الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Projet de Fin d'Études

présenté par

LAAZAB KHEIRA

&

BENTOUTA ASMA

pour l'obtention du diplôme Master en Électronique Spécialité Traitement d'Information et Système

Thème

Analyse de différents systèmes chaotiques .synthèse d'observateur

Proposé par : Mr. FERDJOUNI Abdelaziz

Année Universitaire 2013-2014

Remerciements

Nous remercions ALLAH de nous avoir donné la volonté et le courage

qui nous ont permis de réaliser ce travail, veuille t'il nous guider dans

le droit chemin.

Nos chaleureux remerciements vont tout d'abord à nos familles, qui

nous ont soutenu par leur Amour, leurs encouragements et leur

intérêt tout au long de nos études.

Nous exprimons nos sincères remerciements à notre promoteur Mr A.FERDJOUNI pour avoir accepté d'encadrer ce travail et pour son aide et son précieux encouragement pour la mise en œuvre de ce modeste travail.

Un grand merci à Mr CHIKHI, Mr NEDJMI et Mr OUSSAMA

ainsi qu'à tout le personnel de l'administration

Notre infinie gratitude va à tous nos enseignants qui ont contribué à notre travail et à tous les enseignants de département d'électronique et en précisant les jurys

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

Dédicace

Je dédie ce modeste travail

À la meilleure maman du monde, artisane de ma réussite aucune dédicace ne saurait exprimer ma reconnaissance, mon grand attachement et mon profond amour. Tu as toujours tout fait pour me préserver une part de bonheur et de réconfort. Que dieu puisse m'aider à te prouver ma sincère gratitude pour ces nombreuses années de sacrifice. Je n'oserais prétendre, que ce travail soit à la hauteur des longues nuits de veille et des moments d'angoisse que tu as endure pendant toutes mes années d'étude.

À mon très cher père, à qui je dois ce que je suis. Tu as toujours été le meilleur papa et tu m'as appris à aider les autres. Aucun mot ne saurait témoigner de l'étendue des sentiments que j'éprouve toujours à ton égard.

A mes frères et mes sœurs qu'Allah les protège.

A mes chères amies : Mohamed, affaf, leila, sans oublier ma binôme: Asma avec lequel j'ai partagée des moments de joie et de peur ainsi que des moments parfois agréables et parfois difficiles

A tous mes amis.

Kheira.

Dédicace

A mon chère père ***Abd Errahmane,*** avec toute ma reconnaissance,, lui qui m'a tant aidé et soutenu dans mes études.

* A la lumière de ma vie, ma très chère mère, ***Abesse Souhila*** pour son amour et ses sacrifices sans limites .

*A mes très chères grandes mères ***Hanifa*** et ***Zhour.***

*A mes frères **Mohamed** , **Zin Eddine** et son épouse **Nassima**.

*A mes sœurs **Nor Elhouda** et **Narimane** sur tout la petite **Rimasse** et ***toute ma famille*** particulièrement **Warima, Amina,Ichrak,Djazia,Razika Lamia,Amel, Djamila ,Wahiba,Saida ,Nessrine,Nessrine,Rihab Nihad,Rania,Souad,Hadil,Khiro,Touhami ,Wassim,Farok,Ahmed,*Dr Allaa*** pour leur amour indéfectible et leur soutien moral.

* son oublier mon très chèr amie **ALI AISSA**

* A tous mes amis particulièrement ***Khaoula, Asma, Yasmine, Ayàt,Faiza,Amira,Ouidad,Fella,Abdo,Hicham,Ahcen, Abd Elkarime ,Raouf,Toufik,*** *et* A Ma chère binôme ***Kheira***

A tous ce qui m'ont aidé et soutenu pendant tout mon cursus universitaire.

Asma

ملخص:

في هذا العمل، نظام نقل آمن للمعلومات على أساس التزامن بين نظامين فوضويان .ونحن نركز على التزامن باستخدام مراقب بوضع الانزلاق يعمل خطوة خطوة، ثم يتم توليد إشارات الفوضى (لورنز , لو ,روسلر) وتحليل الانظمة، ثم حلهم باستعمال طريقة عددية رونج-كوتا 4. بعد ذلك ينم إجراء المحاكاة باستعمال سميلينك وكذلك باستعمال برنامج ماتلاب، وهذا لعرض نتائج الاشارات ومقارنتها. وفي الأخير التنفيذ الرقمي لهذه الانظمة على بطاقة DSP TMS320F2812 لمشاهدة الإشارات في الوقت الحقيقي.

كلمات المفاتيح : نظم الفوضى , الفوضى, طريقة رونج كوتا , بطاقة DSP ,التزامن , ملاحظة من انظمة غير خطية, المراقب .

Résumé :

Dans ce travail, un système de transmission sécurisé d'information basé sur la synchronisation de deux systèmes chaotiques .nous nous intéressons a la synchronisation à l'aide d'un observateur a mode glissant fonction étape par étape, ensuite générer un signal chaotique (Lorenz, lu et Rossler). D'abord ces derniers sont analysés, ensuite, sa résolution est effectuée à l'aide de la méthode numérique de Runge-Kutta d'ordre 4. La simulation des systèmes (Rössler, lorenz et lu) est effectuée à l'aide de Simulink, ensuite à l'aide d'un programme Matlab. Enfin, une implémentation digitale de ces systèmes est réalisée sur la carte DSP TMS320F2812 pour voir les signaux eu temps réel.

Mots clés : systèmes chaotiques, chaos, méthode de ronge kutta, la carte DSP, synchronisation, observabilité des systèmes non linéaires, observateur.

Abstract:

In this work, a secure transmission system of information which is based on the synchronization of the tow chaotic system. We are interested to the synchronization with a gliding mode, functioned stage by stage, then generate a chaotic signal (Rossler, lù, Lorenz) at first, this last is analyzed, next his resolution is performed with the help of the numerical method of Range Kutta of filth. The Simulink of towards with the help of Matlab program. Finally a digital Implementation of these systems is realized on DSPTMS320F2812 card (mal) to see the signals in actual time.

Keywords: Chaotic systems, chaos, method of Range kutta, the DSP card, synchronization, observability of nonlinear systems, observer.

Listes des acronymes et abréviations

FCO : Forme canonique d'observabilité.

Rk4: Runge Kutta d'ordre 4.

DSP: Digital Signal Processor.

FPGA: Field Programmable Gate Array.

CCS: Code Composer Studio.

EVA: Event Manager A.

EVB: Event Manager B.

JTAG: Joint Test Action Group.

SPI: Serial Peripheral Interface.

SCI: Serial Communication Interface.

CAN: Controller Area Network.

GPxMUX: general purpose multiplexer.

GPxDIR: general purpose direction.

PWM: pulse with modulation.

CMPR: register de comparison.

GPIO: general purpose input-output.

INT: les interruptions.

UAL : Unité Arithmétique et Logique.

GPTCONA: General Purpose Timer Control register A.

TxCNT: Timer x Counter register.

TxCMPR: Timer x Compare register buffer.

TxPR: Timer x period register buffer.

Table des matières

Chapitre 1

Systèmes dynamiques et chaos

Chapitre 2

Synchronisation des Systèmes chaotiques

Liste des figures

Introduction générale

L'analyse des systèmes dynamiques en général, et des systèmes non linéaire en particulier, a été menée par beaucoup de chercheurs et scientifiques de différentes disciplinés. Les systèmes dynamiques non linéaires offrent la particularité. Dans certaines situations, d'avoir un comportement imprévisible ; dépendant de la variation de paramètres et des conditions initiales. Ce comportement n'est pas aléatoire, mais il imprévisible à long terme : c'est le comportement chaotique. L'analyse de tel système permet de déterminer dans quelles conditions le fonctionnement en mode chaotique se produit. Les applications des systèmes chaotiques sont multiples, l'une d'elle est la

Plusieurs types de synchronisation (synchronisation complète ou identique, généralisée, de la phase, projective) ont été étudiés, et beaucoup de méthodes ont été proposées, mais tous ces types et méthodes sont englobés sous 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 : Son principe est de choisir un système générateur de chaos appelé "émetteur". Celui-ci est décrit par des équations récurrentes et caractérisé par ses variables d'état constituant le vecteur d'état. Quelques composantes de ce vecteur sont transmises à un second système dénommé "récepteur". Dans ce travail on va s'intéresser au deuxième mode Méthode de couplage unidirectionnel choisie : « l'approche par observateur ».

La génération de signaux chaotique peut être réalisée grâce à des circuits électroniques analogiques, formés principalement par des amplificateurs

opérationnels et des multiplieurs. 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 (Runge-Kutta, Euler, , etc...), et ensuite implémenter le programme sur une carte DSP ou FPGA

L'objet de notre mémoire est d'étudier la synchronisation unidirectionnel à l'aide d'observateur des systèmes chaotique (Rössler et Lorenz et Lu). D'analyser l'observabilité de système, faire la synthèse d'observateur en mode glissant étape par étape et finalement l'implémentation des systèmes de (Rössler et Lorenz et Lu) sur la carte DSP TMS320F2812.

Ce mémoire est organisé en quatre chapitres, comme suit :

Chapitre 1 : « **Systèmes dynamiques chaotique et chaos** »

Dans ce premier chapitre, nous avons donnée des définitions sur le système dynamique non linéaire, les systèmes chaotique et les méthodes qui permettant d'analyse ces dernier.

Chapitre 2 : « synchronisation des systèmes chaotiques »

Consacre aux différentes méthodes de synchronisation chaotique et base sur la synthèse du récepteur chaotique utilisant un observateur a mode glissant pour la reconstruisons des états de système chaotique (rossler et lorenz et lu).

Chapitre 3 : « Implémentation du système chaotique sur la carte DSP »

Les caractéristiques de la carte DSP TMS320F2812 sont données dans ce chapitre. Sont donnés également quelques généralités sur le logiciel de la carte DSP, qui est le code composer studio. A la fin, la partie d'implémentation de la méthode RK4 appliquée au système de (Rössler, Lorenz et L u) sur la carte DSP est présentée.

Chapitre 4 : « Résultats et commentaires »

Nous présentons et appliquons la méthode de Runge Kutta d'ordre 4 à l'exemple des systèmes de Rössler, Lorenz et Lu. Et nous donnons des résultats et des commentaires de la simulation du système de (Rössler Lorenz et Lu) sur logiciel Matlab et langage C,

ainsi que les résultats obtenus par l'implémentation, afin de comparer ces résultats avec ceux de la simulation sous Matlab.

Chapitre 1 Systèmes dynamiques et chaos

1.1 Introduction :

Les systèmes dynamiques désignent couramment la branche de recherche active des mathématiques, à la frontière de la topologie, de l'analyse, de la géométrie, de la théorie de la mesure et des probabilités, et qui s'efforce d'étudier les propriétés d'un système dynamique. La nature de cette étude diffère suivant le système dynamique étudié, nature qui dépend des outils utilisés (analytiques, géométriques ou probabilistes).

Historiquement, les premières questions relevant des systèmes dynamiques concernaient la mécanique à une époque où elle était incluse dans l'enseignement des mathématiques. Une des questions majeures et qui a motivé la recherche mathématique est le problème de la stabilité du système solaire. Les travaux de Lagrange sur le sujet consistèrent à interpréter l'influence des corps autres que le Soleil sur une planète comme une succession de chocs infinitésimaux : ces travaux retrouvent des échos dans le théorème KAM (Kolmogorov - Arnold - Moser) [1].

En 1963 le météorologue Edward Lorenz expérimentait une méthode lui permettant de prévoir les phénomènes météorologiques. C'est par pur hasard qu'il observa qu'une modification minime des données initiales pouvait changer de manière considérable ses résultats. Lorenz venait de découvrir le phénomène de sensibilité aux conditions initiales.

Les systèmes répondant à cette propriété seront à partir de 1975 dénommés : systèmes chaotiques. C'est donc au cours des années soixante-dix que la théorie du chaos a pris son essor.

Evidemment, les travaux des prédécesseurs de Lorenz ont donc été très importants pour la compréhension du chaos déterministe, mais il faut souligner que ce qui va permettre aux scientifiques une compréhension plus accrue des systèmes chaotiques c'est l'ordinateur.

En effet, les équations différentielles régissant un système chaotique sont nécessairement non linéaires et, sans ordinateur, leur résolution est en général impossible.

1.2 Système dynamique

· **Dynamique** : on appelle système dynamique, tout système dont l'étude ne peut se faire qu'en tenant compte des valeurs passées et présentes du phénomène (équations différentielles).

. **Système dynamique** est un système classique qui évolue au cours du temps de façon à la fois :

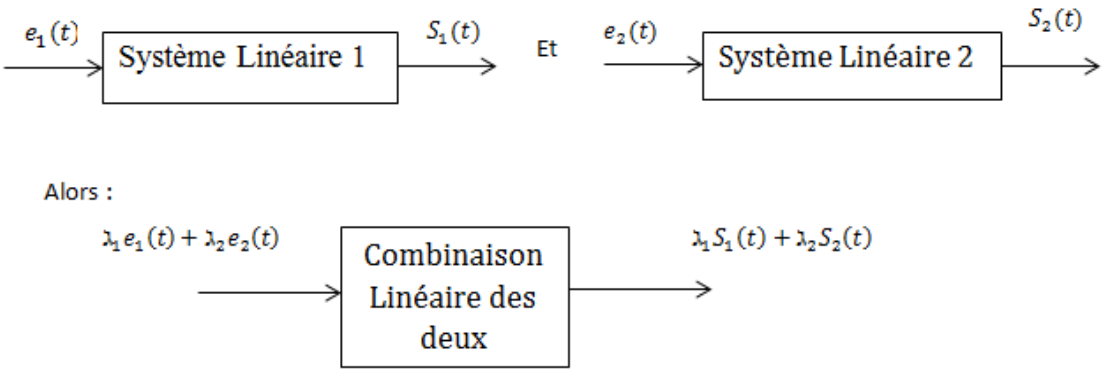
- causale, c'est-à-dire que son avenir ne dépend que de phénomènes du passé ou du présent.
- déterministe, c'est-à-dire qu'à une « condition initiale » donnée à l'instant « présent » va correspondre à chaque instant ultérieur un et un seul état « futur » possible.

On exclut donc ici conventionnellement les systèmes « bruités » intrinsèquement stochastiques, qui relèvent de la théorie des probabilités.

L'évolution déterministe du système dynamique peut alors se modéliser de façon d'une évolution continue dans le temps, représentée par une équation différentielle ordinaire.

- **La proportionnalité et l'additivité**

Si les deux systèmes dynamiques 1 et 2 vérifient :



Figure(1.1) :combinaison linéaire des deux système

Si le système n'est pas linéaire, il ne sera pas possible de l'étudier normalement [2].

1.2.1 Systèmes dynamiques particuliers

Les systèmes dynamiques sont classés selon :

- les systèmes dynamiques à temps discret.
- les systèmes dynamiques à temps continu.

1.2.1.1 Système dynamique à temps discret

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

$$x_{k+1} = f(x_k, v) \tag{1.1}$$

Où $f : \mathbb{R}^n \longmapsto \mathbb{R}^n$ est une fonction au moins continue ou continue par morceaux qui définit la dynamique du système discret. De la même manière si nous associons à cette dynamique un état initial x_0 nous pourrons avoir une solution unique de f [2].

v : Vecteur de paramètre.

1.2.1.2 Système dynamique à temps continu

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

$$\frac{dx}{dt} \equiv \dot{x} = f(x_t, t, v) \tag{1.2}$$

Où f est de classe $C^1 : \mathbb{R}^n \longmapsto \mathbb{R}^n$ définit la dynamique du système continu. Nous pouvons associer une solution unique du système définie à l'aide de l'équation (1.5). L'évolution des ensembles d'états successifs du système à chaque instant t , appelle la trajectoire [2].

1..3 Systèmes autonomes et non autonomes

Soit le système dynamique suivant :

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

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

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

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_{n+1}.

1.4 Points fixes

La détermination des points fixes se fait comme pour un système à une variable. Les points fixes sont définis comme x est un point fixe de R^n

$$\begin{pmatrix}\dot{x} = f(x,y)\\ \dot{y} = g(x,y)\end{pmatrix} \longleftrightarrow \begin{pmatrix}\dot{x} = 0\\ \dot{y} = 0\end{pmatrix} \qquad (1.4)$$

1.4.1 Stabilité des points fixes

Concerne un système différentiel ordinaire ou une itération.

Le système$\frac{dx}{dt} = u(x,y), \frac{dy}{dt} = v(x,y)$ (1.5)

Possède le point fixe (ou point singulier ou position d'équilibre) (x_0, y_0)

Si celui-ci est une solution du système, c'est-à-dire si: $u(x_0,y_0) = 0, v(x_0,y_0) = 0$ (1.6)

Pour une itération $x_{n+1} = f(x_n,y_n)$, $y_{n+1} = g(x_n,y_n)$, il s'agira d'une solution (x_0, y_0) telle que:$f(x_0,y_0) = x_n, g(x_0,y_0) = y_n$ pour tout $t >= 0$.

la matrice Jacobienne J.

$$J = \begin{pmatrix}\frac{\partial u}{\partial x}(x,y) & \frac{\partial u}{\partial y}(x,y)\\ \frac{\partial v}{\partial x}(x,y) & \frac{\partial v}{\partial y}(x,y)\end{pmatrix} \qquad (1.7)$$

$\begin{pmatrix}u_1\\ v_1\end{pmatrix}$ et $\begin{pmatrix}u_2\\ v_2\end{pmatrix}$ $(u_1,u_2),(v_1,v_2)$ Sont les vecteurs propres.

Les deux vecteurs propres de la matrice Jacobienne. Par définition

$$j\begin{pmatrix}u_1\\v_1\end{pmatrix}=\lambda_1\begin{pmatrix}u_1\\v_1\end{pmatrix} \quad \text{et} \quad j\begin{pmatrix}u_2\\v_2\end{pmatrix}=\lambda_2\begin{pmatrix}u_2\\v_2\end{pmatrix}$$

Or, si les vecteurs propres sont distincts, ils forment une base complète et on peut alors exprimer la solution comme leur combinaison linéaire

$$\begin{pmatrix}u(t)\\v(t)\end{pmatrix}=\alpha(t)\lambda_1\begin{pmatrix}u_1\\v_1\end{pmatrix}+\beta(t)\lambda_2\begin{pmatrix}u_2\\v_2\end{pmatrix} \qquad (1.8)$$

En insérant l'équation (1.9) dans l'équation (1.8), nous obtenons

$$\dot{\alpha}(t)=\lambda_1\alpha(t) \qquad \text{et} \qquad \dot{\beta}(t)=\lambda_2\beta(t) \qquad (1.9)$$

Ce qui donne enfin

$$\begin{pmatrix}\boldsymbol{u(t)}\\\boldsymbol{v(t)}\end{pmatrix}=\boldsymbol{\alpha_0}\ell^{\lambda_1 t}\begin{pmatrix}\boldsymbol{U_1}\\\boldsymbol{V_1}\end{pmatrix}+\boldsymbol{\beta_0}\ell^{\lambda_2 t}\begin{pmatrix}\boldsymbol{U_2}\\\boldsymbol{V_2}\end{pmatrix} \qquad (1.10)$$

Où α_0 et β_0 sont deux constantes d'intégration. Les valeurs propres nous renseignent donc sur le type de stabilité, alors que les vecteurs propres indiquent l'orientation des orbites au voisinage du point fixe. A des valeurs propres réelles positives (respectivement négatives) correspondent des solutions qui s'écartent (respectivement se rapprochent) exponentiellement vite du point fixe.

Remarque : $\boldsymbol{\lambda_1}$ ***et*** $\boldsymbol{\lambda_2}$ peuvent être complexes conjuguées.

Puisque les valeurs propres peuvent être complexes conjuguées, on distingue les trois cas suivants :

- Si l'une des valeurs propres est à partie réelle positive : la trajectoire s'éloigne exponentiellement du point d'équilibre.
- Les deux valeurs propres sont à partie réelle négative : la trajectoire converge vers le point de repos.
- Si l'une (ou les deux) des valeurs propres est nulle : une des composantes de l'erreur δx (ou les deux) reste constante.

1.5 Fonction de Lyapunov

Soit $\bar{x}$ un point fixe du système. Soit V : $W \longmapsto \mathbb{R}$ une fonction différentiable définie sur un voisinage W de $\bar{x}$ telle que $\mathrm{V}(\bar{x}) = 0$ et $\mathrm{V}(x) > 0$ si $x \neq \bar{x}$. Posons [2] :

$$\dot{\mathrm{V}} = \sum_{\mathrm{j}=1}^{\mathrm{n}} \frac{\partial \mathrm{V}}{\partial x_{\mathrm{j}}} \dot{x}_{\mathrm{j}} = \sum_{\mathrm{j}=1}^{\mathrm{n}} \frac{\partial \mathrm{V}}{\partial x_{\mathrm{j}}} f_i(x) \qquad (1.11)$$

Théorème de Lyapunov :

- Si $\dot{\mathrm{V}} \leq 0$ dans W$-\{\bar{x}\}$ alors $\bar{x}$ est stable.
- Si $\dot{\mathrm{V}} < 0$ dans W$-\{\bar{x}\}$ alors $\bar{x}$ est asymptotiquement stable.
- Si $\dot{\mathrm{V}} > 0$ dans W$-\{\bar{x}\}$ alors $\bar{x}$ est instable.

Remarque

Il n'y a pas de règle générale pour trouver une fonction de Lyapunov. Dans des problèmes de mécanique, l'énergie est souvent un bon candidat.

1.6 Système chaotique

Le terme « chaos » définit un état particulier d'un système dont le comportement ne se répète jamais qui est très sensible aux conditions initiales et imprédictible à long terme. Des chercheurs d'horizon divers ont alors commencé :

Ainsi, nous nous intéresserons principalement dans ce chapitre aux systèmes dynamiques chaotiques en nous attardant les espaces de phases, les attracteurs étranges et chaotiques et la bifurcation, les exposants de Lyapunov, lesquels nous permettront de mieux comprendre la nature du chaos [4].

1. 6.1 Définitions du chaos

Comme pour beaucoup de limites en science, il n'y a aucune définition standard du chaos.

Néanmoins, les dispositifs typiques du chaos incluent :

- ***La non-linéarité*** : Si le système est linéaire, il ne peut pas être chaotique.
- **Le déterminisme** : Un système chaotique a des règles fondamentales déterministes.

- **La sensibilité aux conditions initiales** : de très petits changements sur l'état initial peuvent mener à un comportement radicalement différent dans son état final.
- **L'imprévisibilité** : En raison de la sensibilité aux conditions initiales, qui peuvent être connues seulement à un degré fini de précision.
- **L'irrégularité** : Ordre caché comprenant un nombre infini de modèles périodiques instables (ou mouvements). Cet ordre caché forme l'infrastructure des systèmes chaotiques.

1.6.2 Historique de chaos

- En1963: Edward Lorenz découvre le premier système chaotique dans la météo ou encore appelé attracteur étrange.
- En1975 : Tien-Yien Li et James A. Yorke ont présenté pour la première fois le terme "chaos" dans un article intitulé "Period three implies chaos".
- En1978 : Mitchell Feigenbaum introduit un nombre universel associé au chaos.
- En 1990 Edward Ott, Celso Grebogi et James A. Yorke. Introduisent la notion de contrôle du chaos.
- En 1990 : Lou Pecora introduit la synchronisation des systèmes chaotiques [7].

1.6.3 Domaine d'application de chaos

- **Engineering** : Contrôle de vibration, stabilisation des circuits, réactions chimiques, turbines, étages de puissance, lasers, combustion, et beaucoup plus.
- **Ordinateurs :** Commutation des paquets dans des réseaux informatiques. Cryptage. Contrôle du chaos dans les systèmes robotiques.
- **Communications** : Compression et stockage d'image. Conception et management des réseaux d'ordinateurs.
- **Médecine et biologie** : Cardiologie, analyse du rythme du cœur, prédiction et contrôle d'activité irrégulière du cœur.
- **Management et finance** *:* Prévisions économiques, analyse financière, et prévision

1.6.4 Les caractéristiques globales de chaos

1. 6.4.1 Sensibilité aux conditions initiales

Poincaré fut l'un des premiers à entrevoir la théorie du chaos. Il remit en cause la façon d'interpréter le hasard et définit, pour la première fois, la sensibilité critique aux conditions initiales.

