République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DU **G**ENIE **E**LECTRIQUE ET D'**I**NFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Sciences et Technologies**
Filière : **Génie électrique**
Spécialité : **Commande des systèmes**

Présenté par
Amine KAKI
Farid KENAS

Thème

Conception et réalisation d'un système de transmission sécurisé de données basé sur le chaos en utilisant la carte Arduino Méga

Mémoire soutenu publiquement le 19/07/ 2016 devant le jury composé de :

M Ahmed MAIDI
Professeur, UMMTO, Président

M Redouane KARA
MCA, UMMTO, Encadreur

M Hamid HAMICHE
MCA, UMMTO, Examinateur

M Aldjia NAIT ABDESSELAM
MAA, UMMTO, Examinatrice

Remerciements

Remerciements

En premier lieu, nous tenons à remercier notre créateur ALLAH pour nous avoir donné la force et le courage pour accomplir ce travail.

Nous tenons à exprimer notre profonde gratitude et sincères remerciements à notre promoteur Monsieur *KARA Redouane* pour son encadrement, ses orientations ainsi que ses conseils combien précieux qui nous ont permis d'accomplir ce modeste travail dans bonnes conditions.

Nos remerciements vont également à Monsieur *HAMICHE Hamid* pour sa disponibilité et son aide précieuse.

Nous sommes aussi reconnaissants à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

Nous exprimons nos vifs remerciements à tous nos professeurs qui nous ont enseignés pendant notre cursus d'étude.

Et enfin nous tenons à remercier chaleureusement les membres du jury de nous faire l'honneur d'accepter d'évaluer ce projet.

Dédicaces

Dédicaces

Je dédie ce travail :

A mes chers parents qui ont toujours été à mes cotés pour me soutenir, m'encourager, me conseiller et veillez à mettre tout à ma disposition pour pouvoir étudier dans les meilleures conditions possibles.

A mes deux sœurs

A ma petite nièce

A tous mes amis qui sauront se reconnaitre

Ainsi à mon binôme Farid

Amine

Dédicaces

Je dédie ce travail :

A mes chers parents qui ont toujours été à mes cotés pour me soutenir, m'encourager, me conseiller et veillez à mettre tout à ma disposition pour pouvoir étudier dans les meilleures conditions possibles.

A mes frères et sœurs

A tous mes amis qui sauront se reconnaitre

Ainsi à mon binôme Amine

Farid

Sommaire

Sommaire

Sommaire

Sommaire

Sommaire

Liste des figures

Liste des figures

Liste des figures

Liste des figures

Liste des figures

Liste des tableaux

Liste des tableaux

Liste des tableaux

Liste des acronymes et abréviations

Liste des acronymes et abréviations

Liste des acronymes et abréviations

ASCII American Standard Code for Information Interchange

CAN Convertisseur Analogique Numérique

GPS Global Positioning System

ICSP In Circuit Serial Programming

LCD Liquid Crystal Display

LED Light Emitting Diode

LSB Least Significant Bit

IDE Integrated Development Environment

MP3 Mpeg Audio Layer 3

MSB Most Significant Bit

PWM Pulse-width modulation

UART Universal Asynchronous Receiver Transmitter

USB Universal Serial Bus

Introduction Générale

Introduction générale

Depuis bien longtemps, le chaos a toujours été associé à une incompréhension des choses, à l'impossibilité de formuler une quelconque loi organisatrice d'un phénomène apparemment inextricable.

La découverte de la dynamique chaotique des systèmes non linéaires remonte aux travaux d'Henri Poincaré sur la mécanique céleste et la mécanique statistique vers 1900 [1]. Ils ont alors suscité peu d'intérêt et sont tombés dans l'oubli.

Il a fallu attendre 1963 pour qu'Edward Lorenz, un météorologue du Massachusetts Institute of Technology, mette en évidence le caractère chaotique des conditions météorologiques [2]. C'est à partir de là que la théorie du chaos a suscité beaucoup d'intérêt dans divers domaines.

Grace aux propriétés des systèmes chaotiques, telles que leurs sensibilités aux conditions initiales et le fait qu'ils évoluent dans une large bande de fréquence, les systèmes chaotiques sont de bons candidats pour la cryptographie [3].

En effet la cryptographie joue un rôle important dans la sécurité et la fiabilité des systèmes de transmission de données [4]. Avec le développement du commerce électronique, les utilisateurs ont besoin de garantir la confidentialité des transactions sur des réseaux publics tels que l'internet.

Il est bien connu que les systèmes de communication traditionnels comportent deux parties, respectivement appelées émetteur et récepteur dont la synchronisation permettra de récupéré le message codé [4].

L'étape de synchronisation est fondamentale pour une transmission réussie ; introduite en 1990 par Pecora et Carroll [4], la synchronisation est une technique ayant pour principe de forcer la trajectoire du récepteur à suivre celle de l'émetteur [5].

Dans ce travail, nous proposons d'utiliser le chaos pour chiffrer une information dans une transmission sécurisée de donnée que nous allons concevoir théoriquement et réaliser sur une carte électronique Arduino. Pour ce faire, nous utiliserons un système chaotique appelé « système de Lozi » [2].

Ce mémoire est organisé de la manière suivante :

Introduction générale

Le chapitre I est consacré aux généralités et aux notions de base sur les systèmes chaotiques.

Le chapitre II est divisé en deux parties, la première partie traitera des généralités sur la synchronisation des systèmes chaotiques. Dans la deuxième partie, on étudiera les différentes méthodes de cryptage et de décryptage.

Le chapitre III est consacré à l'étude théorique du « système de Lozi », puis à la conception du récepteur et à l'insertion du message puis son extraction. A la fin le système de transmission sécurisée obtenu sera simulé sous Matlab.

Dans le chapitre IV, Les différentes étapes qui nous ont menées vers la réalisation pratique du système étudié sont présentées, ainsi que les différents résultats auquel on a abouti.

Enfin, nous terminerons par une conclusion générale.

Chapitre I

Généralités sur les systèmes chaotiques

I. 1 Introduction

Depuis bien longtemps, l'espèce humaine a cherché à comprendre le monde qui l'entoure. Au début les scientifiques ont pensé que l'univers était régi par des lois immuables ne laissant aucune place au hasard. En effet Isaac Newton [1642-1727] parle de déterminisme [1]. Selon cette théorie, tout semblait aussi être parfaitement prédictible. Le futur devenait prévisible : il suffisait de connaitre les conditions initiales.

Le déterminisme triomphant rencontre cependant ses limites au XIX siècle, laissant la place à la théorie des probabilités. Henri Poincaré [1854-1912] propose une approche plus modeste, plus qualitative, en avançant qu'il ne pouvait connaitre l'évolution et prédire le problème de trois corps de la mécanique céleste (exemple : terre, lune et soleil) et ceci malgré leur nature déterministe [2]. De l'impossibilité pratique de prédire le futur en fonction du présent ; le temps qu'il fera demain par exemple, est née la théorie du chaos. Elle sera développée tout au long du XX siècle.

La première visualisation du phénomène du chaos déterministe a été observée par Edward Lorenz [1917-2008], un météorologue du MIT (Massachusetts Institute of Technology) par pur hasard en 1961 [1], alors qu'il cherchait à déterminer des conditions météorologiques futures à partir de données initiales sur son ordinateur, il s'est rendu compte fortuitement qu'une variation minime d'un des paramètres de départ (de l'ordre d'un millième) entraînait des résultats très différents à la fin du calcul. Il a ainsi mis en évidence le phénomène de sensibilité aux conditions initiales. Et c'est lors de sa conférence de 1972 « Prédictibilité : le battement d'ailes d'un papillon au Brésil peut-il provoquer une tornade au Texas ? » que ce phénomène a trouvé sa métaphore emblématique : l'effet papillon [2].

Dans ce chapitre, nous présenterons les aspects généraux des systèmes chaotiques et leurs caractéristiques.

I. 2 Quelques notions et définitions de base

I. 2. 1 Définition d'un système [1]

Un système est un ensemble d'organes, assemblés pour concourir à un résultat. (Exemple : système mécanique : masse-ressort).

I. 2. 2 Système dynamique [2]

C'est un modèle mathématique d'un phénomène évoluant dans le temps, ce phénomène pouvant provenir de la physique, la mécanique, l'économie, la biologie, l'écologie, la chimie, etc.

Il est constitué d'un espace de phase et d'un espace d'état. Un système dynamique en temps continu est décrit par un ensemble d'équations différentielles, alors qu'en temps discret on parle d'équations aux différences finies.

I. 2. 3 Portrait de phase

Le portrait de phase d'un système dynamique est une représentation graphique de plusieurs trajectoires représentatives dans l'espace de phase.

Etant donné un système dynamique $\dot{x} = f(x,t)$, sans résoudre les équations, on peut toujours à un instant t donné, représenter graphiquement (à l'aide de flèches) le champ des $\dot{x}$. La lecture de cette représentation graphique sera très utile pour se faire une idée du comportement du système [1].

I. 2. 4 Système linéaire [3]

Un système est dit linéaire si la fonction qui le décrit vérifie les principes de proportionnalité et de superposition.

a/ Principe de proportionnalité

Si s(t) est la réponse à l'entrée e(t), alors λs(t) est la réponse à l'entrée λe(t).

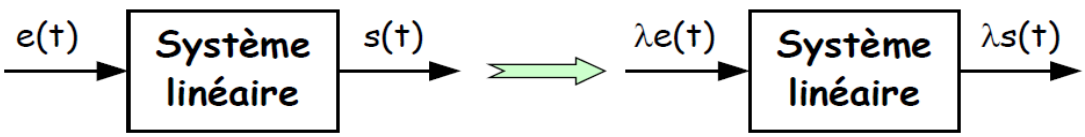


Figure (I.1) : Principe de proportionnalité

b/Principe de superposition

Si $s_1(t)$ est la réponse à l'entrée $e_1(t)$ et $s_2(t)$ est la réponse à l'entrée $e_2(t)$,

alors $[s_1(t) + s_2(t)]$ est la réponse à l'entrée $[e_1(t) + e_2(t)]$

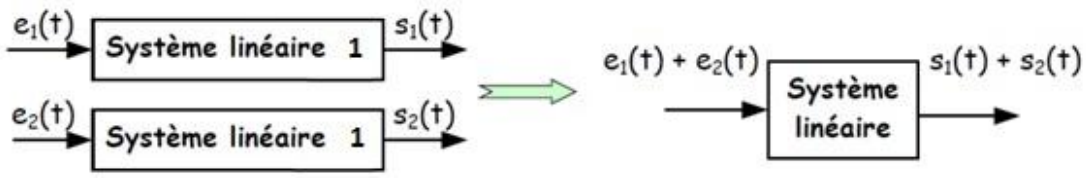


Figure (I.2) : Principe de superposition

Le système linéaire peut être représenté par un système d'équations de la forme suivante :

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t) \qquad \text{(I.1)}$$

Avec :

$x(t)$: Vecteur colonne de variable d'état de dimension n (n : dimension de l'espace d'état).

$y(t)$: Vecteur colonne des sorties du système de dimension p.

A : Matrice d'état ou matrice d'évolution de taille (n×n)

B : Matrice de commande de taille (n×m)

C : Matrice de sortie ou d'observation de taille (p×n)

D : Matrice de transmission de taille (p×m)

I. 2. 5 Système non linéaire [1]

La plupart des systèmes physiques ne sont pas linéaire. En Effet de nombreuses caractéristiques conduisent à des non linéarités.

Un système est dit non linéaire s'il ne respecte pas le principe de superposition et si la relation entre les grandeurs d'entrée et de sortie est une équation différentielle avec des coefficients non constants généralement. Il peut être représenté par un système de la forme suivante :

Equation d'état $\dot{x}\,(t) = f(x(t), u(t))$

Equation de sortie $y(t) = h(x(t), u(t))$ (I.2)

I. 2. 6 Système aléatoire

C'est un système qui évolue de manière probabiliste dans le temps, cela veut dire que la sortie ne peut pas être prédite, car il existe de multiples résultats possibles pour chaque entrée.

I. 2. 7 Système déterministe

Un système est dit déterministe lorsqu'il est possible de prédire (de calculer) son évolution au cours du temps : la connaissance exacte de l'état du système à un instant donné, l'instant initial, permet le calcul précis de l'état du système à n'importe quel autre moment [2].

I. 2. 8 Système autonome

Soit le système suivant :

$$\dot{x} = f(x, v) \quad \text{(I.3)}$$

Où : x est le vecteur d'état et v le vecteur des paramètres

Le système est dit autonome lorsque la fonction f ne dépend pas explicitement du temps [3].

I. 2. 9 Principe de causalité

En physique, un effet ne peut précéder sa cause. Un système est dit causal s'il respecte cette propriété. C'est-à-dire que si le signal d'entrée e(t) est nul pour $t<t_0$, il en est de même pour le signal de sortie $s(t)$ [4] :

$$e(t) = 0 \; \forall \; t \leq t_0 \quad ===> s(t) = 0 \; \forall \; t \leq t_0 \quad \text{(I.4)}$$

I. 2. 10 Exposant de Lyapunov [4]

Il permet de mesurer le taux de convergence ou de divergence de deux trajectoires initialement très proches l'une de l'autre. Il est souvent utilisé pour déterminer si un système est chaotique ou non.

$$Exposant\ de\ Lyapunov: \lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \ln|f'(x_{i-1})| \quad \text{(I. 5)}$$

I. 2. 11 Attracteur

La région de l'espace de phases vers laquelle convergent les trajectoires d'un système dynamique s'appelle '' attracteur ''. Les attracteurs sont des formes géométriques qui caractérisent l'évolution à long terme des systèmes dynamiques. Il en existe quatre types distincts : le point fixe, le cycle limite, l'attracteur quasi-périodique (Tore) et l'attracteur étrange [4].

I. 2. 12 Cycle limite [3]

Un cycle limite est un phénomène typiquement non linéaire. Tout système non linéaire qui a un siège d'oscillations est dit cycle limite, ce dernier est une trajectoire fermée et isolée, caractérisé par leur amplitude et leur fréquence indépendante de la condition initiale x_0 .

(Isolée : signifie que les trajectoires voisines ne sont pas fermées, elles tournent autour du cycle limite en s'en éloignant ou en s'en approchant).

I. 2. 13 Tore [4]

C'est un cas particulier de cycle limite, le système est caractérisé au moins par deux périodes simultanées dont le rapport est aléatoire, la trajectoire de phase ne s'annule pas sur elle-même.

I. 3 Le chaos

Le terme chaos définit un état particulier d'un système modélisé par des équations non linéaire dont le comportement ne se répète jamais et qui est très sensible aux conditions initiales, et imprédictible à long terme [5].

I. 4 Propriétés des systèmes chaotiques

I. 4. 1 Sensibilité aux conditions initiales

C'est une des propriétés essentielles du chaos. En effet, pour un système chaotique, s'il y a une petite modification des conditions initiales, les réponses restent proches pendant un certain moment puis à partir d'un instant, les réponses deviennent complètement différentes.

Pour illustrer cette propriété, on prend l'exemple du système de Rössler [5]

$$\begin{cases} \dot{x} = -(y+z) \\ \dot{y} = x + ay \\ \dot{z} = b + z(x-c) \end{cases} \quad \text{(I.6)}$$

Pour deux conditions initiales très proches :

$(x_0, y_0, z_0) = (1\ ;\ 1\ ;\ 1)\ et\ (x_0, y_0, z_0) = (1\ ;\ 1.000001\ ;\ 1)$

La simulation sur Matlab a donné les trajectoires de la figure suivante :

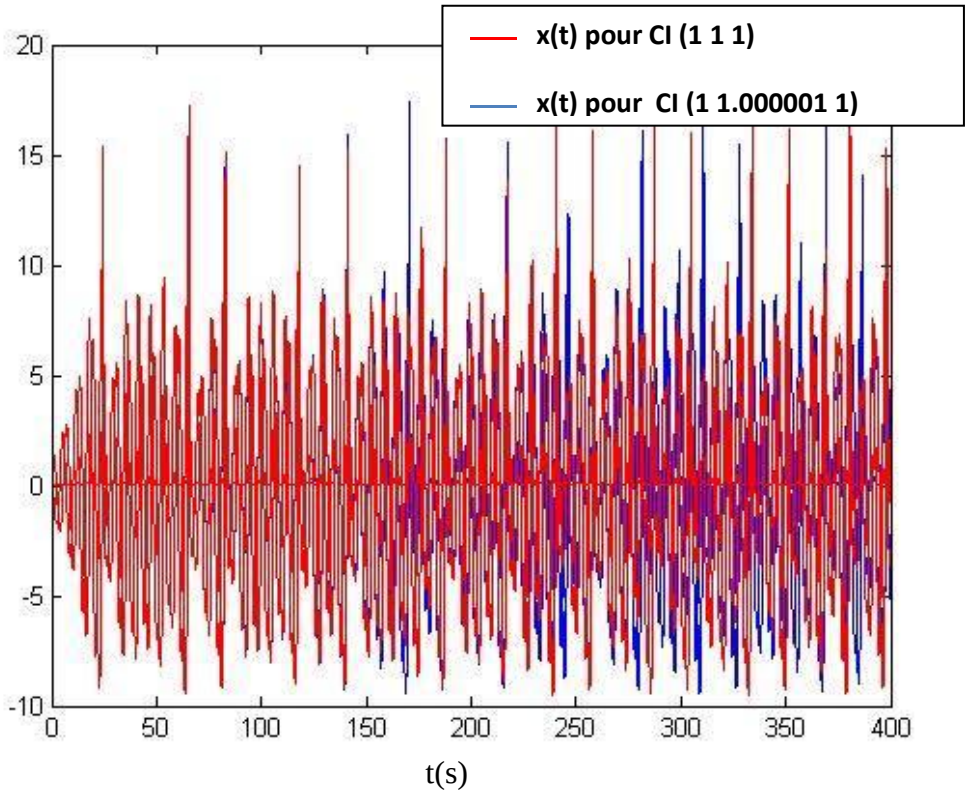


Figure (I.3) : Sensibilité aux conditions initiales pour l'état $\boldsymbol{y(t)}$.

I. 4. 2 Aspect aléatoire

En plus de la sensibilité aux conditions initiales, une autre caractéristique des systèmes chaotiques est que le comportement est imprévisible et possède un aspect aléatoire. Celui-ci correspond à une évolution complexe, non périodique et non prédictible [6].

La figure suivante illustre le caractère de l'évolution de l'une des composantes du système de Lorenz par rapport au temps.

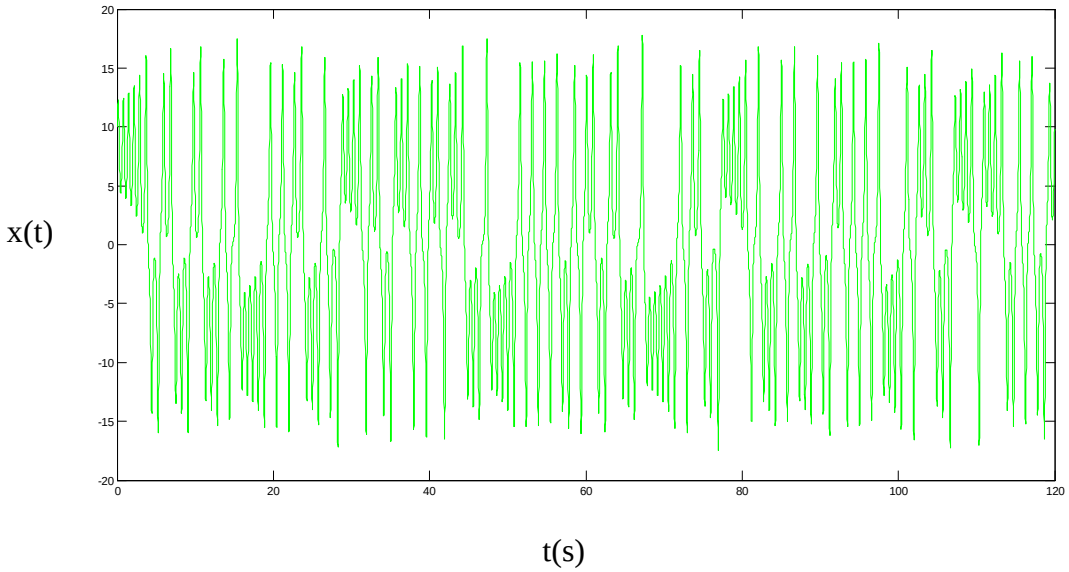


Figure (I.4) : Aspect aléatoire du système de Lorenz

I. 4. 3 Attracteur étrange

La découverte des attracteurs étrange dans les années 1960 a permis de représenter et d'analyser plus facilement des systèmes complexes comme les systèmes chaotiques. Le développement des ordinateurs a permis la représentation de ces objets mathématiques abstraits. Ils sont formés d'une suite de points : $x_0, x_1, x_2, \dots, x_n$, qui dépendent de la valeur initiale x_0 [6].

La forme d'un attracteur étrange n'est ni une courbe, ni une surface, ni continue, elle est constituée d'un ensemble dense de points avec des espaces inoccupés entre eux [7].Il s'agit d'un ensemble fractals.

La figure suivante illustre l'attracteur étrange de Lorenz :

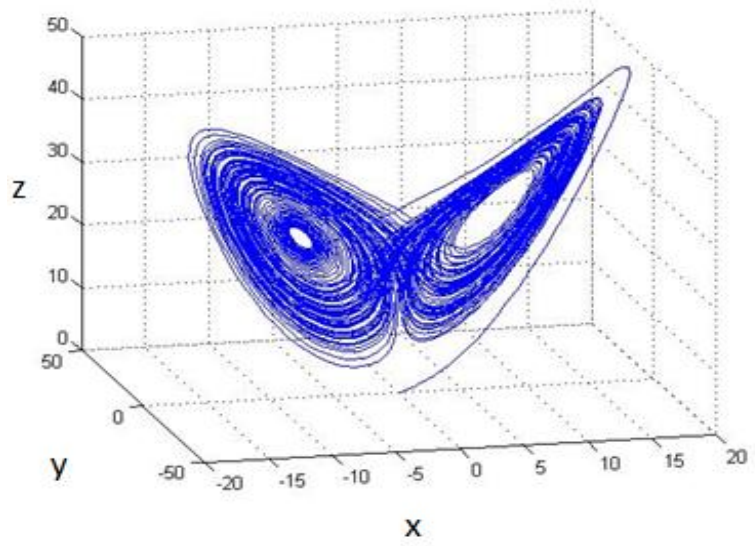


Figure (I.5) : Représentation des états x,y et z du modèle de Lorenz dans l'espace de phase.

I. 4. 4 Le Spectre de puissance

Une façon simple de caractériser le chaos consiste à calculer le spectre de puissance de l'évolution temporelle d'une des variables du système.

Nous avons calculé sous Matlab le spectre de puissance du modèle de Lorenz et il est donné par la figure (I.4).

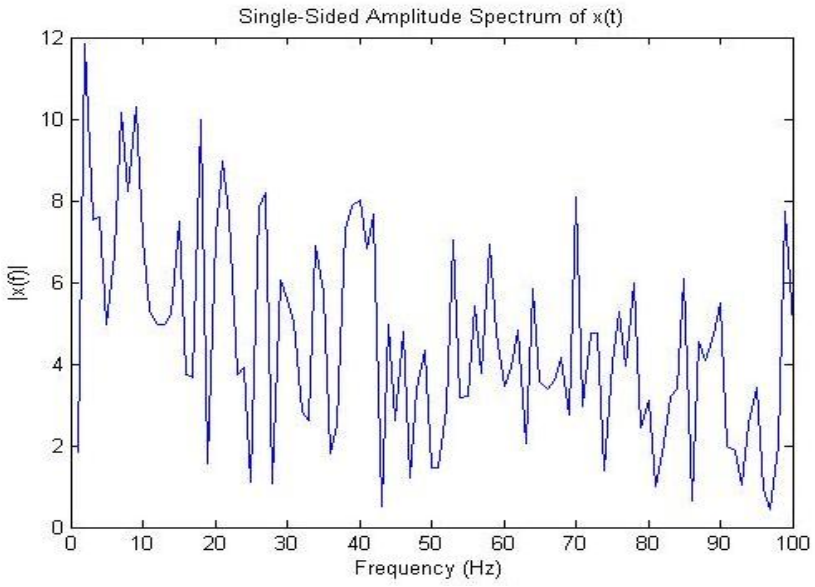


Figure (I.6) : Spectre de puissance du modèle de Lorenz

Remarque

Le spectre est continu et a une large bande de fréquence.

I. 4. 5 Exposants de Lyapunov

L'évolution d'un flot chaotique (annexe A) est d'autant plus difficile à saisir que la divergence des trajectoires sur l'attracteur est rapide. C'est pourquoi on essaie d'estimer ou même de mesurer la vitesse de divergence ou de convergence. On utilise pour cela un paramètre qui s'appelle exposant de Lyapunov [8].

Pour qu'un système ait une dynamique chaotique, trois conditions sont nécessaires :

- Au moins un exposant de Lyapunov est positif qui fait diverger la trajectoire.
- Au moins un exposant de Lyapunov est négatif qui fait replier la trajectoire.
- La somme des exposants doit être négative pour un système dissipatif (système qui n'emmagasine pas de l'énergie).

Un système discret chaotique possède au moins un exposant de Lyapunov positif [5].

Etat	Attracteur	Spectre	Exposant de Lyapunov
Point d'équilibre	Point	0	$\lambda_n \leq \cdots \leq \lambda_1 \leq 0$
Périodique	Cycle limite	1 raie	$\lambda_1 = 0$ $\lambda_n \leq \cdots \leq \lambda_2 \leq 0$
Période d'ordre 2	Tore	2 raies	$\lambda_1 = \lambda_2 = 0$ $\lambda_n \leq \cdots \leq \lambda_3 \leq 0$
Chaotique	Etrange	Spectre large	$\lambda_1 > 0$ $\sum_{i=1}^{n} \lambda_i < 0$
Hyper Chaotique	Etrange	Spectre large	$\lambda_1 > 0$ $\lambda_2 > 0$ $\sum_{i=1}^{n} \lambda_i < 0$

Tableau (I.1) : Classification des régimes permanents en fonction du spectre de Lyapunov.

I. 4. 6 Le Diagramme de Bifurcation

La bifurcation signifie un changement qualitatif de la dynamique du système, qui résulte du changement d'un des paramètres du système.

Par exemple ; déstabilisation d'un équilibre stable, apparition ou disparition d'un cycle ou d'un attracteur, un point d'équilibre dans le plan de phase devient un cycle limite.

La valeur pour laquelle la bifurcation se produit est nommée le point de bifurcation [9].

Le diagramme de bifurcation est un outil efficace pour évaluer rapidement l'ensemble des solutions possibles d'un système en fonction des variations de l'un de ses paramètres. Il permet de repérer les valeurs particulières du paramètre qui induisent des bifurcations [10].

I. 5 Les Routes vers le chaos [11]

Il est reconnu qu'il existe différentes routes par lesquelles un système devient chaotique. En générale, elles résultent de différentes bifurcations. Cela étant dit, « rien ne permet d'énoncer avec précision sous quelles conditions nécessaires et/ou suffisantes ces routes prennent place »

Les routes vers le chaos peuvent être les suivantes:

1. Le doublement de période :

En faisant varier (en augmentant) l'un des paramètres d'un système qui a un mouvement périodique fondamental, le processus arrive à une bifurcation où l'évolution du processus double la période jusqu'à l'apparition du chaos.

