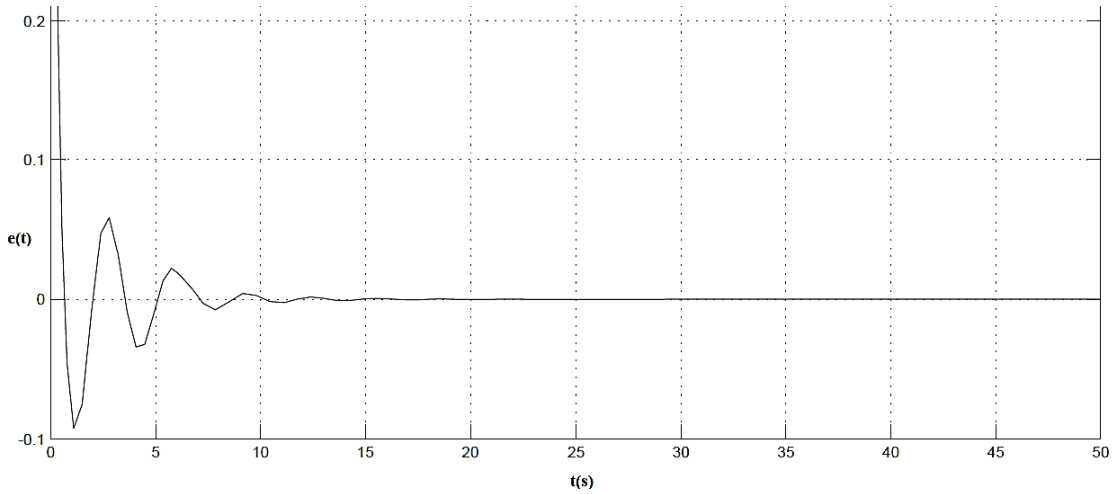


Figure 3.13 : L'erreur e(t) = $x_1(t)$-$x_2(t)$

La figure 3.14 représente le signal x_2 en fonction de x_1. Dans le cas ou K=0,5 notre système est non synchronisé.

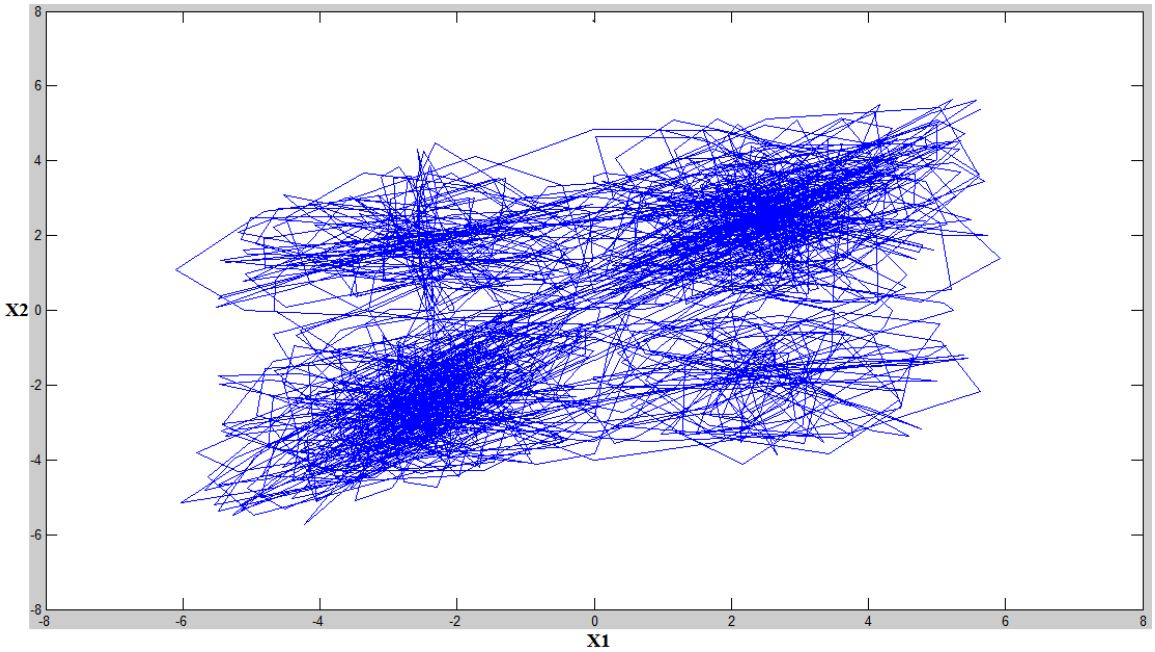


Figure 3.14 : Signal x_2 en fonction de x_1 avec K=0,5

3.5 Conclusion :

Dans ce chapitre, nous avons présenté les différents types de couplages (unidirectionnels et bidirectionnels), ainsi que les différentes méthodes de synchronisation. Ensuite nous avons étudié le récepteur chaotique de Sprott avec l'approche utilisée pour la synchronisation en utilisant la méthode par boucle fermée. La

condition de synchronisation a été étudiée en calculant la valeur minimale K_{min} du coefficient de couplage. Les résultats de simulation ont été présentés et la récupération du message a été obtenue grâce à la synchronisation chaotique.

Chapitre 4 : Implémentation sur circuit FPGA de la transmission chaotique

4.1 Introduction :

De nos jours, grâce aux avancées spectaculaires qu'a connu la technologie, de nombreuses applications complexes qui relèvent du domaine du traitement du signal numérique ont vu le jour, Parmi ces derniers, on peut citer : la téléphonie mobile, la sécurisation de données (cryptographie), etc.

Ces applications sont développées soit à base de microcontrôleurs, DSPs (Digital Signal Processor) ou FPGAs(Field Programmable Gate Arrays ou "réseaux logiques programmables"). Aussi dans le cadre de notre projet nous avons opté pour la technologie FPGA pour la réalisation de notre système de transmission chaotique de Sprott. En effet l'avantage de cette technique est de se passer d'une réalisation couteuse qui consisterait en la réalisation d'un émetteur et d'un récepteur en éléments analogiques discrets.

4.2 Présentation des circuits FPGA :

Les FPGA sont des composants entièrement reconfigurables, ce qui permet de les reprogrammer à volonté afin d'accélérer notablement certaines phases de calculs.

L'avantage de ce genre de circuit est sa grande souplesse qui permet de les réutiliser à volonté dans des algorithmes différents en un temps très court [12].

4.2.1 Architecture des FPGA :

Les circuits FPGA sont constitués d'une matrice de blocs logiques programmables entourés de blocs d'entrée sortie programmable IOB. L'ensemble est relié par un réseau d'interconnexions programmable.

La figure 4.1 présente l'architecture générique d'un circuit FPGA.

4.2.2 Technologies des FPGA :

Il existe actuellement plusieurs fabricants de circuits FPGA tel que : Xilinx, Microsemi (ex. Actel), Altera, Atmel, Cypress, LatticeSemiconductor, Nallatech, QuickLogic, SiliconBlue, Tabula Inc., TierLogic …,etc [13].

Chacun de ces constructeurs utilisent différentes technologies FPGA reprogrammables pour la fabrication des FPGA, Parmi ces technologies reprogrammables, en peut citer :

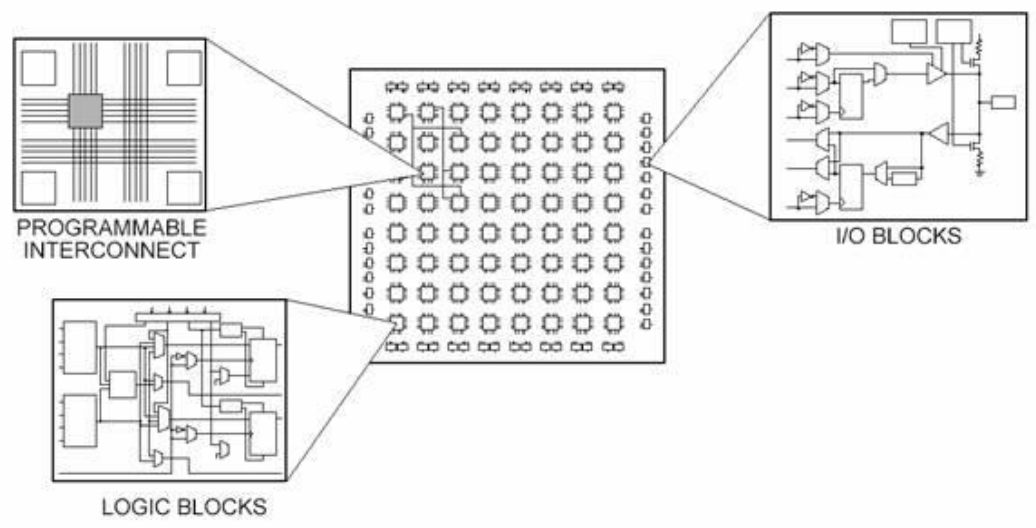


Figure 4.1 : Architecture générique d'un circuit FPGA.

