

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Mention Électronique
Spécialité Traitement de l'Information et Systèmes Électroniques

présenté par

AZIRA HIBA

ET

KHETTAL MAHDIA

Analyse et implémentation du système Chaotique de Qi

Proposé par : Mr FERDJOUNI Abdelaziz

Année Universitaire 2015-2016

Remerciements

Nous tenons à remercier tout d'abord «ALLAH » tout puissant qui nous donné durant toutes ces années la santé, le courage et la fois en nous-mêmes pour en arriver là.

Nous exprimons nos sincères remerciements à notre promoteur

Mr. FERDJOUNI ABD EL AZIZ pour avoir accepté d'encadrer ce travail et pour son aide et son précieux encouragement pour la mise en œuvre de ce projet.

Nous voudrions aussi remercier tous les membres du laboratoire LabSET qui nous ont apporté leur soutien tout au long de ce travail.

Notre infinie gratitude va à tous nos enseignants qui ont contribué à notre formation et à tous les enseignants du département d'électronique et en particulier les membres du jury.

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

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

ملخص

نظرية الفوضى تتعامل مع الأنظمة الديناميكية القطعية بدقة . أولاً، نبدأ مع وصف السمات الرئيسية للأنظمة الديناميكية الفوضوية تليها دراسة النظام الفوضوي Qi. من أجل مزامنة النظامين Qi وقد وضعت واحدة رئيسية والعبد و قد تم تحليل ثلاثة قوانين تحكم. وأجريت دراسة محاكاة مع Matlab لتحليل أدائها. وأخيراً، يتم إجراء تطبيق هذا النظام على بطاقة Virtex-5 FPGA لمشاهدة الإشارات في الوقت الحقيقي ، و مقارنتها مع نتائج المحاكاة.

مفتاح الكلمات

التزامن. النظام فوضى Qi, بطاقة FPGA

Résumé

La théorie du chaos traite des systèmes dynamiques déterministes. D'abord, on commence par donner une description des principales caractéristiques des systèmes dynamiques chaotique suivie d'une étude du système chaotique de Qi. Dans le but de synchroniser deux systèmes de Qi l'un maître et l'autre esclave on a développé trois lois de commande. Une étude en simulation sous Matlab Simulink a été effectuée pour analyser leurs performances. Enfin, une implémentation digitale de ce système est réalisée sur la carte FPGA Virtex-5 pour voir les signaux en temps réel et pour pouvoir les comparer avec les résultats de simulation.

Mots clés : Systèmes dynamiques, système chaotique de Qi, synchronisation, carte FPGA.

Abstract

Chaos theory treats dynamic systems determinists. We first give a description of the main features of chaotic dynamical systems followed by a study of the chaotic system of Qi. In order to synchronize the two systems Qi one master and one slave was developed three laws. Finally, a digital implementation of this system is performed on the Virtex-5 FPGA to see real-time signals and to compare them with simulation results.

Keywords: Chaos, chaotic system Qi, synchronization, FPGA.

Listes des acronymes et abréviations

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

R^n : Ensemble des nombres réels entiers.

R^+ : Ensemble des nombres réels positifs.

x_0 : L'état initial de la variable x.

x_k : L'état de x au temps $t=kT$.

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

Z^+ : Ensemble des nombres rationnels positifs.

λ_i : Exposant de Lyapunov d'ordre i.

\bar{x} : Point fixe.

FPGA : Field Programmable Gate Array.

HDL : Hardware Description Language.

ISE : Integrated Software Environment.

VHDL : Very High Density Logic.

ASIC : circuit logique programmable lors de sa fabrication.

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

SRAM : Static Random Access Memory.

LCD : Liquid-Crystal display.

RTL : Register Transfer Logic

Table des matières

Introduction générale.....	1
----------------------------	---

Chapitre 1 : GENERALITES SUR LES SYSTEMES CHAOTIQUES

1.1	Introduction.....	3
1.2	Les systèmes dynamiques.....	3
1.2.1	Systèmes dynamiques continus.....	4
1.2.2	Systèmes dynamiques discrets.....	4
1.2.3	Systèmes autonomes et non autonomes.....	5
1.3	Les systèmes dynamiques chaotiques.....	5
1.3.1	Définition du chaos.....	5
1.3.2	Caractéristique du chaos.....	6
1.4	L'espace de phase.....	6
1.4.1	Notion d'attracteur.....	7
1.5	Exposants de Lyapunov.....	8
1.6	Points fixes	9
1.7	Section de Poincaré	9
1.8	Diagramme de Bifurcation	10
1.9	Conclusion.....	11

Chapitre 2 : ETUDE DU SYSTEME HYPER CHAOTIQUE DE QI

2.1	Introduction.....	12
2.2	Description du système.....	12
2.3	Analyse du système hyper chaotique de Qi.....	13
2.3.1	Etude des points fixes.....	13
2.3.2	Evolution du système de Qi en fonction du temps.....	14
2.3.3	Plan de phase.....	17
2.3.4	Attracteur étrange.....	20

2.3.5	Exposants de Lyapunov.....	22
2.3.6	Section de Poincaré.....	24
2.3.7	Diagramme de bifurcation.....	24
2.4	Conclusion.....	26
Chapitre 3 : SYNCHRONISATION DU SYSTEME HYPER CHAOTIQUE DE QI		
3.1	Introduction.....	27
3.2	La synchronisation.....	27
3.2.1	Synchronisation unidirectionnelle.....	27
3.2.3	Synchronisation bidirectionnelle.....	28
3.2.3	Nécessité de la synchronisation.....	28
3.3	Méthodes de synchronisation.....	29
3.3.1	Synchronisation par la méthode du contrôle actif.....	30
3.3.2	Synchronisation de deux systèmes hyper chaotique en utilisant la fonction de lyapunov.....	35
3.3.2.1	Synchronisation par la méthode du contrôle actif.....	35
3.3.2.2	Synchronisation par la méthode du contrôle adaptatif.....	40
3.4	Conclusion.....	46
Chapitre 4 : IMPLEMENTATION DU SYSTEME QI SUR FPGA		
4.1	Introduction.....	47
4.2	Description des composants FPGA.....	47
4.3	Processus d'implémentation.....	48
4.4	Présentation des différents outils de travail.....	49
4.4.1	Présentation du logiciel ISE.....	49
4.4.2	Plate-forme de développement Virtex-5.....	50
4.5	Réalisation expérimentale de l'implémentation.....	52
4.5.1	Programmation en VHDL.....	52
4.5.2	visualisation des signaux.....	55

4.6	Conclusion.....	60
	Conclusion générale.....	61

Liste des figures

Figure 1.1. Attracteur de Rossler.

Figure 1.2. Représentation de la section de Poincaré pour le système de Rossler.

Figure 1.3. Diagramme de bifurcation du système de Rossler.

Figure 2.1. Représentation du système de Qi en Matlab Simulink.

Figure 2.2. L'état x en fonction du temps t .

Figure 2.3. L'état y en fonction du temps t .

Figure 2.4. L'état z en fonction du temps t .

Figure 2.5. L'état w en fonction du temps t .

Figure 2.6. Les états x, y, z et w en fonction du temps t .

Figure 2.7. Plan de phase y en fonction de x du système de Qi.

Figure 2.8. Plan de phase z en fonction de x du système de Qi.

Figure 2.9. Plan de phase w en fonction de x du système de Qi.

Figure 2.10. Plan de phase z en fonction de y du système de Qi.

Figure 2.11. Plan de phase w en fonction de y du système de Qi.

Figure 2.12. Plan de phase w en fonction de z du système de Qi.

Figure 2.13. Attracteur étrange de Qi en fonction de x, y et z .

Figure 2.14. Attracteur étrange de Qi en fonction de x, y et w .

Figure 2.15. Attracteur étrange de Qi en fonction de x, z et w .

Figure 2.16. Attracteur étrange de Qi en fonction de y, z et w .

Figure 2.17. L'interface de l'outil MATDS.

Figure 2.18. Création du système de Q_i en MATDS.

Figure 2.19. La dynamique des exposants de Lyapunov du système de Q_i .

Figure 2.20. La section de Poincaré sur l'attracteur de Q_i .

Figure 2.21. Diagramme de bifurcation de y en fonction de b .

Figure 2.22. Les états x, y, z et w ont la même période lorsque $b=5$.

Figure 2.23. Plan de phase y en fonction de w lorsque $b=5$.

Figure 3.1. Le couplage unidirectionnel.

Figure 3.2. Le couplage bidirectionnel.

Figure 3.3. Évolution de deux systèmes chaotiques issus des conditions initiales très proches.

Figure 3.4. Plan de phase y_1 en fonction de x_1 .

Figure 3.5. Schéma de synchronisation sous Matlab.

Figure 3.6. Synchronisation des signaux x_1, y_1 .

Figure 3.7. Synchronisation des signaux x_2, y_2 .

Figure 3.8. Synchronisation des signaux x_3, y_3 .

Figure 3.9. Synchronisation des signaux x_4, y_4 .

Figure 3.10. Les erreurs de synchronisation e_1, e_2, e_3 et e_4 .

Figure 3.11. Schéma de synchronisation sous Matlab.

Figure 3.12. Synchronisation des signaux x_1, y_1 .

Figure 3.13. Synchronisation des signaux x_2, y_2 .

Figure 3.14. Synchronisation des signaux x_3, y_3 .

Figure 3.15. Synchronisation des signaux x_4, y_4 .

Figure 3.16. Les erreurs de synchronisation e_1, e_2, e_3 et e_4 .

Figure 3.17. Schéma de synchronisation sous Matlab.

Figure 3.18. Synchronisation des signaux x_1, y_1 .

Figure 3.19. Synchronisation des signaux x_2, y_2 .

Figure 3.20. Synchronisation des signaux x_3, y_3 .

Figure 3.21. Synchronisation des signaux x_4, y_4 .

Figure 3.22. Les erreurs de synchronisation e_1, e_2, e_3 et e_4 .

Figure 3.23. L'estimation de paramètre a.

Figure 3.24. L'estimation de paramètre b.

Figure 3.25. L'estimation de paramètre c.

Figure 3.26. L'estimation de paramètre g.

Figure 3.27. L'estimation de paramètre f.

Figure 3.28. L'estimation de paramètre h.

Figure 4.1. Programmation d'un FPGA.

Figure 4.2. L'interface Project Navigator de l'ISE 14.2.

Figure 4.3. Plateforme de développement Virtex-5(vu de dessus).

Figure 4.4. Plateforme de développement Virtex-5(vu de dessous).

Figure 4.5. Interface IP (CORP Generator & Architecture Wizard).

Figure 4.6. Exemple d'un bloc d'addition.

Figure 4.7. visualisation des signaux sous ModelSim.

Figure 4.8. La fenêtre pour la synthèse.

Figure 4.9. Interface permettant la programmation du FPGA.

Figure 4.10. Implantation du système hyper chaotique sur la carte FPGA.

Figure 4.11. L'état de x en fonction du temps.

Figure 4.12. L'état de y en fonction du temps.

Figure 4.13. L'état de z en fonction du temps.

Figure 4.14. L'état de w en fonction du temps.

Figure 4.15. Plan de phase y en fonction de x .

Figure 4.16. Plan de phase z en fonction de x .

Figure 4.17. Plan de phase w en fonction de x .

Figure 4.18. Plan de phase z en fonction de y .

Figure 4.19. Plan de phase w en fonction de y .

Figure 4.20. Plan de phase w en fonction de z .

Figure 4.21. Ressources consommées par l'implémentation.

Bibliographie

- [1] L.M. Pacora, T.L.Caroll. "synchronizing nonautonomous chaotic circuits". IEEE transactions on circuits and systems 2: analog and digital signal processing, oct 1993:646-650
- [2] O.Megherbi, ' Etude et réalisation d'un système sécurisé à base de systèmes chaotiques ', Thèse de magister, Université Mouloud Mammeri Tizi-Ouzou, Algérie, 2013.
- [3] Abdelkrim Boukabou : 'Méthodes de contrôle des systèmes chaotique d'ordre élevé et leur application pour la synchronisation : Contribution à l'élaboration de nouvelles approches', Thèse de doctorat d'Université de Constantine, Algérie,2006.
- [4]J.Oden, 'Le chaos dans les systèmes dynamiques', Rapport Université Paris XI, France,2007.
- [5] F. Alain, 'Contribution à la prédiction et au contrôle des comportements apériodiques dans le convertisseurs électromécaniques. Application de la théorie du chaos', Université de Reims France, 2005.
- [6] Tayeb Hamaizia 'système dynamique chaotique : Application à l'optimisation à l'aide d'algorithmes chaotique '. Université de Constantine-1-, Avril 2013.
- [7] G.Y. Qi. M.A. van Wyk, B.J. van Wyk. et al, On a new hyperchaotic system. Physics letters A 372 (2008) 124-136.
- [8] Hadj-youcef Brahim & Laazab Kheira « analyse des systèmes dynamiques chaotiques », Mémoire d'ingénieur d'état en électronique université de Blida 2011.
- [9] Yaoyao Lan, Qingguo Li « chaos synchronization of a new hyperchaotic system »,Applied Mathematics and Computation 2010.
- [10] Pr. Mohamed Djemai : analyse, commande & observation des systemes non lineaires.blida 2011.

[11] V.A.Pedroni « Circuit design with VHDL » , MIT Press, 2004.

[12] Can EROGLU: implementation of synchronized chaotic systems by field programmable gate array, Izmir institute of technology.M.SC in electrical and electronics engineering 2007.

Dédicace

Je dédie ce modeste travail à mes très chers et magnifiques parents « IBRAHIM » et « NACIRA » pour l'éducation qu'ils ont su me donner et qui m'a permis avec l'aide de DIEU d'arriver là où je suis « Maman, Papa je vous aime beaucoup » et mes deux frères « Mohamed Lamine » et « Abd el Madjide » que dieu vous protège tous.

Je dédie ce mémoire à une personne qui compte beaucoup pour moi, à toi « MOHAMED », qui m'a vraiment soutenu dans les moments difficiles et qui a été tous le temps présent pour moi.

A toute la famille qui m'ont soutenu durant les moments difficiles surtout ma grande mère « Zehore » et mon oncle « Aziz ». A mes très chères tantes paternelles et maternelles : Ibtissam, Houria, Habiba, Nour el Din, Rabah... sans oublier Tata Nabila et Tata Naima. A tous mes cousines et cousines : Nariman, Zaki, Khadidja, Mohamed et Walid... merci du fond du cœur.

A mes très chères amies : Nassima, Fatima, Samira, Ilham, Yasmine, Nessrin, Zahra, Nassima26.

A mon binôme « Mehdi » avec qui j'ai passée des bons moments.

AZIRA HIBA

Introduction générale

Le monde qui nous entoure paraît souvent imprévisible. Une partie de cette complexité du monde est maintenant devenue scientifiquement compréhensible grâce à la théorie du chaos déterministe.

Le chaos, avant 1963 ce terme signifiait que le désordre. Mais avec l'arrivée de Edward Lorenz, l'homme qui a découvert le chaos et ses principes caractéristiques. Après le météorologue Lorenz beaucoup de scientifiques ont développé la théorie du chaos. Ces scientifiques ont inventé leurs propres systèmes chaotiques. Il a également développé son propre système [7] .

Récemment, les chercheurs se sont intéressés à la synchronisation des systèmes chaotiques. Et comme ces derniers sont caractérisés par la sensibilité aux conditions initiales, la synchronisation entre deux systèmes chaotiques paraissait impossible. Mais la découverte de Pecora et Carrol (en 1990) montre le contraire [1]. Ils ont montré que deux systèmes chaotiques identiques avec des conditions initiales différentes peuvent éventuellement se synchroniser s'ils sont couplés d'une manière convenable. Il faut noter que la synchronisation joue un rôle important dans les schémas de communication.

La génération de signaux chaotique peut être réalisée grâce à des circuits électroniques analogiques. Elle peut aussi être envisagée grâce à des circuits digitaux. Dans cette situation, il sera nécessaire de résoudre le système chaotique à l'aide d'une méthode numérique, et ensuite implémenter le programme sur une carte FPGA.

Cette thèse est organisée de la façon suivante :

-Le premier chapitre présente un rappel sur les systèmes dynamiques en général et chaotiques en particulier. Il énoncera quelques concepts sur la théorie du chaos.

-dans le deuxième chapitre, nous avons étudié le comportement du system hyper chaotique de Qi et en vue toutes ses caractéristiques à l'aide des outils de simulation.

-Dans le troisième chapitre, des lois de commande sont calculées pour obtenir la synchronisation. Une première loi de commande, basée sur la linéarisation de l'erreur de synchronisation est développés. Ensuite, une seconde loi est calculée à l'aide de la théorie de stabilité de lyapunov, pour tenir compte de la différence entre les paramètres de l'émetteur et ceux du récepteur. Une troisième loi de commande est développée, elle permet l'identification des paramètres de l'émetteur grâce à une approche de commande adaptative.

-Le quatrième chapitre présente un aperçu sur la technologie FPGA et avec laquelle on implémente notre système en langage VHDL.

Enfin, la conclusion reprend les principales points abordés dans ce travail et montre l'importance de l'étude des systèmes chaotiques.

Chapitre 1 Généralité sur les systèmes chaotiques

1.1 Introduction

Le but de ce chapitre est de présenter conditions nécessaires pour qu'un système ait un comportement chaotique. Nous partirons de quelques définitions préliminaires permettant de cerner les caractéristiques essentielles des systèmes dynamiques. L'étude rigoureuse de ce type de système étant en général délicate, nous nous limiterons à une étude qualitative purement graphique. Pour cela, nous introduisons le diagramme de bifurcation, l'exposant de Lyapunov et la section de Poincaré.

1.2 Les systèmes dynamiques

Un système dynamique est une structure qui évolue au cours du temps de façon à la fois :

- Causale, où son avenir ne dépend que de phénomènes du passé ou du présent
- Déterministe, c'est-à-dire qu'à partir d'une condition initiale donnée à l'instant « présent » va correspondre à chaque instant ultérieur un et un seul état « futur » possible.
- L'évolution déterministe du système dynamique peut alors se modéliser de deux façons distinctes :
 - Une évolution continue dans le temps, représentée par une équation différentielle ordinaire.
 - Une évolution discrète dans le temps, représentée par une équation aux différences.

L'étude théorique de ces modèles discrets est fondamentale, car elle permet de mettre en évidence des résultats importants, qui se généralisent souvent aux évolutions dynamiques continues.

1.2.1 Systèmes dynamiques continus

Un système dynamique continu est décrit par un système d'équations différentielles de la forme :

$$\dot{x}(t) = F(x(t), t) \quad (1.1)$$

Où $F : R^n \times R^+ \rightarrow R^n$ représente la dynamique du système.

Si on associe à cette dynamique un état initial : $x_0 = x(t_0)$, alors pour chaque couple (x_0, t_0) on peut identifier une solution unique $\phi(\cdot; x_0, t_0) : R^+ \rightarrow R^n$ telle que :

$$\begin{aligned} \phi_F(t_0; x_0, t_0) &= x_0 \\ \dot{\phi}_F(t; x_0, t_0) &= F(\phi_F(t; x_0, t_0), t) \end{aligned} \quad (1.2)$$

L'évolution des ensembles d'états successifs occupés par le système à chaque instant t représente la trajectoire du système.

1.2.2 Système dynamiques discrets

Un système dynamique discret est représenté par des équations aux différences de la forme :

$$X(k+1) = G(x(k), k) \quad (1.3)$$

Où $G : R^n \times Z^+ \rightarrow R^n$ désigne la dynamique du système en temps discret.

Si on associe à cette dynamique un état initial $x_0 = x(k_0)$, alors pour chaque couple (x_0, k_0) donné, on peut identifier une solution unique

$$\phi_G(\cdot; x_0, k_0) : Z^+ \rightarrow R^n \quad (1.4)$$

Telle que :

$$\phi_G(k_0; x_0, k_0) = x_0 \quad \text{et} \quad \dot{\phi}_G(k+1; x_0, k_0) = G(\phi_G(k; x_0, k_0), k) \quad (1.5)$$

1.2.3 Systèmes autonomes et non autonomes

Soit le système dynamique suivant :

$$\dot{x}(t) = \frac{dx(t)}{dt} = F(x, t) \quad (1.6)$$

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

En temps discret on définit également le système autonome comme une dynamique qui ne dépend pas de l'instant k :

$$x(k+1) = G(x(k)) \quad (1.7)$$

Remarque

Par un changement de variable approprié, on peut toujours transformer un système dynamique non autonome de dimension n en un système dynamique autonome équivalent de dimension $n + 1$.

1.3 Les systèmes dynamiques chaotiques

1.3.1 Définition du chaos

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

Les définitions et propriétés suivantes permettent de comprendre qualitativement les points marquants des systèmes chaotiques [2].

1.3.2 Caractéristique du chaos

Caractériser le chaos est un sujet vaste, et les tests les plus utiles pour ce type de comportement sont les suivants [3] :

- a) Les mouvements chaotiques sont plus compliqués que stationnaire, périodique ou quasi-périodique, et ils ont des formes très complexes, appelés attracteurs étranges.
- b) Sensibilité aux conditions initiales Certains phénomènes dynamiques non linéaires sont si sensibles aux conditions initiales que, même s'ils sont régis par des lois rigoureuses et parfaitement déterministes, les prédictions exactes sont impossibles. Il est clair que la moindre erreur ou imprécision sur la condition initiale interdit de décider à tout temps quelle sera la trajectoire effectivement suivie et de faire une prédiction sur l'évolution à long terme du système [4].
- c) La non-linéarité : Un système chaotique est un système dynamique non linéaire. Un système linéaire ne peut pas être chaotique.
- d) Le déterminisme : L'aspect déterministe signifie la capacité de prédire le futur d'un phénomène à partir d'un événement passé ou présent, Un système chaotique a donc des règles fondamentales déterministes plutôt que probabilistes.
- e) L'imprévisibilité : L'imprévisibilité est due à la sensibilité aux conditions initiales ; et en pratique on peut les connaître seulement à un degré fini de précision.
- f) L'irrégularité : Ordre caché comprenant un nombre infini de modèles périodiques instables (ou mouvements).

1.4 L'espace de phase

Dans un système dynamique de dimension n , l'espace x_1, x_2, \dots, x_n est appelé espace de phases ou espaces d'états. L'évolution par rapport au temps du système se traduit alors par un déplacement du point représentatif dans l'espace de phase, traçant ainsi une trajectoire de phase et x_1, x_2, \dots, x_n sont les états du système. Par

un point de l'espace de phase ne passe qu'une seule trajectoire. Par conséquent, deux trajectoires avec deux conditions initiales différentes ne coïncident jamais au cours du temps.

1.4.1 Notion d'attracteur

Un attracteur est un objet géométrique vers lequel tendent toutes les trajectoires de l'espace des phases, c'est-à-dire, une situation ou un ensemble de situations vers lesquelles évolue un système, quelles que soient ses conditions initiales.

Dans un espace des phases à deux dimensions, les attracteurs sont soit des points, soit des cycles limites.

1.4.1.1 Attracteur régulier

Pour tous les attracteurs réguliers, c'est-à-dire pour tous les systèmes non chaotiques des trajectoires qui partent de « point » proches l'un de l'autre dans l'espace de phase restent indéfiniment voisines. On sait donc prévoir l'évolution de ces systèmes, à partir d'une situation connue.

1.4.1.2 Attracteur chaotique

L'attracteur chaotique dit aussi étrange n'est pas une surface lisse, mais une surface repliée plusieurs fois sur elle-même. En effet, les trajectoires des points divergent (puisque, par définition deux points ne peuvent avoir la même évolution), mais comme l'attracteur a des dimensions finies, il doit se replier sur lui-même.

Un attracteur chaotique possède notamment la propriété suivante : la trajectoire ne repasse jamais par un même état (cette propriété est très remarquable). Ce qui signifie que cette trajectoire passe par une infinité d'états, la figure 1.1 représente l'attracteur de Rossler [9].

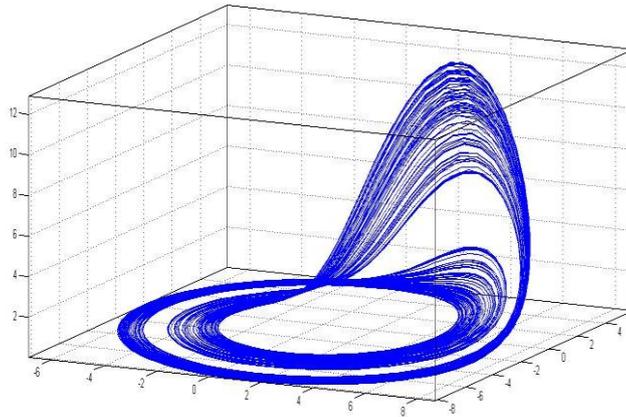


Figure 1.1. Attracteur de Rossler.

1.5 Exposants de Lyapunov

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

Un système de dimension n possède n exposants de Lyapunov qui mesurent le taux de divergence suivant un des axes de l'espace de phase. L'apparition du chaos exige l'existence d'un exposant positif selon au moins un axe [5], tout en rendant compte que la somme des exposants est négative (respectivement nulle) pour les systèmes dissipatifs (respectivement conservatifs).