2. Intermittence :

Cette route se caractérise par l'apparition d'un comportement irrégulier dans un comportement régulier, c'est-à-dire ; un système dont l'évolution est périodique et stable mais entrecoupé (perturbé) par des phases chaotiques.

A un moment donné ces irrégularités prennent le dessus et le système devient chaotique.

3. Quasi-périodicité :

Cette route vers le chaos fait intervenir pour un système périodique l'apparition d'une autre période dont le rapport avec la première n'est pas rationnelle.

-La figure qui suit illustre le diagramme de bifurcation de la suite logistique suivante

$$x_{n+1} = a x_n (1 - x_n) \qquad \text{(I.7)}$$

Avec : $x_n > 0 \quad , n = 0,1,2 \ldots$

$$0 < a < 4$$

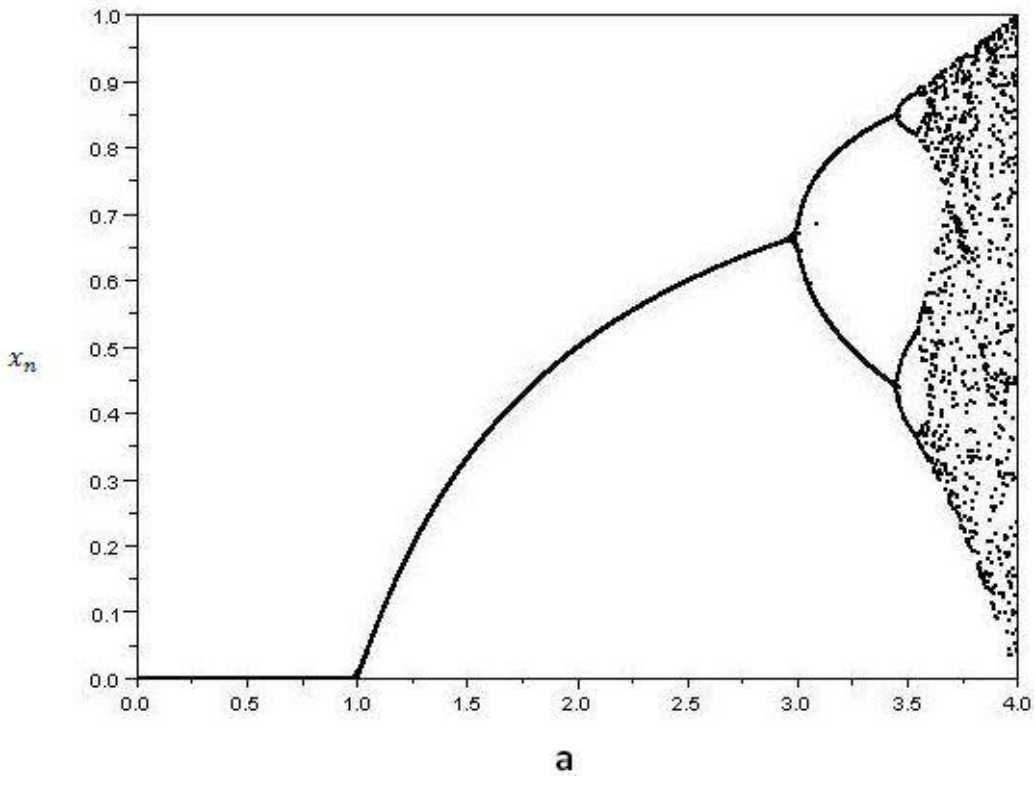


Figure I.7 : Diagramme de bifurcation de la suite logistique

I. 6 Etude de quelques systèmes chaotiques

Dans cette partie, nous exposons les exemples les plus connus et les plus étudiés des systèmes chaotiques.

I. 6. 1 Systèmes chaotiques à temps continu

Le système de Lorenz est un exemple célèbre de systèmes différentiels au comportement chaotique pour certaines valeurs de paramètres.

Le modèle implique trois équations différentielles [12]

$$\begin{cases} \dot{x}(t) = a(y - x) \\ \dot{y}(t) = -xz + bx - y \\ \dot{z}(t) = xy - cz \end{cases} \quad \text{(I.8)}$$

Avec: (x, y, z) vecteur d'état

a =10

c=8/3

b : est un paramètre de contrôle

On fait varier le paramètre b dans le but d'avoir une dynamique chaotique, ainsi pour différentes valeurs de b le comportement du système de Lorenz est représenté sur les figures qui suivent :

-Les conditions initiales sont fixées à [0.02 ; 0.02 ; 0.02]

-Pour b=12

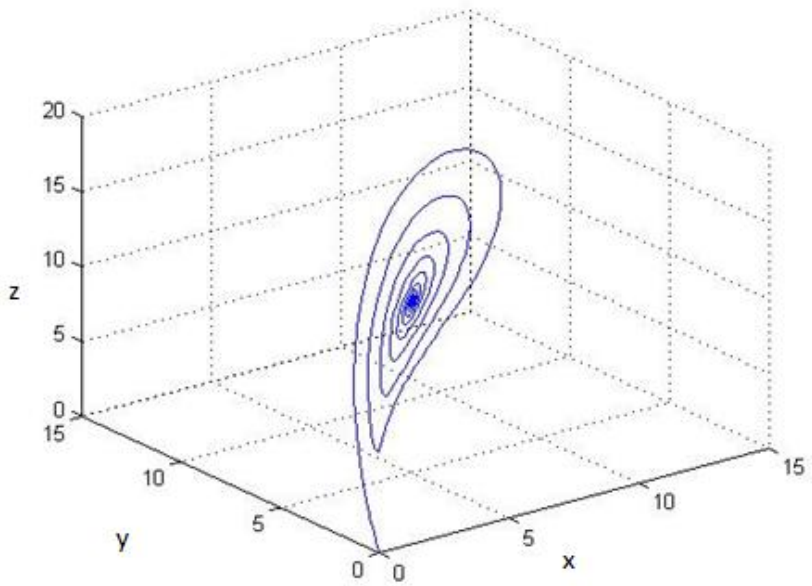


Figure (I.8) : Point fixe

-La trajectoire converge vers un point d'équilibre, dans ce cas, on a pas de dynamique chaotique.

-Pour b= 28

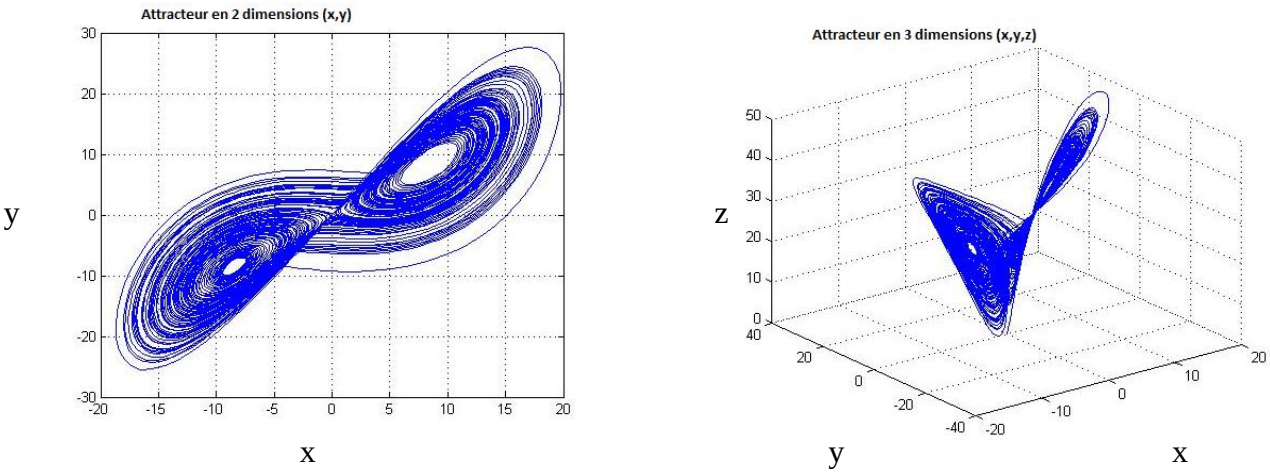


Figure (I.9) : Attracteur étrange (effet papillon)

-On observe que le portrait de phase des trajectoires ressemble à un papillon, c'est un attracteur étrange, qui ''attire'' la trajectoire.

-La dynamique du système est chaotique.

-Pour b= 180

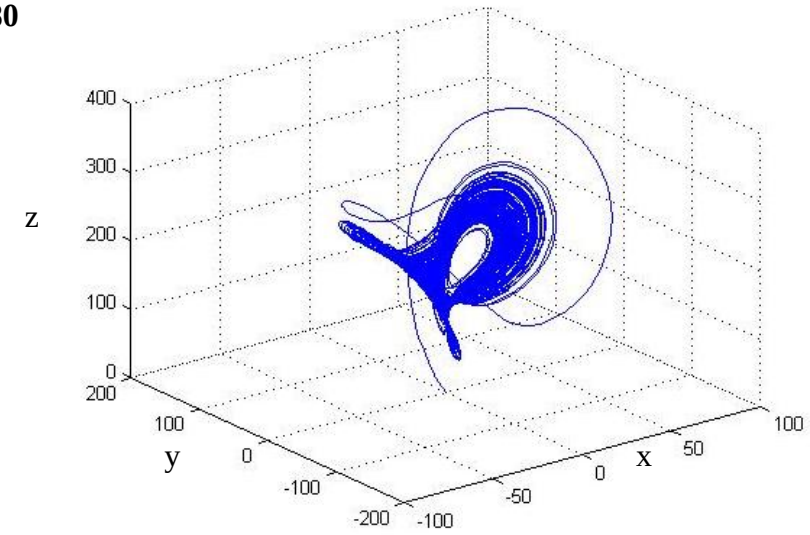


Figure (I.10) : Tore

-La trajectoire converge vers un Tore

-La figure (I.9) illustre la propriété de sensibilité aux conditions initiales avec une différence de l'ordre de σ= 10 $^{-6}$

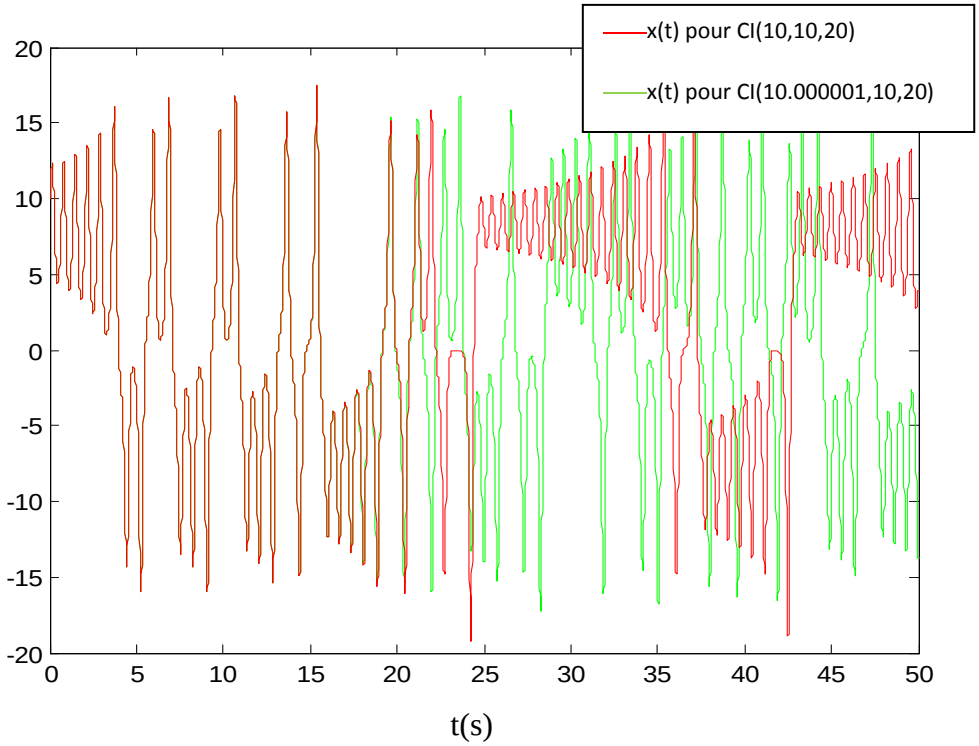


Figure (I.11) : Sensibilité aux conditions initiales pour l'état $\boldsymbol{x(t)}$du système de Lorenz.

I. 6. 2 Systèmes chaotiques à temps discret

Le système de Hénon est un modèle proposé en 1976 par le mathématicien Michel Hénon. Le modèle d'état associé est [13]

$$\begin{cases} x_{k+1} = a - x_k^2 + by_k \\ y_{k+1} = x_k \end{cases} \quad \text{(I.9)}$$

Avec : (x, y) vecteur d'état

$a\ et\ b$: Paramètres du système

k : Le nombre d'itérations

Les valeurs des paramètres proposées par Hénon pour observer le phénomène chaotique sont : a=1.4 et b=0.3

Pour simuler l'attracteur de Hénon on a pris pour conditions initiales [0.1 ; 0.1]

Ainsi les figures suivantes représentent respectivement les trajectoires de x, y et l'attracteur de Hénon.

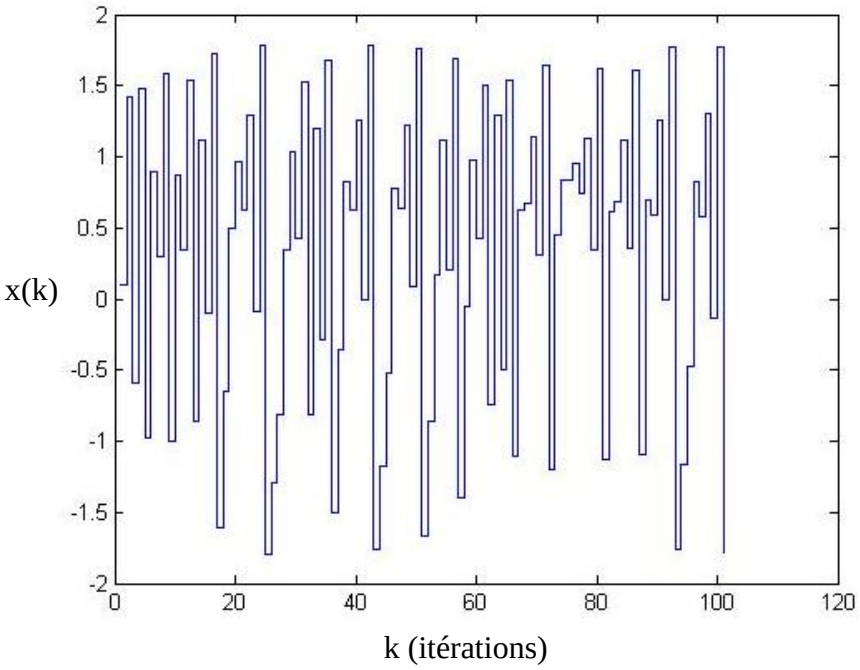


Figure (I.12) : Evolution aléatoire de x(k) du système de Hénon.

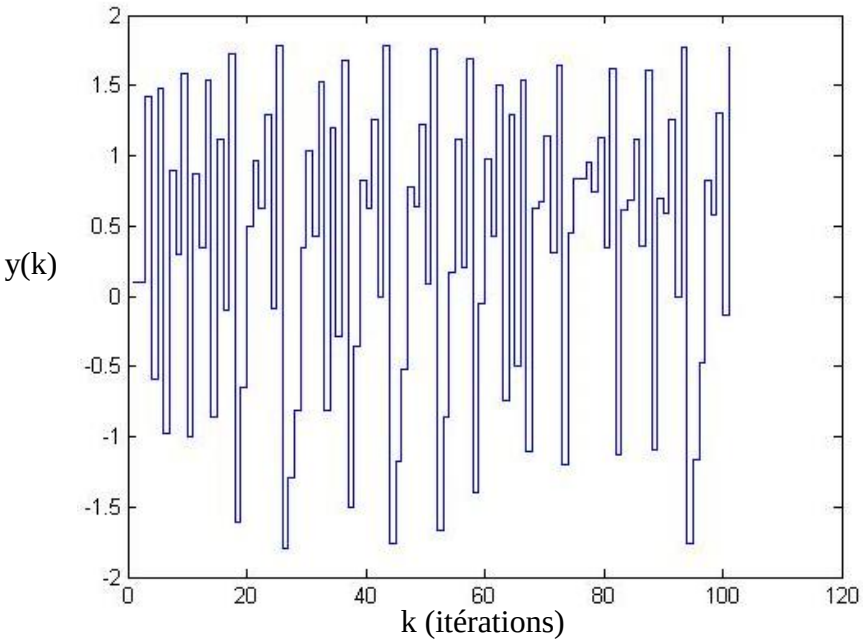


Figure (I.13) : Evolution aléatoire de y(k) du système de Hénon.

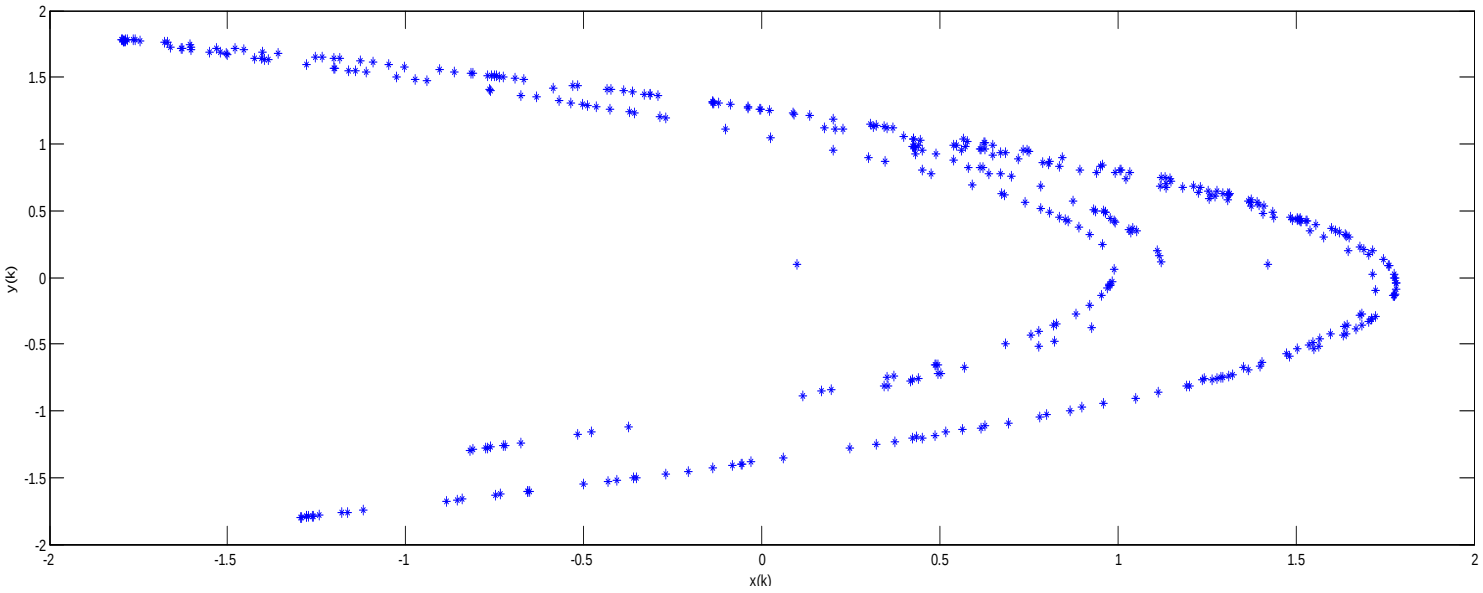


Figure (I.14) : Attracteur de Hénon

-La figure (I.13) illustre la propriété de sensibilité aux conditions initiales.

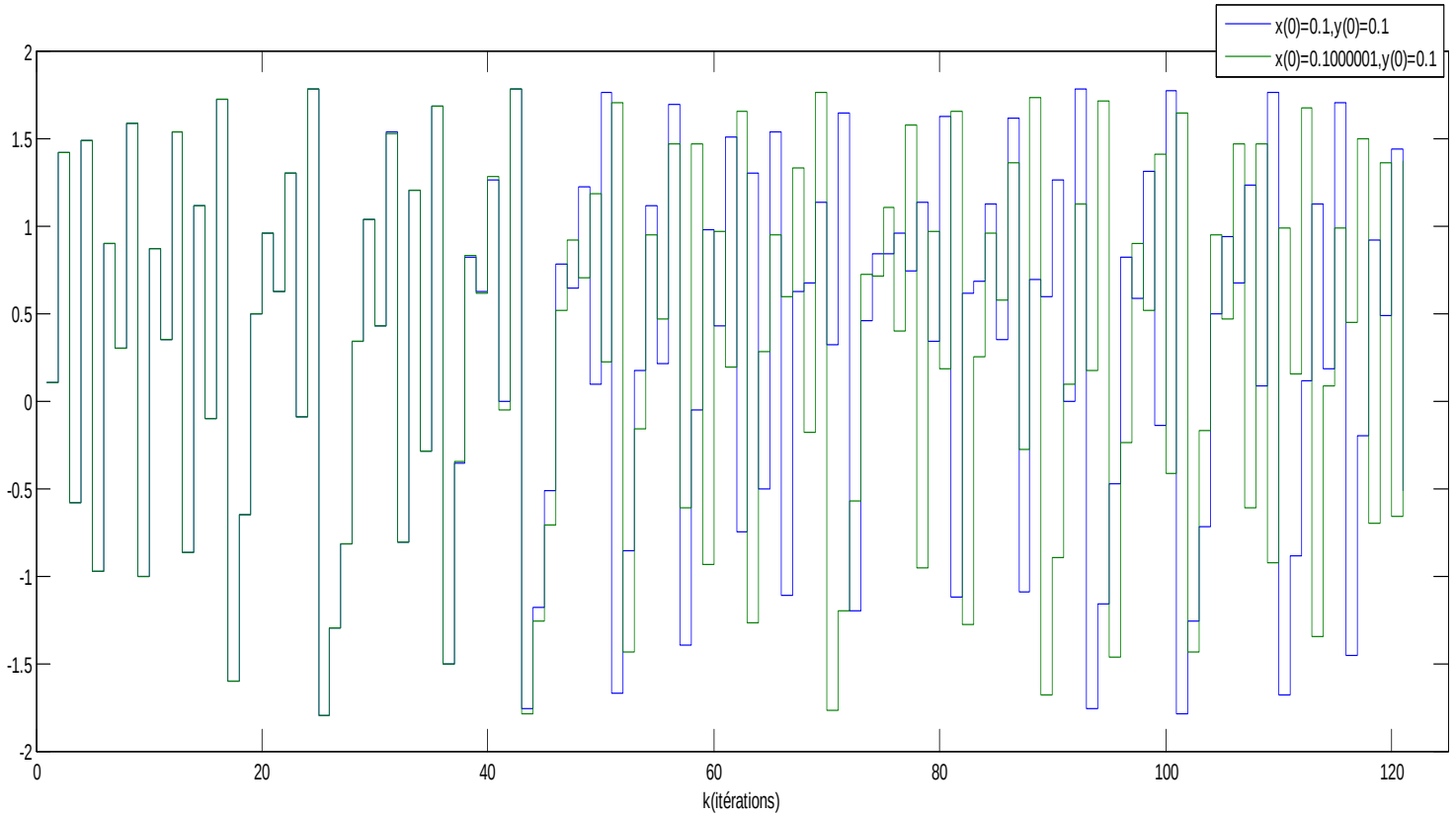


Figure (I.15) : Sensibilité aux conditions initiales du système de Hénon.

I. 7 Utilisation des systèmes chaotiques [14]

La théorie du chaos est utilisée dans divers domaines tels que :

- La biologie, elle permet d'expliquer les variations des populations animales, les oscillations du cerveau, ainsi que les arythmies cardiaques et bien d'autres.
- En économie, les mouvements commerciaux et les marchés financiers, ainsi que les cycles économiques, peuvent être expliqués en partie par cette théorie.
- Dans le domaine de l'art, depuis les années 1980, la beauté des fractales est exploitée et appréciée, et on voit des expositions se multiplier avec pour thème des images fascinantes.
- Et enfin nous la retrouvons aussi dans le domaine de transmission de données sécurisé qui fait l'objet de notre thème.

I. 8 Conclusion

Dans ce chapitre nous avons présenté quelques notions et définitions sur les systèmes chaotiques, ainsi que leurs caractéristiques. On a vu que ces systèmes sont utilisés dans divers domaines dont la cryptographie.

Le chapitre suivant sera consacré à l'étude de la synchronisation des systèmes chaotiques.

Chapitre II

Synchronisation du chaos et méthodes de cryptage

Chapitre II Synchronisation du chaos et méthodes de cryptage

II. 1 Introduction

Depuis quelques années, de nombreux chercheurs étudient la possibilité d'utiliser des signaux chaotiques pour transmettre des données. En effet, l'emploi du chaos dans les systèmes de communication permet de renforcer la sécurité de transmission de l'information et de réduire la probabilité d'interception. De nombreuses techniques de transmission ont été proposées telle que la méthode par addition [5] par inclusion [9] ou commutation [15].

Dans les systèmes de communication, la synchronisation est fondamentale pour une transmission réussie. L'aspect pseudo aléatoire du chaos nous amène à penser qu'il est impossible de le synchroniser. Cette hypothèse a été proposée par Fujisaka et Al en 1983 [15]. Ce n'est qu'en 1990 que les deux chercheurs Pecora et Carroll [26] ont montré que deux systèmes chaotiques identiques peuvent se synchroniser. Cette découverte a ouvert la voie pour des applications du chaos aux télécommunications pour sécuriser l'information et encore d'autres méthodes pour synchroniser le chaos tel que la synchronisation par boucle fermée ou bien la synchronisation par observateur.

Dans ce chapitre nous allons présenter les différentes méthodes utilisées dans la synchronisation du chaos, ainsi que celles utilisées pour le cryptage de l'information.

II. 2 La synchronisation

Le phénomène de synchronisation a été étudié depuis longtemps. Au 17 ^{iéme} siècle, Huygens (1629-1695) remarqua ce phénomène en étudiant deux horloges de fréquences légèrement différentes. Il constata qu'en les reliant l'une à l'autre avec un morceau de bois, elles affichaient toutes les deux la même heure : elles se synchronisaient. Des exemples de synchronisation existent dans la nature et dans le domaine technique [15].

Soient les deux systèmes suivants :

$$\begin{cases} S_1: & \dot{x} = f_1(x,u) \\ S_2: & \dot{x}' = f_2(x',u) \end{cases} \qquad \text{(II.1)}$$

Les deux systèmes sont dit synchronisés si :

$$\lim_{t\to\infty} e(t) = \lim_{t\to\infty} |x(t) - x'(t)| = 0 \qquad \text{(II. 2)}$$

Avec : $x\ et\ x' \in \mathbb{R}^n\ et\ f_1, f_2 \in \mathbb{R}^n * \mathbb{R}^n \rightarrow \mathbb{R}$ sont des champs de vecteur non linéaires, et $e(t)$ est l'erreur de synchronisation.