- SRAM - (StaticRandom Access Memory) : pour cette technologie, les connexions sont réalisées en rendant les transistors passants ce qui permet une reconfiguration rapide du circuit FPGA.
- EPROM (UVPROM) - (Erasable Programmable Read-Only Memory). Peuvent être effacés (et reprogrammés) par exposition aux rayons ultra-violets.
- EEPROM - (Electrically Erasable Programmable Read-Only Memory). Peuvent être effacés et reprogrammés à volonté par source électrique.
- Flash - (Flash-erase EPROM). Mêmes propriétés qu'EEPROM mais avec une densité supérieure (donc avec un coût inférieur pour une complexité donnée).
- Fusible - Programmables une seule fois. Technologie bipolaire.
- Anti-fusible - Ne sont programmables qu'une seule fois.

Le tableau 4.1 montre les différents avantages et inconvénients de chaque technologie :

4.2.3 Application des FPGA :

Les FPGA sont utilisés dans de nombreuses applications dont on peut citer comme exemple :

- Prototypage de nouveaux circuits.
- Fabrication de composants spéciaux en petite série.

- Adaptation aux besoins rencontrés lors de l'utilisation.
- Systèmes de commande à temps réel.
- Cryptographie.
- Imagerie médicale.

Technologie	Avantages	Inconvénients
SRAM	Reprogrammation rapide	Nécessité d'une grande surface
EPROM	Reprogrammable à volonté	Nécessité d'une source UV pour la programmation
EEPROM	Reprogrammable à volonté électriquement	-
Flash	Sauvegarde le programme en cas de coupure de l'alimentation	Nombre de reconfigurations limité
Fusible	/	Programmables une seule fois
Anti-fusible	Reprogrammable à volonté	-

Tableau 4.1 : Avantages et inconvénients des technologies FPGA.

4.3 Processus d'implémentation :

La conception des architectures de commande s'effectue en utilisant les outils de Conception Assistée par Ordinateur (CAO). La saisie est effectuée graphiquement ou via un langage de description matériel de haut niveau, nommé également langage HDL (Hardware Description Language). 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. La figure 4.3 résume les différentes étapes de programmation d'une FPGA.

Le synthétiseur des outils CAO génère dans un premier temps une Netlist qui décrit la connectivité de l'architecture. Puis l'outil de placement-routage place de façon optimale tous les composants et effectue le routage entre les différentes cellules logiques. Ces deux étapes permettent de générer un fichier de configuration à télécharger dans la mémoire de configuration du FPGA. Ce fichier est appelé Bitstream et peut être directement chargé sur FPGA à partir d'un ordinateur hôte [14].

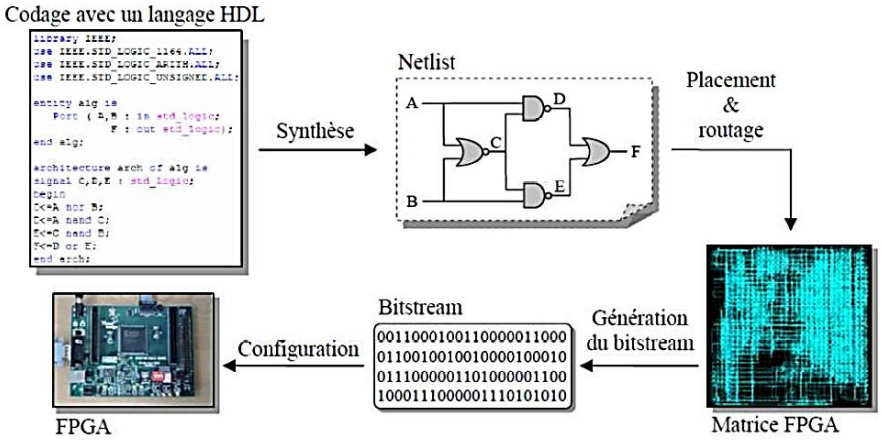


Figure 4.2 : Programmation d'un FPGA.

4.3.1 Présentation de l'outil System Generator et du flot de conception du logiciel ISE:

4.3.1.1 Le SystemGenerator :

Le Xilinx System Generator pour DSP est un plug-in pour MATLAB(Simulink) qui permet aux concepteurs de développer des systèmes DSP de haute performance pour les FPGA Xilinx. Les concepteurs peuvent concevoir et simuler un système utilisant Xilinx sous bibliothèque de modèles MATLAB(Simulink). L'outil génère alors automatiquement un HDL (Hardware Description Language) C'est un code mappé à Xilinx sous forme d'algorithme pré-optimisé et synthétisable. Cette conception de HDL peut être synthétisée pour la mise en œuvre dans les différentes plateformes, comme par exemple: Virtex-II Pro FPGA et Spartan-IIIE FPGA. Par conséquent, les concepteurs peuvent définir une représentation abstraite d'une conception de niveau système et facilement transformer ce code source unique en une représentation de niveau de la porte. En outre, il fournit la génération automatique d'un banc d'essai de HDL, ce qui permet la vérification de la conception à la mise en œuvre.

4.3.1.2 Présentation du logiciel ISE :

Le logiciel Xilinx ISE est un logiciel de description, de simulation et de programmation de circuits et systèmes numériques sur des composant programmables. Le logiciel ISE permet :

- la description de circuits numériques sous forme de schémas logiques, de machines à état finis ou en langages de description matériel (VHDL, Verilog, ABEL) ;
- la compilation, la simulation comportementale ;

- la synthèse, le placement routage et l'implémentation ;
- la simulation temporelle et l'analyse de timing,
- la programmation sur les circuits programmables de Xilinx (CPLD et FPGA).

La figure 4.3 représente l'interface Project Navigator de l'ISE 12.3 permettant l'accès à toutes les ressources d'un projet ainsi qu'aux outils de l'implémentation.

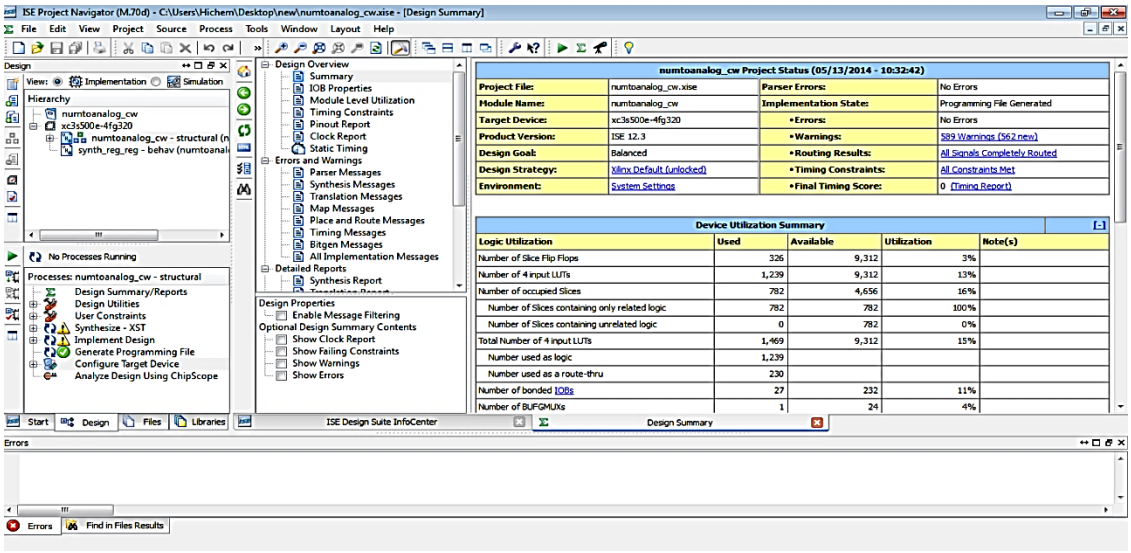


Figure 4.3 : L'interface Project Navigator de l'ISE 12.3

4.4 Réalisation expérimentale de l'implémentation :

Pour pouvoir réaliser l'implémentation expérimentale de la transmission chaotique (émetteur et récepteur) il nous faut :

- Un convertisseur analogique-numérique (CAN)8 bits pour l'insertion du message à transmettre au niveau de l'émetteur.

  La plate-forme de développement FPGA Spartan3^E pour l'implémentation de la transmission chaotique.
- Deux convertisseurs numériques analogiques(CNA) de 12 bits pour la visualisation des signaux : message informatif m(t), signaux chaotiques $x_1(t)$ et $x_2(t)$, signal crypté s(t), signal informatif décrypté r(t)sur l'oscilloscope.

La figure 4.4 représente l'implémentation sur carte FPGA du signal informatif audio et sa récupération avec haut-parleur.

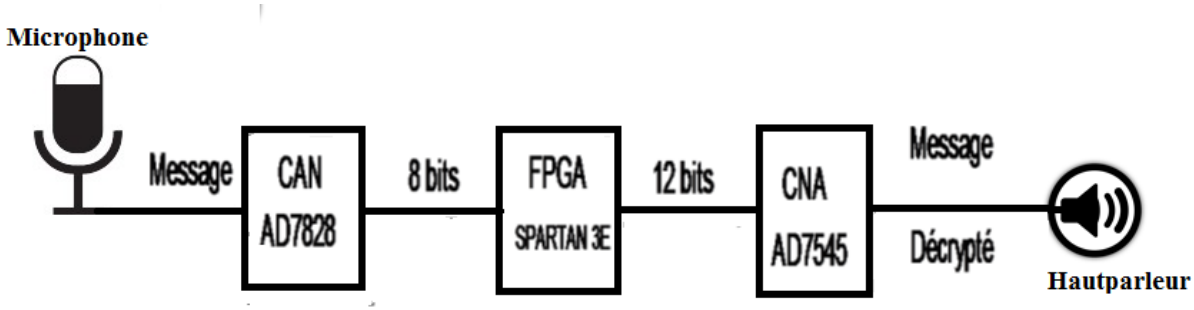


Figure 4.4 : Architecture de l'implémentation de la transmission chaotique.