Un exposant de Lyapunov positif (respectivement négatif) selon une direction indique, qu'une divergence entre deux trajectoires voisines augmente (respectivement diminue) exponentiellement avec le temps [5].

1.6 Points fixes

On appelle point fixe (ou point stationnaire ou point d'équilibre ou point critique) du système (1.1), le point x^* de l'espace de phase tel que :

$$F(x^*) = 0 \quad (1.8)$$

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

1.7 Section de Poincaré

La section de Poincaré est un outil mathématique simple permettant de transformer un système dynamique continu en un système dynamique discret [6].

Faire une section de Poincaré revient à couper la trajectoire dans l'espace des phases, afin d'étudier les intersections de cette trajectoire avec un plan. La figure (1.2) illustre en bleu l'attracteur de Rossler avec en rouge sa section de Poincaré correspondant au plan x - y .

L'ensemble des points d'intersection, situé sur la surface représente la section de Poincaré.

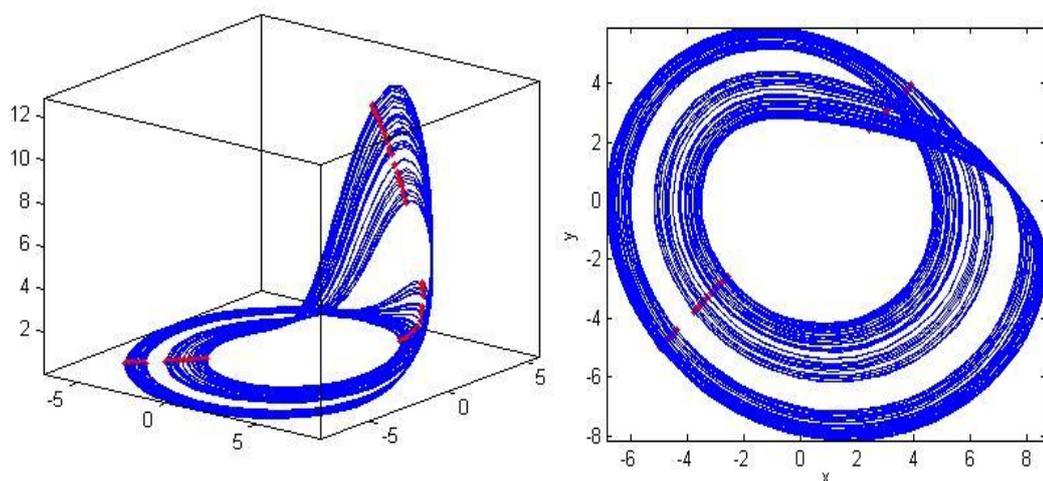


Figure 1.2. Représentation de la section de Poincaré pour le système de Rossler.

Puisque le résultat est une courbe discrète et non un cercle ou un point, on a un comportement chaotique.

1.8 Diagramme de Bifurcation

La bifurcation est un changement d'un système tel que la stabilité, le nombre de points d'équilibre ou la nature du régime permanent, lors d'une variation d'un paramètre. Les valeurs des paramètres au moment du changement sont appelées valeur de bifurcation.

Le diagramme de bifurcation est un outil qui permet de repérer les valeurs du paramètre qui induisent des bifurcations. C'est un diagramme qui porte les valeurs du paramètre en abscisse et des valeurs particulières d'une variable d'état. En fixant les valeurs des autres paramètres du système, on obtient le diagramme de bifurcation par rapport au paramètre considéré. La figure (1.3) illustre cette bifurcation pour le système de Rossler.

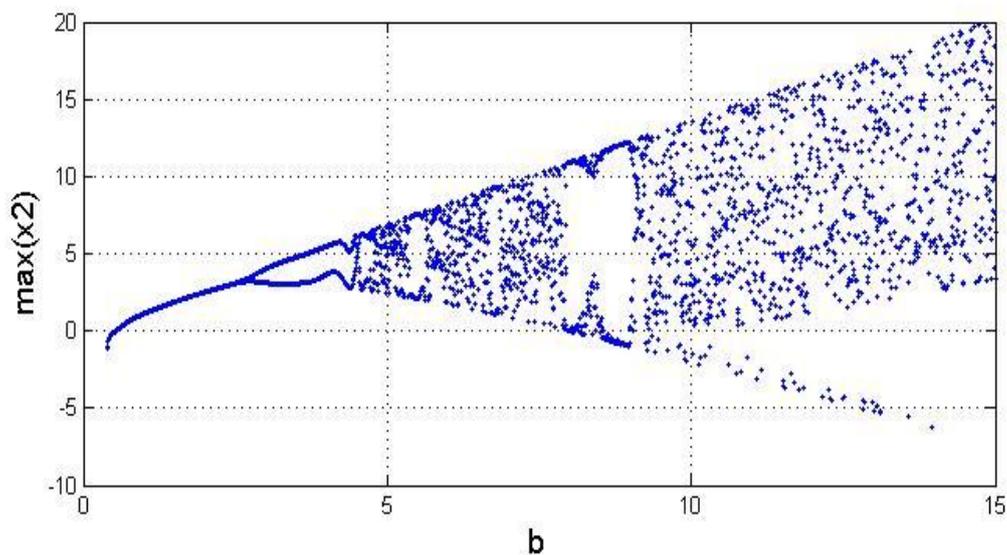


Figure 1.3. Diagramme de bifurcation du système de Rossler.

- Pour $1 < b < 3$, le système possède un point fixe stable (un cycle d'ordre 1).
- Pour $3 < b < 4$, le point fixe se déstabilise et un cycle d'ordre 2 apparaît (doublement de période).

- Pour $4 < b < 4.3$, le cycle d'ordre 2 se déstabilise et un cycle d'ordre 4 apparait et ainsi de suite.
- Pour $5 < b$, l'apparition du chaos.

1.9 Conclusion

Dans ce chapitre, nous avons présenté quelques notions et définitions de base des systèmes dynamiques et chaos, telles que les différents types de systèmes dynamiques. Nous avons aussi décrit les principales caractéristiques du chaos : section de Poincaré, la notion de bifurcation et les exposants de Lyapunov. Dans le chapitre suivant, nous analysons le fonctionnement du système hyper chaotique de Qi.

Chapitre 2 Etude du système hyper chaotique de Qi

2.1 Introduction

Les systèmes dynamiques chaotiques sont depuis longtemps connus dans le domaine des mathématiques, mais c'est seulement au cours de ses dernières années que les applications concrètes se sont multipliées. Ce présent chapitre a pour objectif d'étudier les différentes caractéristiques du système hyper chaotique de Qi (diagramme de bifurcation, point fixe, sections de Poincaré, etc...), à l'aide des outils de simulation tel que (MATLAB, Simulink et MATDS).

2.2 Description du système

Le système hyper chaotique de Qi est donné par [7] :

$$\begin{cases} \dot{x} = a(y - x) + yz. \\ \dot{y} = b(x + y) - xz. \\ \dot{z} = -cz - gw + xy. \\ \dot{w} = -fw + hz + xz. \end{cases} \quad (2.1)$$

Pour la simulation numérique on utilise les paramètres suivants :

$a = 50, b = 24, c = 13, f = 8, g = 33, h = 30$ Pour qu'on puisse trouver l'attracteur hyper chaotique de Qi.

2.3 Analyse du système hyper chaotique de Qi

2.3.1 Etude des points fixes

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

$$\frac{dx}{dt} = f(x) = 0 \quad (2.2)$$

C'est-à-dire :

$$\begin{cases} \dot{x} = a(y - x) + yz = 0 \\ \dot{y} = b(x + y) - xz = 0 \\ \dot{z} = -cz - gw + xy = 0 \\ \dot{w} = -fw + hz + xz = 0 \end{cases} \quad (2.3)$$

On obtient alors une solution : $x = y = z = w = 0$, cette solution n'est pas unique.

Pour étudier la stabilité de ce point fixe, on détermine les valeurs propres de la matrice jacobéenne.

Le calcul de la matrice jacobéenne au point fixe $D(0\ 0\ 0\ 0)$ donne :

$$J_0 = \begin{bmatrix} -a & a & 0 & 0 \\ b & b & 0 & 0 \\ 0 & 0 & -c & -g \\ 0 & 0 & h & -f \end{bmatrix} \quad (2.4)$$

Son équation caractéristique est donné par : $\det(\lambda I - J_0) = 0$, soit :

$$[(\lambda + a)(\lambda - b) - ba][(\lambda + c)(\lambda + f) + gh] = 0$$

Avec : I la matrice identité (4*4).

La résolution de l'équation caractéristique nous donne les racines suivantes :

$$\lambda_1 = -46.5150 + i19.1012, \lambda_2 = -46.5150 - i19.1012,$$

$$\lambda_3 = 23.2150 + i22.1781, \lambda_4 = 23.2150 - i22.1781$$

On a quatre valeurs propres complexes, deux à partie réelle positive et les deux autres ont à partie réelle négative, donc on a un point fixe instable.

2.3.2 Evolution du système de Qi en fonction du temps

Nous avons utilisé MATLAB Simulink pour visualiser les différents signaux issus du système de Qi, tels que : les états x_i en fonction du temps, les plans de phase et tracer les courbes de l'attracteur chaotique en 3D.

Pour cela, on a réalisé un circuit qui définit le système hyper chaotique de Qi à partir des équations (2.1).

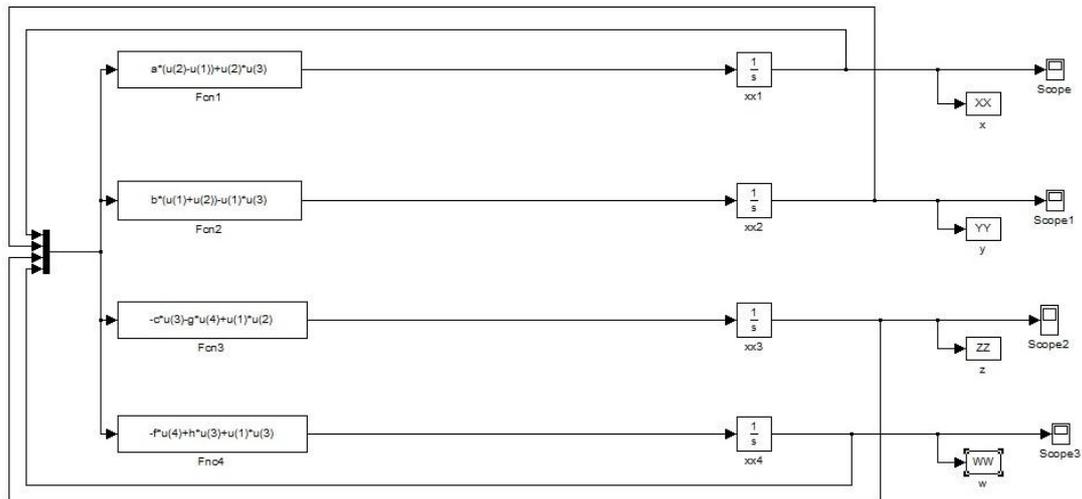


Figure 2.1. Représentation du système de Qi en Matlab Simulink.

Les figures de 2.2 à 2.6 représentent les courbes des états x , y , z et w en fonction du temps.

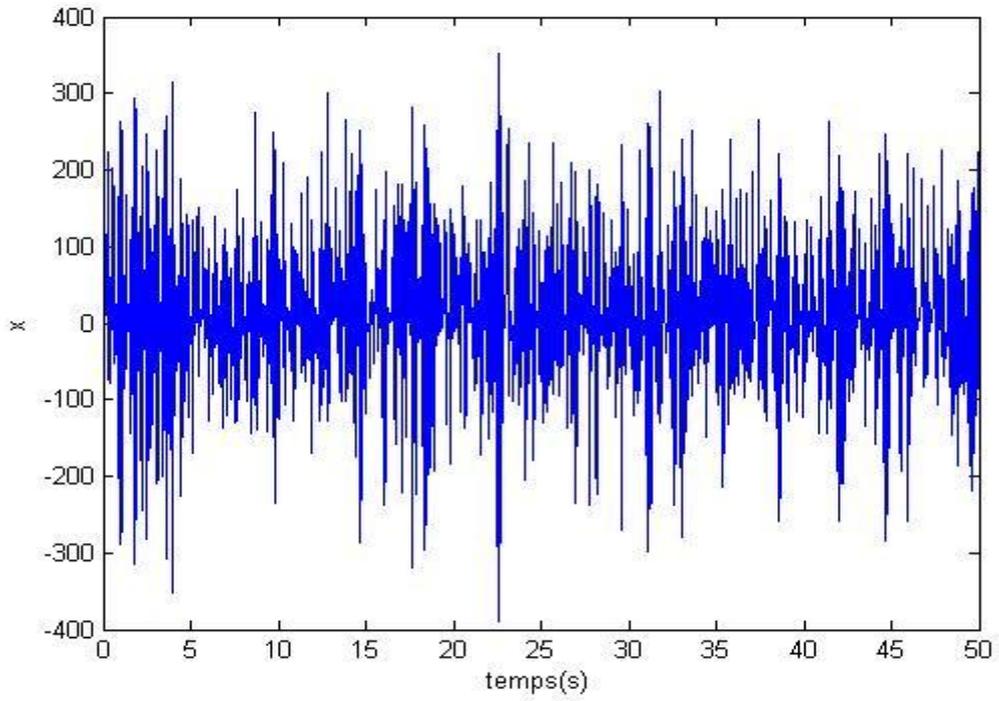


Figure 2.2. L'état x en fonction du temps t .

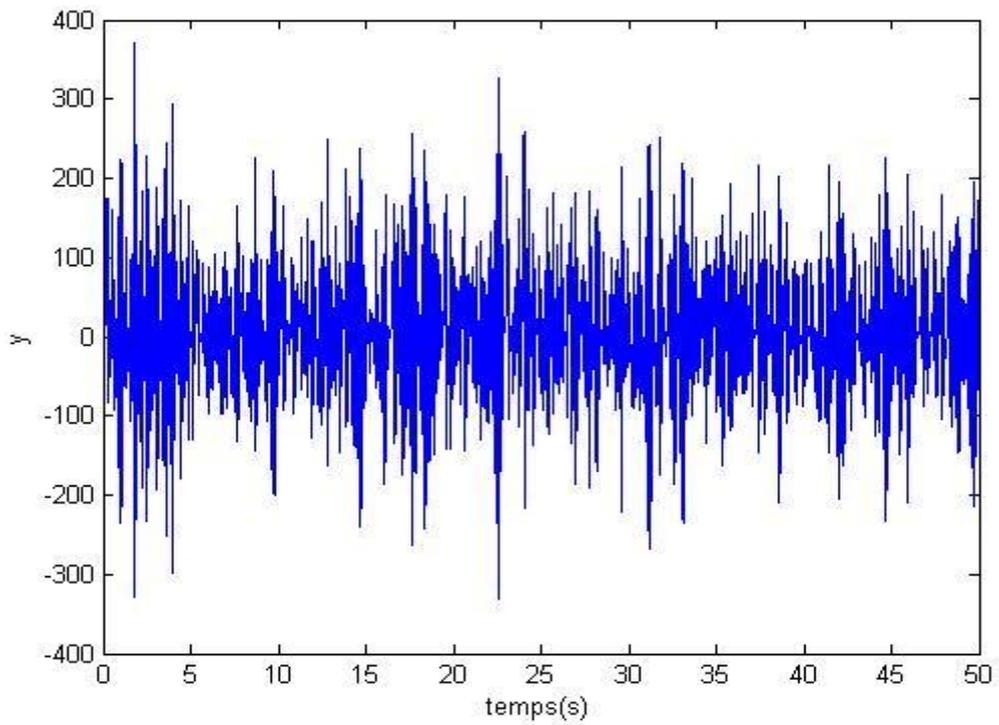


Figure 2.3. L'état y en fonction du temps t .

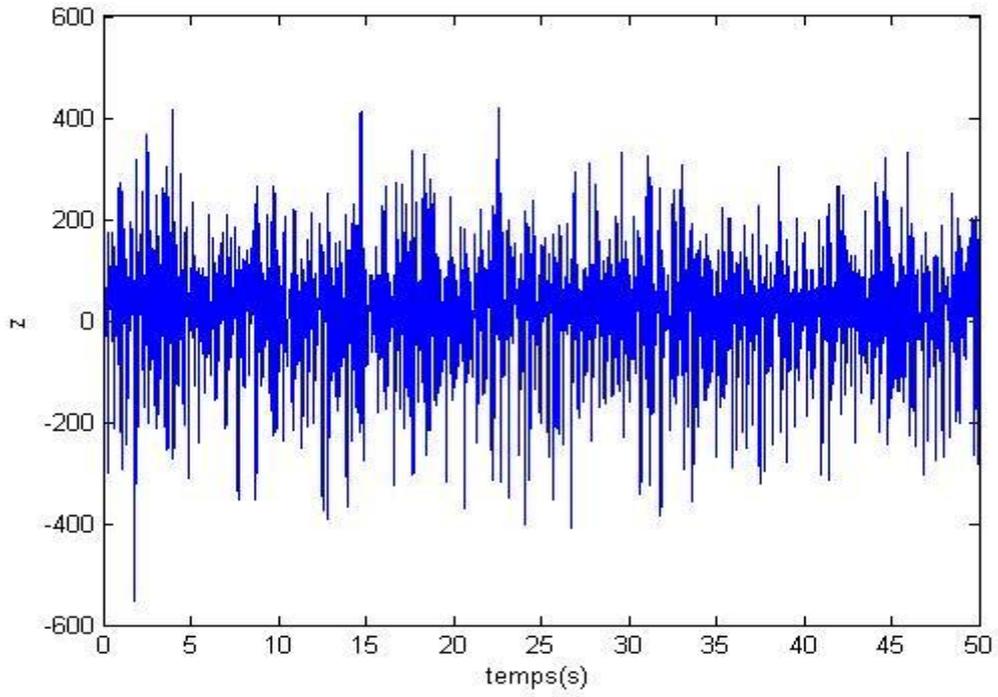


Figure 2.4. L'état z en fonction du temps t .

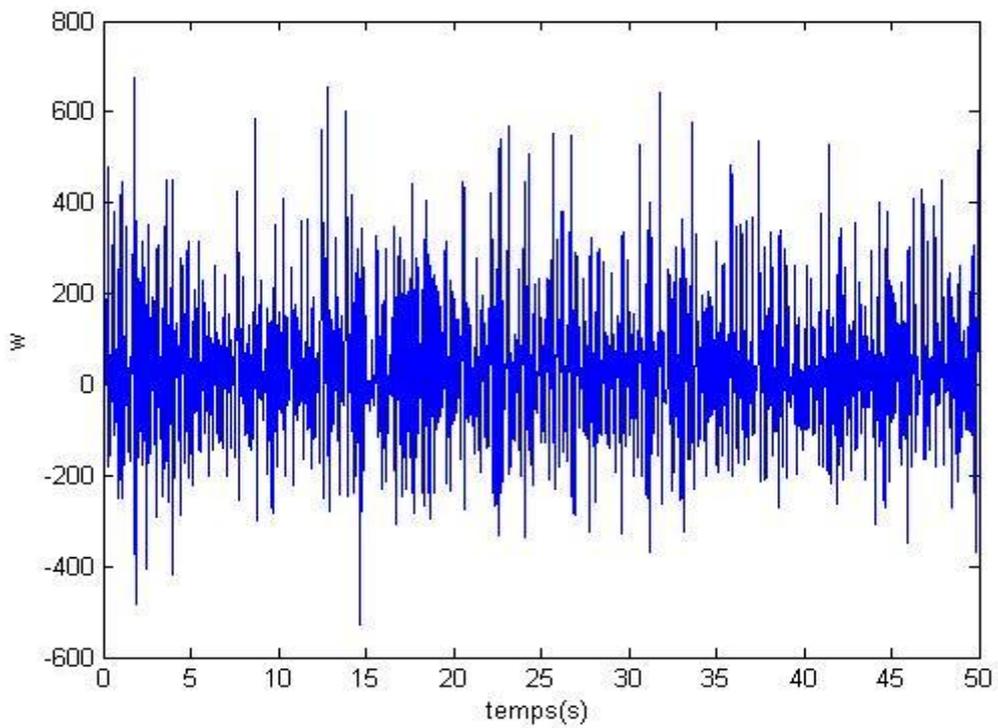


Figure 2.5. L'état w en fonction du temps t .

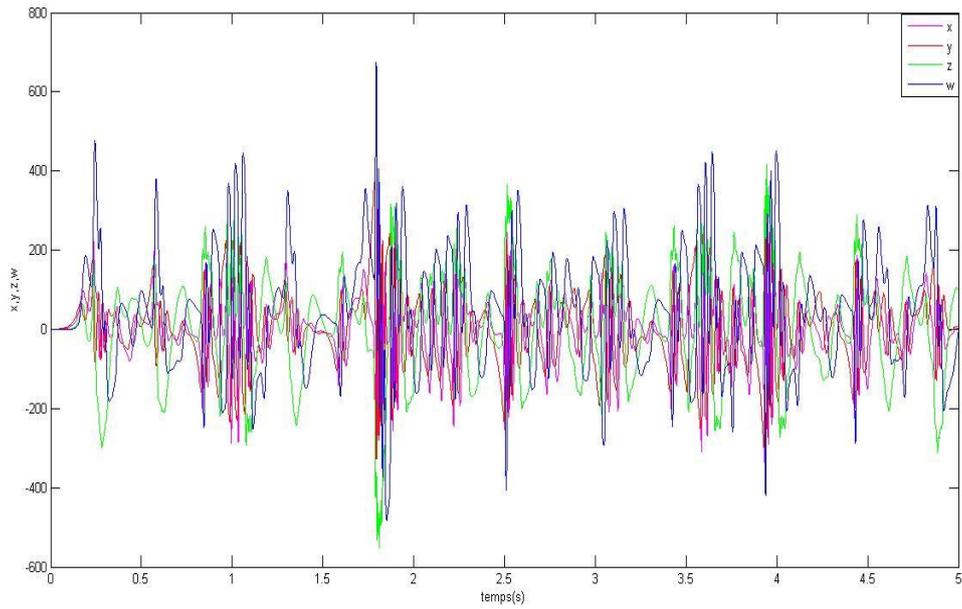


Figure 2.6. Les états x, y, z et w en fonction du temps t .

2.3.3 Plan de phase

Les figures 2.7 à 2.12 représentent les différents plans de phase du système hyperchaotique de Qi.

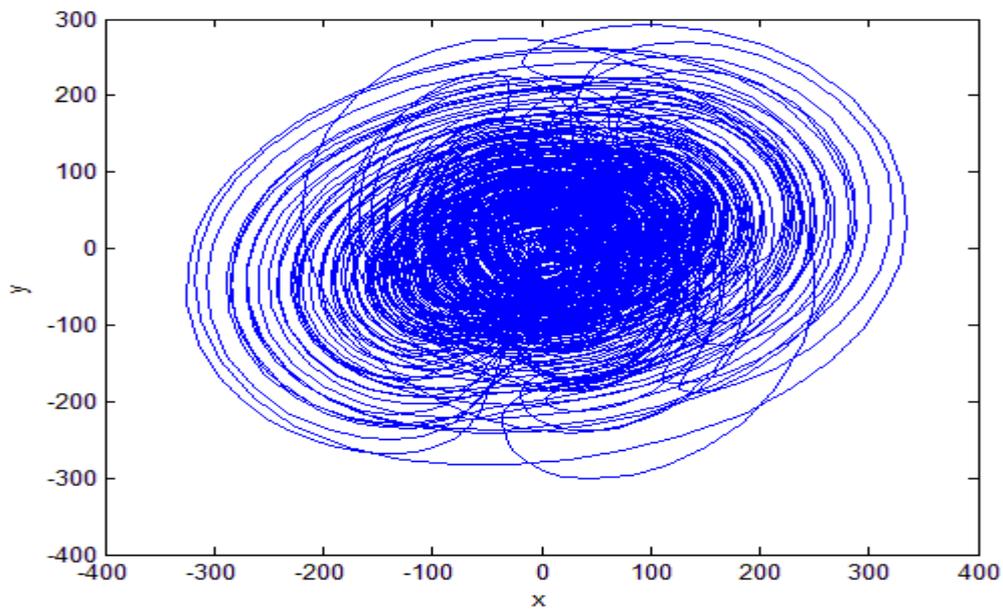


Figure 2.7. Plan de phase y en fonction de x .