L'utilisation des circuits identiques dans la synchronisation des systèmes chaotiques est la base des méthodes traditionnelles, on supposera deux systèmes chaotiques identiques qui oscillent de manière indépendante, s'ils échangent de l'énergie entre eux, action de « couplage », ils finiront par avoir un comportement commun, ils se synchronisent par « couplage unidirectionnel » ou « couplage bidirectionnel » [15].

A ce jour, différentes méthodes de synchronisation ont été proposées et explorées. On citera par exemple la synchronisation en boucle fermée ainsi que la synchronisation à l'aide d'observateur.

II. 2. 1 Méthodes de synchronisation

Il est possible de coupler deux systèmes chaotiques dans un sens (couplage unidirectionnel) ou dans les deux sens (couplage bidirectionnel). Dans le cas d'une synchronisation unidirectionnelle, le couplage entre les deux systèmes chaotiques est réalisé à l'aide d'un élément fonctionnant dans un seul sens, (ex : suiveur). Par contre, dans le cas de la synchronisation bidirectionnelle, le couplage entre les deux systèmes est réalisé à l'aide d'un élément permettant l'échange d'énergie dans les deux sens (ex : résistance). Les deux types de couplage peuvent être appliqués pour des systèmes non identiques [5].

II. 2. 1. 1 Synchronisation par boucle fermée [16]

On l'appelle aussi méthode de synchronisation par contre réaction. Le principe de cette technique consiste à utiliser l'erreur entre l'émetteur et le récepteur pour corriger le comportement du récepteur afin de réaliser la synchronisation.

Supposons les deux systèmes suivants :

Emetteur :

$$\begin{cases} \dot{x} = f(x) \\ y = h(x) \end{cases} \qquad \text{(II.3)}$$

Récepteur :

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

Où g est une fonction de l'erreur entre y et $\hat{y}$, g est choisie de telle sorte à garantir la synchronisation.

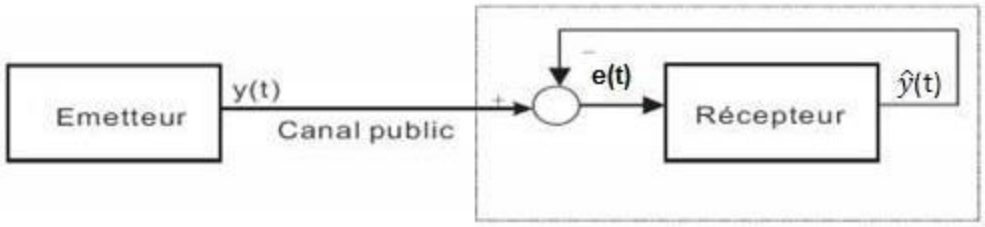


Figure (II.1) : Principe de la synchronisation par boucle fermée

II. 2. 1. 2 Synchronisation identique

C'est la méthode proposée par Pecora et Carroll [26]. L'idée consiste à décomposer un système chaotique en deux sous-systèmes, l'un maître et l'autre esclave. Ces derniers peuvent se synchroniser sous l'effet d'un couplage avec un signal commun. Cette technique a l'avantage de représenter une solution simple et performante de synchronisation, dont l'objectif est que le système esclave reproduise le plus fidèlement possible l'état du système maître, après un régime transitoire [17].

On considère le système chaotique suivant :

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

Où $x = [x_1, x_2, \dots\dots\dots\dots x_n]^T$ est le vecteur d'état.

On divisera le système initial en deux sous systèmes (S_1, S_2):

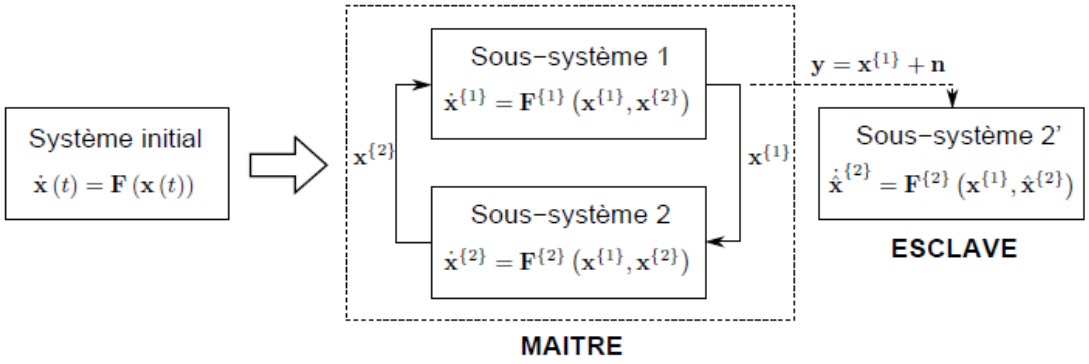
$$\begin{cases} S_1 = \dot{x}_1 = f_1(x_1, x_2) \\ S_2 = \dot{x}_2 = f_2(x_1, x_2) \end{cases} \quad \text{(II.6)}$$

Par la suite nous allons concevoir un nouveau sous système $S_2^{'}$ qui présente une même dynamique que S_2 et dont l'entrée est x_1.

$$S_2^{'} = \dot{\hat{x}}_2 = f_2(x_1, \hat{x}_2) \tag{II.7}$$

Le sous système $S_2^{'}$ est un candidat qui peut se synchroniser avec la dynamique complète initiale, une synchronisation parfaite peut s'accomplir si ce dernier est stable, c'est-à-dire que l'ensemble des coefficients de Lyapunov du sous-système $S_2^{'}$ sont négatifs.

On appellera le système (S_1, S_2)maître et le sous-système $(S_2^{'})$ esclave.



Avec n : est un bruit associé au canal de communication

Figure (II.2) : Principe de la synchronisation identique

II. 2. 1. 3 Synchronisation à l'aide d'observateur

Un observateur est un système qui permet d'estimer les états inconnus d'un système dynamique qui ne peuvent pas être mesurés directement : soit inaccessible, soit par économie.

Après la découverte de Pecora-Carroll [26], le problème de synchronisation a été rapidement relié au problème plus général de l'observation d'état non linéaire classique. Dans cette approche, le système maître est un système chaotique quelconque et le système esclave est un observateur d'état correspondant.

Pour ce principe, on dit que l'émetteur et le récepteur sont synchronisés si le système $\dot{\hat{x}} = \hat{f}(\hat{x}, u)$est un observateurqui estime les états du système $\dot{x} = f(x, u)$.

$y = h(x)$. Autrement dit, le problème de synchronisation revient à déterminer une fonction $\hat{f}$ telle que :

$$\lim_{t\to\infty} |x(t) - \hat{x}(t)| = 0$$

$x(t)$: Etat du système

$\hat{x}(t)$: Etat estimé

La figure (II.3) illustre ce principe de synchronisation [18]

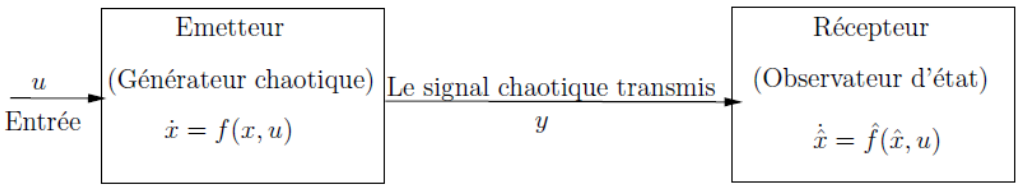


Figure (II.3) : Principe de synchronisation à base d'observateur

a. Observabilité

Soit le système suivant :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \\ x(t_0) = x_0 \end{cases} \quad \text{(II.8)}$$

Où : $x \in \mathbb{R}^n, y \in \mathbb{R}^p, u \in \mathbb{R}^m, A, B, C, et\ D$ sont des matrices de dimension appropriées.

Le système est dit complètement observable, s'il existe un temps fini $t_1 > t_0$, tel que la connaissance de l'entrée $u(t)$ et de la sortie $y(t)$ pour $t \in [t_0, t_1]$ permet de reconstruire l'état $x(t)$ [16].

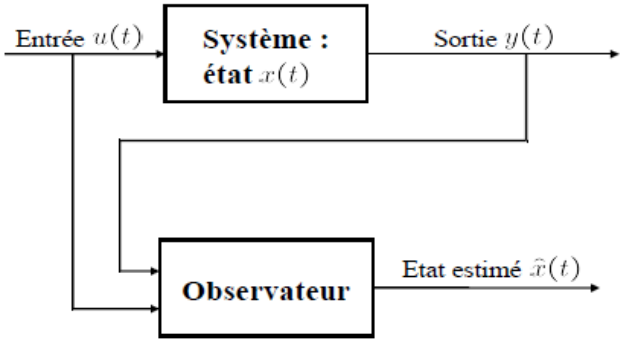


Figure (II.4) : Principe de l'observateur

a.1 Observabilité des systèmes linéaire

Soit le système linéaire suivant :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad \text{(II.9)}$$

Où les vecteurs x(t) $\in \mathbb{R}^n, u(t) \in \mathbb{R}^m$ et $y(t) \in \mathbb{R}^p$ représentent respectivement l'état, la commande et la sortie du système. Les matrices A, B, C et D sont des matrices constantes de dimensions appropriées. L'observabilité du système linéaire est garantie si et seulement si :

$$Rang(O) = Rang\begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{pmatrix} = n \quad \text{(II.10)}$$

Le système (II.9) est observable si et seulement si le rang de la matrice d'observabilité O est égal à la dimension n de ce système, dans le cas où le rang de la matrice O est inférieur à n on parle alors d'observabilité partielle.

a.2 Observabilité des systèmes non linéaire [15]

Cas continu

On considère le système non linéaire suivant :

$$\begin{cases} \dot{x}(t) = f(x) + g(x)u(t) \\ y(t) = h(x) \end{cases} \quad \text{(II.11)}$$

Où $x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m$ sont respectivement les vecteurs d'état et de commande.

$h(x)$ est la sortie du système.

La dérivée de Lie est utilisée pour définir l'observabilité d'un système non linéaire. Elle est définie comme suit :

$$Lfh(x) = \sum_{i=1}^{n} \frac{\partial h(x)}{\partial x_i} f_i(x) = \frac{\partial h(x)}{\partial x_1} f_1(x) + \frac{\partial h(x)}{\partial x_2} f_2(x) + \dots\dots\dots\dots \frac{\partial h(x)}{\partial x_n} f_n(x) \quad \textbf{(II.12)}$$

avec :

$$f(x) = \begin{pmatrix} \\ f_1(x) \\ f_2(x) \\ . \\ . \\ . \\ f_n(x) \end{pmatrix}$$

Le système (II.11) doit satisfaire la condition du rang d'observabilité $rang(M) = n$

Où $M = \left[dh; dL_f h; \dots\dots\dots dl_f^{n-1} h\right]$

$$rang(M) = Rang \begin{pmatrix} dh \\ dL_f h \\ \vdots \\ dL_f^{n-1} h \end{pmatrix} = n \quad \text{(II.13)}$$

Avec : n est la dimension du système

Cas discret

Soit le système non linéaire à temps discret suivant :

$$\begin{cases} x_{k+1} = f(x_k, u_k) \\ y_k = h(x_k) \end{cases} \quad \text{(II.14)}$$

Où $x_k \in \mathbb{R}^n, y_k \in \mathbb{R}^p$ et $u_k = (u_{1k}, u_{2k}, \dots\dots\dots, u_{mk})^T \in \mathbb{R}^m$.

y_k est la sortie du système.

Comme pour le cas continu, l'observabilité des systèmes en temps discret se vérifie par le rang d'observabilité

$$\dim\big(doh(x_0)\big) = n \qquad \text{(II.15)}$$

Avec : n dimension du système

Ceci peut être reformulé comme suit :

$$rang\ [span\{dh, d(foh), \dots\dots\dots, d(f^{n-1}o\ h)\}] = n \qquad \text{(II.16)}$$

b. Différents types d'observateurs

1. Observateur de Luenberger

On considère le système suivant :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y = Cx(t) \end{cases} \qquad \text{(II.17)}$$

Luenberger propose l'observateur suivant pour le système (II.17) :

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)) \\ \hat{y}(t) = C\hat{x}(t) \end{cases} \qquad \text{(II.18)}$$

$$==> \begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t)) \\ \dot{\hat{x}}(t) = (A - LC)\hat{x}(t) + Bu(t) + Ly(t) \end{cases} \qquad \text{(II.19)}$$

L'évolution de l'état est corrigée grâce au modèle en fonction de l'écart entre la sortie mesurée et la sortie reconstruite par l'observateur$(y(t) - \hat{y}(t))$.

L'état x en fonction de la commande u et des mesures y est reconstruit par l'observateur, et la matrice de gain L est choisie de manière à ce que l'erreur converge exponentiellement vers zéro.

$$\lim_{t\to\infty} e(t) = \lim_{t\to\infty} |x(t) - \hat{x}(t)| = 0 \qquad \text{(II. 20)}$$

Pour que le système soit stable, et en utilisant la technique de placement de pôles, on choisit le gain L de l'observateur de telle sorte que les valeurs propres de la matrice $(A - LC)$ soient dans le demi plan complexe gauche (à partie réel négative), de ce fait on choisit une dynamique d'erreur plus rapide que celle de processus.

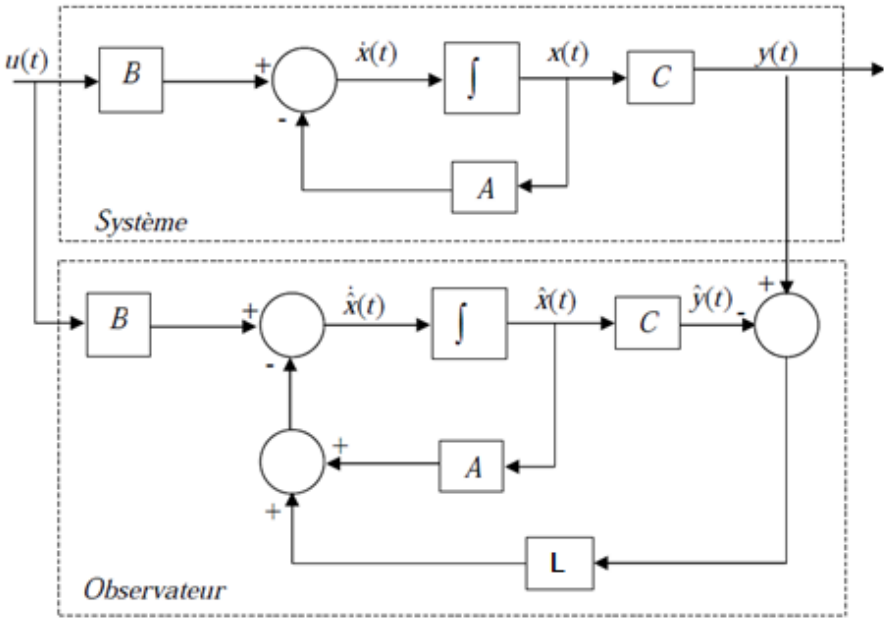


Figure (II.5) : Schéma structurelle de l'observateur de Luenberger

2. Observateur impulsif [5]

Soit le système suivant :

$$\begin{cases} \dot{x}_1(t) = f_1(x_1, x_2, t) \\ \dot{x}_2(t) = f_2(x_1, x_2, t) \\ y(t_k) = x_1(t_k) \end{cases} \quad \text{(II.21)}$$

Avec : $x_1(t) \in \mathbb{R}^p, x_2(t) \in \mathbb{R}^{n-p}$ sont les états du système et $y(t_k) \in \mathbb{R}^p$ est le vecteur de sortie.

L'observateur impulsif du système (II.21) est donné comme suit :

$$\begin{cases} \dot{\hat{x}}_1(t) = f_1(\hat{x}_1, \hat{x}_2, t) \\ \dot{\hat{x}}_2(t) = f_2(\hat{x}_1, \hat{x}_2, t) \\ \hat{y}_1(t_k^+) = x_1(t_k) \end{cases} \quad \text{(II.22)}$$

L'objectif consiste à contraindre l'observateur à suivre l'évolution du système original à des instants (t_k), un état du système est transmis sous forme d'impulsions pour réduire les redondances du signal.

A partir des systèmes (II.21) et (II.22), on obtient le système d'erreur d'observation suivant :

$$\begin{cases} \dot{e}_1(t) = f_1(x_1, x_2, t) - f_1(\hat{x}_1, \hat{x}_2, t) \\ \dot{e}_2(t) = f_2(x_1, x_2, t) - f_2(\hat{x}_1, \hat{x}_2, t) \\ e_1(t_k^+) = 0 \end{cases} \quad \text{(II.23)}$$

3. Observateur à mode glissant [19]

Un observateur à modes glissants est un observateur dont le terme correcteur est une fonction **sign**. Il s'agit de contraindre, à l'aide des fonctions discontinues, les dynamiques du système à converger vers une "surface de glissement".

Soit le système suivant :

$$\begin{cases} \dot{x}(t) = f(x) + g(x)u \\ y(t) = h(x) \end{cases} \quad \text{(II.24)}$$

L'observateur à modes glissants pour ce système s'écrit de la façon suivante :

$$\begin{cases} \dot{\hat{x}}(t) = f(\hat{x}) + g(\hat{x})u + K\, sgn(y - \hat{y}) \\ \hat{y} = h(\hat{x}) \end{cases} \quad \text{(II.25)}$$

Où K est une matrice de gain de dimension $n * p$.

Dans ce cas, on impose l'évolution des dynamiques du système sur une variété, sur laquelle l'erreur d'estimation de la sortie $e = (y - \hat{y})$ est nulle. Ainsi, cette erreur converge vers zéro au bout d'un temps fini, et la dynamique du système se réduit de n à $n - p$.

La figure qui suit illustre le principe d'un observateur à mode glissants

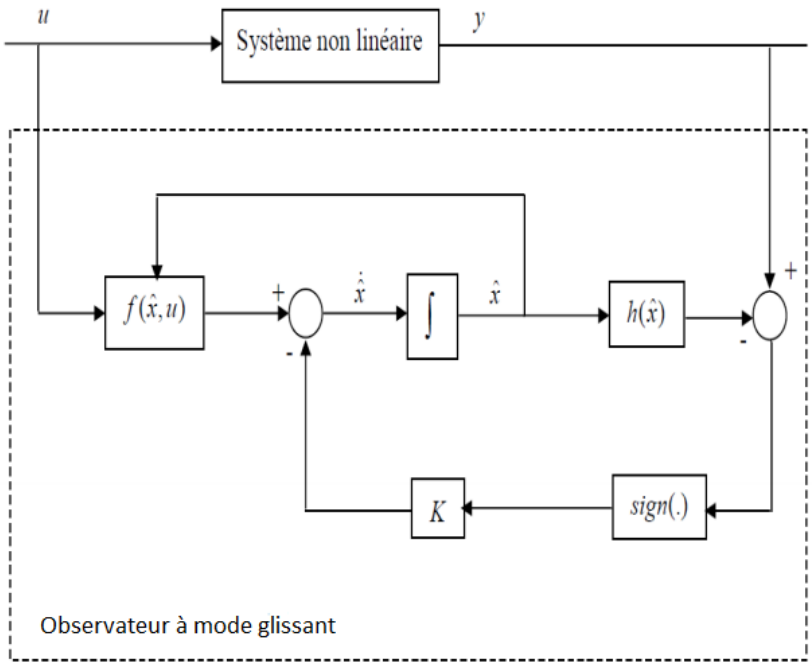


Figure (II.6) : Schéma structurel de l'observateur à modes glissants

4. Observateur à entrée inconnue [15]

Le schéma de la figure (II.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 $u(t)$. Différentes techniques de synthèse d'observateurs à entrées inconnues ont été utilisées dans la littérature, et peuvent être utilisées à des fins de décryptage, comme ceci est illustré par la figure (II.7) ci-dessous.

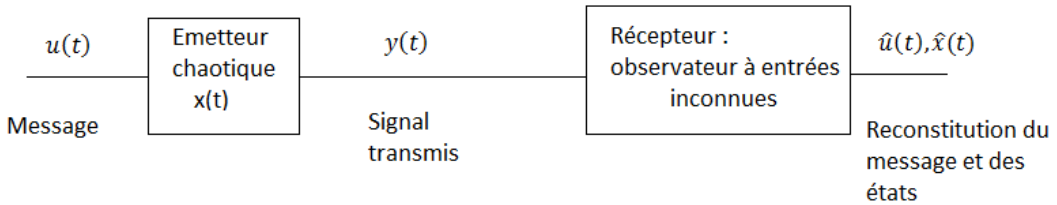


Figure (II.7) : Observateurs à entrées inconnues

II. 3 Utilisation du chaos pour la transmission sécurisée

Dans cette partie, on s'intéresse aux techniques de transmission sécurisée d'informations qui reposent sur le principe de synchronisation chaotique. Le point commun constaté dans la majorité des techniques développées dans la littérature est l'utilisation de la configuration maître-esclave pour laquelle on dispose d'un émetteur chaotique (système maître) qui génère le signal du texte chiffré transmis dans la canal de communication vers un système récepteur (système esclave) qui a pour objectif de synchroniser avec le système maître et de restaurer le signal d'information grâce à une clé partagé avec l'émetteur [20].

II. 3. 1 Quelques définitions [5]

II. 3. 1. 1 Cryptanalyse

C'est une étude qui détermine les éventuelles faiblesses des systèmes cryptographiques et ce en étudiant les probabilités de succès des attaques. Le principal objectif de la cryptanalyse est de déchiffrer le message dans le but de le rendre claire, pour cela il est nécessaire de se mettre dans la peau du pirate.

II. 3. 1. 2 Cryptographie

C'est une étude qui consiste à protéger le message à transmettre et ce en lui appliquant une transformation qui le rend incompréhensible, ce qui est appelé chiffrement.

II. 3. 1. 3 Crypter (chiffrer)

C'est l'opération de transformer un texte claire en un texte codé, le résultat est appelé chiffrement.

II. 3. 1. 4 Décrypter (déchiffrer)

C'est l'opération qui consiste à traduire un message chiffré en un message claire, et c'est en connaissant la clé, donc seul le destinataire légitime peut déchiffrer le message, car il est le seul détenteur de la clé.

II. 3. 2 Méthodes de cryptage

Il existe de nombreuses méthodes de cryptage par chaos. Parmi ces méthodes on peut citer :

II. 3. 2. 1 Cryptage par addition

Dans cette méthode appelée, masquage chaotique, l'émetteur est un système chaotique autonome dont le signal de sortie y(t) est ajouté au signal du message m(t). La somme de deux signaux est transmise au récepteur à travers le canal de transmission, qui est un canal public. Le récepteur est constitué d'un système chaotique identique à l'émetteur et un simple soustracteur. Ainsi, après la synchronisation des deux systèmes chaotiques (émetteur et récepteur), le message est extrait à l'aide d'une opération de soustraction [15].

La figure (II.8) illustre la méthode de cryptage par addition

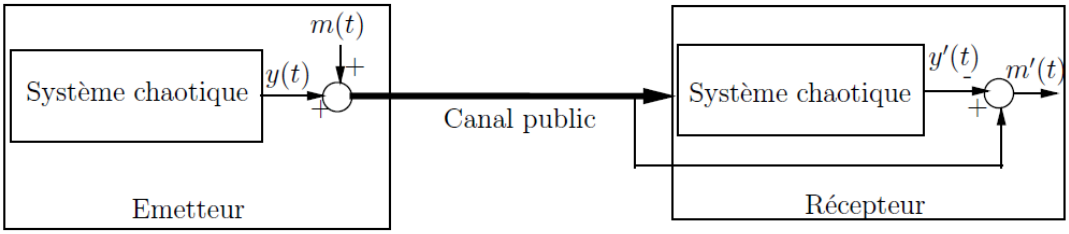


Figure (II.8) : Principe de cryptage par addition

II. 3. 2. 2 Cryptage par commutation [17]

Cette méthode (en anglais Chaos Shift Keying, CSK) est utilisée pour transmettre un message binaire. L'émetteur est composé de deux systèmes chaotiques et pour chaque niveau de message m(t) (0 ou 1), l'un des systèmes envoie sa sortie sur la ligne de transmission. Ainsi, le signal transmis commute entre deux attracteurs étrange.

Le récepteur est constitué de deux systèmes chaotiques identiques à ceux de l'émetteur et un bloc de comparaison permet de relever la valeur du message noté m'(t).
La figure (II.9) illustre le principe de commutation pour un système binaire.

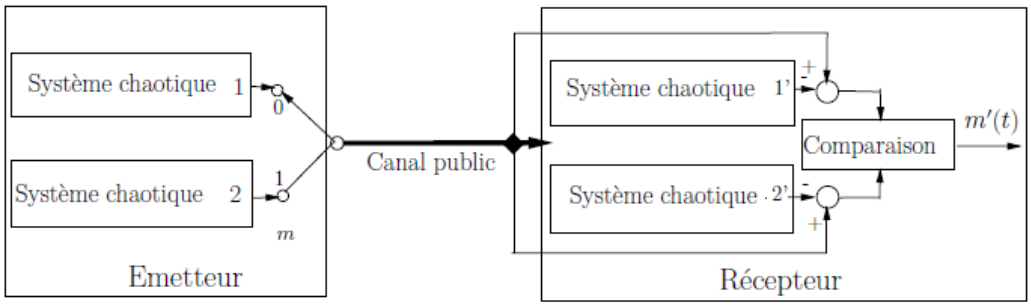


Figure (II.9) : Principe de cryptage par commutation

Cette méthode a d'énormes avantages, la robustesse au bruit en est un : en effet, au niveau du récepteur, on détermine la valeur exacte du message soit en évaluant l'erreur de synchronisation, soit par corrélation entre le signal envoyé et le signal récupéré.

II. 3. 2. 3 Cryptage par inclusion

Dans le cryptage par inclusion, le message source est inclus dans la structure du système chaotique du côté de l'émission. Dans ce cas, on observateur doit être utilisé à la réception pour récupérer le message original. Cette classe de méthode nécessite un seul canal de transmission. Ainsi elle présente beaucoup d'avantages et reste très utilisée en pratique [21].

La figure (II.10) illustre la méthode de cryptage par inclusion.

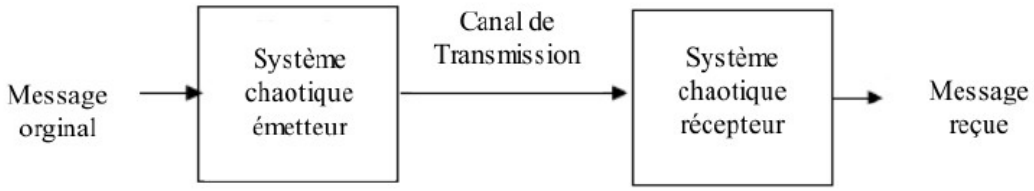


Figure (II.10) : Principe de cryptage par inclusion

II. 3. 2. 4 Transmission à deux voies [21]

Le principe de la transmission par deux voies est illustré dans la figure (II.11). L'idée de base consiste à séparer les tâches de synchronisation et de cryptage en utilisant deux voies de communication. L'émetteur chaotique génère un signal chaotique $y(t)$ transmis dans le premier canal de communication (Canal 1) vers le récepteur qui doit se synchroniser avec le système maître. L'émetteur génère également un autre signal chaotique utilisé dans une fonction de cryptage qui produit le signal chiffré $C(t)$ transmis dans un deuxième canal de transmission (Canal 2).

