

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البلدية
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Électronique
Spécialité Instrumentation

présenté par

TERKHOUCHE NESRINE

&

HAOUA IMENE

Transmission sécurisée par synchronisation d'un système chaotique de type hyperjerk

Proposé par : Mr. Ferdjouni Abdellaziz

Année Universitaire 2019-2020

Remerciements

A l'issue du cycle de notre formation nous tenons remercier Dieu le tout puissant de nous avoir donnée courage, patience et force durant toutes ces années d'étude.

Nos remerciements les plus sincères vont à: monsieur **Ferdjouni Aziz**

Pour sa disponibilité, sa rigueur scientifique, son suivi durant tout notre travail.

Nous voudrions remercier également monsieur **Radouane Kara** maitre de conférences à l'université de tizi ouzou pour son soutien.

Nos vifs remerciements vont aux membres de jury d'avoir accepté d'évaluer ce travail de mémoire.

Nos remerciements à tous le corps enseignant de Département d'électronique de Blida qui ont participé à nos formations.

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

Du profond de mon cœur, je dédie ce travail à tous ceux qui me sont chère,

A la mémoire de mon père, puisse dieu, le tout puissant, l'avoir en sa sainte miséricorde.

A Ma chère Mère, aucun hommage ne pourrait être à la hauteur des sacrifices que vous avez consenti pour mon instruction et mon bien être. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours. Puisse dieu, le très haut vous accorder santé, bonheur et longue vie.

A ma chère sœur ryma et à mon frère abdèsslam qui m'avez toujours soutenu et encouragé durant ces années d'étude.

A toute ma famille, mes amis particulièrement : chaima et asma, à mon binôme Haoua imene et à tous mes collègues de ma promo avec qui j'ai partagé de joyeux moments.

Terkhouche nesrine

A mes chers parents et ma grande mère pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.

A mes chères sœurs Hadjer, Khadija et Hiba pour leurs encouragements permanents, et leur soutien moral. A mon cher frère, Badre Eddine pour leur appui et leur encouragement. A mon fiancé Salah Eddine, pour sa patience, son soutien et ses encouragements durant ce travail et dans la vie.

A toute ma famille, et mes amis particulièrement : Badro, Lamia, Romaiassa. A mon binôme Terkhouche Nesrine pour ses efforts. Et à tous ceux qui ont contribué de près ou de loin que ce projet soit une réussite, je vous dis merci.

IMENE HAOUA

ملخص: في هذا العمل يتم انشاء نظام اتصال آمن يعتمد على مزامنة الأنظمة الفوضوية بواسطة المراقب. يتكون هذا النظام من جهاز إرسال وهو نظام فرط التشابه من نوع hyperjerk و جهاز استقبال وهو مراقب الوضع المنزلق. يتم حقن الرسالة وتشفيرها على مستوى المرسل باستخدام تقنيتي تشفير هما تشفير التضمين وتشفير الإضافة. ثم يتم استرجاعها على مستوى المتلقي من خلال تقارب المراقب خطوة بخطوة. بالإضافة إلى ذلك ، تم إجراء محاكاة للنظام في اطاري معملي (Simulink) لضمان حسن سير النظام متبوعًا بتنفيذ جهاز الإرسال على بطاقة arduino Uno

كلمات المفاتيح: نظام فوضوي ; التزامن المراقب ; الارسال .

Résumé : Ce travail consiste à réaliser un système de communication sécurisée basé sur la synchronisation des systèmes chaotiques par observateur. Ce système est composé d'un émetteur qui est un système hyperchaotique de type hyperjerk et d'un récepteur qui est un observateur à mode glissant. Le message est injecté et crypté au niveau de l'émetteur par deux techniques de cryptage à savoir le cryptage par inclusion et le cryptage par addition. Il est récupéré ensuite au niveau du récepteur grâce à la convergence étape par étape de l'observateur. Par ailleurs, une simulation de système sous matlab(Simulink) a été réalisée afin d'assurer le bon fonctionnement du système suivie d'une implémentation de l'émetteur sur une carte arduino Uno.

Mots clés : système chaotique ; synchronisation ; observateur ; transmission.

Abstract : This work consists in realizing a secure communication system based on the synchronization of chaotic systems by observer. This system consists of a transmitter that is a hyperchaotic hyperjerk-type system and a receiver that is a sliding-mode observer. The message is injected and encrypted at the transmitter level by two encryption techniques: encryption by inclusion and encryption by addition. It is then recovered at the receiver level thanks to the step-by-step convergence of the observer. In addition, a system simulation under matlab (Simulink) was carried out to ensure the proper operation of the system followed by an implementation of the transmitter on Arduino Uno board.

Keywords : chaotic system ; synchronization ; observer ; transmission.

Table des matières

INTRODUCTION GENERALE	1
CHAPITRE 1 GENERALITE SUR LES SYSTEMES DYNAMIQUES CHAOTIQUES..	4
1.1 Introduction.....	4
1.2 Système dynamique.....	4
1.2.1 Définition d'un système dynamique	4
1.2.2 Système dynamique non linéaire	5
1.2.3 Systèmes autonomes et non autonomes.....	5
1.3 Le chaos	5
1.3.1 Définition du chaos.....	5
1.3.2 Caractéristique des systèmes chaotiques	6
1.3.3 Applications du chaos.....	7
1.3.4 Domaine d'application du chaos	7
1.4 Les outils d'étude d'un système chaotique	8
1.4.1 Le point d'équilibre	8
1.4.2 L'espace de phase	11
1.4.3 Les attracteurs.....	12
1.4.4 Exposant de Lyapunov.....	15
1.4.5 Section de Poincaré.....	17
1.4.6 Diagramme de bifurcation	18
1.5 Conclusion	19
CHAPITRE 2 SYNCHRONISATION DES SYSTEMES CHAOTIQUES	20
2.1 Introduction.....	20
2.2 Concept de synchronisation.....	21
2.3 Types de synchronisation.....	21
2.3.1 Couplage unidirectionnel	21
2.3.2 Couplage bidirectionnel	22
2.4 Les méthodes de synchronisation	22
2.4.1 Synchronisation identique.....	22
2.4.2 Synchronisation généralisé	23
2.4.3 Synchronisation projective.....	23
2.4.4 Synchronisation retardé.....	24
2.5 Synchronisation chaotique à l'aide d'un observateur	24
2.5.1 Observabilité des systèmes non linéaires	25
2.5.2 L'observateur	27

2.5.3	Observateur à mode glissant.....	28
2.6	Conclusion	32
CHAPITRE 3 INSERTION ET RECUPERATION DU MESSAGE.....		34
3.1	Introduction.....	34
3.2	Propriétés des systèmes de communication à base du chaos	35
3.2.1	Spectre à large bande.....	35
3.2.2	Comportement apériodique	35
3.3	Techniques de transmission chaotique	36
3.3.1	Le masquage additif	36
3.3.2	Modulation paramétrique.....	36
3.3.3	La commutation chaotique	37
3.3.4	Cryptage mixte	38
3.3.5	Transmission à deux voies.....	39
3.3.6	Cryptage par inclusion.....	40
3.4	Étude d'émetteur et récepteur chaotique.....	41
3.4.1	cyptage par inclusion :.....	41
3.4.2	cyptage par addition :.....	49
3.5	Conclusion	52
CHAPITRE 4 IMPLEMENTATION HARDWARE		53
4.1	Introduction.....	53
4.2	présentation de la carte Arduino Uno	53
4.3	La fiche technique de la carte arduino Uno	54
4.4	Le microcontrôleur ATmega328	55
4.5	Programmation de la carte Arduino.....	56
4.5.1	Structure d'un programme IDE	57
4.5.2	téléchargement du programme sur la carte	57
4.6	Programmation de l'émetteur	58
4.6.1	Numérisation.....	58
4.6.2	Programmes :	59
4.7	Conclusion	62
CONCLUSION GENERALE		63
BIBLIOGRAPHIE.....		65

Liste des figures

Figure 1.1 système (1.4) sous Simulink.....	10
Figure 1.2 La réponse temporelle du système.....	11
Figure 1.3: espace de phase du système	12
Figure 1.4: Les trois types d'un attracteur régulier.....	13
Figure 1.5: quelques attracteurs étranges.....	14
Figure 1.6: attracteurs étranges du système	15
Figure 1.7:L'interface de l'outil MATDS.	16
Figure 1.8: Diagramme d'exposant de Lyapunov.....	16
Figure 1.9: principe de la section de Poincaré	17
Figure 1.10 : la section de Poincaré	18
Figure 1.11: diagramme de bifurcation du système	19
Figure 2.1: Schéma de couplage unidirectionnel.....	21
Figure 2.2: schéma de couplage bidirectionnel	22
Figure 2.3: Principe de synchronisation à base d'observateurs.....	25
Figure 3.1: principe de la communication sécurisé a base du chaos.....	35
Figure 3.2: schéma représentatif de la technique de masquage additif	36
Figure 3.3: schéma représentatif de la technique de modulation paramétrique	37
Figure 3.4: schéma représentatif de la technique commutation chaotique	38
Figure 3.5: schéma représentatif de la technique Cryptage mixte.....	39
Figure 3.6: schéma représentatif de la technique transmission à deux voies.....	39
Figure 3.7: Observateurs a entrées inconnues	40
Figure 3.8 : Principe de cryptage par inversion.....	40
Figure 3.9: l'émetteur chaotique sous matlab (Simulink) avec insertion de message par inclusion	42
Figure 3.10: message crypté	42
Figure 3.11: Schéma de transmission (par inclusion)	46
Figure 3.12: message reconstruit.....	47
Figure 3.13: la réponse temporelle et l'erreur entre x_1, \hat{x}_1	47
Figure 3.14: la réponse temporelle et l'erreur entre x_2, \hat{x}_2	48
Figure 3.15: la réponse temporelle et l'erreur entre x_3, \hat{x}_3	48
Figure 3.16: la réponse temporelle et l'erreur entre x_4, \hat{x}_4	49
Figure 3.17: l'émetteur sous Matlab (Simulink) avec insertion de message par addition	49
Figure 3.18: Message émis, message crypté.....	50
Figure 3.19: Schéma de récupération du message	50
Figure 3.20: message reconstruit.....	51
Figure 3.21: la réponse temporelle du x_4, \hat{x}_4	51
Figure 3.22: l'erreur de synchronisation e_4	52
Figure 4.1: la carte Arduino Uno	54
Figure 4.2: les états x_1, x_2, x_3 en fonction du temps.....	61
Figure 4.3: les attracteurs du système.....	61
Figure 4.4: les espaces de phase du système.....	62

Liste des tableaux

. Tableau4.1 : caractéristiques de la carte arduino	56
---	----

Listes des acronymes et abréviations

IDE: Integrated Development Environment

USB: Universal Serial Bus

PWM: Pulse Width Modulation

RAM: Random Access Memory

SRAM: Static Random-Access Memory

EEPROM: Electrically Erasable Programmable Read Only Memory

Introduction générale

Un système de communication numérique comporte toutes les actions visant à transmettre des données d'un émetteur à un récepteur à travers un support physique qu'on appelle canal de transmission. Avec le développement des systèmes de communication et l'arrivée des réseaux de télécommunication modernes, la sécurité de transmissions de données risque d'être menacée. En effet, la protection des données est devenue une obligation et une précaution qu'il faudra prendre, en vue de réduire le risque des intrusions malveillantes et de protéger les données contre l'interception, la modification ou le vol et garantir la fiabilité, la confidentialité, l'authenticité, ainsi que l'intégrité des données transmises. La cryptographie est une technique permettant d'assurer la sécurité de transmissions des données.

Au cours de ces dernières décennies, la théorie du chaos a été utilisée dans la cryptographie grâce à ses propriétés, principalement sa forte sensibilité aux conditions initiales, son déterminisme, son aspect pseudo-aléatoire, ainsi que son comportement complexe qui offrent la possibilité de sécuriser la transmission de données. L'idée d'utiliser le chaos dans les systèmes de transmission est survenue lors des travaux de Pecora et Carroll en 1990 qui ont démontré que deux systèmes chaotiques un maître et un esclave peuvent se synchroniser sous certaines conditions. Cette synchronisation est utilisée pour pouvoir récupérer l'information. Le cryptage chaotique s'effectue au niveau de l'émetteur par de nombreuses techniques permettant d'insérer le message dans un signal chaotique. Le message crypté résultant sera ensuite transféré à travers le biais d'un canal de transmission au récepteur. Ce dernier connaît les caractéristiques du générateur de chaos, il va décrypter à son tour le signal reçu pour restituer le message.

La synchronisation par observateur est une méthode largement employée pour la récupération du message. Elle présente l'avantage de l'atténuation du bruit et des perturbations.

En 1997 Henk Nijmeijer and Iven M. Y. Mareels ont montré que la théorie d'observateur peut être utilisée dans la synchronisation des systèmes chaotique. En choisissant un système esclave sous la forme d'observateur et un système maître qui satisfait certaines conditions et de ce fait la synchronisation peut être garantie [1]. Différents types d'observateur sont proposés. Certains de ces observateurs sont destinés uniquement à la reconstruction des états de l'émetteur, et d'autres à la récupération de l'information en plus de la synchronisation des états [2].

Dans ce mémoire, nous avons proposé un système de transmission sécurisée basé sur le chaos. Le système émetteur est un système hyper chaotique de type hyperjerk. L'information émise est un signal sinusoïdal. Deux méthodes sont utilisées pour insérer ce message. La première consiste à additionner le message à l'une des variables du système. Pour la seconde méthode, le message est porté dans l'un de dynamiques de système émetteur. Le récepteur est un observateur à mode glissant, qui s'emploie à récupérer les états de système et l'information émise. Le système émetteur sera ensuite implémenté sur une carte Arduino.

Le premier chapitre a été consacré à la présentation des généralités sur les systèmes dynamiques non linéaire, les caractéristiques d'un système chaotique et à l'étude de système émetteur en utilisant différents outils d'analyse.

Le deuxième chapitre présente les différentes méthodes de synchronisation de deux systèmes chaotiques en s'intéressant beaucoup plus à la méthode choisie, synchronisation à l'aide d'un observateur, pour récupérer l'information émise.

Le troisième chapitre traite les différentes techniques de cryptage par le chaos. Puis nous avons conçu le schéma de transmission de notre système de communication basé sur les méthodes d'insertion par addition et par inclusion. La synthèse de l'observateur à mode glissant est présentée. Enfin, les résultats de simulation sont présentés sous le logiciel Matlab.

Le quatrième chapitre est destiné à l'implémentation de l'émetteur sur une carte Arduino Uno. La carte Arduino et son microcontrôleur sont présentés, avec les différentes étapes qui mènent à la programmation de la carte tant qu'émetteur chaotique.

chapitre 1 Généralité sur les systèmes

dynamiques chaotiques

1.1 Introduction

Entre la fin des années 1960 et le début des années 1980, la reconnaissance du fait que des lois dynamiques simples peuvent donner naissance à des comportements très compliqués a été souvent ressentie comme une vraie révolution concernant plusieurs disciplines en train de former une nouvelle science, la “science du chaos” [3].

On appelle chaos tout phénomène complexe, qui dépend de plusieurs paramètres et caractérisé par une extrême sensibilité aux conditions initiales. Le chaos après plusieurs années de sa découverte ne cesse de se développer, on le trouve dans divers domaines tels que la biologie, la médecine, l'économie, la communication. Le présent travail focalise sur la transmission sécurisée de l'information. Dans cette perspective ce chapitre est destiné à l'étude de systèmes dynamiques chaotiques.

1.2 Système dynamique

1.2.1 Définition d'un système dynamique

Un système dynamique est décrit par un ensemble de variables d'état. L'état d'un système dynamique est caractérisé par sa position et la vitesse de changer sa position dans un espace nommé espace d'état. Le système dynamique est donné par une loi d'évolution temporelle, qui détermine d'une façon unique l'avenir de ses états en fonction des états passés.

1.2.2 Système dynamique non linéaire

La dynamique non-linéaire a une représentation d'état, en temps continu :

$$\begin{cases} \dot{x} = f(x(t)) \\ y = h(x(t)) \end{cases} \quad (1.1)$$

En temps discret :

$$\begin{cases} x_{k+1} = f(x_k) \\ y_k = h(x_k) \end{cases} \quad (1.2)$$

Où : $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ est une fonction non linéaire désignant le champ de vecteur et $h : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction éventuellement non linéaire désignant le vecteur de

sortie. $x(t) = [x^{(1)}(t) \dots x^{(n)}(t)]^T$ Est un vecteur d'état en temps continu.

Et $x_k = [x_k^{(1)} \dots x_k^{(n)}]^T$ est le vecteur d'état en temps discret.

\mathbb{R}^n est l'espace de phase. Dans le cas où la dimension ($n=2$), l'espace de phase est dit plan de phase.

1.2.3 Systèmes autonomes et non autonomes

Soit le système dynamique suivant :

$$\dot{x}(t) = \frac{dx(t)}{dt} = f(x, t) \quad (1.3)$$

Le système est dit autonome lorsque le champ de vecteur f ne dépend pas explicitement du temps. Dans le cas contraire le système est dit non autonome.

1.3 Le chaos

1.3.1 Définition du chaos

Il n'existe pas encore de définition universelle du terme chaos. Cependant le chaos est défini généralement comme un comportement désordonné (complexe) qui présente l'activité d'un état de système dynamique déterministe non linéaire, qui est imprédictible et sensible aux conditions initiales.

1.3.2 Caractéristique des systèmes chaotiques

On appelle un système dynamique chaotique, un système qui dépend de plusieurs paramètres ainsi que son évolution dans le temps est très sensible aux conditions initiales. Il évolue dans une région bornée, et possède une infinité de trajectoires non périodiques. Ci-dessous, nous présentons quelques caractéristiques qui permettent de distinguer un système chaotique [4].

a. La non linéarité

Un système chaotique est un ensemble d'équations dynamiques non linéaires. Si le système est linéaire, il ne peut pas être chaotique.

b. Sensibilité aux conditions initiales

Un système chaotique est très sensible aux conditions initiales, tels que toute incertitude au début mène à un comportement futur totalement différent. Cette propriété a été observée pour la première fois par E. Lorenz sur son modèle météorologique. Elle est connue sous le nom populaire d'effet papillon. Pour deux conditions initiales très proches, les deux trajectoires correspondantes à ces données initiales divergent exponentiellement [5].

c. Déterminisme

La notion de déterminisme signifie la capacité de prédire le futur d'un phénomène à partir d'un événement passé ou présent. Un système chaotique obéit à des lois rigoureuses qui décrivent son mouvement. Il est régi par des équations différentielles non linéaires connues.

d. L'imprédictibilité

Comme il est impossible de connaître les conditions initiales avec précision. La prévision à long terme l'est également, même si les lois physiques sont déterministes et exactement connues [6].

e. L'irrégularité

Ordre caché comprenant un nombre infini de modèles périodiques instables (ou mouvements).

1.3.3 Applications du chaos

- **Contrôle** : Première application du chaos est le contrôle du comportement irrégulier dans les circuits et les systèmes.
- **Synchronisation** : Communication sécurisée, cryptage, radio.
- **Traitement d'informations** : Codage, décodage et stockage d'information dans des systèmes chaotiques, tel que les éléments de mémoires et les circuits.

1.3.4 Domaine d'application du chaos

Parmi les domaines d'application du chaos, nous citons principalement les domaines suivants [7].

- **Ingénierie** :

On utilise le chaos dans contrôle de vibrations, stabilisation des circuits, turbines, réaction chimique, laser...

- **Économie** :

Dans le domaine de l'économie, c'est surtout sur la marche boursière : Prévisions économiques, analyse financière et prévision de marché.

- **Médecine et biologie** :

Le chaos se trouve dans le domaine de la médecine plus précisément en cardiologie, analyse du rythme de cœur (EEG), prédiction et contrôle d'activité irrégulière. Et en domaine de la science de la vie dans la réplication de l'ADN au niveau des cellules.

- **Communication** :

Parmi les domaines d'application du chaos en communication : chiffrement et déchiffrement de messages, compression et stockage d'images, conception et management des réseaux d'ordinateurs.

1.4 Les outils d'étude d'un système chaotique

Soit le système hyperjerk décrit par l'équation (1.4) ci-dessous :

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = -x_1 - x_2 - ax_3 - x_1^4 x_4 \end{array} \right. \quad (1.4)$$

Avec : $a = 3.6$.

Les différents outils d'études des systèmes chaotiques sont utilisés pour analyser le système (1.4).

1.4.1 Le point d'équilibre

Le point d'équilibre ou point fixe est Le point \bar{x} de l'espace de phase, il est obtenu en résolvant le système d'équations :

$$f(\bar{x}) = 0 \quad (1.5)$$

Les points d'équilibre jouent un rôle important dans l'application, ils permettent de caractériser les trajectoires voisines. Ils peuvent être stables ou instables suivant que les trajectoires voisines convergent ou divergent entre elles.

a. Étude de stabilité de point fixe

Pour déterminer le point fixe du système, on va résoudre le système d'équations suivant : $\frac{dx}{dt} = f(x) = 0$.

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 = 0 \\ \dot{x}_2 = x_3 = 0 \\ \dot{x}_3 = x_4 = 0 \\ \dot{x}_4 = -x_1 - x_2 - ax_3 - x_1^4 x_4 = 0 \end{array} \right.$$

Nous avons un seul point d'équilibre : $\bar{x} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

La stabilité des points d'équilibres est déterminée en linéarisant le système, et en discutant les valeurs propres notées λ_i , de la matrice jacobéenne résultante de la linéarisation.