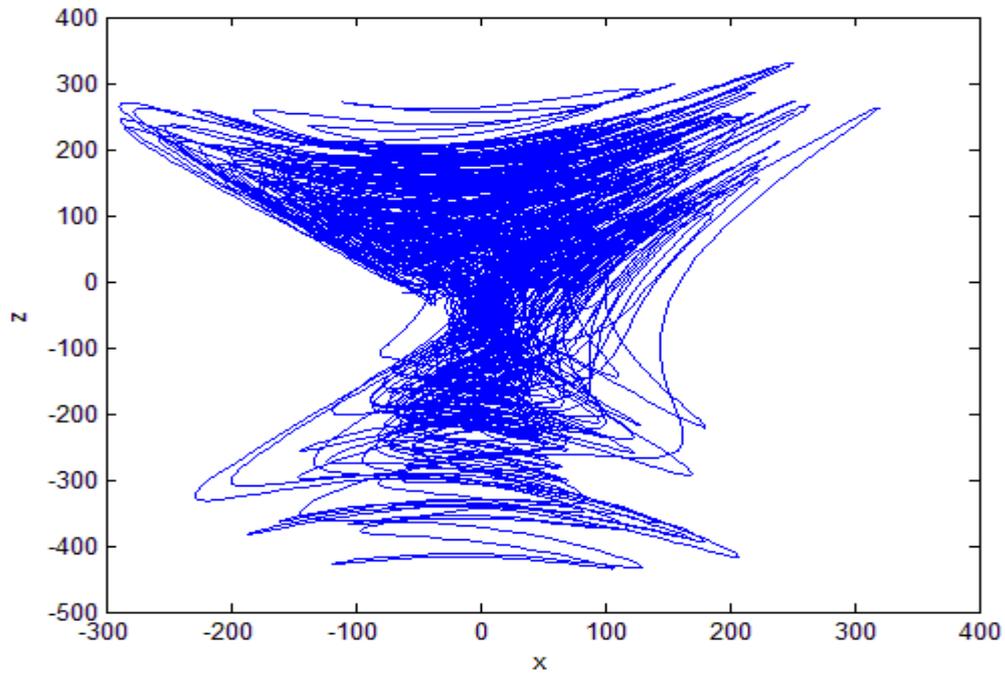


Figure 2.8. Plan de phase z en fonction de x .

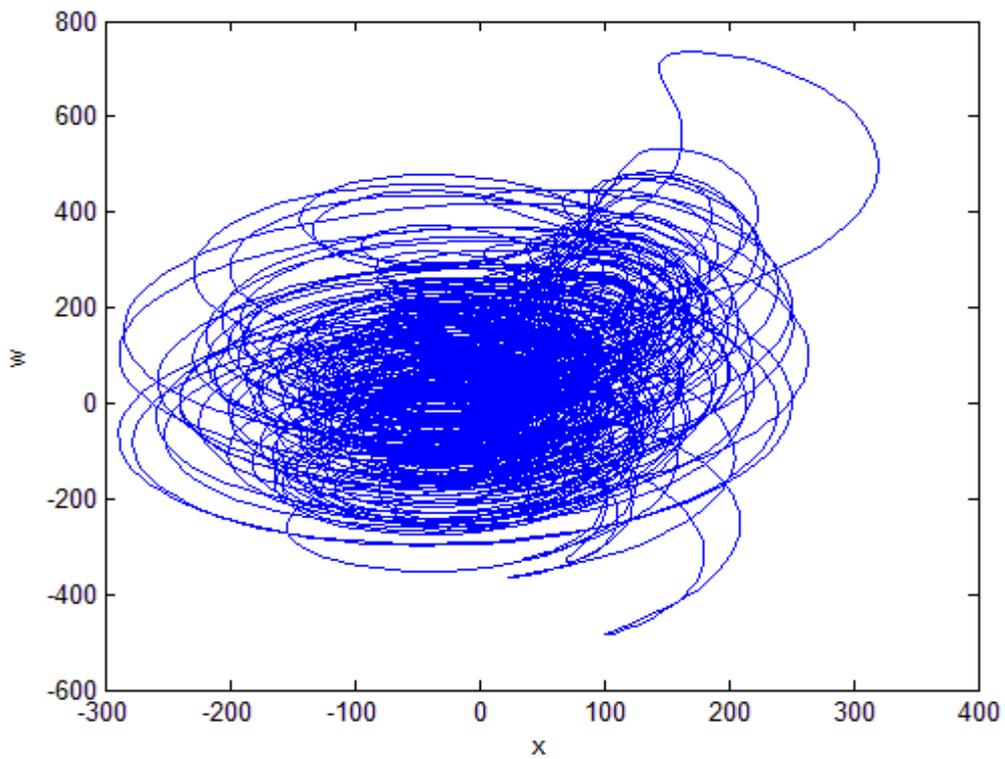


Figure 2.9. Plan de phase w en fonction de x .

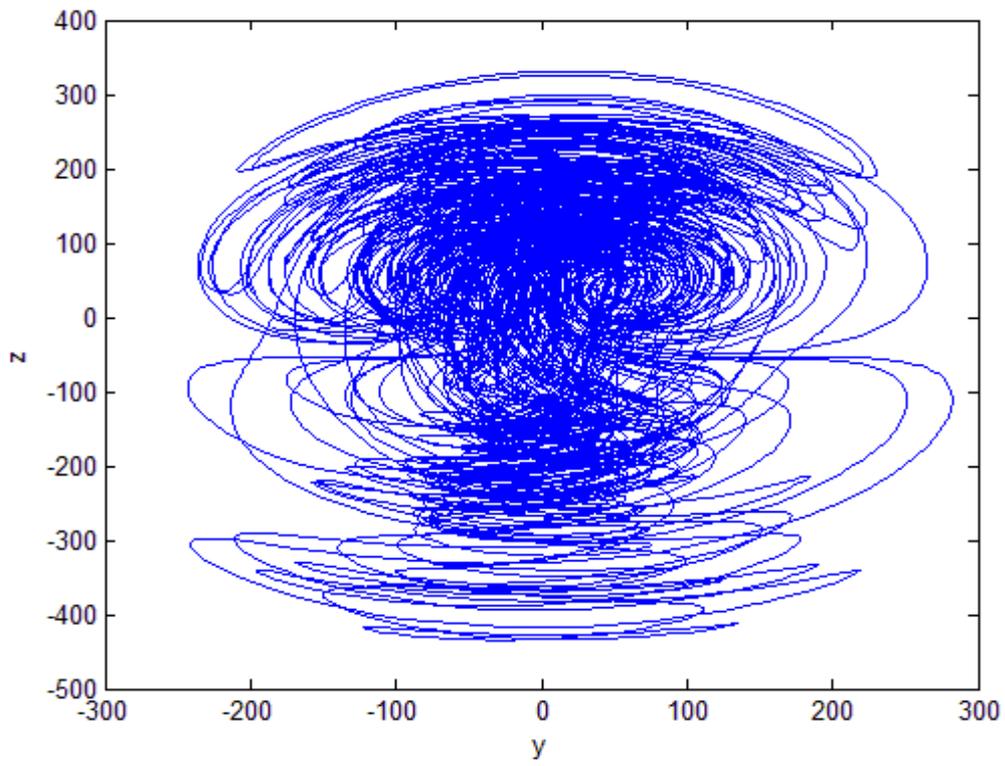


Figure 2.10. Plan de phase z en fonction de y .

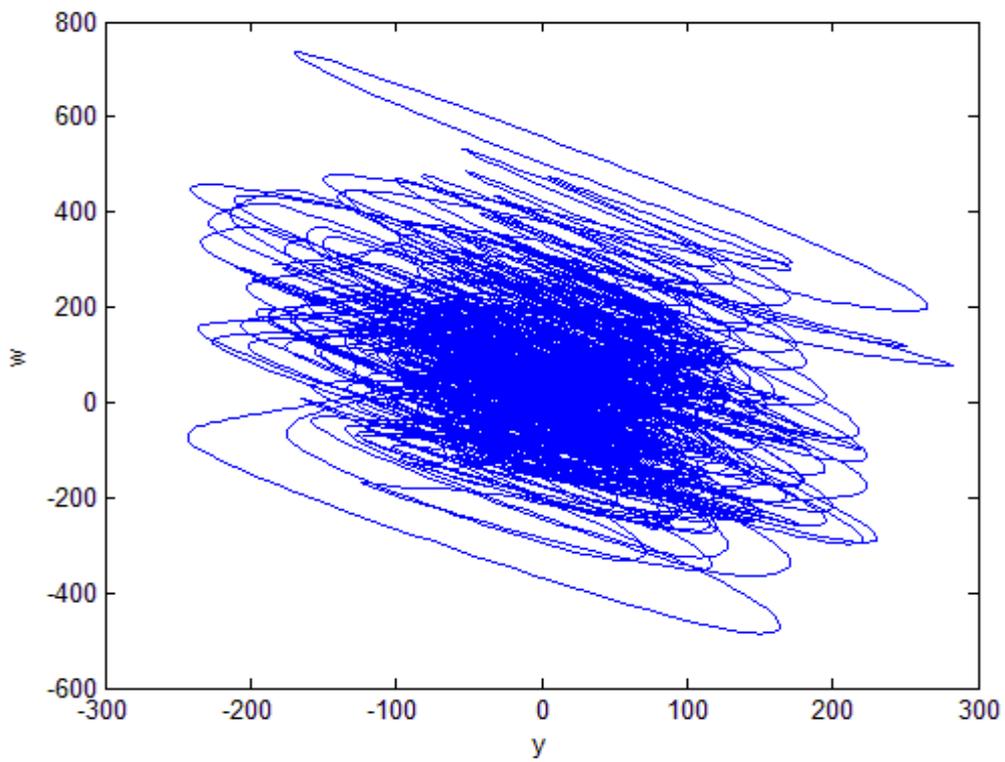


Figure 2.11. Plan de phase w en fonction de y .

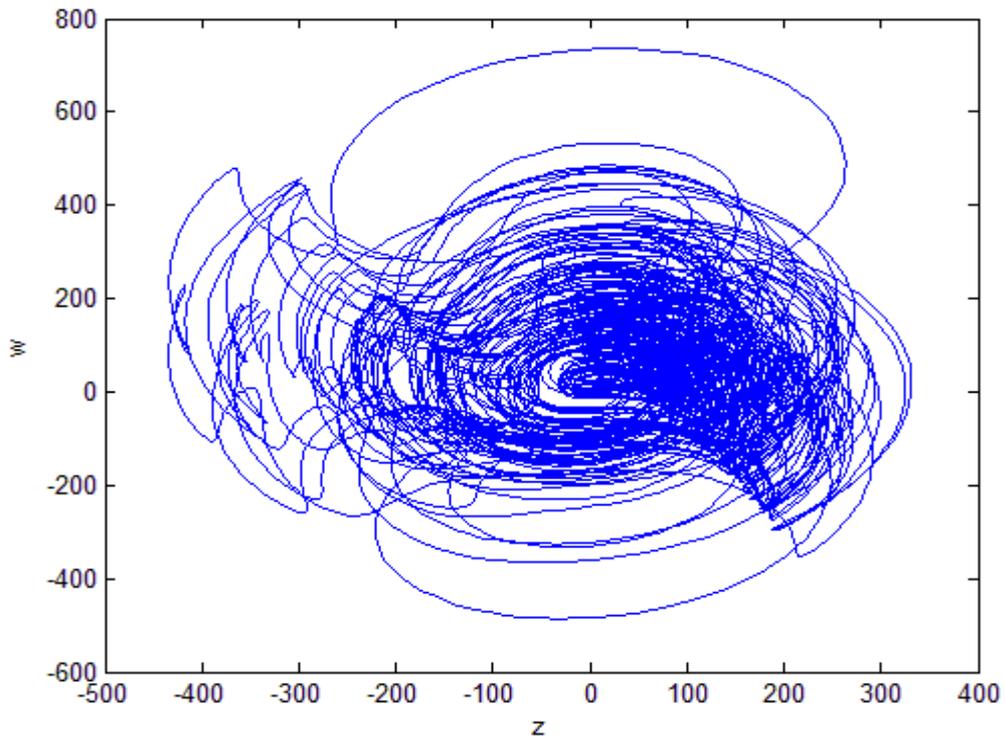


Figure 2.12. Plan de phase w en fonction de z .

2.3.4 Attracteur étrange

Les figures 2.13 à 2.16 représentent l'attracteur hyper chaotique de Q_i en 3D.

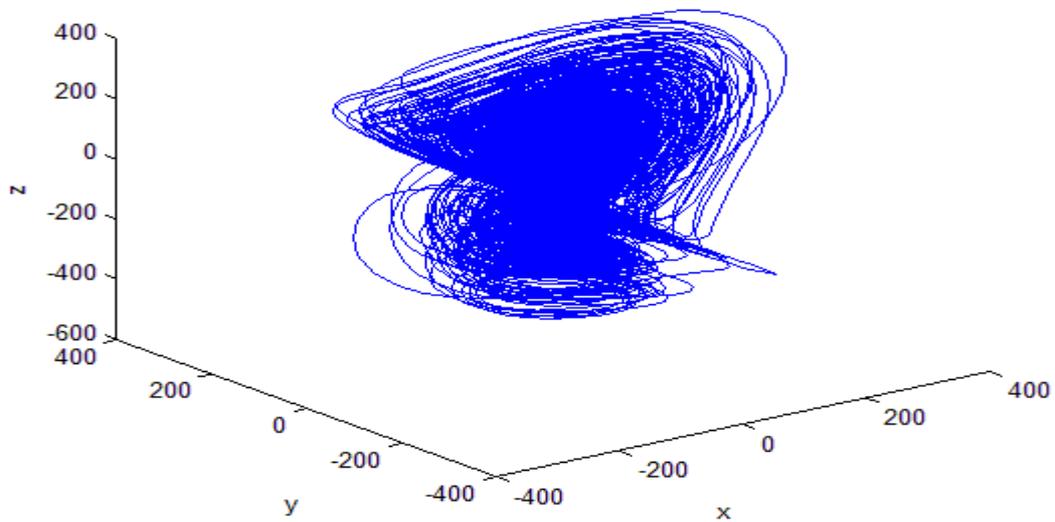


Figure 2.13. Attracteur étrange de Q_i en fonction de x, y et z .

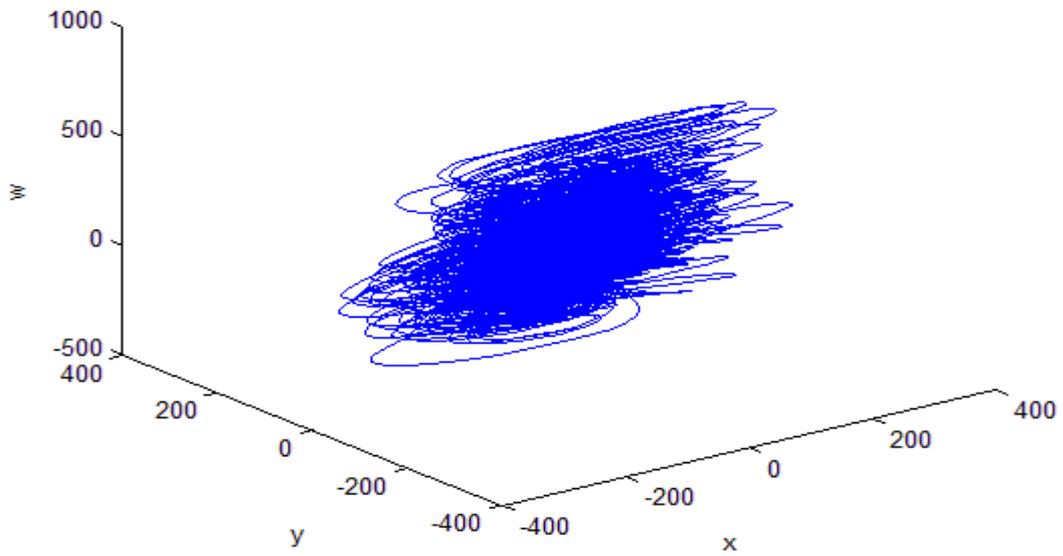


Figure 2.14. Attracteur étrange de Qi en fonction de x , y et w .

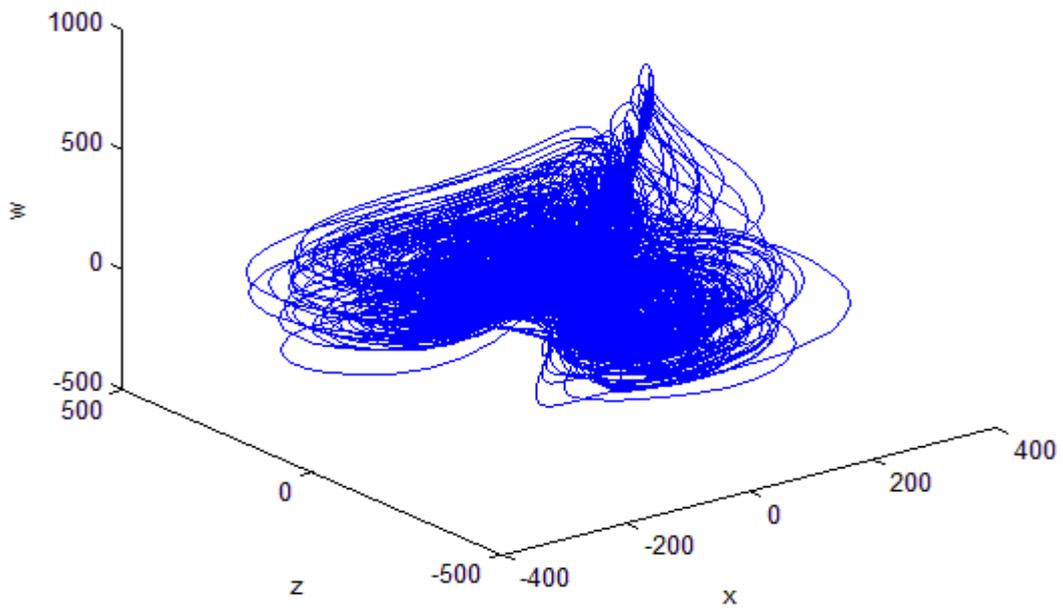


Figure 2.15. Attracteur étrange de Qi en fonction de x , z et w .

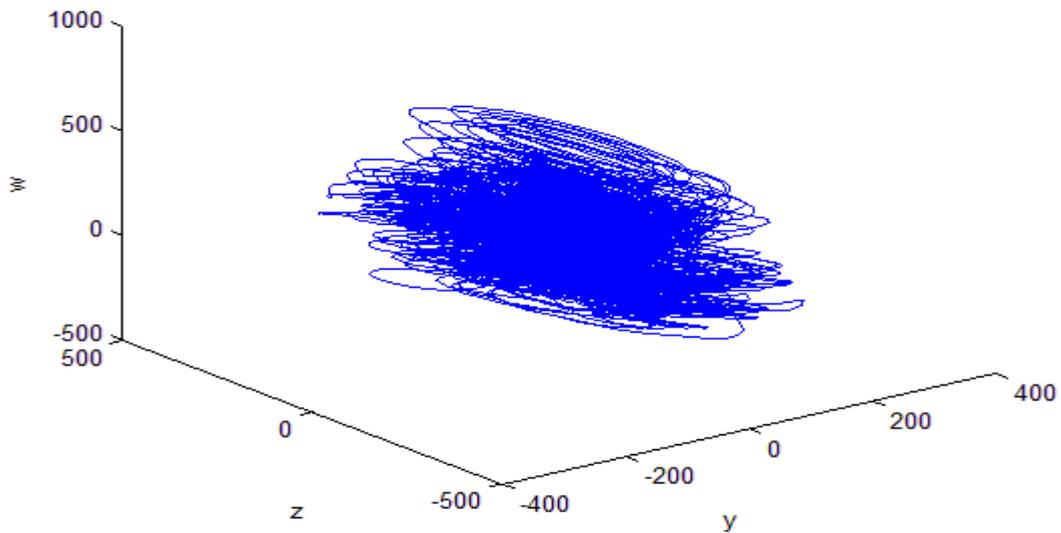


Figure 2.16. Attracteur étrange de Qi en fonction de y, z et w .

2.3.5 Exposants de Lyapunov

Nous allons utiliser l'outil MATDS qui travaille sous MATLAB, et qui nous permet d'étudier les systèmes dynamiques.

Tous d'abord, on va commencer par la présentation de l'outil MATDS :

Lors ce qu'on lance le MATDS la fenêtre suivante va apparaître.

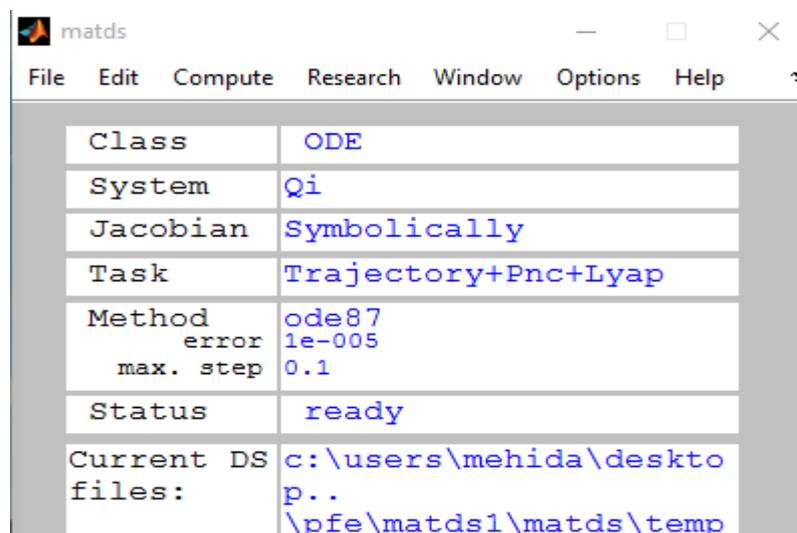


Figure 2.17. L'interface de l'outil MATDS.

- On va créer par la suite notre système hyper chaotique de Qi en faisant entrer les équations et les paramètres suivants :

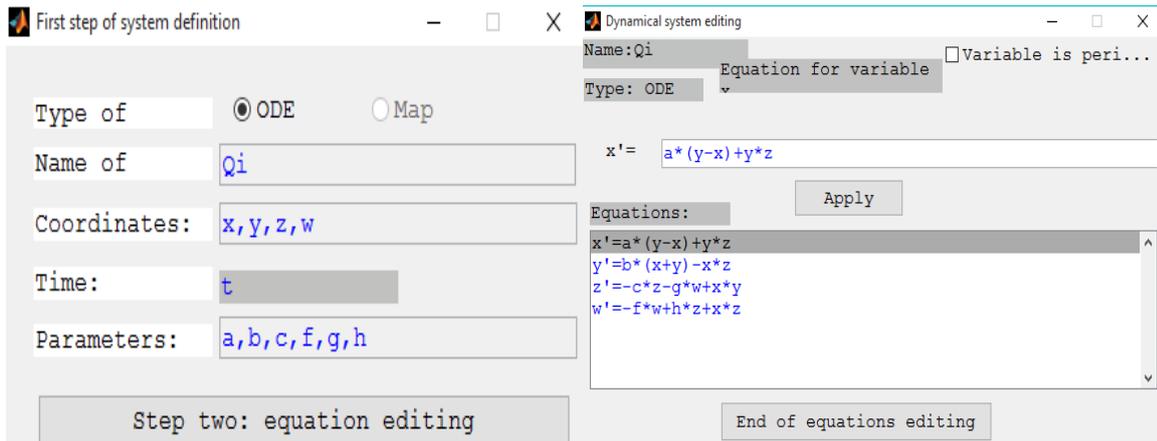


Figure 2.18. Création du système de Qi en MATDS.

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

On sait que pour un attracteur étrange (chaotique), il faut que la somme des exposants de Lyapunov soit négative et qu'au moins deux de ses exposant soient positifs. Pour notre système hyper chaotique de Qi, les exposants de Lyapunov sont représentés sur la figure 2.19.

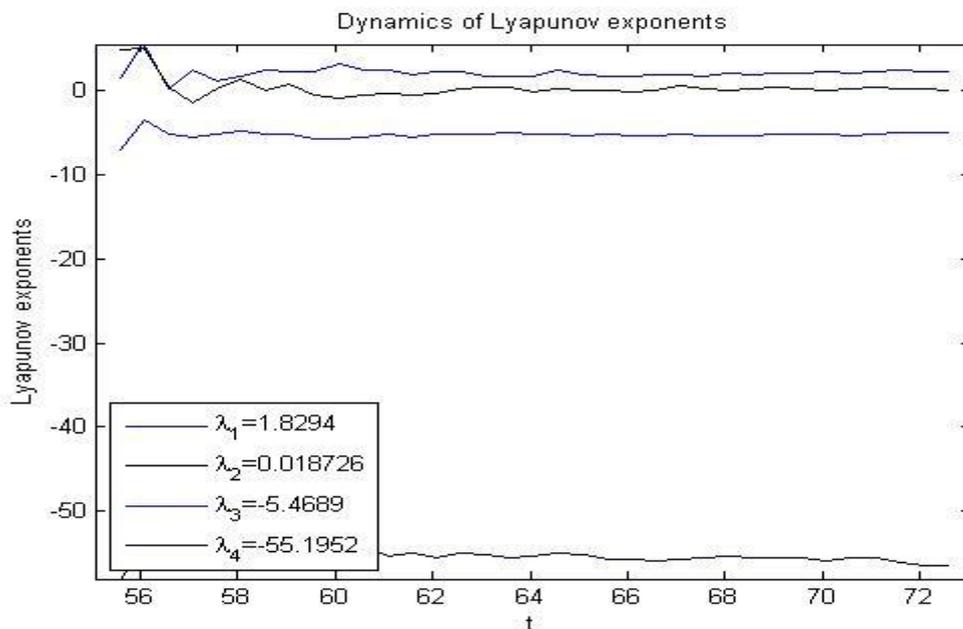


Figure 2.19. La dynamique des exposants de Lyapunov du système de Qi.