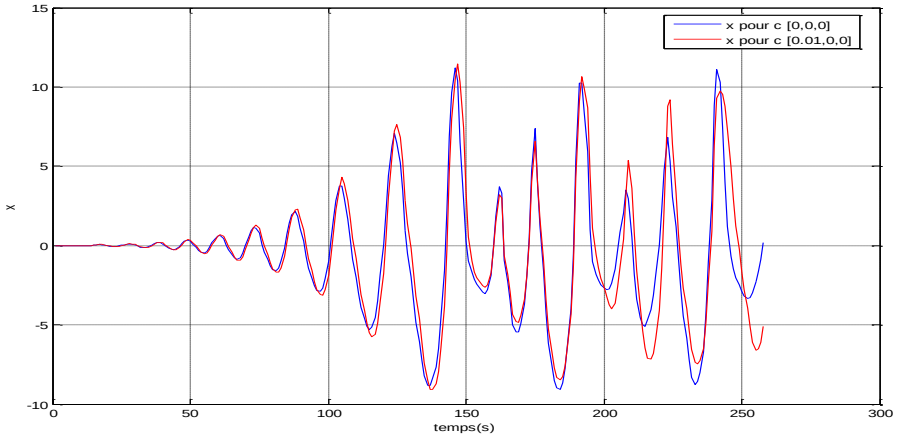


Figure (1.2) : Evolution dans le temps pour deux conditions initiales très proches d'un système chaotique de Lorenz.

1.7 Espace des phases, variables d'état et portrait de phases

De manière simplifiée, l'espace des phases permet de traduire des séries de nombre en une représentation spatiale, de dégager l'essentiel de l'information d'un système en mouvement et de dresser la carte de toutes ses possibilités. L'espace des phases est un espace mathématique souvent multi dimensionnel. Chaque axe de coordonnées de cet espace correspond une variable d'état du système dynamique étudié et chaque variable d'état caractérise le système à un instant donné. Pour chaque instant donné, le système est donc caractérisé par un point de cet espace. A l'instant suivant, il sera caractérisé par un autre point et ainsi de suite. Si l'espace des phases est représenté en trois dimensions, cette suite de points peut montrer graphiquement l'évolution du système dans le temps. L'ensemble des trajectoires possibles constitue le portrait de phases. Celui-ci peut aider à percevoir l'attracteur du système.

1.8 Les sections de Poincaré

Faire une section de Poincaré revient à couper la trajectoire dans l'espace des phases, afin d'étudier les intersections de cette trajectoire avec, par exemple en dimension trois, un plan. On passe alors d'un système dynamique à temps continu à un système dynamique à temps discret. Les mathématiciens ont bien sûr démontré que les propriétés du système sont conservées après la réalisation d'une section de Poincaré judicieusement choisie [8].

Exemples de sections de Poincaré

La section de Poincaré est de couper la trajectoire dans l'espace des phases par un plan (en dimension trois) ou par une droite (en dimension deux).

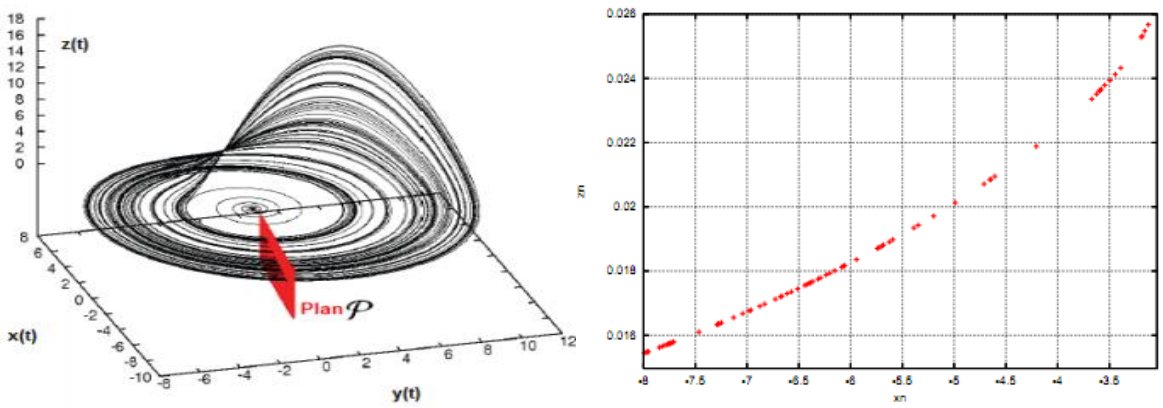


Figure (1.3) : Intersections de la trajectoire de l'attracteur de Rössler avec un plan P d'équation , y = 0 (x ≤ 0).

1.9 Bifurcation

1.9.1 La théorie de bifurcation

La théorie des bifurcations s'intéresse aux familles d'équations différentielles dépendant de paramètres μ :

$\frac{dx}{dt} = g_\mu(x)$, $x \in U \subset \mathbb{R}^n$, $\mu \in \mathbb{R}^k$ constant (1.12)

La bifurcation introduit par **Henri Poincaré** au début du siècle dans ses travaux sur les systèmes d'équations différentielles. Lorsqu'on crée des tourbillons dans un fluide, on observe une « bifurcation » de l'état de repos du fluide vers l'état convectif.

La bifurcation veut dire division d'une branche principale en au moins deux branches Le comportement d'un système dynamique linéaire peut changer quand un paramètre du système change. Ce changement de comportement correspond à un phénomène de bifurcation, il est accompagné d'un changement de type de stabilité.

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, ...

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

Il existe plusieurs types de bifurcations, parmi lesquelles on peut citer:

1 -Bifurcation Nœud-col (ou pli).

2 -Bifurcation Transcritique.

3 -Bifurcation fourche (sous-critique ou sur-critique).

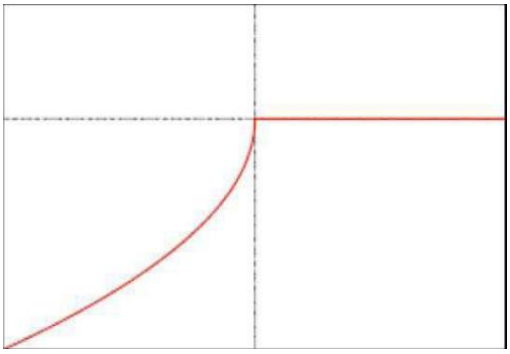


Figure (1.4) : Diagramme de Bifurcation Nœud-col

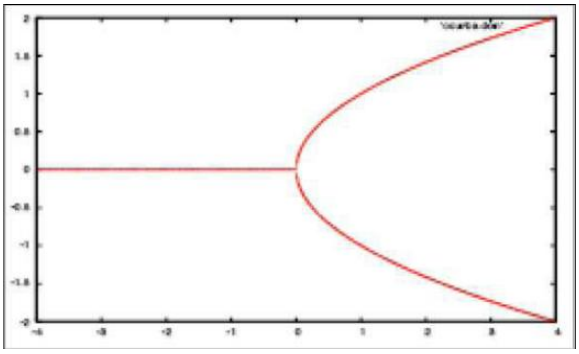


Figure (1.5) : Diagramme de Bifurcation fourche

Les trois premiers types de bifurcations correspondent à des bifurcations statistiques où le point de bifurcation sépare des branches de points fixes.

Les bifurcations de Hopf sont des bifurcations dynamiques où le point critique délimite dans l'espace de contrôle d'état des branches de points fixes et un cycle limite [8].

1.9.2 Diagramme de bifurcations

Un diagramme de bifurcation est une portion de l'espace des paramètres sur laquelle sont représentés tous les points de bifurcation. L'objectif d'un analyse de bifurcation est d'arriver à un, ou plusieurs, diagrammes de bifurcations.

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. C'est un diagramme qui porte les valeurs du paramètre en abscisse et des valeurs particulières d'une des variables d'état en ordonnée lorsque le régime asymptotique est atteint [9].

Exemple :

Le diagramme de bifurcation du système de Rössler écrit par l'équation (1.6) [9] :

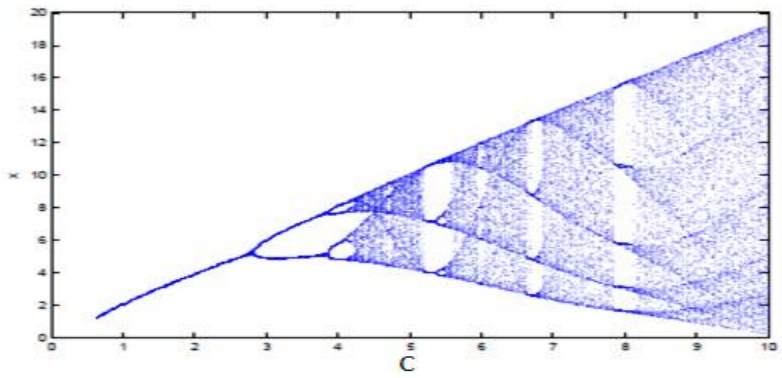


Figure (1.6) : Diagramme de bifurcation du système de Rössler

1.10 Attracteur

La région de l'espace de phases vers laquelle convergent les trajectoires d'un système dynamique dissipatif 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 : un point, un cycle limite, un tore ou avoir une structure encore plus complexe de type fractale [10].

1. L'attracteur "point fixe" est un point de l'espace de phase vers lequel tendent les trajectoires, c'est donc une solution stationnaire constante.

2. L'attracteur "cycle limite" est une trajectoire fermée dans l'espace des phases vers laquelle tendent les trajectoires. C'est donc une solution périodique du système.

3. L'attracteur "tore" représente les mouvements résultant de deux ou plusieurs oscillations indépendantes que l'on appelle parfois "mouvements quasi périodiques".

4. Les attracteurs étranges sont bien plus complexes que les autres, ils seront définis ultérieurement, on parle d'attracteur étrange lorsque la dimension fractale n'est pas entière.

Afin de bien rendre compte de ce qui peut être observé dans un espace des phases à trois dimensions, voici la représentation de l'attracteur de Lorenz. Il s'agit d'un attracteur chaotique (ou attracteur étrange), c'est-à-dire que cette figure géométrique est la représentation dans l'espace des phases d'un système chaotique.

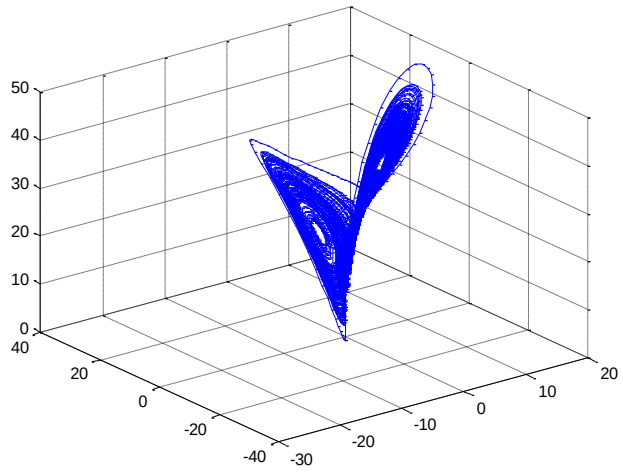


Figure (1.7) *:* Système chaotique de Lorenz dans l'espace des phases

L'objet géométrique observe (fig. 1.7) est relativement complexe et dégage la richesse d'informations que contient le système de Lorenz. Un attracteur chaotique possède notamment la propriété remarquable suivante : la trajectoire ne repasse jamais par un même état. Ce qui signifie, entre autres, que cette trajectoire passe par une infinité d'états.

- L'attracteur chaotique ou étrange est une forme géométrique plus complexe qui caractérise l'évolution des systèmes dynamiques chaotiques. **Voici quelques exemples :**

L'attracteur de Lorenz

L'attracteur de Lorenz tient son nom du météorologue Edward Lorenz qui l'a étudié le premier. C'est une simplification à l'extrême d'équations régissant les mouvements

atmosphériques. Lorenz les a étudiés afin de mettre en évidence sur un système simple la sensibilité aux conditions initiales qu'il avait observée.

Les équations de ce système sont les suivantes:

$$\begin{cases} \frac{dx}{dt} = a(y-x) \\ \frac{dy}{dt} = bx - y - xz \\ \frac{dz}{dt} = xy - cz \end{cases} \quad (1.13)$$

On prendra: *a= 10*, *b = 28* et *c = 8/3*. Ces valeurs impliquent un comportement chaotique (Figure(1.7)) [9].

L'attracteur de Rössler

Proposé par l'Allemand Otto Rössler(1974), le système de Rössler est lié à l'étude de l'écoulement des fluides, il découle des équations de Navier-Stokes. Les équations de ce système ont été découvertes à la suite de travaux en cinétique chimique.

Les équations de ce système sont les suivantes:

$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + xz - cz \end{cases} \quad (1.14)$$

Les dérivées des premiers membres sont des dérivées partielles par rapport au temps, *a*, *b* et *c* sont des constantes réelles. On prendra : *a = 0.398*, *b = 2* et *c = 4*.

On est alors en présence d'un système chaotique [9].

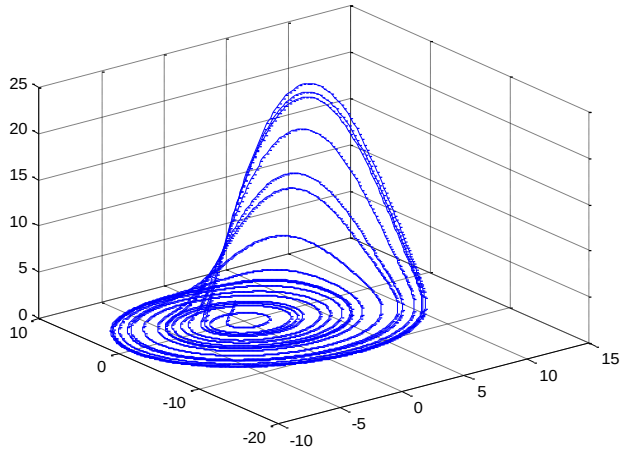


Figure (1.8) Attracteur de Rössler : (a, b, c) = (0.398, 2, 4).

Attracteur de lu

Les équations de ce système sont les suivantes:

$$\begin{cases} \frac{dx}{dt} = ay - ax \\ \frac{dy}{dt} = -x\,z + cy \\ \frac{dz}{dt} = xz - bz \end{cases} \quad (1.15)$$

Les dérivées des premiers membres sont des dérivées partielles par rapport au temps, *a*, *b* et *c* sont des constantes réelles. On prendra : *a = 36*, *b = 3* et *c = 20*.

On est alors en présence d'un système chaotique [9].

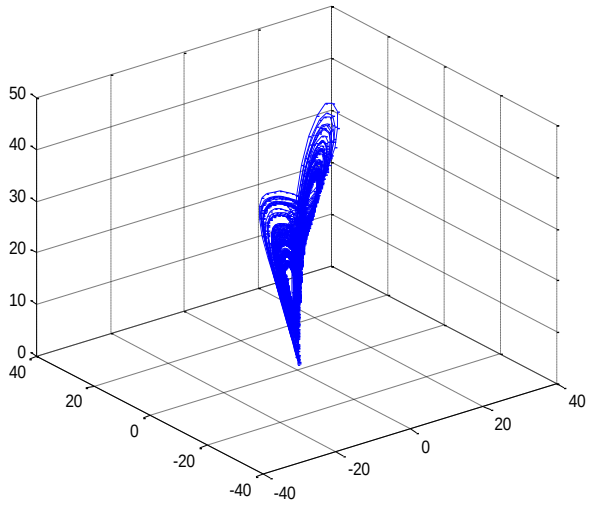


Figure (1.9) *:* Attracteur de lu : (a, b, c) = (36, 3, 20).

1.11 Conclusion

Dans ce chapitre nous avons présenté quelques notions et définitions de base de systèmes dynamiques et chaos, telles que les différents types de système dynamique, les systèmes a temps discret et les systèmes a temps continue, section de Poincaré.

Puis on a donné un aperçu de la théorie du chaos passant par définition, historique et caractéristiques du chaos, la notion de bifurcations. la dernière section du chapitre est les attracteurs, les différents types d'attracteurs.

Chapitre 2 synchronisation **des Systèmes chaotiques**

2.1 Introduction :

Depuis quelques années, la théorie des systèmes chaotiques a été appliquée dans le domaine des communications. 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. Tout changement de paramètre dans un système chaotique pourrait conduire à une divergence entre ces trajectoires.

En 1983, Chua se posa la question du chaos et de la synchronisation de systèmes chaotiques, qu'il aborde avec des circuits électriques linéaires par morceaux [11]. Dans les années 90, Pecora et Carroll ont montré que deux systèmes chaotiques pourraient se synchroniser sous certaines condition, si l'un deux est piloté par au moins une composante (une ou plusieurs variables d'état) de l'autre [12, 13].etc.

Depuis les années 90, de nombreux ouvrage ont été publiés au sujet de la synchronisation chaotique [14], [15], etc.

Une raison importante de l'intérêt des chercheurs vers la synchronisation des systèmes chaotiques est son application à la communication sécurisée [16], [17], etc. Les travaux de Pecora et Caroll ont permis de suggérer que les systèmes chaotiques pourraient être utilisés dans la communication, ou leur nature semblable aux bruits améliorerait la sécurité et le rejet de perturbation. En effet, une fois la synchronisation entre l'émetteur et le récepteur atteinte, il est possible de récupérer un message

masqué par l'émetteur chaotique. C'est ainsi qu'en 1990, Parlitz proposa le couplage de deux attracteurs étranges identiques dans le but de camoufler un message confidentiel en le superposant à un signal chaotique, et en le restaurant à la réception. Ce n'est donc qu'en 1992 que les ingénieurs ont réalisé des systèmes de communications chaotiques sécurisées [18, 19]. Dans [20], Oppenheim et al ,ont proposé des méthodes qu'ils ont appelées commutation chaotique ou modulation et masquage chaotique. Koracer et al, dans [21] proposent de noyer le message dans le système chaotique et d'utiliser le concept de synchronisation afin d'augmenter la sécurité de la communication.

Ce chapitre présente les principales méthodes de la synchronisation de systèmes chaotiques et en particulier l'utilisation d'observateurs pour reconstruire les états de l'émetteur chaotique. Pour cela nous vérifiant l'observabilité des différents systèmes chaotiques (Rossler, Lorenz, Sprott, Lu). Nous utilisons ensuite un observateur à mode glissants que nous avons étudié par simulation et appliquée à système chaotique (Lorenz). Les résultat de simulation seront présentés et commentés à la fin de ce chapitre.

2 .2 Méthodes de synchronisation

Les méthodes traditionnelles de synchronisation chaotique sont en général basées sur l'utilisation des circuits identiques.

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

Figure (2.1) : couplage unidirectionnel

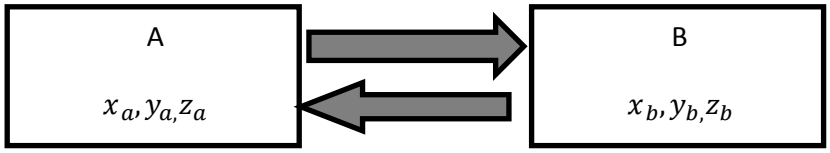


Figure (2.2) : couplage bidirectionnel

Dans le cas d'un couplage unidirectionnel, l'énergie est transférée d'un système à l'autre, à l'aide d'un ´élément de couplage fonctionnant dans un seul sens comme par exemple un suiveur.

Par contre, dans le couplage bidirectionnel, l'élément de couplage permet l'échange de l'énergie dans les deux sens. Ceci peut être par exemple une simple résistance.

Les deux types de couplage « unidirectionnel et bidirectionnel » peuvent aussi être appliqués aux systèmes non identiques.

En plus du couplage simple (par résistance ou suiveur), d'autres méthodes ont été proposées pour la synchronisation des systèmes chaotiques. Ainsi pour la synchronisation unidirectionnelle on peut citer la méthode :

- Décomposition du système [22, 23, 24],
- La synchronisation impulsive [25,26],
- La synchronisation par des méthodes itératives [27]
- La synchronisation par la boucle fermée. Cette dernière méthode peut être considérée comme une conception d'observateur.

Dans la majorité des cas, les deux systèmes doivent avoir des structures identiques, ce qui n'est pas tout à fait réalisable en pratique. Un petit écart entre les valeurs des composants peut entrainer un écart considérable entre les comportements des deux circuits et détruire le phénomène de synchronisation [28]

2 .3 Méthode de couplage unidirectionnel choisie : l'approche par observateur

2.3.1 Observabilité des systèmes linéaires

On considère le système linéaire suivant :

$$\begin{cases} \dot{x} = Ax + Bu \\ y = cx \end{cases} \quad (2.1)$$

Où : $x \in R^n$: vecteur qui représente les n variables d'état.

$u(t) \in R^m$: vecteur qui représente les m commandes.

$y(t) \in R^p$: vecteur qui représente les p mesures.

- Le système (2.1) est dit observable, si et seulement si on peut reconstruire l'état *x* par la seule connaissance de l'entrée u et de la sortie y.

2.3.1.1 Critère d'observabilité de kalman

Le système (2.1) est observable si et seulement si la matrice d'observabilité $O_{(A, C)}$ est régulière ou encore : le rang d'observabilité du système (2.1) est égal à n ; n étant la dimension de l'espace d'état. On dit alors que la paire (A, C) est observable.

$$rang\ O_{(A,C)} = rang[C \quad (CA) \quad (CA^2) \quad (CA^{n-1})]^T = n \quad (2.2)$$

Il s'agit d'une condition nécessaire et suffisante. Si la matrice d'observabilité $O_{(A,\ C)}$ est inversible (son déterminant est non nul), alors le système (2.1) est complètement observable [29].

$$det[O\ (A,C)] = det[C \quad (CA) \quad (CA2) \ldots. \ (CAn-1)\ ]^T \neq 0$$

2.3.2 Observabilité des systèmes non-linéaires

Soit un système non linéaire de la forme :

$$\begin{cases} \dot{\mathrm{x}} = \mathrm{f}(\mathrm{x}(\mathrm{t}), \mathrm{u}(\mathrm{t})) \\ \mathrm{y} = \mathrm{h}(\mathrm{x}(\mathrm{t})) \end{cases} \quad (2.3)$$

Où $x(t) \in \mathrm{R}^{\mathrm{n}}$ représentent l'état, $u(t) \in R^m$ l'entrée et $y(t) \in \mathrm{R}^{\mathrm{p}}$ la sortie. $f(\mathrm{x}(\mathrm{t}), \mathrm{u}(\mathrm{t}))$ et $h(\mathrm{x}(\mathrm{t}))$ sont des fonctions analytiques.

La notion d'observabilité d'un système non linéaire peut être définie à partir de la notion d'indiscernabilité d'une paire d'état, tel que le système non linéaire (2.3) est dit observable s'il n'admet pas de paires indiscernables.

Deux états initiaux $x_2(t0) = x_1$ et $x_2(t_0) = x_2$ sont dit indiscernables pour le système (2.3) si $\forall \in t[t_0; t_1]$ les sorties correspondantes $y_1(t)$ et $y_2(t)$ sont indiscernables quelle que soit l'entrée admissible u(t) du système (2.3).

Donc, contrairement au cas linéaire, l'observabilité d'un système non linéaire dépend de l'entrée appliquée. On doit donc prendre en compte le problème des entrées.

On appelle entrée universelle, l'entrée qui permet de discerner tout couple d'états initiaux par examen de la sortie. Dans le cas contraire, elle sera appelée entrée singulière. Lorsque le système non linéaire est dépourvu d'entrées singulières, il est appelé système uniformément observable.

Les critères permettant de déterminer l'observabilité d'un système non linéaire sont plus compliqués que dans le cas linéaire. Cependant, une notion d'observabilité faible locale peut être caractérisée par une condition de rang équivalant au cas linéaire.

On dit que le système non linéaire (2.3) satisfaisant la condition de rang d'observabilité, si :

$$\forall x \in \boldsymbol{R}^n \dim(\mathrm{d_o(x)}) = \mathrm{n} \qquad (2.4)$$

Tel que :

d_o : l'espace des différentielles des éléments de *o.*

o: l'espace d'observabilité.

L'espace d'observabilité noté *'O'* est le plus petit sous espace vectoriel de fonction de R^n à valeur dans l'espace de sortie, contenant les sorties h_1, h_2, … h_p et qui soit fermé sous l'opération de la dérivation de Lie par rapport au champ de vecteur f(x, u), u étant fixée.