La photo de la figure 4.5 représente l'environnement de la réalisation expérimentale.

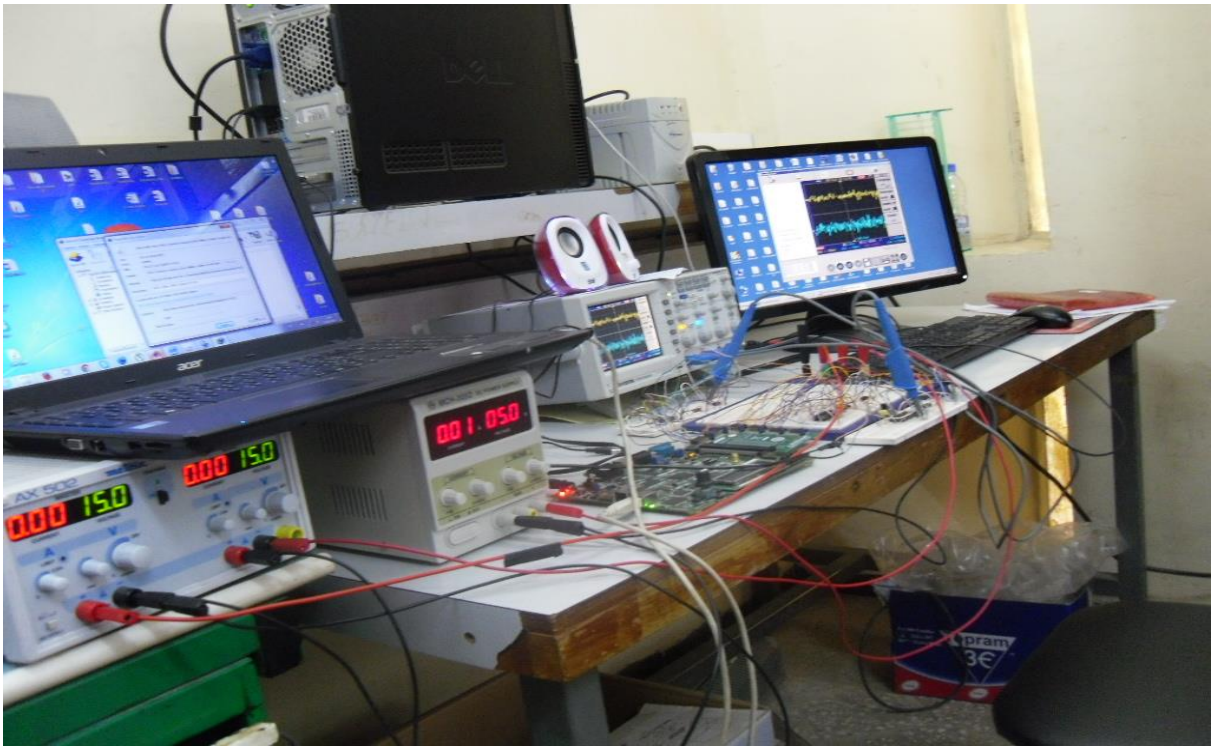


Figure 4.5 : Réalisation expérimentale de l'implémentation.

4.4.1 Convertisseur analogique-numérique :

Pour pouvoir insérer notre message informatif m(t) que l'on souhaite crypter à travers la transmission chaotique dans la carte FPGA, il faut le convertir en un signal numérique grâce au convertisseur analogique-numérique. Ce dernier est basé sur l'utilisation d'un convertisseur AD7828 de résolution 8 bits (figure 4.6) [15].

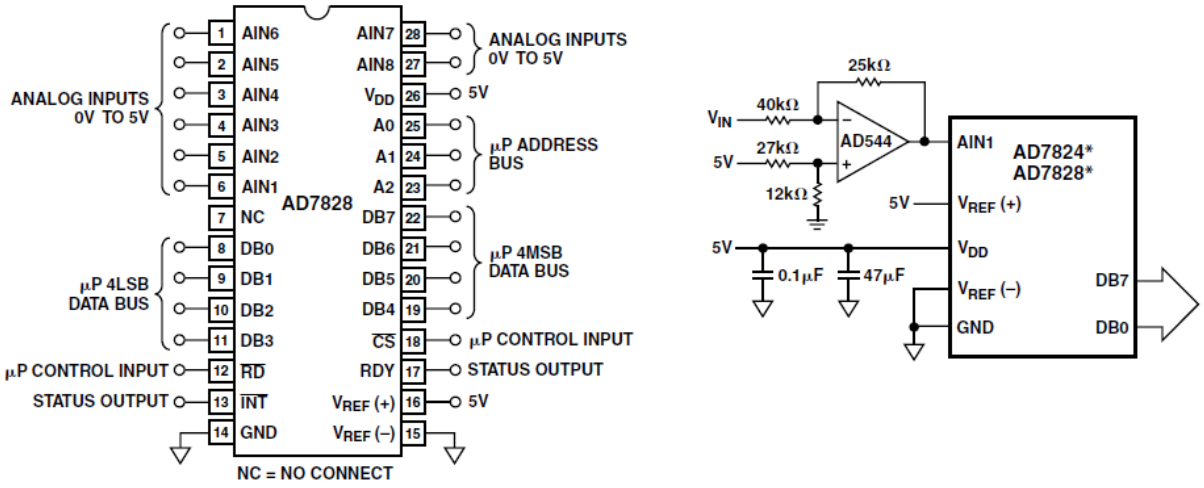


Figure 4.6 : Conversion A/N pour l'acquisition du message.

4.4.2 Plate-forme de développement Spartan3^E:

La Carte de développement Spartan 3E fournit une plateforme de développement autonome puissante et très perfectionnée pour les conceptions ciblant le FPGA Spartan 3E de Xilinx. Elle dispose d'un circuit FPGA Spartan 3^E de 500.000 portes logiques capable de mettre en oeuvre de grands systèmes numériques complexes dont le processeur softcore RISC MicroBlaze à 32 bits avec interfaces DDR. La carte propose la programmation JTAG via le port USB2 intégré avec des câbles USB ou de type parallèle externes. Le FPGA prend en charge la configuration via la mémoire flash intégrée de la plate-forme Xilinx, une mémoire Intel StrataFlash, une mémoire flash série ST Microelectronics et de nombreuses autres options. Il est totalement compatible avec toutes les versions des outils Xilinx ISE, dont le kit Web gratuit [16].

4.4.2.1 Connecteurs de la carte FPGA Spartan 3^E :

Les connections de la carte Spartane3E sont composées de: Connecteur FX2 Hirose à 100 broches ; trois connecteurs Pmod à 6 broches ;VGA ;Clavier PS/2 ; deux connecteurs RS ;Port Ethernet RJ-45 ; embase à 16 broches pour modules LCD optionnels ; connecteur SMA pour entrée d'horloge à haute vitesse.

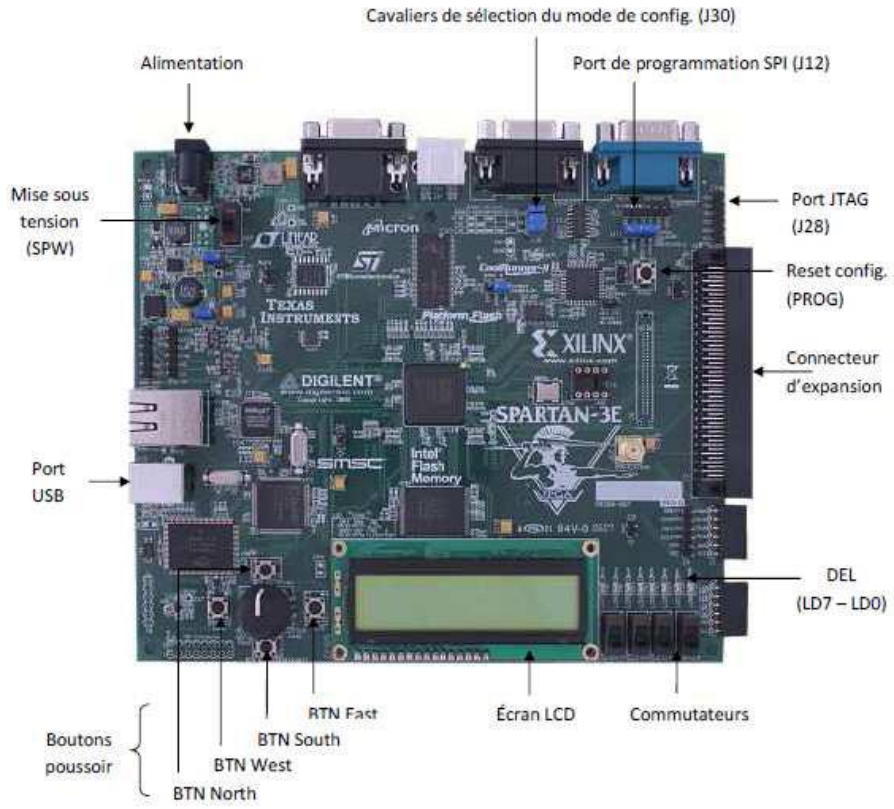


Figure 4.7 : Plateforme de développement Spartan3E.