Notre système hyper chaotique de Qi a quatre équations différentielles donc on obtient quatre exposants de Lyapunov :

$$\lambda_1=1.8294, \lambda_2=0.018726, \lambda_3= - 5.4689, \lambda_4= - 55.1952$$

2.3.6 Section de Poincaré

On va tracer l'intersection du plan de phase y en fonction de x avec le plan

$x = 0$, ainsi on obtient la section de Poincaré mentionnée par les points rouges sur la figure 2.20.

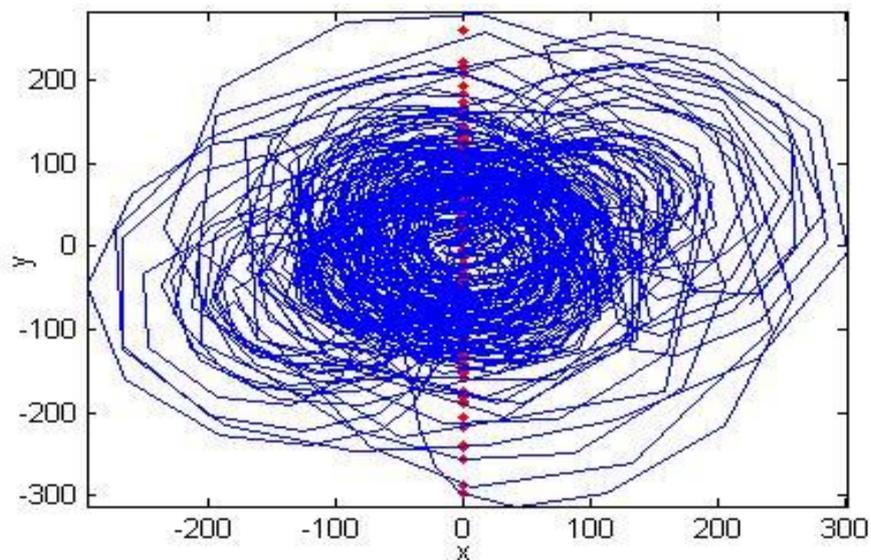


Figure 2.20. La section de Poincaré sur l'attracteur de Qi.

2.3.7 Diagramme de bifurcation

Pour le tracé du diagramme de bifurcation, un programme MATLAB a été écrit et les résultats obtenus sont représentés sur la figure 2.21. Le paramètre variable utilisé est b .

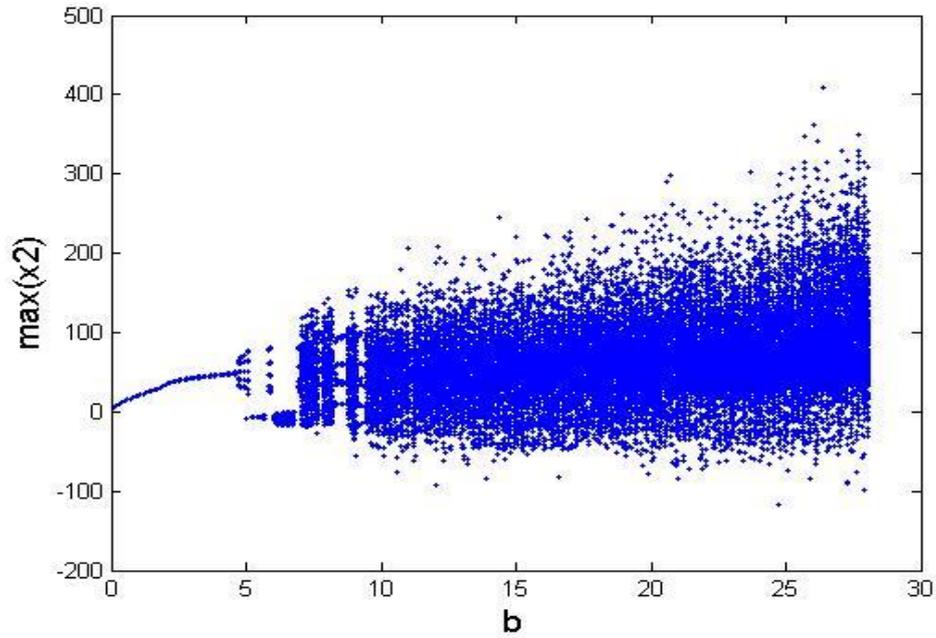


Figure 2.21. Diagramme de bifurcation de y en fonction de b .

Lorsque $0 < b < 5$ le système est périodique.

Lorsque $5 < b < 10$: on a une cascade de doublement de périodes c'est la route vers le chaos.

Lorsque $10 < b < 28$ le système est chaotique les figures précédentes le confirme.

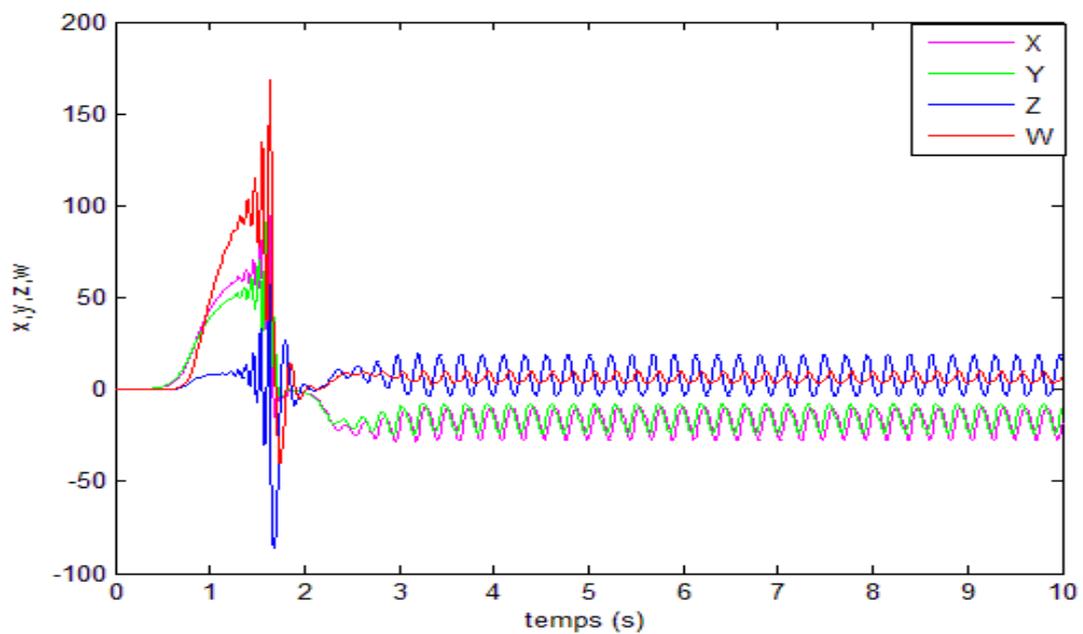


Figure 2.22. Les états x, y, z et w ont la même période lorsque $b=5$.

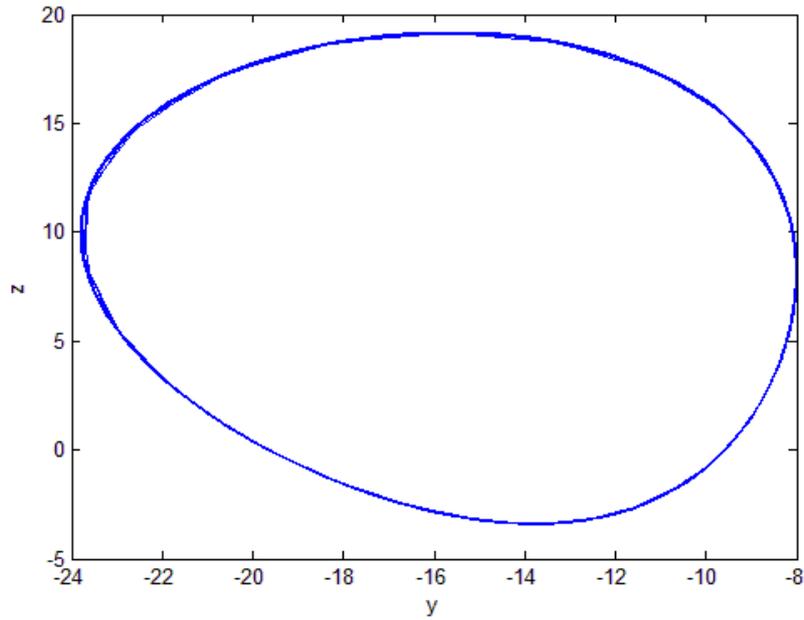


Figure 2.23. Plan de phase y en fonction de z lorsque $b=5$.

2.4 Conclusion

Dans ce chapitre on a étudié le système hyper chaotique de Q_i , son comportement et ses caractéristiques, on a vu la simulation de ce système sur Matlab, Simulink et MATDS et on a confirmé les résultats de la simulation avec les résultats obtenus avec l'oscilloscope. Dans le chapitre suivant nous présente les différentes méthodes de synchronisation du système Q_i puis présenter les résultats de simulation par MATLAB de l'émetteur et le récepteur.

Chapitre3 Synchronisation du système hyper chaotique de Qi

3.1 Introduction

Durant la décennie passée, plusieurs types de synchronisation ont été étudiés, et différentes méthodes ont été proposées. On distingue deux modes de synchronisation. Le premier mode repose sur un couplage mutuel entre deux systèmes chaotiques ou plus. Le second est appelé maître-esclave ou couplage unidirectionnel [8].

Dans ce chapitre nous développons trois lois de commande permettant la synchronisation maître-esclave de deux systèmes de Qi.

3.2 La synchronisation

La synchronisation de deux systèmes dynamiques signifie qu'un système évolue suivant le comportement d'autre système.

Il existe deux classes de synchronisation suivant la manière avec laquelle les deux systèmes chaotiques sont couplés ; on distingue la synchronisation unidirectionnelle et la synchronisation bidirectionnelle.

3.2.1 Synchronisation unidirectionnelle

Le maître : est un système indépendant.

L'esclave : est un système qui est dépendant du système maître.

Dans ce cas, on n'a qu'un seul système maître qui est l'émetteur et un seul système esclave qu'est le récepteur. Donc la synchronisation est dans un seul sens.

Dans ce cas le couplage entre les deux systèmes maître et esclave est représenté par la figure 3.1:

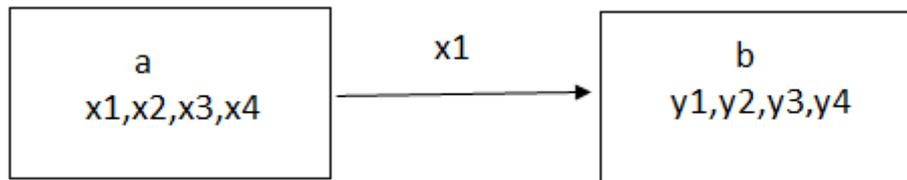


Figure 3.1. Le couplage unidirectionnel.

3.2.2 Synchronisation bidirectionnelle

Dans ce cas chaque système (l'émetteur et le récepteur) peut jouer le rôle d'un maître et d'un esclave en même temps, c'est-à-dire que la synchronisation est dans les deux sens.

Dans ce cas le couplage entre les deux systèmes maître et esclave est représenté par la figure 3.2:

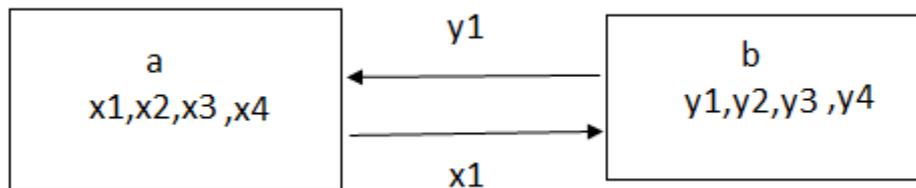


Figure 3.2. Le couplage bidirectionnel.

3.2.3 Nécessité de la synchronisation

La synchronisation des systèmes chaotiques semble impossible dans un premier temps, notamment à cause de la sensibilité de ces systèmes aux conditions initiales.

De plus, un système chaotique n'est pas asymptotiquement stable, c'est-à-dire que les trajectoires issues des conditions initiales voisines (légèrement différentes) divergent exponentiellement avec le temps.

Deux systèmes chaotiques pourraient se synchroniser sous certaines conditions.

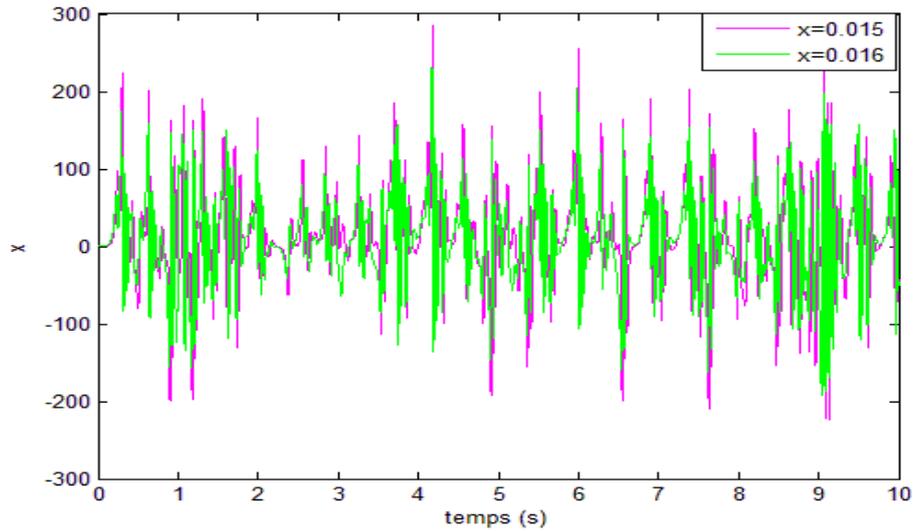


Figure 3.3 Évolution de deux systèmes chaotiques issus des conditions initiales très proches.

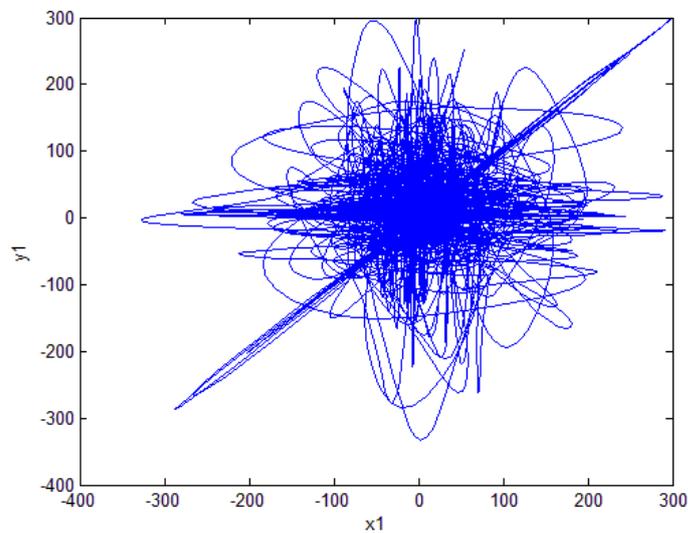


Figure 3.4 Plan de phase y_1 en fonction de x_1 .

3.3 Méthodes de synchronisation

Plusieurs méthodes de synchronisation ont été proposées dans la littérature. Dans ce qui suit nous appliquons trois approches permettant d'obtenir la synchronisation de deux systèmes de Q_i .

3.3.1 Synchronisation par la méthode du contrôle actif

L'application du contrôle actif pour la synchronisation des systèmes chaotiques a été proposée dans [9], c'est une technique efficace qui a montré sa puissance pour la synchronisation des systèmes hyper chaotiques. De plus, cette méthode offre une simplicité remarquable pour l'implémentation de l'algorithme. Le principe de l'algorithme est le suivant :

Soit deux systèmes hyper chaotiques à synchroniser. Un système maître défini par le système d'ODE 3.1 :

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1) + x_2x_3 \\ \dot{x}_2 = b(x_1 + x_2) - x_1x_3 \\ \dot{x}_3 = -cx_3 - gx_4 + x_1x_2 \\ \dot{x}_4 = -fx_4 + hx_3 + x_1x_3 \end{cases} \quad (3.1)$$

Où x_1, x_2, x_3 et x_4 désignent les variables d'état du système et a, b, c, f, g, h des paramètres réels constants et positifs.

Et un système identique esclave donné par le système d'ODE 3.2 :

$$\begin{cases} \dot{y}_1 = a(y_2 - y_1) + y_2y_3 + U_1 \\ \dot{y}_2 = b(y_1 + y_2) - y_1y_3 + U_2 \\ \dot{y}_3 = -cy_3 - gy_4 + y_1y_2 + U_3 \\ \dot{y}_4 = -fy_4 + hy_3 + y_1y_3 + U_4 \end{cases} \quad (3.2)$$

Où y_1, y_2, y_3 et y_4 désignent les variables d'état du système, et $U = [U_1, U_2, U_3, U_4]^T$ représente le vecteur de commande.

L'erreur de synchronisation entre les deux systèmes s'écrit sous la forme :

$$e_1 = y_1 - x_1, e_2 = y_2 - x_2, e_3 = y_3 - x_3, e_4 = y_4 - x_4$$

Et sa dynamique :

$$\dot{e}_1 = \dot{y}_1 - \dot{x}_1, \dot{e}_2 = \dot{y}_2 - \dot{x}_2, \dot{e}_3 = \dot{y}_3 - \dot{x}_3, \dot{e}_4 = \dot{y}_4 - \dot{x}_4$$

Pour que les deux systèmes se synchronisent, il faut que $\lim_{t \rightarrow \infty} e_i \rightarrow 0, i = 1, 2, 3$ et 4. Dans ce cas le problème de synchronisation se transforme à un problème de contrôle de l'erreur pour qu'elle se stabilise à la valeur zéro.

Après quelques étapes de calcul, la dynamique de l'erreur est obtenue comme suit (3.3) :

$$\begin{cases} \dot{e}_1 = a(e_2 - e_1) + e_2e_3 + x_3e_2 + x_2e_3 + U_1 \\ \dot{e}_2 = b(e_1 + e_2) - e_1e_3 - x_3e_1 - x_1e_3 + U_2 \\ \dot{e}_3 = -ce_3 - ge_4 + e_1e_2 + x_2e_1 + x_1e_2 + U_3 \\ \dot{e}_4 = -fe_4 - he_3 + e_1e_3 + x_3e_1 + x_1e_3 + U_4 \end{cases} \quad (3.3)$$

Il existe plusieurs choix possibles pour le contrôleur U. on choisit les fonctions de contrôle actif $U_i(t)$ ($i = 1,2,3,4$) de la manière suivante :

$$\begin{aligned} U_1 &= -e_2e_3 - x_3e_2 - x_2e_3 - k_1e_1 \\ U_2 &= e_1e_3 + x_3e_1 + x_1e_3 - k_2e_2 \\ U_3 &= -e_1e_2 - x_2e_1 - x_1e_2 - k_3e_3 \\ U_4 &= -e_1e_3 - x_3e_1 - x_1e_3 - k_4e_4 \end{aligned} \quad (3.4)$$

La substitution de (3.4) en (3.3), on obtient la dynamique de l'erreur sous forme linéaire :

$$\begin{cases} \dot{e}_1 = a(e_2 - e_1) - k_1e_1 \\ \dot{e}_2 = b(e_1 + e_2) - k_2e_2 \\ \dot{e}_3 = -ce_3 - ge_4 - k_3e_3 \\ \dot{e}_4 = -fe_4 + he_3 - k_4e_4 \end{cases} \quad (3.5)$$

D'où k_i ($i = 1,2,3,4$) sont des gains de rétroaction. Par conséquent, le système d'erreur (3.5) peut être réécrit sous la forme :

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \end{bmatrix} = \begin{bmatrix} -(a + k_1) & a & 0 & 0 \\ b & (b - k_2) & 0 & 0 \\ 0 & 0 & -(c + k_3) & -g \\ 0 & 0 & h & -(f + k_4) \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} \quad (3.6)$$

L'équation caractéristique du système 3.5 est donnée par :

$$\det(\lambda I - A) = [(\lambda + (a + k_1))(\lambda - (b - k_2)) - ab][(\lambda + (c + k_3))(\lambda + (f + k_4)) + gh]$$

Où I est la matrice identité de dimension (4x4).

Selon le critère de Routh-Hurwitz, pour n'importe quelle valeur de k_i ($i = 1,2,3,4$), le système bouclé est asymptotiquement stable si les conditions (3.7) sont remplies :

$$\begin{cases} (a + k_1) - (b - k_2) > 0 \\ -(a + k_1)(b - k_2) - ab > 0 \\ (c + k_3) + (f + k_4) > 0 \\ (c + k_3)(f + k_4) + gh > 0 \end{cases} \quad (3.7)$$

3.3.1.1 Résultats de simulation

Pour illustrer l'efficacité de la méthode proposée, quelques résultats de simulation sont représentés par les figures 3.5 à 3.10.

Un comportement chaotique du système (3.1) est obtenu pour les valeurs suivantes des paramètres : $a=50$, $b=24$, $c=13$, $g=33$, $f=8$ et $h=30$.

Au départ les deux systèmes évoluent à partir de deux conditions initiales différentes :

$$x_1 = 0.1, x_2 = 0.1, x_3 = 0.05, x_4 = 0.2$$

Et : $y_1 = 0.1, y_2 = 0.7, y_3 = 0.18, y_4 = 0.35$

Les gains k_i peuvent être choisis comme suit :

$$k_1 = k_2 = 50, k_3 = k_4 = 0.$$

Les figures 3.6 à 3.10 représentent la synchronisation des systèmes (maître, esclave)

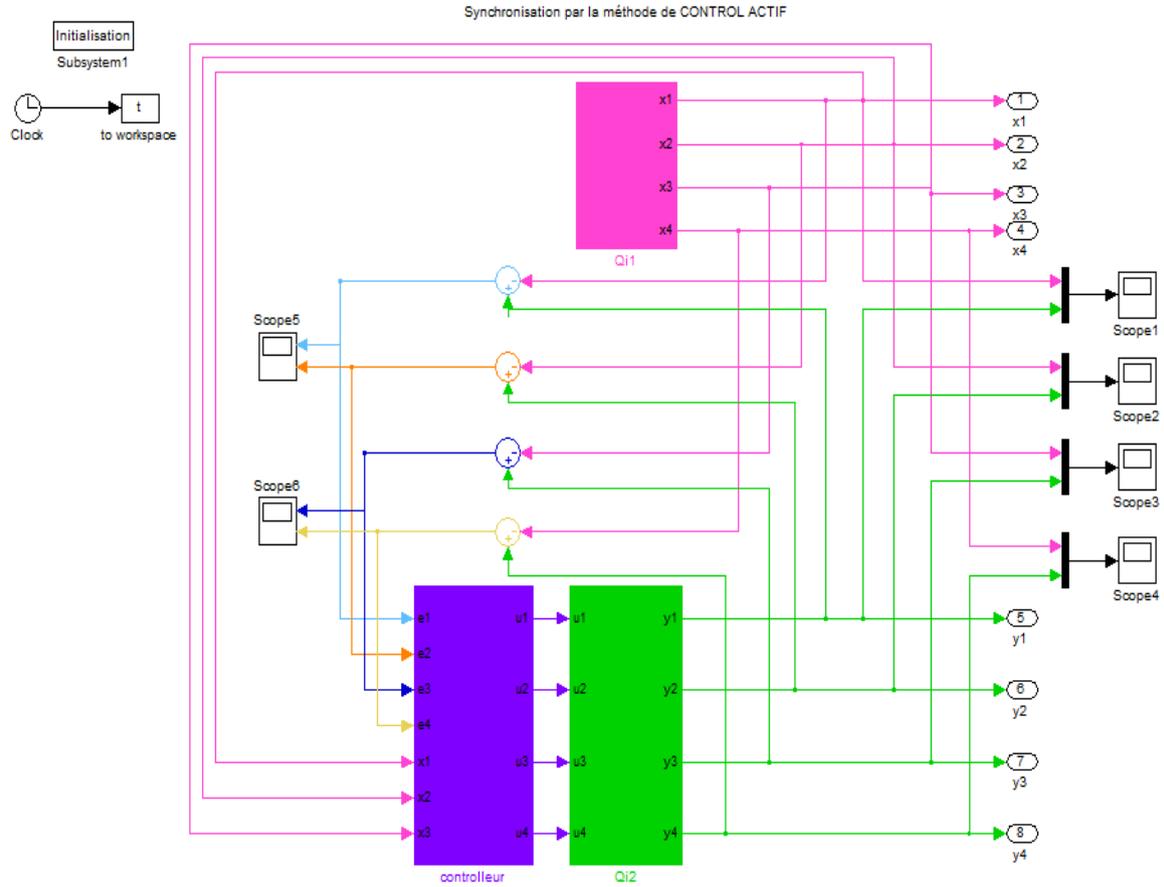


Figure 3.5. Schéma de synchronisation sous Matlab.