Dérivée de Lie

Pour tout $u \in$ U notons par f_u le champ de vecteurs défini par $f(x) = f(x,u)$. Si φ est une fonction différentiable sur *M*, la dérivée de Lie de φ par rapport à f_u est notée par $L_{fu}(\varphi)$.

Dans un système de coordonnées $(x_1, x_2, \ldots, x_n)$ le champ s'écrit :

$$f_u = \sum_{i=1}^{n} \frac{\partial}{\partial x_i} \tag{2.5}$$

Alors :

$$l_f(\theta) = \sum_{i=1}^{n} \frac{\partial \theta}{\partial x_i} \tag{2.6}$$

Si *O* est un espace d'observabilité générique on peut dire que le système (2.3) est génériquement observable si et seulement si :

$$Dim\ (O) = n \tag{2.7}$$

Cette condition est appelée condition de rang d'observabilité générique. Si cette condition est satisfaite, on peut alors vérifier :

$$\boldsymbol{Rang}\begin{pmatrix} \boldsymbol{dh} \\ \boldsymbol{dl_f h} \\ \vdots \\ \boldsymbol{dl_f^{n-1} h} \end{pmatrix} \boldsymbol{= n} \tag{2.8}$$

L_f est l'opération de dérivée de Lie.

Cela implique que l'état x peut être déduit de la connaissance de la sortie et d'un nombre finie de ses dérivées.

Un seul critère est suffisant est que :

$$Det\ (J) \neq 0 \tag{2.9}$$

2.3.3 Analyse de l'observabilité de quelques systèmes chaotiques

2.3.3.1 Analyse de l'observabilité du système de Lu

Soit le système de Lu

$$\dot{x} = f(x) = \begin{cases} \dot{x_1} = 36x_2 - 36x_1 \\ \dot{x_2} = -x_1x_3 + 20x_2 \\ \dot{x_3} = x_1x_3 - 3x_3 \end{cases} \quad (2.10)$$

- **1^er^cas : la sortie est $x_1 (y = x_1)$**

$$\boldsymbol{O} = \begin{cases} y = x_1 \\ \dot{y} = \dot{x_1} = 36x_2 - 36x_1 \\ \ddot{y} = \ddot{x_1} = 1296x_1 - 1296x_2 - 108x_3 - 36x_1x_3 \end{cases} \quad (2.11)$$

Le calcul de la matrice Jacobienne donne :

$$\frac{\delta o}{\delta f} = \begin{pmatrix} \frac{\delta y}{\delta x_1} & \frac{\delta y}{\delta x_2} & \frac{\delta y}{\delta x_3} \\ \frac{\delta \dot{y}}{\delta x_1} & \frac{\delta \dot{y}}{\delta x_2} & \frac{\delta \dot{y}}{\delta x_3} \\ \frac{\delta \ddot{y}}{\delta x_1} & \frac{\delta \ddot{y}}{\delta x_2} & \frac{\delta \ddot{y}}{\delta x_3} \end{pmatrix} \Rightarrow \quad \frac{\sigma o}{\delta f} = \begin{pmatrix} 1 & 0 & 0 \\ -36 & 36 & 0 \\ 1296 - 36x_3 & -1296 & -108 \end{pmatrix} \quad (2.12)$$

Le déterminant de la matrice : $\det\frac{\delta O}{\delta f} = -3888 \neq 0$

- **2^eme^ cas : la sortie est $x_2 (y = x_2)$**

$$\boldsymbol{O} = \begin{cases} y = x_2 \\ \dot{y} = \dot{x_2} = -x_1x_3 + 20x_2 \\ \ddot{y} = \ddot{x_2} = -400 * x_2 - 128x_1x_3 + 108x_2x_3 + 36x_3x_1^2 - 36x_2 \end{cases} \quad (2.13)$$

Le calcul de la matrice jacobienne donne

$$\frac{\delta o}{\delta f} = \begin{pmatrix} \frac{\delta y}{\delta x_1} & \frac{\delta y}{\delta x_2} & \frac{\delta y}{\delta x_3} \\ \frac{\delta \dot{y}}{\delta x_1} & \frac{\delta \dot{y}}{\delta x_2} & \frac{\delta \dot{y}}{\delta x_3} \\ \frac{\delta \ddot{y}}{\delta x_1} & \frac{\delta \ddot{y}}{\delta x_2} & \frac{\delta \ddot{y}}{\delta x_3} \end{pmatrix} \Rightarrow$$

$$= \begin{pmatrix} 0 & 1 & 0 \\ -x_3 & 20 & -x_1 \\ -108x_3 + 72x_2x_3 - 36x_2x_3 & 108x_3 - 36x_1x_3 + 400 & -128x_1 + 108x_2 - 36x_1x_2 + 36x_1^2 \end{pmatrix} \quad (2.14)$$

$\det\frac{\delta O}{\delta f} \neq 0 \Longrightarrow 108x_2 - 36x_1^2x_3 \neq 0 \Longrightarrow$

$x_2 \neq \frac{36*x_1^2*x_3}{108}$ et $x_3 \neq \frac{108*x_2}{36*x_1^2}$ et $x_1 \neq \sqrt{\frac{108*x_2}{36*x_3}}$.

- **3eme cas : la sortie est x_3 ($y = x_3$)**

$\boldsymbol{O}$

$$= \begin{cases} y = x_3 \\ \dot{y} = \dot{x_3} = x_1x_3 - 3x_3 \\ \ddot{y} = \ddot{x_3} = -9x_3 + 105x_1x_3 - 108x_2x_3 + 36x_1x_2x_3 - 36x_1^2x_3 \end{cases} \quad (2.15)$$

Le calcul de la matrice jacobienne donne :

$$\frac{\delta o}{\delta f} = \begin{pmatrix} \frac{\delta y}{\delta x_1} & \frac{\delta y}{\delta x_2} & \frac{\delta y}{\delta x_3} \\ \frac{\delta \dot{y}}{\delta x_1} & \frac{\delta \dot{y}}{\delta x_2} & \frac{\delta \dot{y}}{\delta x_3} \\ \frac{\delta \ddot{y}}{\delta x_1} & \frac{\delta \ddot{y}}{\delta x_2} & \frac{\delta \ddot{y}}{\delta x_3} \end{pmatrix} \Longrightarrow$$

$$\begin{pmatrix} 0 & 0 & 1 \\ x_2 & 0 & x_1 - 3 \\ 36x_2x_3 - 72x_1x_3 + 96x_3 & -108x_3 + 36x_1x_3 & 105x_1 - 108x_2 + 108x_1^2 + 36x_1x_2 - 9 \end{pmatrix} \quad (2.16)$$

$\det\frac{\delta O}{\delta f} \neq 0 \Longrightarrow 108x_3^2 - 36x_3^2x_1 \neq 0 \Longrightarrow x_3 \neq 0$ et $x_1 \neq 3$

Alors

D'après l'étude qu'on a faite, on constate que $y = x_1$ est la meilleure solution pour réaliser l'observateur

2.3.3.2 Analyse de l'observabilité ou système de Sprott

Soit le système de sprott :

$$\dot{x} = f(x) = \begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = -x_1 + x_2x_3 \\ \dot{x_3} = 1 - x_3^2 \end{cases} \quad (2.17)$$

❖ **1^er cas : la sortie est $x_1 (y = x_1)$**

$$\boldsymbol{O} = \begin{cases} y = x_1 \\ \dot{y} = \dot{x_1} = x_2 \\ \ddot{y} = \ddot{x_1} = -x_1 + x_2 x_3 \end{cases} \quad (2.18)$$

Le calcul de la matrice jacobienne donne :

$$\frac{\delta o}{\delta f} = \begin{pmatrix} \frac{\delta y}{\delta x_1} & \frac{\delta y}{\delta x_2} & \frac{\delta y}{\delta x_3} \\ \frac{\delta \dot{y}}{\delta x_1} & \frac{\delta \dot{y}}{\delta x_2} & \frac{\delta \dot{y}}{\delta x_3} \\ \frac{\delta \ddot{y}}{\delta x_1} & \frac{\delta \ddot{y}}{\delta x_2} & \frac{\delta \ddot{y}}{\delta x_3} \end{pmatrix} \Longrightarrow \frac{\sigma o}{\delta f} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & x_3 & x_2 \end{pmatrix} \quad (2.19)$$

Le déterminant de la matrice :

$$\det \frac{\delta O}{\delta f} \neq 0 \Longrightarrow x_2 \neq 0$$

❖ **2^eme cas : la sortie est $x_2 (y = x_2)$**

$$O = \begin{cases} y = x_2 \\ \dot{y} = \dot{x_2} = -x_1 + x_2 x_3 \\ \ddot{y} = \ddot{x_2} = -x_1 - x_2 + x_2 x_3 \end{cases} \quad (2.20)$$

Le calcul de la matrice jacobienne donne :

$$\frac{\delta o}{\delta f} = \begin{pmatrix} \frac{\delta y}{\delta x_1} & \frac{\delta y}{\delta x_2} & \frac{\delta y}{\delta x_3} \\ \frac{\delta \dot{y}}{\delta x_1} & \frac{\delta \dot{y}}{\delta x_2} & \frac{\delta \dot{y}}{\delta x_3} \\ \frac{\delta \ddot{y}}{\delta x_1} & \frac{\delta \ddot{y}}{\delta x_2} & \frac{\delta \ddot{y}}{\delta x_3} \end{pmatrix} \Longrightarrow$$

$$\begin{pmatrix} 0 & 1 & 0 \\ -1 & x_3 & x_2 \\ -1 + x_3^2 + 20 x_1 x_2 & -1 + x_3 + 10 x_3^2 & 2 x_3 x_1 + x_2 - 3 x_2 x_3^2 \end{pmatrix} \quad (2.21)$$

$$\det \frac{\delta O}{\delta f} \neq 0 \Longrightarrow 2 x_1 x_3 - 2 x_2 x_3^2 \neq 0 \Longrightarrow x_1 \neq \sqrt{\frac{108 * x_2}{36 * x_3}}\ et x_2 \neq \frac{36 * x_1^2 * x_3}{108}\ \text{et} x_3 \neq \frac{108 * x_2}{36 * x_1^2}$$

❖ **3^eme cas : la sortie est $x_3 (y = x_3)$**

$$O = \begin{cases} y = x_3 \\ \dot{y} = \dot{x_3} = 1 - x_3^2 \\ \ddot{y} = \ddot{x_3} = -2 x_3^2 x_3^4 \end{cases} \quad (2.22)$$

Le calcul de la matrice jacobienne donne :

$$\frac{\delta o}{\delta f}=\begin{pmatrix}\frac{\delta y}{\delta x_1} & \frac{\delta y}{\delta x_2} & \frac{\delta y}{\delta x_3}\\ \frac{\delta \dot{y}}{\delta x_1} & \frac{\delta \dot{y}}{\delta x_2} & \frac{\delta \dot{y}}{\delta x_3}\\ \frac{\delta \ddot{y}}{\delta x_1} & \frac{\delta \ddot{y}}{\delta x_2} & \frac{\delta \ddot{y}}{\delta x_3}\end{pmatrix} \Longrightarrow \begin{pmatrix}0 & 0 & 1\\ 0 & 0 & -2x_3\\ 0 & 0 & -4x_3+10x_3^2\end{pmatrix} \quad (2.23)$$

$\det\frac{\delta O}{\delta f}=0$

Alors d'après l'étude qu'on a faite on constate que $y=x_1$ est la meilleur solution pour réaliser l'observateur.

2.3.3.3 Analyse de l'observabilité du système de Rossler

Soit le système de Rossler

$$\dot{x}=f(x)=\begin{cases}\dot{x_1}=-x_2-x_3\\ \dot{x_2}=x_1+0.398x_2\\ \dot{x_3}=2+x_1x_3-4x_3\end{cases} \quad (2.24)$$

- **1^er^cas : la sortie est $x_1(y=x_1)$**

$$\boldsymbol{O}=\begin{cases}y=x_1\\ \dot{y}=\dot{x_1}=-x_2-x_3\\ \ddot{y}=\ddot{x_1}=-x_1-0.398-2-x_1x_3+4x_3\end{cases} \quad (2.25)$$

Le calcul de la matrice jacobine donne :

$$\frac{\delta o}{\delta f}=\begin{pmatrix}\frac{\delta y}{\delta x_1} & \frac{\delta y}{\delta x_2} & \frac{\delta y}{\delta x_3}\\ \frac{\delta \dot{y}}{\delta x_1} & \frac{\delta \dot{y}}{\delta x_2} & \frac{\delta \dot{y}}{\delta x_3}\\ \frac{\delta \ddot{y}}{\delta x_1} & \frac{\delta \ddot{y}}{\delta x_2} & \frac{\delta \ddot{y}}{\delta x_3}\end{pmatrix} \Longrightarrow \frac{\sigma o}{\delta f}=\begin{pmatrix}1 & 0 & 0\\ 0 & -1 & -1\\ -1-x_3 & -0.398 & 4-x_1\end{pmatrix} \quad (2.26)$$

Le déterminant de la matrice :

$$\det\frac{\delta O}{\delta f}=-4.398+x_1\neq 0 \Longrightarrow x_1\neq 0$$

- **2^eme^cas : la sortie est $x_2(y=x_2)$**

$$\boldsymbol{O}=\begin{cases}y=x_2\\ \dot{y}=\dot{x_2}=x_1+0.398x_2\\ \ddot{y}=\ddot{x_2}=0.398x_1-0.841596x_2\end{cases} \quad (2.28)$$

Le calcul de la matrice jacobienne donne :

$$\frac{\delta o}{\delta f}=\begin{pmatrix}\frac{\delta y}{\delta x_1} & \frac{\delta y}{\delta x_2} & \frac{\delta y}{\delta x_3}\\ \frac{\delta \dot{y}}{\delta x_1} & \frac{\delta \dot{y}}{\delta x_2} & \frac{\delta \dot{y}}{\delta x_3}\\ \frac{\delta \ddot{y}}{\delta x_1} & \frac{\delta \ddot{y}}{\delta x_2} & \frac{\delta \ddot{y}}{\delta x_3}\end{pmatrix} \Longrightarrow \quad \frac{\delta o}{\delta f}=\begin{pmatrix}0 & 1 & 0\\ 1 & 0.398 & 0\\ 0.398 & -0.398 & -1\end{pmatrix} \qquad (2.29)$$

$\det\frac{\delta O}{\delta f}=1\neq 0$

- **3eme cas : la sortie est x_3 ($y=x_3$)**

$$\boldsymbol{O}=\begin{cases} y=x_3\\ \dot{y}=\dot{x_3}=2+x_1x_3-4x_3\\ \ddot{y}=\ddot{x_3}=-6+14x_3-2x_2-4x_1x_3+4x_2x_3+4x_3^2-x_1x_3^2\end{cases} \qquad (2.30)$$

Le calcul de la matrice jacobienne donne

$$\frac{\delta o}{\delta f}=\begin{pmatrix}\frac{\delta y}{\delta x_1} & \frac{\delta y}{\delta x_2} & \frac{\delta y}{\delta x_3}\\ \frac{\delta \dot{y}}{\delta x_1} & \frac{\delta \dot{y}}{\delta x_2} & \frac{\delta \dot{y}}{\delta x_3}\\ \frac{\delta \ddot{y}}{\delta x_1} & \frac{\delta \ddot{y}}{\delta x_2} & \frac{\delta \ddot{y}}{\delta x_3}\end{pmatrix} \Longrightarrow \begin{pmatrix}0 & 0 & 1\\ x_2 & 0 & -\frac{8}{3}\\ -4x_3-4_3^2 & -2+4x_3 & 14-4x_1+8x_3-2x_1x_3\end{pmatrix} \qquad (2.31)$$

$\det\frac{\delta O}{\delta f}\neq 0 \Longrightarrow -2x_3-4x_3^2\neq 0 \Longrightarrow x_3\neq 0\ et x_3\neq -1/2$

2.3.3.4 Analyse de l'observabilité ou système de Lorenz

Soit le système de Lorenz

$$\dot{x}=f(x)=\begin{cases}\dot{x_1}=10x_2-10x_1\\ \dot{x_2}=28x_1-x_2-x_1x_3\\ \dot{x_3}=-\frac{8}{3}x_3+x_1x_2\end{cases} \qquad (2.32)$$

- **1ercas : la sortie est x_1 ($y=x_1$)**

$$O=\begin{cases} y=x_1\\ \dot{y}=\dot{x_1}=10x_2-10x_1\\ \ddot{y}=\ddot{x_1}=180x-110x_2-10x_1x_{3_1}-110x_2-10x_1x_3\end{cases} \qquad (2.33)$$

Le calcul de la matrice jacobienne donne :

$$\frac{\delta o}{\delta f}=\begin{pmatrix}\frac{\delta y}{\delta x_1} & \frac{\delta y}{\delta x_2} & \frac{\delta y}{\delta x_3}\\ \frac{\delta \dot{y}}{\delta x_1} & \frac{\delta \dot{y}}{\delta x_2} & \frac{\delta \dot{y}}{\delta x_3}\\ \frac{\delta \ddot{y}}{\delta x_1} & \frac{\delta \ddot{y}}{\delta x_2} & \frac{\delta \ddot{y}}{\delta x_3}\end{pmatrix} \Longrightarrow \quad \frac{\sigma o}{\delta f}=\begin{pmatrix}1 & 0 & 0\\ -10 & 10 & 0\\ 180-10x_3 & 110 & -10x_1\end{pmatrix} \qquad (2.34)$$

Le déterminant de la matrice : $\det\frac{\delta O}{\delta f}\neq 0 \Longrightarrow \; x_1\neq 0$

- **2eme cas : la sortie est x_2 ($y=x_2$)**

$$O=\begin{cases} y=x_2\\ \dot{y}=\dot{x_2}=28x_1-x_2-x_1x_3\\ \ddot{y}=\ddot{x_2}=380*x_1-281*x_2-\frac{77}{3}x_1x_3+\frac{80}{3}x_2x_3+10x_1x_2^2+10x_1^2x_2\end{cases} \qquad (2.35)$$

Le calcul de la matrice jacobienne donne

$$\frac{\delta o}{\delta f}=\begin{pmatrix}\frac{\delta y}{\delta x_1} & \frac{\delta y}{\delta x_2} & \frac{\delta y}{\delta x_3}\\ \frac{\delta \dot{y}}{\delta x_1} & \frac{\delta \dot{y}}{\delta x_2} & \frac{\delta \dot{y}}{\delta x_3}\\ \frac{\delta \ddot{y}}{\delta x_1} & \frac{\delta \ddot{y}}{\delta x_2} & \frac{\delta \ddot{y}}{\delta x_3}\end{pmatrix} \Longrightarrow$$

$$=\begin{pmatrix}0 & 1 & 0\\ 28-x_3 & -1 & -x_1\\ -308-\frac{77}{3}x_3+10x_2^2+20x_1x_2 & 281+\frac{80}{3}x_3+20x_1x_2+10x_1^2 & -\frac{77}{3}x_1+\frac{80}{3}x_2\end{pmatrix} \qquad (2.36)$$

$$\det\frac{\delta O}{\delta f}\neq 0 \Longrightarrow \quad -\frac{1232}{3}x_1-x_1x_3-\frac{2140}{3}x_2+\frac{80}{3}x_2x_3-10x_1x_2^2-10x_2x_1^2\neq 0$$

- **3eme cas : la sortie est x_3 ($y=x_3$)**

$$\boldsymbol{O}=\begin{cases} y=x_3\\ \dot{y}=\dot{x_3}=-\frac{8}{3}x_3+x_1x_2\\ \ddot{y}=\ddot{x_3}=\frac{64}{9}x_3-\frac{862}{3}x_1x_2-10x_2^2-10x_1x_2x_3-280x_1^2+10x_1^2x_3\end{cases} \qquad (2.37)$$

Le calcul de la matrice jacobienne donne :

$$\frac{\delta o}{\delta f}=\begin{pmatrix}\frac{\delta y}{\delta x_1} & \frac{\delta y}{\delta x_2} & \frac{\delta y}{\delta x_3}\\ \frac{\delta \dot{y}}{\delta x_1} & \frac{\delta \dot{y}}{\delta x_2} & \frac{\delta \dot{y}}{\delta x_3}\\ \frac{\delta \ddot{y}}{\delta x_1} & \frac{\delta \ddot{y}}{\delta x_2} & \frac{\delta \ddot{y}}{\delta x_3}\end{pmatrix} \Longrightarrow$$

$$\begin{pmatrix} 0 & 0 & 1 \\ x_2 & x_1 & -\frac{8}{3} \\ -\frac{862}{3}x_2 - 10x_2x_3 - 560x_1 + 20x_1x_2 & -\frac{862}{3}x_1 - 20x_2 - 10x_1x_3 & \frac{64}{3} - 10x_2x_3 + 10x_1^2 \end{pmatrix} \quad (2.38)$$

$$\det\frac{\delta O}{\delta f} \neq 0 \Longrightarrow 560x_1^2 - 20x_2^2 - 20x_1^2x_2 \neq 0 \Longrightarrow x_1 \neq \sqrt{\frac{20x_2^2}{560 - 20x_2}}$$

Alors

D'après l'étude qu'on a faite on constate que $y = x_1$ est la meilleure solution pour réaliser l'observateur.

2 .4 Observateur

Pour reconstituer l'état complet du système, l'idée repose sur l'utilisation d'un capteur logiciel, appelé observateur.

Un observateur est un système dynamique qui à partir de $u(t)$du système (la commande), de la sortie $y(t)$ mesurée, ainsi que d'une connaissance à priori du modèle, fournira en sortie un état estimé $\hat{x}(t)$ qui devra tendre vers l'état réel $x(t)$[30].

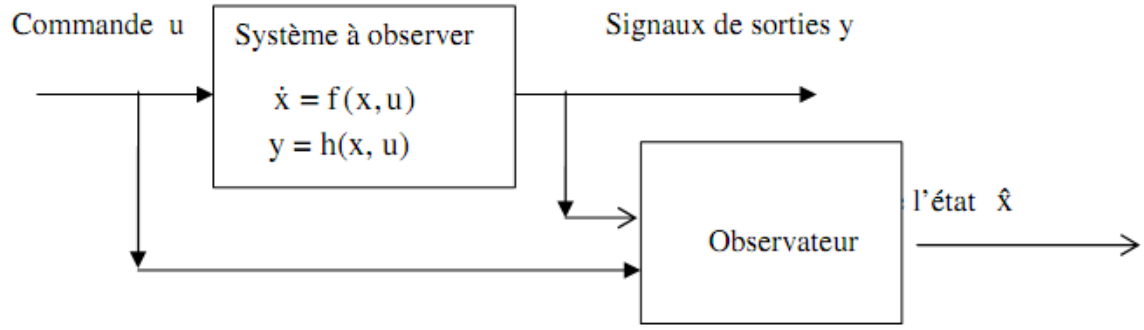


Figure (2.3) : Principe d'un observateur

L'observation se fait en deux phases ; la première est une étape d'estimation et la seconde est une étape de correction.

L'estimation se fait par le calcul des grandeurs d'états à l'aide d'un modèle du système.

La correction se fait par l'addition de la différence entre les états estimés et ceux mesurés (erreur d'estimation) que l'on multiplie par un gain K.

2.4.1 Observateurs des systèmes linéaires

Une solution au problème de l'estimation de l'état des systèmes linéaires a été proposée par Luenberger [31] dans le cadre déterministe, et par Kalman [32] dans le cadre stochastique.

2.4.2 Observateur des systèmes non linéaires

Une fois l'observabilité du système non linéaire est assurée, on passe à la synthèse de l'observateur.

La plupart des observateurs synthétisés pour les systèmes non linéaires et qui existent dans la littérature ont la structure suivante [33] :

$$\begin{cases} \dot{\hat{x}} = f(\hat{x}, u) + \eta(y, \hat{x}) \\ y = h\big(x(t)\big) \end{cases} \tag{2.39}$$

C'est-à-dire une copie du modèle plus un terme correcteur $\eta(y,\hat{x})$ qui établit la convergence de l'état estimé $\hat{x}$ vers l'état réel x en un temps fini. En général le gain d'observation et la stabilité de l'observateur synthétisé pour les systèmes non linéaires dépendent de l'entrée [34].

La synthèse des observateurs pour les systèmes non linéaires reste un sujet encore ouvert pour la recherche.