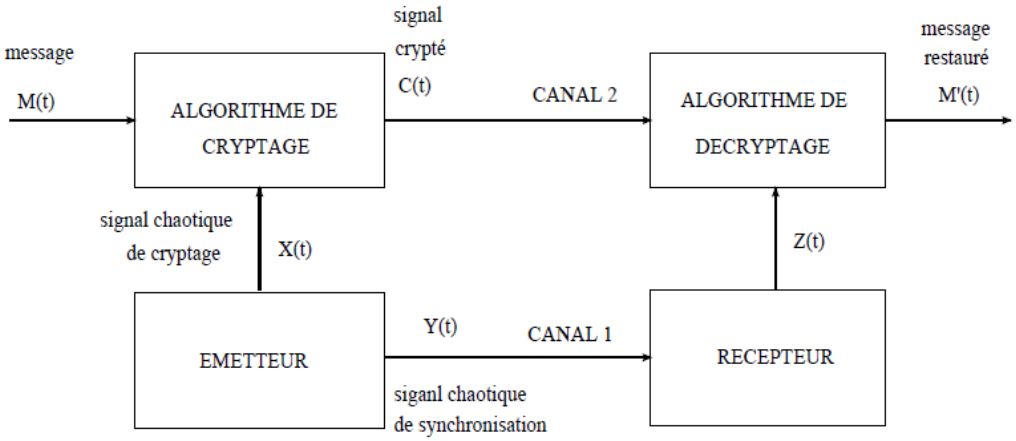


Figure (II.11) : Principe de la technique de transmission à deux voies

II. 4 Conclusion

Dans ce chapitre on a pu constater l'intérêt de la théorie du chaos dans le domaine de la transmission sécurisée de données. En effet un signal chaotique peut être utilisé pour crypter un message confidentiel.

Dans la première partie de ce chapitre, nous avons expliqué le concept de synchronisation des systèmes chaotiques ainsi que les différents modes de synchronisation. Ajouté à cela nous avons fait un tour d'horizon des différents observateurs existants. Ensuite dans la deuxième partie de ce chapitre, nous avons présenté le principe des techniques de cryptage exploitant le chaos, à savoir le cryptage par addition, par commutation, par inclusion et pour terminer par la technique de transmission à deux voies.

Chapitre III

Synchronisation du système de Lozi

III. 1 Introduction

La synchronisation de deux systèmes (émetteur et récepteur) par la technique des observateurs nécessite un signal d'injection généré par l'émetteur et transmis au récepteur afin que ce dernier se synchronise avec l'émetteur. L'envoi d'un signal d'injection à des instants discrets et de manière minutieuse sous forme d'impulsions est un grand avantage pour la transmission de données car cela permet de ne pas saturer le canal public.

Ce chapitre est consacré à la synchronisation de deux systèmes chaotiques identiques (émetteur et récepteur) à l'aide d'un observateur impulsif. Dans un premier temps nous allons définir un oscillateur. Par la suite, on étudiera un oscillateur chaotique appelé « système de Lozi ». Pour la simulation on utilisera **Matlab/Simulink.**

III. 2 L'émetteur

Afin de générer un signal chaotique et noyer un message, nous avons besoin d'un oscillateur électronique qu'on utilisera comme émetteur.

III. 2. 1 Définition et structure générale d'un oscillateur électronique

Un oscillateur électronique est un montage électronique autonome (pas de signal de commande), dont la fonction principale est de produire un signal périodique lors de sa mise sous tension. La forme du signal peut être sinusoïdale, carré, voire en dent de scie, ou d'une quelconque forme.

Un oscillateur comporte toujours un élément actif (circuit amplificateur) associé à un circuit passif (un filtre). L'élément actif est souvent un transistor bipolaire ou un amplificateur opérationnel. Sa structure est celle d'un système bouclé dans lequel une fonction du signal de sortie est ramenée à l'entrée pour l'auto entretien des oscillations.

D'une façon générale, un oscillateur électronique est représenté par le schéma bloc ci-dessous [24]

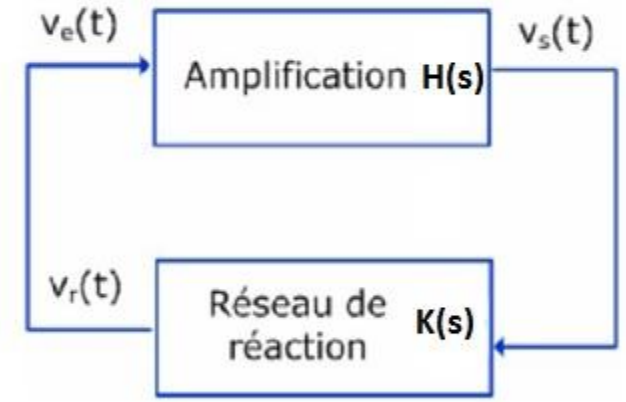


Figure $(\boldsymbol{III}.\mathbf{1})$ **:** Schéma de représentation d'un oscillateur électronique

Pour qu'un système bouclé oscille, il faut qu'il existe une fréquence f_0 ou une pulsation ω_0 pour laquelle le gain de boucle soit égal à 1 : c'est la condition de Barkhausen :

$$T(j\omega_0) = H(j\omega_0).K(j\omega_0) = 1$$

III. 2. 2 Caractérisation du chaos dans le système de Lozi

Dans ce travail notre étude se porte sur le système de Lozi qui est un oscillateur chaotique, il est régit par les équations suivantes [9].

$$\begin{cases} x_1(k+1) = 1 - a|x_1(k)| + bx_2(k) \\ x_2(k+1) = x_1(k) \\ y(k) = x_1(k) \end{cases} \qquad \text{(III. 1)}$$

Où a et b sont des constantes positives et k le nombre d'itération

Remarques

-La seule différence entre le système de Hénon et de Lozi est que le terme non linéaire x^2 du système de Hénon est remplacé par le terme $|x|$ dans le système de lozi.

-Si $a = 0$, L'application de Lozi est une application linéaire.

a) Exposant de Lyapunov

La figure $(III.2)$ représente les exposants de Lyapunov pour le système de Lozi. On peut observer qu'il y a deux exposants de Lyapunov sur tout un intervalle pour les valeurs des paramètres ($a = 1.7$et $b = 0.5$). Le premier exposant varie aux voisinages de $L_1 \approx 0.4694$, et le deuxième exposant qui est négatif et varie aux alentours de $L_2 \approx -1.1625$.

Ces résultats montrent que le système de Lozi est de nature chaotique.

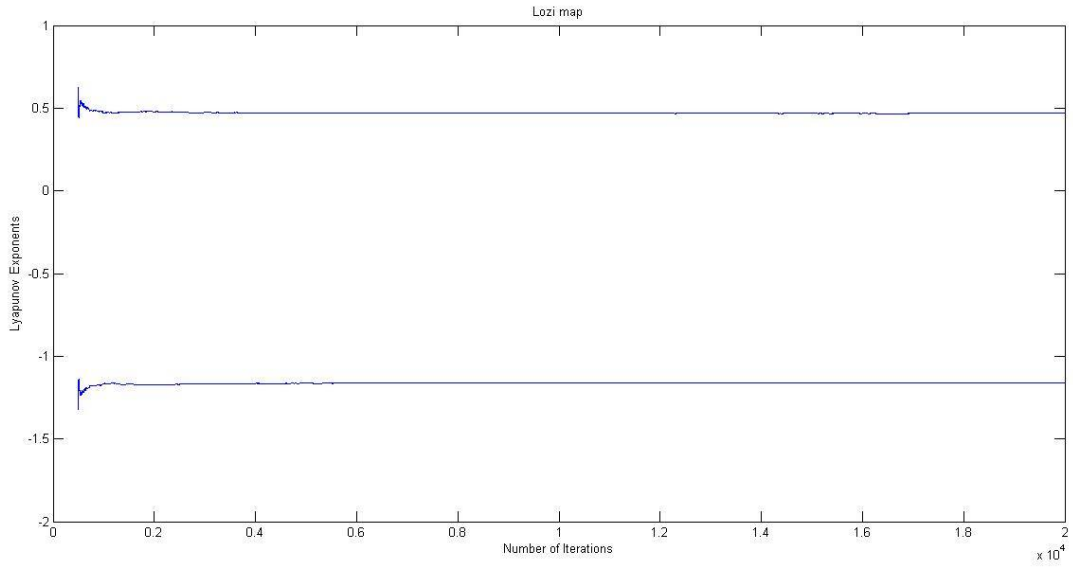


Figure $(III.2)$ **:** Les exposants de Lyapunov du système de Lozi

b) Spectre de puissance

Le spectre de puissance est le résultat de la transformée de Fourier de la fonction d'autocorrélation (Annexe A). Le spectre de puissance d'un système chaotique est continu.

La figure $(III.3)$ représente le spectre de puissance du système de Lozi. On remarque que le spectre de puissance de ce système a une large bande de fréquence.

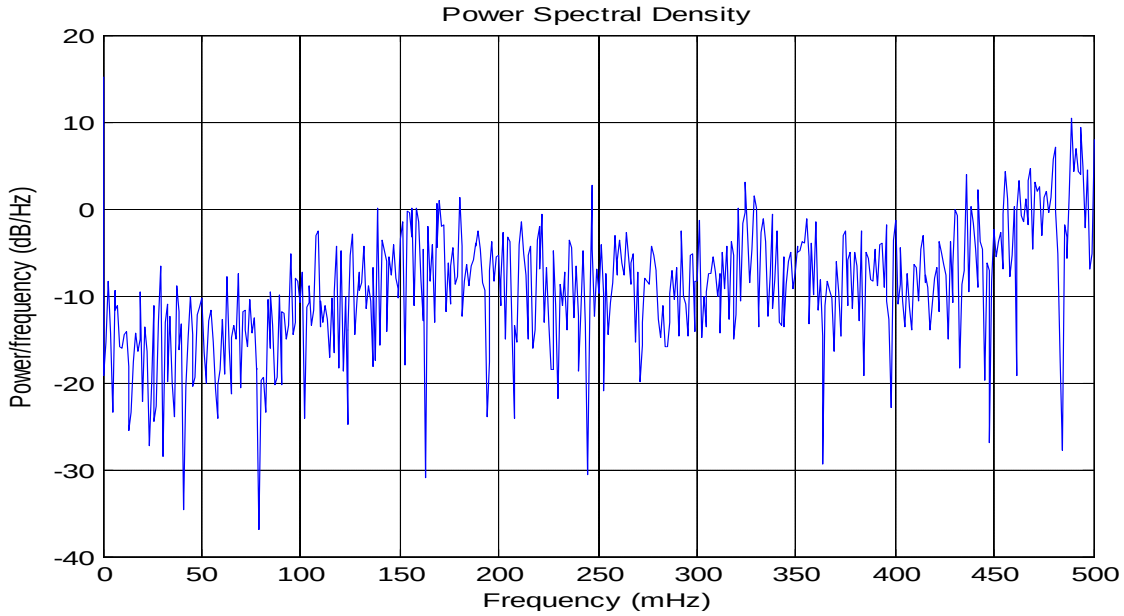


Figure $(\boldsymbol{III.3})$ **:** Le spectre de puissance pour le système de Lozi.

III. 3 Simulation sur Matlab

Pour la simulation du système de Lozi on utilise **Matlab/Simulink**. On fixe Les valeurs des paramètres de ce système à :$a = 1.7$ et $b = 0.5$, ainsi nous obtenons les figures suivantes :

a) Visualisation des états

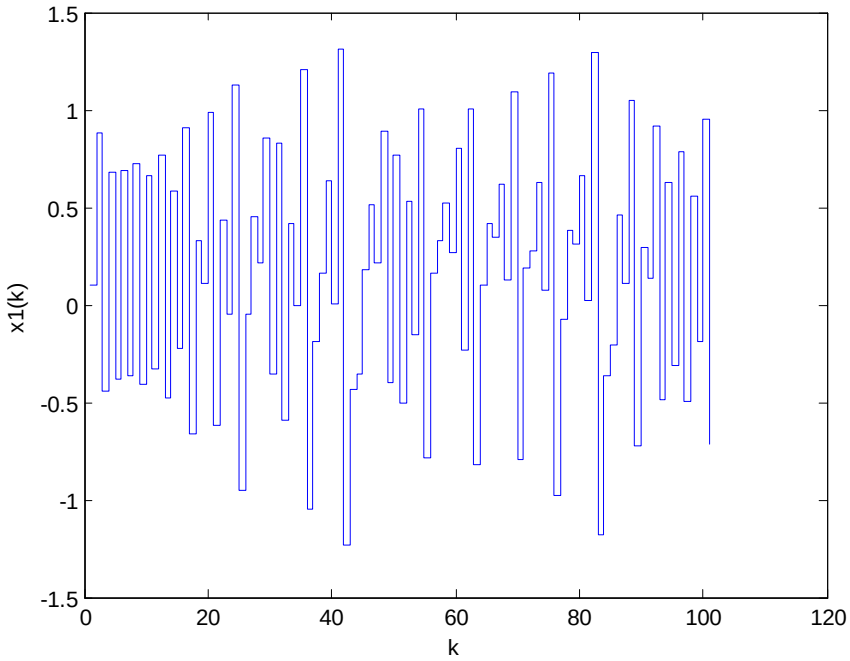


Figure $(\boldsymbol{III.4})$ **:** Graphe de l'état $\boldsymbol{x_1(k)}$ du système de Lozi.

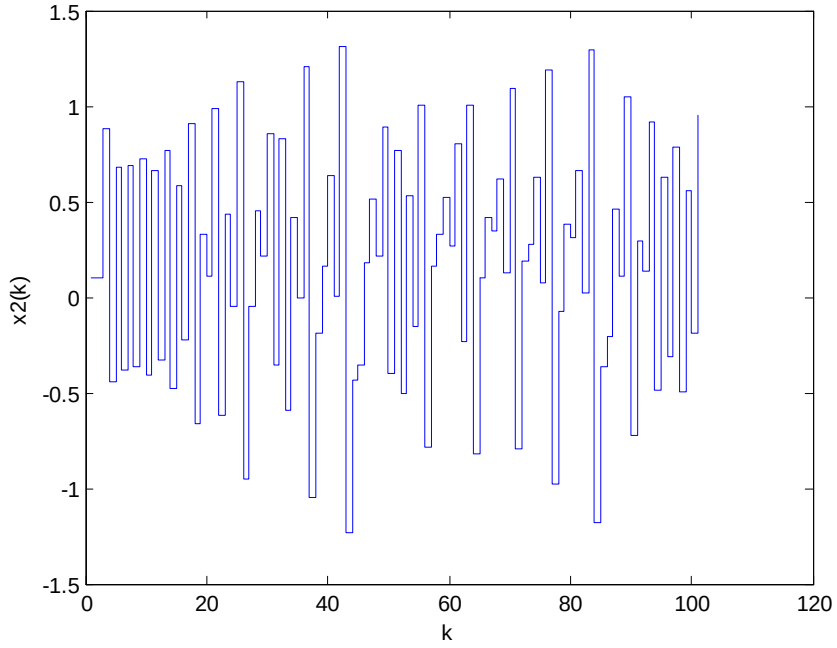


Figure $(\boldsymbol{III.5})$ **:** Graphe de l'état $\boldsymbol{x_2(k)}$ du système de Lozi.

Remarques

La présence d'oscillations apériodiques et irrégulières montrent que les signaux $x_1(k)$ et $x_2(k)$ sont des signaux chaotiques.

b) Visualisation de l'attracteur

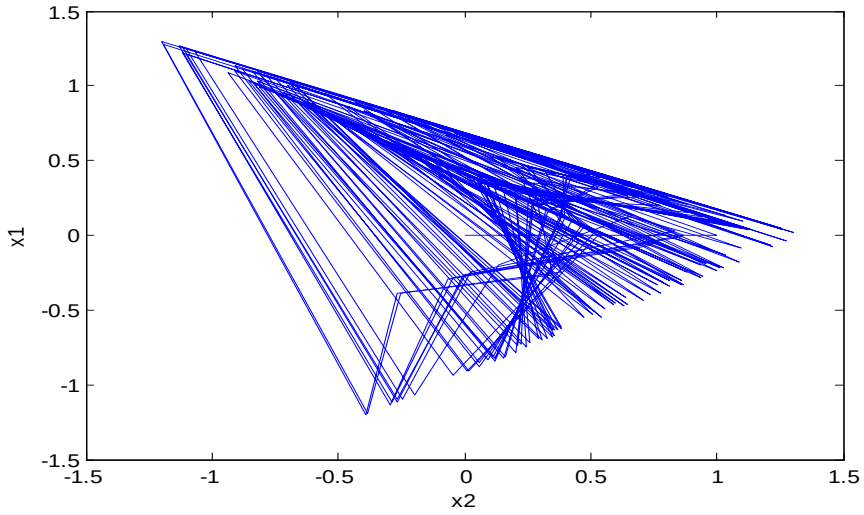


Figure $(\boldsymbol{III.6})$ **:** L'Attracteur de Lozi

Remarques

Cette figure montre l'aspect chaotique du système de Lozi. L'objet est dit étrange en raison de sa structure.

III. 4 Le récepteur

Un système dynamique est dit observable si on peut récupérer toutes ses grandeurs par une combinaison de mesures et de leurs dérivées. Nijmeijer et Mareels [27] étaient les premiers à avoir montré que la synchronisation unidirectionnelle de deux systèmes chaotiques peut être considérée comme un problème d'observateur non linéaire.

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

III. 4. 1 Condition du rang d'observabilité

Pour l'étude de l'observabilité d'un système non linéaire, il suffit de calculer et d'étudier le rang de la matrice d'observabilité (la Jacobienne). On dit qu'un système est observable si et seulement si sa matrice d'observabilité est égal à la dimension du système.

On reprend le système $(III.1)$ précédent :

$$\begin{cases} x_1(k+1) = 1 - a|x_1(k)| + bx_2(k) \\ x_2(k+1) = x_1(k) \\ y(k) = x_1(k) \end{cases} \quad \text{(III. 1)}$$

On peut écrire le système$(III.1)$sous la forme suivante :

$$\begin{cases} x(k+1) = f(x) \\ y(k) = h(x) \end{cases} \quad \text{(III. 2)}$$

La matrice d'observation (O) du système $(III.1)$ est définie comme suit :

$$O = \begin{pmatrix} dh \\ dfoh \end{pmatrix}$$

Nous avons

$$dh = [1 \quad 0]$$

Et

$$foh = 1 - a|x_1(k)| + bx_2(k) \qquad \text{(III. 3)}$$

Pour $x_1(k) > 0$

$$foh = 1 - ax_1(k) + bx_2(k)$$

$$dfoh = [-a \quad b] \qquad \text{(III. 4)}$$

Pour $x_1(k) < 0$

$$foh = 1 + ax_1(k) + bx_2(k)$$

$$dfoh = [a \quad b] \qquad \text{(III. 5)}$$

La Jacobienne est donnée par :

$$O = \begin{pmatrix} 1 & 0 \\ -asign(x_1) & b \end{pmatrix} \qquad \text{(III. 6)}$$

$$\det(O) = b$$

Le système $(III. 1)$ est observable si $b \neq 0$.

III. 4. 2 Synchronisation du système de Lozi

Le modèle d'observateur impulsif pour le système de lozi est représenté par :

$$\begin{cases} \hat{x}_1(k+1) = 1 - a|\hat{x}_1(k)| + b\hat{x}_2(k) \\ \hat{x}_2(k+1) = \hat{x}_1(k) \\ \hat{x}_1(t_k^+) = x_1(t_k) \qquad\qquad k = t_k \end{cases} \qquad \text{(III. 7)}$$

t_k *ensemble discret des instants de temps tell que* $0 < t_1 < t_2 < \cdots < t_i < t_{i+1} < \cdots$

$$et\ i \in \mathbb{Z}$$

A partir des systèmes$(III.1)$et $(III.7)$on obtient le système d'erreur d'observation suivant :

$$\begin{cases} e_1(k+1) = x_1(k+1) - \hat{x}_1(k+1) \\ e_2(k+1) = x_2(k+1) - \hat{x}_2(k+1) \\ e(t_k^+) = 0 \end{cases} \quad \text{(III.8)}$$

Le schéma du principe est représenté par la figure suivante [5]

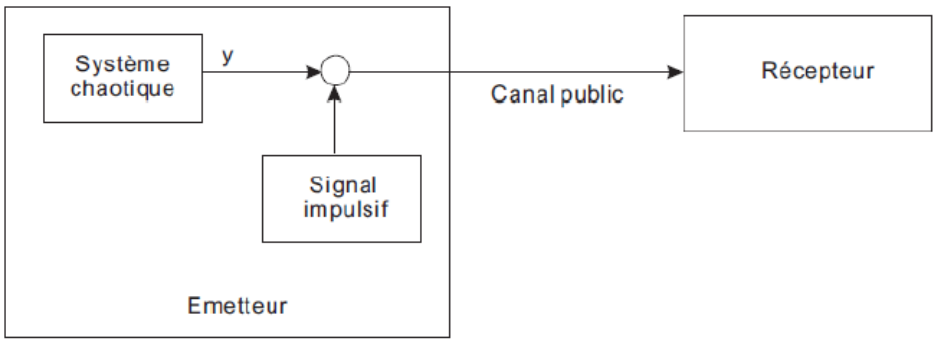


Figure $(III.7)$ **:** Synchronisation impulsive

III.4.2.1 Simulation sur Matlab

La simulation se fait sous Matlab/Simulink, on transmet $x(k)$ dont on a étudié l'observabilité, le signal est injecté via un Switch à chaque période de manière à forcer le système récepteur à suivre la trajectoire de l'émetteur.

Ainsi on obtient les résultats suivants :

a) Visualisation des états estimés

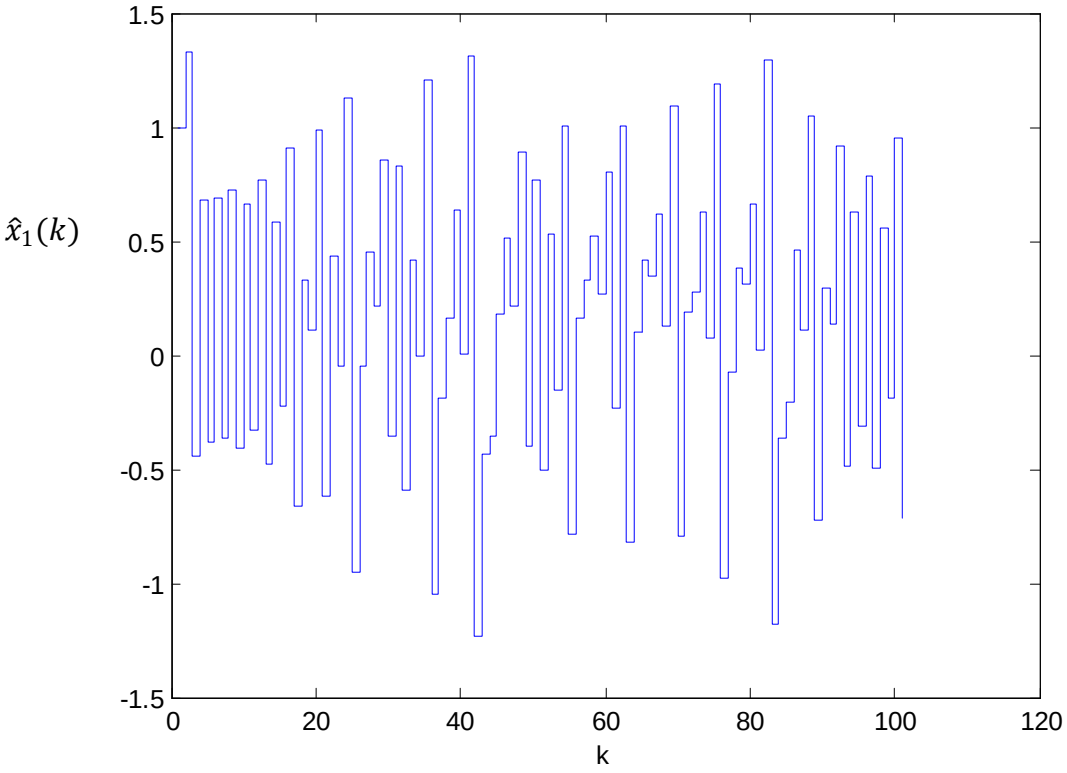


Figure $(III.8)$ **:** Graphe de l'état $\hat{x}_1(k)$

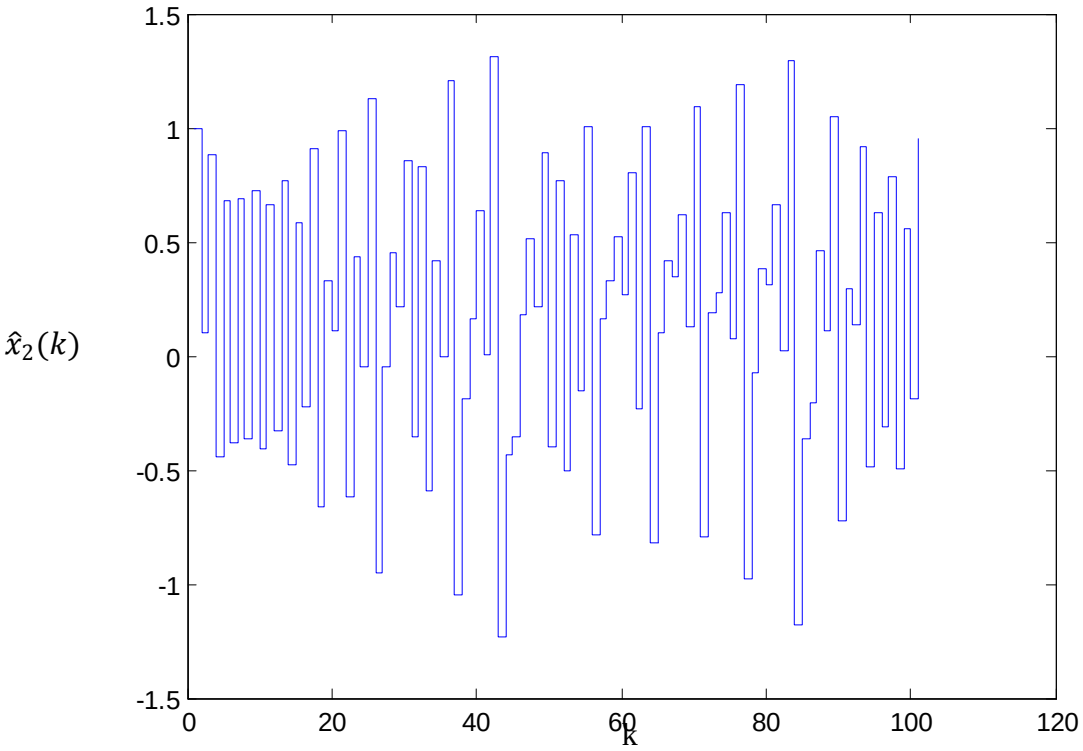


Figure $(III.9)$ **:** Graphe de l'état $\hat{x}_2(k)$

Remarques

La présence d'oscillations apériodiques et irrégulières montre que les signaux $\hat{x}_1(k)$et $\hat{x}_2(k)$ sont des signaux chaotiques.

b) Visualisation des écarts entre les états des deux systèmes

-Erreur entre $x_1(k)$et $\hat{x}_1(k)$:

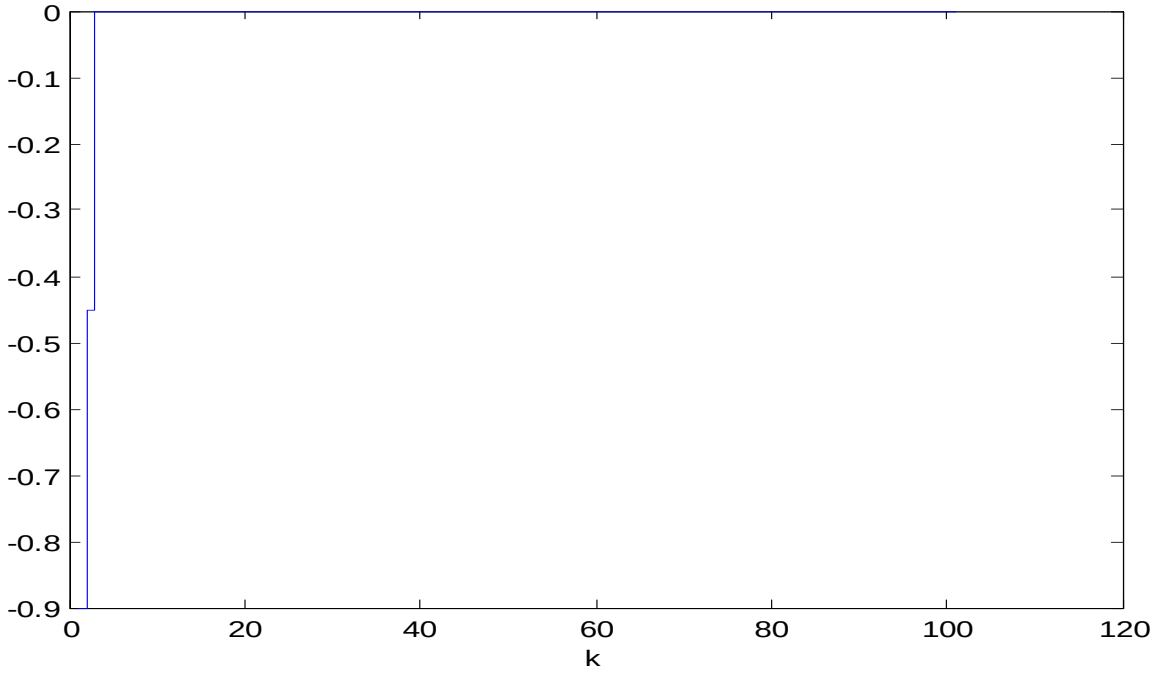


Figure $(III.10)$ **:** Graphe de l'écart$(x_1 - \hat{x}_1)$

-Erreur entre $x_2(k)$et $\hat{x}_2(k)$:

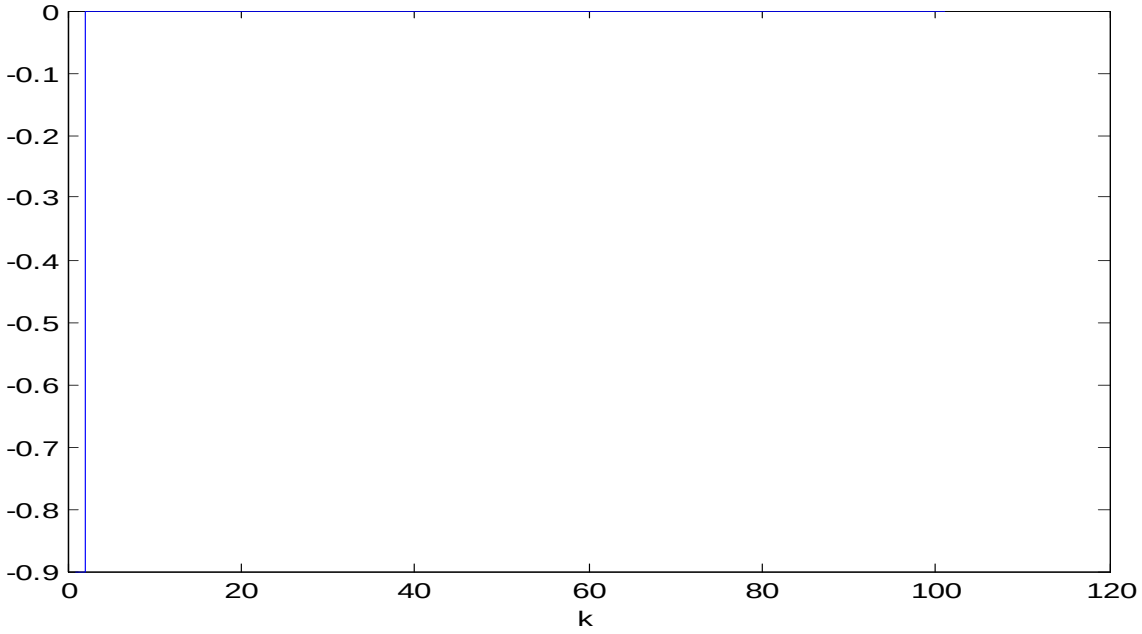


Figure $(III.11)$ **:** Graphe de l'écart$(x_2 - \hat{x}_2)$

Remarques

Les figures ($III.10$) et ($III.11$) nous montre la convergence des erreurs de l'état vers zéro quand k tend vers l'infini, ce qui montre la synchronisation des deux systèmes.

c) Récupération du message

Comme le montre la figure ($III.12$), on choisit un signal carré qu'on va additionner à l'un des états de l'émetteur, du côté récepteur on fait l'opération inverse.

Le message crypté est représenté sur la figure ($III.13$), celui récupéré sur la figure ($III.14$)

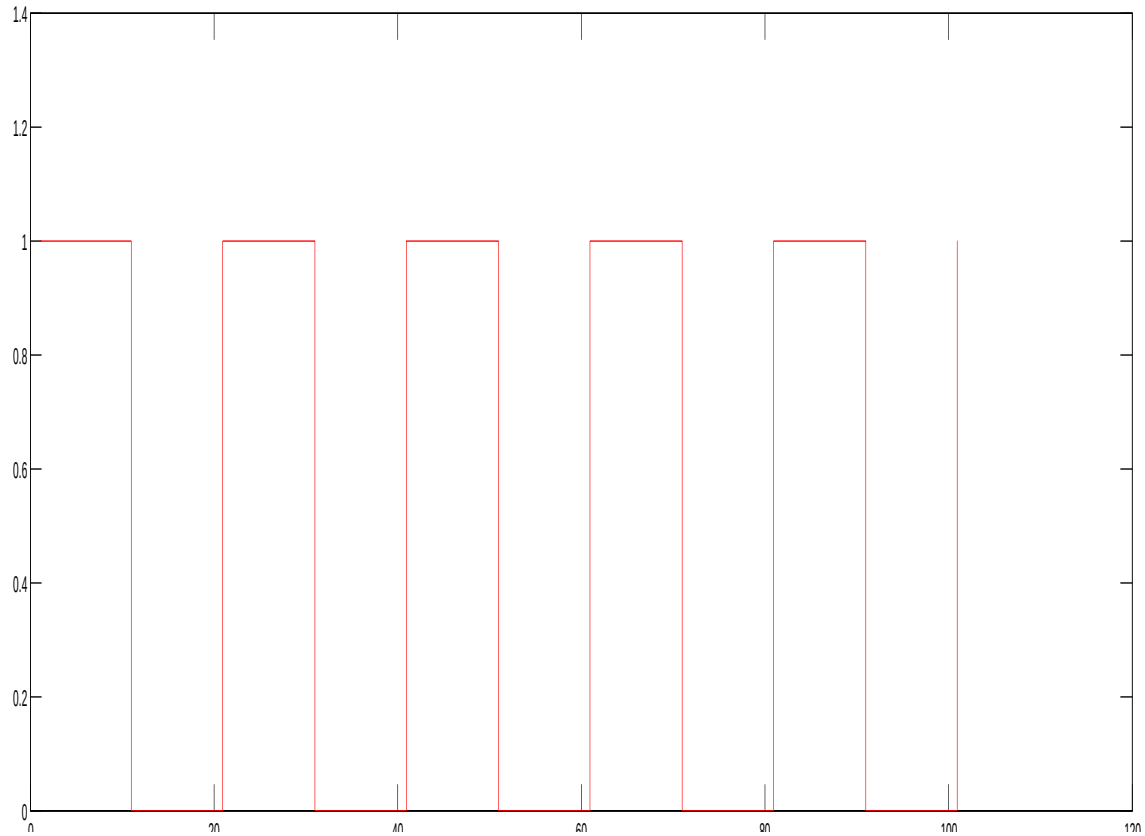


Figure ($III.12$) : Message original

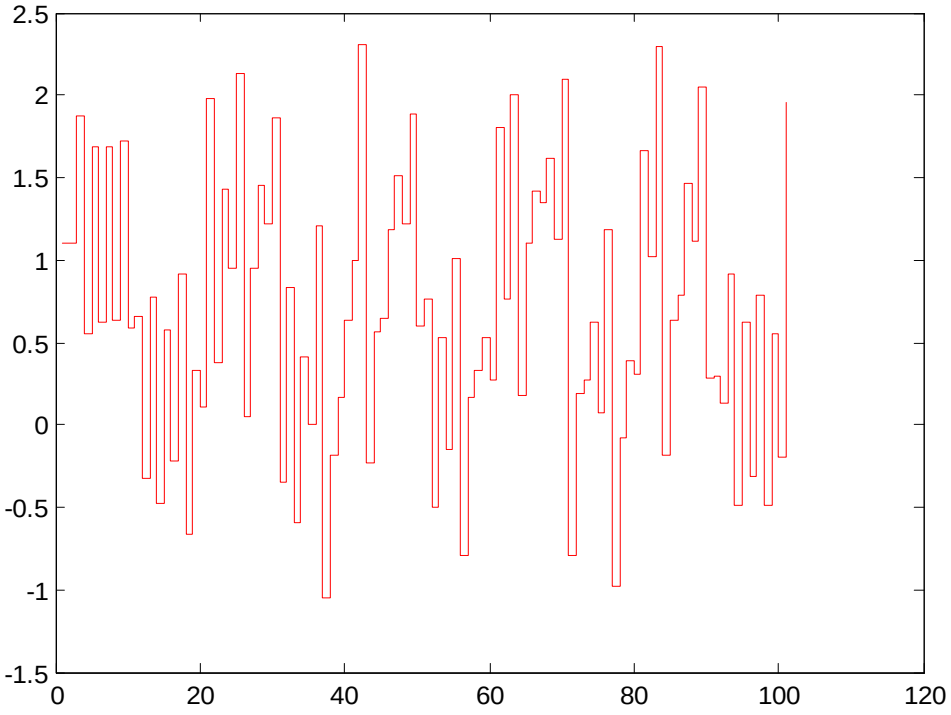


Figure (***III***. **13**) **:** Message crypté

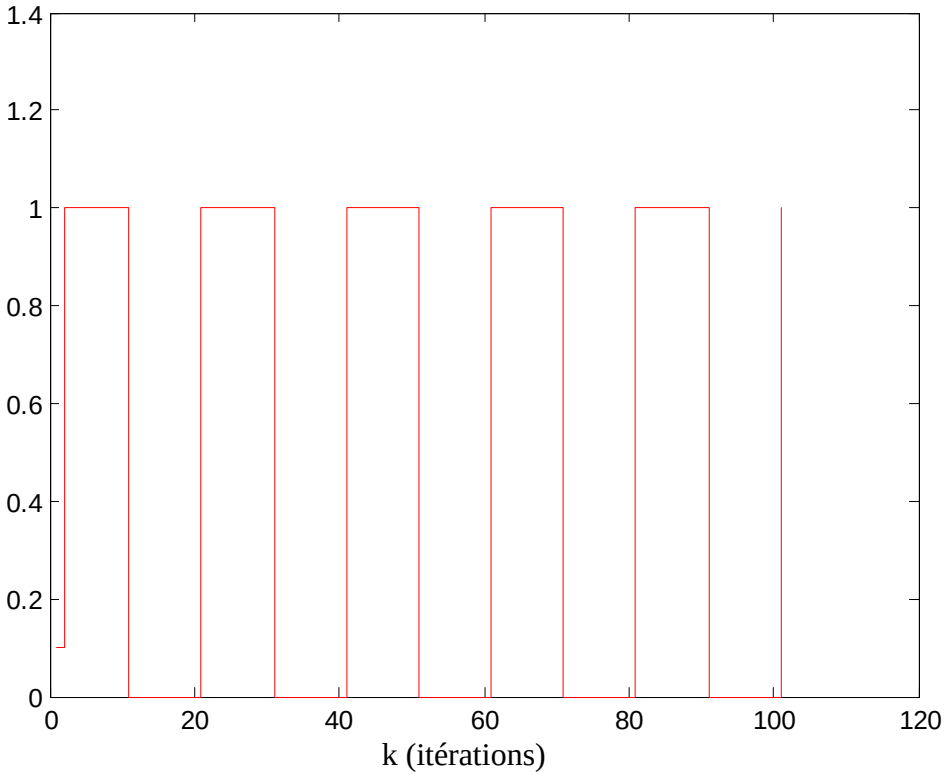


Figure (***III***. **14**) **:** Message récupéré

Remarque

Ces figures montre que le message est bien noyé dans le signal chaotique et que le message envoyé a été récupéré.

III. 5 Conclusion

Dans ce chapitre nous avons réussi à synchroniser deux systèmes chaotiques de Lozi à l'aide d'un observateur impulsif. L'objectif est de réaliser une transmission sécurisée de données, en noyant par addition le message dans le signal chaotique généré par l'émetteur. La récupération du message original s'est faite par soustraction du signal émis par l'émetteur et celui reconstruit par l'observateur impulsif. Les résultats obtenus par simulation montrent l'efficacité de cette approche.

Dans le chapitre qui suit nous allons passer à la réalisation du système de transmission de données sécurisées en utilisant une carte Arduino Mega.

Chapitre IV

Réalisation du système de transmission

IV. 1 Introduction

Nous avons vu dans le chapitre précédent ce qu'est un système de Lozi ainsi que ses propriétés et son fonctionnement théorique et simulé. Ayant acquis les notions nécessaires, nous allons dans un premier temps présenter le schéma de transmission sécurisée adopté. Par la suite nous allons expliquer et réaliser cette transmission à l'aide d'une carte électronique nommé Arduino Méga (Annexe C). Aussi, on abordera la manière de programmation de la carte avec Matlab Simulink. Enfin nous allons présenter les différents tests et résultats obtenus.

IV. 2 Présentation du système de transmission sécurisée de données à base d'oscillateurs de Lozi sur carte Arduino

Dans cette partie, nous allons présenter le diagramme bloc de la synchronisation impulsive adopté dans l'objectif de concevoir un système de transmission de données sécurisées constitué de deux blocs, un bloc pour l'émission et un autre bloc pour la réception.

L'émetteur contient une carte Arduino contenant les équations du système de Lozi, elle sert à la génération et au cryptage du message.

Le récepteur contient également un oscillateur de Lozi identique à celui de l'émetteur, son objectif c'est le décryptage du message reçu.

Les blocs émetteur et récepteur sont reliés par deux canaux, un canal sert à l'envoi du signal de synchronisation $y = x(k)$, et l'autre sert à l'envoi du message crypté par addition avec le signal $y(k)$, la période T des impulsions est choisie de manière à assurer une bonne synchronisation.

La figure (IV. 1) illustre le diagramme bloc de la transmission utilisant la synchronisation impulsive.

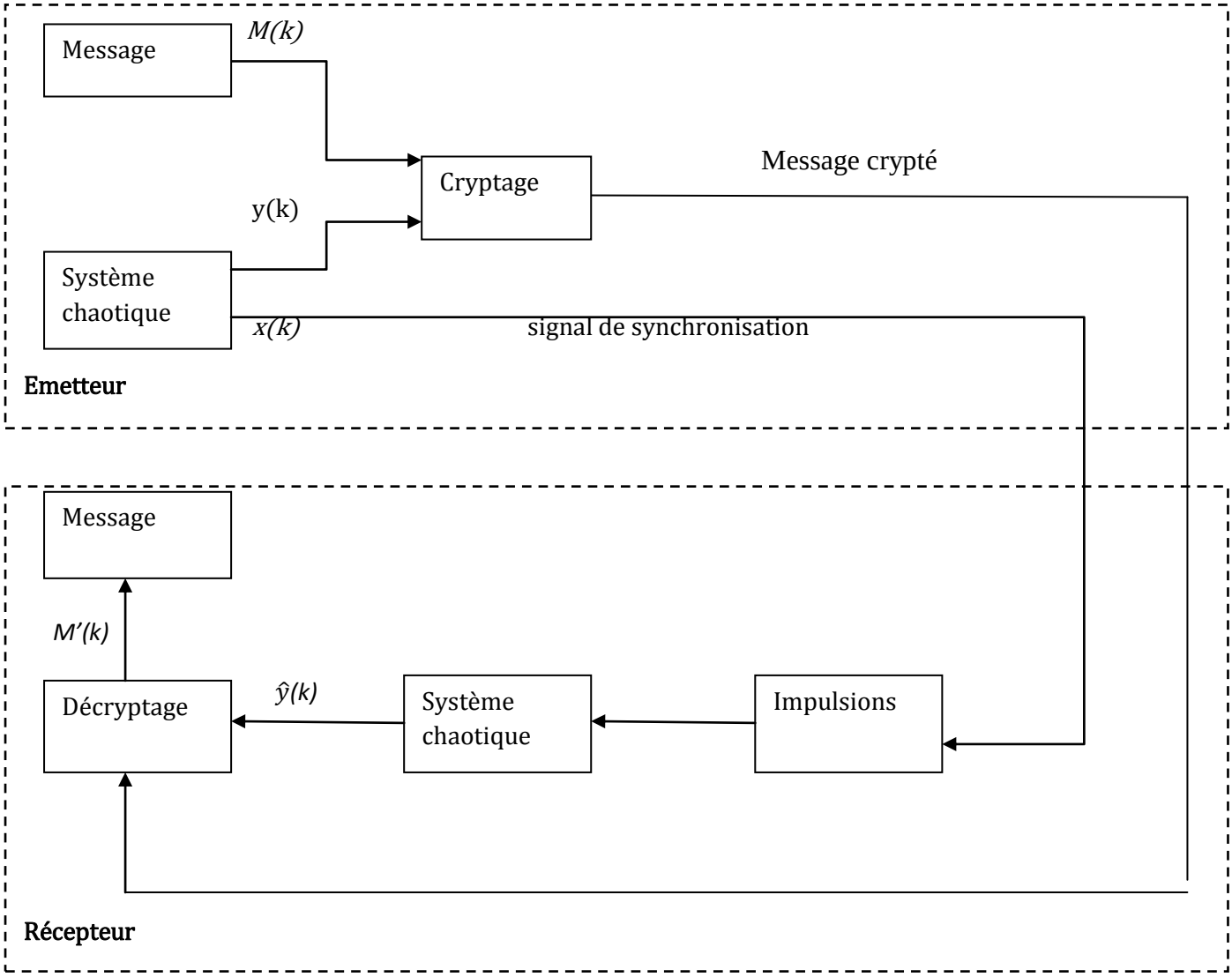


Figure (**IV**. **1**) **:** Diagramme bloc de la transmission utilisant la synchronisation impulsive

IV. 3 La carte Arduino

Arduino est un projet créé par une équipe de développeurs, composée de six individus : Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti. Cette équipe a crée le "système Arduino". C'est un outil qui va permettre aux débutants, amateurs ou professionnels de créer des systèmes électroniques plus au moins complexes [22].

Le système Arduino, nous donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, elle permet de programmer des systèmes électroniques. Le gros avantage de l'électronique programmée c'est qu'elle simplifie grandement les schémas électroniques et par conséquent, le coût de la réalisation, mais aussi la charge de travail à la conception d'une carte électronique [22].

Elle est basée sur un microcontrôleur AtmelATmega 328 ou ATmega 2560. Elle dispose dans sa version de base de 8 Ko de mémoire vive, et 256 Ko de mémoire flash pour stocker ses programmes. Elle peut être connectée à 54 entrées ou sorties numériques, dont 15 PWM.

IV. 3. 1 Présentation de La carte Arduino Mega 2560 [23]

La carte Arduino Mega 2560 est une carte à microcontrôleur basée sur un ATmega2560.

Cette carte dispose :

- de 54broches numériques d'entrées/sorties (dont 14 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)),
- de 16 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),
- de 4 UART (port série matériel),
- d'un quartz 16Mhz,
- d'une connexion USB,
- d'un connecteur d'alimentation jack,
- d'un connecteur ICSP (programmation "in-circuit")
- et d'un bouton de réinitialisation (reset).

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur. Pour pouvoir l'utiliser et la lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB).

La carte Arduino Mega 2560 est compatible avec les circuits imprimés prévus pour les cartes Arduino Uno, Duemilanove ou Diecimila.

Ce tableau nous donne les caractéristiques de la carte Arduino Mega

Microcontrôleur	ATmega 2560
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	54 (dont 14 disposent d'une sortie PWM)
Broches d'entrées analogiques	16 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	256 KB dont 8 KB sont utilisés par le bootloader
Mémoire SRAM (mémoire volatile)	8 KB
Mémoire EEPROM (mémoire non volatile)	4 KB
Vitesse d'horloge	16 MHz

Tableau IV. 1 : Caractéristique de la carte Arduino Mega 2560

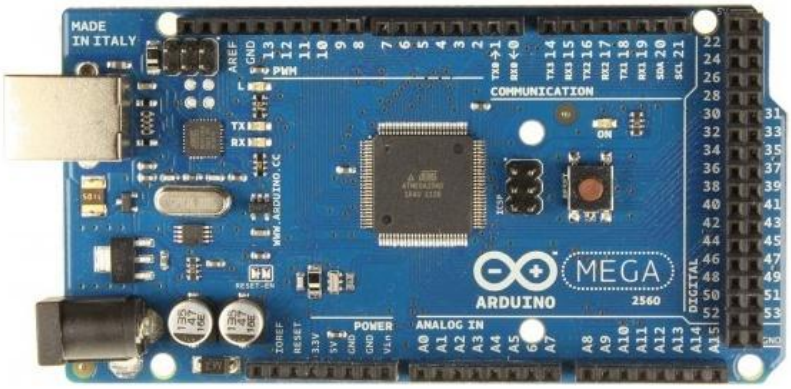


Figure (IV. 2) : Illustration d'une carte Arduino Mega 2560

IV. 3. 2 Le langage de programmation [23]

La programmation se fait dans un langage propre à Arduino dont la structure s'apparente aux langages C/C++. Mais lorsque on évoque une fonction Arduino, non standard C/C++, et pourtant reconnue et coloriée comme un mot-clé dans l'éditeur, on fait appel à une ou plusieurs bibliothèques rédigées en C ou C++ qui seront incluses à la compilation.

En développant le langage C/C++ de cette manière, les concepteurs de l'Integrated Development Environment (IDE) ont pu simplifier sa syntaxe et l'adapter aux possibilités de la carte. De nombreuses fonctionnalités de haut niveau sont ainsi proposées à l'utilisateur novice qui n'a plus à se soucier de la logique interne de microcontrôleur.

Il existe plusieurs possibilités d'interfaçage avec d'autres logiciels notamment **Matlab Simulink** grâce à plusieurs bibliothèques, on cite **Arduino_io** qui nous donne la possibilité d'utiliser la carte en temps réel avec Simulink et donc d'avoir accès aux entrées/sorties, et aussi la bibliothèque **Support Package for Arduino Hardware** qui inclue un compilateur qui permet la compilation direct du fichier MLD, et le téléversement sur la carte.

IV. 3. 3 Arduino_io

La bibliothèque **Arduino_io** permet d'utiliser la carte arduino comme une interface d'entrées/sorties. Ce package permet de communiquer Matlab ou Simulink avec la carte Arduino via un câble USB. Le principe consiste à pré-charger un programme dans la carte Arduino afin que celle-ci fonctionne en serveur. Ce programme consiste à "écouter" les requêtes envoyées via la liaison série (USB) et de répondre à ces requêtes en renvoyant l'état d'une entrée ou en modifiant l'état d'une sortie. Ces mêmes entrées/sorties sont vues dans Matlab comme des entrées logiques ou analogiques (utilisation du CAN) ou des sorties analogiques (mode PWM).

a) Installation du package arduino_io

-Télécharger la package Arduino_io sur le site mathworks.com.

-Décompressé le fichier zip téléchargé précédemment.

-Lancer Matlab 2013 en tant qu'administrateur et se placer dans ce même répertoire arduinoio.

-Sur le workspace on exécute la commande install_arduino.

-Relancer Matlab et Simulink.

-Dans la bibliothèque on retrouve les blocs dans arduinoio Library.

b) La bibliothèque Arduino_IO

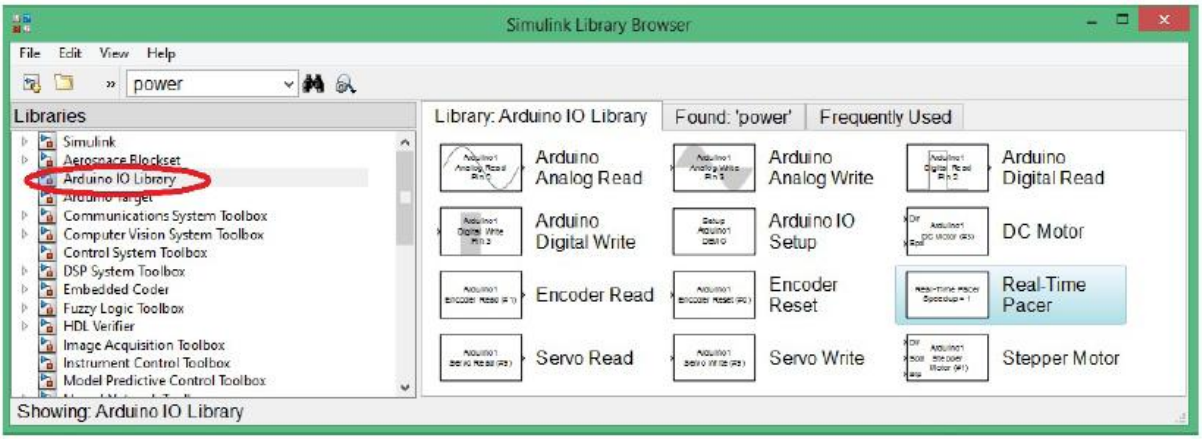


Figure (IV. 3) : Bibliothèque Arduino/Simulink

c) Programmation du « Firmware » de l'arduino

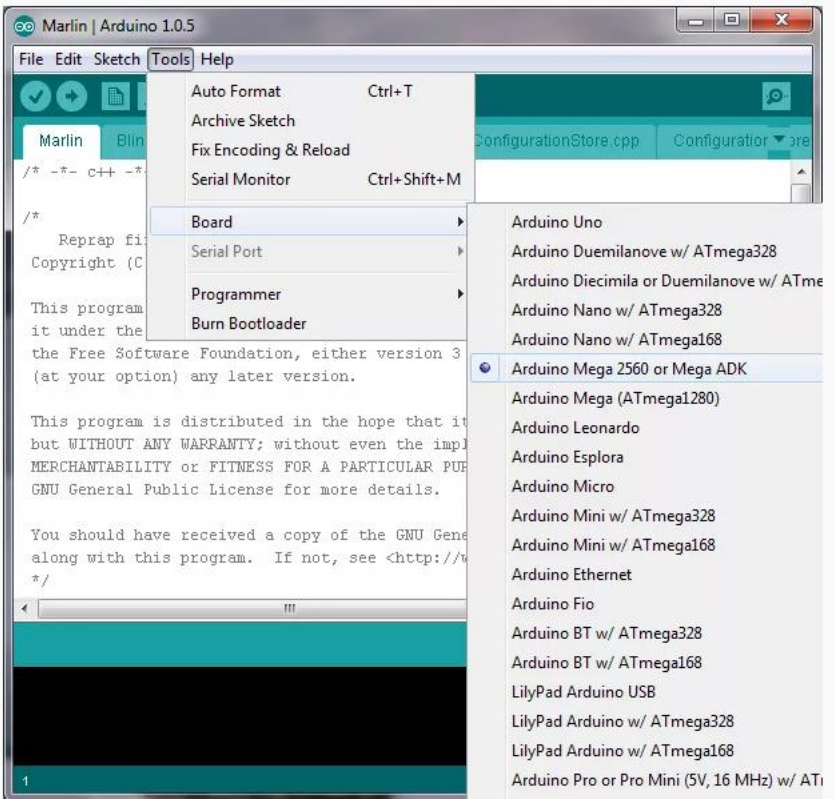


Figure (IV. 4) : L'interface du logiciel de programmation de la carte Arduino

Avant l'utilisation il faut d'abord programmer la carte arduino*srv.pde* (qui se trouve dans le sous répertoire « *pde* ». l'exécution se fait sur l'IDE arduino.