4.4.2.2 Implémentation sur la carte FPGA Spartan 3E :

Les figures 4.8 à 4.12 représentent le processus d'implémentation du Programme HDL générer avec MATLAB(System Generator) sur la carte FPGA Spartan 3E avec le logiciel ISE.

Après la réalisation expérimentale dans MATLAB(système generator)nous allons générer notre Programme HDL à partir du jeton 

. En cliquant sur ce jeton on obtient la fenêtre suivante :

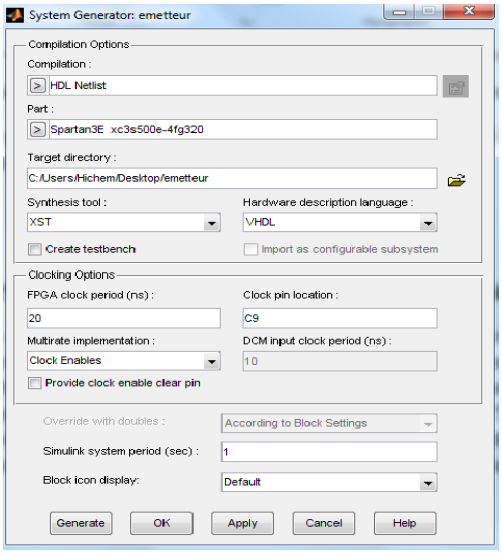


Figure 4.8 : La fenêtre du System Generator

On règle quelques paramètres(le dossier de destination, clock pin location, le langage HDL…) puis on clique sur « GENERATE » le programme va générer le code VHDL de notre Projet.

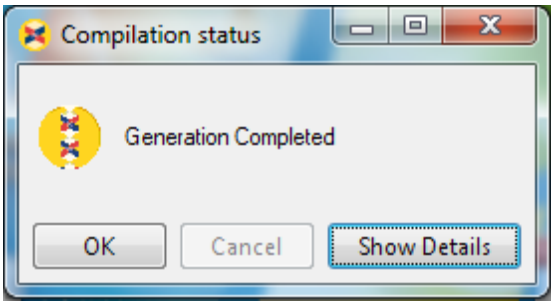


Figure 4.9 : La fenêtre de la génération du système avec succès

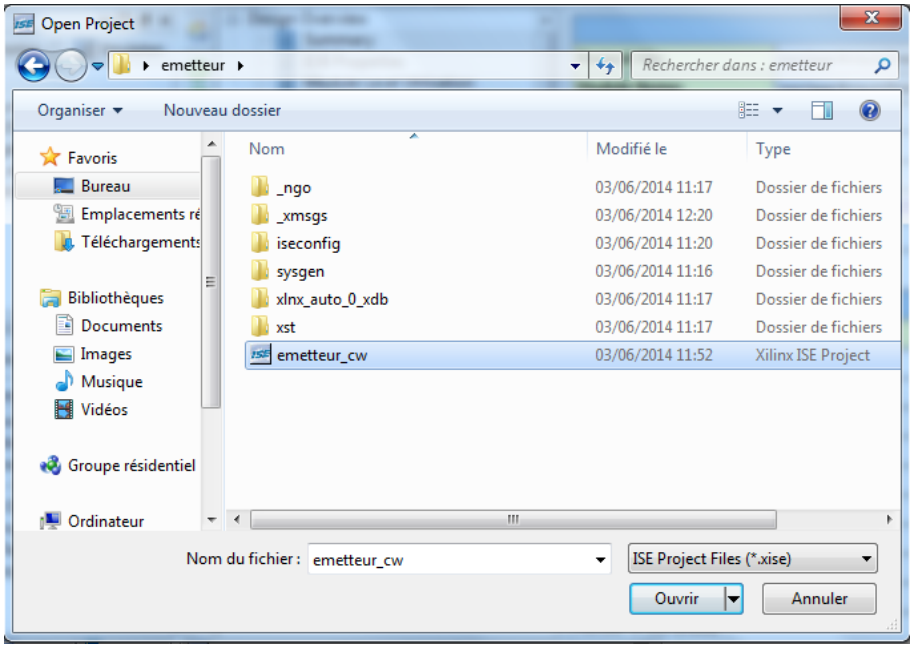


Figure 4.10 : La fenêtre du open project

Apres la génération du programme VHDL, A partir du logiciel ISE, une interface représentée sur la figure 4.10. On clique sur « File » puis « open Project » pour ouvrir le programme HDL qu'on a généré.

On ouvre le fichier qui a l'extension « cw ». Ensuite on effectue une opération de synthèse pour vérifier le bon fonctionnement du montage.

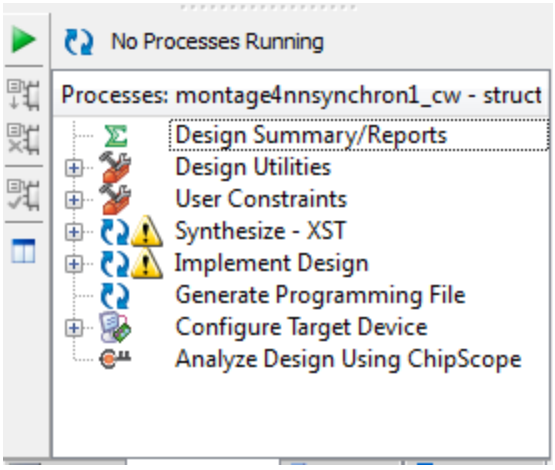


Figure 4.11 : La fenêtre pour la synthèse

Apres avoir synthétisé notre projet on va implémenter ce dernier dans la carte FPGA Spartan 3^E. Pour cela nous allons connecter cette dernière par câble USB. Ensuite on

clique sur « configure Target Device ». Une fenêtre s'ouvre permettant d'implémenter notre projet dans la carte FPGA Spartan 3^E : on clique sur bouton droit « initialize chain » puis sur « program» (figure 4.12).

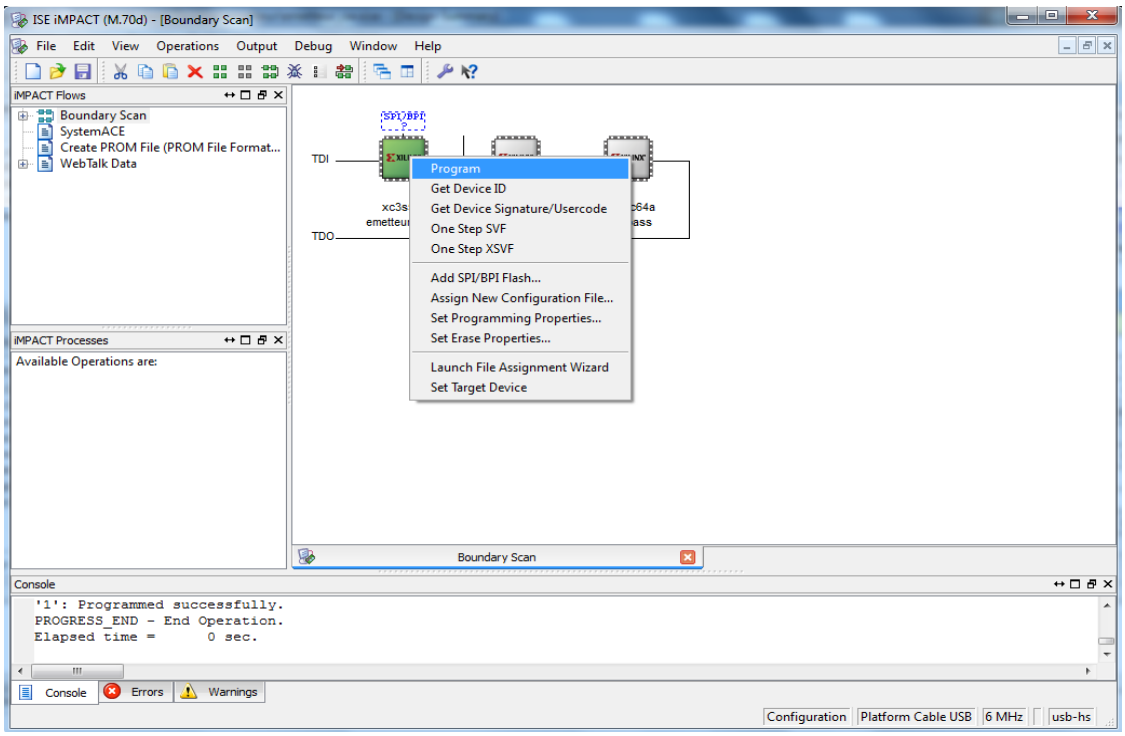


Figure 4.12 : Interface permettant la programmation du FPGA

4.4.3 Convertisseurs numérique analogique :

Les deux convertisseurs numériques analogiques utilisés dans notre projet sont constitués du convertisseur AD7545 de résolution 12 bits . La conversion se fera comme suit : les données à convertir sont recueillies à partir de la carte FPGA. Le courant de sortie out A est proportionnel au code binaire des 12 bits à l'entrée du convertisseur AD7545. Il est donc possible d'avoir une tension analogique Vanalog image des 12 bits d'entrée du CNA. La figure 4.13 présente le montage réalisé pour une conversion numérique analogique [17].