La matrice jacobéenne ($m * n$) est définie par :

$$J_f(M) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

La matrice jacobéenne du système (1.4) s'écrit comme suivant :

$$J = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & -1 & -a & 0 \end{bmatrix}; \text{ Avec } a = 3.6.$$

Les valeurs propres sont données par la solution de l'équation caractéristique suivante :

$$\det(\lambda I - J) = 0 \tag{1.6}$$

$$\det(\lambda I - J) = \left(\begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & -1 & -a & 0 \end{bmatrix} \right) = 0$$

Avec :

I: est la matrice identité ($4 * 4$).

λ : est la valeur propre de la matrice.

Les valeurs propres obtenues par la résolution de l'équation caractéristique sont données ci-dessous.

$$\begin{cases} \lambda_1 = 0.1604 + 1.8395i \\ \lambda_2 = 0.1604 - 1.8395i \\ \lambda_3 = -0.1604 + 0.5172i \\ \lambda_4 = -0.1604 - 0.5172i \end{cases}$$

Les valeurs propres sont complexes conjugués. La partie réelle de deux solutions est positive donc il s'agit d'un point fixe instable.

La figure (1.1) représente la simulation sous Matlab / Simulink du système donné par l'équation (1.4).

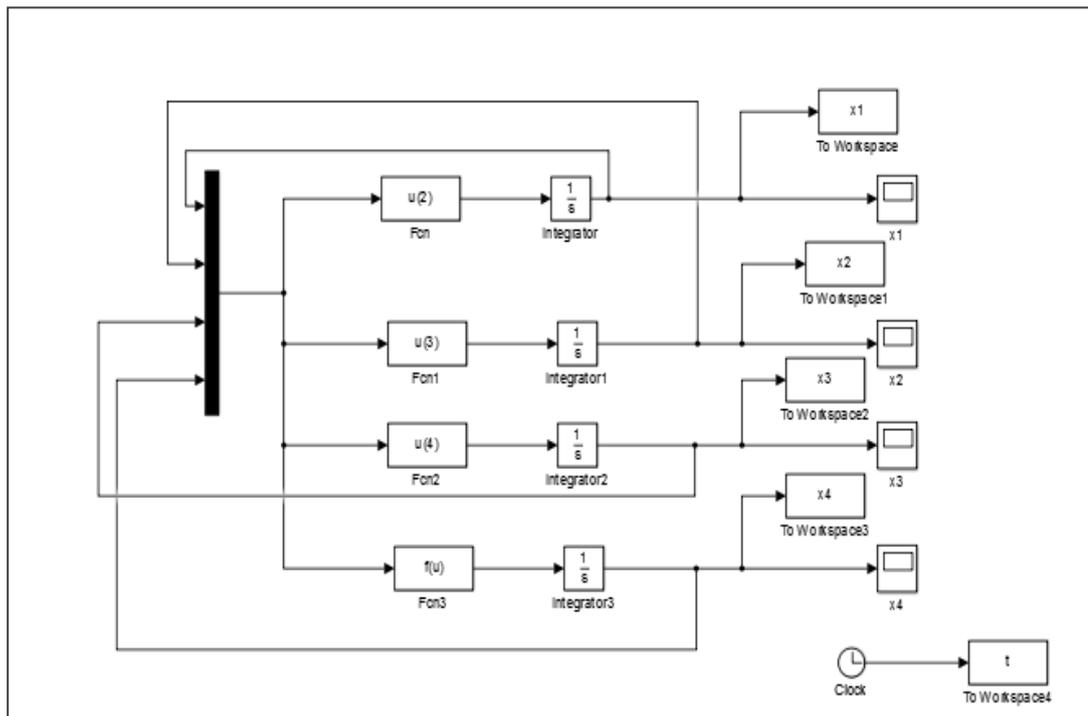


Figure 1.1.1 système (1.4) sous Simulink.

La figure (1.2) représente Les réponses temporelles de système en fonction du temps.

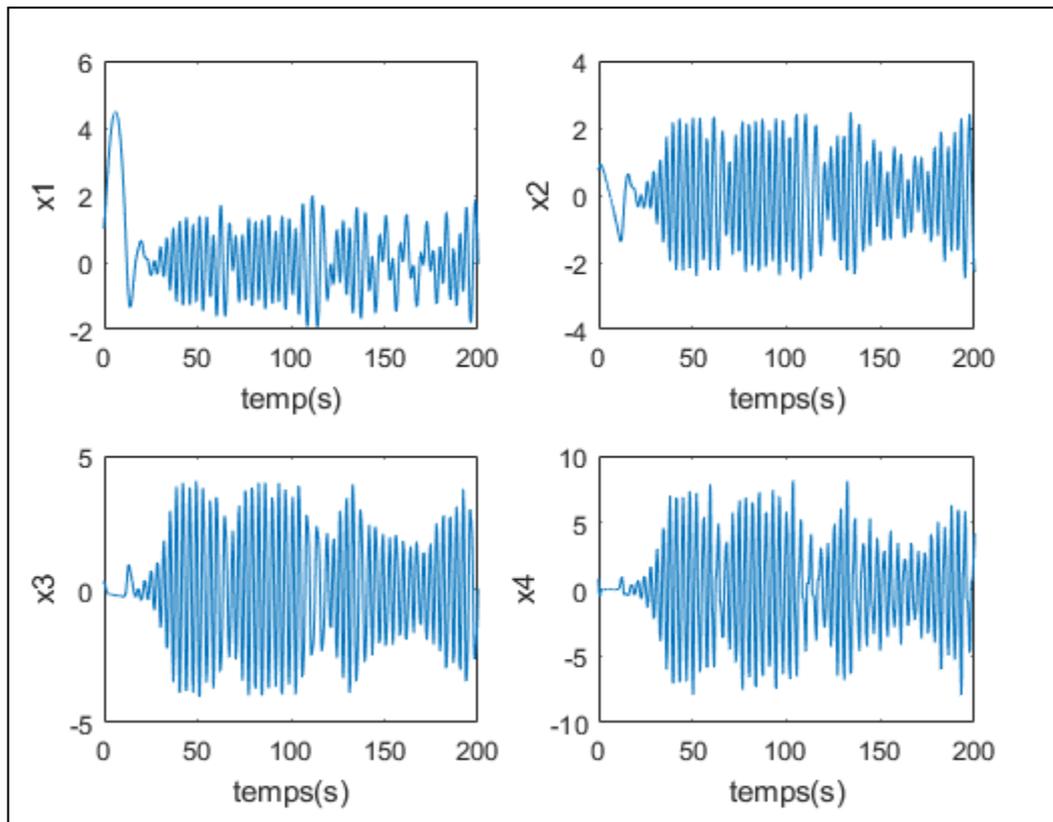


Figure 1.2 La réponse temporelle du système

1.4.2 L'espace de phase

Le comportement d'un système dynamique est caractérisé par l'évolution de ses variables d'état qui décrit complètement le système à un instant donné, à partir de diverses conditions initiales dans un espace [2]. Cet espace est appelé espace de phase qui est un système de coordonnées dont les axes sont définis par les variables d'état du système étudié. Il fait apparaître le comportement de système sous une forme géométrique [8]. Chaque point dans l'espace de phase représente un état de système et le déplacement de ce point décrit une trajectoire [6].

La figure (1.3) représente différents plans de phase du système (1.4).

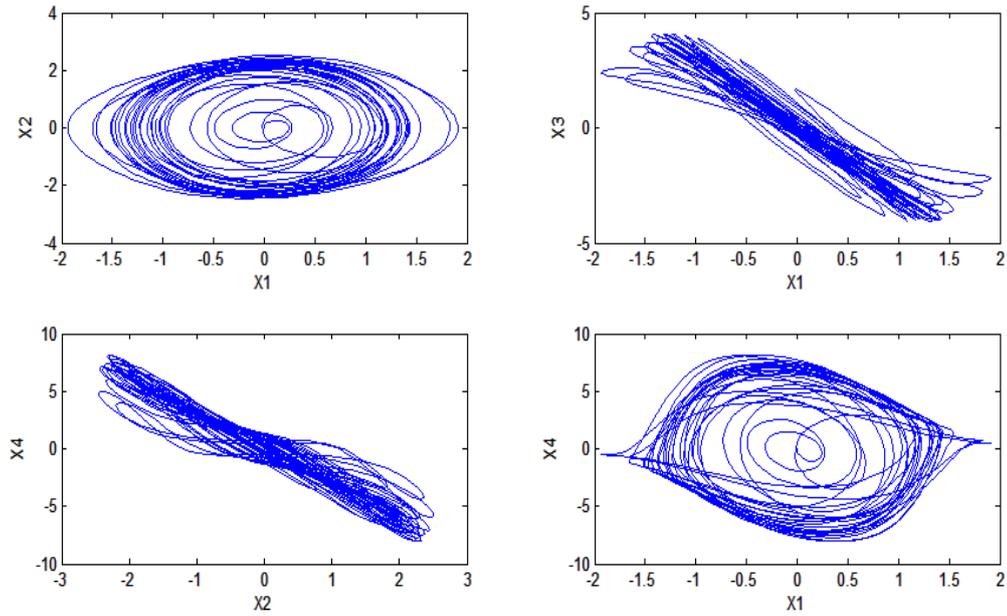


Figure 1.3: espace de phase du système

1.4.3 Les attracteurs

a. Définitions d'un attracteur/ Bassin d'attraction

Les attracteurs sont des formes géométriques qui caractérisent l'évolution à long terme des systèmes dynamiques [8] pour différentes conditions initiales. Elles représentent une zone de l'espace de phase vers lequel tend un ensemble de trajectoires des points de l'espace de phase.

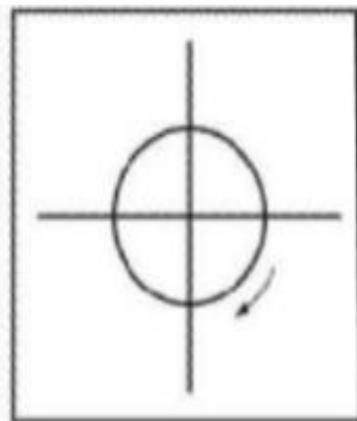
Le bassin d'attraction est la région de l'espace de phase dans laquelle l'ensemble des points contenu dans cette région sont attirés vers l'attracteur.

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

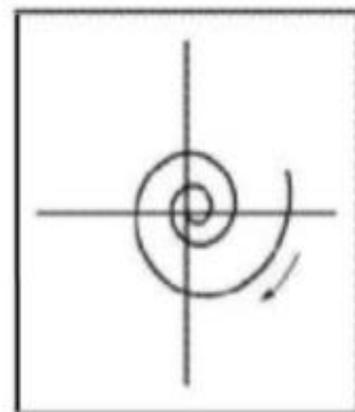
b. L'attracteur régulier

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

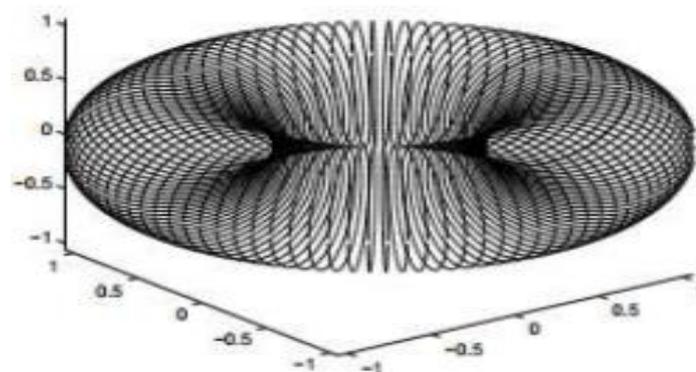
- 1) Le point fixe : C'est l'attracteur le plus simple et le plus courant, dans lequel le système évolue vers un état de repos (point). Seuls les puits peuvent être des attracteurs.
- 2) Le cycle limite périodiques : est une boucle fermée dans l'espace des phases [9]. Il ne peut arriver que si la trajectoire de phase se referme sur elle-même. L'évolution temporelle est alors cyclique. Il intervient dans la description des systèmes présentant des oscillations périodiques.
- 3) Le cycle limite pseudo périodique (Tore) : C'est presque un cas particulier du précédent. Le système présente au moins deux périodes simultanées dont le rapport est irrationnel. La trajectoire de phase ne se referme pas sur elle-même, mais s'enroule sur une variété de dimension 2.



a. Attracteur Cycle limite.



b. Attracteur point fixe.



c. Attracteur Tore.

Figure 1.4: Les trois types d'un attracteur régulier

c. L'attracteur étrange

L'attracteur étrange est une forme géométrique qui est très complexe, qui caractérise l'évolution des systèmes dynamiques chaotiques. Il a été introduit par David Ruelle et Floris Takens en 1971 pour caractériser le comportement quasi-aléatoire ou chaotique. L'attracteur étrange se caractérise essentiellement par la sensibilité aux conditions initiales tels que la distance entre deux trajectoires initialement proches augmente au fur et à mesure à une vitesse exponentielle. Chaque trajectoire de l'attracteur étrange est apériodique (c'est-à-dire qu'elle ne passe pas deux fois par le même point).

Parmi les attracteurs étranges On peut citer par exemple : l'attracteur de Rossler, l'attracteur de Lorenz, l'attracteur de Chen, l'attracteur de Chua, qui sont représentés sur la figure (1,5) [9].

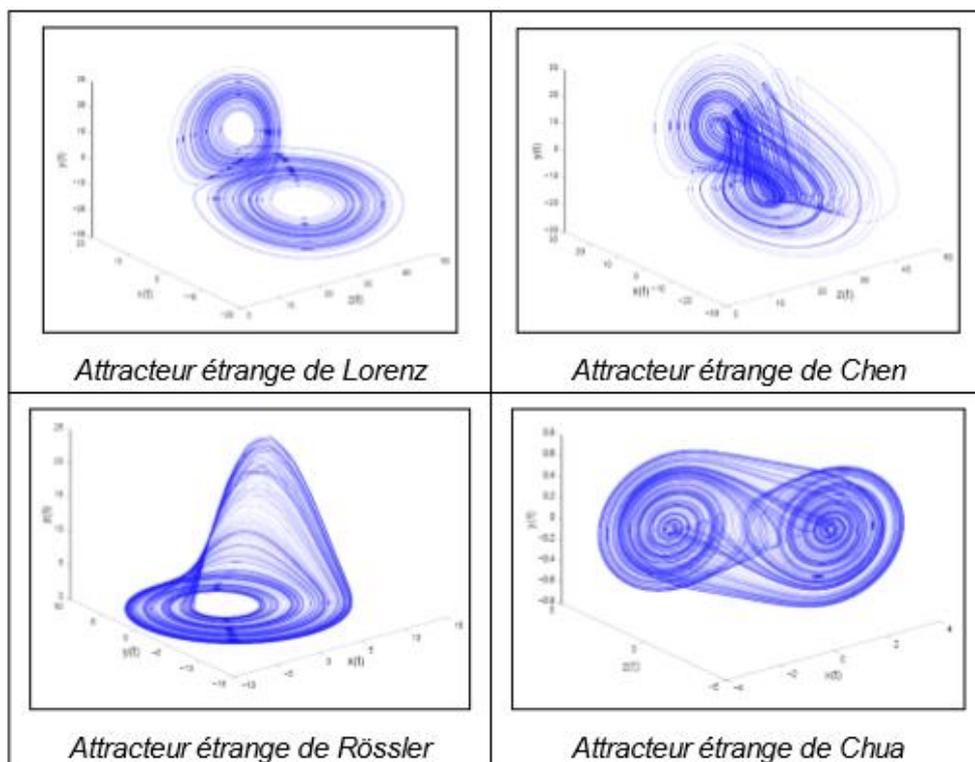


Figure 1.5: quelques attracteurs étranges

La figure (1.6) montre les attracteurs étranges du système (1.4)

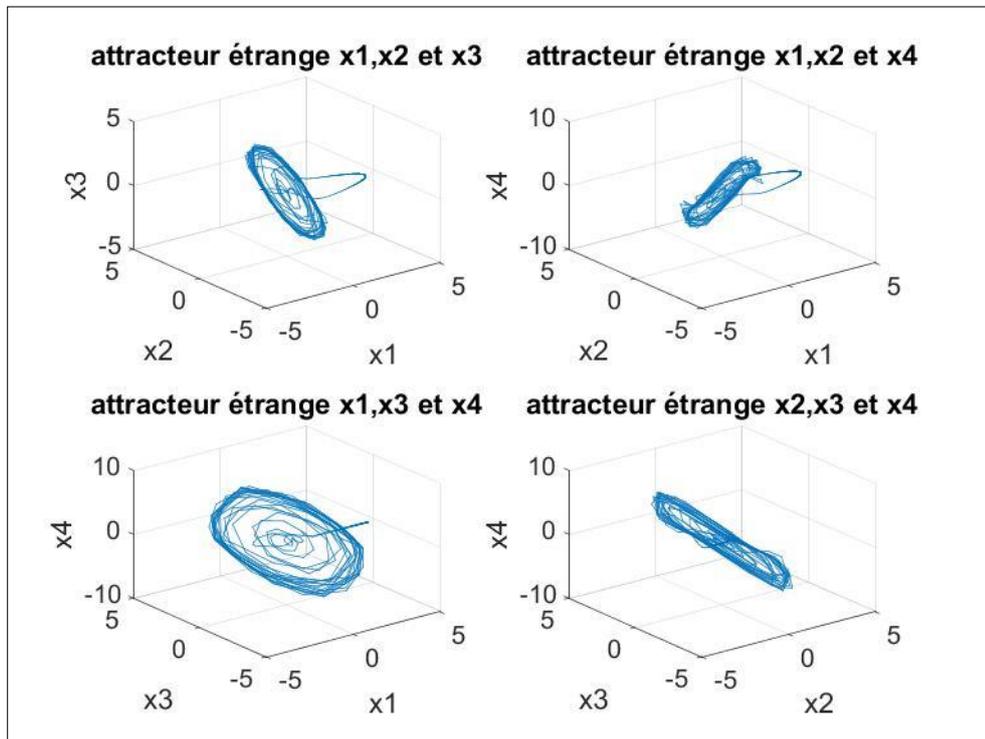


Figure 1.6: attracteurs étranges du système

1.4.4 Exposant de Lyapunov

Les systèmes chaotiques sont sensibles à de petites variations des conditions initiales, les trajectoires de deux conditions initiales proches évoluent différemment et rapidement, de ce fait l'évolution chaotique est difficile à appréhender.

Alexandre Lyapunov a introduit un outil permettant de mesurer la vitesse de divergence en le donnant le nom de l'exposant de Lyapunov. Ce dernier caractérise le taux de divergence de deux trajectoires d'un système dynamique non linéaire, soumis à des conditions initiales proches et par conséquent il permet de caractériser le degré de stabilité des systèmes. Les exposants de Lyapunov caractérisent ainsi le comportement de système non linéaire et notamment son caractère chaotique ou hyper chaotique.

Pour un attracteur non chaotique, les exposants de Lyapunov sont tous inférieurs ou égaux à zéro et leur somme est négative. Un attracteur étrange possèdera toujours au moins trois exposants de Lyapunov, dont un au moins doit être positif.

Pour un système chaotique : $\lambda_1 > 0, \sum_{i=0}^n \lambda_i < 0$.

Pour un système hyper chaotique : $\lambda_1 > 0$ et $\lambda_2 > 0, \sum_{i=0}^n \lambda_i < 0$.

λ_i : sont les exposants de Lyapunov.

Pour visualiser les Exposant de Lyapunov du système nous utilisons l'outil MATDS qui travaille sous MATLAB, et qui permet de faire l'étude d'un système dynamique.

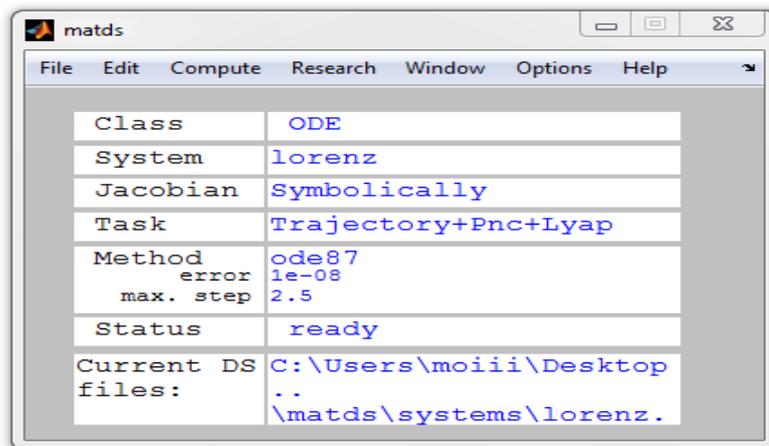


Figure 1.7: L'interface de l'outil MATDS.

Pour le système (1.4), les exposants de Lyapunov sont représentés sur la figure (1.8) suivante.