Parmi ces observateurs, on cite :

- Les observateurs dits à ''grand gain''.
- Observateurs adaptatifs.
- Observateurs pour systèmes sous une forme canonique.
- Linéarisation étendue.
- Observateurs par modes glissants.
- Les observateurs basés sur l'optimisation.

Dans ce qui suit on s'intéressera au mode glissant étape par étape.

2.5 Les modes glissants

Le contrôle par mode glissant est un mode de fonctionnement particulier des systèmes à structures variables. Le terme système à structure variables apparait à cause de la structure particulière du système ou de correcteur utilisé. Celui-ci change sa structure d'une façon discontinue entre au moins deux structures.

Les circuits d'électroniques de puissances constituent un exemple type de système à structure variable. Pour chaque position de l'interrupteur, le système est gouverné par une équation différentielle différente. Le contrôle par mode de glissement possède des avantages incontestables. Cette méthode a aussi quelques inconvénients dus au fait qu'elle nécessite une forte sollicitation de l'organe de commande.

2.5.1 Synthèse de l'observateur à mode glissant étape par étape

Un observateur à mode glissant est un observateur dont le terme correcteur est une fonction **sign** discontinu. On considère ici une approche reposant sur la théorie du système à structure variable, conduisant à la synthèse d'observateur appelé observateur à mode glissant.

Les propriétés qui nous intéressons dans ce type d'observateurs sont la convergence en temps fini vers la ou les surfaces de glissement, la réduction de la dynamique totale de *n* à *n-p* état quand l'état est sur la surface de glissement et le plus important est sa robustesse vis-à-vis des perturbations [35].

Remarque

L'observateur à mode glissant étape par étape ne peut pas être appliqué directement pour les systèmes chaotiques ; donc an applique la transformation à la forme canonique [36].

2.5.2 Forme canonique d'observabilité (FCO)

La FCO est utilisée pour observer l'état continu x sans avoir besoin d'aucune information concernant la dernière dynamique du système

Pour transforme le système (2.6) a la forme canonique on faire changement de coordonnées

$$\begin{cases} \dot{\boldsymbol{z}}_{\boldsymbol{1}} = \boldsymbol{x}_{\boldsymbol{2}} \\ \dot{\boldsymbol{z}}_{\boldsymbol{1}} = \boldsymbol{x}_{\boldsymbol{3}} \\ \quad . \\ \quad . \\ \dot{\boldsymbol{z}}_{\boldsymbol{n}} = \boldsymbol{f}(\boldsymbol{z}) \\ \boldsymbol{y} = \boldsymbol{x}_{\boldsymbol{1}} \end{cases} \tag{2.40}$$

2.5.3 Observateur triangulaire

$$\begin{cases} \dot{\hat{\boldsymbol{z}}}_{\boldsymbol{1}} = \hat{\boldsymbol{z}}_{\boldsymbol{2}} + \boldsymbol{\lambda}_{\boldsymbol{1}} * \boldsymbol{E}_{\boldsymbol{1}} * \boldsymbol{sign}(\boldsymbol{z}_{\boldsymbol{1}} - \tilde{\boldsymbol{z}}_{\boldsymbol{1}}) \\ \dot{\hat{\boldsymbol{z}}}_{\boldsymbol{2}} = \hat{\boldsymbol{z}}_{\boldsymbol{3}} + \boldsymbol{\lambda}_{\boldsymbol{2}} * \boldsymbol{E}_{\boldsymbol{2}} * \boldsymbol{sign}(\tilde{\boldsymbol{z}}_{\boldsymbol{2}} - \hat{\boldsymbol{z}}_{\boldsymbol{2}}) \\ \qquad\qquad . \\ \qquad\qquad . \\ \qquad\qquad . \\ \dot{\hat{\boldsymbol{z}}}_{\boldsymbol{N}} = \boldsymbol{f}(\boldsymbol{z}) + \boldsymbol{\lambda}_{\boldsymbol{N}} * \boldsymbol{E}_{\boldsymbol{N}} * \boldsymbol{sign}(\tilde{\boldsymbol{z}}_{\boldsymbol{N}} - \hat{\boldsymbol{z}}_{\boldsymbol{N}}) \end{cases} \tag{2.41}$$

Avec les conditions suivantes :

Si $(z_1 = \tilde{z}_1)$ alors $E_1 = 1$sinon $E_1 = 0$ ainsi lorsque $E_1 = 1$, et l'observateur synchronise l'état z_2.

Aussi si $\tilde{z}_2 = \hat{z}_2$ et $E_1 = 1$ $sinon$ $E_2 = 0$ et l'observateur synchronise l'état z_3 …etc Avec les états auxiliaires comme suit

$$\tilde{\boldsymbol{z}}_{\boldsymbol{j}} = \tilde{\boldsymbol{z}}_{\boldsymbol{j}} + \boldsymbol{\lambda}_{\boldsymbol{j-1}} * \boldsymbol{sing}(\tilde{\boldsymbol{z}}_{\boldsymbol{j-1}} - \hat{\boldsymbol{z}}_{\boldsymbol{j-1}}) \qquad \boldsymbol{j=2: n} \tag{2.42}$$

Il existe un choix λ_N et α_N tel que l'état observe $\dot{\hat{z}}_N$ converge en temps fini [37] :

$$\begin{cases} \alpha_{\mathrm{N}} > \mathrm{f}^{+} \\ \lambda_{\mathrm{N}} > (\alpha_{\mathrm{N}} + \mathrm{f}^{+}) \sqrt{\dfrac{2}{(\alpha_{\mathrm{N}} - \mathrm{f}^{+})}} \end{cases} \tag{2.43}$$

Avec f^{+}= max$|f(z)|$

Remarque : En pratique on prend

$E_N = 1$ SI $|\tilde{e}_N|$=$|\tilde{z}_N - \hat{z}_N| \leq \varepsilon$, $sinon\ E_N = 0$

2.5.4 Avantages de l'observateur en mode glissement

- Réduction de l'ordre du système
- Convergence en temps fini
- Robustesse vis-à-vis desincertitude paramétrique et perturbation

2.5. 5 Inconvénient de l'observateur en mode glissement

- Phénomène de chattering⇒ mauvaise prise de décision

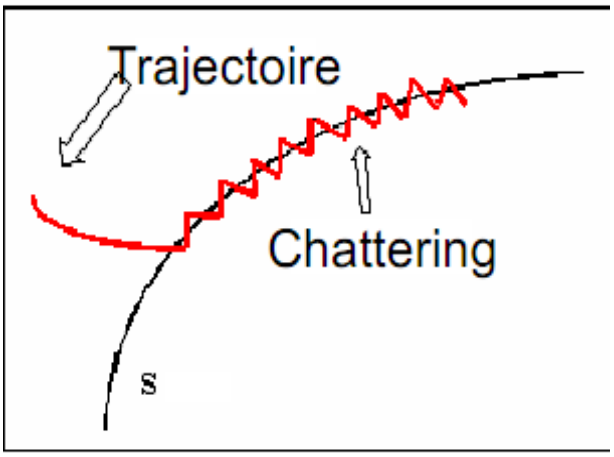


Figure (2.4) : Phénomène de chattering

2.6 Conception et réalisation de l'observateur à modes glissants pour système de Lorenz

2.6.1Transformé a la forme canonique d'observabilité

On applique la forme triangulaire

$$\begin{cases} z_1 = x_1 \\ z_2 = \dot{x}_1 \\ z_3 = \ddot{x}_1 \end{cases} \tag{2.44}$$

$$\dot{z} = f(z) = \begin{cases} \dot{z}_1 = \dot{x}_1 = z_2 \\ \dot{z}_2 = \ddot{x}_1 = z_3 \\ \dot{z}_3 = \dddot{x}_1 = \mathrm{f(z)} \end{cases} \tag{2.45}$$

Avec

*f(z)= $\dot{z}_3$ =773.3*z_1 -29.3 *z_2- 13.6 *z_3 -10*(z_1)^3 -(z_1 ^2)* z_2 + (1+10)*(z_2)^2)/ z_1 +*

z_2*z_3 /z_1

On obtient la forme canonique d'observable :

$$\begin{pmatrix}\dot{z}_1\\ \dot{z}_2\\ \dot{z}_3\end{pmatrix}=\begin{pmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}z_1\\ z_2\\ z_3\end{pmatrix}+\begin{pmatrix}0\\ 0\\ f(z)\end{pmatrix}$$
$$Y=\begin{pmatrix}1 & 0 & 0\end{pmatrix}\begin{pmatrix}z_1\\ z_2\\ z_3\end{pmatrix} \qquad (2.46)$$

2.6.2 Observateur triangulaire (Mode glissant étape par étape)

Soit le système (2.45) avec la sortie $y = z_1$ le but est de concevoir un observateur à modes glissant étape par étape qui permet, a partir de l'état z_1 , de reconstruire touts les états du système (appelée $\hat{z}_1, \hat{z}_2, \hat{z}_3$). Les équations de l'observateur sont données dans (2 .46)

$$\begin{cases}\dot{\hat{z}}_1 = \hat{z}_2 + \lambda_1 * E_1 * sign(z_1 - \tilde{z}_1)\\ \dot{\hat{z}}_2 = \hat{z}_3 + \lambda_2 * E_2 * sign(\tilde{z}_2 - \hat{z}_2)\\ \dot{\hat{z}}_3 = f(z) + \lambda_3 * E_3 * sign(\tilde{z}_3 - \hat{z}_3)\end{cases} \qquad (2.47)$$

On calcule maintenant les états auxiliaires comme suit :

$\tilde{z}_j = \tilde{z}_j + \lambda_{j-1} * sing(\tilde{z}_{j-1} - \hat{z}_{j-1})$ j=2 : n

On obtient les états auxiliaires $(\tilde{z}_2, \tilde{z}_3)$

$$\begin{cases}\tilde{z}_2 = \hat{z}_2 + \lambda_1 * sign(z_1 - \hat{z}_1)\\ \tilde{z}_3 = \hat{z}_3 + \lambda_2 * sing(\tilde{z}_2 - \hat{z}_2)\end{cases} \qquad (2.48)$$

Donc

$$\begin{cases}(\tilde{z}_2 - \hat{z}_2) = \lambda_1 * \mathrm{sign}(z_1 - \hat{z}_1)\\ (\tilde{z}_3 - \hat{z}_3) = \lambda_2 * \mathrm{sign}\,(\lambda_1 * \mathrm{sign}(z_1 - \hat{z}_1))\end{cases} \qquad (2.49)$$

Alors

$$\begin{cases}\dot{\hat{z}}_1 = \hat{z}_2 + \lambda_1 * E_1 * \mathrm{sign}(z_1 - \tilde{z}_1)\\ \dot{\hat{z}}_2 = \hat{z}_3 + \lambda_2 * E_2 * \mathrm{sign}(\lambda_1 * \mathrm{sign}(z_1 - \hat{z}_1))\\ \dot{\hat{z}}_3 = f(z) + \lambda_3 * E_3 * \mathrm{sign}\big(\lambda_2 * \mathrm{sign}\,(\lambda_1 * \mathrm{sign}(z_1 - \hat{z}_1))\big)\end{cases} \qquad (2.50)$$

L'observateur à mode glissant fonctionne étape par étape : la première étape consiste à reconstruire le signal z_1 . Une fois synchronisé le signal, et lorsque l'erreur $e_1 = z_1 - \hat{z}_1$ converge vers zéro, l'observateur reconstruira l'état suivant, soit z_2 ;

La dernière étape consiste à reconstruire z_3, et cela lorsque $e_3 = z_3 - \hat{z}_3$ converge vers zéro.

Remarque

Pour trouve le système original (x_1, x_2, x_3) utiles la transformait invars :

$$\dot{\hat{x}}_1 = f(\hat{z})^{\cdot} = \begin{cases} \dot{\hat{x}}_1 = \dot{\hat{z}}_2 \\ \dot{\hat{x}}_2 = \dot{\hat{z}}_1 + \dot{\hat{z}}_2/10 \\ \dot{\hat{x}}_3 = (29 * \dot{\hat{z}}_1 - 1.1 * \dot{\hat{z}}_2 - 0.1 * \dot{\hat{z}}_3)/\dot{\hat{z}}_1 \end{cases} \quad (2.51)$$

2.6.3 Schéma de simulation sous matlab/Simulink

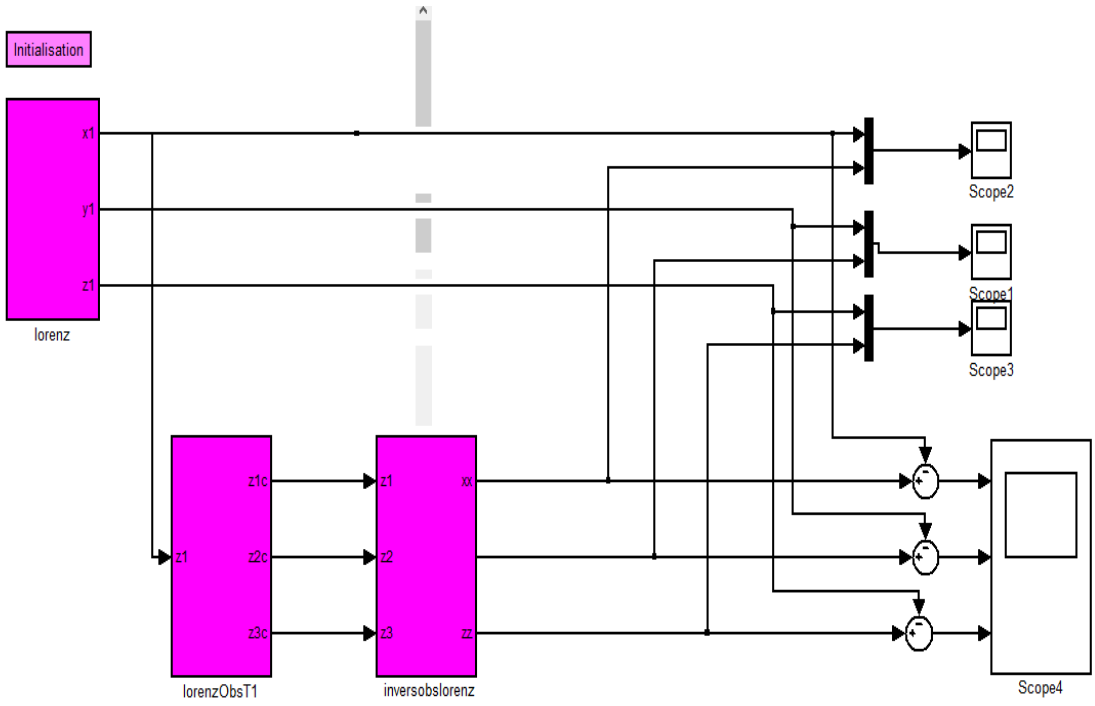


Figure (2.5) : schéma de synchronisation par observateur en mode glissante sur Simulink

2.6.4 Résultat de simulation sous matlab /Simulink

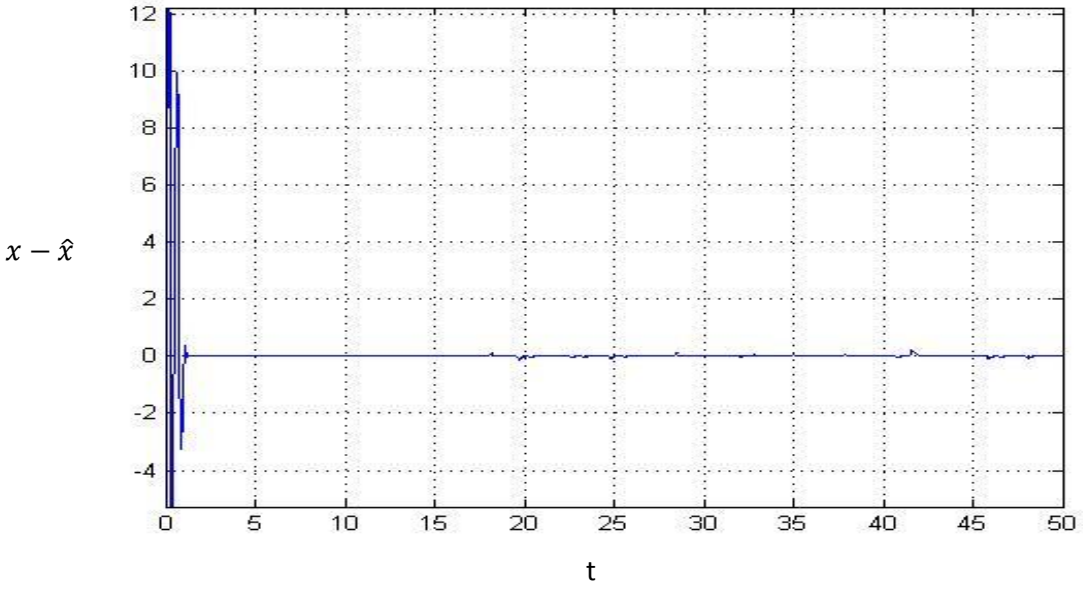


Figure (2.6) : la trajectoire d'erreur avec la synchronisation en mode glissant(x, $\hat{x}$).

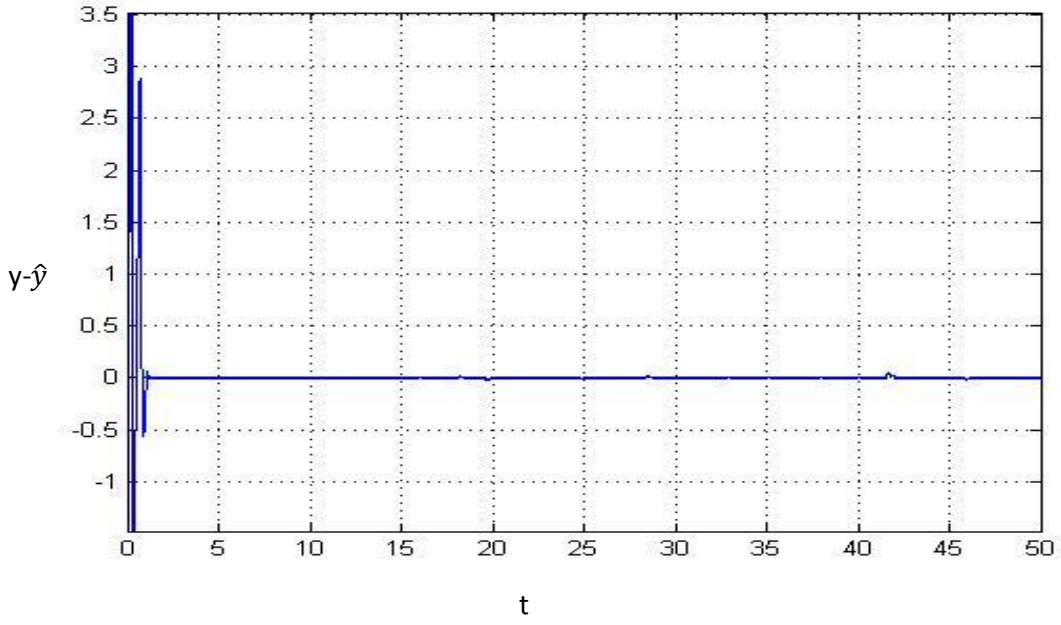


Figure (2.7) : la trajectoire d'erreur avec la synchronisation en mode glissant (y,$\hat{y}$)

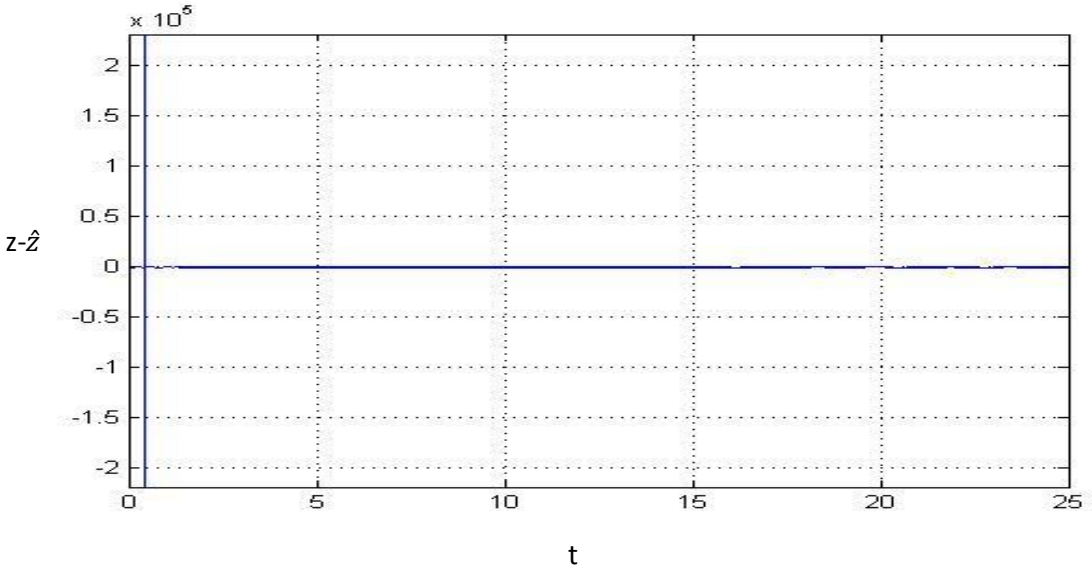


Figure (2.8) : la trajectoire d'erreur avec la synchronisation en mode glissant (z,$\hat{z}$)

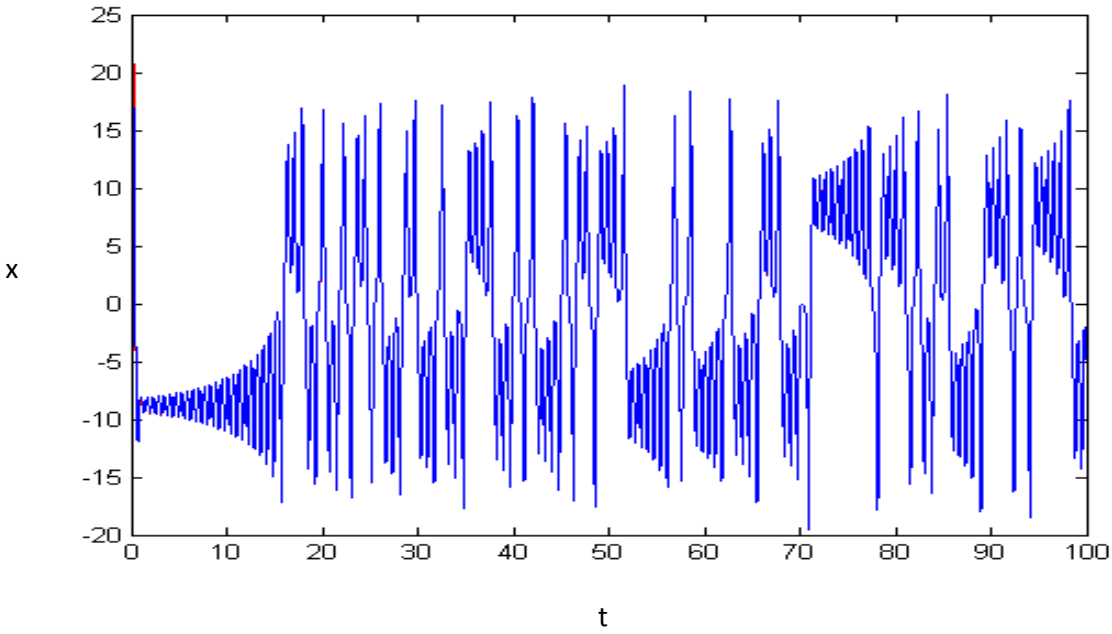


Figure (2.9) : la synchronisation en mode glissant (x, $\hat{x}$)

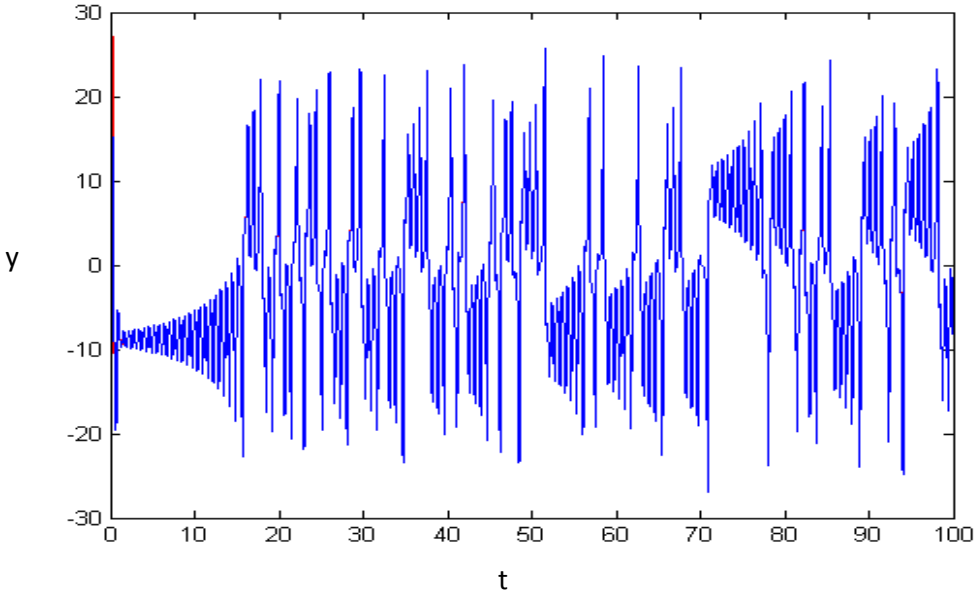


Figure (2.10) : la synchronisation en mode glissant (y, $\hat{y}$)

2.7 Commentaires

Les figures ((2.6), (2.7), (2.8)) : représentent la trajectoire d'erreur avec la synchronisation en mode glissant entre les états de systèmes (maître et esclave), avec les valeurs de $(k_1 = 60, k_2 = 60, k_3 = 60)$ qu'on a choisi avec des valeurs initiales de maître (1, 5,1) et esclave (5, 1,0.5).