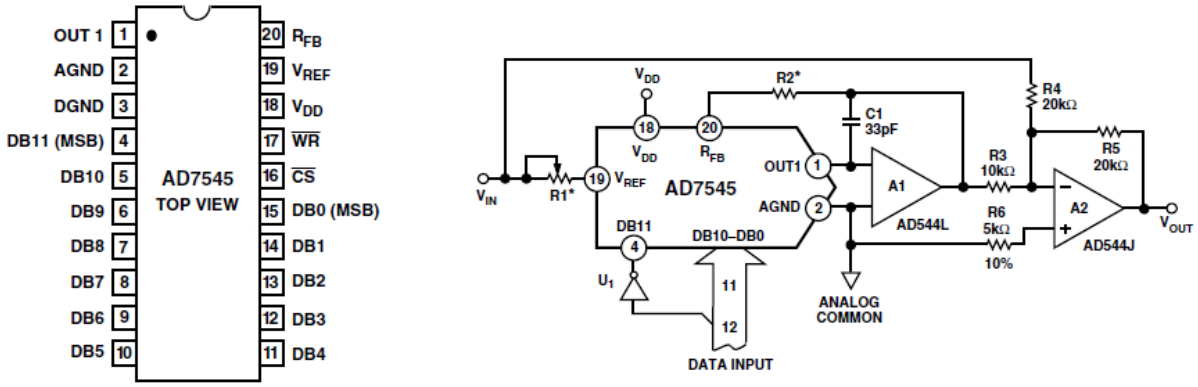


Figure 4.13 : Montage de conversion numérique analogique

4.4 Implémentation de l'émetteur sur circuit FPGA :

Nous rappelons le système normalisé de l'oscillateur chaotique de Sprott obtenu dans le chapitre 2 :

$$\dot{x} = -Q_1 x + Q_2 z \tag{2.6}$$

$$\dot{y} = Q_3 x - Q_4 z + Q_5 sign(x) \tag{2.7}$$

$$\dot{z} = Q_6 y - Q_7 z \tag{2.8}$$

Certaines fonctions n'étant pas disponibles dans la bibliothèque de System Generator, nous les avons synthétisées à l'aide des blocs disponibles tel que :

- le bloc intégrateur (figure 4.14a).
- le bloc réduction de résolution et inversion du bit le plus significatif pour la carte de conversion numérique analogique (figure 4.14b).
- la fonction sign a l'aide du bloc Mcode (figure4.14c).

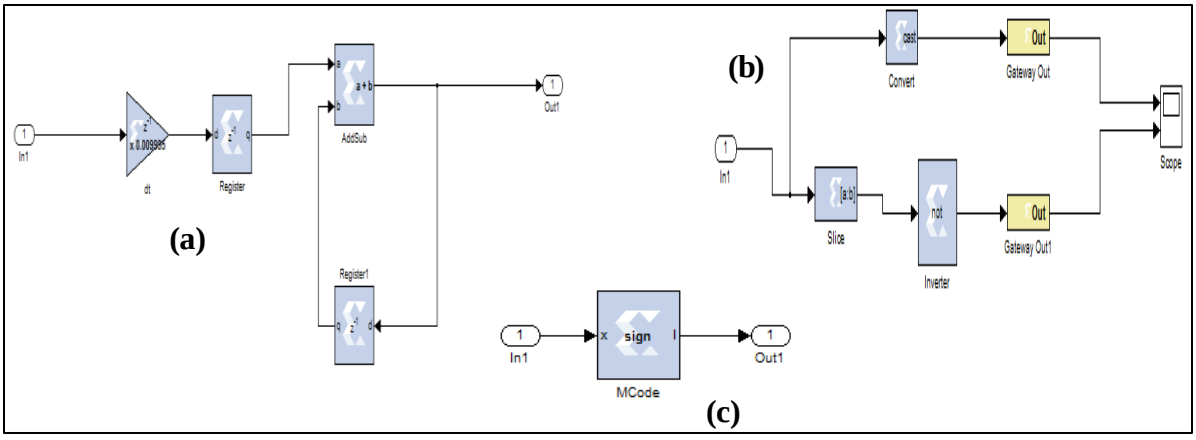


Figure 4.14 : Fonctions spécifiques synthétisées comprenant l'intégrateur (a) , la conversion du format (b) et la fonction sign (c)

La figure 4.15 représente l'implémentation de l'émetteur chaotique incluant l'oscillateur chaotique de Sprott et l'insertion du message à transmettre. Le bloc Resource « Estimator » permet de déterminer les ressources utilisées lors de l'implémentation.

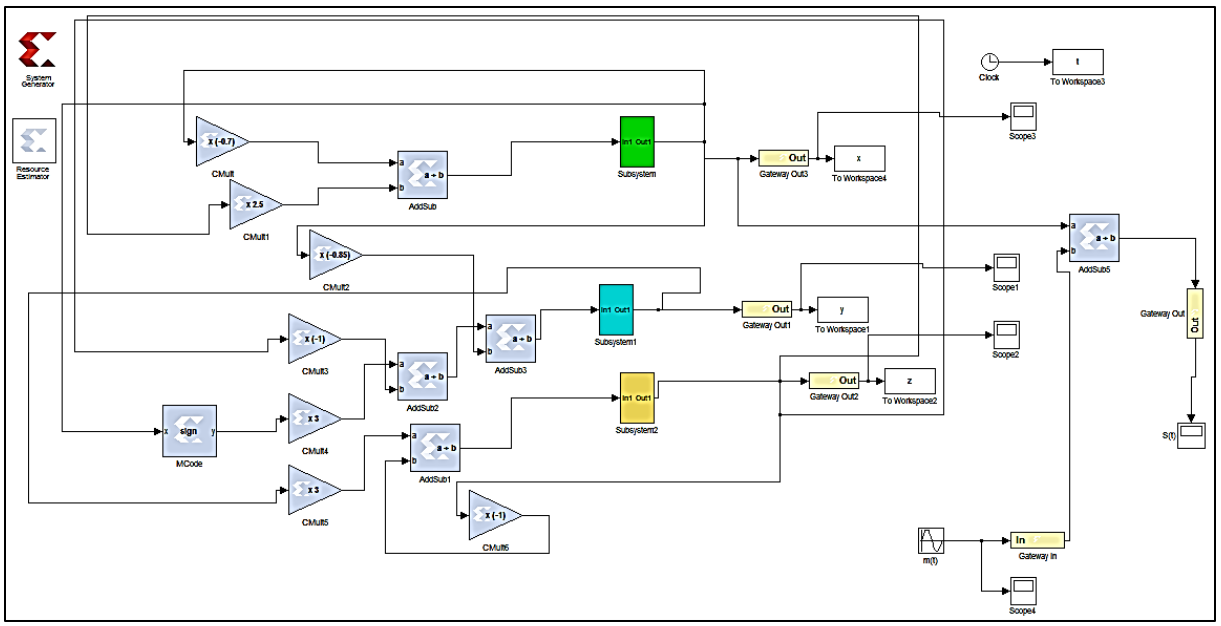
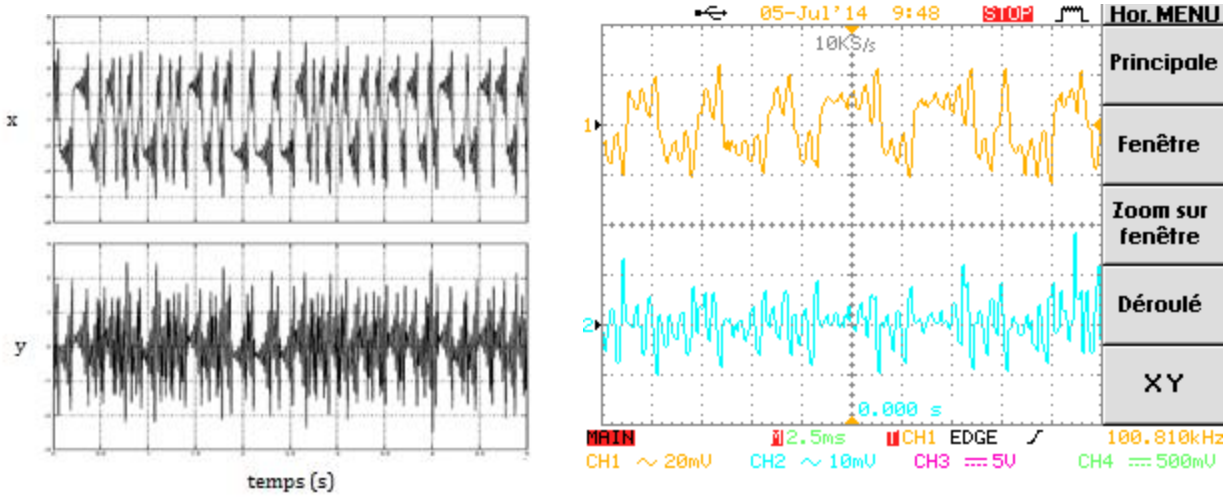