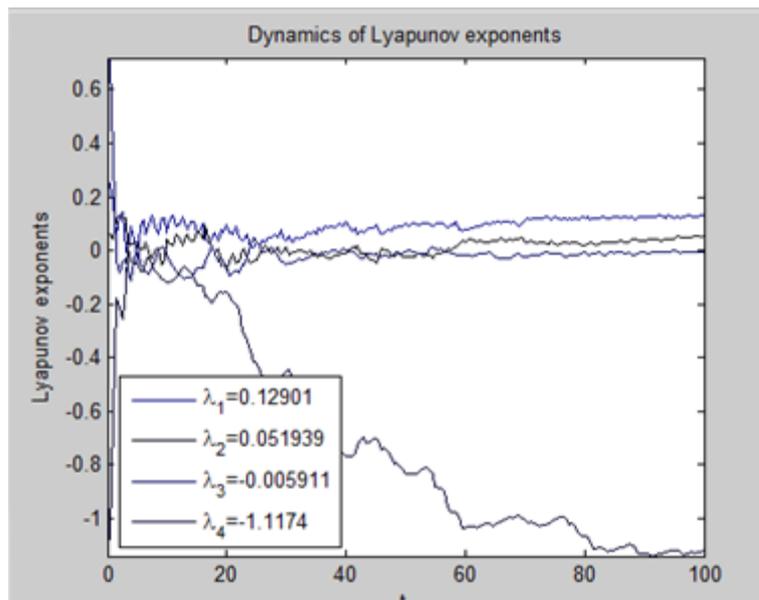


Figure 1.8: Diagramme d'exposant de Lyapunov

D'après les résultats du diagramme de l'exposant de Lyapunov nous avons quatre solutions dont deux positifs et deux négatifs et dont leur somme est négative :

Nous déduisons que le système (1,4) est un système hyper chaotique.

1.4.5 Section de Poincaré

La section de Poincaré est un outil mathématique géométrique a pour rôle de simplifier la description géométrique de la dynamique en réduisant le dimensionnement de l'espace de phase [6]. Cela se fait en coupant la trajectoire dans l'espace des phases, par exemple Pour un espace des phases tridimensionnel, la section Poincaré est un Plan dimensionnel, et ensuite d'étudier les points d'intersections de cette trajectoire avec le plan judicieusement choisi. De ce fait, la section de Poincaré permet de passer de l'analyse d'un système en temps continu à celle d'un système en temps discret.

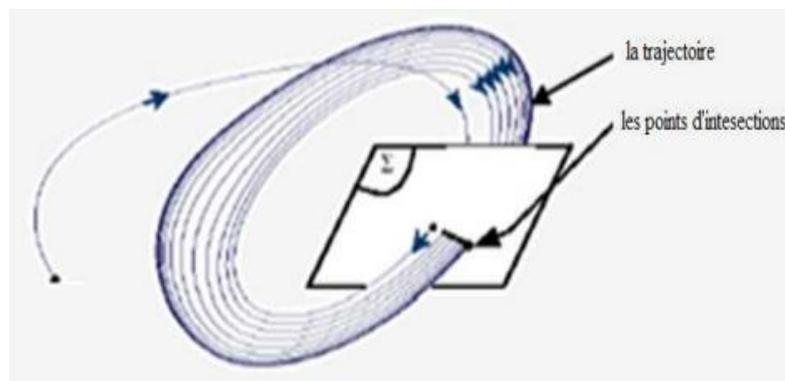


Figure 1.9: principe de la section de Poincaré

La section Poincaré peut être :

- **Un unique point** : le système est périodique.
- **Un petit nombre de points** : le système est périodique.
- **Une courbe fermée** : le système est quasi-périodique.
- **Un nuage de points** : le système est chaotique.

Pour le système (1.4) la section de Poincaré est donnée par la figure (1.10).

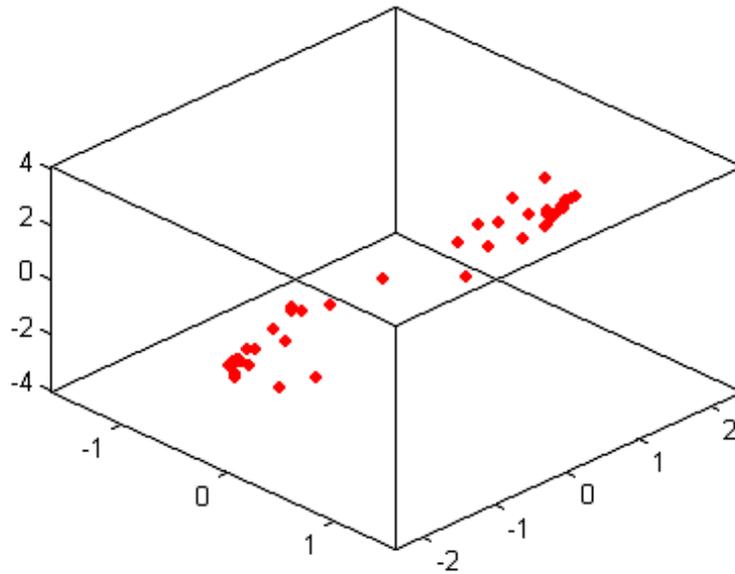


Figure 10 : la section de Poincaré du système (1.4)

1.4.6 Diagramme de bifurcation

La théorie de bifurcation est utilisée dans l'étude de la dynamique non linéaire pour décrire tout Changement dans le comportement du système lorsque certains paramètres sont modifiés [6], que signifie toute variation quantitative d'un paramètre du système engendre un changement qualitatif des propriétés de même système. Par exemple, déstabilisation d'un équilibre stable, apparition ou disparition d'un cycle ou d'un attracteur [5]. Les valeurs des paramètres au moment du changement sont appelées valeurs de bifurcation.

Dans les systèmes dynamiques, un diagramme de bifurcation est un tracé montrant les comportements possibles d'un système, à long terme, en fonction des paramètres de bifurcation. Le diagramme de bifurcation montre la division du comportement du système en deux régions : l'une au-dessus, l'autre en dessous de la valeur du paramètre particulier où le changement se produit [6]. La figure (1.11) Représente le diagramme de bifurcation du système dynamique (1.4).

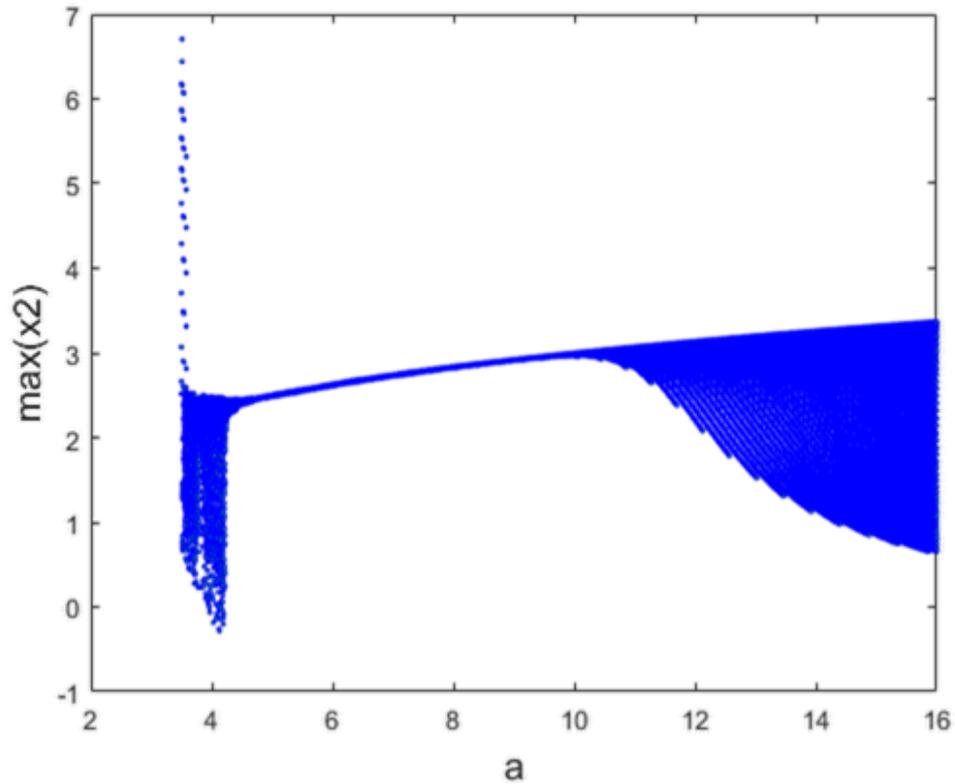


Figure 1.11: diagramme de bifurcation du système

$3.5 < a < 4.1$: le système est chaotique.

$4.1 < a < 10$: le système est périodique.

$10 < a < 16$: le système est chaotique.

1.5 Conclusion

Dans ce chapitre nous avons présenté de façon brève les généralités d'un système dynamique. Nous avons défini le chaos, donné son historique, ses domaines d'application et introduit ses caractéristiques fondamentales. Nous avons ensuite étudié un système chaotique par différents outils : point d'équilibre, exposant de Lyapunov, bifurcation, etc. Dans le prochain chapitre nous allons voir le principe et les méthodes de synchronisation des systèmes chaotiques en particulier par observateurs à mode glissant.

chapitre 2 Synchronisation des systèmes

chaotiques

2.1 Introduction

Le développement et le progrès du chaos ont conduit à son utilisation dans divers domaines, notamment dans le domaine de la communication sécurisée. Pour récupérer les données transmises le chaos utilise la synchronisation, c'est à dire que le récepteur doit suivre le comportement de l'émetteur afin de restaurer une image exacte de message émis.

La synchronisation a été observée pour la première fois par Huygens en 1673, en étudiant un système de deux pendules couplées, et depuis, plusieurs types de synchronisation ont été distingués [10].

Les caractéristiques du chaos ont conduit à la pensée que la synchronisation des systèmes chaotiques était impossible, jusqu'à ce que les deux chercheurs Pecora et Carroll sont réussis en 1990 à appliquer le phénomène de synchronisation sur deux systèmes chaotiques identiques [11]. Ils ont montré que deux systèmes chaotiques maître et esclave peuvent se synchroniser en utilisant la méthode de décomposition en sous-systèmes [10], de ce fait le chaos pourrait être utilisé dans les systèmes de transmission sécurisée, dont l'information portée sur le système maître sera récupérée par synchronisation avec le système esclave.

Ce chapitre est consacré à l'étude de la synchronisation des systèmes chaotiques. Nous commençons par présenter les types de synchronisation, puis les méthodes en basant sur la méthode de synchronisation par d'observateur à mode glissant. Pour cela, nous vérifions d'abord l'observabilité de notre système, puis nous passons à l'étude de l'observateur.

2.2 Concept de synchronisation

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

2.3 Types de synchronisation

La synchronisation est classée en deux types, suivant la manière avec laquelle les deux systèmes chaotiques sont couplés.

2.3.1 Couplage unidirectionnel

Dans le couplage unidirectionnel, l'énergie est transférée d'un système (maitre) à un autre (esclave), à l'aide d'un élément de couplage fonctionnant dans un seul sens comme par exemple un suiveur. Elle est basée sur l'injection d'une partie du signal d'erreur dans le système esclave et cela se traduit mathématiquement par les deux systèmes ci-dessous :

$$\begin{cases} \dot{x} = f_1(x) \\ \dot{z} = f_2(z) + \alpha(x - z) \end{cases} \quad (2.1)$$

Où : $x, z \in \mathbb{R}^n$; α matrice $n * n = [\alpha_1, \dots, \alpha_n]$ et $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ une fonction non linéaire continue.

Le problème de synchronisation consiste alors à trouver α tel que

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

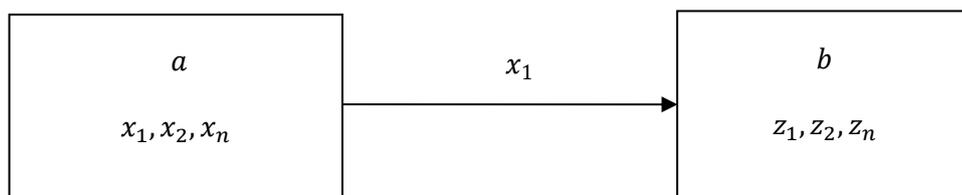


Figure 2.1: Schéma de couplage unidirectionnel

2.3.2 Couplage bidirectionnel

Dans le couplage bidirectionnel, l'élément de couplage permet l'échange de l'énergie dans les deux sens. Dans ce cas une partie de signal d'erreur est injecté dans les deux systèmes comme montré ci-dessous :

$$\begin{cases} \dot{x} = f_1(x) + \lambda(z - x) \\ \dot{z} = f_2(z) + \mu(x - z) \end{cases} \quad (2.2)$$

Où : $x, z \in \mathbb{R}^n$; λ, μ sont des matrices $n \times n$ et $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ une fonction non linéaire continue.

Le problème de synchronisation consiste à trouver λ et μ à coefficients arbitraires tel que : $\lim_{t \rightarrow \infty} \|z(t) - x(t)\| = 0$.

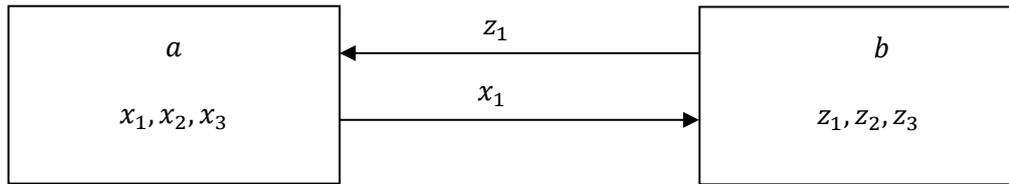


Figure 2.2: schéma de couplage bidirectionnel

2.4 Les méthodes de synchronisation

Depuis les travaux de Picora et Carroll [11], différentes méthodes de synchronisation ont été distinguées.

2.4.1 Synchronisation identique

La synchronisation identique se réalise que sur les systèmes identiques [12], si leur trajectoire convergent vers 0 lorsque le temps tends vers l'infini on dit qu'elles sont synchronisées. Prenons deux systèmes identiques, représentés par les équations suivantes :

$$\dot{x}_m(t) = f_1(x_m(t), t) \quad (2.3)$$

$$\dot{x}_s(t) = f_2(x_m(t), x_s(t), t) \quad (2.4)$$

$x_m(t), x_s(t) \in \mathbb{R}^n$: sont les états de système maitre et esclave respectivement et $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

On dit que les deux systèmes sont synchronisés si la condition suivante est vérifiée :

$$\lim_{t \rightarrow \infty} \|x_s(t) - x_m(t)\| = 0 \quad (2.5)$$

Si les deux systèmes sont différents, il s'agit alors de la synchronisation généralisée [13][14].

2.4.2 Synchronisation généralisé

La synchronisation généralisée s'applique sur systèmes chaotiques de dimension et de modèles différents [15]. Elle apparait s'il existe une relation fonctionnelle entre le système maitre et esclave qui vérifie pour une fonction g telle que [16] :

$$\lim_{t \rightarrow \infty} x_s = g(x_m) \quad (2.6)$$

Où : x_s, x_m : Sont des vecteurs d'état des systèmes maitre et de l'esclave.

2.4.3 Synchronisation projective

La synchronisation projective se manifeste lorsque les variables d'états du système chaotique esclave se synchronisent avec une constante multiple de l'état du système chaotique maitre. C'est-à-dire il faudra que la condition suivante soit vérifiée :

$$\exists \alpha_i \text{ tel que } \lim_{k \rightarrow \infty} |x_s(k) - \alpha_i x_m(k)| = 0, \forall x_s(0), x_m(0), \forall i = 1, \dots, n$$

Où : x_s, x_m : Sont des vecteurs d'état des systèmes maitre et esclave de dimension n et α : est une constante.

Le cas où tous les α_i sont égaux à 1 représente un cas de synchronisation complète [17].

La synchronisation projective est généralement utilisée pour des systèmes partiellement linéaires et permet la synchronisation aux états qui ne peuvent pas être synchronisé à un facteur près [18][19].

2.4.4 Synchronisation retardé

La synchronisation retardée a été étudiée dans le cas des systèmes chaotiques non identiques. Elle apparait si un retard $\tau > 0$ existe et vérifie la condition suivante [16] :

$$\lim_{t \rightarrow \infty} \|x_s(t) - x_m(t - \tau)\| = 0 \quad (2.7)$$

Avec :

x_m : L'état de système maitre.

x_s : L'état de système esclave.

2.5 Synchronisation chaotique à l'aide d'un observateur

Dans de nombreuses applications, le procédé doit être contrôlé ou surveillé en temps réel et ceci nécessite de pouvoir accéder aux variations en fonction du temps des variables d'état. La mesure de tout l'état n'est pas toujours possible pour des raisons technologiques, économiques, ou encore de fiabilité. De ce fait, les variables d'entrée et de sortie accessibles sont utilisées pour reconstituer l'état complet du système. L'idée repose sur l'utilisation d'un observateur d'état.

Un système dynamique est dit observable si on peut récupérer toutes ses grandeurs (de façon statique ou dynamique) par une combinaison de mesures de ses sorties et de leurs dérivés [2].

La figure (2.3) illustre le principe de synchronisation à base d'observateurs dans lequel l'émetteur est un système chaotique et le récepteur est un observateur d'état.

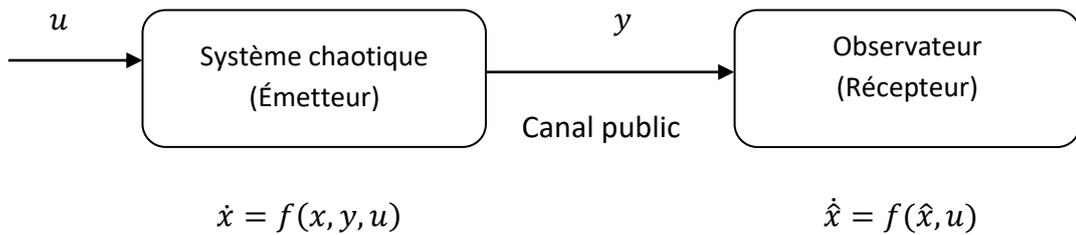


Figure 2.3: Principe de synchronisation à base d'observateurs

Pour avoir la synchronisation entre l'émetteur et le récepteur la condition suivante doit être vérifiée :

$$\lim_{t \rightarrow \infty} \|x(t) - \hat{x}(t)\| = 0 \quad (2.8)$$

2.5.1 Observabilité des systèmes non linéaires

Considérons le système décrit sous la forme suivante :

$$\begin{cases} \dot{x} = f(x(t), u(t)) \\ y = h(x(t)) \end{cases} \quad (2.9)$$

Où :

$x \in \mathbb{R}^n$ Représente le vecteur d'état.

$u \in \mathbb{R}^m$ Représente le vecteur d'entrée ou la commande.

$y \in \mathbb{R}^p$ Représente le vecteur de sortie.

Les fonctions f et h sont des champs de vecteurs supposés suffisamment continûment dérivables sur x .

L'entrée u est localement bornée et mesurable.

L'observabilité du système non linéaire (2.9) est caractérisée par le fait qu'à partir de la sortie mesurée, il faut être capable de discerner les différents états initiaux [21]. Un système est observable s'il n'admet aucune paire indiscernable [10]. L'observabilité est donc définie à partir de la notion d'indiscernabilité dont voici la définition.

- **Indiscernabilité**

Deux états initiaux $x(t_0) = x_1$ et $x(t_0) = x_2$ du système non linéaire (2.9) sont dits indiscernables sur l'intervalle de temps $[t_0, t_1]$ si, pour toute entrée $u(t)$, leurs sorties respectives $y_1(t)$ et $y_2(t)$ sont identiques sur cet intervalle [22].

- **Espace d'observabilité**

L'espace d'observabilité d'un système de la forme (2.9), est défini par le plus petit sous espace vectoriel contenant les sorties h_1, h_2, \dots, h_3 notée $O(h)$, et qui est fermé par l'opération de la dérivation de Lie par rapport au champ de vecteur $f(x, u)$, u étant fixe. On note $dO(h)$ l'espace des différentielles des éléments de $O(h)$ [23].

- **Rang d'observabilité**

Pour l'étude de l'observabilité d'un système non-linéaire, nous pouvons utiliser les dérivées de Lie. Le système est localement observable si la condition de rang d'observabilité est vérifiée, c'est-à-dire que la matrice d'observabilité (O) définie ci-dessous est de rang n [24] :

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

Avec la dérivée de lie :

$$L_f h(x) = \sum_{i=1}^n f_i(x) \frac{\partial h(x)}{\partial x_i} \quad (2.10)$$

L'écriture de $dL_f^k h$ est donnée par le co-vecteur :

$$dL_f^k h = \left(\frac{\partial L_f^k h}{\partial x_1}, \frac{\partial L_f^k h}{\partial x_2}, \dots, \frac{\partial L_f^k h}{\partial x_n} \right) \quad (2.11)$$

a- Etude d'observabilité du système

Prenons le système donné par l'équation (2.12).

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = -x_1 - x_2 - ax_3 - x_1^4 x_4 \\ y = x_1 \end{cases} \quad (2.12)$$

On utilise l'algèbre de Lie pour vérifier l'observabilité du système. On calcul la matrice d'observabilité O ci-dessous :

On prend $h=y$:

$$h = [x_1 \ 0 \ 0 \ 0], \quad dh = [1 \ 0 \ 0 \ 0].$$

On obtient alors la matrice O de la forme suivante :

$$O = \begin{pmatrix} dh \\ dL_f h \\ dL_f^2 h \\ dL_f^3 h \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

On trouve que le rang (O) = 4 = n, donc le système est observable.