2.8 Conclusion

La majorité des méthodes classiques des synthèses d'observateurs pour les systèmes non linéaires sont basées, directement ou indirectement, sur le fait que leur approximation linéaire est observable ou au moins détectable. Dans ce chapitre, nous avons montré qu'il est possible de synthétiser des observateurs pour des systèmes non linéaire, La solution que nous avons apporté à ce problème est d'une part l'analyse d'observabilité et d'autre part la synthèse d'observateur à modes glissants étape par étape, Ainsi on a déduit leurs avantages dans l'estimation des états du système à partir des états mesurés. Grâce à ces derniers l'observateur a modes glissant a été proposés dans différentes application, entre autres la synchronisation des systèmes chaotiques, qui a fait l'objet de notre travail.

Chapitre 3 Implémentation du système chaotique sur la carte DSP TMS320F2812

3.1 Introduction

Grâce à sa rapidité, sa flexibilité ainsi de sa simplicité de conception, et avec le progrès de la microélectronique le traitement numérique devient de plus en plus important et très utilisable dans beaucoup de domaines. Pour cela plusieurs, dispositifs de traitement numérique ont été développés au vingtième siècle comme le microprocesseur, le microcontrôleur, DSP (digital signal processor) et le FPGA (Field Programmable Gate Array).

Dans les domaines d'applications du traitement numérique du signal, le DSP présente des avantages qui l'on ne trouve pas dans d'autres processeurs classiques comme le calcule en temps réel et la rapidité d'exécution. Il contient aussi des périphériques spécialisés pour le traitement des signaux.

Dans ce chapitre on propose une implémentation de la méthode RK4 appliquée aux systèmes de Rössler, Lorenz et lu sur la carte DSP TMS320F2812.

Pour cela on va présenter les différentes étapes d'implémentation et la programmation des algorithmes, ainsi que la configuration des différents registres de DSP.

3.2 Présentation des DSP

Un DSP (Digital signal Processor) processeur de traitement numérique du signal est un composant électronique programmable de type processeur. Optimisé pour exécuter des applications de traitement numérique du signal (filtrage, extraction de signaux, etc.) le plus rapidement possible.

Les DSP sont utilisés dans la plupart des applications du traitement numérique du signal en temps réel**.**

Les applications des DSP sont utilisé dans nombreuses domaines voici quelques exemples des applications les plus courantes suivants [38]:

- **Télécommunications**

Modem, multiplexeurs, récepteurs de numérotation DTMF, télécopieurs, codeurs de parole GMS, …

- **Militaire**

Guidage missiles, navigation, communications cryptée, radar, …

- **Médical**

Compression d'image médicale (IRM, échographie…), traitements des signaux biophysiques,

- **Les multimédias et du grand public**

Le traitement de la parole (compression, reconnaissance et synthèse), la compression des images fixes ou animées, les cartes multimédias pour PC ou stations de travail, les jeux, …

3.3 Caractéristiques principales des DSP

Ce qui distingue deux DSP différents sont, pour les caractéristiques principales :

3.3.1 Son type

Deux familles principales de dsp se partagent le marché virgule fixe / virgule flottante :

3.3.1.1 les dsp à virgule fixe

Les données sont représentées comme étant des nombres fractionnaires à virgule fixe, (exemple -1.0 à +1.0), ou comme des entiers classiques. La représentation de ces nombres fractionnaires s'appuie la méthode du « complément à deux ». De cette représentation est de permettre facilement l'addition et la soustraction binaires de nombres aussi bien positifs que négatifs.

Un DSP à virgule fixe est un peu plus compliqué à programmer qu'un DSP à virgule flottante. Dans un DSP à virgule fixe, les nombres sont codés sur 16 bits (rappel : des entiers classiques ou des fractionnaires). Toutefois, sur ce DSP, les calculs sont effectués avec des accumulateurs de 32 bits. Lorsque les résultats doivent être stockés en mémoire, les 16 bit les moins significatifs sont perdus. Ceci permet de limiter les erreurs d'arrondis cumulatives. Il est toujours possible de stocker séparément en mémoire les 16 bit faibles, puis les 16 bit forts, s'il n'y a plus de registres libres lors d'une étape de calcul [39].

3.3.1.2 dsp a virgule flottante :

Les données sont représentées en utilisant une mantisse et un exposant.

La représentation de ces nombres s'effectue selon la formule suivante :

$$\mathrm{n} = \mathrm{mantisse} * 2^{\mathrm{exposant}}$$

Généralement, la mantisse est un nombre fractionnaire (-1.0 à +1.0), et l'exposant est un entier indiquant la place de la virgule en base 2 (c'est le même mécanisme qu'en base 10) [39].

3.3.2 Sa vitesse *:*

exprimée en MIPS (Méga Instructions par Seconde), qui n'est pas obligatoirement représentative des performances du DSP. En effet, le nombre de cycles d'horloge par instruction peut varier d'un DSP à l'autre. De plus, cela dépend aussi de la complexité des instructions. Les instructions peuvent être basiques (lecture/écriture d'une donnée dans un registre) ou plus étoffées (chargement d'une donnée, élévation au carré, addition d'un registre avec l'accumulateur) compte tenu de l'architecture même du DSP.

3.3.3 Sa quantité de mémoire interne

DRAM –RAM - ROM - FLASH.

3.3.4 Ses entrées/sorties

Ports série ou ports parallèles et leurs vitesses respectives

3.3.5 Son architecture interne des processeurs

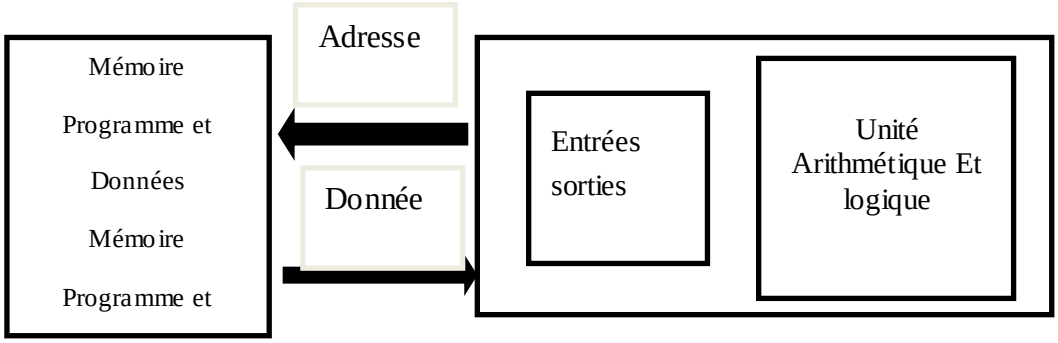
L'architecture d'un microprocesseur, et donc d'un dsp, est un élément important qui conditionne directement les performances d'un processeur.

Il existe deux types fondamentaux de structures, dites « von neuman » et « Harvard »[40*].*

3.3.5.1 structures de von neuman :

Un microprocesseur basé sur une structure Von Neuman stocke les programmes et les données dans la même zone mémoire. Une instruction contient le code opératoire et l'adresse de l'opérande. Ce type de microprocesseur incorpore principalement deux unités logiques de base :

- L'unité Arithmétique et Logique (ou ALU en anglais), chargée de réaliser les opérations centrales (de type multiplications, soustractions, rotation, etc.) ;
- L'unité en charge des Entrées/Sortie, qui commandent le flux de données entre le cœur du microprocesseur et les mémoires ou les ports



Figure(3.1) : Principe de l'architecture de Von Neuman

3.3.5.2 :structure de harvard :

Cette structure se distingue de celle de Von Neuman par le fait que les mémoires programme est données sont séparées. L'accès à chacune des deux mémoires se fait via deux chemins distincts. Cette organisation permet de transférer une instruction et une donnée simultanément, elle est considérée comme deux fois plus rapide que celle de Von Neuman, mais ce gain de vitesse se fait au prix d'une sérieuse complication de l'électronique puisque toute l'architecture des bus est doublée.

Pour réduire le coût de la structure Harvard, certains DSP utilisent la structure de Harvard modifiée. A l'extérieur, le DSP ne propose qu'un seul bus de donnée et un bus d'adresse, comme la structure Von Neuman. Toutefois, à l'intérieur, la puce DSP dispose de deux bus distincts de données et de deux bus distincts d'adresses.

Le transfert des données entre les bus internes et les bus externes est effectué par un multiplexage temporel [40].

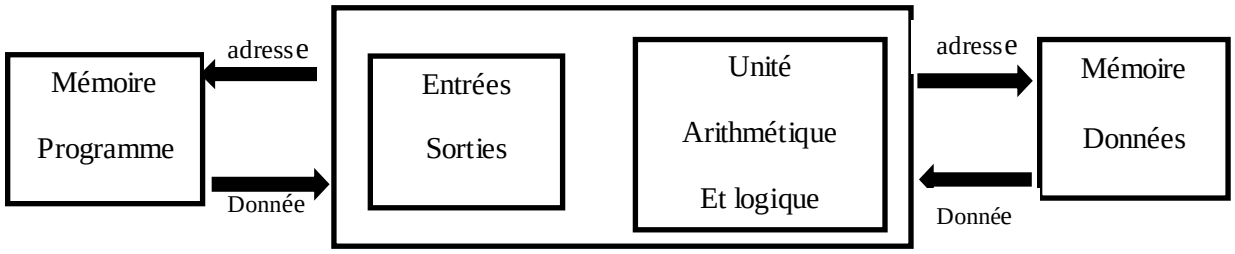


Figure (3.2) : Principe de l'architecture de Harvard

Remarque

Le choix d'un DSP pour une application particulière sera conditionné par le coût relatif du DSP par rapport au BOM (Bill of Matériel) de l'application, et de la complexité des opérations à effectuer. Il est clair qu'un DSP introduit dans un appareil grand public est plus souvent à virgule fixe. Si les coûts de développement d'une application en virgule fixe sont beaucoup plus importants qu'un design en virgule flottante, les volumes viendront amortir cette dépense... Il faut l'espérer

3.6 Programmation

Dans un appareil équipé de DSP, la vitesse d'exécution des calculs dans le DSP est généralement la partie déterminante de la vitesse d'exécution du travail effectué par la machine. Il s'agit souvent de programmes très courts, de quelques centaines de lignes au maximum dont certaines parties (boucles) doivent être optimisées au maximum.

C'est pourquoi beaucoup de programmeurs de DSP utilisent l'assembleur et analysent en détail le schéma d'exécution du code par la machine.

Cependant, certains DSP ont une architecture tellement complexe qu'il devient long et difficile pour le programmeur d'optimiser manuellement l'exécution. Il peut alors écrire son programme en C et laisser au compilateur le soin de réaliser l'optimisation du code.

S'il le souhaite, le programmeur pourra alors analyser le code généré par le compilateur et y apporter les dernières retouches permettant d'obtenir du DSP la meilleure performance possible [41].

Le choix entre ces deux langages se fera donc en fonction de la complexité du programme, de la vitesse de traitement souhaitée, du nombre de programmeurs qui travailleront sur le projet et du coût du produit. D'autres facteurs comme l'expérience personnelle et les outils que propose le constructeur doivent aussi être considérés.

3.7 Le DSP TMS320F2812

Le TMS320F2812 est un DSP contrôleur de 32 bit à virgule fixe avec la mémoire flash intégrée. L'unité centrale de traitement fonctionne avec un signal d'horloge de30MHz.

Le progrès de la microélectronique a permis d'avoir des calculateurs fonctionnant à de très grandes vitesses (150 millions d'opération par second), et d'une taille de quelques centimètres carrés. La génération des DSP C28x est efficace en exécutant des programmes en C et en C++, ce qui permet de développer des algorithmes de commande dans des langages de haut niveau. En plus de la mémoire flash, le F2812

embarque sur la même puce des périphériques utilisés pour la commande et la communication, telle qu'un gestionnaire d'évènement (générateur des signaux PWM), un convertisseur analogique numérique, des entrées / sorties numériques à usage général, et une interface de communication série [39].

3.7.1 Caractéristique du processeur

Le TMS320F2812 est un processeur de signaux en virgule fixe travaillant sur des mots de 32 bits. Ce dernier arrive de la famille C281x.Le DSP contient une unité centrale composée d'une unité arithmétique et logique dans laquelle la majorité des instructions s'exécute en un seul cycle d'horloge (6.67 ns), d'un multiplicateur hardware de 32x32 bits, d'un registre à décalage de 32 bits, et d'un accumulateur de 32 bits, de trois Timers de 32 bits.

Le DSP dispose aussi des registres auxiliaires qu'on utilise pour l'adressage indirect des données ou bien pour le stockage temporaire de celles-ci. Cette configuration de l'unité centrale nous permet l'implémentation des algorithmes les plus complexes utilisés dans le domaine de la commande [39].

3.7.2 Les interruptions

Une interruption est une rupture de séquence asynchrone (qui n'est pas synchronisée avec le déroulement normale du programme). Le TMS320F2812 possède 45 interruptions classées par ordre de priorité, rangé sur douze groupes (INT1, ….INT12) avec huit vecteurs masqué (INTx1…INTx8) à travers le registre IER. Ces interruptions seront générées par le software ou le hardware. Plus clairement chacune des périphériques du DSP peut générer une ou plusieurs interruptions en réponse à plusieurs événement, pour pouvoir effectuer un sous-programme à chaque interruption, elles ont était réunis selon leur source, ainsi à chaque périphérique correspond un vecteur d'interruption celui-ci peut être masqué (désactiver) ou bien branché à un sous-programme s'il n'est pas masquer (activer) [39].

3.8 Description du processeur

3.8.1 Description de la mémoire

Pour plus de rapidité et de flexibilité, le TMS320F2812 est basé sur une architecture Harvard modifié. Dans l'architecture Harvard conventionnelle les mémoires programme et data sont deux blocs différents, afin de permettre l'exécution des instructions et le transfert de donné simultanément. Tandis que ce processeur contient trois champs de mémoires, une mémoire programme pour les instructions, une autre (data) pour les variables et une troisième (entrée/sortie) pour la communication avec les différents périphériques. Cette configuration lui permet de faire trois transferts de donné en parallèle, en un seul cycle, tout en effectuant une opération arithmétique, aussi complexe soit elle [39].

3.8.2 Descriptions des Périphériques

a) Contrôleur de Réseau(CAN)

Le CAN est un module de 16 bits destiner au contrôle de l'environnement du DSP, il contient une boite aux lettres dans laquelle des informations peuvent être stocké et comparer à d'autre données transmises instantanément au processeur, par exemple la température [39].

b) Interface de Communication Série (SCI)

Le module SCI est conçu pour une communication digitale entre l'unité centrale de traitement et d'autres périphériques asynchrones utilisant le format non-return-to-zéro [39].

c) **Interface de Périphérique Séries (SPI)**

Le DSP contient quatre pins reliés au module SPI pour une communication à grande vitesse avec un port série synchrone. Ce module sert à la communication entre le DSP et des périphériques externes comme un autre convertisseur analogique / numérique, ou bien un autre processeur [39].

d) Gestionnaire d'Evénement (EVA et EVB)

Les DSP de la famille C28X sont équipés de deux modules identiques digestion d'événements appelés respectivement EVA et EVB lui permettant de générer 16 signaux PWM et aussi de détecter les transitions de 6 signaux logiques variables dans le temps provenant des codeurs incrémentaux par exemple [39].

e) La configuration des ports GPIO

Tous les E / S numériques sont regroupées dans des «ports», appelé GPIO-A, B, D, E, Fet G.GPIO« General Pur pose Input Output »signifie «général d'entrée-sortie fin ».

LeC28xest équipé d'autant d'unités internes. Tous ces pin sont relier aux ces unités (périphériques), donc pour les utilisé comme E / S général la solution et le multiplexage, soit on lui relier avec les périphériques interne soit on l'utilise comme des E / S général, ce multiplexage se fait avec le registre GPxMUX (avec x = A, B, D, E, Fet G).

Une fois on a choisi le mode E / S, il faut qu'on doit préciser la direction (entré ou sortie) et ça ce fait avec le registre GPxDIR (avec x = A, B, D, E, Fet G) [39].

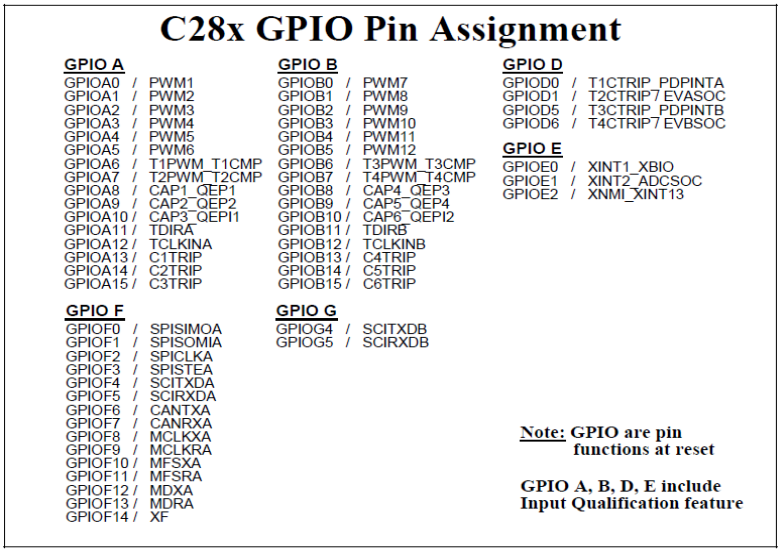


Figure (3.3) : Les entrées/sorties de GPIO

f) Les registres de GPIO

Le schéma synoptique de Les registres de GPIO est donné par la figure ci-dessous.

C28x GPIO MUX/DIR Registers

Address	Register	Name
70C0h	GPAMUX	GPIO A Mux Control Register
70C1h	GPADIR	GPIO A Direction Control Register
70C2h	GPAQUAL	GPIO A Input Qualification Control Register
70C4h	GPBMUX	GPIO B Mux Control Register
70C5h	GPBDIR	GPIO B Direction Control Register
70C6h	GPBQUAL	GPIO B Input Qualification Control Register
70CCh	GPDMUX	GPIO D Mux Control Register
70CDh	GPDDIR	GPIO D Direction Control Register
70CEh	GPDQUAL	GPIO D Input Qualification Control Register
70D0h	GPEMUX	GPIO E Mux Control Register
70D1h	GPEDIR	GPIO E Direction Control Register
70D2h	GPEQUAL	GPIO E Input Qualification Control Register
70D4h	GPFMUX	GPIO F Mux Control Register
70D5h	GPFDIR	GPIO F Direction Control Register
70D8h	GPGMUX	GPIO G Mux Control Register
70D9h	GPGDIR	GPIO G Direction Control Register

Figure (3.4) : Les registres GPXMUX de GPIO

Le registre GPXMUX est utilisé pour le choix de mode E/S ou mode périphérique.

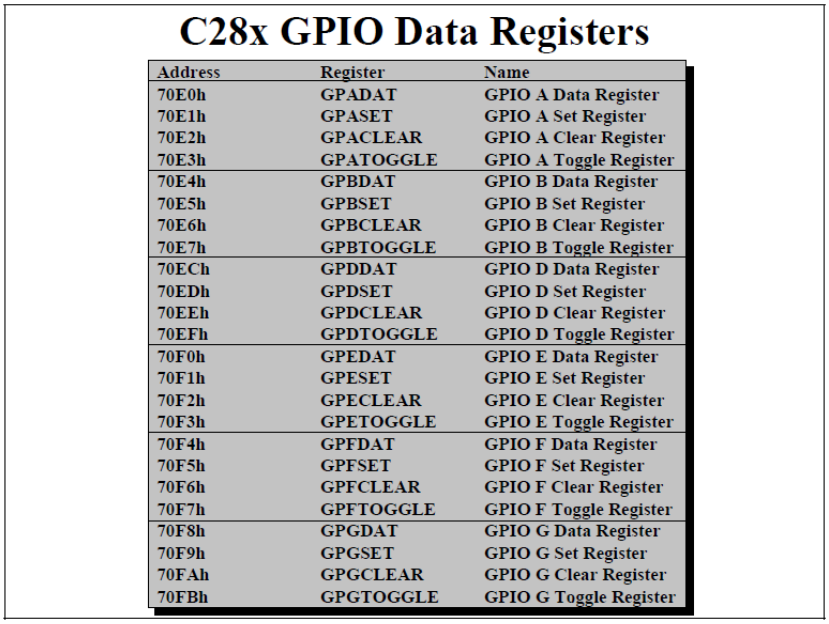
- SiGPXMUX.bit=0, alors la broche est configurée en mode E/S.
- SiGPXMUX.bit=1, alorslabrocheest reliée en mode périphérique.

Le registre GPXDIR est utilisé pour le choix soit comme entrée ou comme sortie.

Chaque port d'E / S dispose d'un registre de contrôle de direction. Le registre direction contrôle si le correspondant broche I/O est configurée comme une entrée ou une sortie. A réinitialiser tous les broches GP E/S sont configurées comme des entrées [39].

- SiGPxDIR.bit=0, alorslabrocheest configurée en tant qu'entrée.
- SiGPxDIR.bit=1, alorslabrocheest configurée en tant que sortie.

Le schéma synoptique de registre GPXDAT est utilisé pour l’écriture dans la sortie est donné par la figure ci-dessous.



C28x GPIO Data Registers

Address	Register	Name
70E0h	GPADAT	GPIO A Data Register
70E1h	GPASET	GPIO A Set Register
70E2h	GPACLEAR	GPIO A Clear Register
70E3h	GPATOGGLE	GPIO A Toggle Register
70E4h	GPBDAT	GPIO B Data Register
70E5h	GPBSET	GPIO B Set Register
70E6h	GPBCLEAR	GPIO B Clear Register
70E7h	GPBTOGGLE	GPIO B Toggle Register
70ECh	GPDDAT	GPIO D Data Register
70EDh	GPDSET	GPIO D Set Register
70EEh	GPDCLEAR	GPIO D Clear Register
70EFh	GPDTOGGLE	GPIO D Toggle Register
70F0h	GPEDAT	GPIO E Data Register
70F1h	GPESET	GPIO E Set Register
70F2h	GPECLEAR	GPIO E Clear Register
70F3h	GPETOGGLE	GPIO E Toggle Register
70F4h	GPFDAT	GPIO F Data Register
70F5h	GPFSET	GPIO F Set Register
70F6h	GPFCLEAR	GPIO F Clear Register
70F7h	GPFTOGGLE	GPIO F Toggle Register
70F8h	GPGDAT	GPIO G Data Register
70F9h	GPGSET	GPIO G Set Register
70FAh	GPGCLEAR	GPIO G Clear Register
70FBh	GPGTOGGLE	GPIO G Toggle Register

Figure (3.5) : Les registres GPXDATA de GPIO

Chaque port d'E / S dispose d'un registre de données. Le registre de données est un registre R/W qui reflète l'état actuel de l'entrée I/Ode signal après la qualification. Ecrit à l'état du registre correspondant de n'importe quel signal d'entrée / sortie qui est configuré comme.