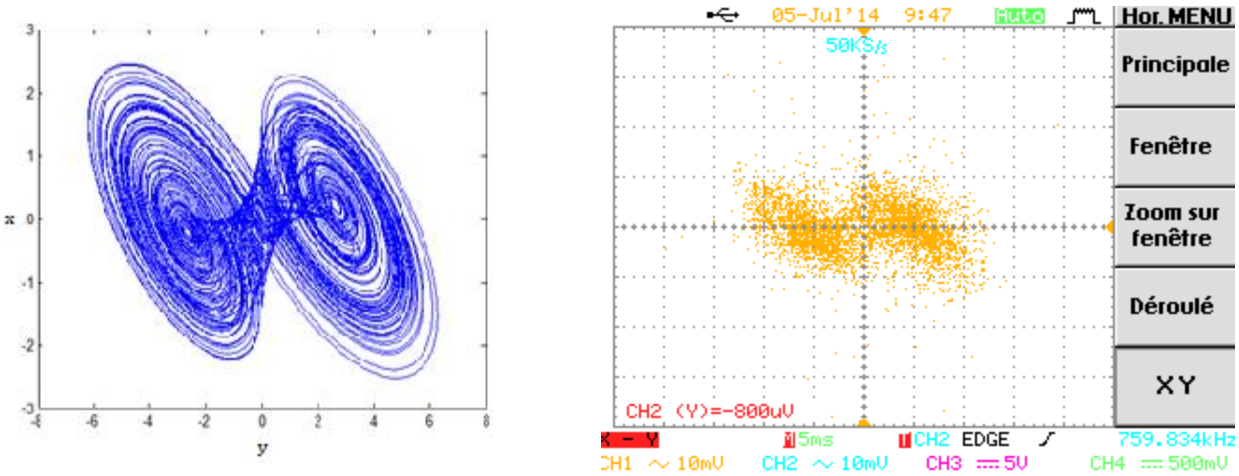
Figure 4.15 : Implémentation de l'émetteur chaotique.

Les figures 4.16, 4.17 et 4.18 représentent les différents signaux relevés sous Simulink-système Generator et sur la carte expérimentale à l'aide de l'oscilloscope numérique (GWINSTEK) :



(a) **(b)**

Figure 4.16 : Signaux x (t) et y(t) simulés(a) et expérimentaux (b).



(a) **(b)**

Figure 4.17 : L'attracteur étrange simulé (a) et expérimental (b)

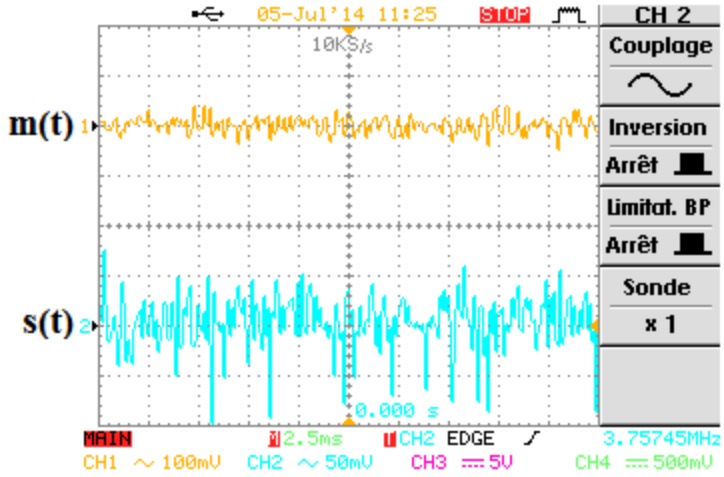


Figure 4.18 : Les signaux informatif m(t) et crypté s(t)

Le signal informatif utilisé m(t) est un signal audio obtenu à partir d'un microphone.

4.5 Implémentation de la transmission chaotique sur FPGA :

Le système de transmission chaotique constitué de l'émetteur chaotique (oscillateur de Sprott incluant le message) et du récepteur chaotique (avec la synchronisation en boucle fermée permettant la récupération du message) a été implanté sur la carte FPGA Spartan 3^E associée à une carte de conversion analogique-numérique et numérique-analogique. La figure 4.19 représente l'implémentation de l'émetteur et du récepteur sur FPGA

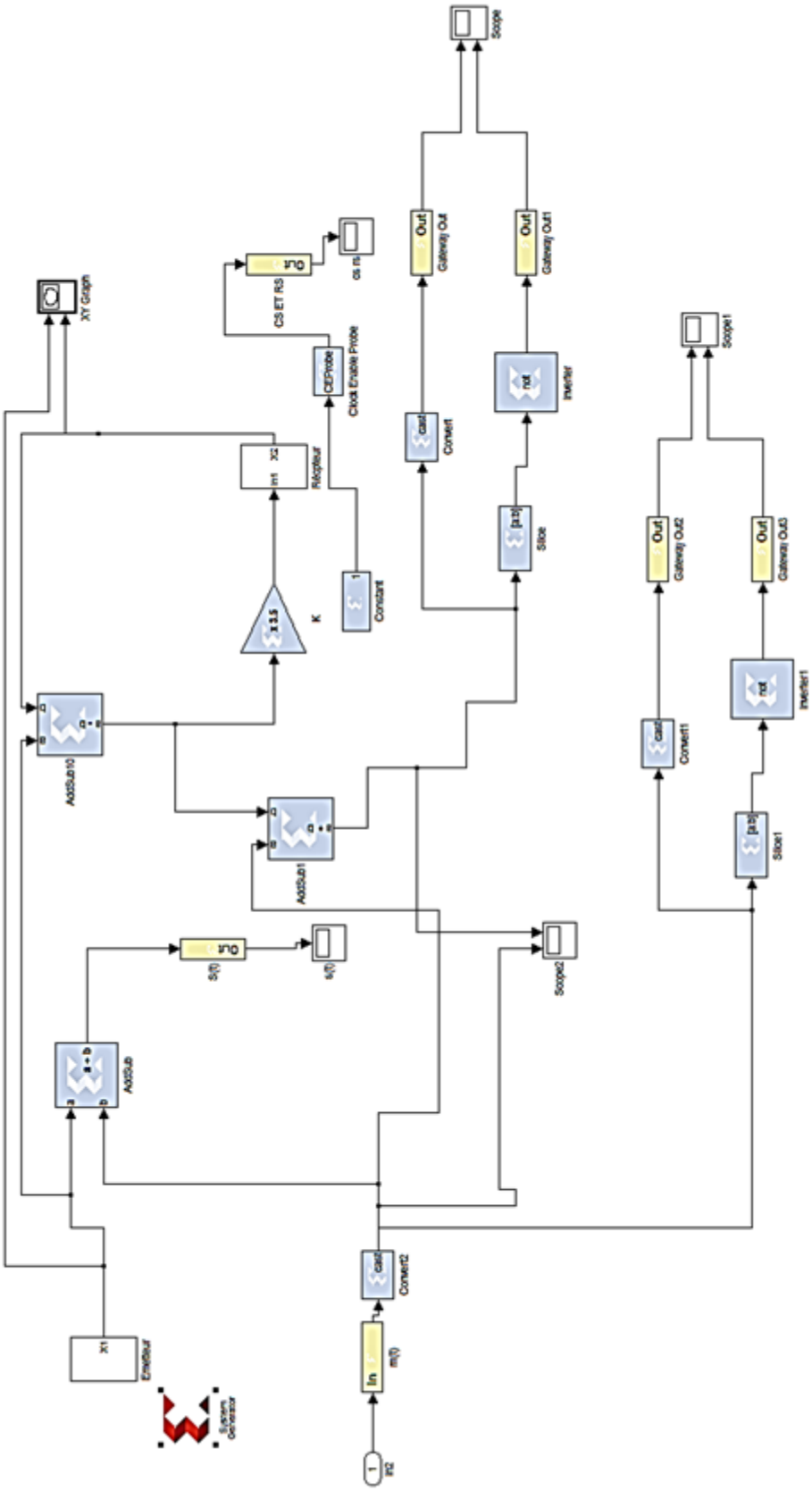
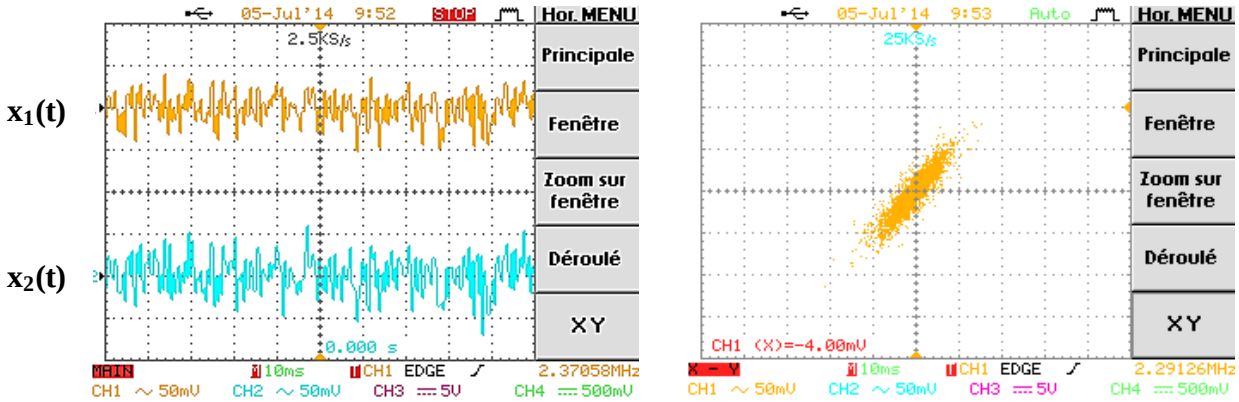