2.5.2 L'observateur

Un observateur est un système dynamique qui à partir de l'entrée $u(t)$ du système et de la sortie $y(t)$ mesurée, fournira en sortie un état estimé $\hat{x}(t)$ qui devra tendre vers l'état réel $x(t)$ [10], en constituant un système dynamique auxiliaire.

Il existe plusieurs types d'observateurs non linéaire dans la littérature [2]. Nous nous intéressons sur notre étude aux observateurs à modes glissants.

2.5.3 Observateur à mode glissant

Un observateur à mode glissant est un observateur dont le terme correcteur est une fonction signe discontinue. Le principe d'un observateur à mode glissants consiste à contraindre, à l'aide d'une fonction discontinue, les dynamiques des erreurs d'estimation d'un système non linéaire d'ordre n ayant p sorties à converger vers une variété s d'ordre $(n - p)$ dite surface de glissement [25].

La surface de glissement est définie par :

$$S = \{x \in \mathbb{R}^n : S(x) = 0\} \quad (2.13)$$

Dans le cas des observateurs à mode glissant, les dynamiques concernées sont celles des erreurs d'observation $e(t) = x(t) - \hat{x}(t)$. A partir de leurs valeurs initiales $e(0)$, ces erreurs convergent vers les valeurs d'équilibre, en imposant l'évolution de ces dynamiques sur une variété S sur laquelle l'erreur d'observation entre la sortie de l'observateur et la sortie de système réel $\epsilon = y(t) - \hat{y}(t)$ est nulle. Les dynamiques sur cette variété sont stabilisées, ou assignées, de manière à borner ou annuler tout le reste de l'erreur d'observation [26]. De cette manière les erreurs convergent vers les valeurs d'équilibre.

a. Synthèse d'un observateur à mode glissant

Considérons un système d'état non linéaire d'ordre n :

$$\begin{cases} \dot{x} = f(x, u) \\ y = h(x) \end{cases} \quad (2.14)$$

L'observateur mode glissant est défini avec la structure donnée par l'équation (2.15).

$$\begin{cases} \dot{\hat{x}} = f(\hat{x}, u) - K\Gamma_s \\ \hat{y} = h(\hat{x}) \end{cases} \quad (2.15)$$

Où :

K est la matrice de gain de dimension $(n \times p)$.

Γ_s est un vecteur de dimension $(p \times 1)$ donnée par :

$$\Gamma_s = [\text{sign}(\hat{y}_1 - y_1) \dots \text{sign}(\hat{y}_p - y_p)]^T$$

$e = x - \hat{x}$: Le vecteur d'état des erreurs d'observation.

$s = e_y = y - \hat{y}$: La surface de glissement.

Le terme de correction utilisé est proportionnel à la fonction discontinue sign appliquée à l'erreur de sortie. Où la fonction $\text{sign}(x)$ est définie par [27]:

$$\text{sign}(x) = \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{si } x < 0 \end{cases} \quad (2.16)$$

L'attractivité et l'invariance de la surface de glissement sont assurées par deux conditions appelées conditions de glissement. Si ces conditions sont vérifiées, l'état observé converge vers l'état réel.

- La première condition présente le mode d'atteinte. Elle garantit l'attractivité de la surface de glissement $s = 0$ de dimension p . Celle-ci est attractive si seulement si la fonction de Lyapunov $V(x) = S^T S$ vérifie la condition : $\dot{V}(x) < 0$ si $S \neq 0$.
- La deuxième condition présente le mode de glissement, durant cette étape, la trajectoire des erreurs d'observation glisse sur la surface de glissement avec des dynamiques imposées de manière à annuler toutes les erreurs d'observation. La matrice des gains correctifs doit agir de manière à satisfaire la condition d'invariance suivante :

$$\begin{cases} \dot{S} = 0 \\ S = 0 \end{cases} \quad (2.17)$$

Durant ce mode, les dynamiques du système sont réduites et le système d'ordre n devient un système équivalent d'ordre $(n - p)$. Ces critères permettent la synthèse de l'observateur à mode glissant et déterminent son fonctionnement.

b. Observateurs à mode glissant étape par étape

L'observateur à mode glissant étape par étape a été développé pour des systèmes pouvant se mettre sous la forme, appelée forme triangulaire d'observation, suivante [28] :

$$\begin{cases} \dot{x}_1 = x_2 + g_1(x_1, u) \\ \dot{x}_2 = x_3 + g_2(x_1, x_2, u) \\ \vdots \\ \dot{x}_{n-1} = x_n + g_{n-1}(x_1, x_2, \dots, x_{n-1}, u) \\ \dot{x}_n = f_n(x_1, x_2, \dots, x_n) + g_n(x_1, x_2, \dots, x_n, u) \\ y = x_1 \end{cases} \quad (2.18)$$

Où : f_n et g_n pour $i = 1, \dots, n$, sont des fonctions scalaires, x_i sont les états du système, u est le vecteur d'entrée et y est la sortie.

La structure de l'observateur proposée est :

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 + g_1(x_1, u) - \lambda_1 \text{sign}_1(x_1 - \hat{x}_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + g_2(x_1, \tilde{x}_2, u) - \lambda_2 \text{sign}_2(\tilde{x}_2 - \hat{x}_2) \\ \vdots \\ \dot{\hat{x}}_{n-1} = \hat{x}_n + g_{n-1}(x_1, \tilde{x}_2, \dots, \tilde{x}_{n-1}, u) - \lambda_{n-1} \text{sign}_{n-1}(\tilde{x}_{n-1} - \hat{x}_{n-1}) \\ \dot{\hat{x}}_n = f_n(x_1, \tilde{x}_2, \dots, \tilde{x}_n) + g_{n-1}(x_1, \tilde{x}_2, \dots, \tilde{x}_n, u) - \lambda_n \text{sign}_n(\tilde{x}_n - \hat{x}_n) \\ y = x_1 \end{cases} \quad (2.19)$$

Avec :

$\hat{x}_i, i=1\dots n$: Sont les variables reconstruites par l'observateur.

Avec : $\tilde{x}_1 = x_1$

$\tilde{x}_i = \hat{x}_i + \lambda_{i-1} \text{sign}_{eq\ i-1}(\tilde{x}_{i-1} - \hat{x}_{i-1})$ Pour $i > 1$.

Avec sign_{eq} désigne la fonction $\text{sign}(\cdot)$ classique filtrée par un filtre passe bas. La fonction sign_i est définie de manière à imposer que le terme correctif ne soit actif que si $(\tilde{x}_j - \hat{x}_i) = 0$, pour $j = 1, \dots, i$ c'est à dire, s'il existe $j \in \{1, \dots, i-1\}$ tel que $(\tilde{x}_j - \hat{x}_i) = 0$ alors la fonction sign_i est mise à zéro sinon elle est égale à la fonction $\text{sign}(\cdot)$ usuelle. La convergence des erreurs d'observation en temps fini n'est assurée que si le système est à entrées bornées et à états bornées pour une durée finie. Si cette condition est vérifiée, alors les λ_i peuvent être choisis tel que

l'état de l'observateur \hat{x} converge en un temps fini vers l'état x réel du système. Cependant cette convergence se fait par étapes :

Étape 1 : dans cette étape on assure la convergence $e_1 = x_1 - \hat{x}_1$ vers zéro dans un temps $< t_1$. Pour $i > 1$ toutes les fonctions $sign_i$ sont égales à zéro, les dynamiques des erreurs d'observation $e = x - \hat{x}$ sont donc :

$$\left\{ \begin{array}{l} \dot{e}_1 = e_2 - \lambda_1 sign_1(x_1 - \hat{x}_1) \\ \dot{e}_2 = e_3 + g_2(x_1, x_2, u) - g_2(x_1, \hat{x}_2, u) \\ \vdots \\ \dot{e}_{n-1} = e_n + g_{n-1}(x_1, x_2, \dots, x_{n-1}, u) - g_{n-1}(x_1, \hat{x}_2, \dots, \hat{x}_{n-1}, u) \\ \dot{e}_n = f_n(x_1, x_2, \dots, x_n) + f_n(x_1, \hat{x}_2, \dots, \hat{x}_n) - g_n(x_1, x_2, \dots, x_n, u) - g_n(x_1, \hat{x}_2, \dots, \hat{x}_n, u) \end{array} \right. \quad (2.20)$$

On considère la fonction de Lyapunov définie positive :

$$v_1 = \frac{1}{2} e_1^2 \quad (2.21)$$

Alors : $\dot{v}_1 = e_1 \dot{e}_1$; $\dot{v}_1 = e_1 (e_2 - \lambda_1 sign(e_1))$

En choisissant $\lambda_1 > |e_2|_{max}$, l'erreur d'observation e_1 converge vers zéro en un temps fini t_1 . Après cet instant, e_1 reste égale à zéro et on obtient alors :

$e_2 = \lambda_1 sign(x_1 - \hat{x}_1)$ Ceci implique que $\tilde{x}_2 = x_2$.

Étape 2 : L'objectif dans cette étape est d'atteindre la surface de glissement $e_2 = x_2 - \hat{x}_2 = 0$. Pour rester sur la surface $e_1 = 0$, il faut que $\lambda_1 > |e_2|_{max}$, et cela est vérifié le fait que e_2 est strictement décroissante après t_1 .

Les dynamiques des erreurs d'observation sont alors :

$$\left\{ \begin{array}{l} \dot{e}_1 = e_2 - \lambda_1 sign(x_1 - \hat{x}_1) = 0 \\ \dot{e}_2 = e_3 + g_2(x_1, x_2, u) - g_2(x_1, \hat{x}_2, u) - \lambda_2 sign(x_1 - \hat{x}_2) = e_3 - \lambda_2 sign(e_2) \\ \vdots \\ \dot{e}_{n-1} = e_n + g_{n-1}(x_1, x_2, \dots, x_{n-1}, u) - g_{n-1}(x_1, \hat{x}_2, \dots, \hat{x}_{n-1}, u) \\ \dot{e}_n = f_n(x_1, x_2, \dots, x_n) + f_n(x_1, \hat{x}_2, \dots, \hat{x}_n) - g_n(x_1, x_2, \dots, x_n, u) - g_n(x_1, \hat{x}_2, \dots, \hat{x}_n, u) \end{array} \right. \quad (2.22)$$

En choisissant la fonction de Lyapunov :

$$v_2 = \frac{1}{2} (e_1^2 + e_2^2) \quad (2.23)$$

On aura : $\dot{v}_2 = e_1(e_2 - \lambda_1 \text{sign}(e_1)) + e_2(e_3 - \lambda_2 \text{sign}(e_2)) = e_2(e_3 - \lambda_2 \text{sign}(e_2))$

Si $\lambda_2 > |e_3|_{max}$ alors e_2 converge vers zéro après un temps fini $t_2 > t_1$. L'erreur d'observation est strictement décroissante durant la période $[t_1, t_2]$ ce qui implique qu'après un temps fini t_2 , $\tilde{x}_3 = x_3$.

Ainsi, étape par étape nous obtenons la convergence de toutes les composantes de l'erreur d'observation vers zéro et celles de \hat{x} vers x pour tous $i < n$, sous condition, que $\lambda_i > |e_{(i-1)}|_{max}$ durant $[t_i, t_{(i+1)}]$.

Étape n : cette étape commence à l'instant $(t_{(n-1)})$ à cet instant $e_k = 0$ pour tous $k < n$.

$$\left\{ \begin{array}{l} \dot{e}_1 = e_2 - \lambda_1 \text{sign}(x_1 - \hat{x}_1) = 0 \\ \dot{e}_2 = e_3 - \lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2) = 0 \\ \dot{e}_{n-1} = e_n - \lambda_{n-1} \text{sign}(\tilde{x}_n - \hat{x}_n) \\ \dot{e}_n = f_n(x_1, x_2, \dots, x_n) + f_n(x_1, \hat{x}_2, \dots, \hat{x}_n) - g_n(x_1, x_2, \dots, x_n, u) - g_n(x_1, \hat{x}_2, \dots, \hat{x}_n, u) - \lambda_n \text{sign}(x_n - \hat{x}_n) \\ \quad = -\lambda_n \text{sign}(e_n) \end{array} \right. \quad (2)$$

De la même façon on choisit la fonction de Lyapunov :

$$v_n = \frac{1}{2}(e_1^2 + e_2^2 + \dots + e_n^2) \quad (2.25)$$

On obtient alors :

$$\dot{v}_n = e_n(e_{n+1} - \lambda_n \text{sign}(e_n)) \quad (2.26)$$

Ainsi, e_n converge vers zéro en un temps fini $t_n > t_{(n-1)}$ pour toutes valeurs de $\lambda_n > 0$. Si évidemment toutes les conditions sur λ_k , $k < n$ sont elles aussi vérifiées.

2.6 Conclusion

Dans ce chapitre nous avons présenté le concept de synchronisation et le principe de couplage des systèmes chaotiques. Ensuite, différents types de synchronisation ont été décrits. Nous avons présenté la synchronisation à l'aide d'un observateur qui est un observateur à mode glissant, et nous avons expliqué son principe de fonctionnement. Dans le prochain chapitre nous allons utiliser ces

notions pour transmettre les données pour lequel l'observateur sera utilisé en tant que récepteur dans notre système de communication.

chapitre 3 Insertion et récupération du message

3.1 Introduction

À l'arrivée des systèmes de communications modernes (Internet, etc...) la protection des données échangées est devenue une nécessité, pour assurer la fiabilité, l'authenticité et la confidentialité du message. Le cryptage ou chiffrement est l'outil majeur permettant d'assurer la sécurité des données, il fait transformer un message clair à un message crypté incompréhensible, et donc on ne peut pas accéder aux données de messages, sauf le destinataire qui doit avoir une clé de cryptage pour décrypter le signal transmis et reconstruit le message.

Après les travaux de Pecora et Carroll en 1990 qui ont montré que les systèmes chaotiques pouvaient être synchronisés, plusieurs applications du chaos ont été développées. Les propriétés des systèmes chaotiques, telles que la sensibilité aux conditions initiales et l'aspect pseudo-aléatoire répondent aux exigences de sécurité et permis l'application des systèmes chaotiques en cryptographie.

Le principe de la cryptographie chaotique est de noyer le message à transmettre dans un signal porteur chaotique et de le transmettre à un récepteur qui connaît les caractéristiques du générateur de chaos, et qui pourra par synchronisation avec l'émetteur de récupérer le message émis. Le diagramme principal de la communication sécurisée par le chaos est montré sur la figure (3.1).

Plusieurs techniques de cryptage ont été proposées par les chercheurs afin de cacher un message dans un générateur chaotique et ensuite le restaurer, parmi ces techniques nous citons le masquage chaotique, la modulation chaotique, la commutation chaotique...

Dans ce chapitre nous allons exposer les différentes techniques de la transmission chaotiques qui reposent sur la synchronisation chaotique. Ensuite, nous allons introduire un système de communication sécurisée basé sur deux techniques de cryptage, par inclusion et par addition, et la synchronisation par observateur a mode glissant.

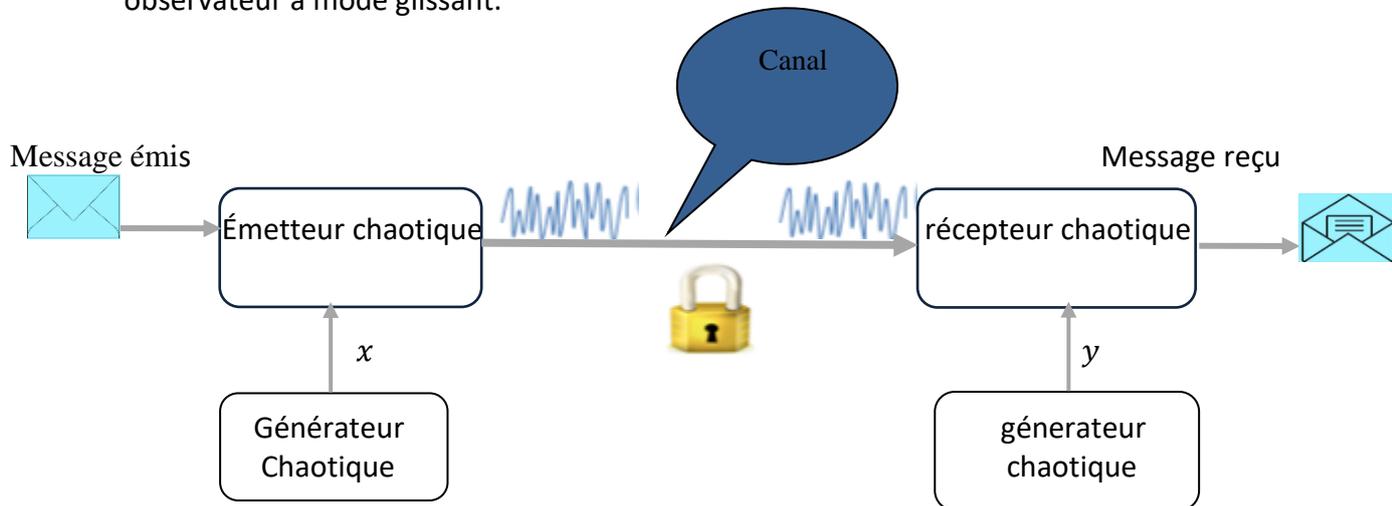


Figure 3.1: principe de la communication sécurisée à base du chaos

3.2 Propriétés des systèmes de communication à base du chaos

Dans cette section on va étudier les propriétés des systèmes de communication chaotiques [29].

3.2.1 Spectre à large bande

Le spectre du signal chaotique est de nature large bande, c'est une propriété essentielle d'un système chaotique, elle est favorable dans les applications qui ont besoin une importante robustesse face aux interférences et une faible probabilité de détection.

3.2.2 Comportement apériodique

Un signal chaotique est non périodique et son évolution ne peut être prédite sur un long intervalle de temps. La non-périodicité est une propriété attractive pour la sécurisation des communications puisque la périodicité, dans la communication

sécurisée engendre des pics spectraux indésirables. De plus il est plus difficile de développer un modèle de prévisions pour les dynamiques non périodiques.

3.3 Techniques de transmission chaotique

Il existe plusieurs techniques qui peuvent servir comme moyen de masquage de l'information dans le chaos ; nous en décrivons ici quelques-unes :

3.3.1 Le masquage additif

Le masquage additif appelé aussi le masquage chaotique est la technique la plus simple et la toute première des techniques de cryptage. Elle consiste à effectuer une simple addition entre le signal d'information $m(t)$ de nature binaire ou analogique et un signal porteur chaotique $x(t)$ généré par le système émetteur. La somme des deux signaux constitue le message chiffré $y(t)$ qui va être transmis à travers le canal de transmission vers le système récepteur. Au niveau du récepteur et grâce à une synchronisation avec le système émetteur, le message d'information sera reconstitué par une simple soustraction entre le signal transmis et le signal porteur chaotique estimé ($\hat{m}(t) = y(t) - \hat{y}(t)$). La figure (3.2) illustre le principe de cette technique.

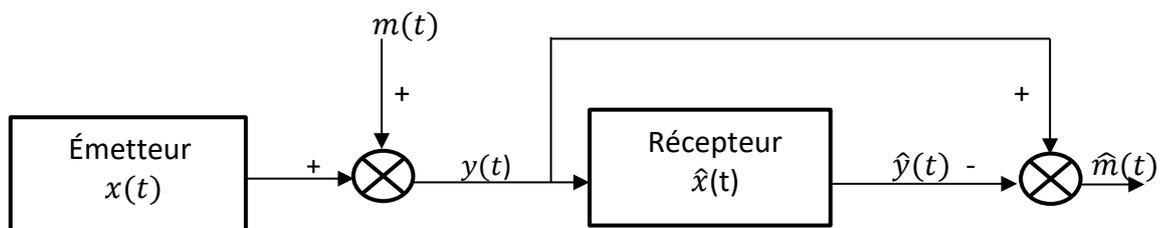


Figure 3.2: schéma représentatif de la technique de masquage additif

3.3.2 Modulation paramétrique

Le principe de modulation paramétrique consiste à utiliser le signal d'information généralement de nature binaire, pour moduler l'un des paramètres du système chaotique émetteur. Au niveau du récepteur, le système synchronise d'une manière

adaptative avec l'émetteur tout en suivant le changement du paramètre modulé. Le signal d'information est ensuite reconstitué par l'intermédiaire d'une loi d'adaptation. La figure (3.3) représente le schéma de la technique de modulation paramétrique.