- Si GPxDAT.bit=0, et la broche est une sortie, la broche retiré en bas
- Si GPxDAT.bit= 1, et la broche est une sortie, la broche retirez en haute

Le registre GPXSET est utilisé pour fixer le registre à sa valeur.

Le registre GPXCLEAR est utilisé pour effacé le contenu de registre.

3.9 Présentation de la carte « eZdsp TMF2812 »

Le DSP Starter Kit est un module de développement conçu par la société Spectrum Digital, sur lequel est implanté le processeur de signaux TMS320F2812.

Ce module servira à l'implantation des programmes de plusieurs applications. La figure (3.8) montre la photo du starter kit eZdsp F2812 [39].

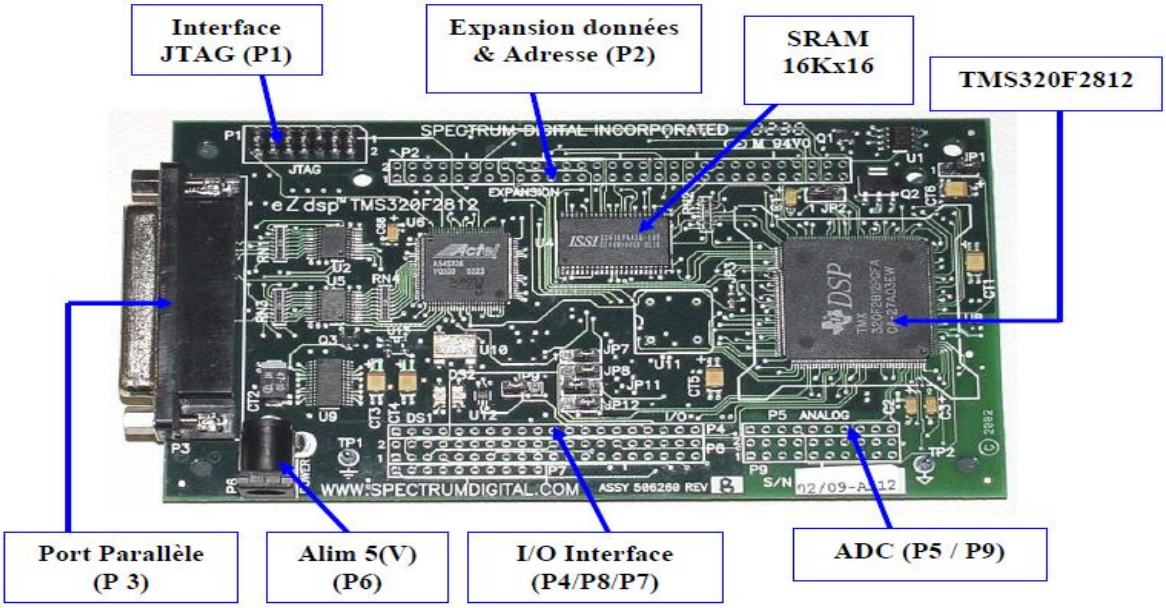


Figure (3.6) : Les connecteurs sur la carte DSP

Le module est équipe de 5 connecteurs qui sont :

- **P1 :** Interface JTAG : c'est une interface standard qui est utilisée par les émulateurs JTAG pour interfacer les DSP de Texas Instruments.
- **P2 :** port d'expansion.
- **P3 :** Port parallèle pour le control de l'interface JTAG : cette interface possède un port parallèle standard qui peut supporter les communications bidirectionnelles de type ECP, EPP, et SPP8.
- **P6 :** Connecteur pour l'alimentation en 5V DC de la carte.
- **P4, P8, P7 :** Ce sont trois connecteurs qui permettent un accès direct aux pins d'entrée/sortie du DSP
- **P5, P9 :** Ce sont deux connecteurs qui permettent d'accéder aux pins des entrées analogiques pour les conversions analogiques numériques.

3.10 Outils de développement logiciel du DSP

L'outil de développement du logiciel retenu est tiré de l'Environnement de Développement Intégré : «Code Composer studio» (CCS).

3.10.1 Logiciel code composé studio :

Ce logiciel Code Composer studio (CCS) fournit donc plusieurs outils pour faciliter la construction et la mise au point des programmes utilisant les DSP. Il comprend un éditeur code source, un compilateur de langage c/c++, un assembleur de code, un éditeur de liens, et un environnement d'exécution permettant de télécharger

un programme exécutable sur une carte cible, de l'exécuter et de le déboguer au besoin.

Le CCS comprend aussi des outils qui permettent l'analyse en temps réel d'un programme en cours d'exécution et de donner la structuration des résultats produits. Finalement, il fournit un environnement de gestion de fichiers, qui facilite la construction et la mise au point des programmes [42].

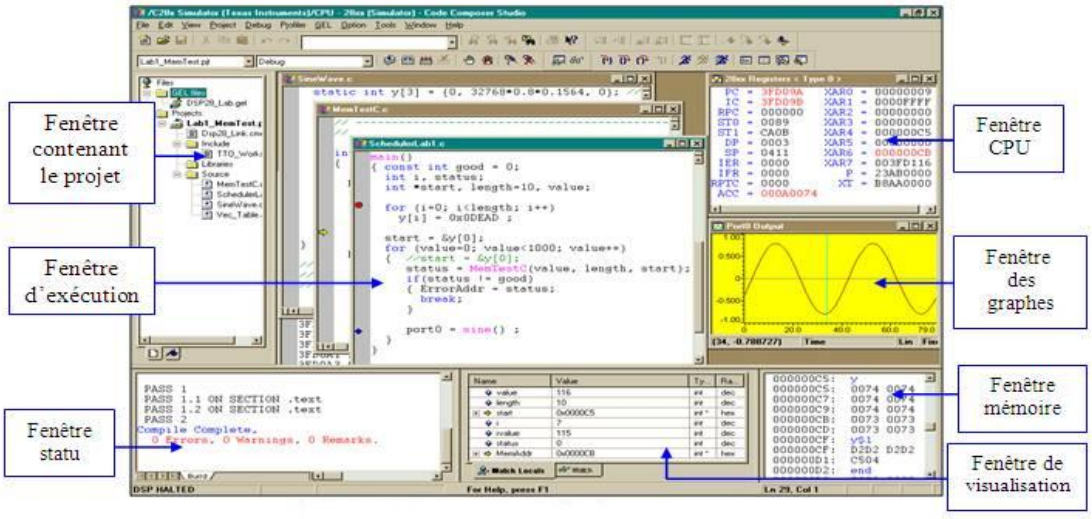


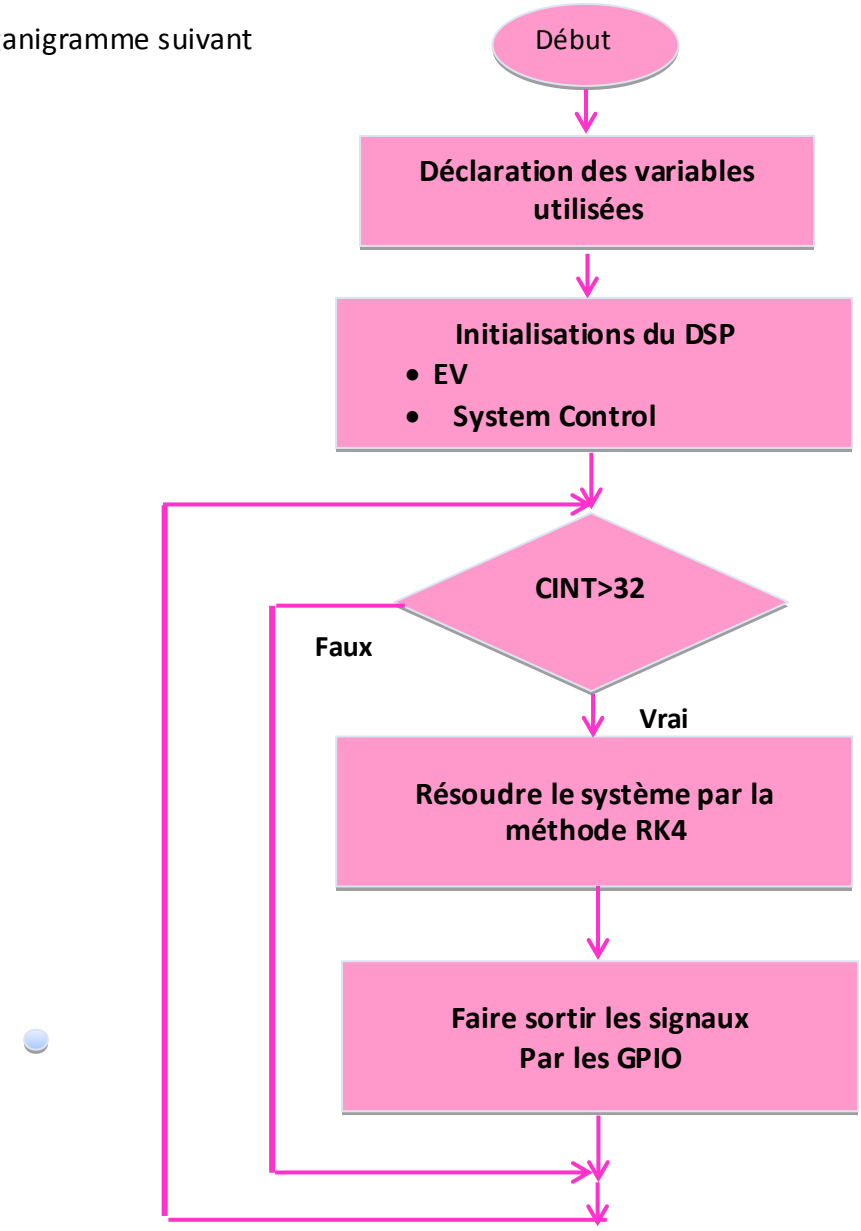
Figure (3.7) : La fenêtre de l'outil de développement CCS

La carte est équipée de deux diodes électroluminescente :
La première diode DS1 indique la présence de la tension continue du 5 volts.
La diode DS2 indique l'état du bit XF du DSP (la diode s'allume lorsque ce bit est à
L'état logique haut).

3.11 Implémentation de la méthode RK4 appliquée au système de Rössler, Lorenz et Lu sur la carte DSP

L'implémentation de la méthode RK4 appliquée aux systèmes de Rössler, Lorenz et Lu sur la carte DSP sera faite sur plusieurs étapes. En commence par effectuer le programme en langage c/c++. La deuxième étape consiste à créer un projet en l'environnement de développement Code Composer Studio (CCS V3.1) contient le programme c/c++ de la méthode RK4, La dernière étape implantée le programme sur la carte DSP pour visualiser les signaux. On résume l'algorithme de l'implémentation

de la méthode RK4 appliquée au système chaotique ((Rössler, Lorenz et Lu)) dans le DSP par l'organigramme suivant



Figure(3.8) :L'organigramme de la réalisation de l'implémentation

- **Déclaration des variables utilisés :** dans cette étape on doit déclarer tous les variables utilisés dans le programme.
- **Initialisations du DSP :** Dans cette étape les périphériques utilisés sont configurés et les périphériques qui ne sont pas utilisés, doivent être désactivés. Les interruptions utilisées doivent être aussi activées.
  - ❖ **La configuration de gestionnaire d'événement :** dans cette étape on doit faire **la** configuration de GPTimer1 pour crée une interruption et pour choisir la fréquence de l'incrémentation des Timers.

- **La configuration du système de contrôle :** on sait que le DSP est entrainé par un oscillateur externe de fréquence de 30MHZ, pour atteindre une fréquence plus grand on passe par le module PLL (multiplicateur de fréquence). La fréquence à la sortie de la PLL qu'on a choisi est 150MHZ. Puis on doit configurer la fréquence HSPCLK, on obtient à la fin une horloge de 75MHZ.

- **La précision de la fréquence d'échantillonnage :** on sait que le Timer1 nous fournit une interruption chaque fois qu'il atteint la valeur période, d'autre part le Timer1 s'incrémente avec une fréquence HSPCKL égale à 75MHZ, la valeur de la période qu'on a choisi est égale 586, on a fait un compteur d'interruption CINT, pour obtenir une période d'échantillonnage a 1 ms.

$\frac{75000000/4}{586}$ ⟶ $3.12533*10^{-5}$ ⟶ 1 interruption

1000 HZ ⟶ 1 ms ⟶ 32 interruptions

Donc notre programme doit être exécuté chaque fois le CINT compte 32 interruptions.

- **Calcul les paramètre de la méthode RK4**

Calcul des k_{jx}, k_{jy} , k_{jz} ; $1 \leq j \leq 4$.

Calcul des x_{i+1} , y_{i+1} , z_{i+1} .

- **Faire sortir les signaux par les GPIO**

Dans cette étape on doit configurer les entrées et les sorties, pour cela on agit sur les registres suivants :

```
GpioMux Regs.GPAMUX.all=0x0 ;          // mode entré\sortie
GpioMux Regs.GPADIR.all = 0xFFFF ;     // configuration comme des sorties
GpioMux Regs.GPADATA.all = X / Y / Z   // écriture dans les pins de sortie
```

3.12 Conclusion

Dans ce chapitre nous avons présenté quelques généralités sur la carte DSP, comme les caractéristiques principales des DSP (architecture des DSP, les types de DSP, vitesse) et la carte DSP TMS320F2812. Puis, un aperçu sur le logiciel de la carte DSP, Code Composer Studio (CCS) a été donné. Après, nous avons présenté la partie d'implémentation de la méthode RK4 appliquée aux systèmes de Rössler Lorenz et Lu sur la carte DSP TMS320F2812.

Dans le chapitre suivant nous allons montrer tous les résultats de simulation et de l'implémentation pratique, afin de les comparer.

Chapitre 4 Résultats et commentaires

4.1 Introduction

L'implémentation de la génération des signaux chaotiques fournis par les systèmes de Rössler, Lorenz et Lù sera faite sur plusieurs étapes. On commence par le résoudre des systèmes par la méthode numérique (Rang kutta) puis on écrit le programme sous matlab et en langage C, finalement on termine par une application sur la carte DSP. On utilisant ce Logiciel Code Composer Studio(CCS). Dans ce chapitre nous présentons et appliquons la méthode de Range kutta sur les systèmes (Rossler, Lorenz et Lù) et représentons les résultats de la simulation sous Matlab et les résultats pratiques de l'implémentation des systèmes.

4.2 Méthodes numérique de Résolutions des équations différentielles non linéaires

Les équations différentielles constituent un type d'équations qui trouvent de nombreuses applications dans la modélisation des systèmes physiques. Elles jouent un rôle fondamental dans la théorie des systèmes asservis linéaires et non linéaires. C'est pourquoi, il est important de savoir établir les équations différentielles, les résoudre et analyser leurs solutions.

Dans le cas d'équations différentielles non linéaires, sauf pour quelques cas considérés comme particulier, on passe nécessairement à une résolution numérique.

Il existe plusieurs méthodes numériques pour résoudre ces équations :

Dans ce qui suit on s'intéressera à la méthode de Runge Kutta .

4.2.1 La méthode de Runge-Kutta

Les méthodes de Runge-Kutta sont des méthodes d'analyse numérique d'approximation de solutions d'équations différentielles. Ces méthodes reposent sur le principe de l'itération, c'est-à-dire qu'une première estimation de la solution est utilisée pour calculer une seconde estimation, plus précise, et ainsi de suite. Il existe plusieurs façons d'appliquer cette méthode que l'on différencie grâce à « L'ordre d'application ». On obtient ainsi (de la méthode la moins précise à la plus précise) [43]

Dans ce qui suit-on s'intéressera La méthode de Runge Kutta d'ordre 4 (RK-4)

4.2.1.1 Runge-Kutta d'ordre 4 (RK-4)

C'est un cas particulier d'usage très fréquent, noté RK4.Considérons le problème suivant :

$$\dot{y} = \mathcal{F}(\mathrm{x},\mathrm{y}) \tag{4.1}$$

Avec la condition : $\mathrm{y}(x_0) = y_0$.

La méthode RK4 est donnée par l'équation :

$$\mathrm{y}_{n+1} = \mathrm{y}_n + \frac{1}{6} * (k_1 + 2 * k_2 + 2 * k_3 + k_4) \tag{4.2}$$

ou

$$\begin{cases} k_1 = \mathcal{F}(x,y) \\ k_2 = \mathcal{F}\left(x + \frac{h}{2}, y + \frac{h}{2} * k_1\right) \\ k_3 = \mathcal{F}\left(x + \frac{h}{2}, y + \frac{h}{2} * k_2\right) \\ k_4 = \mathcal{F}(x + h, y + h * k_3) \end{cases} \tag{4.3}$$

- $\boldsymbol{h}$**:** Le pas de discrétisation en x (pas de calcul).
- $\boldsymbol{k_1}$: est la pente au début de l'intervalle ;
- $\boldsymbol{k_2}$: est la pente au milieu de l'intervalle, en utilisant la pente $\boldsymbol{k_1}$ pour calculer la valeur de y au point $t_n + \frac{h}{2}$ par le biais de la méthode d'Euler ;
- $\boldsymbol{k_3}$: est de nouveau la pente au milieu de l'intervalle, mais obtenue cette fois en utilisant la pente $\boldsymbol{k_2}$ pour calculer y ;
- $\boldsymbol{k_4}$: est la pente à la fin de l'intervalle, avec la valeur de y calculée en utilisant $\boldsymbol{k_3}$.

Dans la moyenne des quatre pentes, un poids plus grand est donné aux pentes au point milieu. $\text{Pente} = \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$.

La méthode RK4 est une méthode d'ordre 4, ce qui signifie que l'erreur commise à chaque étape est de l'ordre de h^5, alors que l'erreur totale accumulée est de l'ord re de h^4 [43].

4.2.2 Algorithme de la méthode Runge Kutta 4

C'est un algorithme itératif qui nécessite la connaissance des valeurs initiales des variables. Les valeurs des variables au pas $i+1$ sont déterminées de celles du pas i, moyennant le calcul de quatre termes intermédiaires k_1, k_2, k_3, k_4 .

1. Pour commencer, on évalue la dérivée $\frac{k_1}{h}$ au point (x_n, t_n).
2. On utilise cette dérivée pour obtenir un premier point médian $\frac{x_n + k_1}{2} = k_2$
3. On calcule ensuite la dérivée $\frac{k_2}{h}$ à ce point médian.
4. On calcule une deuxième estimation $\frac{x_n + k_2}{2} = k_3$ de ce point médian et on y calcule encore une fois la dérivée $\frac{k_3}{h}$
5. On calcule ensuite la dérivée $\frac{k_4}{h}$ à une première estimation du point final
6. Enfin, le point final estimé par la méthode est obtenu par une combinaison des quatre dérivées calculées aux étapes précédentes [43].

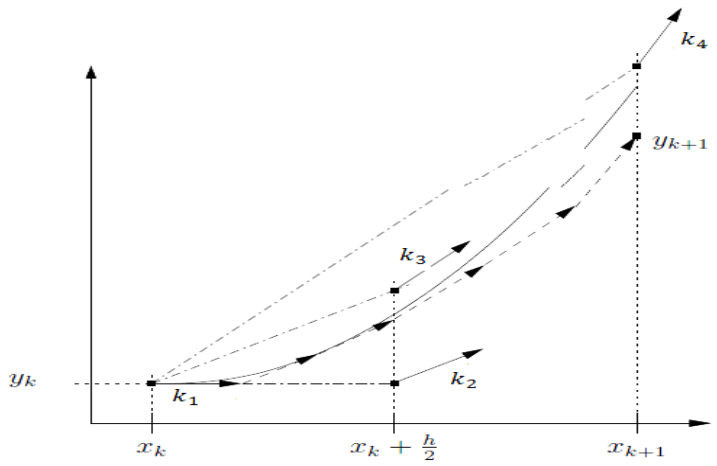


Figure (4.1) *:* Schéma de Runge Kutta d'ordre 4

4.2.3 Organigramme général : La simulation nécessite la connaissance des paramètres du système et le pas d'intégration $\boldsymbol{h}$, et la condition d'arrêt. Le pas d'intégration est un paramètre important. Généralement, il agit sur la précision de la méthode. Il est choisi petit et les résultats du calcul stockés dans des vecteurs, à la fin, on trace ces vecteurs.

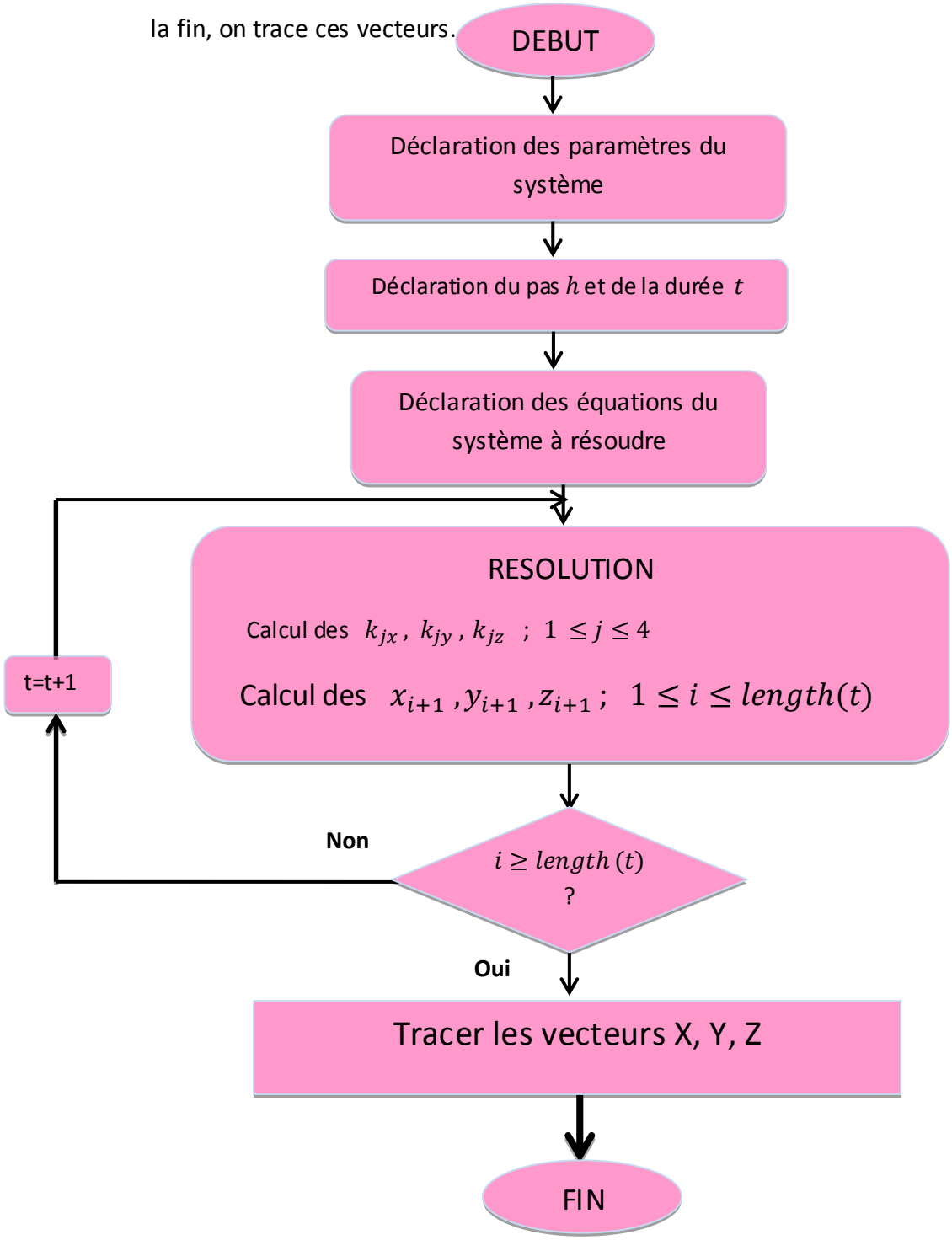


Figure (4.2) : Organigramme de la méthode RK4 appliqué sur le système non linéaire.