Figure 4.19 : Implémentation de l'émetteur et du récepteur sur carte FPGA

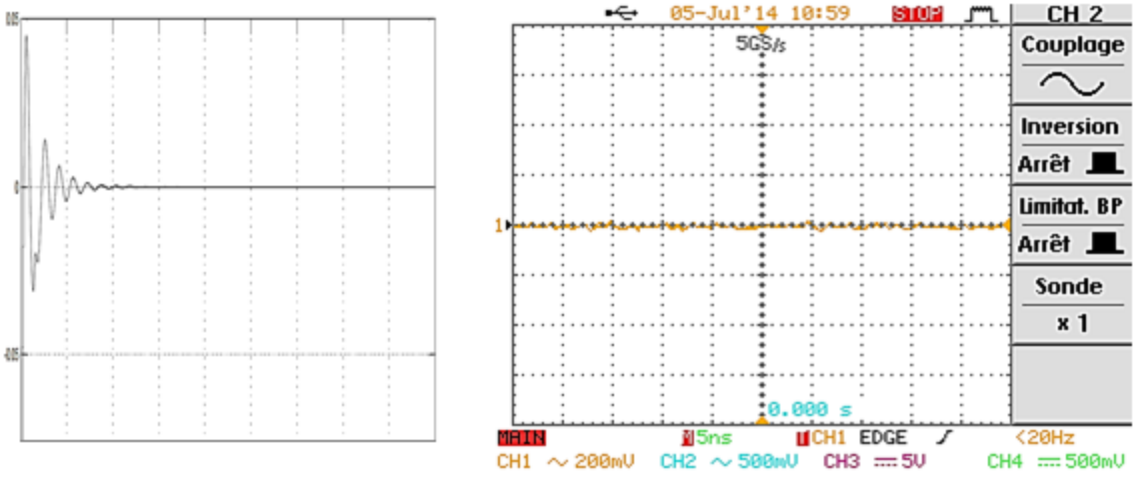
Les figures 4.20 à 4.22 montrent les différents oscillogrammes relevés au niveau de la carte FPGA associée avec les deux cartes de conversion CNA. La visualisation de ces différents signaux nous permet de:

- vérifier la synchronisation de l'émetteur et du récepteur (figures 4.20).
- mesurer l'erreur entre les signaux estimés au niveau du récepteur et de l'émetteur (figures 4.21).
- montrer la récupération du message associé à une légère déformation (figure 4.22), nécessitant l'utilisation d'un filtrage numérique que l'on pourra implémenter dans la carte.



(a) (b)

Figure 4.20 : Synchronisation des signaux $x_1(t)$ et $x_2(t)$ (a) et représentation du signal x_2 en fonction de x_1(b).



(a) (b)

Figure 4.21 : Erreur de synchronisation e(t)= $x_1(t)$- $x_2(t)$ simulé (a) et expérimental (b)

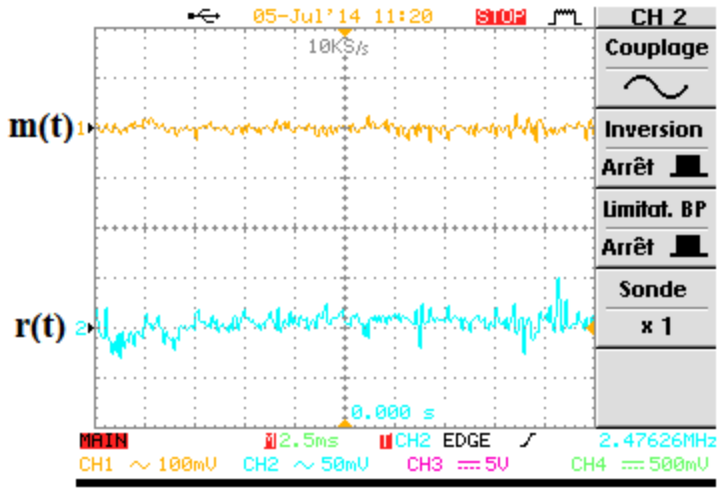
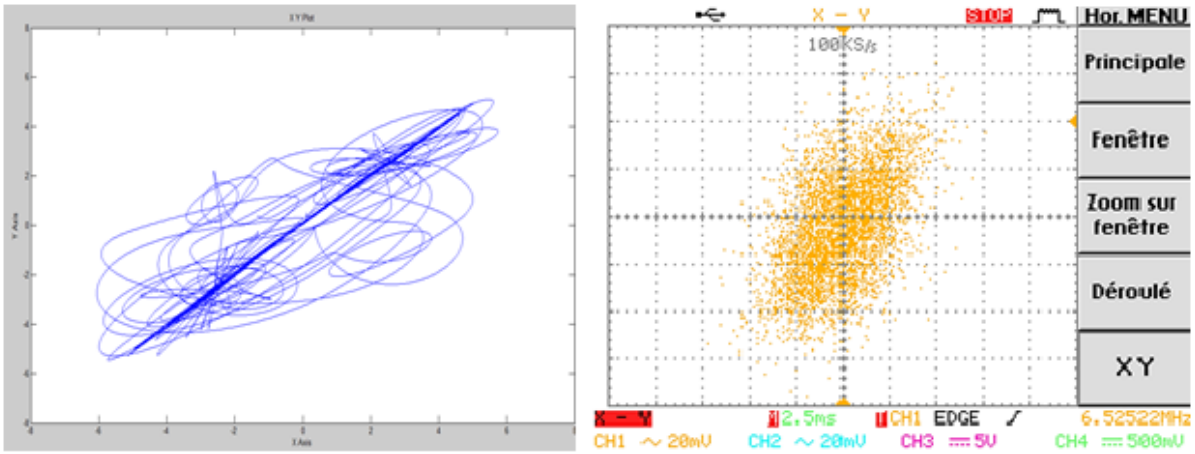


Figure 4.22 : Les signaux informatif m(t) et récupéré r(t) avec synchronisation.

Lorsque l'émetteur et le récepteur ne sont pas synchronisés, la caractéristique $x_1(t)$ en fonction de $x_2(t)$ n'est plus une droite mais une courbe aléatoire. La figure 4.23 représente par exemple la caractéristique en prenant K<3.5 en simulation (figure 4.23a) et en expérimentation (figure 4.23b).



(a) (b)

Figure 4.23 : Désynchronisation entre l'émetteur et le récepteur en simulation (a) et expérimentation (b).

L'environnement ISE fournit un rapport d'implémentation sous forme de tableaux contenant les informations utiles liées au design. La figure 4.24 représente les

ressources utilisées au niveau de la carte lors de l'implémentation de la transmission chaotique.

montage4_cw Project Status (06/03/2014 - 12:05:34)			
Project File:	montage4_cw.xise	**Parser Errors:**	No Errors
Module Name:	montage4_cw	**Implementation State:**	Programming File Generated
Target Device:	xc3s500e-4fg320	**•Errors:**	No Errors
Product Version:	ISE 12.3	**•Warnings:**	852 Warnings (816 new)
Design Goal:	Balanced	**•Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**•Timing Constraints:**	X 1 Failing Constraint
Environment:	System Settings	**•Final Timing Score:**	4212534 (Timing Report)

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	501	9,312	5%	
Number of 4 input LUTs	2,632	9,312	28%	
Number of occupied Slices	1,831	4,656	39%	
Number of Slices containing only related logic	1,831	1,831	100%	
Number of Slices containing unrelated logic	0	1,831	0%	
Total Number of 4 input LUTs	3,141	9,312	33%	
Number used as logic	2,632			
Number used as a route-thru	509			
Number of bonded IOBs	69	232	29%	
Number of BUFGMUXs	1	24	4%	

Figure 4.24 : Ressources consommées par l'implémentation.

La figure 4.25 est un aperçu du circuit implémenté sur la carte Spartan3^{E} avec les routages et l'emplacement des ressources utilisées.