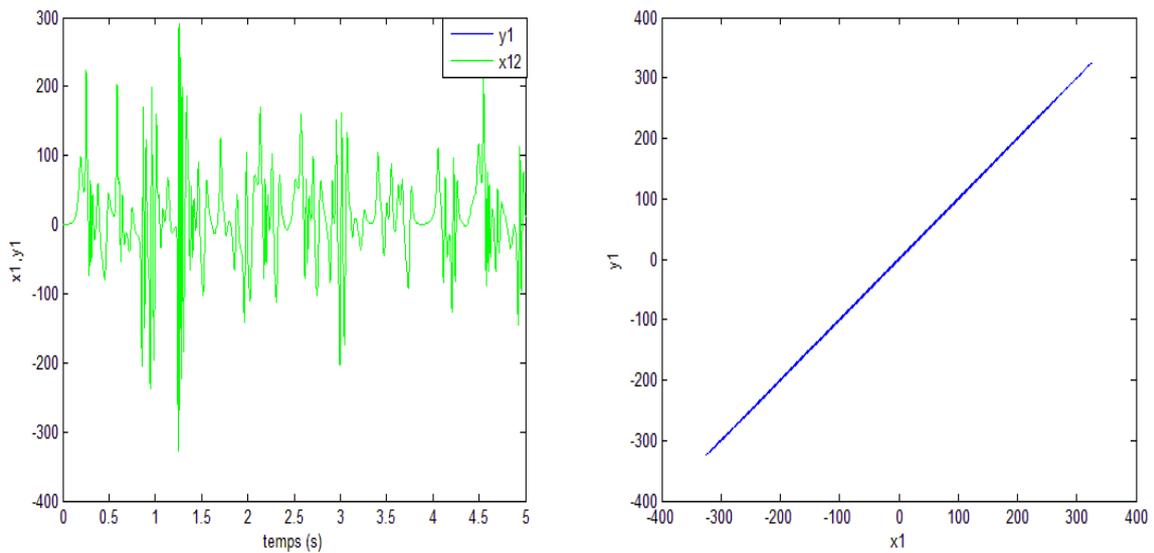


Figure 3.6. Synchronisation des signaux x_1 et y_1 .

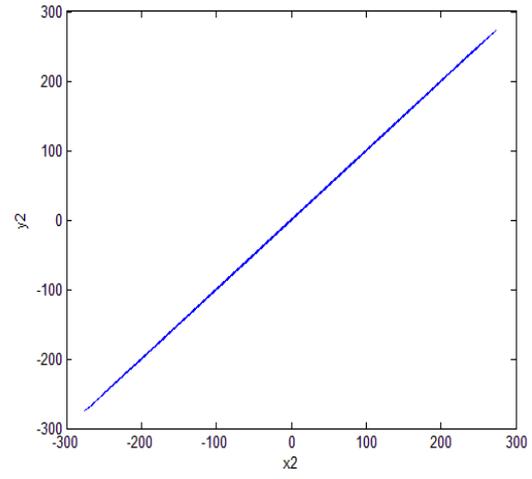
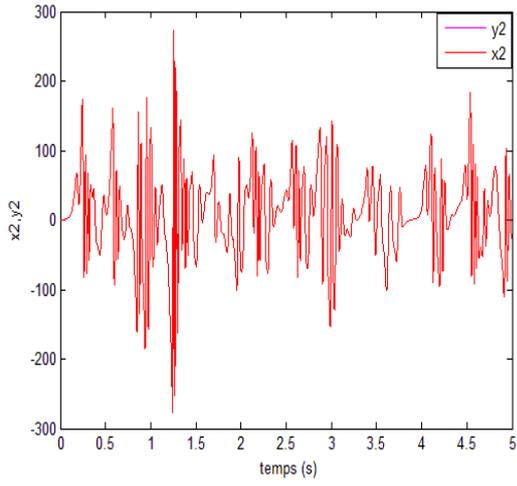


Figure 3.7. Synchronisation des signaux x_2 et y_2 .

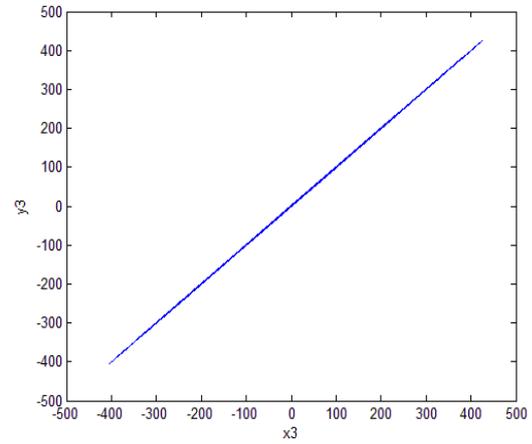
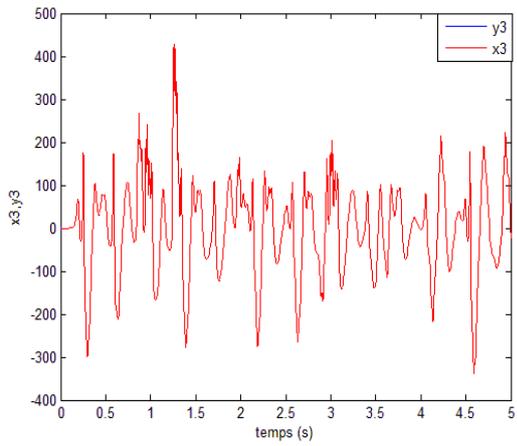


Figure 3.8. Synchronisation des signaux x_3 et y_3 .

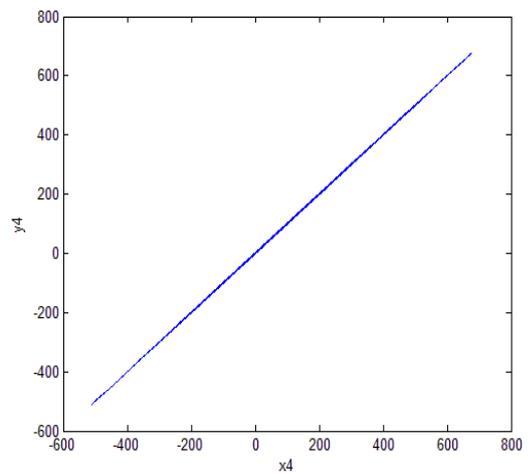
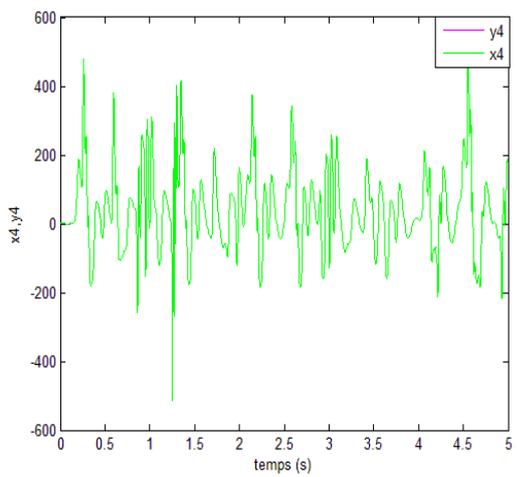


Figure 3.9. Synchronisation des signaux x_4 et y_4 .

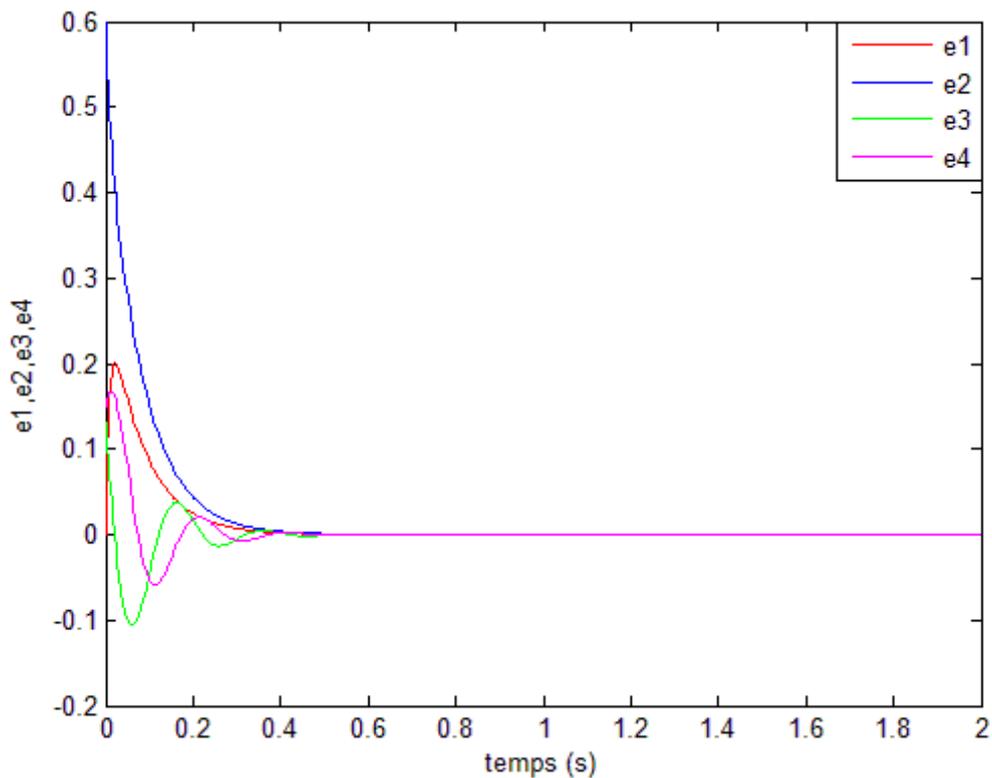


Figure 3.10. Les erreurs de synchronisation e_1 , e_2 , e_3 et e_4 .

On remarque qu'au début, les deux systèmes varient selon une dynamique différente, mais après un temps très court les trajectoires commencent à se rapprocher et deviennent identiques.

3.3.2 Synchronisation de deux systèmes hyper chaotique en utilisant la fonction de Lyapunov

3.3.2.1 La synchronisation par la méthode du contrôle actif

L'expression du contrôleur $U=[U_1, U_2, U_3, U_4]^T$ mentionnée dans le paragraphe précédent, on peut voir que, pour réaliser la synchronisation, il faut accéder à toutes les variables d'état du système. Mais dans de nombreux problèmes pratiques, les variables de système ne sont pas toutes mesurées. En outre, la méthode de synchronisation est meilleure lorsque le nombre de contrôleurs de processus sont réduits. Dans la suite, nous abordons un autre schéma de synchronisation pour lequel l'erreur n'est pas linéaire.

Le système maître est donné par l'équation (3.1). Le système esclave est représenté par (3.8).

$$\begin{aligned}
 \dot{y}_1 &= a(y_2 - y_1) + y_2 y_3 + W_1 \\
 \dot{y}_2 &= b(y_1 + y_2) - y_1 y_3 + W_2 \\
 \dot{y}_3 &= -c y_3 - g y_4 + y_1 y_2 + W_3 \\
 \dot{y}_4 &= -f y_4 + h y_3 + y_1 y_3
 \end{aligned} \tag{3.8}$$

Nous contrôlons les trois premiers états du système esclave à l'aide des contrôles W_1, W_2 et W_3 .

La dynamique de l'erreur est définie par :

$$\begin{aligned}
 \dot{e}_1 &= a(e_2 - e_1) + e_2 e_3 + x_3 e_2 + x_2 e_3 + W_1 \\
 \dot{e}_2 &= b(e_1 + e_2) - e_1 e_3 - x_3 e_1 - x_1 e_3 + W_2 \\
 \dot{e}_3 &= -c e_3 - g e_4 + e_1 e_2 + x_2 e_1 + x_1 e_2 + W_3 \\
 \dot{e}_4 &= -f e_4 + h e_3 + e_1 e_3 + x_3 e_1 + x_1 e_3
 \end{aligned} \tag{3.9}$$

On pose $W_2 = -2b e_2$, la dynamique de l'erreur devient :

$$\begin{cases}
 \dot{e}_1 = a(e_2 - e_1) + e_2 e_3 + x_3 e_2 + x_2 e_3 + W_1 \\
 \dot{e}_2 = b(e_1 - e_2) - e_1 e_3 - x_3 e_1 - x_1 e_3 \\
 \dot{e}_3 = -c e_3 - g e_4 + e_1 e_2 + x_2 e_1 + x_1 e_2 + W_3 \\
 \dot{e}_4 = -f e_4 + h e_3 + e_1 e_3 + x_3 e_1 + x_1 e_3
 \end{cases} \tag{3.10}$$

On choisit la fonction de Lyapunov (3.11). C'est une fonction définie positive :

$$V = \frac{1}{2}(e_1^2 + e_2^2 + e_3^2) \tag{3.11}$$

La dérivée de la fonction de Lyapunov s'écrit (3.12) :

$$\begin{aligned}
 \frac{dV}{dt} &= e_1 \dot{e}_1 + e_2 \dot{e}_2 + e_3 \dot{e}_3 \\
 &= -a e_1^2 - b e_2^2 - c e_3^2 + e_1 [(a + b) e_2 + W_1] + e_3 (2x_2 e_1 + e_1 e_2 - g e_4 + W_3)
 \end{aligned} \tag{3.12}$$

Si on choisit :

$$\begin{aligned}
 W_1 &= -(a + b) e_2 \\
 W_3 &= g e_4 - 2x_2 e_1 - e_1 e_2
 \end{aligned}$$

Alors la dérivée de la fonction de Lyapunov devient (3.13):

$$\frac{dV}{dt} = -a e_1^2 - b e_2^2 - c e_3^2 \tag{3.13}$$

qui est une fonction définie négative.

Ce qui assure que le système esclave (3.8) est synchrone asymptotiquement avec le système maître (3.1).

3.3.2.1 Résultats de simulation

Les résultats de simulation de la synchronisation des deux systèmes chaotiques de Qi sont représentés sur les figures 3.11 à 3.16.

Les valeurs des paramètres du système sont choisis comme suit : $a=50$, $b=24$, $c=13$, $g=33$, $f=8$ et $h=30$.

Les états initiaux du système pilote sont : $x_1=0.7$, $x_2=1$, $x_3=0.05$, $x_4=0.2$ et du système esclave sont : $y_1=0.1$, $y_2=0.18$, $y_3=0.1$, $y_4=0.35$.

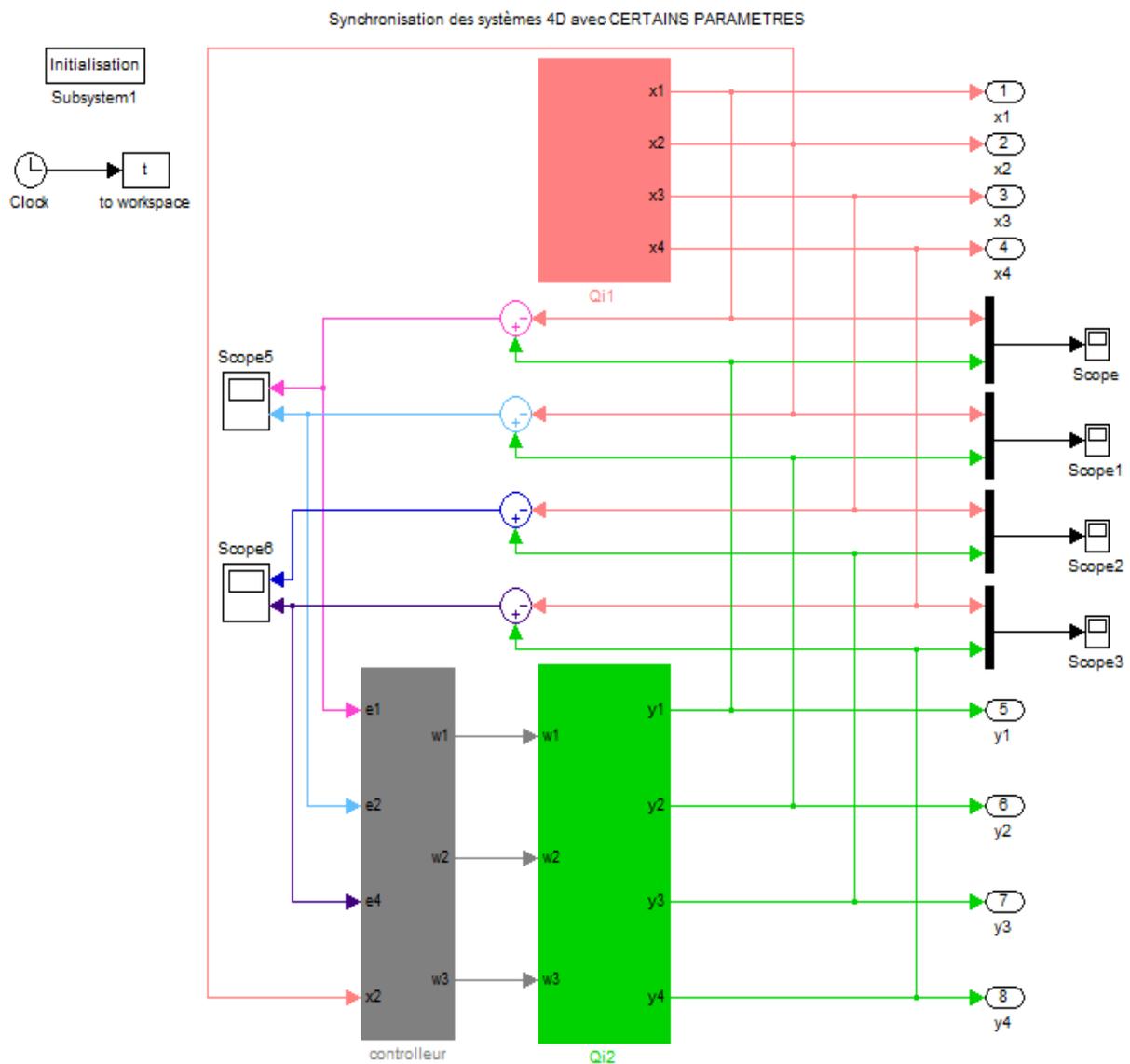


Figure 3.11. Schéma de synchronisation sous Matlab.

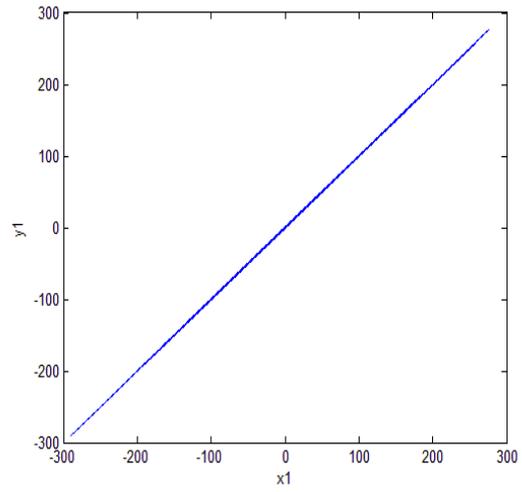
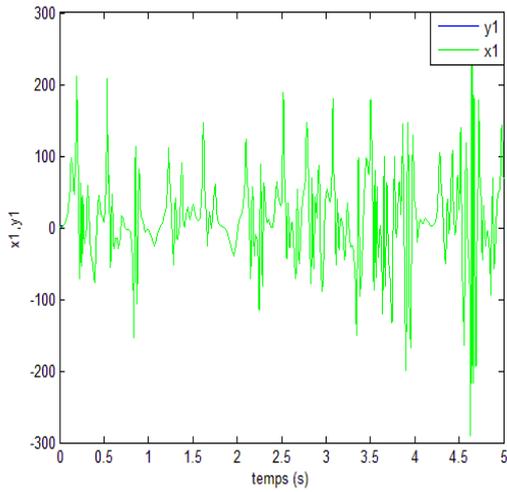


Figure 3.12. Synchronisation des signaux x_1 et y_1 .

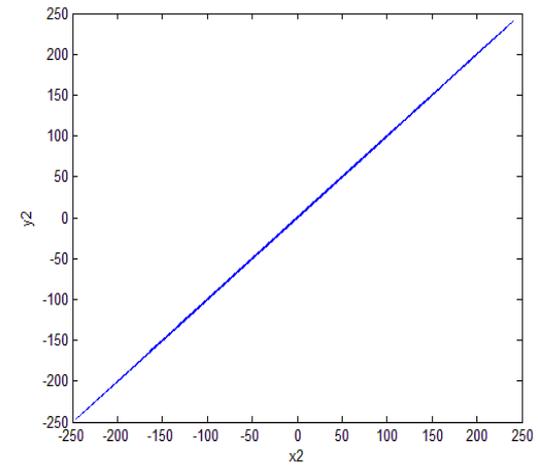
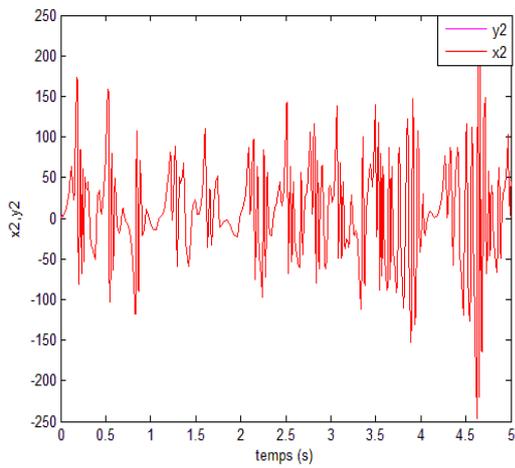


Figure 3.13. Synchronisation des signaux x_2 et y_2 .

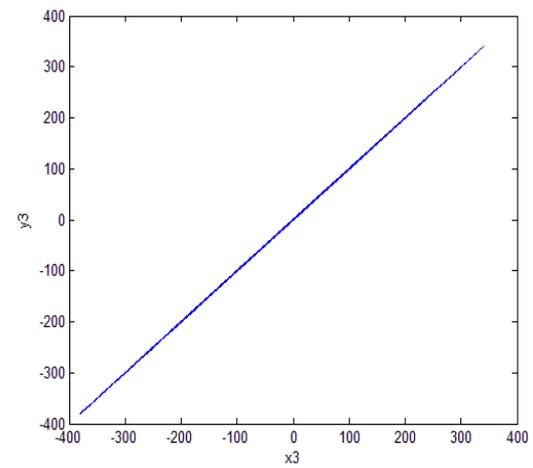
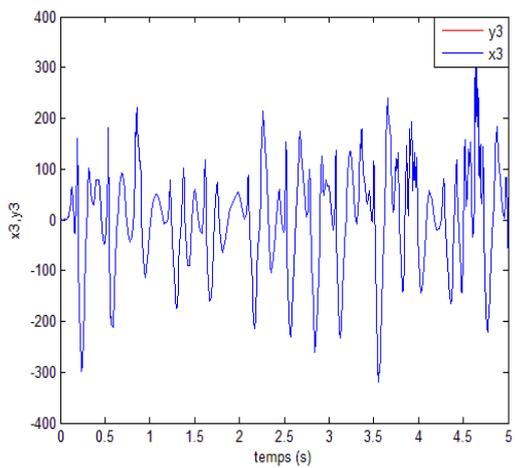


Figure 3.14. Synchronisation des signaux x_3 et y_3 .

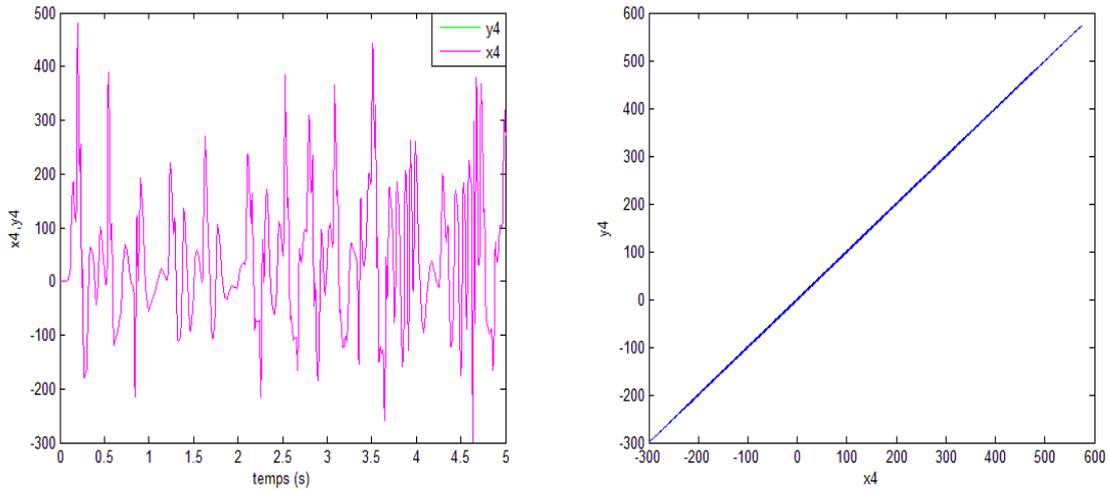


Figure 3.15. Synchronisation des signaux x_4 et y_4 .