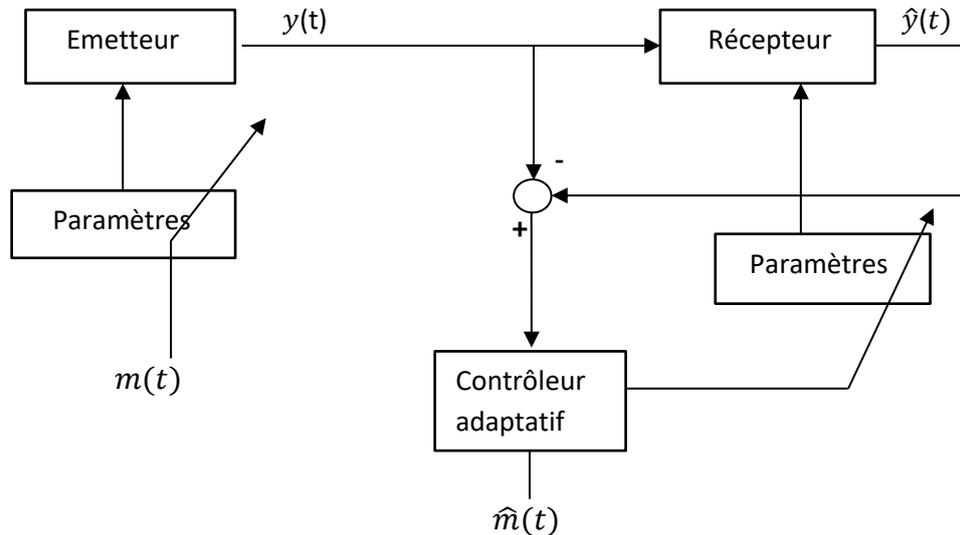


Figure 3.3: schéma représentatif de la technique de modulation paramétrique

Afin de préserver le système maître, la façon d'injecter le message et donc la fonction de modulation des paramètres ne doivent pas altérer le caractère chaotique du signal envoyé au récepteur, 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 [10].

3.3.3 La commutation chaotique

La commutation chaotique, est aussi connue sous nom de « Chaos Shift Keying » ou « chaotic switching », Au niveau de l'émetteur, on dispose de deux systèmes générant les signaux chaotiques $x_0(t)$ et $x_1(t)$. Un signal d'information de type binaire $m(t) = \{0,1\}$ est utilisé pour commuter entre $x_0(t)$ encodant le bit 1 et $x_1(t)$ encodant le bit 0. Le signal crypté résultant $y(t)$ est transmis à travers le canal de transmission vers le système récepteur constitué de deux systèmes esclaves identiques à ceux de l'émetteur. Pour chaque valeur du message, l'un des

deux systèmes se synchronise avec l'émetteur et un bloc de comparaison permet de relever la valeur du message notée $m'(t)$ [30][31].

Le principe de cette technique est lié à la valeur du message $m(t)$ à l'instant t , Si $m(t)$ prend la valeur 0, alors $x_0'(t)$ se synchronise, et $x_1'(t)$ ne se synchronise pas, ainsi l'erreur de synchronisation $e_0(t) = y(t) - \hat{y}_0(t)$ tend vers 0, tandis que l'erreur $e_1(t) = y(t) - \hat{y}_1(t)$ sera d'amplitude non nulle. Le processus est symétrique lorsque $m(t) = 1$. Le schéma du principe de cette technique est représenté par la figure (3.4).

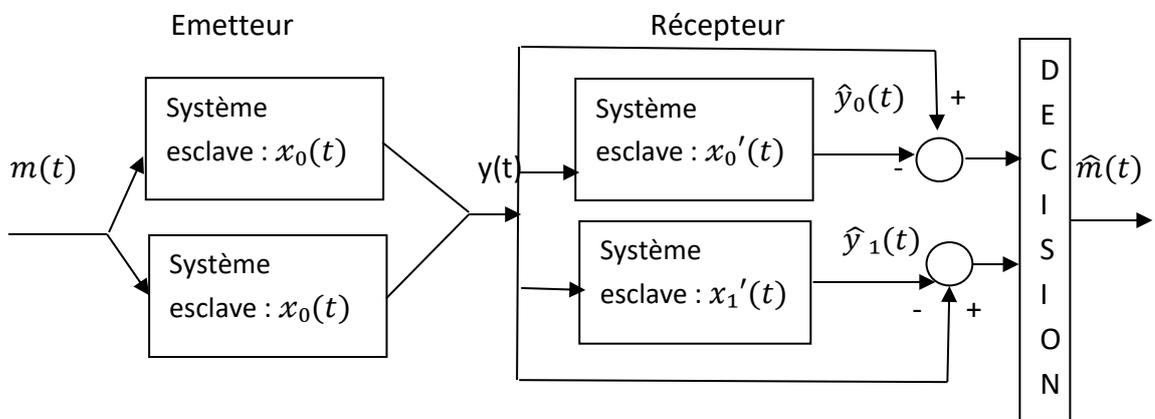


Figure 3.4: schéma représentatif de la technique commutation chaotique

La commutation chaotique présente relativement plus de robustesse au bruit de canal, néanmoins, les crypto systèmes utilisant cette technique possèdent une faible vitesse de transmission car à chaque changement de bit on doit tenir compte du temps de convergence nécessaire pour la mise en place de la synchronisation [10].

3.3.4 Cryptage mixte

Cette technique est un mixage entre les systèmes cryptographiques classiques et la synchronisation chaotique. Pour avoir le cryptage, l'émetteur chaotique génère un signal $c(t)$ appelé aussi clé c'est lui qui va crypter l'information $m(t)$. Le signal crypté $u_c(t)$ résultant est réinjecté dans la dynamique de l'émetteur pour le rendre plus complexe. Un signal $y(t)$, en fonction de variable d'état de l'émetteur sera

transmis au récepteur. Au niveau du récepteur une synchronisation avec l'émetteur va être établie, et la clé est alors reconstruite par le récepteur, qui peut finalement décoder le message. Cette technique est illustrée par la figure (3.5).

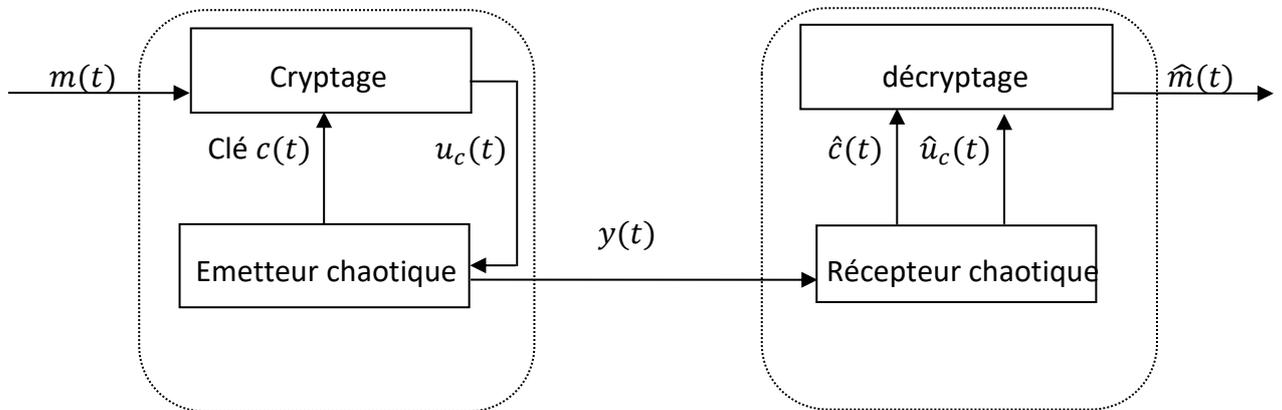


Figure 3.5: schéma représentatif de la technique Cryptage mixte

3.3.5 Transmission à deux voies

Cette technique consiste à séparer la synchronisation et le cryptage en utilisant deux voies de communication, de sorte que le signal de synchronisation est transmis séparément du signal porteur de message. L'émetteur génère 2 signaux un pour la synchronisation qui est transmis dans le 1^{er} canal vers le récepteur et un deuxième crypté qui porte le message transmit dans le 2^{eme} canal. Le schéma de principe est illustré dans la figure (3.6).

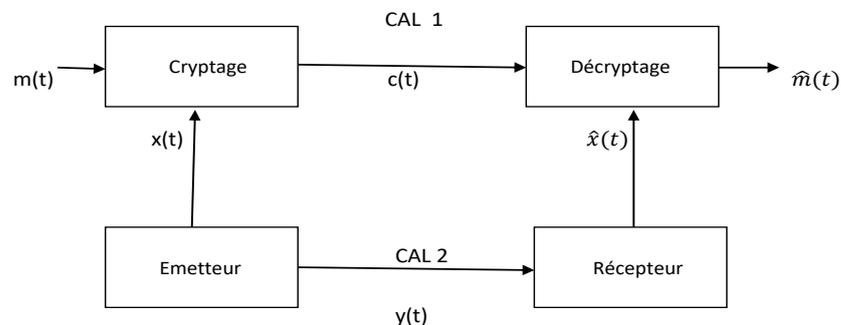


Figure 3.6: schéma représentatif de la technique transmission à deux voies

3.3.6 Cryptage par inclusion

Dans cette technique, le signal d'information est injecté dans la dynamique de l'émetteur. Le récepteur a pour but de synchroniser avec l'émetteur et de reconstruire l'information. Ceci se fait principalement par deux techniques, reposant soit sur les observateurs à entrées inconnues, soit sur l'inversion du système émetteur.

a. Observateurs à entrées inconnues

Le schéma de la figure (3.7) illustre un problème classique d'estimation d'état non linéaire à entrées inconnues ; il faut reconstruire l'état $x(t)$ du système émetteur et également l'entrée inconnue $u(t)$ pour finalement restaurer l'information.

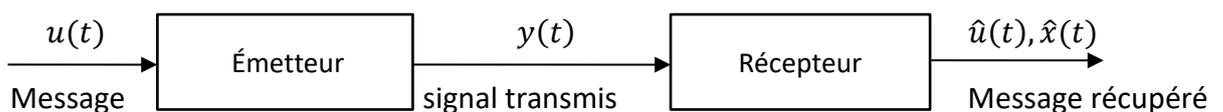


Figure 3.7: Observateurs a entrées inconnues

b. Méthode par inversion

L'article [32] présente un processus de décryptage par inversion, c'est à dire, le récepteur est conçu en inversant le modèle de l'émetteur. La figure (3.8) présente le principe général de cette approche.

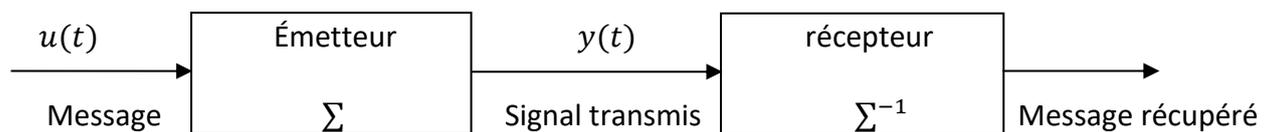


Figure 3.8 : Principe de cryptage par inversion

Cette technique présente beaucoup d'avantages. Elle présente un niveau de sécurité nettement élevé par rapport aux techniques précédentes, puisque le signal

d'information est masqué dans la dynamique du système maître et que le signal chaotique disponible dans le canal public ne porte pas l'information d'une manière directe. Elle reste est très utilisé en pratique [33][34].

3.4 Étude d'émetteur et récepteur chaotique

Dans cette section nous allons étudier un système de communication basé sur un système dynamique hyper chaotique. L'objectif de cette étude est de pouvoir transmettre un message et le récupérer en toute sécurité. Pour ce faire, nous allons utiliser le système émetteur hyperjerk donné par l'équation (3.1). Nous utilisons deux méthodes de cryptage, dans un premier temps nous allons utiliser la technique de cryptage par inclusion pour introduire le message à l'une des dynamiques du système (quatrième dynamique), Puis nous passerons à technique par addition. Dont le message doit être choisi d'une façon à ne pas détruire le comportement chaotique du système. Nous utilisons pour la récupération du message un observateur à mode glissant étape par étape, de la sorte que les états du système et le message seront reconstitué étape par étape.

3.4.1 Cryptage par inclusion:

Au niveau de l'émetteur, le message est inclus dans la quatrième dynamique du système maître puis transmis par la ligne de transmission grâce à la sortie y .

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = -x_1 - x_2 - ax_3 - x_4x_1^4 + m_0 \\ y = x_1 \end{array} \right. \quad (3.1)$$

Le message m_0 est un signal sinusoïdal d'amplitude de $b = 1$ et de fréquence de ($f = 2 * \pi * 10$).

$$m_0 = b \sin (2 * \pi * 10 * t)$$

Les conditions initiales du système maître sont prises comme suit :

$$x_1 = 0.2, x_2 = 0.2, x_3 = 0.3, x_4 = 0.1.$$

La figure (3.9) représente l'émetteur chaotique après l'insertion de message sous Matlab.

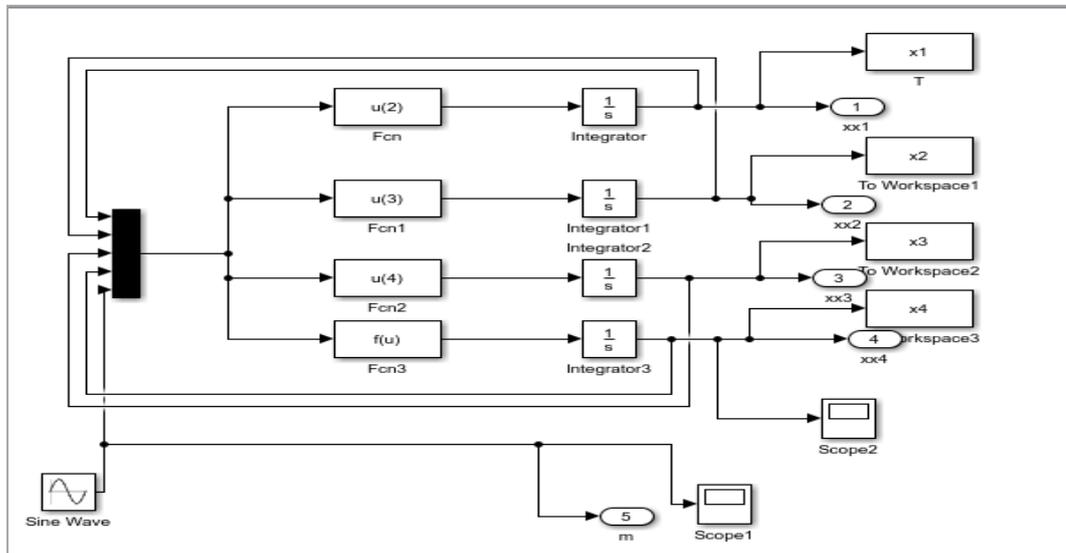


Figure 3.9: l'émetteur chaotique sous matlab (Simulink) avec insertion de message par inclusion

Le message crypté envoyé au récepteur est représenté par la figure (3.10).

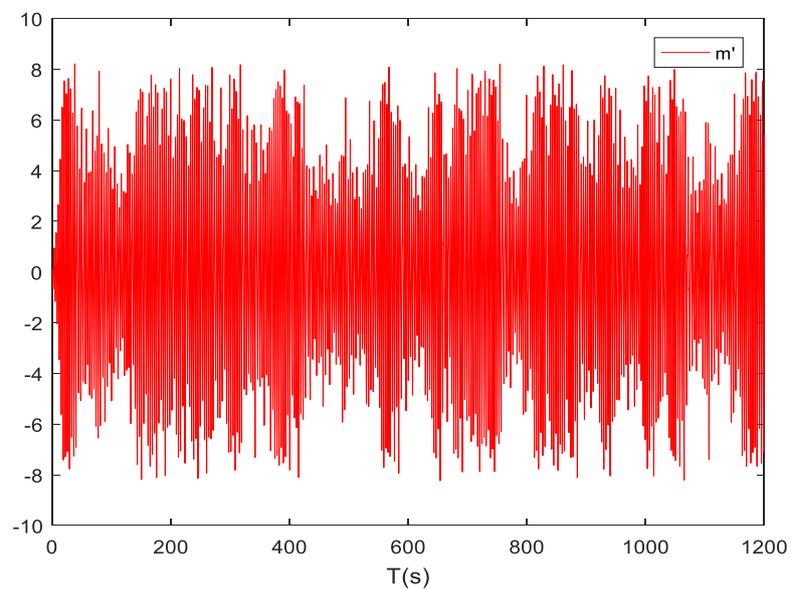


Figure 3.10: message crypté

a. Synthèse de l'observateur

Le récepteur est formé par un observateur. Il est régi par les équations dynamiques suivantes :

$$\begin{cases} \dot{\hat{x}}_1 = \tilde{x}_2 + \lambda_1 \text{sign}(x_1 - \hat{x}_1) \\ \dot{\hat{x}}_2 = \tilde{x}_3 + \lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2)E_1 \\ \dot{\hat{x}}_3 = \tilde{x}_4 + \lambda_3 \text{sign}(\tilde{x}_3 - \hat{x}_3)E_1E_2 \\ \dot{\hat{x}}_4 = -x_1 - \tilde{x}_2 - a\tilde{x}_3 - \tilde{x}_4x_1^4 \end{cases} \quad (3.1)$$

Avec les variables auxiliaires :

$$\begin{cases} \tilde{x}_2 = \hat{x}_2 + \lambda_1 \text{sign}(x_1 - \hat{x}_1) \\ \tilde{x}_3 = \hat{x}_3 + \lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2)E_1 \\ \tilde{x}_4 = \hat{x}_4 + \lambda_3 \text{sign}(\tilde{x}_3 - \hat{x}_3)E_1E_2 \end{cases} \quad (3.2)$$

En remplaçant les variables auxiliaires La dynamique de l'observateur devient :

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 + \lambda_1 \text{sign}(x_1 - \hat{x}_1) + \lambda_1 \text{sign}(x_1 - \hat{x}_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + \lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2) + \lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2)E_1 \\ \dot{\hat{x}}_3 = \hat{x}_4 + \lambda_3 \text{sign}(\tilde{x}_3 - \hat{x}_3) + \lambda_3 \text{sign}(\tilde{x}_3 - \hat{x}_3)E_1E_2 \\ \dot{\hat{x}}_4 = -x_1 - (\hat{x}_2 + \lambda_1 \text{sign}(x_1 - \hat{x}_1)) - a(\hat{x}_3 + \lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2))E_1 - x_1^4 \\ \quad (\hat{x}_4 + \lambda_3 \text{sign}(\tilde{x}_3 - \hat{x}_3))E_1E_2 \end{cases} \quad (3.3)$$

$$\Rightarrow \begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 + 2\lambda_1 \text{sign}(x_1 - \hat{x}_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + 2\lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2) \\ \dot{\hat{x}}_3 = \hat{x}_4 + 2\lambda_3 \text{sign}(\tilde{x}_3 - \hat{x}_3) \\ \dot{\hat{x}}_4 = -x_1 - (\hat{x}_2 + \lambda_1 \text{sign}(x_1 - \hat{x}_1)) - a(\hat{x}_3 + \lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2))E_1 - x_1^4 \\ \quad (\hat{x}_4 + \lambda_3 \text{sign}(\tilde{x}_3 - \hat{x}_3))E_1E_2 \end{cases} \quad (3.4)$$

Les erreurs d'observation sont définies par :

$$\begin{cases} e_1 = x_1 - \hat{x}_1 \\ e_2 = x_2 - \hat{x}_2 \\ e_3 = x_3 - \hat{x}_3 \\ e_4 = x_4 - \hat{x}_4 \end{cases} \quad (3.5)$$

Ainsi la dynamique de l'erreur est donnée par : $\dot{e}_i = \dot{x}_i - \dot{\hat{x}}_i$

$$\begin{cases} \dot{e}_1 = e_2 - 2\lambda_1 \text{sign}(x_1 - \hat{x}_1) \\ \dot{e}_2 = e_3 - 2\lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2)E_1 \\ \dot{e}_3 = e_4 - 2\lambda_3 \text{sign}(\tilde{x}_3 - \hat{x}_3)E_1E_2 \\ \dot{e}_4 = -e_2 - ae_3 - x_1^4 e_4 - \lambda_1 \text{sign}(x_1 - \hat{x}_1) - a\lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2)E_1 \\ \quad - x_1^4 \lambda_3 \text{sign}(\tilde{x}_3 - \hat{x}_3)E_1E_2 + \tilde{m}_0 \end{cases} \quad (3.6)$$

b. Etude de convergence de l'observateur étape par étape :

L'observateur à mode glissant fonctionne étape par étape :

$$E_1 = \begin{cases} 1 & \text{si } \hat{x}_1 = x_1 \\ 0 & \text{si } E_1 = 1 \text{ et } \hat{x}_2 = x_2 \\ 1 & \text{si } E_1 = 1, E_2 = 1 \text{ et } \hat{x}_3 = x_3 \\ 0 & \end{cases} \quad E_2 = \begin{cases} 1 & \text{si } E_1 = 1 \text{ et } \hat{x}_2 = x_2 \\ 0 & \end{cases} \quad E_3 = \begin{cases} 1 & \text{si } E_1 = 1, E_2 = 1 \text{ et } \hat{x}_3 = x_3 \\ 0 & \end{cases}$$

Pour montrer la stabilité de l'erreur et prouver la convergence de l'observateur nous utilisons une fonction de Lyapunov.

Nous prenons :

$V = \frac{1}{2} S^T S$, cette fonction est positive et sa dérivée (\dot{V}) doit être négative.

$$S^T = (e_1 \quad e_2 \quad e_3 \quad e_4); S = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix}$$

$$\frac{dV}{dt} = S^T \dot{S} = (e_1 \quad e_2 \quad e_3 \quad e_4) \begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \end{pmatrix} = e_1 \dot{e}_1 + e_2 \dot{e}_2 + e_3 \dot{e}_3 + e_4 \dot{e}_4 \leq 0$$

Pour que $\frac{dV}{dt} \leq 0 \Leftrightarrow e_i \dot{e}_i \leq 0$ pour $i = 1 \dots 4$

Etape1 : nous considérons $E_1=0$ (nous ne sommes pas encore dans la surface de glissement).