IV. 3. 4 Support Package for Arduino Hardware

Elle offre une possibilité de programmer une carte Arduino, la bibliothèque génère automatiquement le code à partir du modèle Simulink qui va fonctionner alors sur la carte Arduino de façon autonome.

L'installation est relativement simple sur les dernières versions de Matlab. On prend l'exemple de la version Matlab R 2013a.

Après le démarrage de Matlab en tant qu'administrateur sur la droite dans l'angle **Add-Ons** on sélectionne **Get Hardware Support Package**.

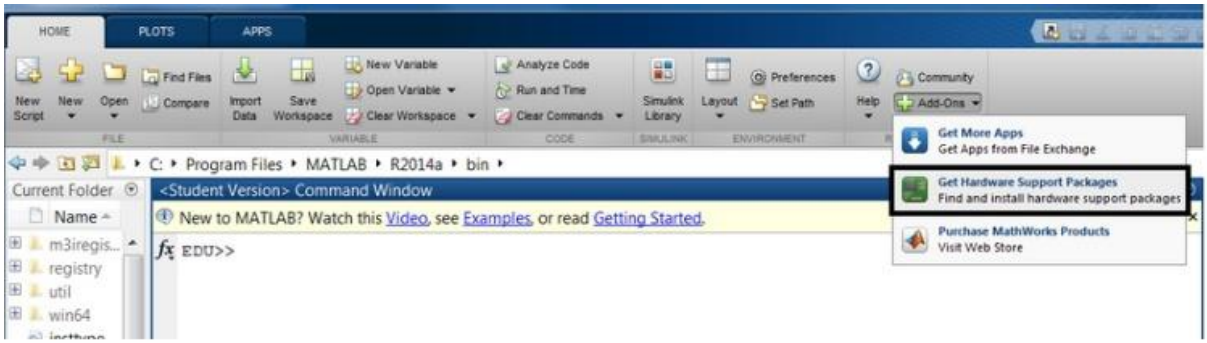


Figure (IV. 5) : Support Package Arduino

La fenêtre suivante s'ouvre, on clic sur « **Next** »

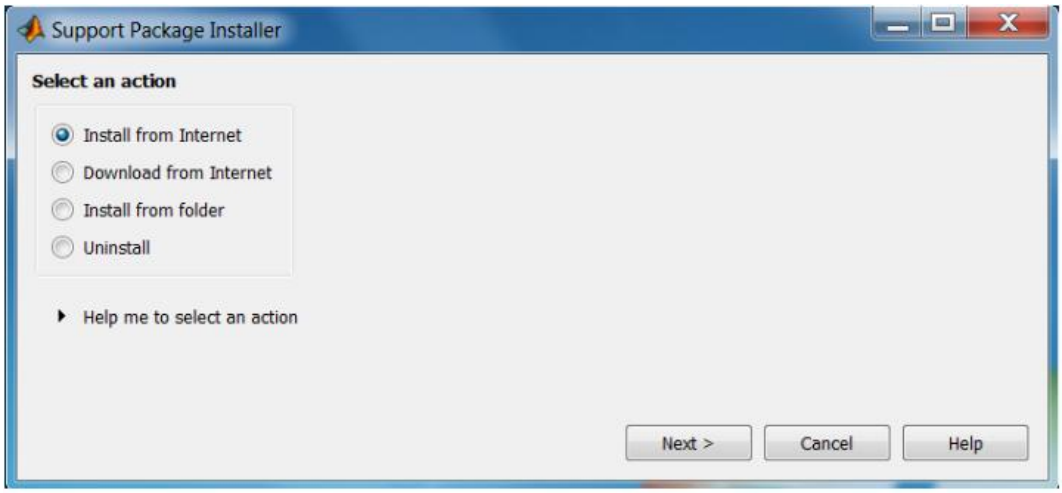


Figure (IV. 6) : Fenêtre de téléchargement

Après avoir sélectionné la source on passe à l'étape suivante qui consiste à sélectionner Arduino sur la liste, comme auparavant on clic sur « **Next** » pour télécharger et installé la bibliothèque comme le montre les figures ci-dessous :

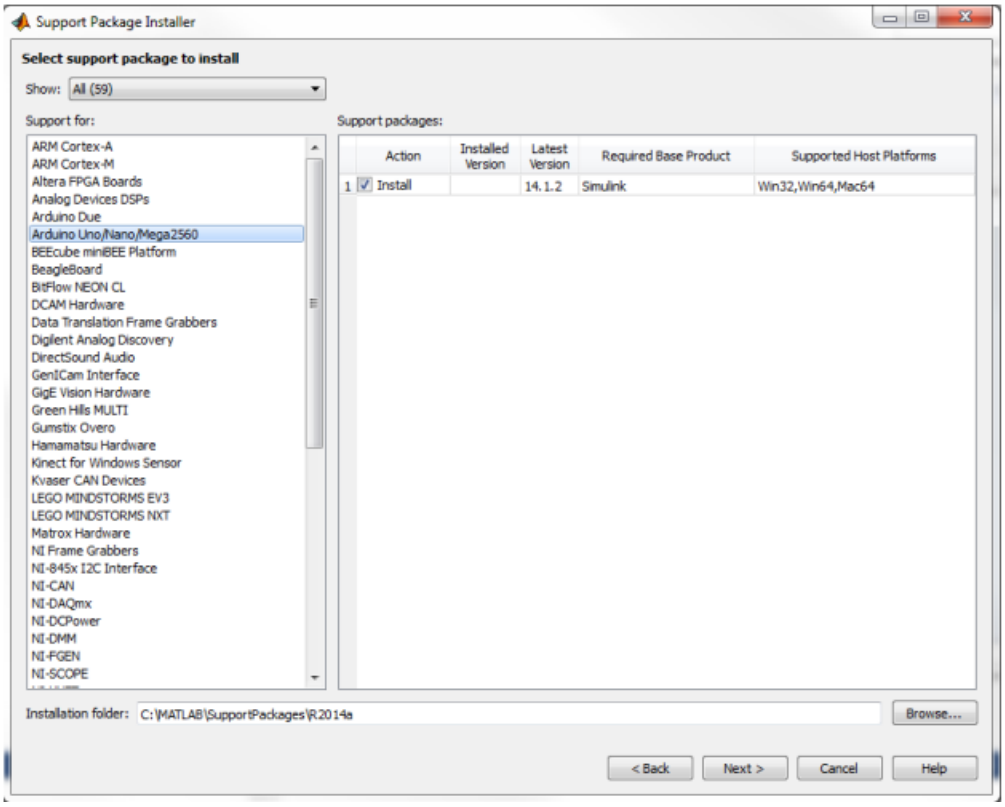


Figure (IV. 7) : Fenêtre de téléchargement

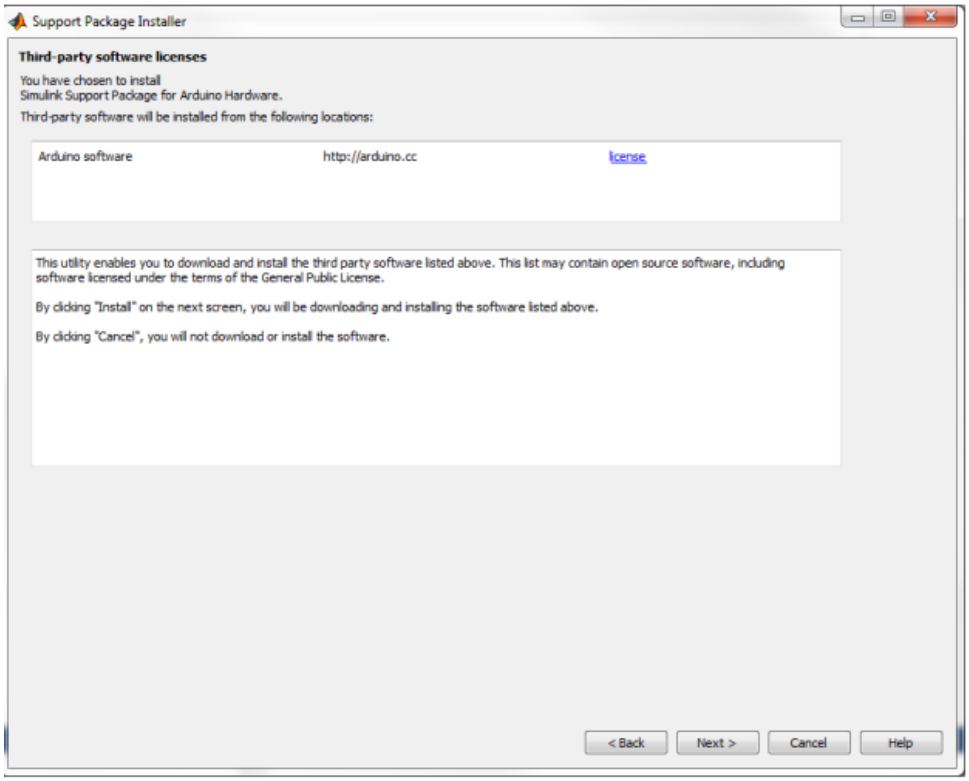


Figure (IV. 8) : Fenêtre de téléchargement

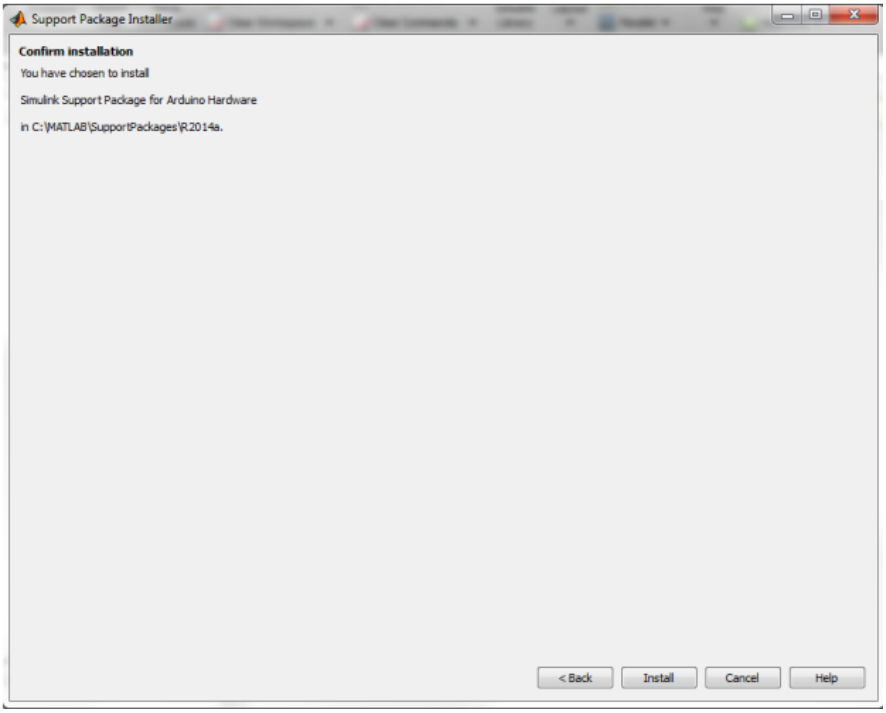


Figure (IV. 9) : Fenêtre de téléchargement

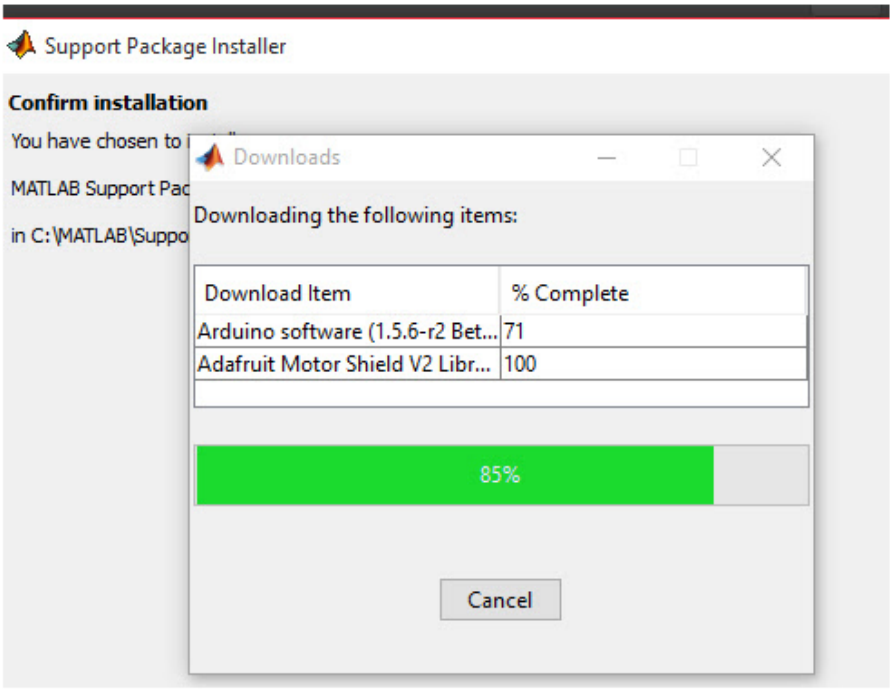


Figure (IV. 10) : Téléchargement et installation en cours

Une fois le téléchargement terminé, on retrouve la Tools box dans Simulink comme le montre la figure ci-dessous :

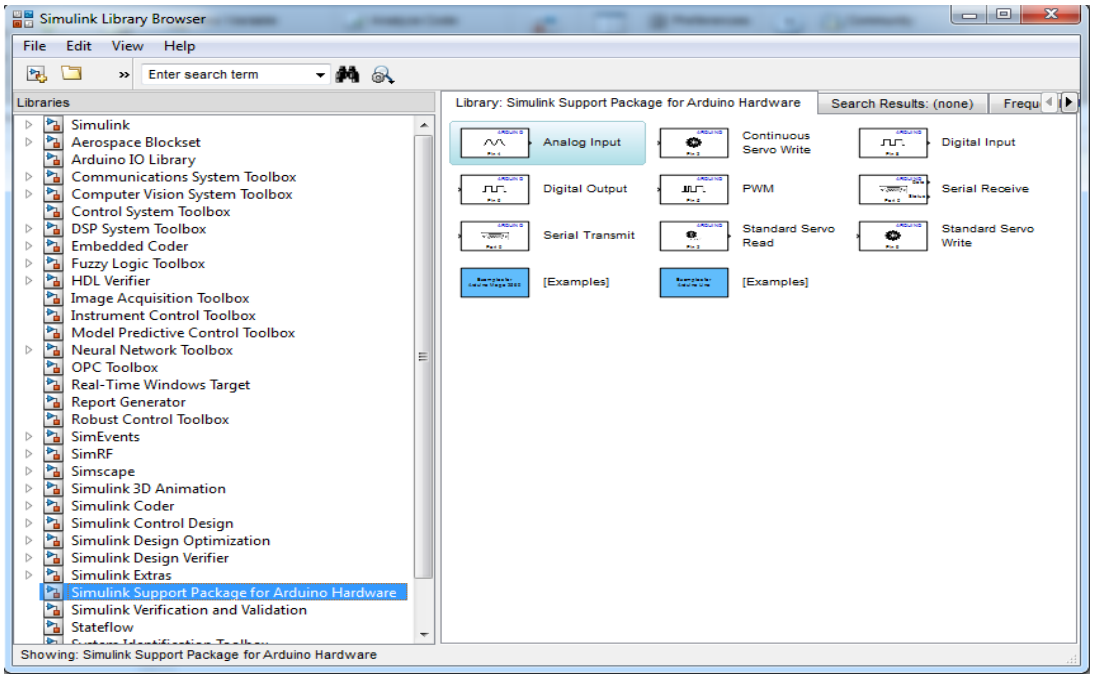


Figure (IV. 11) : Fenêtre de la Tools box dans Simulink

IV. 3. 5 La voie série UART [23]

Le principal objectif de la communication et de transmettre de l'information, elle nécessite l'utilisation d'un langage commun ou d'un code commun : type de liaison, vitesse, format des données, détection d'erreurs.

a) Emetteur et récepteur

Lorsque l'on communique des informations, il faut nécessairement un émetteur, qui va transmettre les informations à communiquer, et un récepteur, qui va recevoir les informations pour les traiter.

b) Les différents types de communication

On peut citer trois types de communications entre deux interlocuteurs :

- Simplex : L'un transmet des données, l'autre les reçoit sans répondre.
- Half-duplex : Chaque interlocuteur transmet des données à tour de rôle.
- Full-duplex : Chaque interlocuteur transmet en même temps que l'autre.

Arduino est capable de faire des communications de type full-duplex. Le protocole de communication est un ensemble de règles qui régissent la façon dont communiquent deux dispositifs entre eux, cela définit le rythme de la conversation, le débit de données, l'ordre des informations envoyées.

c) Application de la norme

Elle définit le signal électrique, et tout ce qui est lié à la connectique, le câblage, etc.

On prend l'exemple de la figue ($IV.12$) :

- Le premier câble est la référence électrique. Cela permet de prendre les mesures de tensions en se fixant un même référentiel.
- Les deux autres câble permettent la transmission des données l'un sert à l'envoi et l'autre à la réception.

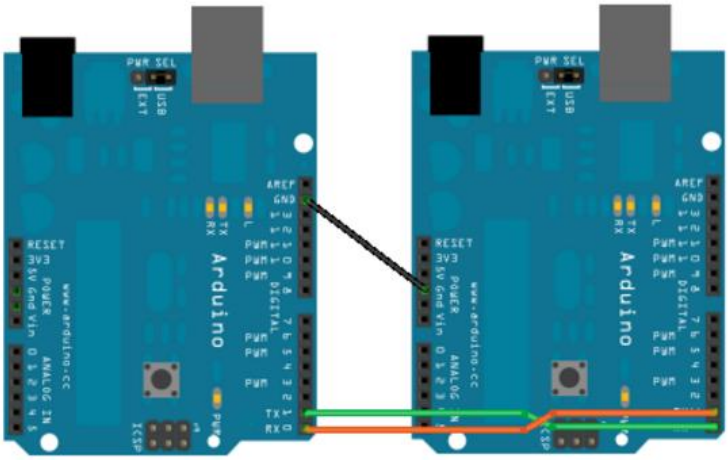


Figure (**IV**. **12**) **:** Schéma illustrant la liaison entre deux cartes Arduino

d) Les tensions utilisées

Les bits sont des niveaux de tension imposée par la norme, ces derniers sont cités dans le tableau suivant :

	Niveau logique 0	Niveau logique 1
Tension électrique minimale	+3V	-3V
Tension électrique maximale	+25V	-25V

Tableau (**IV**. **2**) **:** Les tensions utilisées pour la communication série

Toutes les tensions au-delà des valeurs imposé ou au-dessous sont hors normes, elles sont ignorées car cela permet d'éviter un certain nombre d'erreurs de transmission.

e) Les données

Les données qui transitent par la voie série sont transmises sous une forme binaire codée sur 8 bits selon la **table ASCII**. C'est-à-dire avec des niveaux logiques de 0 et 1.

La donnée commence avec un bit **Start** et se termine avec un bit **Stop** comme illustré dans la figure suivante :

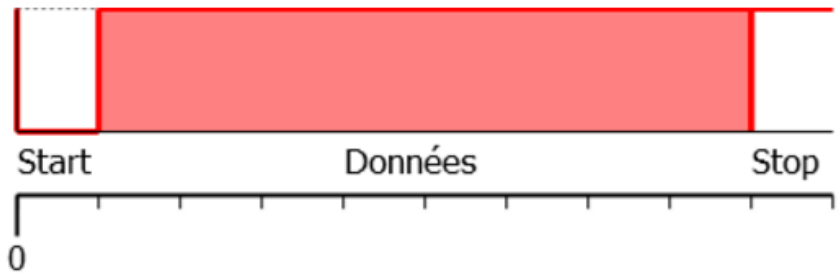


Figure (IV. 13) : Start et stop bit

Les données sont envoyées à l'envers, le bit de donnée qui vient après le bit Start s'appelle le **bit de poids faible** ou **LSB** en anglais pour less significant bit, et le **bit de poids fort** ou **MSB** viens après le stop bit.

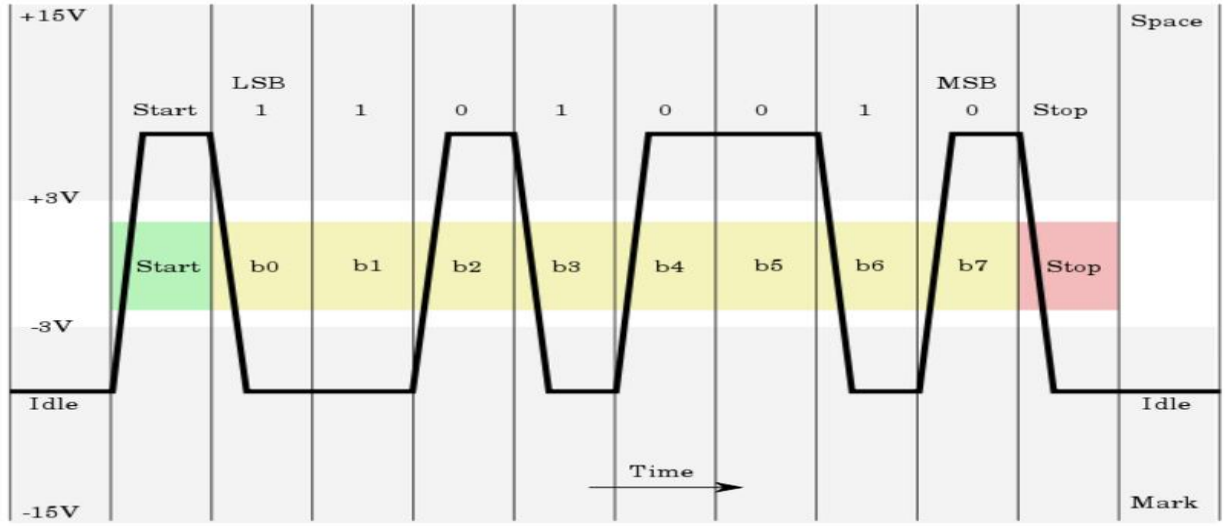


Figure (IV. 14) : Transmission d'un octet

f) La vitesse :

La norme définit la vitesse à laquelle sont envoyées les données. Elles sont exprimées en bits par seconde (bits/s). Elle préconise des vitesses inférieures à 20 000 bits/s. Sauf qu'en pratique, il est très courant d'utiliser des débits supérieurs pouvant atteindre les 115 200 bits/s. Quand on va utiliser la voie série, on va définir la vitesse à laquelle sont transférées les

données. Cette vitesse dépend de deux contraintes qui sont : la longueur du câble utilisé reliant les deux interlocuteurs et la vitesse à laquelle les deux interlocuteurs peuvent se comprendre.

Débit en bit/s	Longueur du câble en mètres (m)
2 400	900
4 800	300
9 600	150
19 200	15

Tableau (IV. 3) : Vitesse de transmission

Plus le câble est court, plus le débit est élevé car moins il y a d'affaiblissement des tensions et le risque de parasites. Quand la distance séparant les deux interlocuteurs est grande, la vitesse de communication diminue de façon significative.

g) Emulation du port série

La voie série est émulée à travers l'USB, c'est une liaison virtuelle de l'RS232, l'émulation est géré par un circuit intégré (entouré en rouge sur la figure ($VI.16$), et le gestionnaire du port USB et périphérique de l'ordinateur.

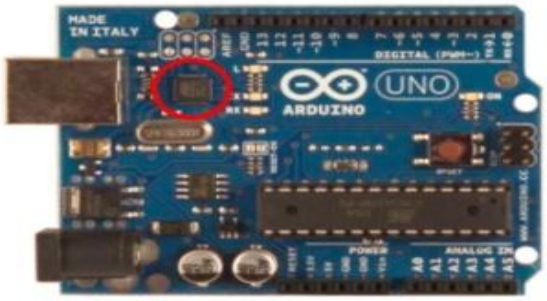


Figure (IV. 15) : Circuit émulateur du port série

IV. 4 Implémentation sur la carte Arduino

IV. 4. 1 Emetteur :

Le schéma Simulink de l'émetteur basé sur l'oscillateur chaotique de Lozi est présenté en Figure (IV,16).