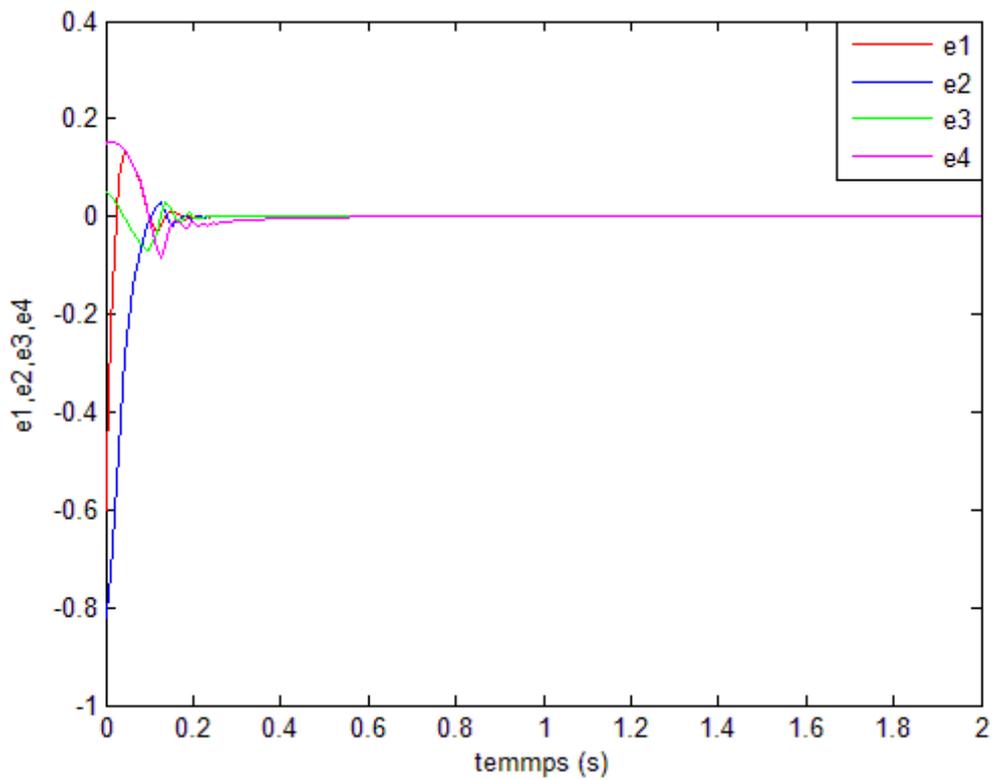


Figure 3.16. Les erreurs de synchronisation e_1, e_2, e_3 et e_4 .

Les différentes figures montrent que les trajectoires des variables d'état des systèmes maître (x_1, x_2, x_3, x_4) et esclave (y_1, y_2, y_3, y_4) convergent rapidement et deviennent identiques après un temps réduit. On remarque que le contrôleur calculé a parfaitement conduit le système esclave à suivre des trajectoires hyper chaotiques du maître. De plus la

stabilité est garantie puisque les lois de commande sont tirées de la fonction de Lyapunov, ce qui assure une stabilité asymptotique.

3.3.2.2 Synchronisation par la méthode du contrôle adaptatif

Pratiquement, une partie ou l'ensemble des paramètres du système sont incertains et peuvent changer au cours du temps. Pour cela, une approche de contrôle adaptatif pour la synchronisation des systèmes hyper chaotique incertains est proposée.

Soit le système maître (3.1) et esclave (3.14) représentés respectivement par les équations suivantes:

$$\begin{cases} \dot{y}_1 = \hat{a}(y_2 - y_1) + y_2 y_3 + W_1 \\ \dot{y}_2 = \hat{b}(y_1 + y_2) - y_1 y_3 + W_2 \\ \dot{y}_3 = -\hat{c}y_3 - \hat{g}y_4 + y_1 y_2 + W_3 \\ \dot{y}_4 = -\hat{f}y_4 + \hat{h}y_3 + y_1 y_3 + W_4 \end{cases} \quad (3.14)$$

Ou a, b, c, f, g et h sont des paramètres réels constants et positives du système maître (3.1)

et $\hat{a}, \hat{b}, \hat{c}, \hat{g}, \hat{f}$ et \hat{h} sont des paramètres réels constants et positives du système esclave (3.14)

$W_i = [w_1, w_2, w_3, w_4]^T$ représente le vecteur de contrôle à calculer.

La dynamique de l'erreur entre les deux systèmes s'écrit sous la forme (3.15) :

$$\begin{cases} \dot{e}_1 = \hat{a}(e_2 - e_1) - \tilde{a}(x_2 - x_1) + e_2 e_3 + x_3 e_2 + x_2 e_3 + W_1 \\ \dot{e}_2 = \hat{b}(e_1 + e_2) - \tilde{b}(x_1 + x_2) - e_1 e_3 - x_3 e_1 - x_1 e_3 + W_2 \\ \dot{e}_3 = -\hat{c}e_3 - \hat{g}e_4 + \tilde{c}x_3 + \tilde{g}x_4 + e_1 e_2 + x_2 e_1 + x_1 e_2 + W_3 \\ \dot{e}_4 = -\hat{f}e_4 + \hat{h}e_3 + \tilde{f}x_4 - \tilde{h}x_3 + e_1 e_3 + x_3 e_1 + x_1 e_3 + W_4 \end{cases} \quad (3.15)$$

On choisit la fonction de Lyapunov (3.16) :

$$V = \frac{1}{2}(e_1^2 + e_2^2 + e_3^2 + e_4^2 + \tilde{a}^2 + \tilde{b}^2 + \tilde{c}^2 + \tilde{g}^2 + \tilde{f}^2 + \tilde{h}^2) \quad (3.16)$$

Avec :

$$\tilde{a} = a - \hat{a}, \quad \tilde{b} = b - \hat{b}, \quad \tilde{c} = c - \hat{c}, \quad \tilde{g} = g - \hat{g}, \quad \tilde{f} = f - \hat{f}, \quad \tilde{h} = h - \hat{h}$$

La dérivée de la fonction de Lyapunov devient (3.17)

$$\frac{dV}{dt} = e_1 \dot{e}_1 + e_2 \dot{e}_2 + e_3 \dot{e}_3 + e_4 \dot{e}_4 + \tilde{a}(-\dot{\hat{a}}) + \tilde{b}(-\dot{\hat{b}}) + \tilde{c}(-\dot{\hat{c}}) + \tilde{g}(-\dot{\hat{g}}) + \tilde{f}(-\dot{\hat{f}}) + \tilde{h}(-\dot{\hat{h}}) \quad (3.17)$$

$$\begin{aligned}
&= e_1(\hat{a}(e_2 - e_1) - \tilde{a}(x_2 - x_1) + e_2e_3 + x_3e_2 + x_2e_3 + w_1) + \tilde{a}(-\dot{\hat{a}}) \\
&+ e_2(\hat{b}(e_1 + e_2) - \tilde{b}(x_1 + x_2) - e_1e_3 - x_3e_1 - x_1e_3 + w_2) + \tilde{b}(-\dot{\hat{b}}) \\
&+ e_3(-\hat{c}e_3 - \hat{g}e_4 + \tilde{c}x_3 + \tilde{g}x_4 + e_1e_2 + x_2e_1 + x_1e_2 + w_3) + \tilde{c}(-\dot{\hat{c}}) + \tilde{g}(-\dot{\hat{g}}) \\
&+ e_4(-\hat{f}e_4 + \hat{h}e_3 + \tilde{f}x_4 - \tilde{h}x_3 + e_1e_3 + x_3e_1 + x_1e_3 + w_4) + \tilde{f}(-\dot{\hat{f}}) + \tilde{h}(-\dot{\hat{h}})
\end{aligned}$$

Les lois de commandes qui réalisent la synchronisation sont :

$$\begin{cases}
w_1 = e_1(\hat{a} - k_1) - \hat{a}e_2 - e_2e_3 + x_3e_2 - x_2e_3 \\
w_2 = -e_2(\hat{b} + k_2) - \hat{b}e_1 + e_1e_3 + x_3e_1 + x_1e_3 \\
w_3 = e_3(\hat{c} - k_3) + \hat{g}e_4 - e_1e_2 - x_2e_1 - x_1e_2 \\
w_4 = e_4(\hat{f} - k_4) - \hat{h}e_3 - e_1e_3 - x_3e_1 - x_1e_3
\end{cases} \quad (3.18)$$

D'où k_i ($i = 1,2,3,4$) sont des gains.

Et les lois d'adaptation des paramètres incertains sont :

$$\begin{cases}
\dot{\hat{a}} = -e_1(x_2 - x_1) \\
\dot{\hat{b}} = -e_2(x_1 + x_2) \\
\dot{\hat{c}} = e_3x_3 \\
\dot{\hat{g}} = e_3x_4 \\
\dot{\hat{f}} = e_4x_4 \\
\dot{\hat{h}} = -e_4x_3
\end{cases} \quad (3.19)$$

La dérivée devient (3.20) :

$$\frac{dV}{dt} = -k_1e_1^2 - k_2e_2^2 - k_3e_3^2 - k_4e_4^2.$$

De toute évidence, V est définie négative. Par conséquent, sur la base de la théorie de la stabilité de Lyapunov, les systèmes maître et esclave sont asymptotiquement synchronisés.

3.3.2.2 Résultats de simulation

Les résultats de simulation de la synchronisation des deux systèmes chaotiques de Qi sont représentés sur les figures 3.17 à 3.28.

Pour la simulation, les paramètres du maître sont : $a=50, b=24, c=13, g=33, f=8, h=30$ et les paramètres incertains de l'esclave sont : $\hat{a} = 45, \hat{b} = 20, \hat{c} = 10, \hat{g} = 30, \hat{f} = 10, \hat{h} = 35$.

Pour des conditions initiales différentes (maître : $x_1=0.1, x_2=0.2, x_3=0.05, x_4=0.2$; l'esclave : $y_1=0.1, y_2=0.1, y_3=0.18, y_4=0.35$), et les gains k_i ($k_1 = k_2 = k_3 = k_4 = 150$) la simulation donne les résultats représentés par les figures 3.18 à 3.28.

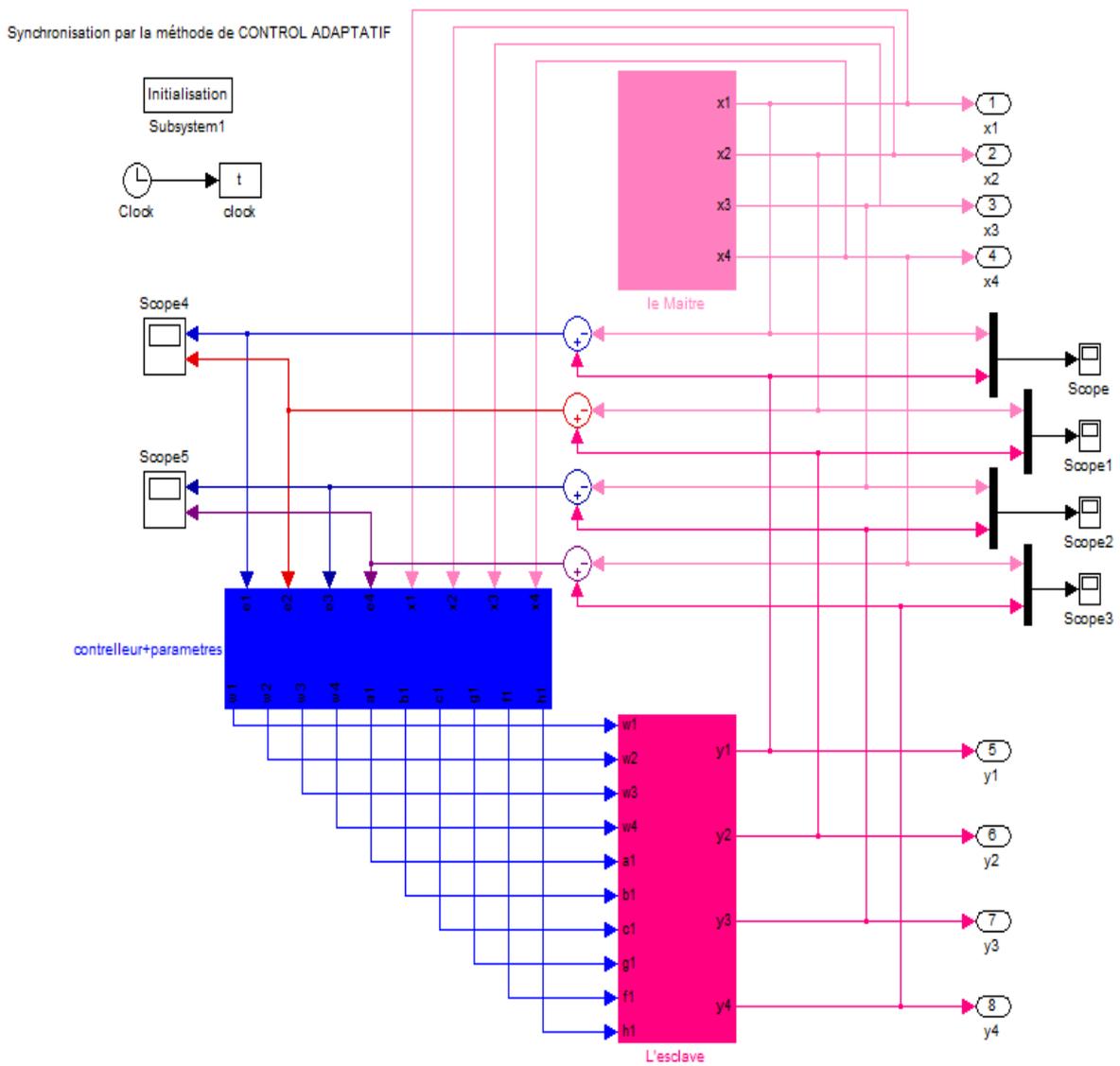


Figure 3.17. Schéma de synchronisation sous Matlab.

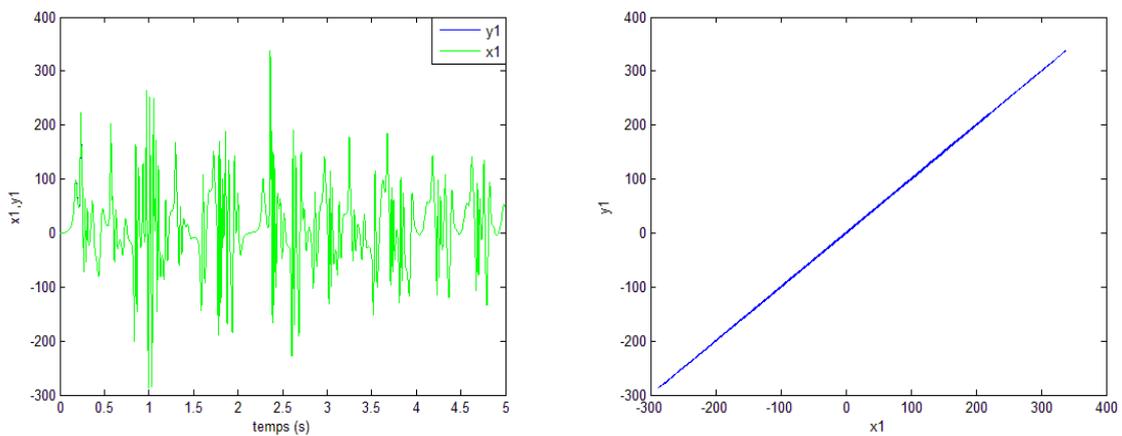


Figure 3.18. Synchronisation des signaux x1 et y1.

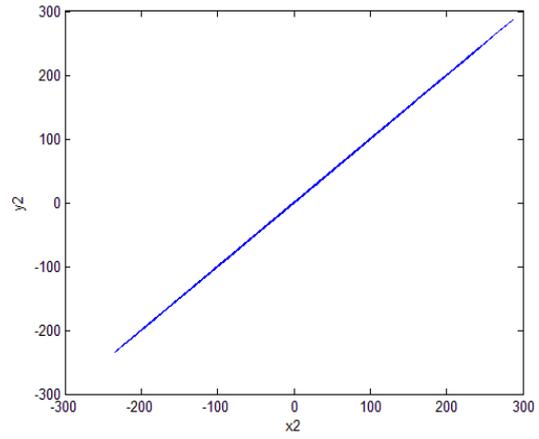
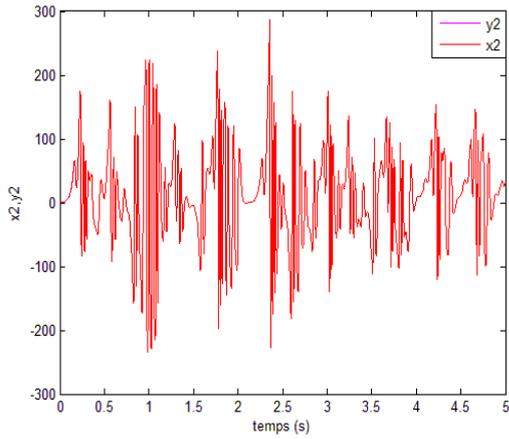


Figure 3.19. Synchronisation des signaux x_2 et y_2 .

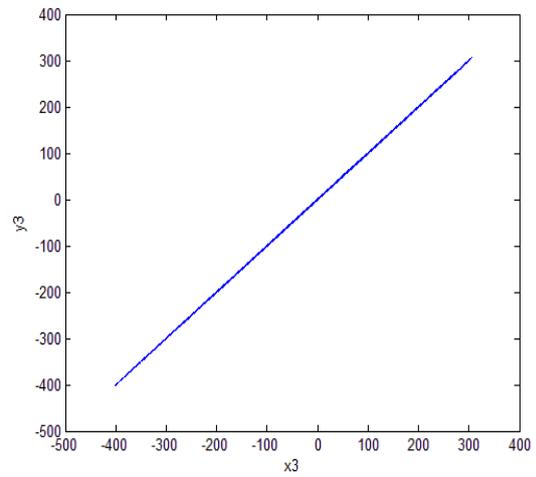
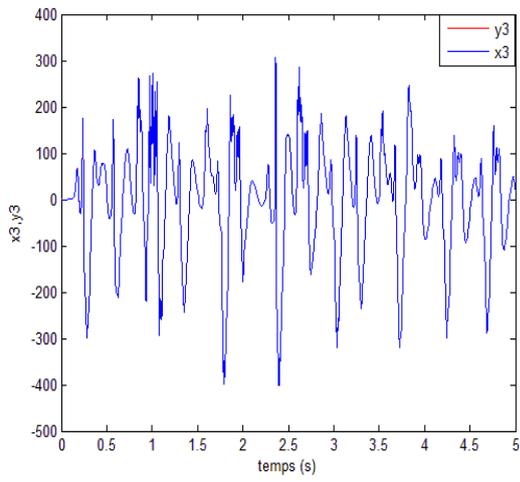


Figure 3.20. Synchronisation des signaux x_3 et y_3 .

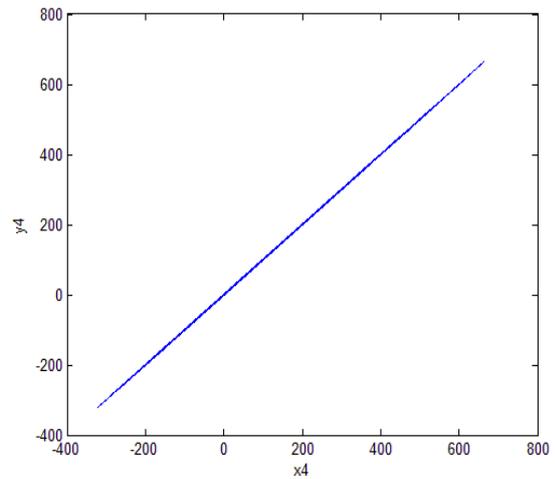
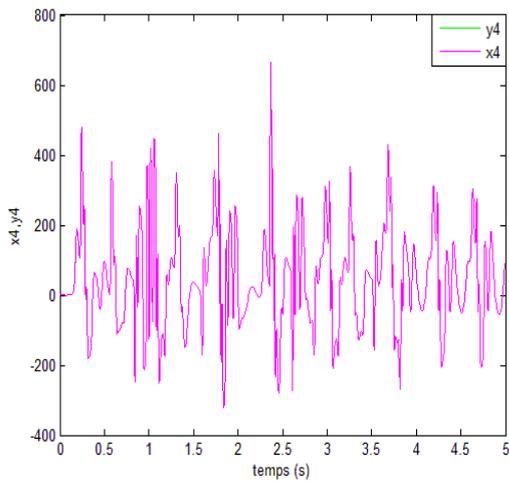


Figure 3.21. Synchronisation des signaux x_4 et y_4 .

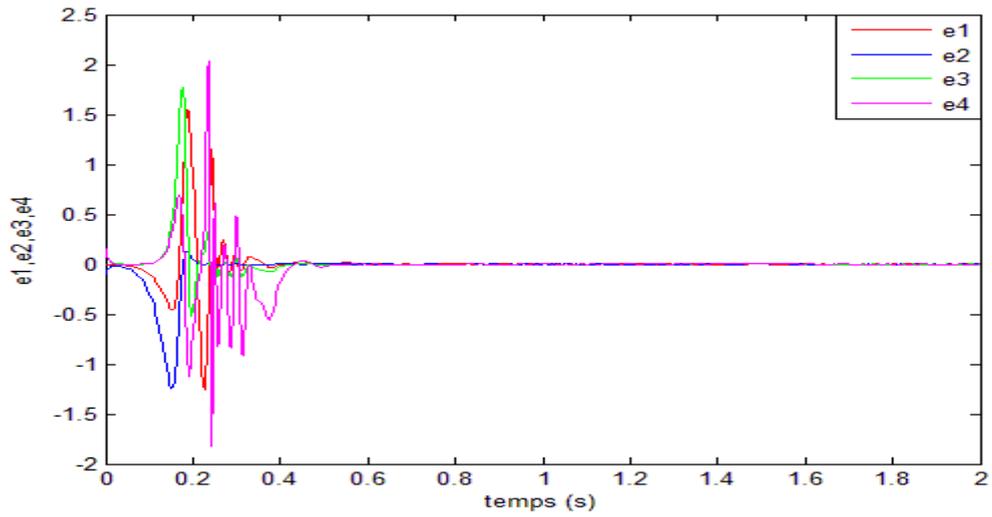


Figure 3.22. Les erreurs de synchronisation e_1 , e_2 , e_3 et e_4 .

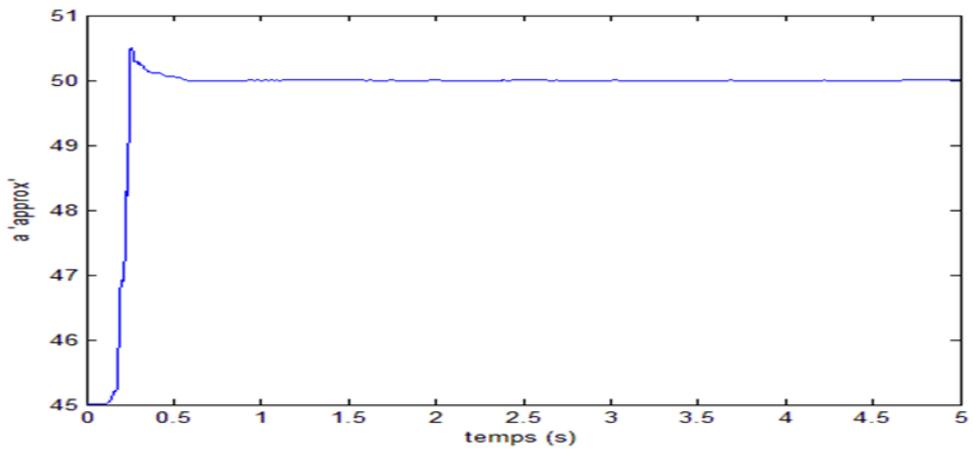


Figure 3.23. L'estimation de paramètre a.

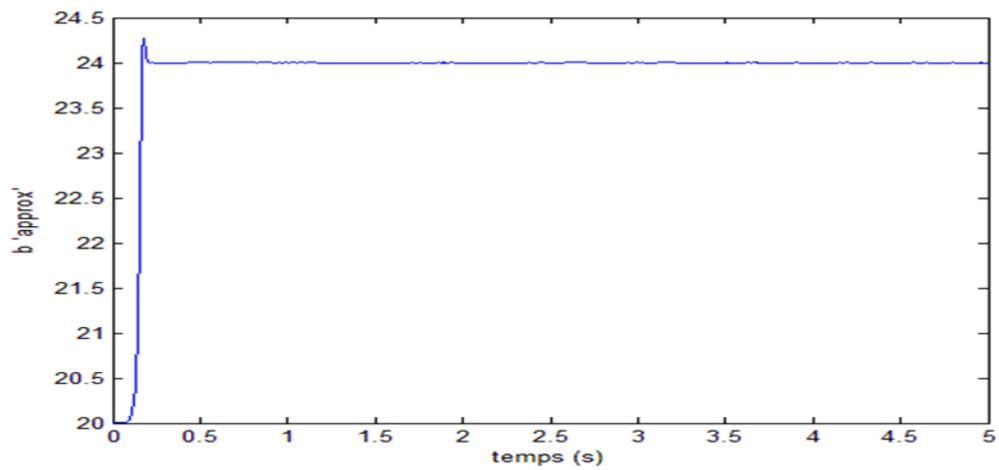


Figure 3.24. L'estimation de paramètre b.