La dynamique de l'erreur est donnée par :

$$\dot{e}_1 = e_2 - 2\lambda_1 \text{sign}(e_1)$$

$$\dot{V}_1 = e_1 \cdot \dot{e}_1 = e_1(e_2 - 2\lambda_1 \text{sign}(e_1))$$

Tous les états sont bornés ainsi on peut choisir $\lambda_1 > |e_2|$ de sorte que la surface de glissement soit atteinte. Après un temps fini t_1 grâce à la convergence en temps fini des modes glissant, Nous obtenons $e_1 = \dot{e}_1 = 0$ et alors $\hat{x}_2 = x_2$ et $E_1 = 1$.

Etape2 : pour $t > t_1$, $\hat{x}_2 = x_2$, $E_1 = 1$, nous avons alors la dynamique :

$$\dot{e}_2 = -e_3 - 2\lambda_2 \text{sign}(e_2)$$

$$\dot{V}_2 = e_1 \cdot \dot{e}_1 + e_2 \cdot \dot{e}_2 = e_1 \cdot \dot{e}_1 + e_2(-e_3 - 2\lambda_2 \text{sign}(e_2))$$

Ainsi, par le choix de $\lambda_2 > |e_3|$, nous obtenons la convergence de e_2 . Après un temps t_2 ($t > t_2 > t_1$) nous auront $e_2 = \dot{e}_2 = 0$ et donc $\tilde{x}_3 = x_3$ et $E_2 = 1$.

Etape3 : pour $\tilde{x}_3 = x_3$ et $E_1 = 1$ on a donc $E_2 = 1$

$$\dot{e}_3 = -e_4 - 2\lambda_3 \text{sign}(e_3)$$

$$\dot{V}_2 = e_1 \cdot \dot{e}_1 + e_2 \cdot \dot{e}_2 + e_3 \dot{e}_3 = e_1 \cdot \dot{e}_1 + e_2 \dot{e}_2 + e_3(-e_4 - 2\lambda_3 \text{sign}(e_3))$$

En choisissant $\lambda_3 > |e_4|$ nous obtenons après un temps fini t_3 , $e_3 = \dot{e}_3 = 0$ et donc $\tilde{x}_4 = x_4$, $E_3 = 1$.

Etape4 : Le but de cette étape est de reconstituer le message.

On considère l'équation de \dot{e}_4 :

$$\begin{aligned} \dot{e}_4 = & -e_2 - ae_3 - x_1^4 e_4 - \lambda_1 \text{sign}(x_1 - \hat{x}_1) - \lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2) E_1 \\ & - x_1^4 \lambda_3 \text{sign}(\tilde{x}_3 - \hat{x}_3) E_1 E_2 + \lambda_4 \text{sign}(\tilde{x}_4 - \hat{x}_4) E_1 E_2 E_3 + m_0 \end{aligned}$$

si pour $t > t_3$: $E_1 = 1, E_2 = 1, \tilde{x}_4 = x_4, \tilde{x}_3 = x_3, \tilde{x}_2 = x_2$ et $\hat{x}_1 = x_1$.

On a $e_4 = \dot{e}_4 = 0$

Lorsque toutes les variables ont convergé : $e_1 = \dot{e}_1 = e_2 = \dot{e}_2 = e_3 = \dot{e}_3 = e_4 = \dot{e}_4 = 0$

Nous reconstituons le message dont son estimation est donnée par :

$$\tilde{m}_0 = \lambda_4 \text{sign}(\tilde{x}_4 - \hat{x}_4) E_1 E_2 E_3$$

c. Résultat et simulation

La reconstitution du message est obtenue après la synchronisation de tous les états dont cette dernière est obtenue étape par étape, commençons respectivement par x_1 jusqu'aux x_4 .

Prenant les paramètres de simulation ci-dessous :

Les conditions initiales de l'observateur :

$$\hat{x}_1(0) = 0.85, \hat{x}_2(0) = 0.7, \hat{x}_3(0) = 0.44, \hat{x}_4(0) = 0.44$$

Les gains de l'observateur :

$$\lambda_1 = 20, \lambda_2 = 15, \lambda_3 = 10, \lambda_4 = 300$$

Le schéma de la transmission complet sous matlab(Simulink) en utilisant la méthode d'inclusion est donné par la figure (3.11).

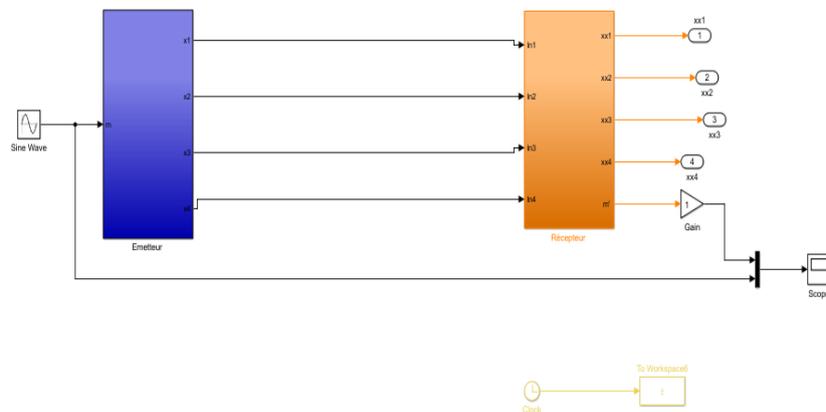


Figure 3.11: Schéma de transmission (par inclusion)

Le message reconstruit est illustré dans la figure (3.12).

Les figures(3.13)(3.14),(13.15) et (3.16) montrent les réponses temporelles ainsi que les erreurs de synchronisation entre l'émetteur et le récepteur.

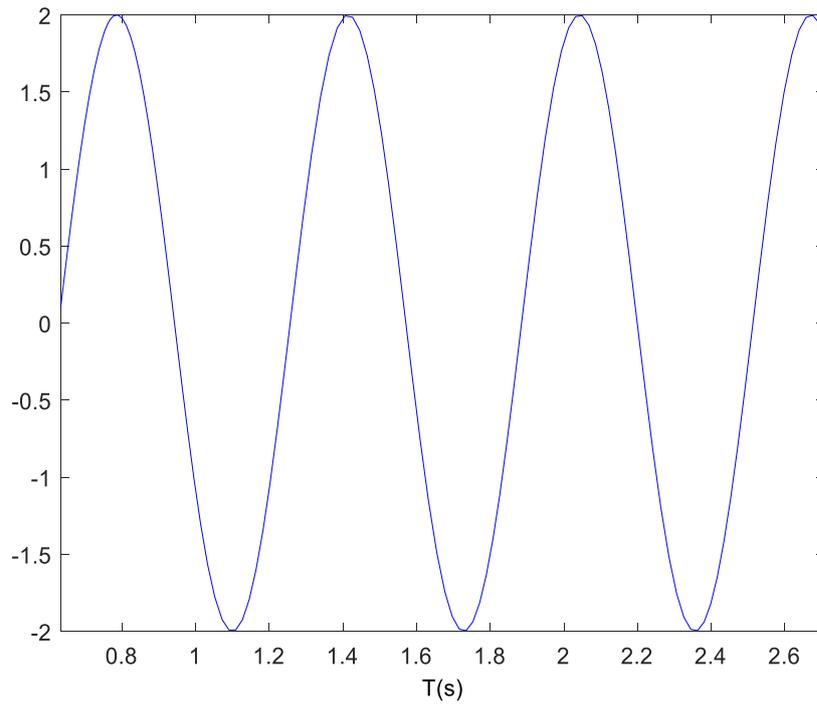


Figure 3.12: message reconstruit.

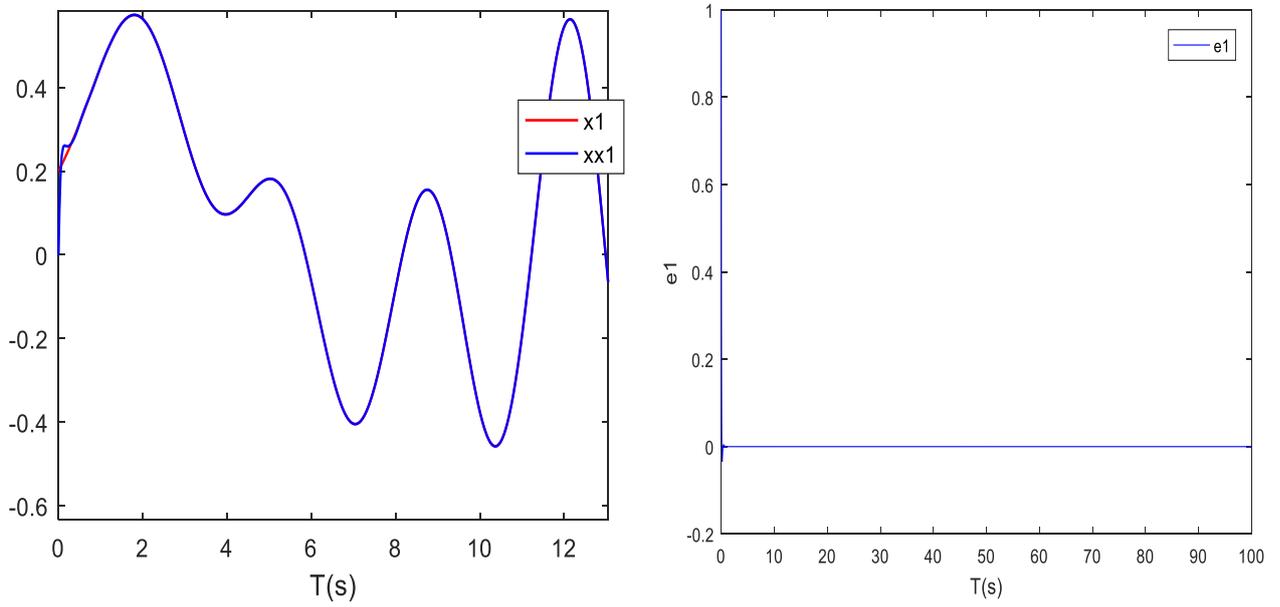


Figure 3.13: la réponse temporelle et l'erreur entre x_1 , \hat{x}_1

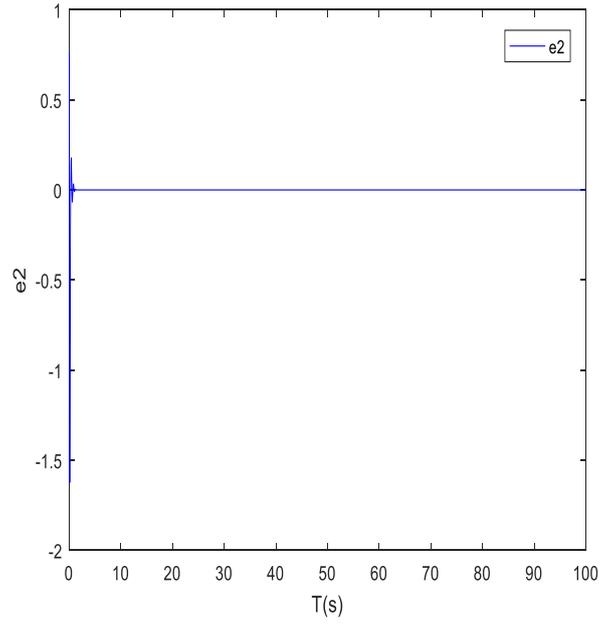
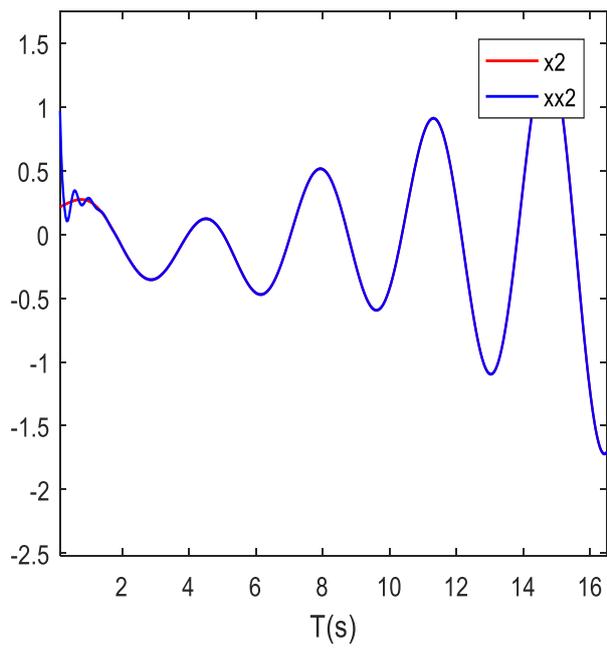


Figure 3.14: la réponse temporelle et l'erreur entre x_2, \hat{x}_2

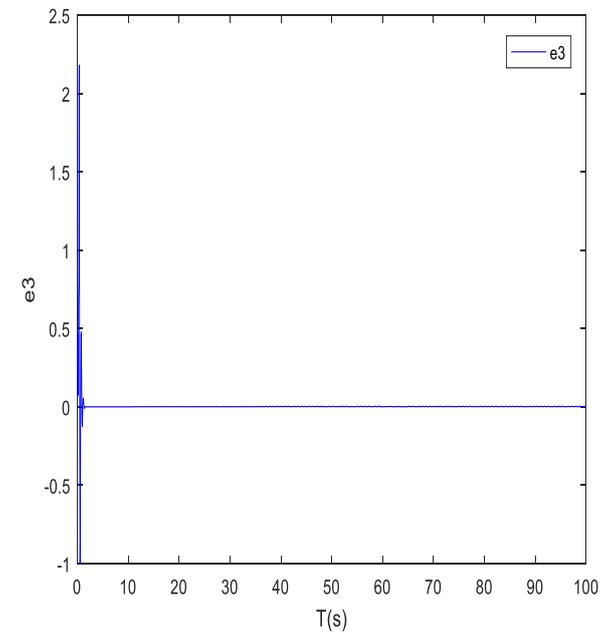
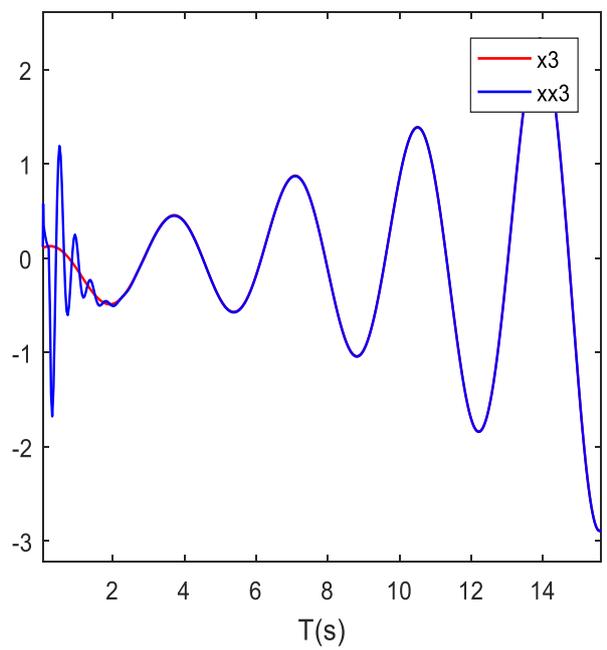


Figure 3.15: la réponse temporelle et l'erreur entre x_3, \hat{x}_3

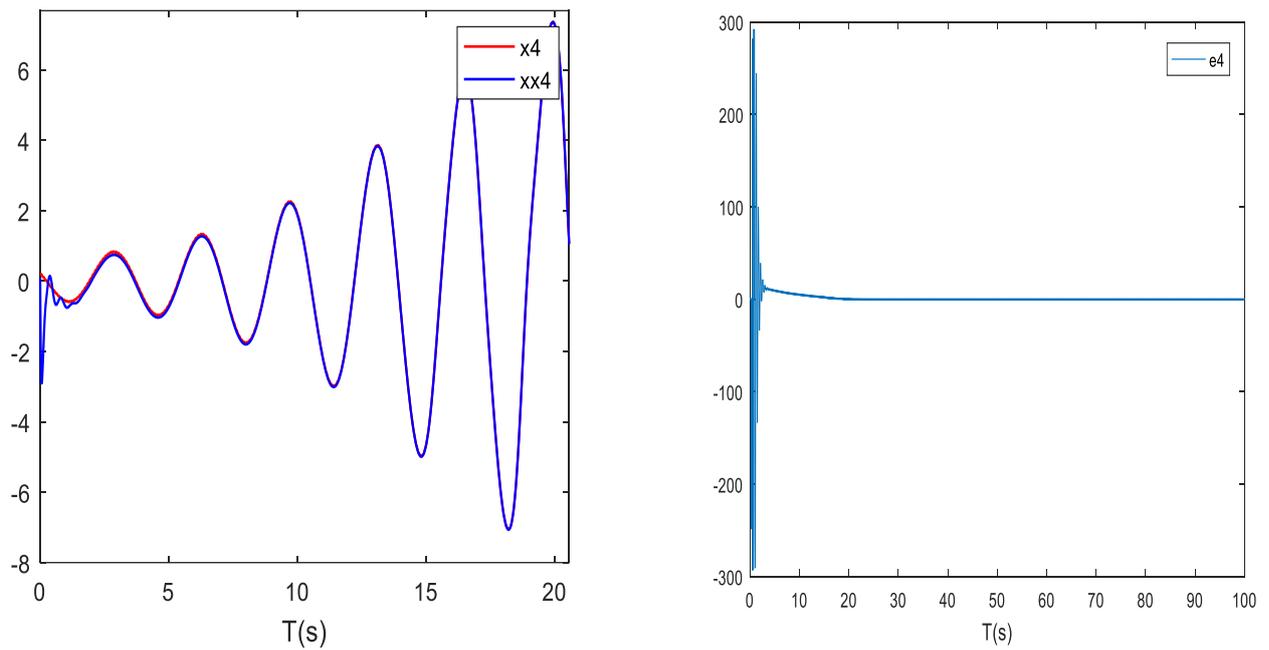


Figure 3.16: la réponse temporelle et l'erreur entre x_4 , \hat{x}_4

3.4.2 Cyptage par addition:

Au niveau de l'émetteur, Le message est ajouté directement par une simple addition à la quatrième dynamique du système maître.

La figure (3.9) représente l'émetteur chaotique après l'insertion de message sous Matlab :

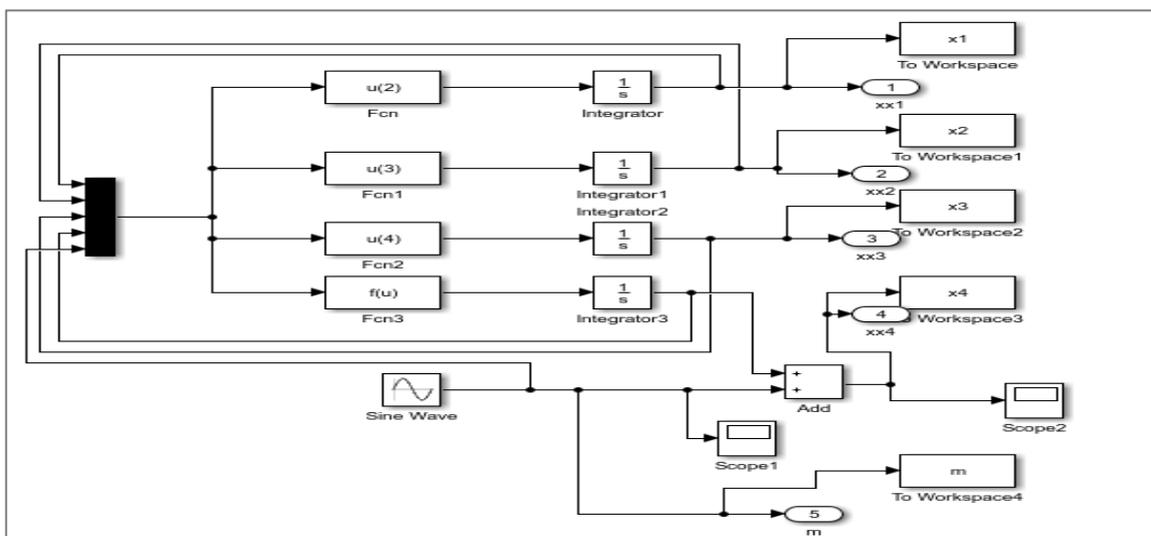
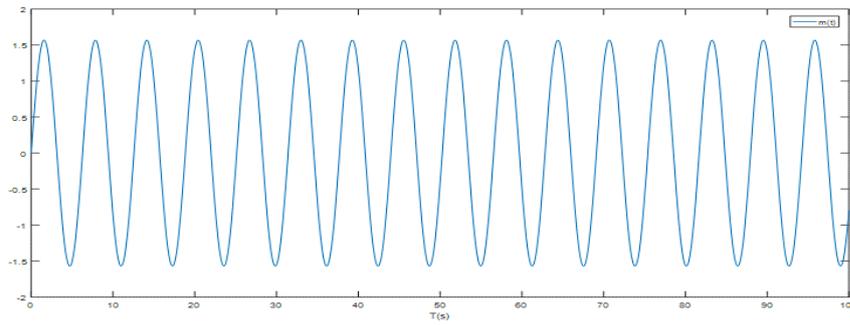
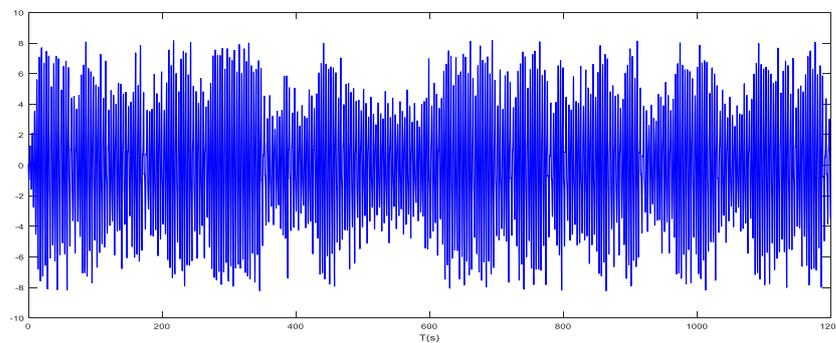


Figure 3.17: l'émetteur sous Matlab (Simulink) avec insertion de message par addition

La figure (3.10) représente le message émis et le message crypté.



a. Message émis



b. Message crypté

Figure 3.18: Message émis, message crypté

Au niveau du récepteur, pour récupérer le message on utilise le même observateur qui nous permet de reconstituer toutes les étapes et ensuite extraire le message par une simple fonction de soustraction :

$$m = x_4 - \hat{x}_4$$

Le schéma de la récupération de message est donné par la figure (3.19).