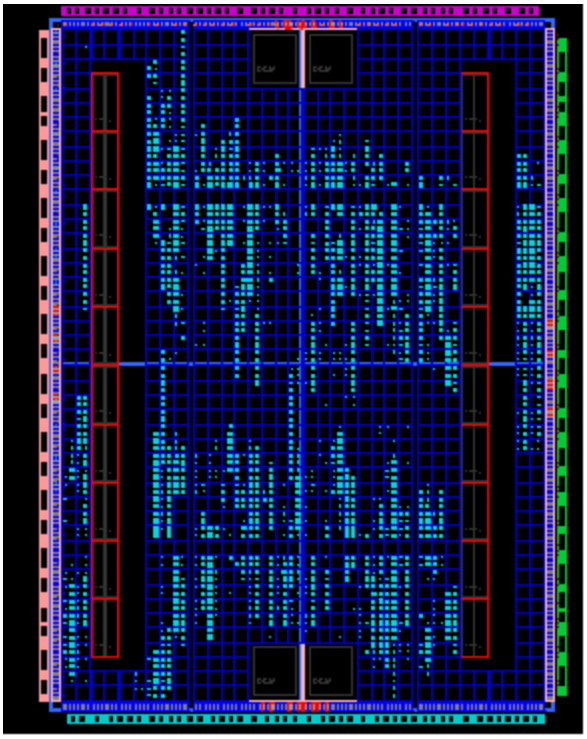


Figure 4.25 : Aperçu du circuit implémenté sur le FPGA Spartan3^{E}.

4.6 Conclusion :

L'objectif de ce chapitre a été l'implantation sur cible FPGA d'une transmission chaotique. L'émetteur constitué de l'oscillateur chaotique de Sprott incluant notre message informatif que nous voulons crypter a été implémenté sur la carte FPGA Spartan3^{E} et une concordance entre les signaux obtenus par simulation et les signaux relevés au niveau de l'oscillateur numérique a été observée. L'implémentation du récepteur a mis en évidence le fonctionnement expérimental avec la synchronisation par boucle fermée permettant la récupération du message crypté associé à une légère déformation qui pourra être minimisée par un filtrage adéquat. Enfin, une estimation des ressources utilisées par les différentes implémentations a été faite.

CONCLUSION GÉNÉRALE

Dans ce mémoire, nous avons étudié un système de communication chaotique construit à l'aide de l'oscillateur de Sprott et basé sur la synchronisation par boucle fermée pour la récupération du message et son implémentation sur carte FPGA Spartan 3 E.

Dans le premier chapitre nous avons évoqué les caractéristiques des systèmes dynamiques chaotiques ainsi que les différentes évolutions possibles de transition vers le chaos (bifurcation) et leur caractéristique grâce au calcul des exposants de Lyapunov. Dans le deuxième chapitre, nous avons étudié l'oscillateur chaotique de Sprott, puis nous avons passé en revue quelques méthodes d'insertion du message que l'on veut crypter. Parmi ces dernières, nous avons choisi la méthode par addition. A l'aide de MATLAB (Simulink) nous avons simulé l'ensemble de notre émetteur chaotique composé de deux blocks : un block pour l'oscillateur de Sprott et le deuxième pour l'insertion du message. Dans le chapitre trois, nous avons démontré l'importance de la synchronisation chaotique même si cette dernière et complexe à réaliser. Les conditions de synchronisation ont été vérifiées par simulation sous MATLAB (Simulink). Le quatrième chapitre a comporté une implémentation de l'ensemble émetteur chaotique – récepteur chaotique sur circuit FPGA. Pour pouvoir visualiser les différents signaux récupérés au niveau du récepteur et en particulier le message sur l'oscilloscope numérique, on a réalisé une carte de conversion constituée de deux convertisseurs (CNA) et d'un convertisseur(CAN) pour l'insertion du message.

Finalement, l'objectif principal de notre travail a été l'implémentation de la transmission chaotique sur circuit FPGA et la récupération du message crypté grâce à la synchronisation par boucle fermée. Nous avons pu constater une convergence des signaux obtenus entre la simulation et la réalisation expérimentale.

Comme perspective à venir, nous suggérons la réalisation d'autres solutions de cryptage chaotique telle que le cryptage mixte pour mieux sécuriser le message informatif et l'insertion d'un filtrage numérique pour diminuer la déformation au niveau du signal décrypté.

ANNEXE

Programme de bifurcation de x en fonction du paramètre Q_3 sous MATLAB

```
functionbifurcationr;
% 3-variable SPROTT model - chaos
clc;
%%%% Number of variable and initial conditions:
nbvar=3;
xini=ones(1,nbvar);
%%%% Time parameters:
trans=100;
tend=200;
tstep=0.01;
%%%% Range (for bifurcation diagram as a function of b):
g=1 ;   % (default value for chaos)
gmin=0.5;
gmax=1.4;
gint=0.01;
grange=[gmingintgmax];
%%%% Task:
integration(xini,trans,tend,tstep,g);
bifurcation(xini,trans,tend,tstep,grange);
% Integration
function output=integration(x0,trans,tend,tstep,g);
 [t,x] = run(x0,trans,tend,tstep,g);
set(figure(1),'Position', [400 400 500 300]);
clf;
plot(t,x(:,1:3));
xlabel('Time','fontsize',18);
ylabel('x y z','fontsize',18);
xlim([0 tend]);
legend('X','Y','Z');
set(figure(2),'Position', [400 400 500 300]);
clf;
plot3(x(:,1),x(:,2),x(:,3));
xlabel('X','fontsize',14);
ylabel('Y','fontsize',14);
zlabel('Z','fontsize',14);
figure(4)
plot(t,x(:,1));
xlabel('Time','fontsize',14);
ylabel('X','fontsize',14);
figure(5)
```

```
plot(t,x(:,2));
xlabel('Time','fontsize',14);
ylabel('Y','fontsize',14);
figure(6)
plot(t,x(:,3));
xlabel('Time','fontsize',14);
ylabel('Z','fontsize',14);
box on;
% Bifurcation
function output=bifurcation(x0,trans,tend,tstep,range);
D=[];  % data (bifurcation diagram)
for g=range(1):range(2):range(3)
fprintf('b=%g...\n',g);
   [t,x] = run(x0,trans,tend,tstep,g);
fori=2:length(x(:,1))-1
if((x(i,1)>x(i-1,1))&&(x(i,1)>x(i+1,1)))
         D=[D; g x(i,1)];
end
end
end
figure(3)
plot(D(:,1),D(:,2),'ro','MarkerEdgeColor','b','MarkerFaceColor','g','MarkerSize',1.5)
xlabel('Q3','fontsize',14);
ylabel('X','fontsize',14);
% Run
function [t,x]=run(x0,trans,tend,tstep,g)
ttrans = [0:tstep:trans];
tspan = [0:tstep:tend];
option = [];
%option = odeset('RelTol', 1e-5);
%option=odeset('OutputS',[1:3],'OutputF','odeplot');
if trans > 0
   [t x] = ode45(@dxdt,ttrans,x0,option,g);
   x0=x(end,:);
end
[t x] = ode45(@dxdt,tspan,x0,option,g);
% dxdt
function y = dxdt(t,x,g)
%%% parameters
%Q = 1.77;
%b = 0.82;
%k = 0.5;
%%% equations
y = [ -1*x(1)+2.5*x(3)
-g*x(1)-x(3)+3*sign(x(1))
3*x(2)-x(3)];
```

Bibliographie

[1]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.

[2] H. Dang-Vu, C.Delcarte : "Bifurcations et chaos : Introduction à la dynamique contemporaine avec des programmes en Pascal, Fortran et Mathématica", Ed. Ellipses, Paru en Septembre 2000.

[3] O. Megherbi, " Etude et réalisation d'un système sécurisé à base de systèmes chaotiques", Thèse de magister, Université Mouloud Mammeri Tizi-Ouzou, Algérie, 2013.

[4] J.Oden," Le chaos dans les systèmes dynamiques", Rapport Université Paris XI, France, 2007.

[5] S. H. Strogatz, "Nonlinear Dynamics and Chaos", Harper-Collins publishers New York, 1994.

[6] E.Goncalvès " Introduction aux systèmes dynamiques et chaos", Rapport, Institut National Polytechnique de Grenoble, France, Avril 2004.

[7] Z.Elhadj : " Étude de quelques types de systèmes chaotiques", Généralisation d'un modèle issu du modèle de Chen', Thèse de doctorat en mathématiques, Université Mentouri Constantine, Algérie, 2006.

[8] B.Nana, P.Woafo, S.Domngang: " Chaotic synchronization with experimental application to secure communications", J. Commun Nonlinear Sci. Numer Simulate, Vol. 14, pp2266-2276, 2009.

[9]L. M. Pecora, and T. L. Caroll,"Synchronization in chaotic systems", Phys. Rev. Lett. 64, p. 821–824, 1990

[10] M. L. Chikhi : 'Application des systèmes dynamiques chaotiques en transmission de données', Thèse de Magister, Université Saad Dahlab, Blida, Algérie, 2012.

[11]R.Tenny. "Symmetric and asymmetric secure communication schemes",Phd Thesis, University of California,San Diego,2003.

[12] www.xilinx.com/products/silicon-devices/fpga/.

[13] http://fr.wikipedia.org/wiki/Circuit_logique_programmable.

[14] V. A. Pedroni" Circuit dessign with VHDL", MIT Press, 2004.

[15] www.alldatasheet.com/Ad7828.

[16] www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf.

[17] www.alldatasheet.com/Ad7545