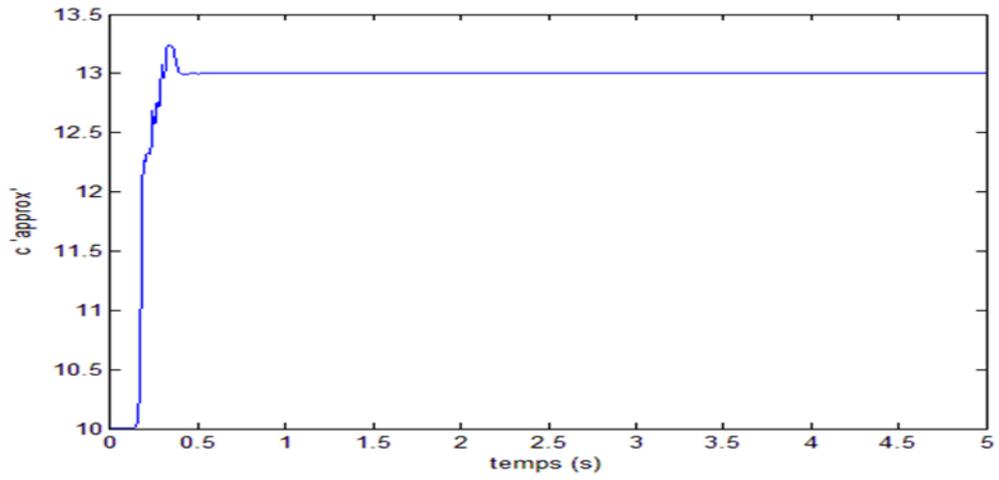


Figure 3.25. L'estimation de paramètre c.

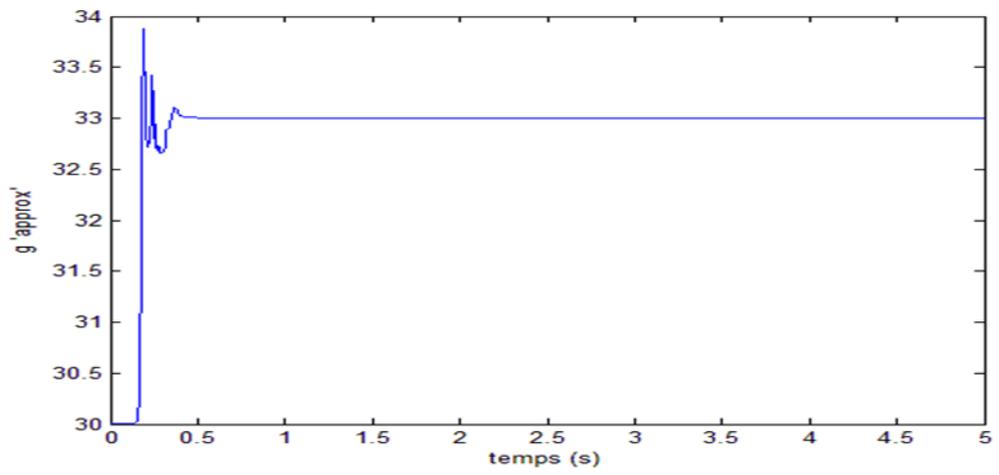


Figure 3.26. L'estimation de paramètre g.

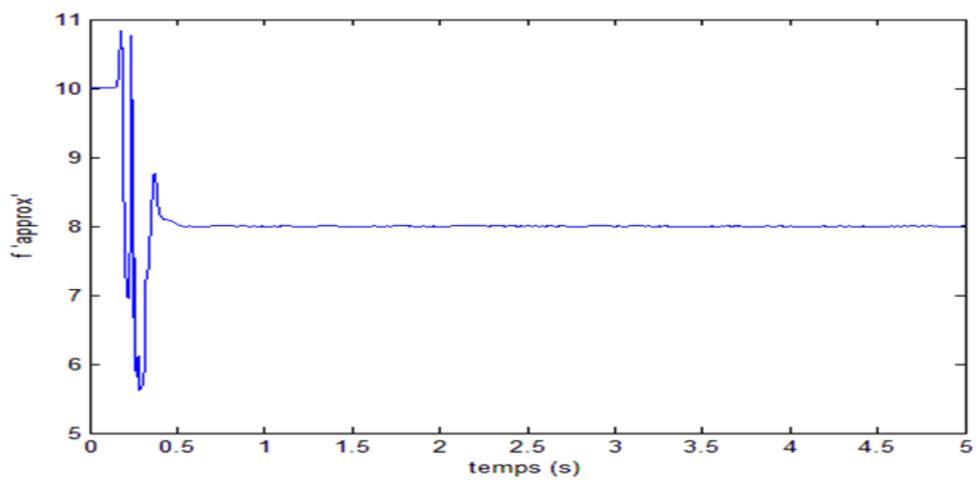


Figure 3.27. L'estimation de paramètre f.

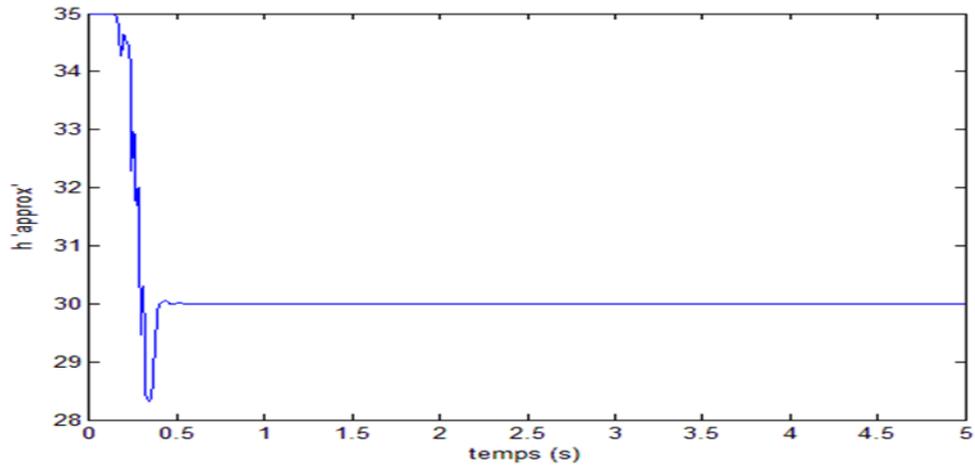


Figure 3.28. L'estimation de paramètre h.

Les figures montrent l'efficacité du contrôle adaptatif. On remarque que le contrôleur calculé a parfaitement conduit le système esclave à suivre des trajectoires hyper chaotiques du maître. De plus la stabilité est garantie puisque les lois de commande sont tirées de la fonction de Lyapunov, ce qui assure une stabilité asymptotique.

3.4 Conclusion

Dans ce chapitre, nous avons présenté les différents types de couplages (unidirectionnels et bidirectionnels), ainsi que les différentes méthodes de synchronisation, la synchronisation par contrôle actif a montré son efficacité sur les hyper chaotique mais pratiquement l'ensemble des paramètres du système sont incertains et même peuvent changer au cours du temps. Pour cela, on a proposé l'approche de synchronisation par contrôle adaptatif. Les résultats de simulation ont été présentés.

Chapitre 4 Implémentation du système Qi sur FPGA

4.1 Introduction

Les circuits FPGA sont les circuits qui ont permis de pousser plus loin les progrès technologiques en électronique. En effet, le degré de maturité des circuits FPGA fait d'eux aujourd'hui une solution pour remplacer les ASICs et les processeurs personnalisés dans des applications de contrôle et de traitement des signaux.

Nous présentons dans ce chapitre une implémentation du système hyper chaotique de Qi, sur la carte FPGA VIRTEX-5. Après avoir décrit brièvement la technologie FPGA, nous présentons le logiciel ISE de Xilinx. Nous décrivons ensuite le processus d'implémentation sur carte FPGA. Par la suite, les résultats de simulation du system de Qi à l'aide de logiciel ISE sont comparés avec ceux obtenu à l'aide de Simulink, obtenus au chapitre 3.

4.2 Description des composants FPGA

Les circuits FPGA (Field Programmable Gate Arrays) ou « réseaux logiques programmables » sont des composants VLSI (Very Large Scale Integration). Ils sont constitués d'une matrice de blocs logiques programmables entourés de blocs d'entrée/sortie programmables. L'ensemble est relié par un réseau d'interconnexions programmables. Entièrement reconfigurables ce qui permet de les reprogrammer à volonté afin d'accélérer notablement certaines phases de calculs. L'avantage de ce genre de circuit est sa grande souplesse qui permet de les réutiliser plusieurs fois dans des algorithmes différents en un temps très court. Le progrès de ces technologies permet de faire des composants toujours plus rapides et à plus haute intégration, ce qui permet de programmer des applications importantes [10].

4.3 Processus d'implémentation

La conception des architectures de commande s'effectue en utilisant les outils de conception assistée par ordinateur (CAO). La saisie est effectuée graphiquement ou via un langage de description matériel de haut niveau, nommé également langage HDL (Hardware Description Language), le langage HDL le plus couramment utilisés est VHDL (Very High Speed Integrated Hardware Description Language). Il est standardisé et offre au concepteur différents niveaux de description, et surtout l'avantage d'être portable et compatible avec toutes les technologies FPGA précédemment introduites. La figure 4.1 résume les différentes étapes de programmation d'une FPGA.

Le synthétiseur des outils CAO génère dans un premier temps une Netlist qui décrit la connectivité de l'architecture, après avoir vérifié la syntaxe, il génère aussi un schéma RTL et un schéma technologique qui décrivent le programme HDL. Puis l'outil de placement-routage place de façon optimale tous les composants et effectue le routage entre les différentes cellules logiques. Ces deux étapes permettent de générer un fichier de configuration à télécharger dans la mémoire de configuration du FPGA. Ce fichier est appelé Bit Stream et peut être directement chargé sur FPGA à partir d'un ordinateur [11].

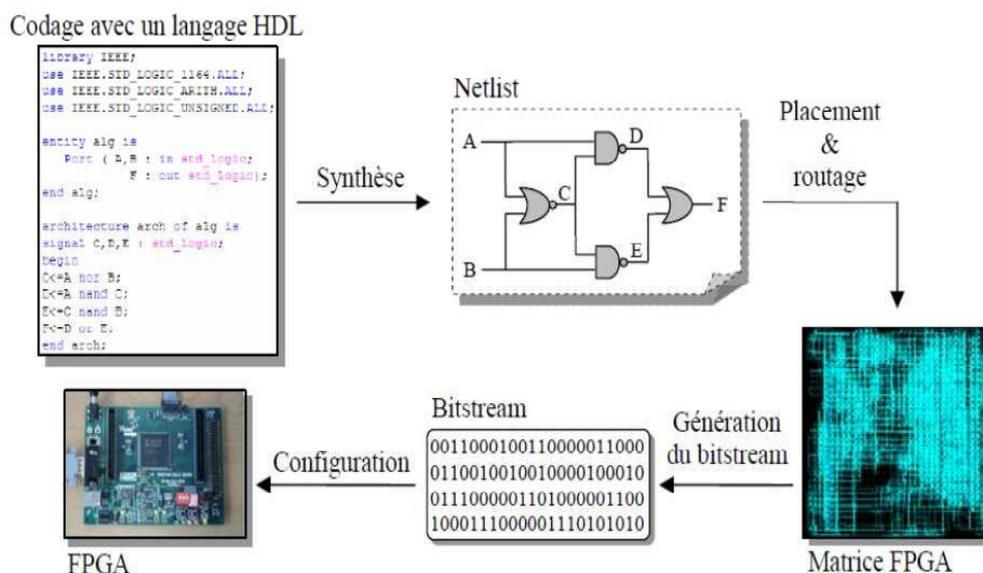


Figure 4.1. Programmation d'un FPGA.

4.4 Présentation des différents outils de travail

4.4.1 Présentation du logiciel ISE

L'environnement ISE (Integrated Software Environment) Xilinx est un logiciel de description, de simulation et de programmation de circuits et systèmes numériques sur des composants programmables [12].

Pour l'implémentation d'une application sur circuit FPGA, quatre étapes sont nécessaires :

- 1) la spécification regroupe les trois modes (schématique, diagrammes d'états ou HDL) de saisie d'un circuit électrique. La spécification HDL est synthétisée pour générer un fichier appelé NETLIST qui décrit les interconnexions entre les registres.
- 2) La vérification du design est une étape parallèle où le concepteur observe le comportement du code et vérifie s'il se comporte tel qu'il est prévu.
- 3) L'implémentation sur le composant en spécifiant les références exactes de celui-ci à savoir : le boîtier, la fréquence de travail et les options spécifiques à chaque composant.
- 4) Le débogage du système : après chargement des interconnexions sur le FPGA, des tests doivent être effectués pour vérifier le bon fonctionnement de l'implémentation.

La figure 4.2 représente l'interface Project Navigator de l'ISE 14.2 permettant l'accès à toutes les ressources d'un projet ainsi qu'aux outils de l'implémentation.

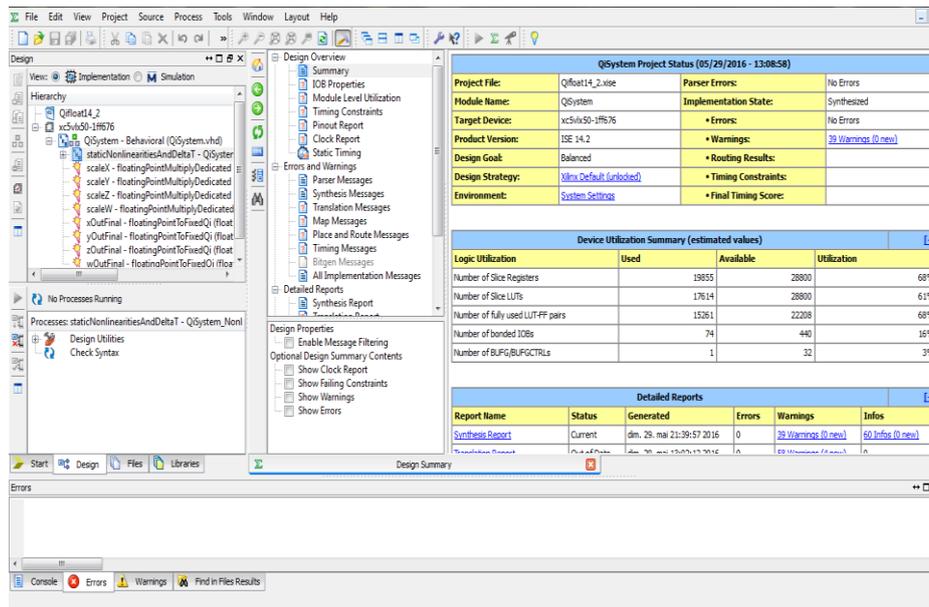


Figure 4.2. L'interface Project Navigator de l'ISE 14.2

4.4.2 Plate-forme de développement Virtex-5

La carte FPGA Virtex-5 a plusieurs caractéristiques et périphériques, on peut citer :

- Référence : Virtex-5 FPGA XC5VLX50-1FFG676.
- Puce de générateur d'horloge programmable du système.
- Horloge de fréquence 100 Mhz
- Commutateurs DIP à usage général, voyants et boutons poussoirs
- Codec audio stéréo AC97 avec line-in, line-out, 50mw casque, entrée microphone, et SPDIF prises audio numérique.
- Port série RS-232.
- Ecran LCD 2 lignes x 16 caractères.
- Connecteur vidéo DVI (VGA pris en charge avec l'adaptateur fourni).
- Connecteurs de souris et clavier PS/2.
- Contrôleur de configuration système ACE avec CompactFlash Type 1 CompactFlash connecteur.
- SRAM synchrone ZBT, 9Mb sur le bus de données 32 bits avec quatre bits de parité
- Intel P30 StatFlash linéaires puces de mémoire flash.
- Serial Peripheral interface (SPI) Flash (2Mo).
- Connecteur RJ45 Ethernet.
- Puce d'interface USB avec ports hôtes et périphériques.

Les figures 4.3 et 4.4 représentent la carte Virtex-5 utilisé dans le cadre de notre projet.

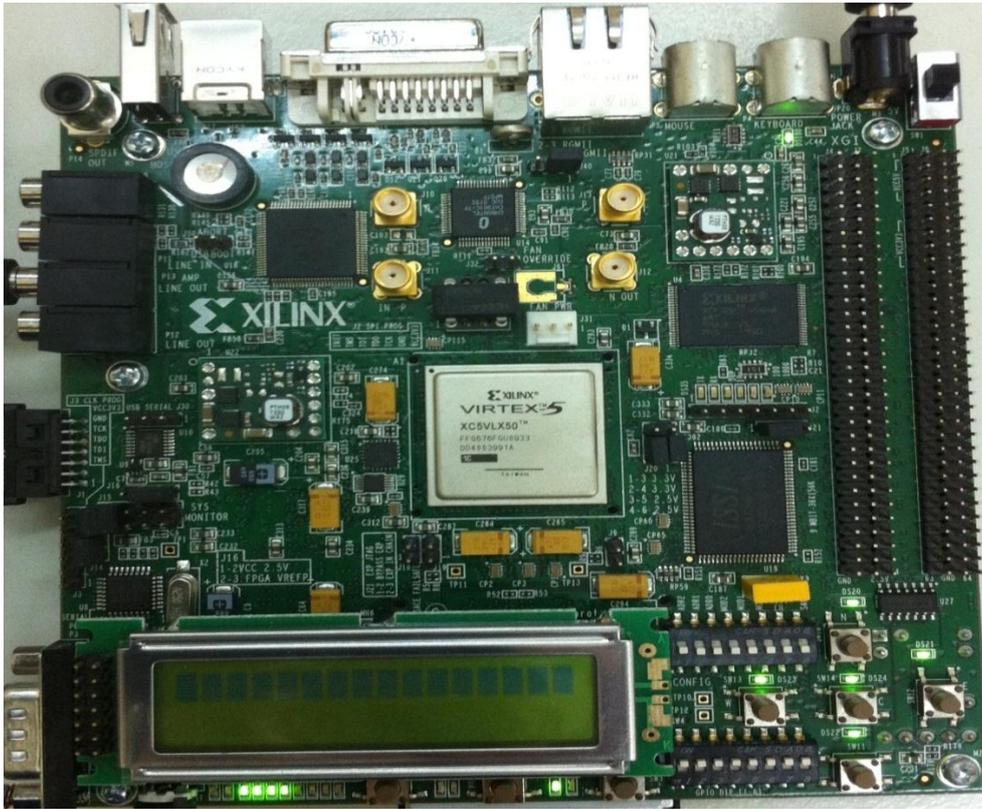


Figure 4.3. Plateforme de développement Virtex-5 (vu de dessus).



Figure 4.4. Plateforme de développement Virtex-5 (vu de dessous).

4.5 Réalisation expérimentale de l'implémentation

4.5.1 Programmation en VHDL

Pour pouvoir réaliser l'implémentation du système de Qi sur circuit FPGA Virtex-5 on utilise le logiciel ISE 14.2 pour générer le programme VHDL, on suit les étapes suivantes :

- on écrit le programme VHDL en utilisant les blocs IP (CORP Generator & Architecture Wizard) en particulier les blocs Math Functions (**Figure 4.5**) pour écrire les équations de système hyper chaotique de Qi, on a utilisé l'approximation d'Euler :

$$\frac{dx}{dt} = f'(x) = \frac{\Delta x}{\Delta t} = \frac{x_{n+1} - x_n}{\Delta t} \quad (4.1)$$

$$x_{n+1} = \Delta t * f'(x) + x_n \text{ Où } \Delta t \text{ représente le pas d'intégration.} \quad (4.2)$$

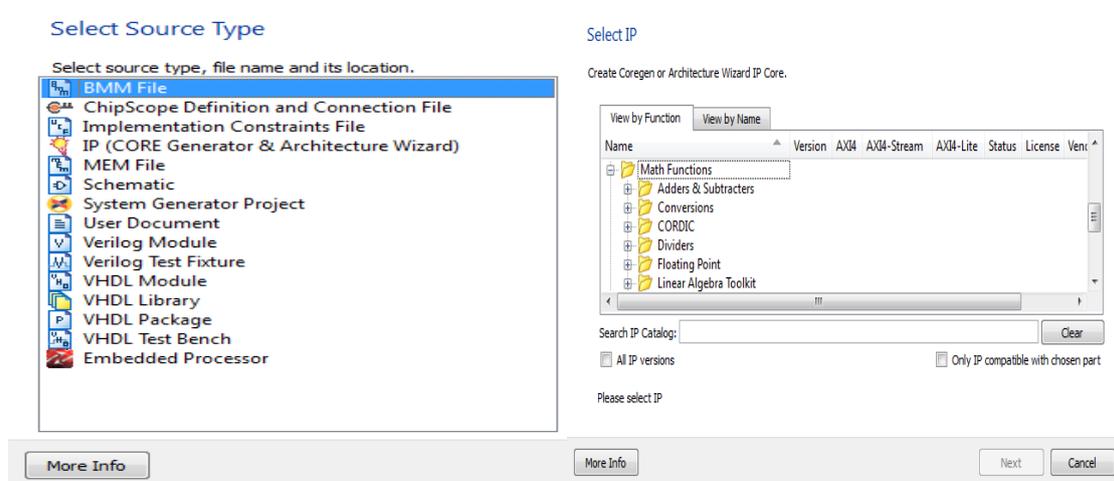


Figure 4.5 Interface IP (CORP Generator & Architecture Wizard).

La figure 4.6 représente un exemple d'un bloc d'addition.

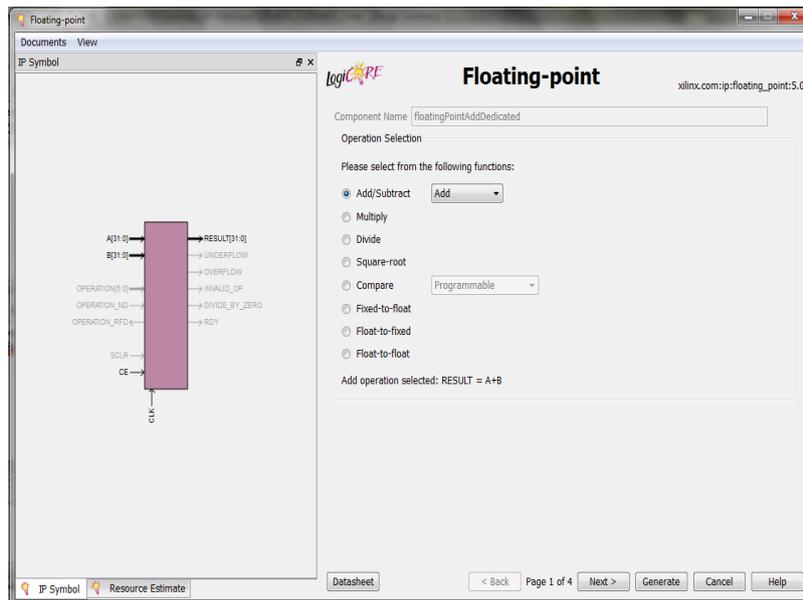


Figure 4.6 Exemple d'un bloc d'addition.

- Après avoir synthétisé notre projet on a utilisé le ModelSim (Figure 4.7) pour visualiser les signaux de système de Qi.



Figure 4.7 visualisation des signaux sous ModelSim.

- Ensuite, on ajoute le programme VHDL du codec AC97, après on effectue une opération de synthèse pour vérifier le bon fonctionnement du montage, et on génère le Bitstream cliquant sur Generate Programming File.

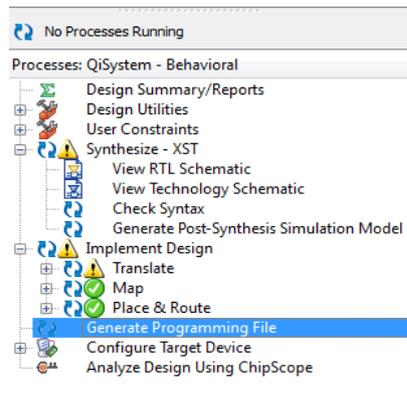


Figure 4.8 La fenêtre pour la synthèse.

- Pour implémenter notre projet dans la carte FPGA Virtex-5 nous allons connecter cette dernière par câble USB au PC. Ensuite on clique sur « configure Target Device », une fenêtre s’ouvre permettant d’implémenter le fichier Bitstream sur la carte FPGA Virtex-5 : on clique sur bouton droit « initialize chain » puis sur « program » (**Figure 4.9**).