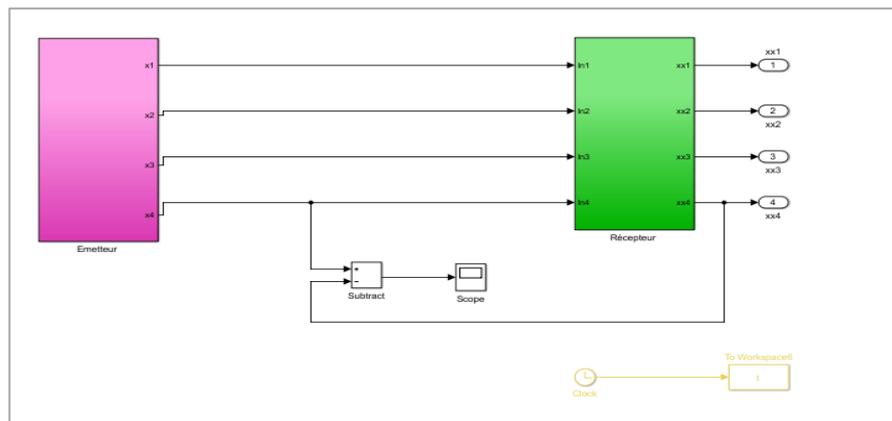


Figure 3.19: Schéma de récupération du message

Les figure (3.20), (3.21) et (3.22) montrent le message reconstruit, la réponse temporelle ainsi que l'erreur de synchronisation de la variable d'état x_4 .

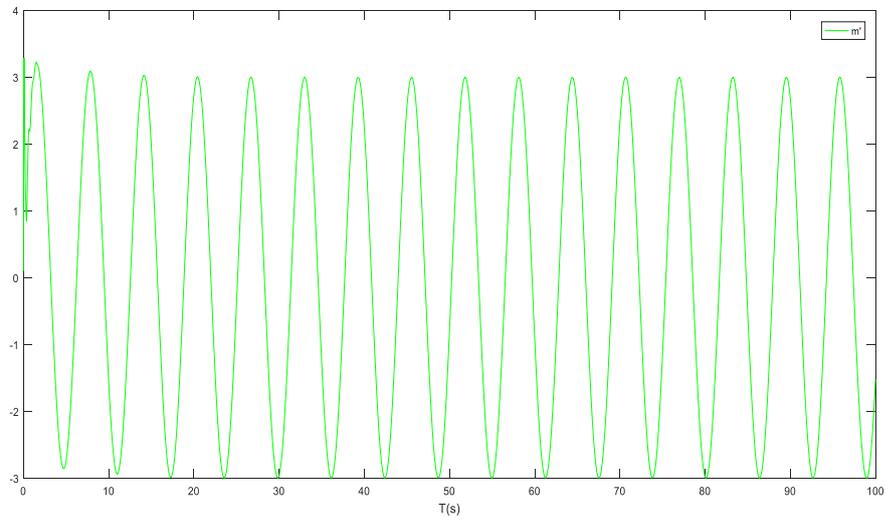


Figure 3.20: message reconstruit

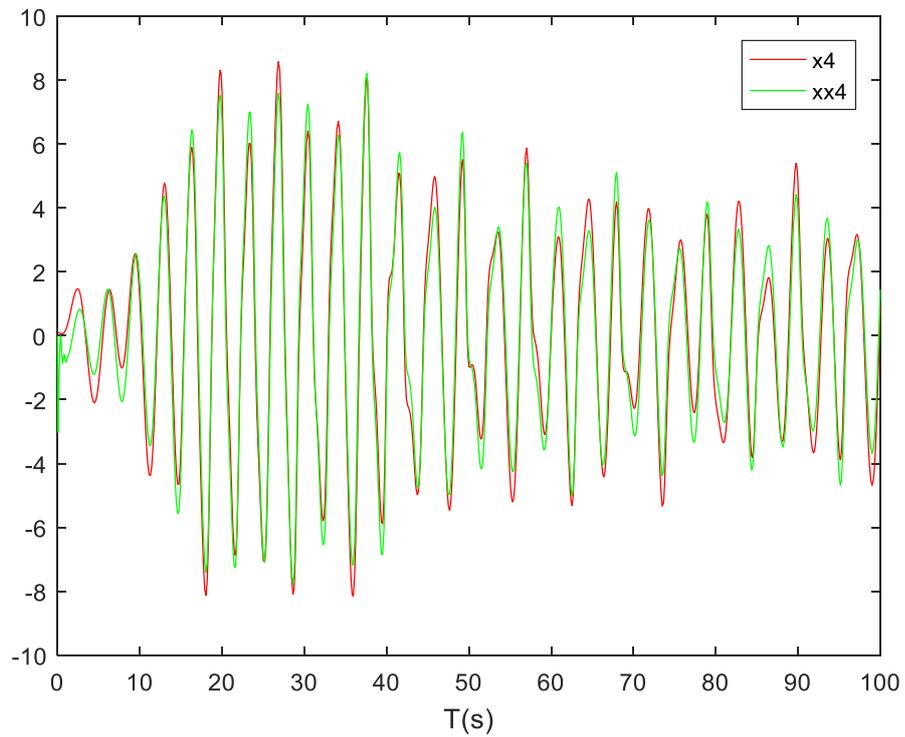


Figure 3.21: la réponse temporelle du x_4 , \hat{x}_4

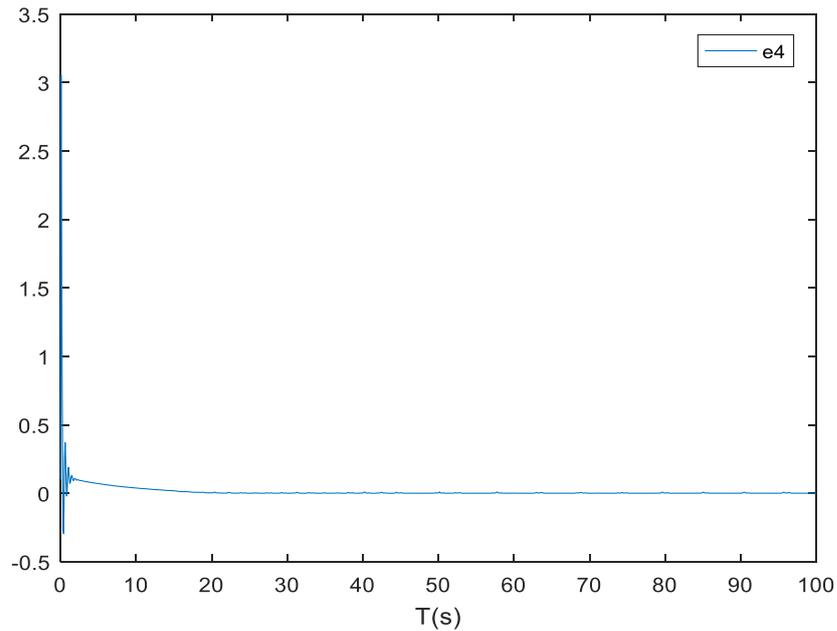


Figure 3.22: l'erreur de synchronisation e4.

Des résultats de simulation, le message a été récupéré avec les deux techniques de cryptage utilisées. Le message a été mieux récupéré avec la méthode d'inclusion car deux voies de transmission sont utilisées, une pour la synchronisation et l'autre pour la transmission de message. Donc cette méthode n'influence pas sur la synchronisation, Contrairement à la deuxième méthode dans lequel le message est injecté dans la dynamique du système puis transférer par une seule voie de transmission.

3.5 Conclusion

Dans ce chapitre nous avons décrit différentes techniques de cryptage d'un système chaotique. Nous avons conçu un système de transmission sécurisé en utilisant un système hyper chaotique dont le message est signal un sinusoïdale, qui a été injecté dans la dynamique de système par deux techniques de cryptage par addition et par inclusion. Puis nous l'avons récupéré à l'aide d'un observateur à mode glissant étape par étape. Les résultats de simulations ont montré que la synchronisation par l'observateur est réussie.

4.1 Introduction

De nos jours, l'électronique programmable a ouvert de nouvelles perspectives pour les systèmes embarqués dans de nombreux domaines industriels, tels que la sécurité, les télécommunications, etc. L'intégration des systèmes embarqués dans les produits industriels est devenue une des principales sources d'innovation [1]. Ces systèmes permettent de générer aisément des signaux issus de la discrétisation d'équation chaotique. Dans le cadre de notre projet nous avons opté pour une carte Arduino uno qui est un ensemble de matériel (carte électronique) et logiciel de programmation caractérisé par sa capacité et la facilité de programmation pour la mise en œuvre de l'émetteur.

4.2 Présentation de la carte Arduino Uno

La carte électronique Arduino Uno est un assemblage de composants sur une plateforme dite circuit imprimée, l'intérêt principal de cette carte est de faciliter la mise en œuvre. Elle est basée sur un microcontrôleur ATmega 328 de la famille AVR. Arduino fournit un environnement de développement s'appuyant sur des outils Open Source comme interface de programmation. Le changement du programme dans la mémoire du microcontrôleur se fait de façon très par port USB. Elle est dotée de 14 entrées/sorties digitales (dont 6 peuvent être utilisé comme sorties PWM), 6 entrées analogiques, un cristal à 16 MHz, une connexion USB, une prise jack d'alimentation et Possède un bouton de réinitialisation. La carte est illustrée sur la figure (4.1) si dessous [35]

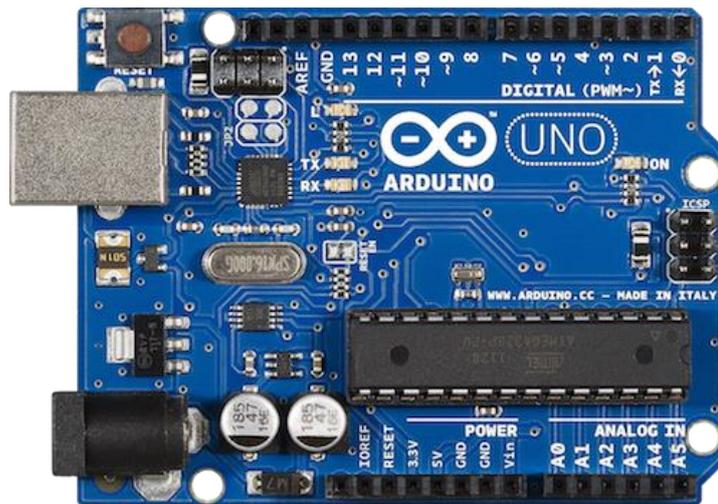


Figure 4.1: la carte Arduino Uno

4.3 La fiche technique de la carte Arduino Uno

Nous présentons dans cette section la fiche technique d'un Arduino Uno [36] :

- Microcontrôleur : ATmega328
- Tension de fonctionnement = 5V
- Tension d'alimentation recommandée= 7 à 12V
- Tension d'alimentation limites = 6 à 20v
- Entrées/ sorties numériques : 14 dont 6 sorties PWM
- Entrées analogiques = 6 (avec une résolution de 10 bits = 1024 valeurs différentes)
- Courant max sur les broches E/S= 40Ma
- Courant max sur la sortie 3.3V = 50MA
- Mémoire flash : 32KB (ATmega328p)
- SRAM : 2KB (ATmega328p)
- EEPROM : 1KB (ATmega328p)
- Vitesse d'horloge :16 MHz
- Longueur : 68.6mm
- Largeur :53.4mm
- Poids :25g

4.4 Le microcontrôleur ATmega328

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur.

- La mémoire : Il en possède de 5 types :

La mémoire Flash (32KB) : c'est celle qui contiendra le programme à exécuter.

RAM (2KB) : c'est la mémoire dite « vive », elle contient les variables de programme. Elle est dite « volatile » car elle s'efface en coupant l'alimentation du microcontrôleur.

EEPROM (1KB) : c'est le disque dur de microcontrôleur dans lequel on enregistre des informations qui ont besoin de survivre dans le temps. Même si la carte doit être arrêtée cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme.

Les registres : c'est un type de mémoire utilisé par le processeur.

Le mémoire cache : c'est une mémoire qui fait la liaison entre les registres et la RAM.

- Le processeur : est le composant principal du microcontrôleur. Il va exécuter le programme à traiter. On le nommant souvent « CPU » à une largeur du bus chemin de données allant de 4 bits pour les modèles les plus basiques à 32 ou 64 bits pour les modèles les plus évolués.

On distingue plusieurs cartes Arduino, dont les caractéristiques sont données par le tableau (4.1).

Nome	Processor	Operating Voltage/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [KB]	SRAM [KB]	Flash [KB]	USB	UART
Uno	ATmega328	5 V/7-12 V	16 Mhz	6/0	14/6	1	2	32	Regular	1
Due	AT91SAM3X8E	3.3 V/7-12 V	84 Mhz	12/2	54/12	-	96	512	2 Micro	4
Leonardo	ATmega32u4	5 V/7-12 V	16 Mhz	12/0	20/7	1	2.5	32	Micro	1
Mega 2560	ATmega2560	5 V/7-12 V	16 Mhz	16/0	54/15	4	8	256	Regular	4
Mega ADK	ATmega2560	5 V/7-12 V	16 Mhz	16/0	54/15	4	8	256	Regular	4
Micro	ATmega32u4	5 V/7-12 V	16 Mhz	12/0	20/7	1	2.5	32	Micro	1
Mini	ATmega328	5 V/7-9 V	16 Mhz	8/0	14/6	1	2	32	-	-
Nano	ATmega168	5 V/7-9 V	16 Mhz	8/0	14/6	0.512	1	16	Mini-B	1
	1					2	32			
Ethernet	ATmega328	5 V/7-12 V	16 Mhz	6/0	14/4	1	2	32	Regular	-
Explora	ATmega32u4	5 V/7-12 V	16 Mhz	-	-	1	2.5	32	Micro	-
ArduinoBT	ATmega328	5 V/2.5-12 V	16 Mhz	6/0	14/6	1	2	32	-	1
Fio	ATmega328P	3.3 V/3.7-7 V	8 Mhz	8/0	14/6	1	2	32	Mini	1
Pro (168)	ATmega168	3.3 V/3.35-12 V	8 Mhz	6/0	14/6	0.512	1	16	-	1
Pro (328)	ATmega328	5 V/5-12 V	16 Mhz	6/0	14/6	1	2	32	-	1
Pro Mini	ATmega168	3.3 V/3.35-12 V	8 Mhz	6/0	14/6	0.512	1	16	-	1
LilyPad	ATmega168V	2.7-5.5 V/2.7-5.5 V	8 Mhz	6/0	14/6	0.512	1	16	-	-
	ATmega328V									
LilyPad USB	ATmega32u4	3.3 V/3.8-5V	8 Mhz	4/0	9/4	1	2.5	32	Micro	-
LilyPad Simple	ATmega328	2.7-5.5 V/2.7-5.5 V	8 Mhz	4/0	9/4	1	2	32	-	-
LilyPad SimpleSnap	ATmega328	2.7-5.5 V/2.7-5.5 V	8 Mhz	4/0	9/4	1	2	32	-	-

Tableau 4.1 : caractéristiques de la carte Arduino

4.5 Programmation de la carte Arduino

La programmation de la carte Arduino se fait d'une façon simple par un logiciel appelé IDE, qui signifie « Integrated Development Environment ». IDE offre une multitude de fonctionnalités, et travaille avec un langage qui est proche du C et du C ++. [37]

1. Pour télécharger le logiciel il faut se rendre sur la page de téléchargement du site : <https://www.arduino.cc/en/Main/Software> . Décompresser le fichier avec un utilitaire de décompression (7-zip, WinRar...). A l'intérieur du dossier se trouvent quelques fichiers de l'exécutable de logiciel.
2. Installer le logiciel.
3. Dézipper le pilote FTDI USB Drivers.zip.
4. Brancher l'Arduino et pointer l'installateur Windows vers le pilote.
5. La carte est prête à accueillir un programme utilisateur.

A l'ouverture de logiciel, l'IDE affiche une fenêtre graphique qui contient un éditeur de texte et tous les outils nécessaires à l'activité de programmation. On peut donc saisir le programme, l'enregistrer, le compiler et le transférer sur la carte. La figure (4.2) montre l'écran initial qui apparaît au lancement de l'IDE.

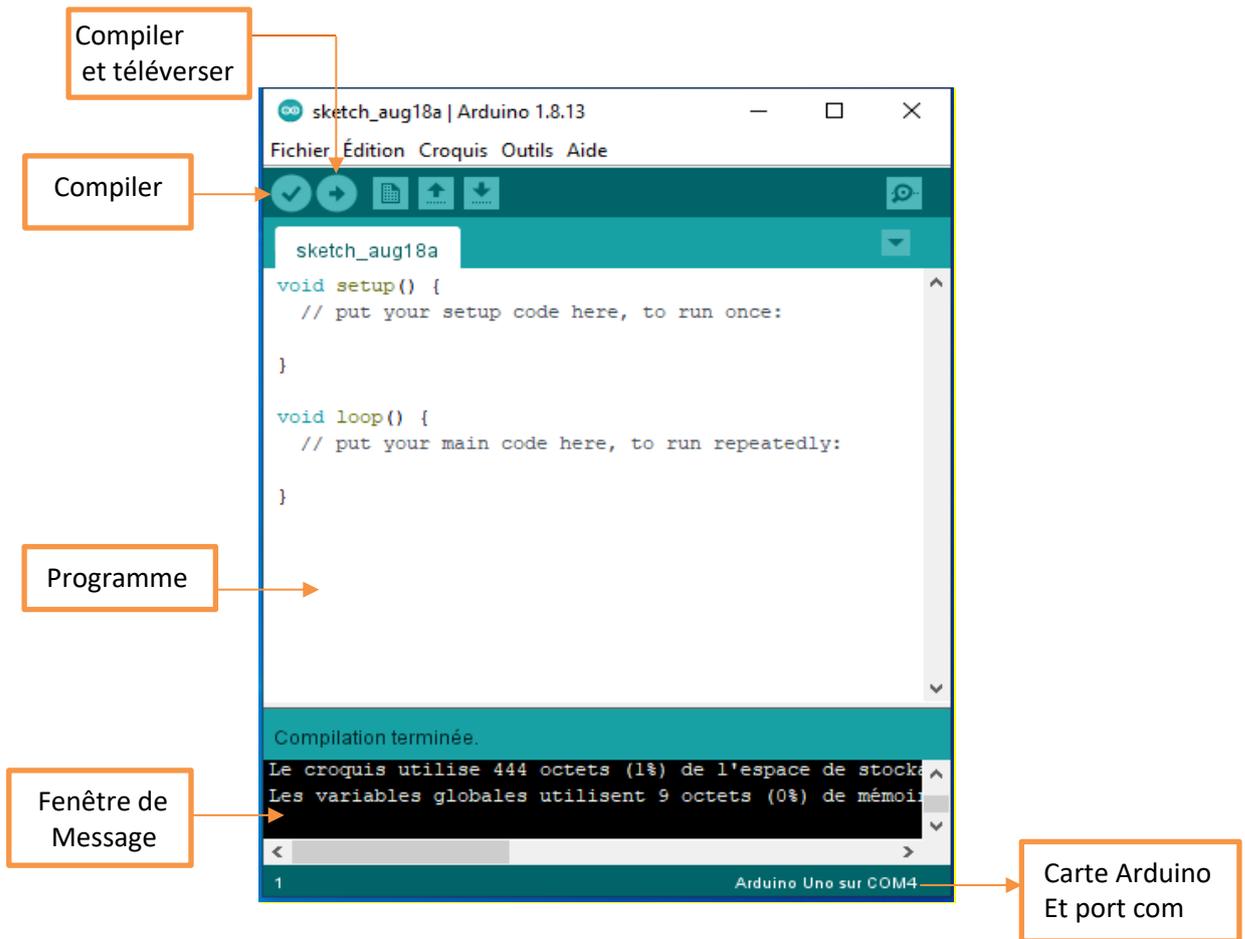


Figure 4.3 : interface de l'IDE Arduino

4.5.1 Structure d'un programme IDE

La structure d'un programme IDE à trois phases :

- 1- Les définitions des constantes et des variables.
- 2- La fonction setup : est une fonction d'initialisation.
- 3- La fonction Loop : est une fonction répétée indéfiniment, elle reboucle sur elle-même. Elle contient le programme à exécuter sur la carte Arduino.