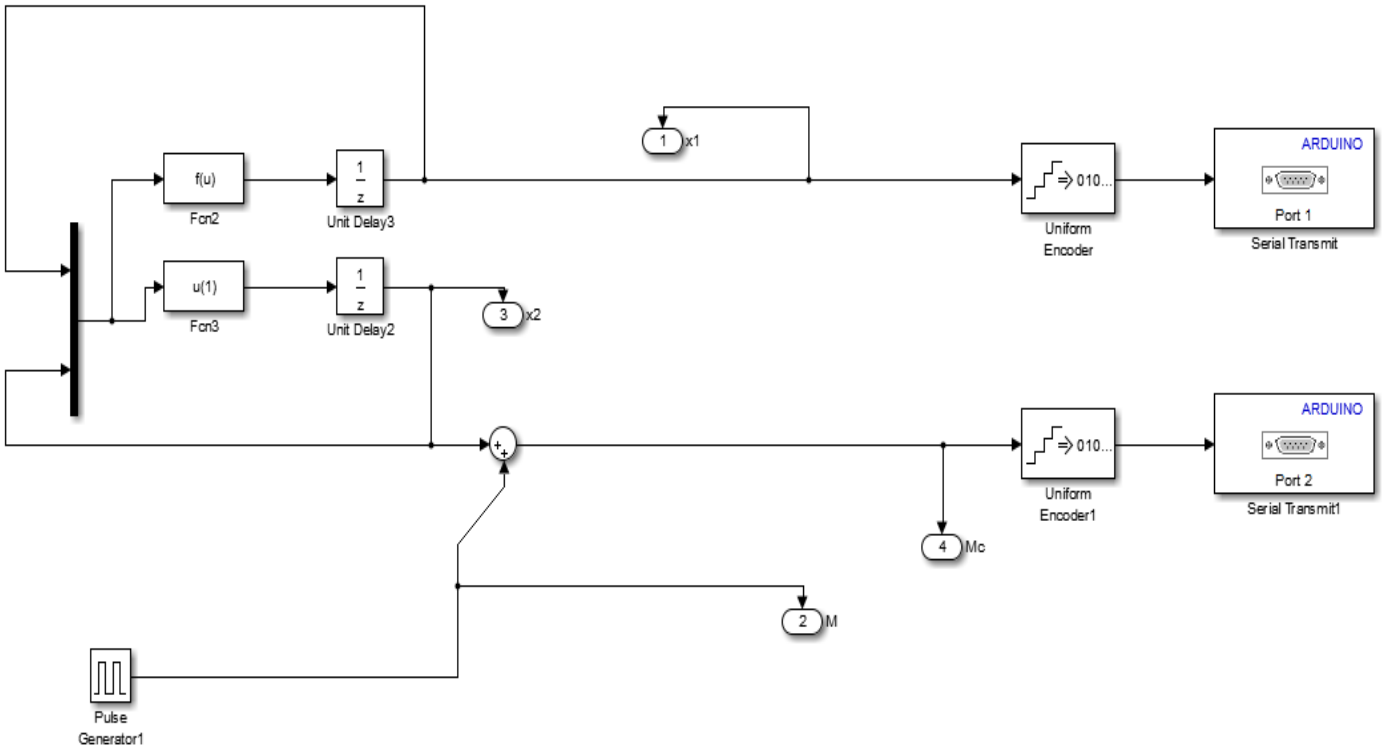


Figure (IV. 16) : Schéma Simulink de l'émetteur

La simulation du système émetteur sous Simulink permet de récupérer les trajectoires x1, x2 et l'attracteur étrange de Lozi présenté dans les figures qui suivent.

a) Visualisation des signaux

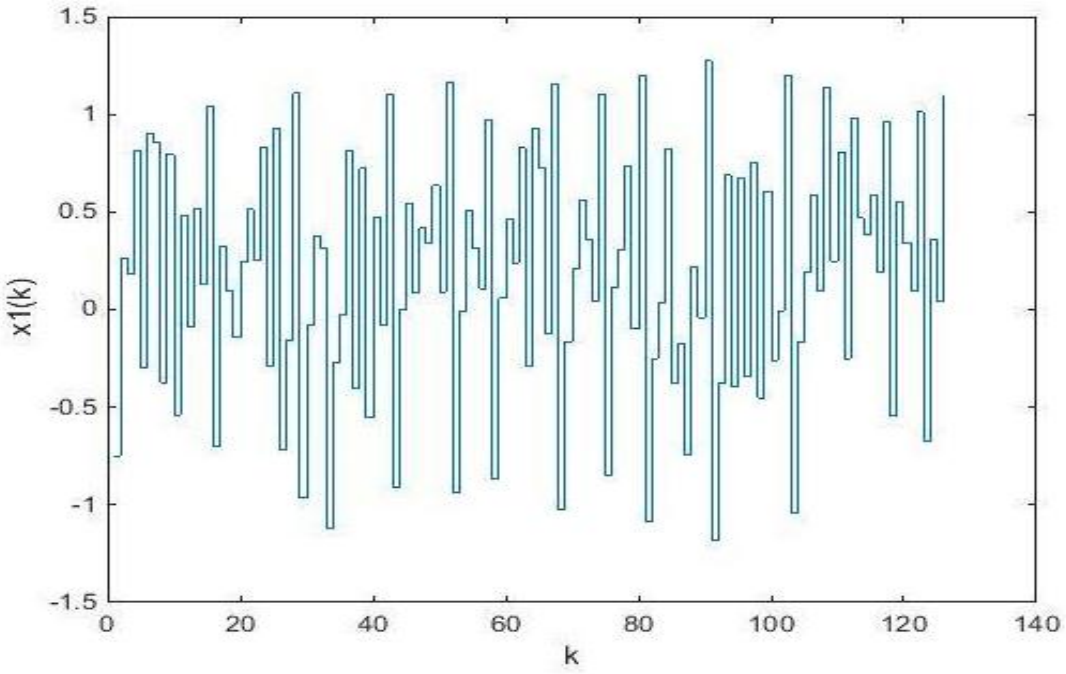


Figure (IV. 17) : Etat $x_1(k)$

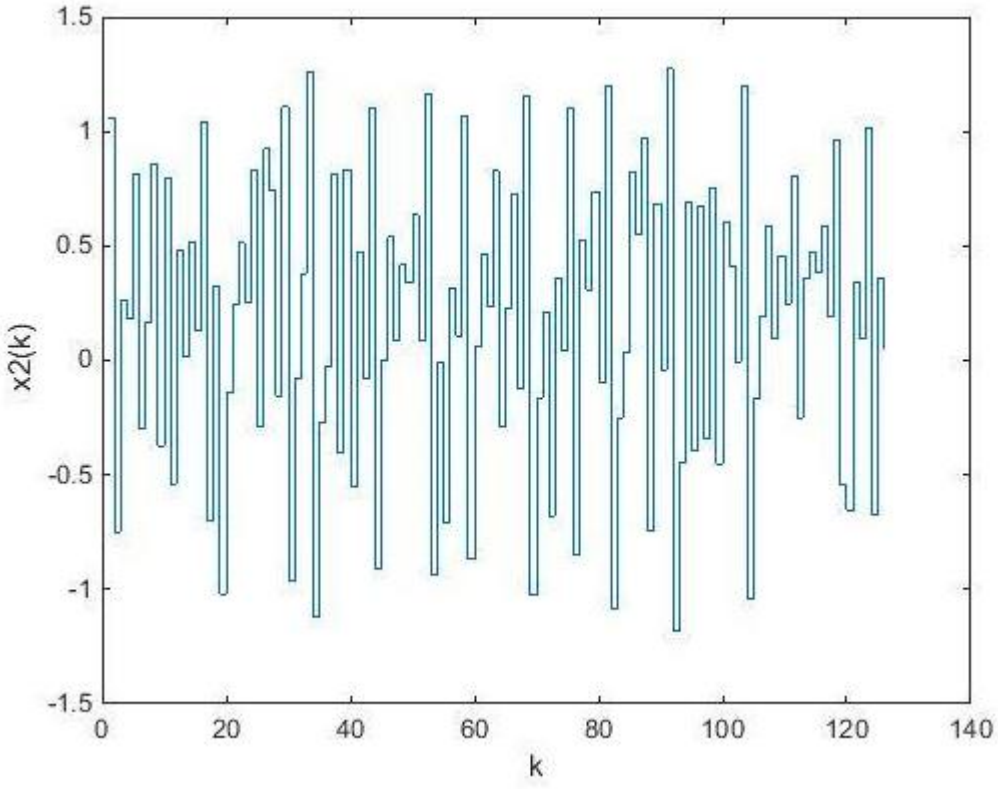


Figure (IV. 18) : Etat $x_2(k)$

-Attracteur

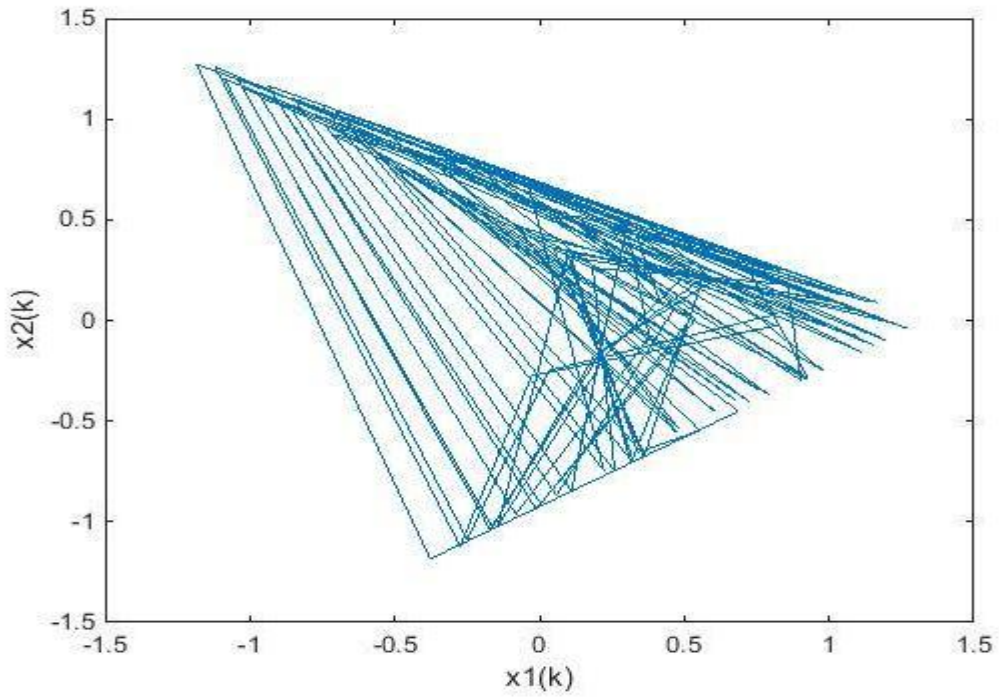


Figure (IV. 19) : Attracteur récupéré sur l'émetteur

Remarques

-Les figures (IV. 17), (IV. 18) montrent que les signaux $x_1(k)$ et $x_2(k)$ possèdent un aspect aléatoire .

-La figure (IV. 19) illustre la forme de l'attracteur de Lozi ce qui démontre l'aspect chaotique du système.

IV. 4. 2 Récepteur

Le schéma simulink du récepteur basé sur l'oscillateur chaotique de Lozi est présenté en Figure (IV,20).

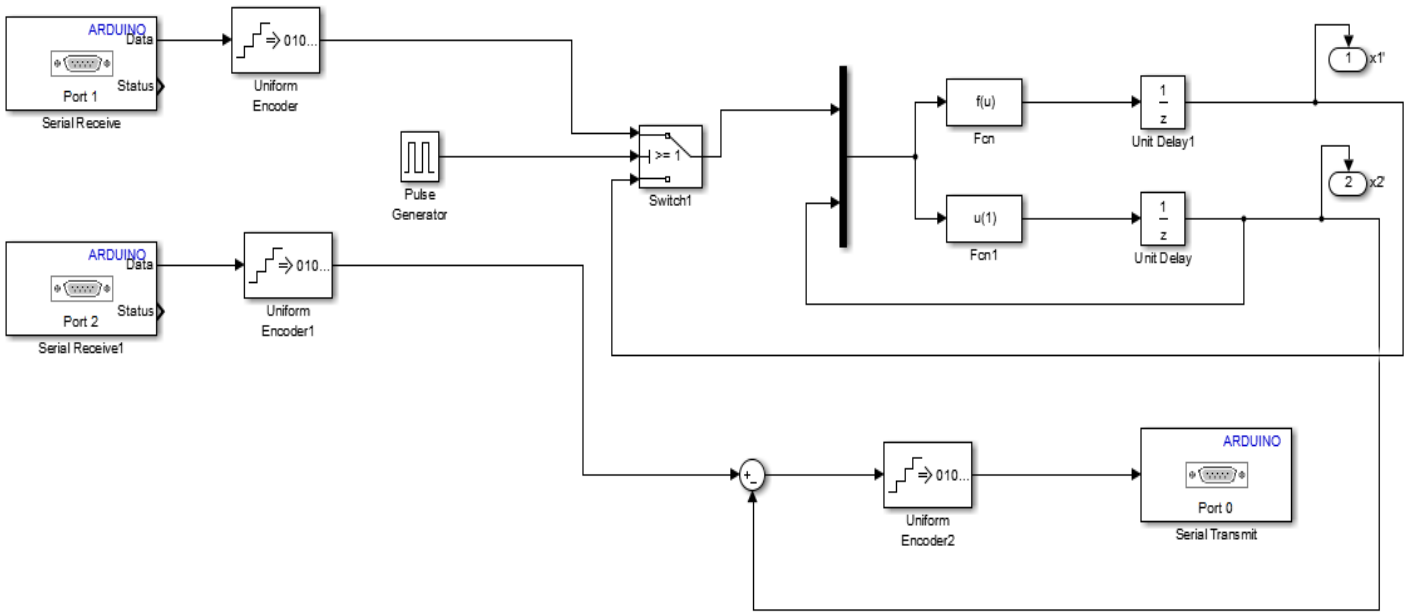


Figure (IV.20) : Schéma Simulink du récepteur

La simulation du système récepteur sous Simulink permet de récupérer les trajectoires $\hat{x}_1$ et $\hat{x}_2$ et l'attracteur étrange de Lozi présenté dans les figures qui suivent.

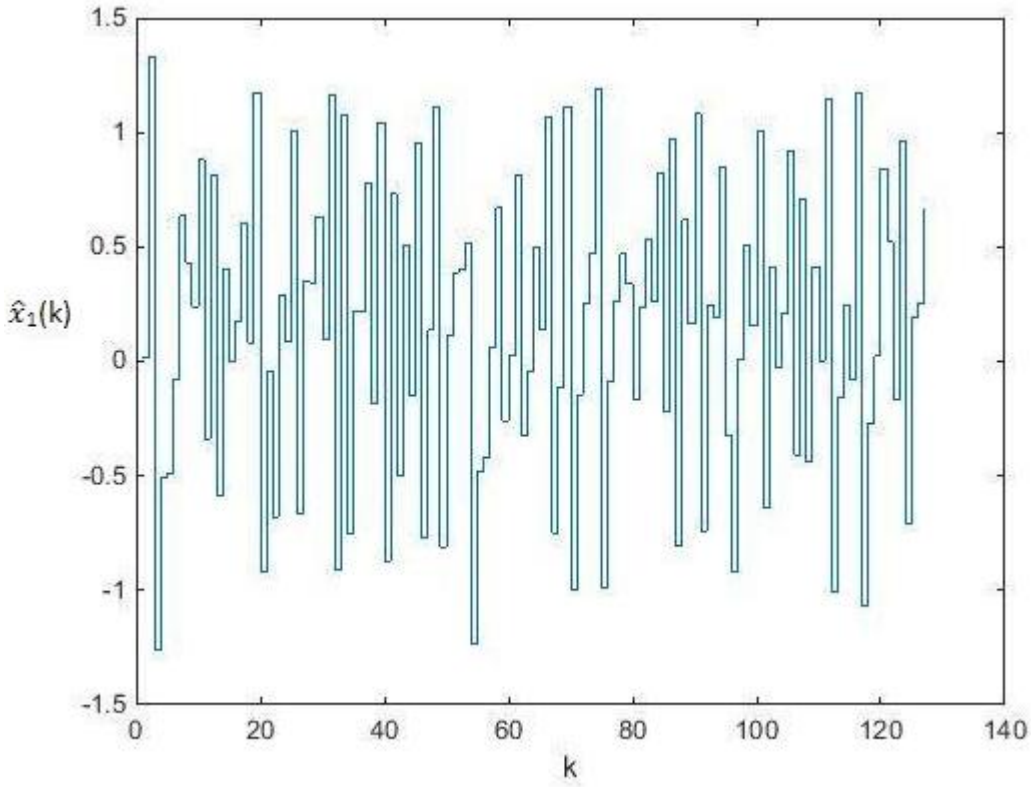


Figure (IV.21) : Etat $\hat{x}_1(k)$

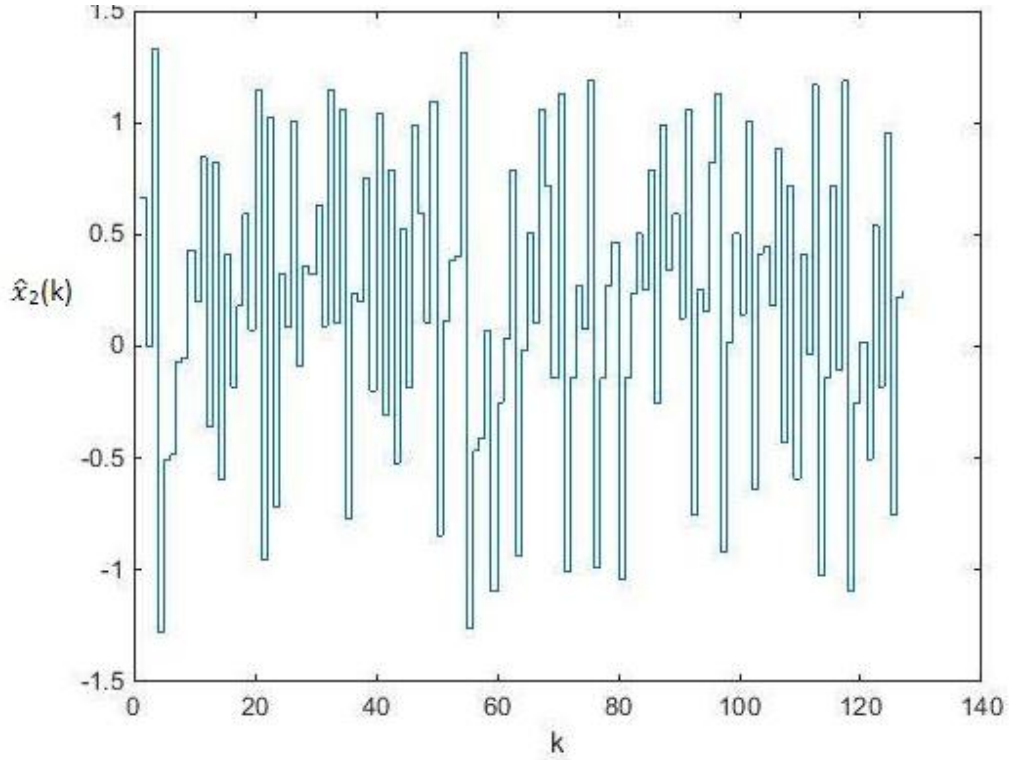


Figure (IV. 22) : Etat $\hat{x}_2(k)$

-Attracteur

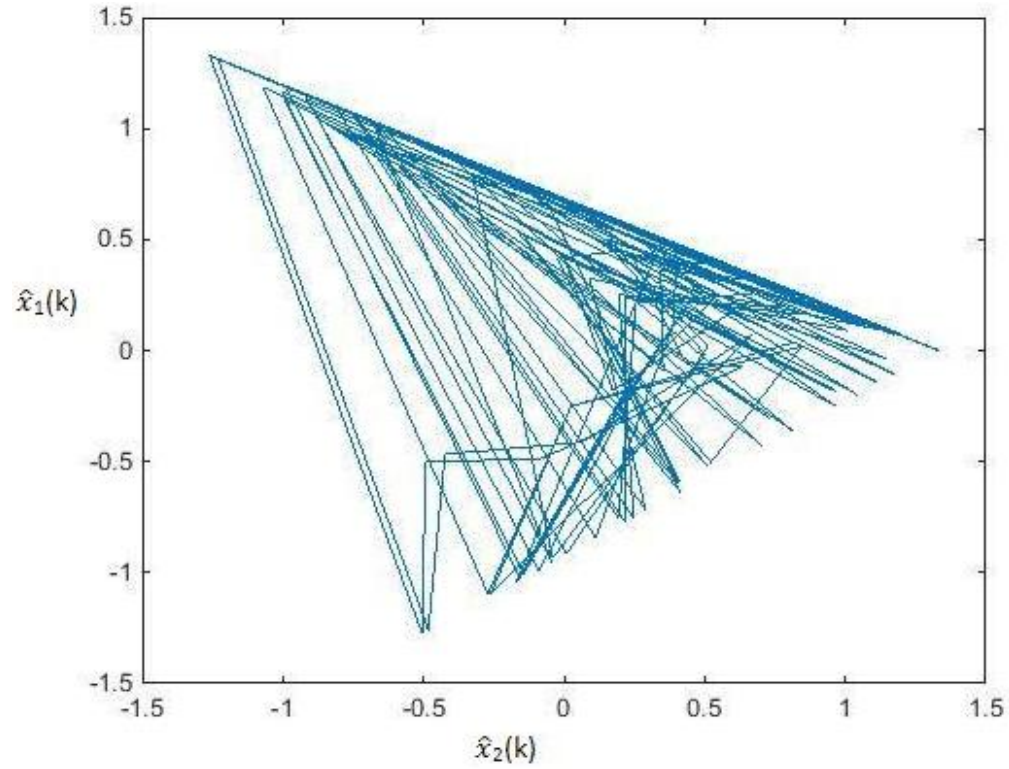


Figure (IV. 23) : Attracteur récupéré sur le récepteur

IV. 5 Visualisation des erreurs et des messages

Le message original est le suivant

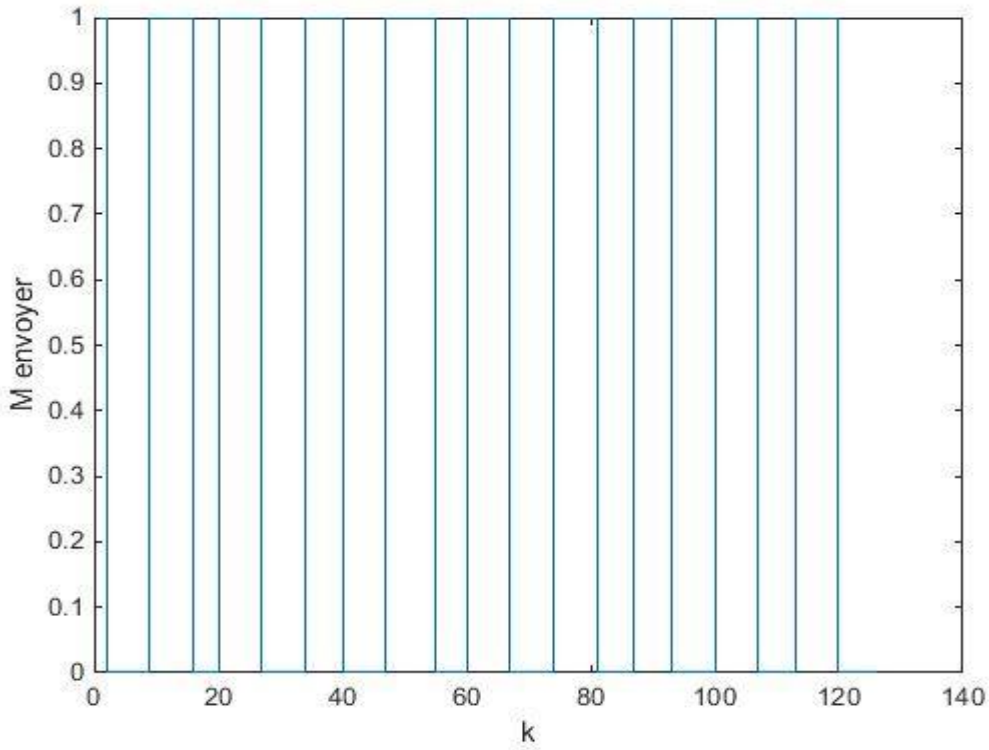


Figure (IV. 24) : Message envoyé

Le message crypté est le suivant

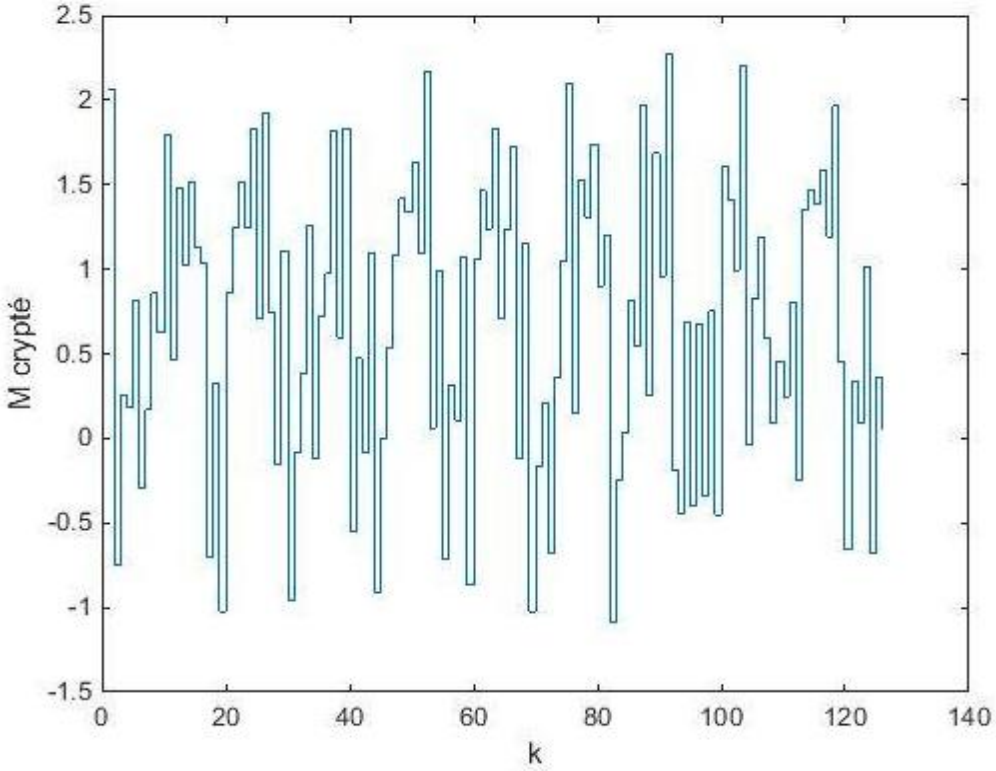


Figure (IV. 25) : Message crypté

Le message récupéré au niveau du récepteur est représenté sur la figure ci-dessous