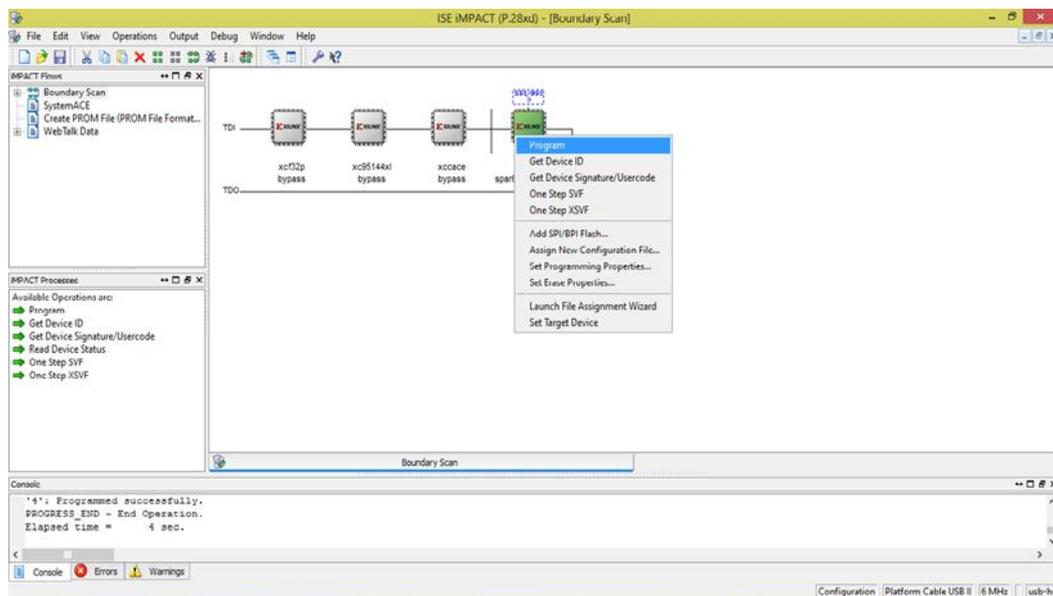


Figure 4.9 Interface permettant la programmation du FPGA.

• Le convertisseur

Le convertisseur utilisé dans notre projet est le convertisseur du codec AC97 de résolution 18 bits interne sur la carte FPGA Virtex-5. La conversion se fera comme suit : les données à convertir sont recueillies à partir de circuit FPGA. Les signaux de sorties du

système réalisé (x_1, x_2, x_3 et x_4, y_1, y_2, y_3 et y_4) de format 18 bits vont subir une conversion numérique analogique.

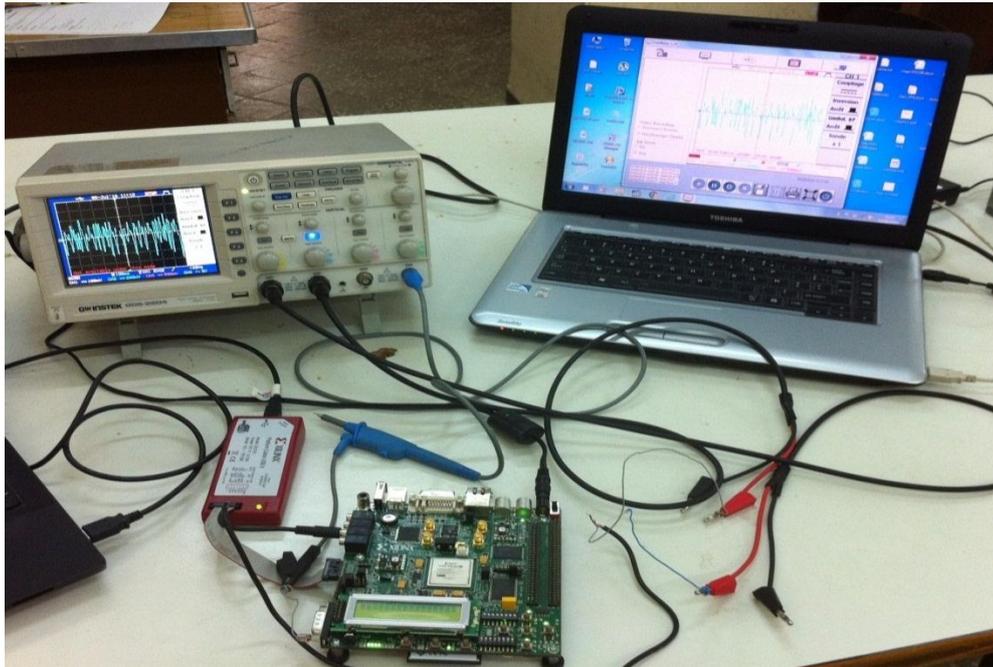


Figure 4.10. Implantation du système hyper chaotique sur la carte FPGA.

4.5.2 visualisation des signaux

Les figures 4.11 à 4.14 Représentent les différents signaux obtenus par la simulation et les signaux expérimentaux obtenus à l'aide de l'oscilloscope numérique.

Les figures 4.15 à 4.20 représentent les courbes issues de l'implémentation du système hyper chaotique de Qi.

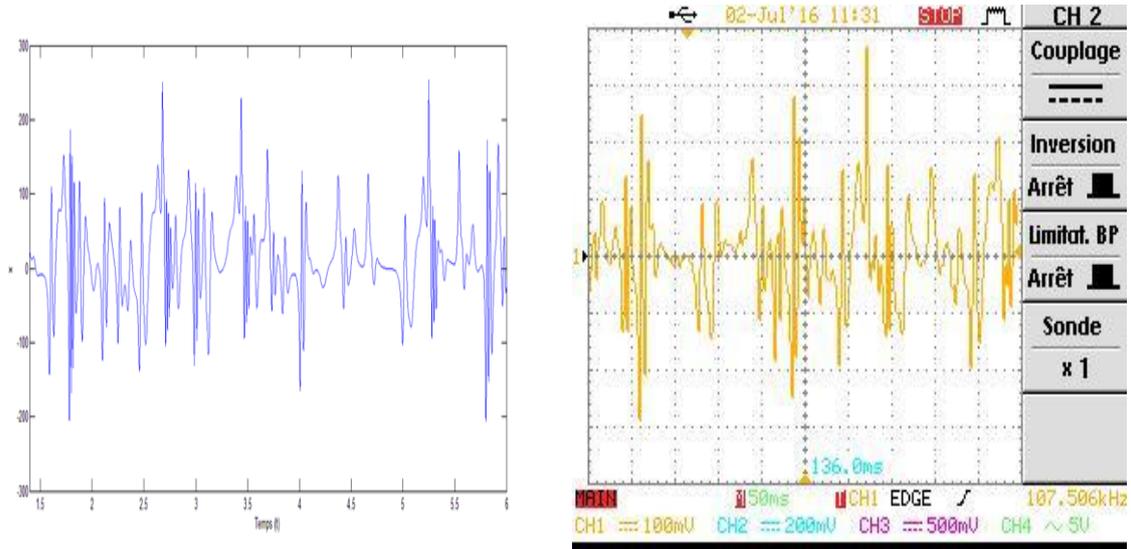


Figure 4.11. l'état de x en fonction du temps.

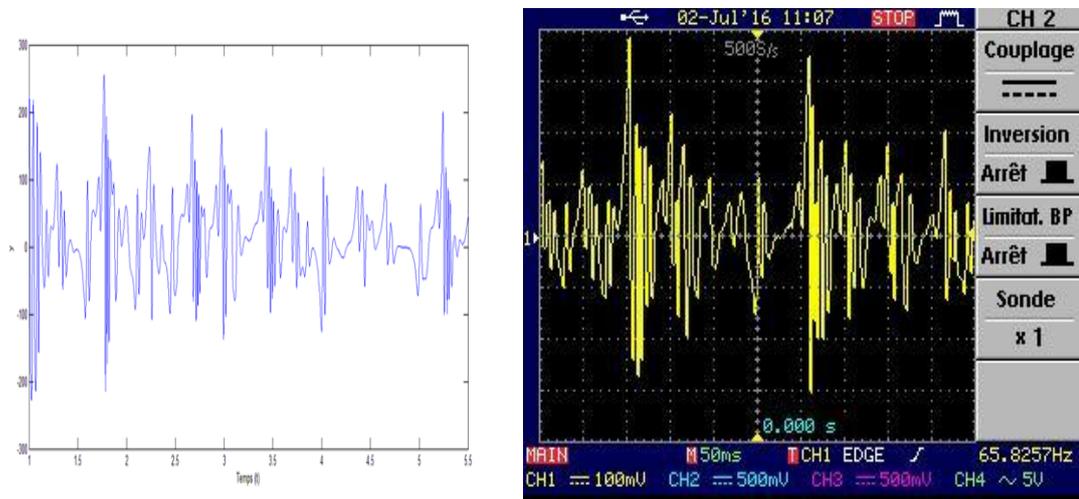


Figure 4.12. l'état de y en fonction du temps.

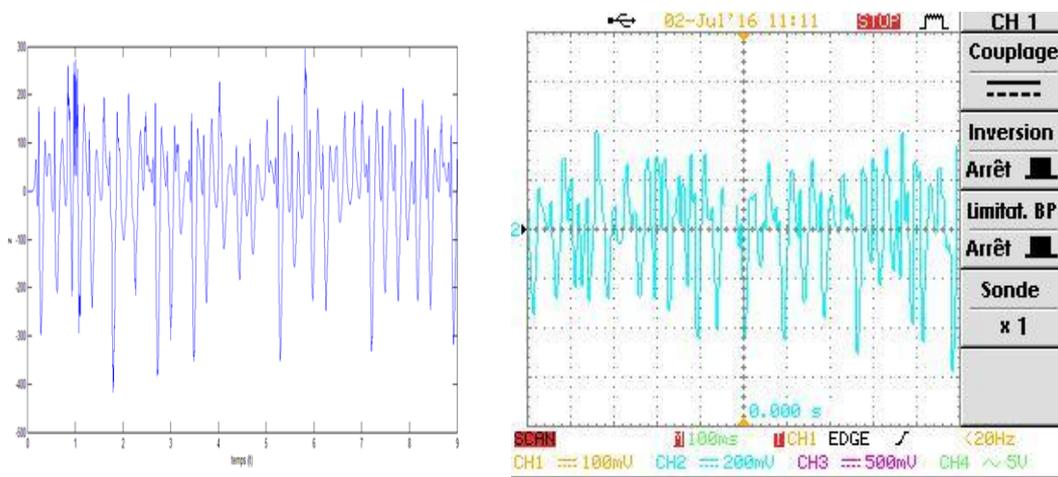


Figure 4.13. l'état de z en fonction du temps.

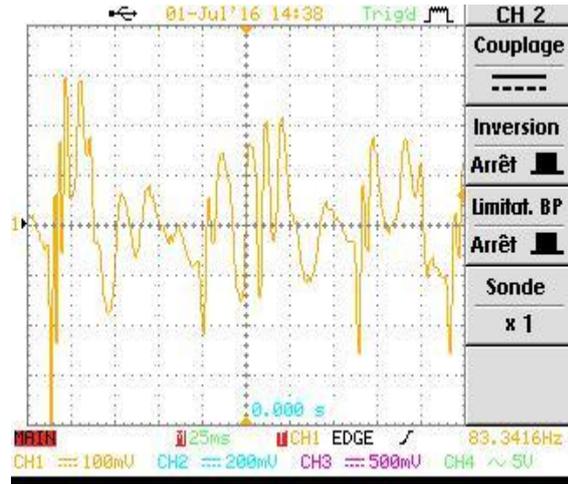
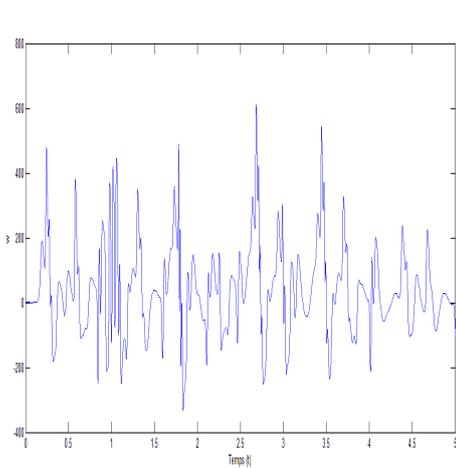


Figure 4.14. l'état de w en fonction du temps.

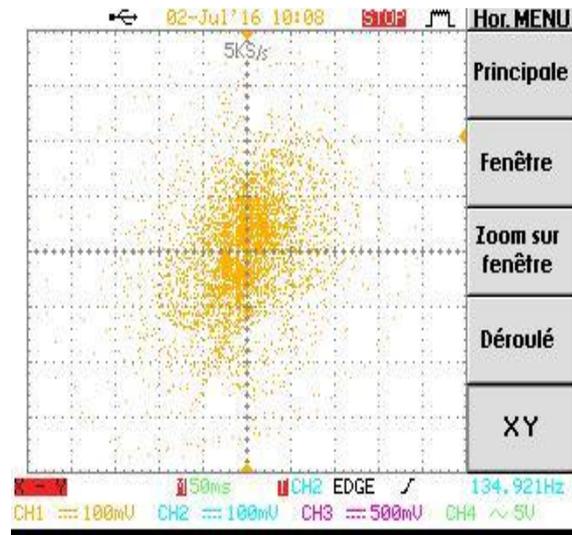
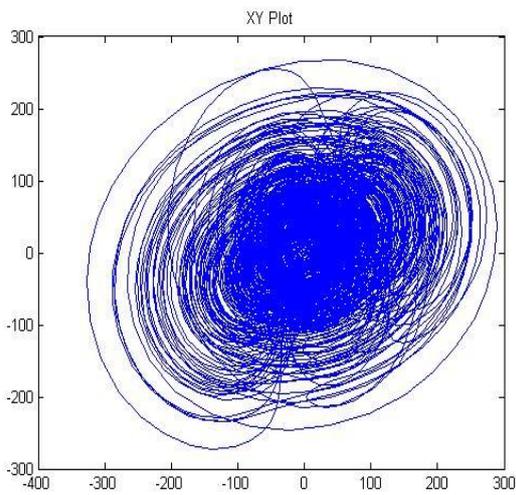


Figure 4.15. plan de phase y en fonction de x .

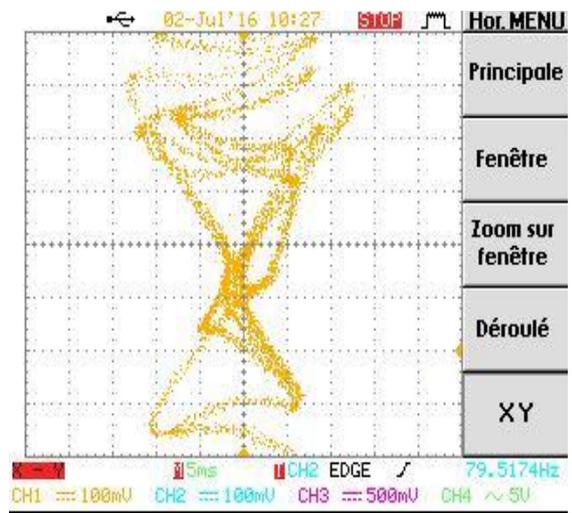
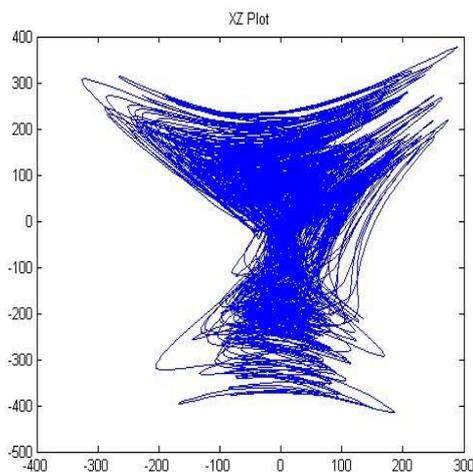


Figure 4.16. plan de phase z en fonction de x .

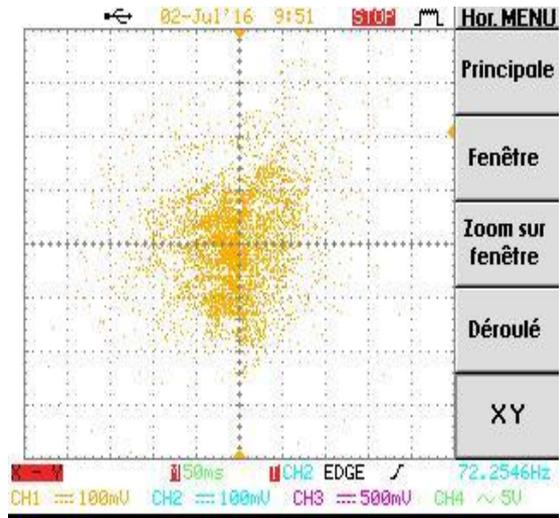
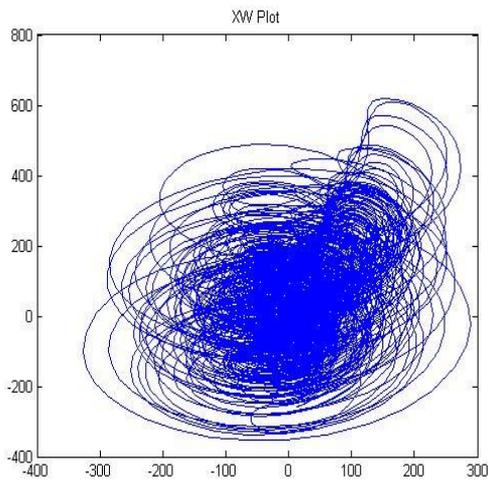


Figure 4.17. plan de phase w en fonction de x .

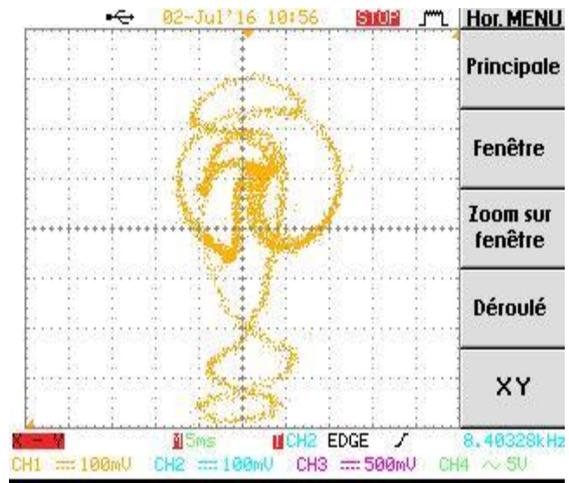
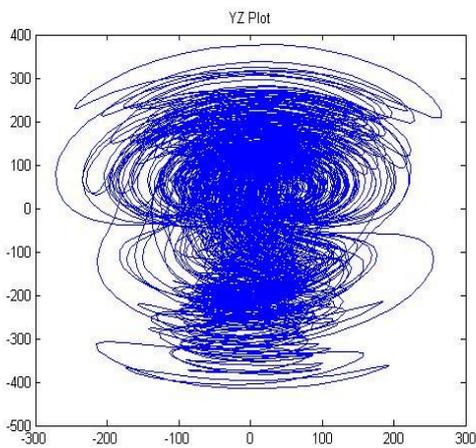


Figure 4.18. plan de phase z en fonction de y .

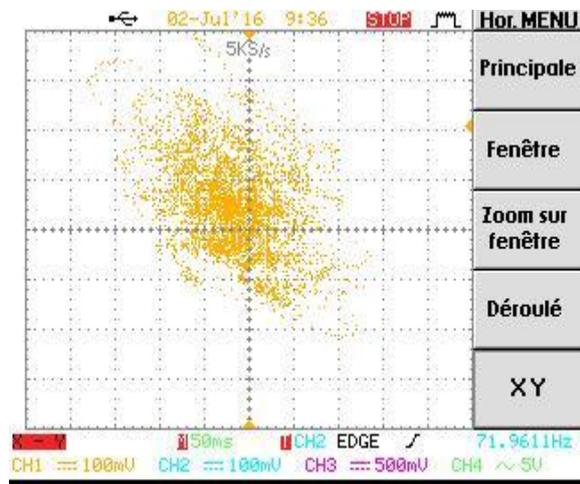
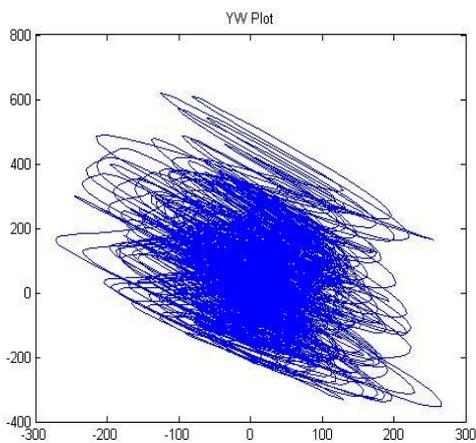


Figure 4.19. plan de phase w en fonction de y .

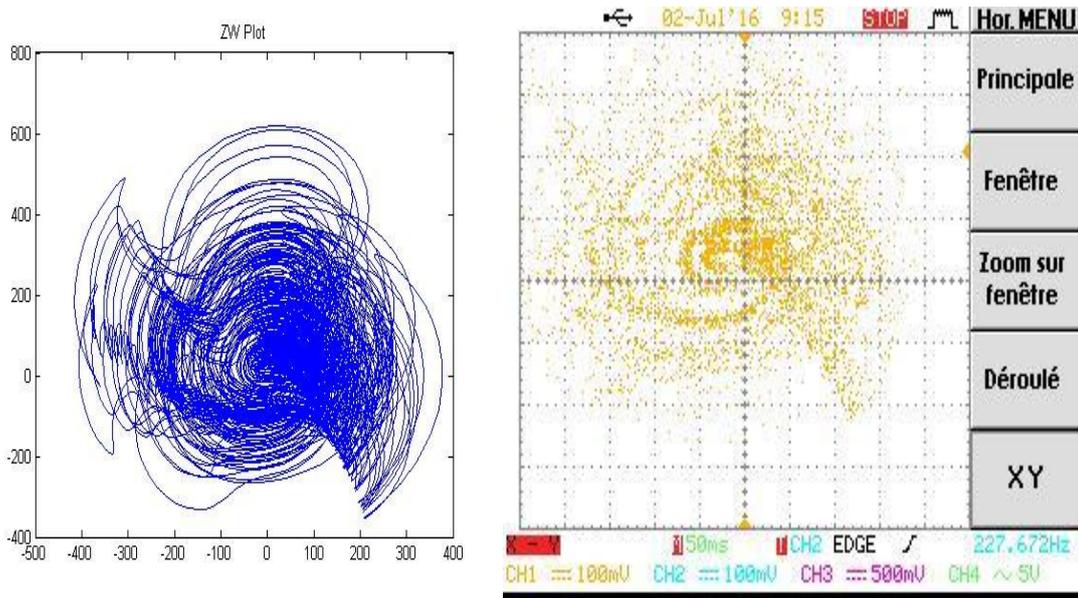


Figure 4.20. plan de phase w en fonction de z.

Les figures 4.10 à 4.19 présentent les résultats en temps réel et montrent le bon fonctionnement du processus de l'implémentation.

L'environnement ISE fournit un rapport donnant toutes les ressources utilisées lors de l'implémentation sur FPGA. La figure 4.20 représente un tableau contenant toutes les ressources utilisées liées au design, il les comptabilise en nombre et en pourcentages.

QISystem Project Status (06/08/2016 - 11:30:08)			
Project File:	Qifloat14_2.xise	Parser Errors:	No Errors
Module Name:	QISystem	Implementation State:	Placed and Routed
Target Device:	xc5vlx50-1ff676	• Errors:	No Errors
Product Version:	ISE 14.2	• Warnings:	97 Warnings (0 new)
Design Goal:	Balanced	• Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	All Constraints Met
Environment:	System Settings	• Final Timing Score:	0 (Timing Report)

Device Utilization Summary				
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	19,157	28,800	66%	
Number used as Flip Flops	19,157			
Number of Slice LUTs	16,660	28,800	57%	
Number used as logic	15,285	28,800	53%	
Number using O6 output only	9,210			
Number using O5 output only	2,262			
Number using O5 and O6	3,813			
Number used as Memory	1,284	7,680	16%	
Number used as Shift Register	1,284			
Number using O6 output only	1,284			

Figure 4.21. Ressources consommées par l'implémentation.

4.6 Conclusion

L'objectif de ce chapitre a été l'implémentation sur circuit FPGA de système hyper chaotique de Qi afin de chercher la concordance entre les signaux obtenus par simulation dans le chapitre 3 et les signaux relevés au niveau de l'oscillateur numérique.

Les résultats obtenus ont montré le bon fonctionnement du processus de l'implémentation.

Conclusion générale

Dans ce mémoire, nous avons étudié le système dynamique hyper chaotique de Qi, ainsi que la synchronisation maître-esclave de deux systèmes de Qi.

Le premier chapitre présente les notions de base sur les systèmes dynamiques chaotiques. Dans le deuxième chapitre, nous avons étudié le comportement du système hyper chaotique de Qi, cette étude a été mise en évidence à l'aide de simulations. Le chapitre 3 a été consacré à l'étude du phénomène de la synchronisation entre deux systèmes de Qi, on s'est basée sur trois lois de contrôle. Dans le dernier chapitre, une implémentation de système hyper chaotique de Qi a été réalisée sur circuit FPGA.

La contribution de ce travail a été la synchronisation de deux systèmes hyper chaotique identiques grâce à une approche par contrôle actif. Des lois de commande sont calculées pour obtenir la synchronisation. Une première loi de commande, basée sur la linéarisation de l'erreur de synchronisation est développés. Ensuite, une seconde loi est calculée à l'aide de la théorie de stabilité de Lyapunov, pour tenir compte de la différence entre les paramètres de l'émetteur et ceux du récepteur. Une troisième loi de commande est développée, elle permet l'identification des paramètres de l'émetteur grâce à une approche de commande adaptative. Nous avons montré par simulation que dans cette approche, les états de système maître et ceux de système esclave sont asymptotiquement synchronisés avec la détermination des paramètres inconnus.