4.2.4 Exemples des systèmes chaotiques

4.2.4.1 Le système de Rössler

Le système de Rössler composé par le système d'équations différentielles non linéaires suivant :

$$\begin{cases} \dfrac{dx}{dt} = -y - z \\ \dfrac{dy}{dt} = x + ay \\ \dfrac{dz}{dt} = b + z\,x - z\,c \end{cases} \tag{4.4}$$

Où x, y et z sont des variables d'état du système, a, b et c sont des paramètres réels.

Les paramètres et les conditions initiales de système ont été choisis de la manière suivante : a = 0.2 ; b = 0.2 ; c = 5.7 avec $(x_0, y_{0,} z_0)$ = (0, 1, 2).

Pour appliquer la méthode de Runge Kutta d'ordre 4 on choisir $h = 0.01$ et on calcule les paramètres : k_1 , k_2 , k_3 , k_4 ; x_{i+1} , z_{i+1} , y_{i+1} .

Pour k_1 :

$$\begin{cases} k_{1x} = h * dx(x,y,z) \\ k_{1y} = h * dy(x,y,z\ ) \\ k_{1z} = h * dz(x,y,z) \end{cases} \Longrightarrow \begin{cases} k_{11} = h * (-y - z) \\ k_{12} = h * (x + a * y) \\ k_{13} = h * (b + (z * x) - c * z) \end{cases} \tag{4.5}$$

Pour k_2 :

$$\begin{cases} k_{2x} = h * dx(x + 0.5 * k_{1x}, y + 0.5 * k_{1y}, z + 0.5 * k_{1z}) \\ k_{2y} = h * dy(x + 0.5 * k_{1x}, y + 0.5 * k_{1y}, z + 0.5 * k_{1z}) \\ k_{2z} = h * dz(x + 0.5 * k_{1x}, y + 0.5 * k_{1y}, z + 0.5 * k_{1z}) \end{cases} \tag{4.6}$$

Alors

$$\begin{cases} k_{21} = h * \big(-(y + k12 * 0.5) - (z + 0.5 * k13)\big) \\ k_{22} = h * (x + 0.5 * k11) + a * (y + 0.5 * k12)) \\ k_{23} = h * (\mathrm{b} + \big((z + 0.5 * k13) * (x + 0.5 * k11)\big) - (c * (z + k13 * 0.5))) \end{cases} \tag{4.7}$$

Pour k_3 :

$$\begin{cases} k_{3x} = h * dx(x + 0.5 * k_{2x}, y + 0.5 * k_{2y}, z + 0.5 * k_{2z}) \\ k_{3y} = h * dy(x + 0.5 * k_{2x}, y + 0.5 * k_{2y}, z + 0.5 * k_{2z}) \\ k_{3z} = h * dz(x + 0.5 * k_{2x}, y + 0.5 * k_{2y}, z + 0.5 * k_{2z}) \end{cases} \tag{4.8}$$

Alors

$$\begin{cases} k_{31} = h * \left(-(y + k21 * 0.5) - (z + 0.5 * k21)\right) \\ k_{32} = h * (x + 0.5 * k21) + a * (y + 0.5 * k22)) \\ k_{33} = h * (\mathrm{b} + \left((z + 0.5 * k23) * (x + 0.5 * k21)\right) - (c * (z + k23 * 0.5))) \end{cases} \quad (4.9)$$

Pour k_4 :

$$\begin{cases} k_{4x} = h * dx(x + k_{3x}, y + k_{3y}, z + k_{3z}) \\ k_{4y} = h * dy(x + k_{3x}, y + k_{3y}, z + k_{3z}) \\ k_{4z} = h * dz(x + k_{3x}, y + k_{3y}, z + k_{3z}) \end{cases} \quad (4.10)$$

Alors

$$\begin{cases} k_{41} = h * \left(-(y + k32 * 0.5) - (z + 0.5 * k33)\right) \\ k_{42} = h * (x + 0.5 * k31) + a * (y + 0.5 * k32)) \\ k_{43} = h * (\mathrm{b} + \left((z + 0.5 * k33) * (x + 0.5 * k31)\right) - (c * (z + k33 * 0.5))) \end{cases} \quad (4.11)$$

Donc :

$$x_{i+1} = x_i + \frac{(k_{1x} + 2 * k_{2x} + 2 * k_{3x} + k_{4x})}{6}$$

$$y_{i+1} = y_i + \frac{(k_{1y} + 2 * k_{2y} + 2 * k_{3y} + k_{4y})}{6}$$

$$z_{i+1} = z_i + \frac{(k_{1z} + 2 * k_{2z} + 2 * k_{3z} + k_{4z})}{6}$$

4.2.4.2 Le système de Lorenz :

Les équations de ce système sont les suivantes

$$\begin{cases} \dfrac{dx}{dt} = a(y - x) \\ \dfrac{dy}{dt} = bx - y - x * z \\ \dfrac{dz}{dt} = x * y - c * z \end{cases} \quad (4.12)$$

Ou x, y et z sont des variables d'état du système, a, b et c sont des paramètres réels.

Les paramètres et les conditions initiales de système ont été choisis de la manière suivante : a = 10 ; b = 28 ; c = 8/3 avec $(x_0, y_{0,} z_0)$ = (0, 1, 2).

Pour appliquer la méthode de Runge kutta d'ordre 4 on choisir $h = 0.01$ et on calcule les paramètres : k_1 , k_2 , k_3 , k_4 ; x_{i+1} , z_{i+1} , y_{i+1} .

k_1 :

$$\begin{cases} k_{11} = h * \left(a * (y - x)\right) \\ k_{12} = h * ((b * x) - y - (x * z)) \\ k_{13} = h * \left((x * y) - (c * z)\right) \end{cases} \quad (4.13)$$

k_2

$$\begin{cases} k_{21} = h * (a * (y + 0.5 * k12) - (x + 0.5 * k11)) \\ k_{22} = h * ((b * (x + 0.5 * k11) - (y + 0.5 * k12) - (x + 0.5 * k11) * (z + 0.5 * k13)) \\ k_{23} = h * \Big((x + 0.5 * k11) * (y + 0.5 * k12) - \big(c * (z + 0.5 * k13)\big)\Big) \end{cases} \quad (4.14)$$

k_3 :

$$\begin{cases} k_{31} = h * (a * (y + 0.5 * k22) - (x + 0.5 * k21)) \\ k_{32} = h * ((b * (x + 0.5 * k21) - (y + 0.5 * k22) - (x + 0.5 * k21) * (z + 0.5 * k23)) \\ k_{33} = h * \Big((x + 0.5 * k21) * (y + 0.5 * k22) - \big(c * (z + 0.5 * k23)\big)\Big) \end{cases} \quad (4.15)$$

k_4 :

$$\begin{cases} k_{41} = h * (a * (y + 0.5 * k32) - (x + 0.5 * k31)) \\ k_{42} = h * ((b * (x + 0.5 * k31) - (y + 0.5 * k32) - (x + 0.5 * k31) * (z + 0.5 * k33)) \\ k_{43} = h * \Big((x + 0.5 * k31) * (y + 0.5 * k32) - \big(c * (z + 0.5 * k33)\big)\Big) \end{cases} \quad (4.16)$$

Donc :

$$x_{i+1} = x_i + \frac{(k_{1x} + 2 * k_{2x} + 2 * k_{3x} + k_{4x})}{6}$$

$$y_{i+1} = y_i + \frac{(k_{1y} + 2 * k_{2y} + 2 * k_{3y} + k_{4y})}{6} \quad (4.16)$$

$$z_{i+1} = z_i + \frac{(k_{1z} + 2 * k_{2z} + 2 * k_{3z} + k_{4z})}{6}$$

4.2.4.3 Le système de Lu :

Les équations de ce système sont les suivantes

$$\begin{cases} \frac{dx}{dt} = a(y - x) \\ \frac{dy}{dt} = -x * z + c * y \\ \frac{dz}{dt} = x * y - b * z \end{cases} \quad (4.17)$$

Ou x, y et z sont des variables d'état du système, a, b et c sont des paramètres réels.

Les paramètres et les conditions initiales de système ont été choisis de la manière suivante :a = 36 ; b = 3 ; c = 20 avec (x_0, y_0, z_0) = (0, 1, 2) .
Pour appliquer la méthode de Runge kutta d'ordre 4 on choisir $h = 0.01$ et on calcule les paramètres : k_1 , k_2 , k_3 , k_4 ; x_{i+1} , z_{i+1} , y_{i+1} .

Pour k_1 :

$$\begin{cases} k_{11} = h * \big(a(y - x)\big) \\ k_{12} = h * ((-x * z) + (c * y)) \\ k_{13} = h * ((x * y) - (b * z)) \end{cases} \quad (4.18)$$

Pour k_2 :

$$\begin{cases} k_{21} = h * \big(a(y + 0.5 * k12) - (x + 0.5 * K11)\big) \\ k_{22} = h * \big((-(x + 0.5 * k11) * (z + 0.5 * k13) + (c * (y + 0.5 * k12)\big) \\ k_{23} = h * \Big(\big((x + 0.5 * k11) * (y + 0.5 * k12)\big) - \big(b * (z + 0.5 * k13)\big)\Big) \end{cases} \quad (4.1)$$

Pour k_3 :

$$\begin{cases} k_{31} = h * \big(a(y + 0.5 * k22) - (x + 0.5 * K21)\big) \\ k_{32} = h * \big((-(x + 0.5 * k21) * (z + 0.5 * k23) + (c * (y + 0.5 * k22)\big) \\ k_{33} = h * \Big(\big((x + 0.5 * k21) * (y + 0.5 * k22)\big) - \big(b * (z + 0.5 * k23)\big)\Big) \end{cases} \quad (4.20)$$

Pour k_4 :

$$\begin{cases} k_{41} = h * \big(a(y + 0.5 * k32) - (x + 0.5 * K31)\big) \\ k_{42} = h * \big((-(x + 0.5 * k31) * (z + 0.5 * k33) + (c * (y + 0.5 * k32)\big) \\ k_{43} = h * \Big(\big((x + 0.5 * k31) * (y + 0.5 * k32)\big) - \big(b * (z + 0.5 * k33)\big)\Big) \end{cases} \quad (4.21)$$

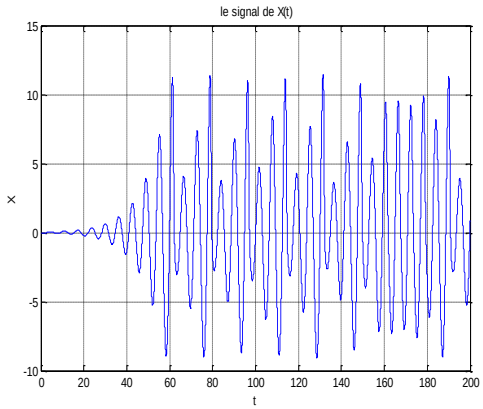
Donc :

$$\begin{aligned} x_{i+1} &= x_i + \frac{(k_{1x} + 2 * k_{2x} + 2 * k_{3x} + k_{4x})}{6} \\ y_{i+1} &= y_i + \frac{(k_{1y} + 2 * k_{2y} + 2 * k_{3y} + k_{4y})}{6} \\ z_{i+1} &= z_i + \frac{(k_{1z} + 2 * k_{2z} + 2 * k_{3z} + k_{4z})}{6} \end{aligned} \quad (4.22)$$

4.3 Résultats de la simulation sous Matlab

4.3.1 La simulation du système de Rossler sous Matlab/Programme :

Les figures [(4.3) (4.4) (4.5) (4.6)] représentent les résultats de la méthode de Runge Kutta d'ordre 4 **appliquée au** système de**s** Rössler avec les conditions initiales de système ont été choisis de la manière suivante : a = 0.2 ; b = 0.2 ; c = 5.et h=0.01.



Figure(4.3): Signal x (t).

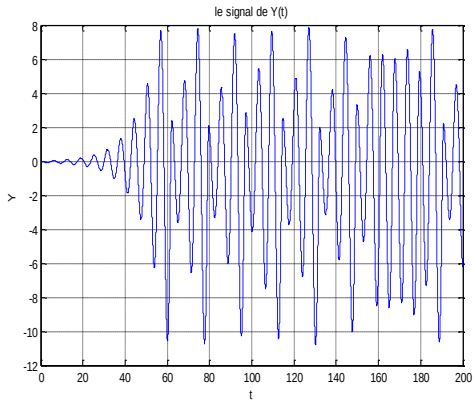


Figure (4.4) : Signal y (t).

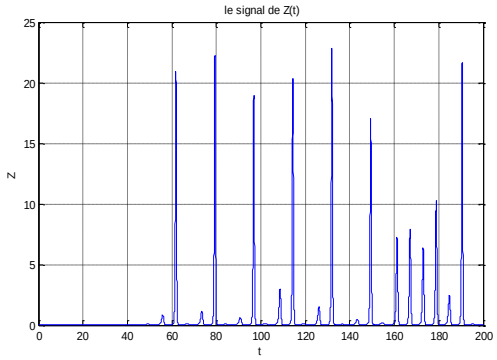


figure ((4.5) : signal z(t)

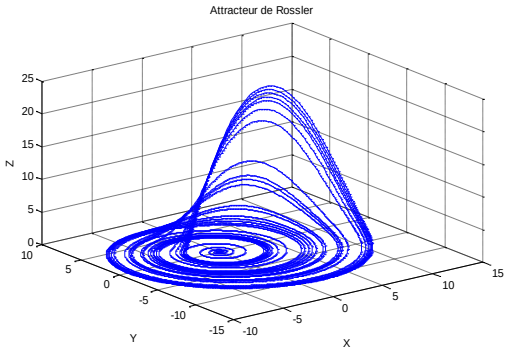


Figure (4.6) : Attracteur de Rössler.

4.3.2 La simulation du système de Rossler sous Matlab/Simulink

Les figures [(4.7) (4.8) (4.9) (4.10)] représentent les résultats de la simulation du système de Rossler sous Matlab/Simulink7 avec $(x_0, y_{0,} z_0)$ = (0, 0, 0).

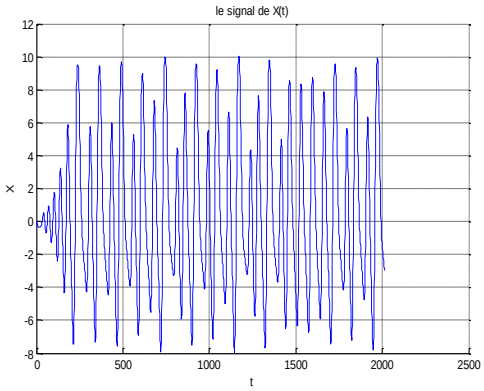


Figure (4.7) : Signal x (t)

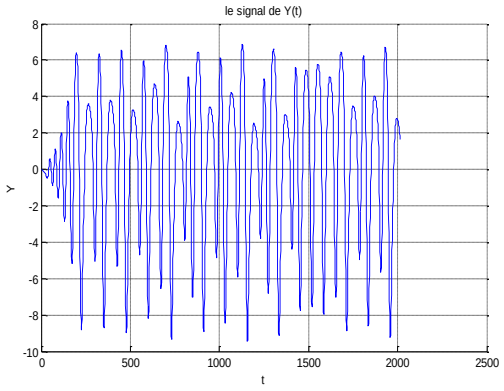
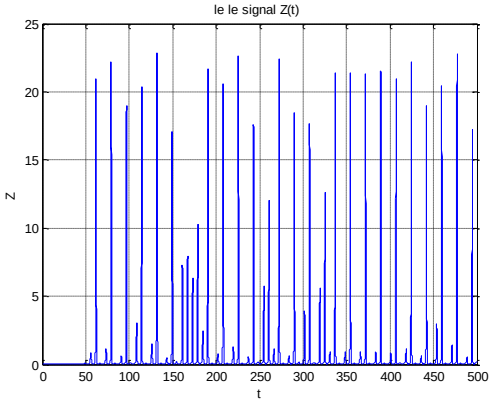


Figure (4.8) : Signal y (t)



Figure(4.9) : Signal z (t)

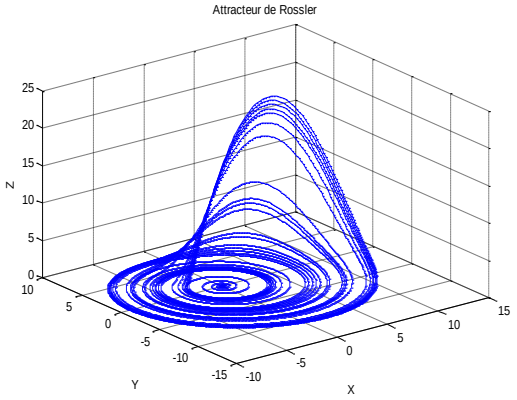


Figure (4.10): Attracteur rossler

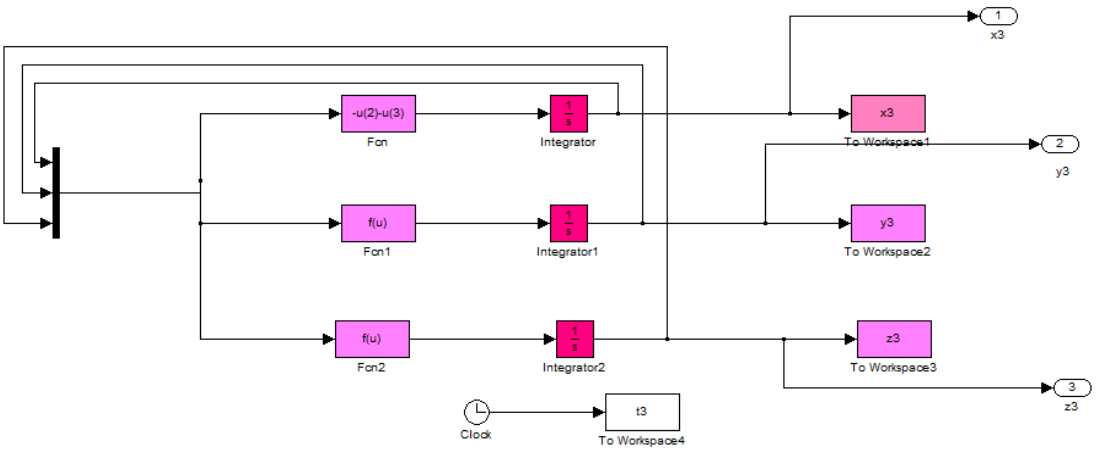


Figure (4.11) *:* Schéma du systèm

e de Rössler SOUS Matlab/Simulink.

4.3.3 la simulation du système de Lorenz sous Matlab/Programme :

Les figures [(4.12) (4.13) (4.14) (4.15)] représentent les résultats de la méthode de Runge Kutta d'ordre 4 appliquée aux systèmes des Lorenz avec h=0.01 programmée sous Matlab.

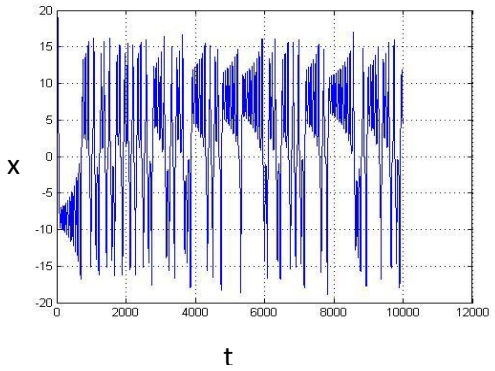


Figure 4.12 : Signal x (t)

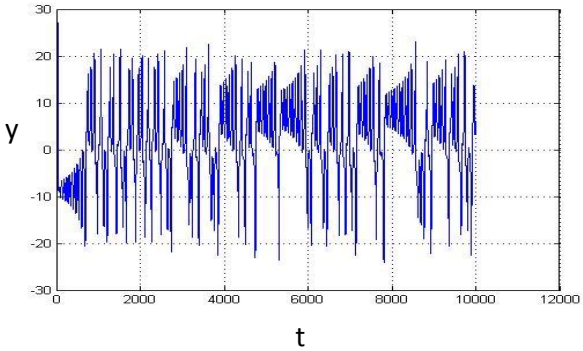


Figure4.13 : Signal y (t)

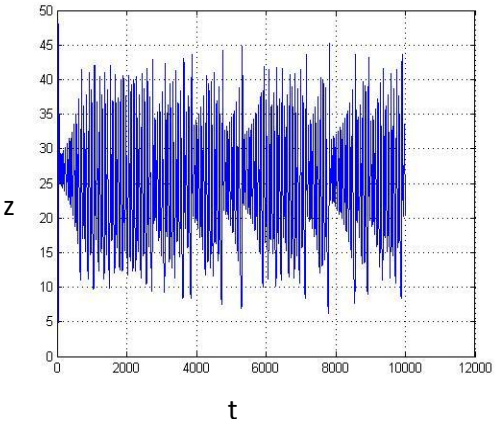


Figure 4.14 : Signal z(t)

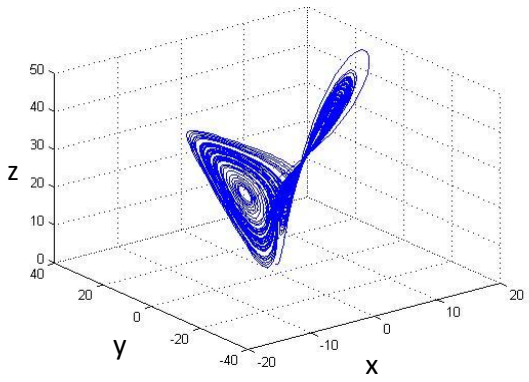


Figure 4.15: Attracteur Lorenz

4.3.4 La simulation du système de Lorenz sous Matlab/Simulink

Les figures [(4.16) (4.17) (4.18) (4.19)] représentent les résultats de la Simulation du système de Lorenz sous Matlab/Simulink avec les conditions initiales de système ont été choisis de la manière suivante $(x_0, y_{0,} z_0)$ = (0.05, 1, 3).