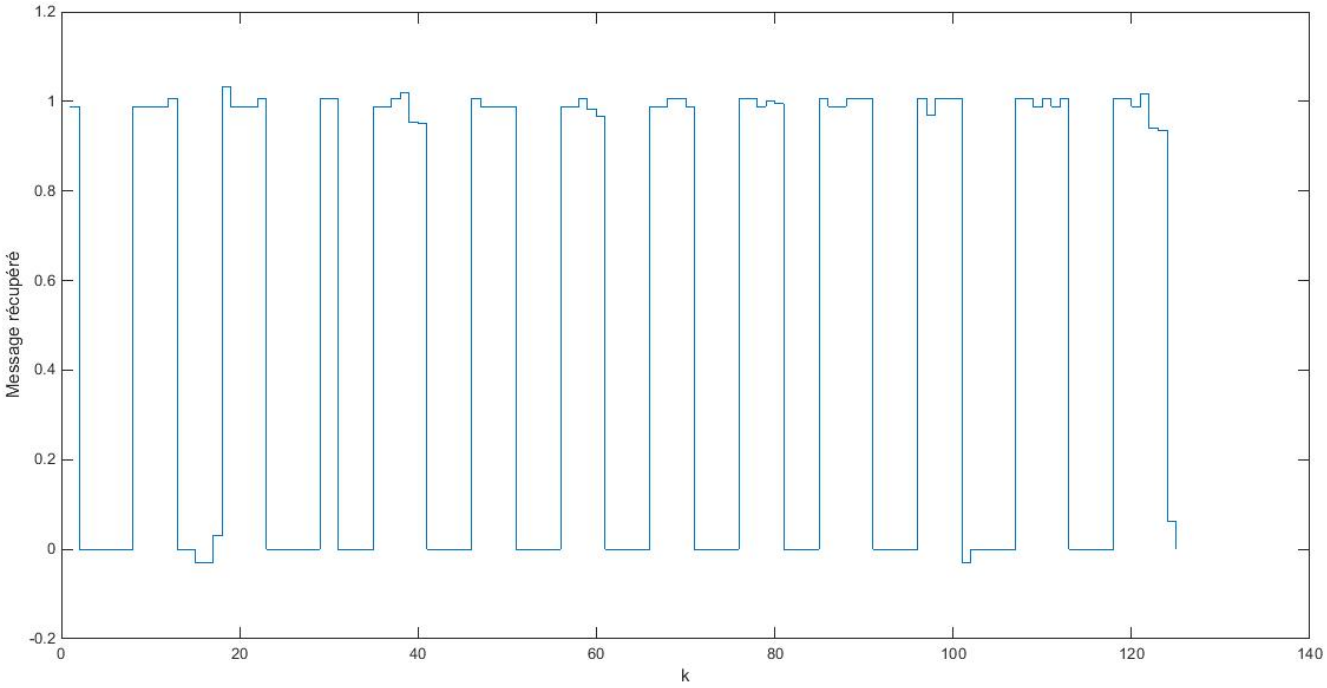


Figure (IV. 26) : Message récupéré

4/ Erreur de synchronisation

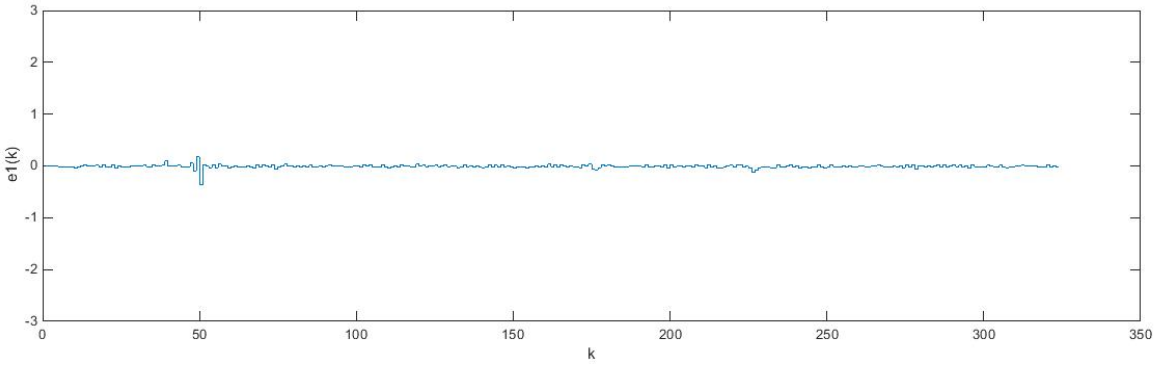


Figure (IV. 27) : Erreur de synchronisation entre $\boldsymbol{x_1(k)}$et $\boldsymbol{\hat{x}_1(k)}$

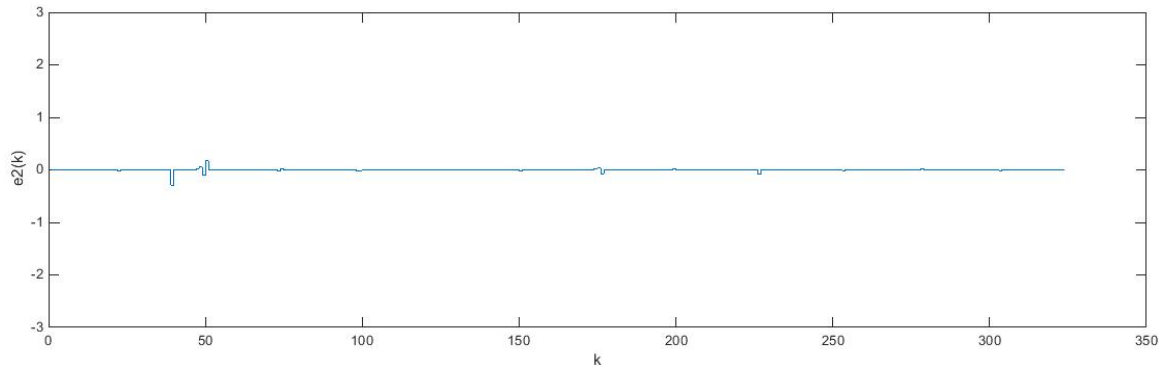


Figure (IV. 28) : Erreur de synchronisation entre $\boldsymbol{x_2(k)}$et $\boldsymbol{\hat{x}_2(k)}$

Remarque

D'après les figures ci-dessus, on a constaté que le message envoyé à été récupéré dans son ensemble à 95 %, cela montre que la synchronisation impulsive qu'on a étudiée et réalisée a bien fonctionnée.

IV. 6 Conclusion

Dans ce chapitre nous avons présenté les différents résultats pratiques qu'on a obtenus. Les différents blocs à savoir : l'émetteur et le récepteur ont été simulés sous Matlab et implémentés dans des cartes Arduino Méga en utilisant la bibliothèque package for Arduino Hardware de Matlab. Ceci nous a permis de tester notre système de transmission, en effet, le message envoyé depuis l'émetteur a bien été récupéré par le récepteur.

Conclusion Générale

Conclusion générale

Dans ce mémoire, nous avons présenté et étudié une transmission de données sécurisée utilisant le chaos. Ce travail a permis d'explorer le phénomène chaotique. Nous avons également présenté tous les points essentiels concernant le chaos, telles que sa définition et ses caractéristiques.

Le premier chapitre a porté sur les caractéristiques des systèmes chaotiques comme la sensibilité aux conditions initiales, le déterminisme ainsi que l'aspect aléatoire de leurs trajectoires. Ajoutons à cela, nous avons aussi expliqué et montré la transition vers le régime chaotique. A la fin de ce chapitre, des exemples célèbres tels que le modèle de Lorenz et le modèle de Hénon ont été mis exergue afin d'illustrer concrètement toutes les notions précédemment citées.

Dans le deuxième chapitre, nous avons exposé les différentes méthodes de synchronisation des systèmes chaotiques les plus rencontrées dans la littérature. En effet cette étape est indispensable pour une transmission de données. Par la suite, nous avons passé en revue les notions de cryptage et de décryptage. Nous avons aussi expliqué comment chiffrer un message en le noyant dans un signal chaotique, puis comment l'envoyer dans un canal public. Nous avons conclu cette partie en montrant comment récupérer le message transmis en faisant une simple soustraction.

Dans le troisième chapitre, nous avons décrit et élaboré notre système de transmission. Nous avons notamment expliqué le fait que nous ayons utilisé comme émetteur le système de Lozi. La conception du récepteur s'est faite par un système de Lozi identique à celui de l'émetteur que nous avons synchronisé à l'aide d'un observateur impulsif. Ensuite nous avons simulé le système de transmission sur Matlab Simulink, et nous avons ensuite inséré un message dans l'émetteur en l'additionnant à un des états du système, et nous avons terminé par la récupération du message au niveau du récepteur.

Le quatrième chapitre met en évidence notre réalisation pratique. Cette dernière s'est faite en utilisant des cartes Arduino Mega. Nous avons pu obtenir des résultats satisfaisants et similaires par rapport à l'étude théorique dans le troisième chapitre.

Nos résultats montrent bien que la transmission sécurisée utilisant le chaos, étudiée par simulation fonctionne en pratique. Ces résultats laissent entrevoir quelques perspectives et élargissements et énormément de possibilités de développement dans le domaine de la communication.

Conclusion générale

En conclusion de ce mémoire : le chaos est un domaine en plein développement et en perpétuelle métamorphose, proposant des innovations de diverses origines. Aujourd'hui la plupart des recherches se concentrent sur l'utilisation du chaos dans des crypto systèmes en vue d'apporter une amélioration de plus en plus exigeante (temps de chiffrement, sécurité).

Annexe A

Annexe A | Quelques définitions

A.1 Définition (trajectoire et flot)

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

Donc toute solutions $\varphi_t(x)$ d'un système autonome considérée comme un ensemble de trajectoires avec différentes conditions initiales, est appelée **flot.**

A.2 Définition (Degré de liberté)

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

A.3 Définition (Dimension fractale)

Pour un système dissipatif, le chaos implique la présence d'un attracteur étrange (dans l'espace de phase) dont la dimension est fractale. Cette dimension fractale donne le nombre de degrés de liberté du système et renseigne sur sa complexité.

A.4 Définition (Le bassin d'attraction)

Le bassin d'attraction est l'ensemble des points de l'espace des phases qui sont sous l'effet de l'attracteur. C'est-à-dire que toutes les trajectoires qui commencent à ces points tendent vers l'attracteur après un temps fini.

A.5 Définition (La fonction d'autocorrélation)

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

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

$$C(\zeta) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} x(t)x(t+\zeta)d\zeta \qquad A.1$$

La fonction d'autocorrélation est construite en faisant varier progressivement l'intervalle ζ, par conséquent si $x(t)$ est constant, périodique ou quasi-périodique $C(\zeta)$ reste non nul quand

$\zeta \to \infty$ car le spectre de puissance est formée de rais distinctes. En particulier pour les oscillations périodiques.

$C(\zeta)$ oscille entre 1 et -1 , dans le cas des oscillations chaotiques ou le spectre de puissance comporte une partie continue, $C(\zeta) \to 0$, un processus chaotique est tel que sa fonction de corrélation décroit vers 0 quand l'horizon tend vers l'infini. Cette propriété assure que les solutions sont divergentes les unes des autres. Si la fonction de corrélation est nulle pour des horizons non nuls, alors c'est un processus non corrélé, et on parle de « Bruit blanc déterministe ».

Annexe B

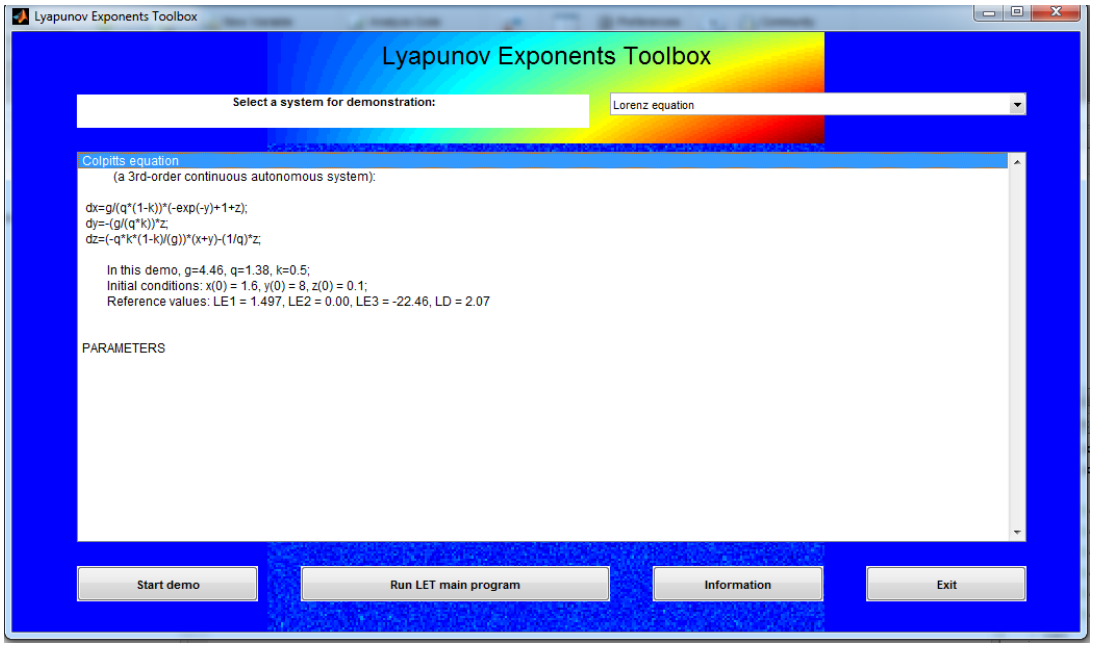
Annexe B LET (Lyapunov Exponents Toolbox)

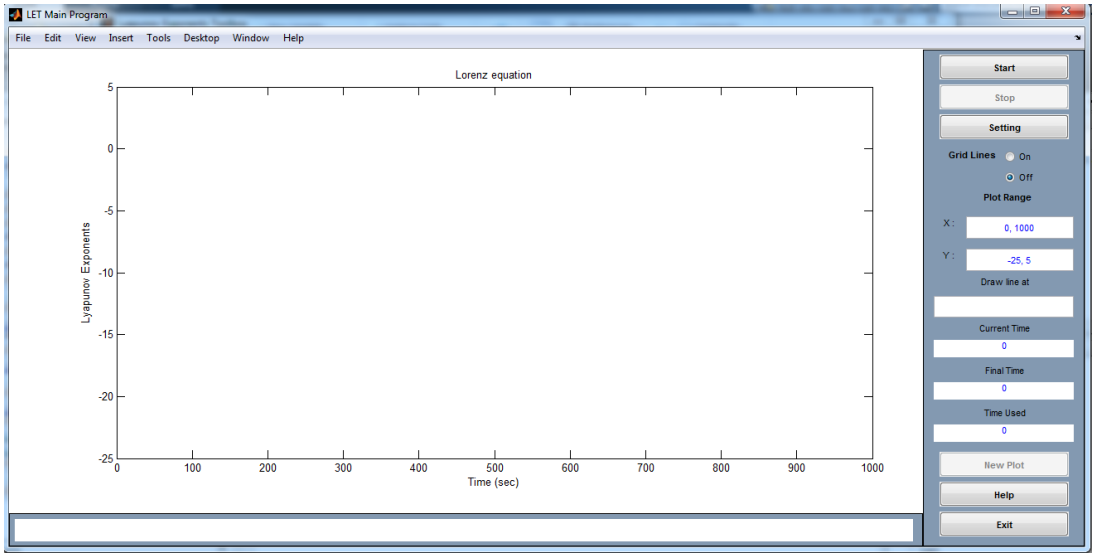
B.1 LET

Lyapunov Exponents Toolbox (LET) offre une interface graphique pour les utilisateurs afin de déterminer les ensembles complets d'exposants de Lyapunov et Lyapunov dimension des systèmes chaotiques continus et discrets.

Cette boite à outils ne peut fonctionner que sur Matlab 5 ou versions supérieurs de Matlab. Il a été testé sous Windows et Unix et peut aussi fonctionner sur d'autres plateformes.

La fenêtre sur la figure ci-dessous s'ouvre et on aura le choix entre plusieurs systèmes chaotiques, préprogrammé. A noter que le système de Lozi n'est pas programmé, mais la modification des algorithmes est possible par conséquent on peut le programmé.





B.2 Comment utiliser le programme :

Pour exécuter le programme, entrez « let » dans Matlab fenêtre workspace. Afin d'exécuter le programme correctement, tous les fichiers de cette boite à outils doivent être dans le dossier de travail. Quand la fenêtre GUI apparait, les utilisateurs peuvent exécuter un programme de démonstration en appuyant sur le bouton '' Start demo'' ou de démarrer le programme principal en appuyant sur le bouton '' RUN LET main program ''.

Cette boite à outils fournit certains systèmes chaotiques connus comme Lorenz ou Hénon. Ainsi les utilisateurs peuvent les sélectionnés dans le menu pop-up.

Pour calculer les exposants de Lyapunov et la dimension d'un système, suivez les instructions suivantes :

1. Ecrire une fonction dans l'ODE qui décrit le système spécifié.
2. Entrer « let » dans Matlab fenêtre de commande.
3. Appuyer sur le ''Run'' dans la fenêtre de démarrage.
4. Appuyer sur le bouton ''Paramètres'' dans la fenêtre principale.
5. Entrer les paramètres souhaités dans la fenêtre de réglage.
6. Une fois terminer, appuyer sur le bouton ''OK''.
7. Appuyer sut le bouton ''Démarrer'' dans la fenêtre principale pour commencé le calcul.

B.3 A propos de l'auteur

L'auteur de LET est Steve Wai Kam SIU, qui travaille actuellement à l'Université de la ville de Hong Kong. Il est au sein du département de génie électronique. Les intérêts de recherche de Steve comprennent l'application du chaos pour sécuriser les communications, le contrôle du chaos ainsi que la synchronisation des systèmes chaotiques non linéaires (utilisations des boucles verrouillées).

Annexe C

C.1 Alimentation de la carte arduino MEGA 2560

La carte Arduino Mega 2560 peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées GND (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Uno est entre 7V et 12V.

La carte Arduino Mega2560 diffère de toutes les cartes précédentes car elle n'utilise par le circuit intégré FTDI usb-vers-série. A la place, elle utilise un Atmega8U2 programmé en convertisseur USB-vers-série.

Les broches d'alimentation sont les suivantes :

- VIN. La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). Vous pouvez alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- 5V. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

- 3.3V. Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA
- GND. Broche de masse (ou 0V).

C.2 Mémoire

L'ATmega 2560 a 256Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le bootloader). L'ATmega 2560 a également 8 ko de mémoire SRAM (volatile) et 4Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM).

C.3 Entrées et sorties numériques

Chacune des 54 broches numériques de la carte Mega peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions pinMode(), digitalWrite() et digitalRead() du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digitalWrite (broche, HIGH).

C.4 Broches analogiques

La carte Mega2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (c'est-à-dire sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction analogRead() du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction analogReference() du langage Arduino.

Note : les broches analogiques peuvent être utilisées en tant que broches numériques.

C.5 Communication

La carte Arduino Mega2560 dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega2560 dispose de quatre UARTs (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'ATmega 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire. Cependant, sous Windows, un fichier .inf est requis.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré ATmega8U2 utilisé en convertisseur USB-vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

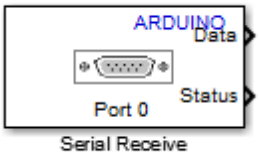
Une librairie Série Logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte UNO.

C.6 Protection du port USB contre la surcharge en intensité

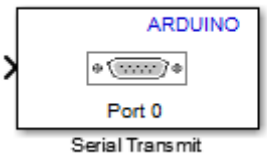
La carte Arduino Mega2560 intègre un polyfusible réinitialisable qui protège le port USB de votre ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.

Annexe D

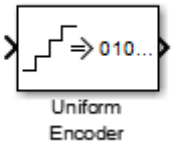
Annexe D | Liste descriptive des blocs Matlab Simulink utilisés



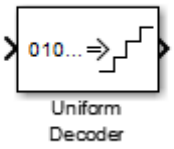
Arduino Serial Receive : reçoit un octet de données par période de la mémoire tampon du port série spécifié, le bloc dispose de deux sorties **data** et **status**, la sortie **status** se met a 1 lorsque une donnée est disponible sur la sortie **data**.



Arduino Serial Transmit : Envoyer des données en mémoire tampon sur le port série spécifié. Le bloc accepte des données sous forme de vecteur ou scalaire **uint8.**



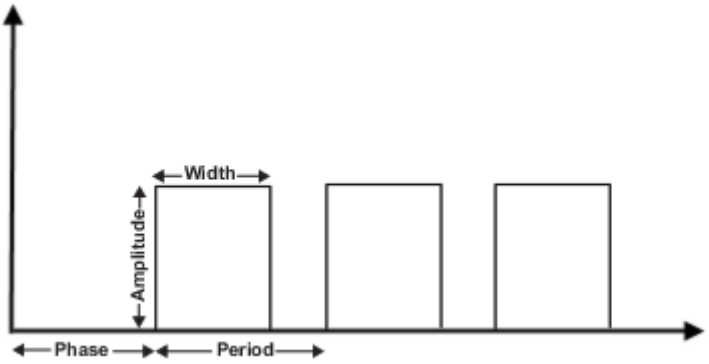
Uniform Encoder : Ce bloc effectue les deux opérations suivantes sur chaque échantillon d'entrée ou matrice, quantifie la valeur avec la même précision et encode la valeur à virgule flottante à une valeur entière.



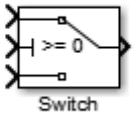
Uniform decoder : fait l'opération inverse de **Uniform Encoder** et reconstruit, quantifie les valeurs a virgules flottante à partir de l'entrée entière codé, les entrées peuvent être des valeurs réelles ou complexes, les types de données traiter (**uint8, uint16, uint32, int8, int16,ou int32**)

Annexe D | Liste descriptive des blocs Matlab Simulink utilisés

Pulse Generator : Le bloc de générateur d'impulsions génère des impulsions carrées à intervalles réguliers. Les paramètres de forme d'onde du bloc, **Amplitude, largeur d'impulsion, Période,** et **retard de phase,** permettent de déterminer la forme d'onde de sortie. Le schéma suivant montre comment chaque paramètre affecte la forme d'onde.



Le générateur d'impulsions peut émettre scalaire, vecteur, ou matrice de tout type de données réel.

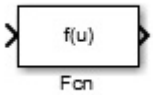


Switch : le bloc fait passer la première entrée ou la troisième entrée sur la base de la valeur de la seconde entrée. Les première et troisième entrées sont appelées des entrées de données. La seconde entrée est appelée entrée de commande. Les tailles des deux entrées de données peuvent être différentes, ce bloc ne supporte pas les signaux d'entrée de taille variable, l'entrée de commande peut être de tout type.

Mux : Le bloc multiplexeur combine en une sortie unique plusieurs entrées. Une entrée peut être un signal scalaire ou vectoriel, toutes les entrées doivent être du même type de données numérique.

Annexe D | Liste descriptive des blocs Matlab Simulink utilisés

Sum, Add, Subtract, sum of Elements : Le bloc **Sum** effectue une addition ou une soustraction sur ses entrées. Ce bloc peut sommer ou soustraire des scalaires, vecteurs, ou les éléments d'une matrice. Il peut aussi inverser les éléments d'un signal.



Fcn : Le bloc Fcn applique l'expression mathématique spécifiée à son entrée. L'expression peux inclure une ou plusieurs de ces composants :

- **u-** L'entrée du bloc, Si u est un vecteur, **u(i)** représente le i ème élément du vecteur ; **u(1)** ou **u** représente le seul premier élément.
- Les constantes numériques.
- Les opérateurs arithmétiques (+-*/^).
- Les opérateur relationnels (== => <> = <= !) –L'expression renvoie **1** si la relation est vraie ; sinon, elle retourne **0.**
- Parenthèses
- Les fonctions mathématiques et **tanh.**

Unit Delay : Le bloc **Unit Delay** retarde l'entrée durant une période spécifiez. Ce bloc est équivalent au z^{-1} opérateur à temps discret. Chaque signal peut être un scalaire ou un vecteur. Si l'entrée est un vecteur, le bloc retarde tous les éléments du vecteur durant la même période.

Bibliographie

Bibliographie

[1] S.H. Strogatz, « Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering », Addison-Wesley, 1994.

[2] E. Goncalves, « introduction aux systèmes dynamiques et chaos », Engineering school, Institut Polytechnique de Grenoble, France, Avril 2004.

[3] A.J. Michaels, « Digital Chaotic Communications », Thèse de Doctorat, Georgia Institute of Technology, 2009.

[4] I. Talbi, « Systèmes dynamiques non linéaires et phénomène du chaos (application à la cryptographie) », Mémoire de Magister, Université Mentouri de Constantine, 2010.

[5] O. Megherbi, « Etude et réalisation d'un système sécurisé à base de systèmes chaotiques », Mémoire de Magister, Université Mouloud Mammeri de Tizi Ouzou, 2013.

[6] F.Anstett, « Les systèmes dynamiques chaotiques pour le chiffrement : synthèse et cryptanalyse », Thèse de Doctorat, Université Henri Poincaré, Nancy, France, 2006.

[7] Kihal Ahmed Ridha, « Systèmes chaotiques pour la transmission sécurisée de données », Mémoire de Magister, Université Mohamed Khider, Biskra, 2013.

[8] E. Polizzi, « Chaos: From Classical to Quantum », Graduate research report, UPS, University of Toulouse, Mai 1996.

[9] T. Hamaizia, « systèmes dynamiques et chaos : "Application à l'optimisation à l'aide d'algorithme chaotique" », Thèse de Doctorat, Université Mentouri de Constantine, 2013.

[10] I. Ameur, « Contrôle, chaotification et hyperchaotification des systèmes dynamiques », Mémoire de Magister, Université Mentouri de Constantine, 2007.

[11] R.C Hilborn, « Chaos and nonlinear dynamics: an introduction for scientists and engineers », Oxford University Press, second edition, 2000.

[12] K.M. Cuomo, A.V. Oppenheim and S.H. Strogatz, « Synchronization of Lorenz based chaotic circuits with applications to communications », Physics, review and letters, Vol. 71, pp. 65-68, October 1993.

[13] D.E. Comer, « Internetworking with TCP/IP: principles, protocols, architecture Vol 1 », Prentice-Hall, New Jersey, USA, 1992.

Bibliographie

[14] Z. EL HADJ, « Etude de quelques types de systèmes chaotiques: généralisation d'un modèle issu du modèle de CHEN », Thèse de Doctorat, Université Mentouri de Constantine, 2006.

[15] H. Hamiche, « Inversion à gauche des systèmes dynamiques Hybrides chaotiques. Application à la transmission sécurisée de données », Thèse de Doctorat, Université Mouloud Mammeri de Tizi Ouzou, 2011.

[16] G. Zheng, « Formes Normales d'Observabilité Paramétrées par les Sorties : Applications au Cryptage par Synchronisation de Systèmes Chaotiques », Thèse de Doctorat, Université de Cergy-Pontoise, France, 2006.

[17] Mihai Bogdan Luca, « Apports du chaos et des estimateurs d'états pour la transmission sécurisée de l'information », Thèse de Doctorat, Université de Bretagne occidentale, Brest, France, 2006.

[18] A. Zemouche, « Sur l'observation de l'état des systèmes dynamiques non linéaires », Thèse de Doctorat, Université Louis Pasteur-Strasbourg I, France, 2007.

[19] M.L'Hernault, A.Ouslimani, J.P.Barbot, « Conception et réalisation d'un observateur à modes glissants pour un oscillateur de Colpitts chaotique », Manifestation des Jeunes Chercheurs francophones dans les domaines des STIC, pp.232-237, Rennes, Novembre 2005.

[20] H. Dimassi, « Synchronisation des systèmes chaotiques par observateurs et applications à la transmission d'informations », Thèse de Doctorat, Université Paris Sud, France, 2012.

[21] E. Cherrier, « Estimation de l'état et des entrées inconnues pour une classe de systèmes non linéaires », Thèse de Doctorat, Institut National Polytechnique de Lorraine, Nancy, France, 2012.

[22] Eskimon et Olyte, « Arduino pour bien commencer en électronique et en programmation », site web : https://openclassrooms.com, 2012.

[23] Simon Landrault, Hippolyte Weisslinger, « Arduino : Premiers pas en informatique embarquée », site web : https://openclassrooms.com, 2014.

[24] De Labachelerie, « Cours de physique optique », École nationale supérieure de mécanique et des microtechniques, 2003.

Bibliographie

[25] M. Halimi, « Observation et détection de modes pour la synchronisation des systèmes chaotiques : une approche unifiée », Thèse de Doctorat, Université de Lorraine, Nancy, France, 2013.

[26] L.M. Pecora, T.L. Carroll, « Synchronization in chaotic systems », Physical Review Letters, pp. 821-825, February 19, 1990.

[27] H. Nijmeijer, M.Y.I. Mareels, « An Observerlooks at Synchronization », Fundamantal theory and application, Vol. 44, pp. 882-890, 1997.