4.5.2 Téléchargement du programme sur la carte

Pour injecter le programme dans la carte Arduino, il faudra d'abord choisir le type de la carte et le port USB sur le menu "outils", puis vérifier le programme en cliquant sur "compiler". S'il n'y a pas d'erreurs on envoi le programme vers

la carte en cliquant sur l'icône "téléverser" et le programme est injecté dans la carte.

4.6 Programmation de l'émetteur

Pour implémenter le système (1.4) sur la carte Arduino, il faut d'abord numériser le système, écrire un programme sur le logiciel IDE puis l'envoyer vers la carte et enfin transférer les données vers matlab pour visualiser les signaux.

4.6.1 Numérisation

Pour numériser le système, nous utilisons la méthode d'Euler qui est une méthode de résolution de façon approximative des équations différentielles ordinaires (EDO) de premier degré. Elle est due au mathématicien et physicien suisse Euler (1707-1783) [38].

a- Principe de la méthode d'Euler

La résolution du problème de Cauchy suivant :

$$\begin{cases} \dot{y} = f(x, y) \\ y(x_0) = y_0 \end{cases} \quad (4.1)$$

On va construire une courbe formée de segments qui vont démarrer du point de coordonnées (x_0, y_0) et dont le second point est obtenu en assimilant la courbe solution avec sa tangente au point (x_0, y_0) , ce qui est possible puisque l'on connaît la pente de la tangente :

$$\dot{y}(x_0) = f(x_0, y_0) \quad (4.2)$$

On obtient alors l'équation de la tangente en $M(x_0, y_0)$:

$$y = f(x_0, y_0)(x - x_0) + y_0 \quad (4.3)$$

On trouve alors, puisque $x_1 - x_0 = p$:

$$y_1 = f(x_0, y_0) \times p + y_0. \quad (4.4)$$

On recommence alors avec le point (x_1, y_1) obtenu sur la tangente avec $x_1 = x_0 + p$, où p est le pas choisi.

C'est une méthode itérative. La valeur y_{i+1} est déterminée en ajoutant Dy_i à la valeur y_i :

$$y_{i+1} = y_i + Dy_i = y_i + pf(x_i, y_i) \quad (4.5)$$

En utilisant cette approximation le système (1.4) devient :

$$\left\{ \begin{array}{l} x_1((k+1)p) = x_1(kp) + px_2(kp) \\ x_2((k+1)p) = x_2(kp) + px_3(kp) \\ x_3((k+1)p) = x_3(kp) + px_4(kp) \\ x_4((k+1)p) = x_4(kp) + p(-x_1(kp) - x_2(kp) - a \times x_3(kp) - (x_1(kp))^4 \times x_4(kp)) \end{array} \right. \quad (4.6)$$

4.6.2 Programmes :

Pour réaliser l'émetteur chaotique, le programme listé ci-dessous est proposé.

- Programme Arduino :

```
//conditions initiales
double x1=0.2;
double x2=0.2;
double x3=0.3;
double x4=0.1;
double a=3.6;
double p=0.001; // pas d'échantillonnage
double y1,y2,y3,y4;
void setup() {
  Serial.begin(9600);
}
void loop () {
  // les équations du système
  y1=x1+p*x2;
```

```

y2=x2+p*x3;
y3=x3+p*x4;
y4=x4-p(x1+x2+a*x3+pow(x1,4)*x4) ;
// état présent=etatfutur
x1=y1;
x2=y2;
x3=y3;
x4=y4;
Serial.println(x1);
Serial.println(x2);
Serial.println(x3);
Serial.println(x4);
}

```

Pour récupérer les données simulant l'émetteur, le programme listé ci-dessous, est utilisé.

- Programme matlab :

Clear all

Clc

x=[data] ;

```

n= length(x) ;
z=n/4;

```

```

x1(1)=x(1) ;
x2(1)=x(2) ;
x3(1)=x(3) ;
x4(1)=x(4) ;

```

```

for i=0:z-1

```

```

    x1(i+1)=x(4*i+1) ;
    x2(i+1)=x(4*i+2) ;
    x3(i+1)=x(4*i+3) ;
    x4(i+1)=x(4*i+4) ;

```

```

end

```

```

t=[1:z];
plot1(t,x1)

```

Une fois l'émetteur Arduino est programmé et les données sont transférer vers matlab, on obtient les résultats montrés sur les figures (4.4) (4.5) (4.6).

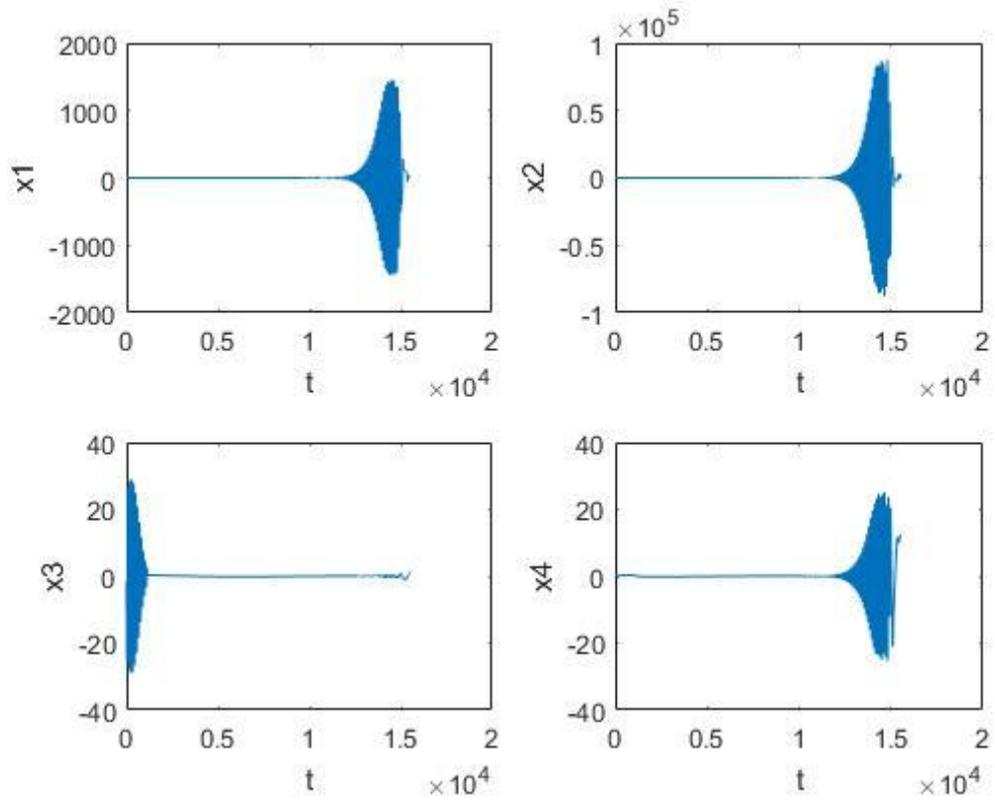


Figure 4.2: les états x_1, x_2, x_3 en fonction du temps

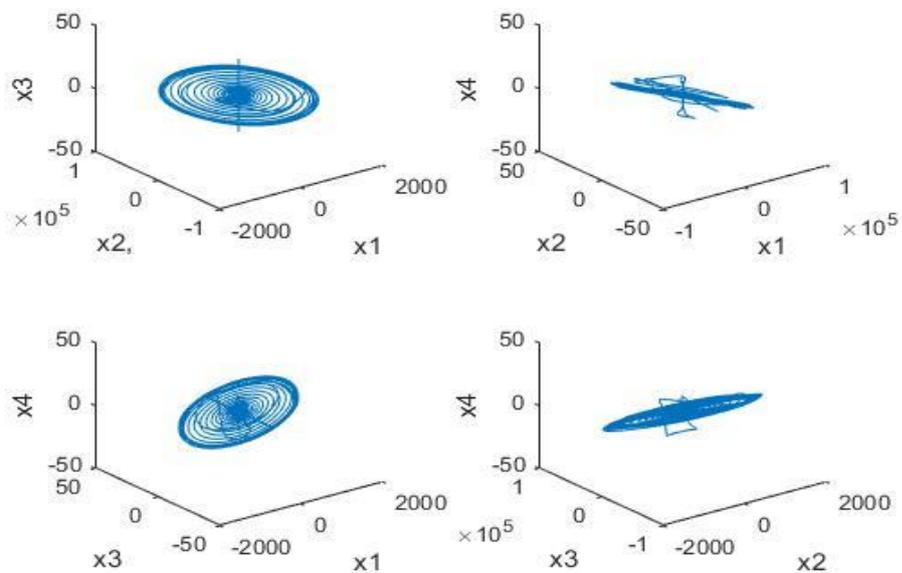


Figure 4.3: les attracteurs du système

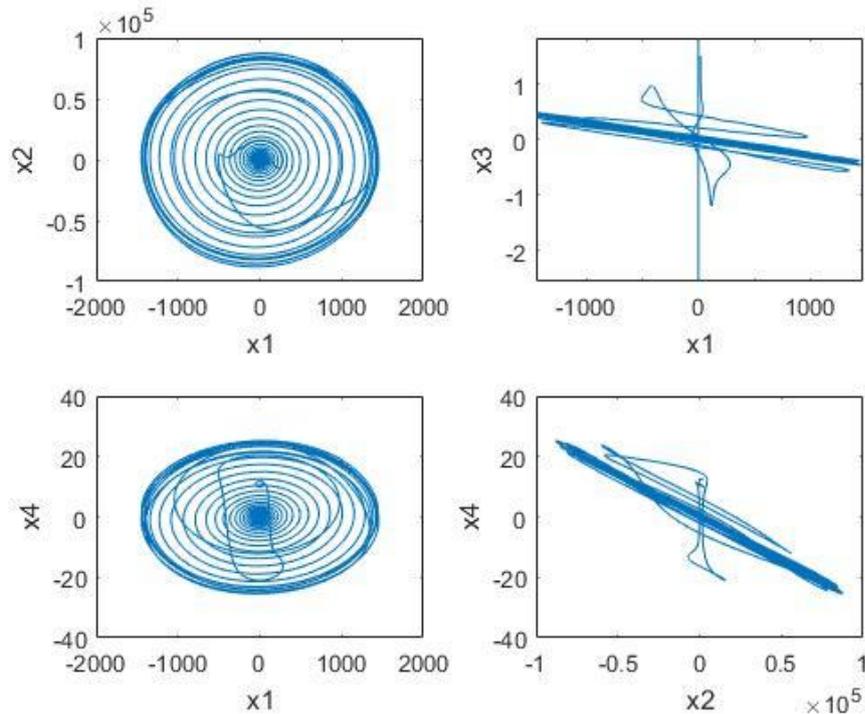


Figure 4.4: les espaces de phase du système

En visualisant les résultats, nous voyons que les états de système évoluent d'une façon qui semble aléatoire, ainsi les attracteurs résultants ce sont des attracteurs étranges, ses trajectoires sont apériodiques. Ce sont des propriétés qui caractérisent un système chaotique.

4.7 Conclusion

Dans de chapitre nous avons présenté la carte Arduino Uno avec ses différentes caractéristiques et son microcontrôleur. Ainsi nous avons vu comment ce fait la programmation de la carte Arduino. En fin nous avons programmé la carte Arduino en tant qu'émetteur hyper chaotique et visualiser les états, les espaces des phases et les attracteurs de l'émetteur.

Conclusion générale

L'objectif de ce mémoire est de réaliser un système de communication sécurisée basé sur le chaos, en utilisant un système hyper chaotique en tant qu'émetteur et un observateur mode glissant en tant que récepteur. L'information est introduite et cryptée au niveau de l'émetteur par deux techniques de transmission chaotique. Elle est reconstruite au niveau du récepteur en toute sécurité, en faisant synchroniser les deux systèmes.

Dans le premier chapitre nous avons présenté les caractéristiques des systèmes dynamiques chaotiques, ainsi que les outils permettant leurs études, en étudiant un système hyper chaotique. Le comportement du système est montré par la simulation sur matlab.

Dans le deuxième chapitre nous avons abordé la question de synchronisation, nous avons passé en revue quelques méthodes de synchronisation en basant sur la méthode par observateur, et en faisant l'étude de l'observateur choisi qui est l'observateur à mode glissant.

Dans le troisième chapitre, nous nous sommes intéressés aux différentes techniques de transmission par le chaos. Nous avons conçu notre système de transmission, deux techniques sont utilisées pour le cryptage de message à savoir le cryptage par addition et le cryptage par inclusion. Le message est ensuite transmis au récepteur. L'observateur mode glissant est à convergence étape par étape, permettant de le synchroniser avec l'émetteur. Nous avons montré la convergence en temps fini de l'observateur.

Dans le dernier chapitre nous avons réalisé l'implémentation de système émetteur, simulé précédemment sur le logiciel matlab/Simulink, sur une carte Arduino Uno et en utilisant la programmation grâce au logiciel IDE Arduino. Nous avons présenté les programmes utilisés avec le microcontrôleur et en montrant les procédures de l'implémentation de l'émetteur sur La carte.

Dans cette perspective, nous comptons réaliser l'implémentation de système de transmission en utilisant deux cartes Arduino afin de récupérer le message émis.

Bibliographie

- [1] H. Nijmeijer, M. Y. Ivan. Mareels: 'An observer looks at synchronization', IEEE transaction on circuits and systems: fundamental theory and application, vol.44, p:882-890, 1997.
- [2] Mr Chikhi Mohamed Lazhar : 'Application des systèmes dynamiques chaotiques en transmission de données'. Mémoire de magister, université Saad Dahleb Blida1, juin 2012.
- [3] Dalmedico, David Aubin and Amy Dahan: 'Writing the History of Dynamical Systems and Chaos: Longue Durée and Revolution', Disciplines and Cultures1. *Historia Mathematica* 29. 2002, Vol. 273-339.
- [4] Hamid Hamiche : 'Inversion à Gauche des Systèmes Dynamiques. Application à la Transmission Sécurisée de Données', *thèse de doctorat*, université Mouloud Mammeri de Tizi Ouzou, 2011.
- [5] Hamaizia Tayeb : ' Systèmes Dynamiques et Chaos Application à l'optimisation à l'aide d'algorithme chaotique', *thèse de doctorat*, université de Constantine-1-, 2013.
- [6] Hilborn Robert: '*Chaos and nonlinear dynamics. An introduction for scientists and engineers*. Oxford : Oxford University Press, 2000.
- [7] <http://guerillot.chez.com/physique/chaos/chaos4.htm> consulté le 10 mars 2020.
- [8] James Crutchfield, Doyne Farmer, Norman Packard et Robert Shaw : 'le chaos', pour la science, Vol. 26-39, année 1987.
- [9] Ibtissem Talbi : 'Systèmes dynamiques non linéaires et phénomènes de chaos', thèse de doctorat, Université Mentouri de Constantine, Juin 2010.
- [10] Dimassi Habib : 'Synchronisation des systèmes chaotiques par observateurs et applications à la transmission d'informations', thèse de doctorat, Université Paris Sud XI – Université Tunis El-Manar, novembre 2012.

- [11] Louis M. Pecora and Thomas L. Carroll: 'Synchronization in chaotic systems', 'Physical review letters', Code 6341, Naval Research Laboratory, Washington, D. C. 20375, volume 64, n: 8, 1990.
- [12] Zouad Fadia : 'Élaboration et implémentation de nouvelles approches pour la sécurisation des transmissions à base des systèmes chaotiques d'ordre fractionnaire', thèse de doctorat, Université Mohamed Seddik Ben Yahia-Jijel, juin 2019.
- [13] Mohammad Ali Khan et Swarup Poria: 'Generalized Synchronization of Bidirectionally Coupled Chaotic Systems', International Journal of Applied Mathematical Research, Science Publishing Corporation, p.303-313, 2012.
- [14] L. Kocarev et U.Parlitz: 'Generalized Synchronization, Predictability, and Equivalence of Unidirectionally Coupled Dynamical Systems', physical review letters, The American Physical Society , Vol.79, n.11, mars 1996.
- [15] Ma, Z., Liu, Z., Zhang, G.: 'Generalized synchronization of discrete systems', Appl.Math.Mech., Vol.28(5), p.609-614,2007.
- [16] Chuandong Li, Xiaofeng Liao et Kwok-wo Wong: ' Chaotic lag synchronization of coupled time-delayed systems and its applications in secure communication', Physica D 194, Elsevier B.V, Vol.194, p.187-202, 2004.
- [17] Adel Ouannas : 'sur la synchronisation des systèmes discrets', thèse de doctorat, Université des Frères Mentouri- Constantine, 2015.
- [18] C.P.silva,A.M.Young : 'introduction to chaos-based communications and signal processing', IEEE Aerospace Conference Proceedings(Cat.NO.00 TH 8484,2000: 279-299
- [19] R.Mainieri, J.Rehacek: 'Projective synchronization in three-dimensional chaotic', Physical Review Letters 1999: 82-3045.
- [20] J. Y. Chen, K. W. Wong, L. M. Cheng, et J. W. Shuai: 'A secure communication scheme based on the phase synchronization of chaotic systems', Chaos: An Interdisciplinary Journal of Nonlinear, Science American Institute of Physics, Vol.13, n.2, juin 2003.

- [21] Oukoum Ryma et BOUMESSID Hanane : 'Transmission sécurisée par modulation CSK (Chaos Shift Keying)', Mémoire de master, Université Saad dahleb-Blida,2015-2016.
- [22] Tian Yang : 'Une contribution à l'observation et à l'estimation des systèmes linéaires', Thèse de doctorat, école centrale de Lille, décembre 2010.
- [23] Marwa Mohamed Moustafa Ezzat : 'commande non linéaire sans capteur de la machine synchrone a aimants permanents', Thèse de doctorat, école centrale de Nantes, mai 2011.
- [24] Belgasem Jaballah : 'Observateurs robustes pour le diagnostic et la dynamique des véhicules', Thèse de doctorat, l'Université Paul Cézanne Aix-Marseille III- école Nationale d'Ingénieurs de Monastir, novembre 2011.
- [25] J.-J. E. Slotine, J. K. Hedrick et E. A. Misawa : 'nonlinear state estimation using sliding observers', IEEE conference on Decision and Control, 1986.
- [26] Ameer Aissa : 'Commande sans capteur de vitesse par dtc d'une machine synchrone à aimants permanents dotée d'un observateur d'ordre complet à modes glissants', mémoire de magister, université de Batna, octobre 2005.
- [27] Larbi Mohamed Elamine : 'Synthèse des Observateurs pour un Dorne Hélicoptère', Thèse de doctorat, université des sciences et de la technologie d'Oran Mohamed Boudiaf, mai 2015.
- [28] W. Perruquetti, J.P. Barbot: 'Sliding Mode Control in Engineering', Marcel Dekker.2002.
- [29] Megherbi Ouerdia : 'Etude et réalisation d'un système sécurisé à base de systèmes chaotiques', mémoire de magister, Université mouloud Mammeri tizi-ouzou,2013.
- [30] N. Corron, D. Hahs: 'A new approach to communication using chaotic signals', IEEE Transactions on Circuit and Systems, vol. 44, pp. 373–382, 1997.
- [31] H. Dedieu H, M.P. Kennedy and M. Hasler, New block Chaos shift keying: 'modulation and demodulation of a chaotic carrier using self-synchronizing Chua's

circuits, IEEE Transactions on Circuits and Systems', Circuits and Systems II: Analog and Digital Signal Processing, vol. 40, pp. 634–642, 1993.

[32] U. Feldmann, M. Hasler and W. Schwarz: 'Communications par chaotic signals: the inverse system approach', International Journal of Circuit: Theory and Applications, vol. 24, p. 551–579, 1996.

[33] M.P. Kennedy: 'Chaos in the Colpitts oscillator' IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Nov 1994: 771-774.

[34] M. L'Hernault, J.P. Barbot, A.Ouslimani: 'Feasibility of Analog Realization of a Sliding Mode Observer: Application to Data Transmission', IEEE Transactions on Circuits and Systems I: Regular Papers, March 2008: 614- 624.

[35] <http://www.acm.uiuc.edu/sigbot/tutorials/2009-11-17-arduino-basics> consulté le 25/08/2019.

[36] Gamal Abdelhamid : 'Système de supervision des temps d'arrêt des machines', mémoire de master, Université Sidi Mohamed Ben Abdellah Fès, 2016.

[37] Hanoune Katia : 'Etude, Simulation et implémentation d'un émetteur hyper chaotique sur une carte Arduino Uno', mémoire de master, Université Mouloud Mammeri Tizi-Ouzou, 2013/2014.

[38] <http://www.tangentex.com/MethodEuler.htm> consulté le 05/09/2020.