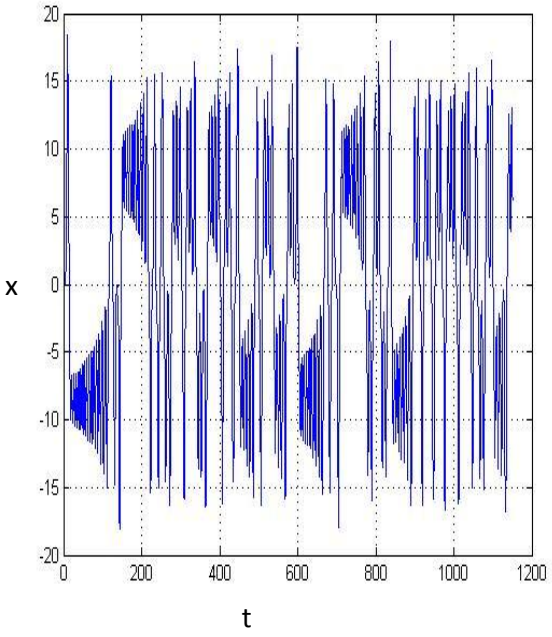


Figure (4.16) : Signal x (t)

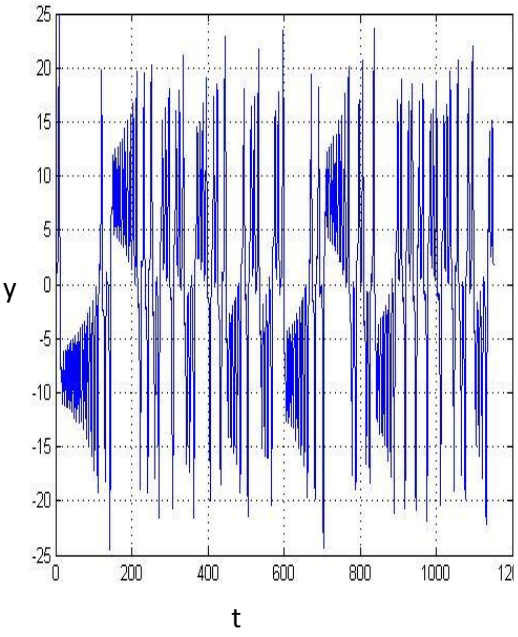


Figure (4.17) : Signal y (t)

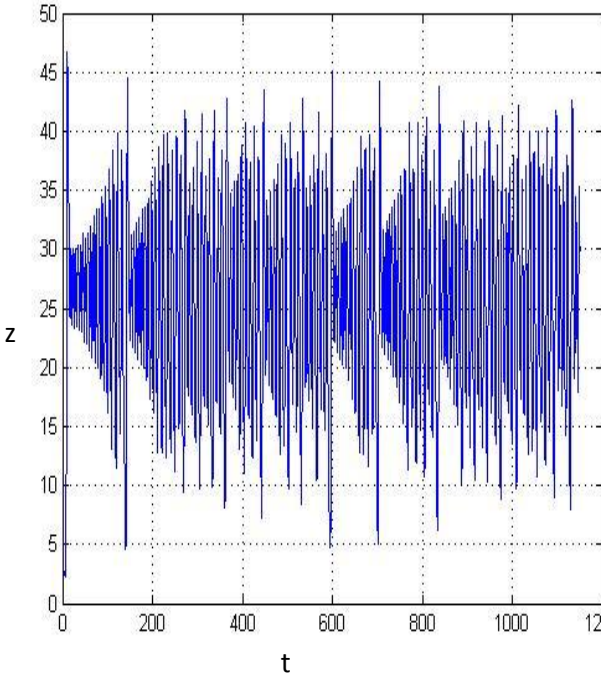


Figure (4.18) : Signal z (t)

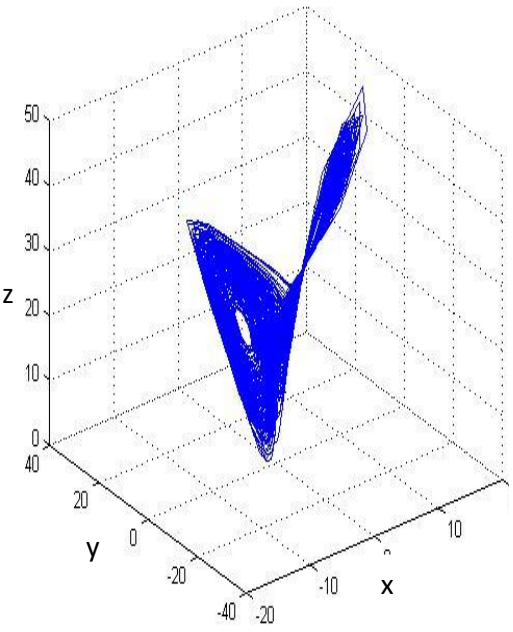


Figure (4.19) : Attracteur Lorenz.

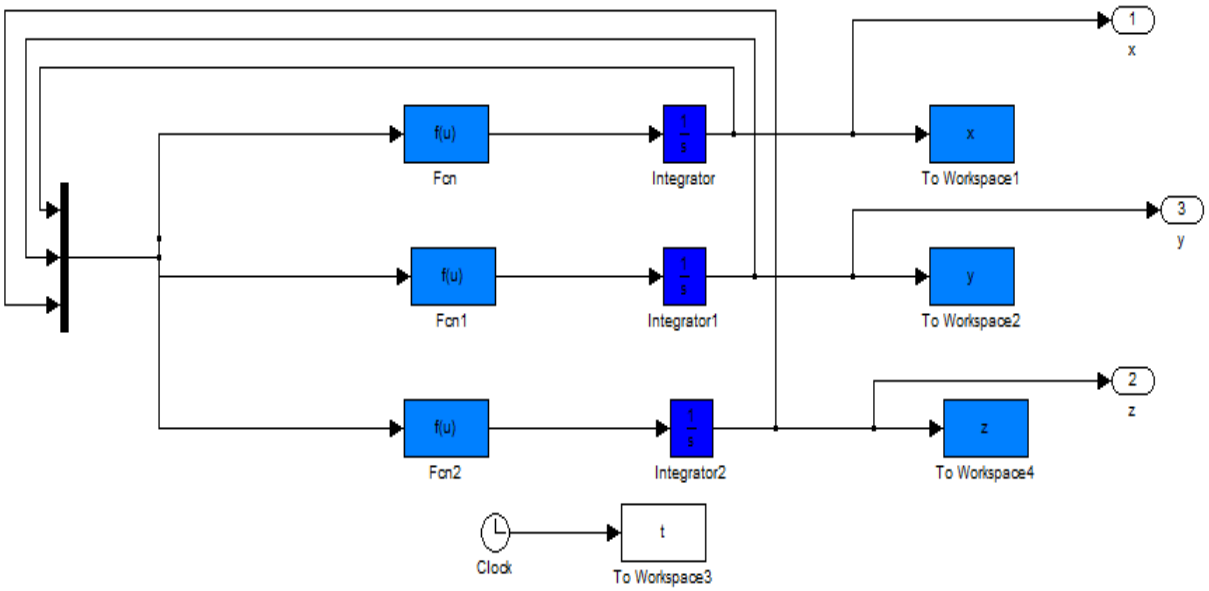


Figure (4.20) : Schéma du système de Lorenz sur Matlab/Simulink.

4.3.6 la simulation du système de lu sous Matlab/Programme :

Les figures [(4.21) (4.22) (4.23) (4.24)] représentent les résultats de la méthode de Runge Kutta d'ordre 4 appliquée aux systèmes des Lu avec h=0.01 programmée sous Matlab.

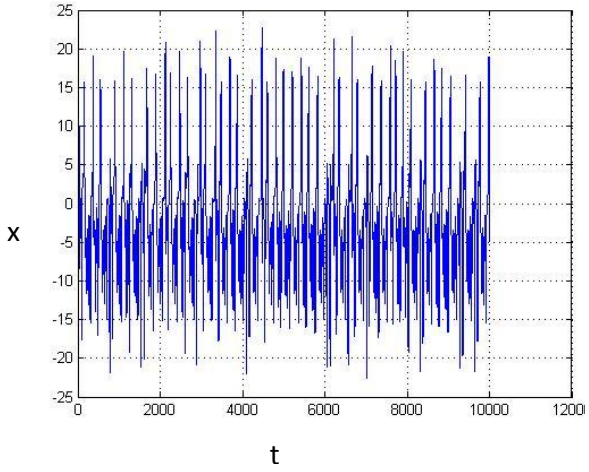


Figure (4.21) : Signal x (t)

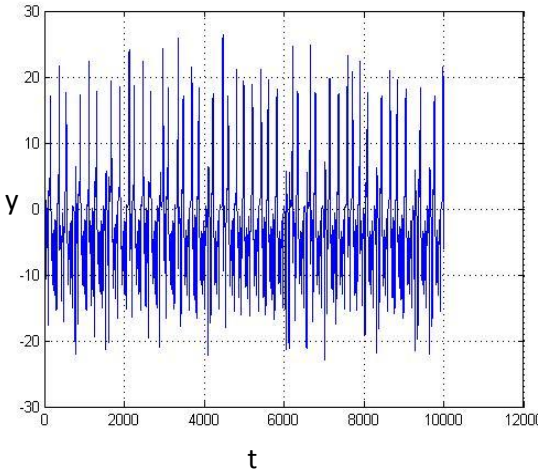
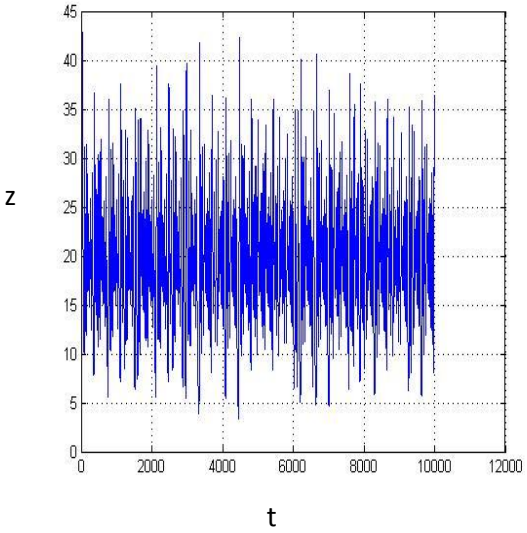


Figure (4.22) : Signal y (t)



Figure(4.23) : Signal x (t)

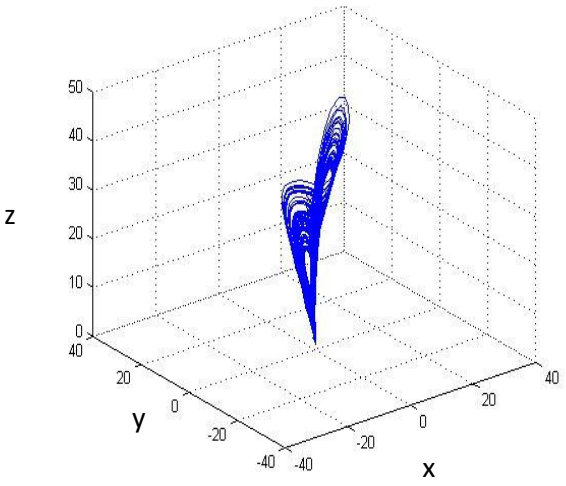


Figure (4.24) : Attracteur lu

1.1.1 4.3.6 La simulation du système de lu sous Matlab/Simulink :

Les figures [(4.25) (4.26) (4.27) (4.28)] représentent les résultats de la Simulation du système de lu sous Matlab/Simulink avec les conditions initiales de système ont été choisis de la manière suivante : $(x_0, y_{0,} z_0)$ = (6, 4,2). :

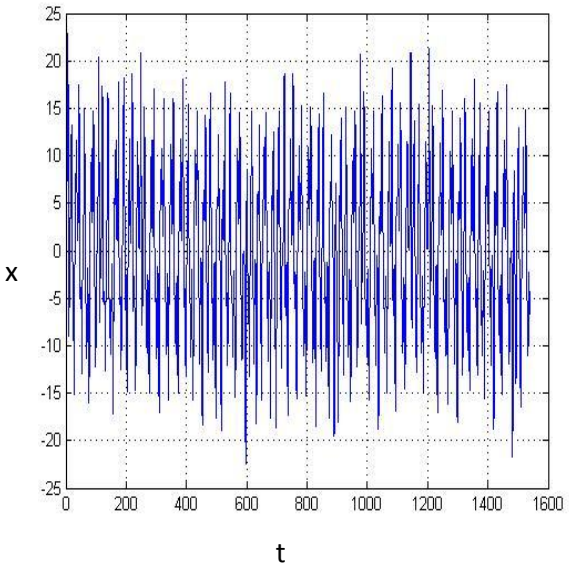


Figure (4.25) : Signal x (t)

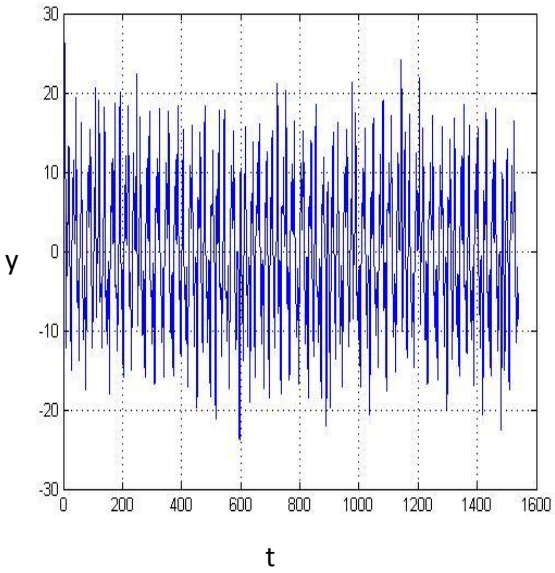


Figure (4.26) : Signal y (t)

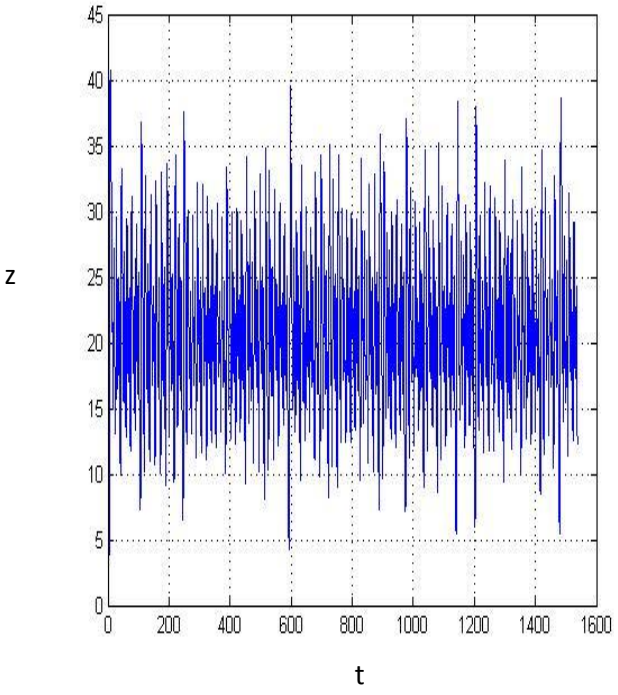
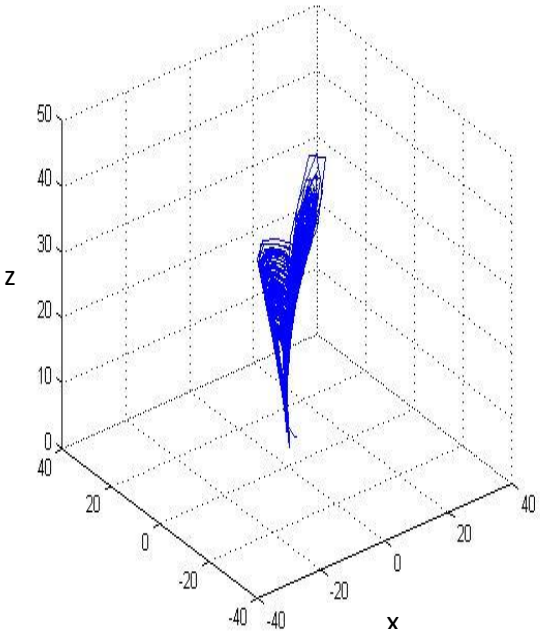


Figure (4.27) : Signal z (t)



Figure(4.28): Attracteur Lorenz.

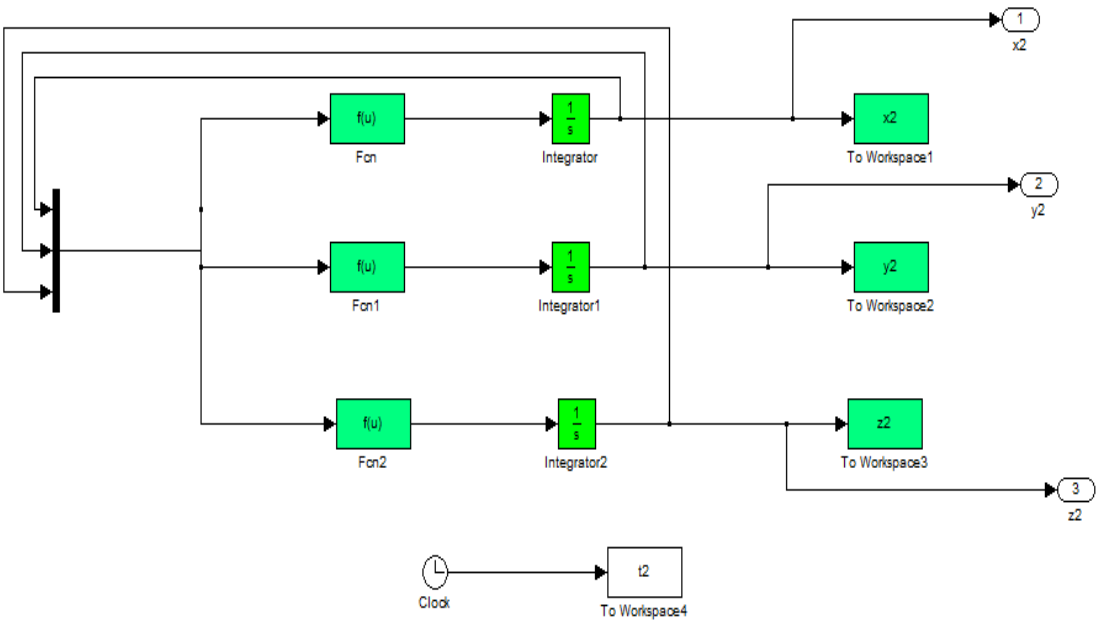


Figure (4.29): Schéma du système de Lu SOUS Matlab/Simulink.

4.4 Résultats de l'implémentation sur la carte DSP « eZdsp TMF2812 »

Les figures [(4.30), (4.31), (4.32)] représentent les résultats de l'implantation de la du système de Rössler sur le Code Composer Studio (CCS v3.1).

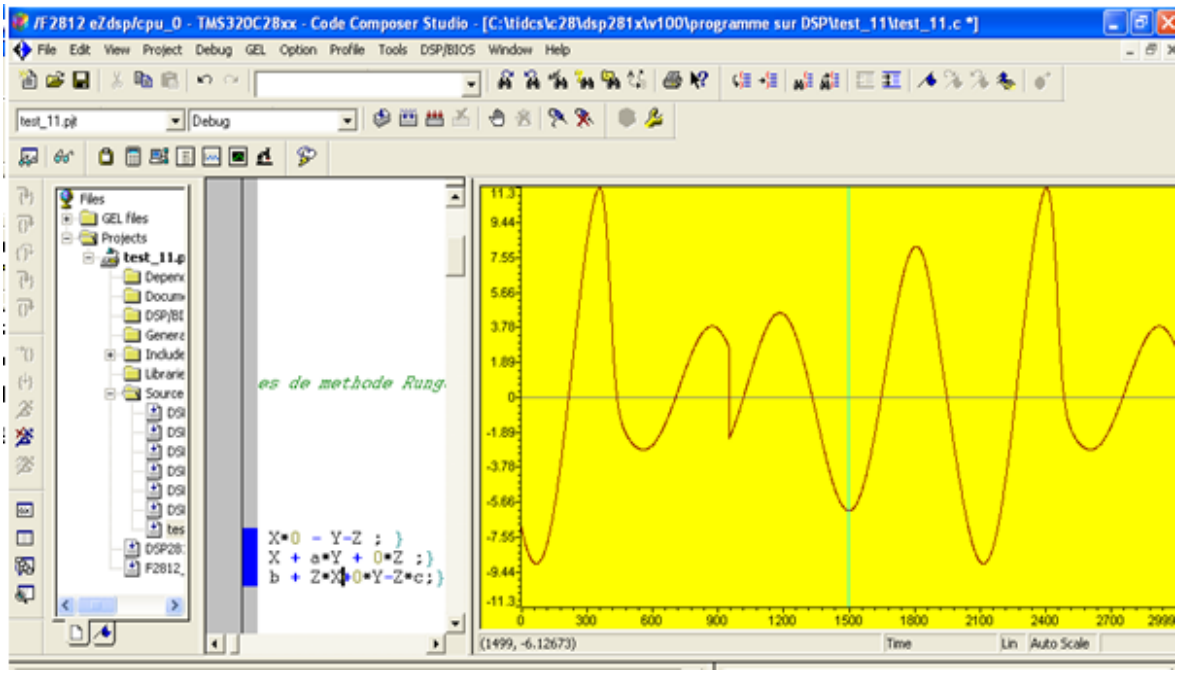


Figure (4.30) : Signal X(t)

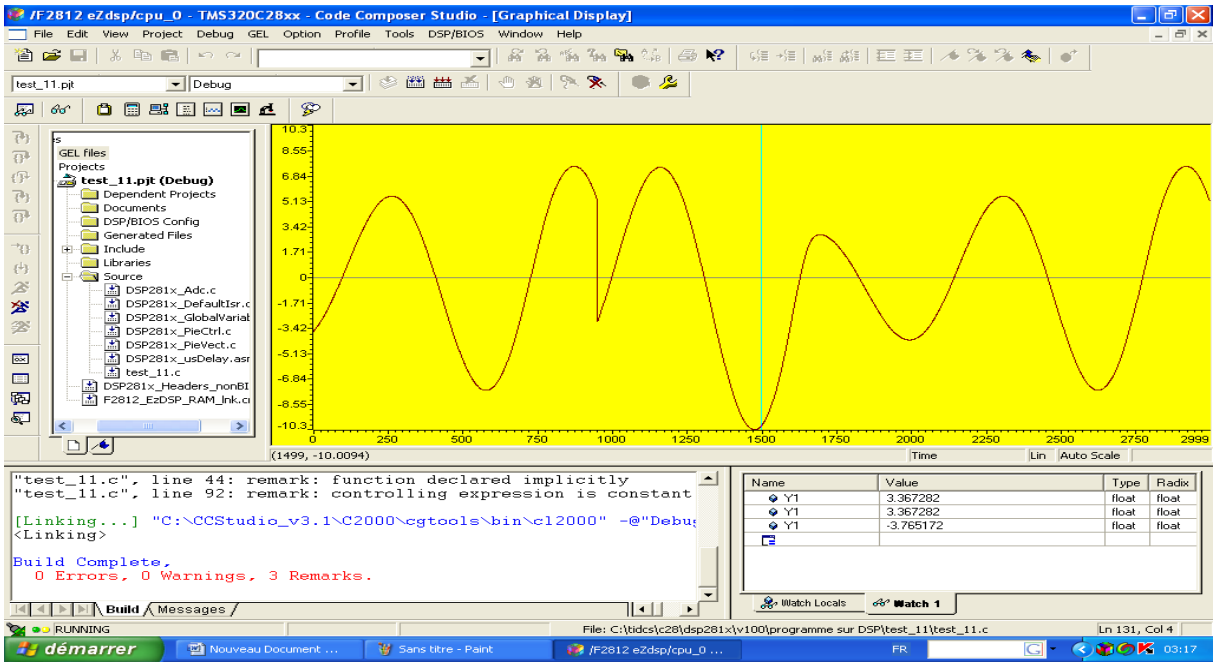


Figure (4.31) : Signal Y(t)

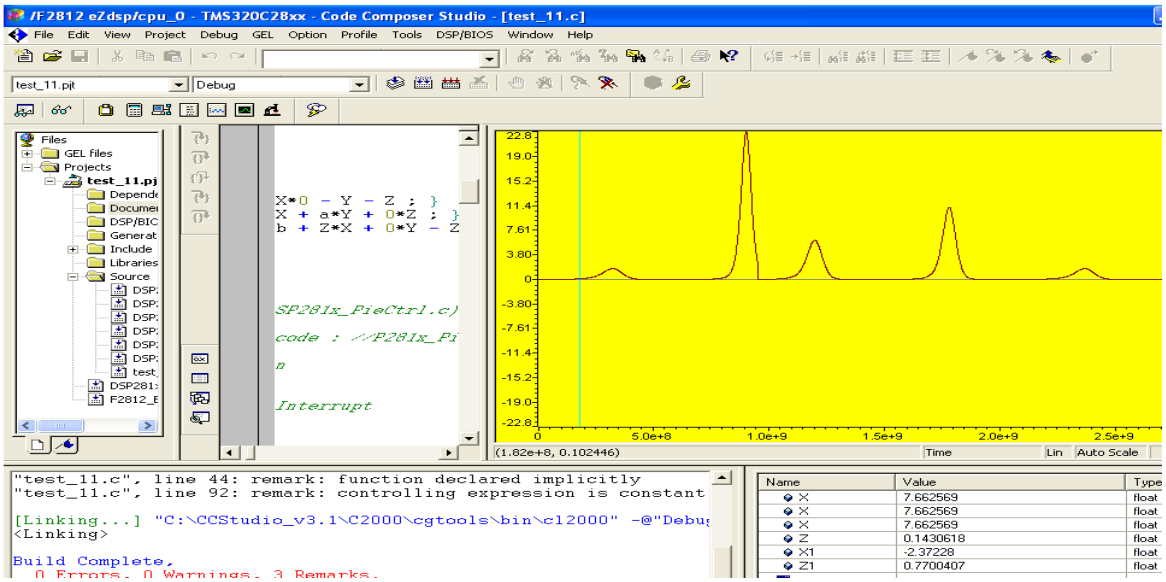


Figure (4.32) : Signal Z(t)

Les figures [(4.33) (4.34) (4.35)] représentent les résultats de l'implantation de la du système de Lorenz sur le Code Composer Studio (CCS v3.1).

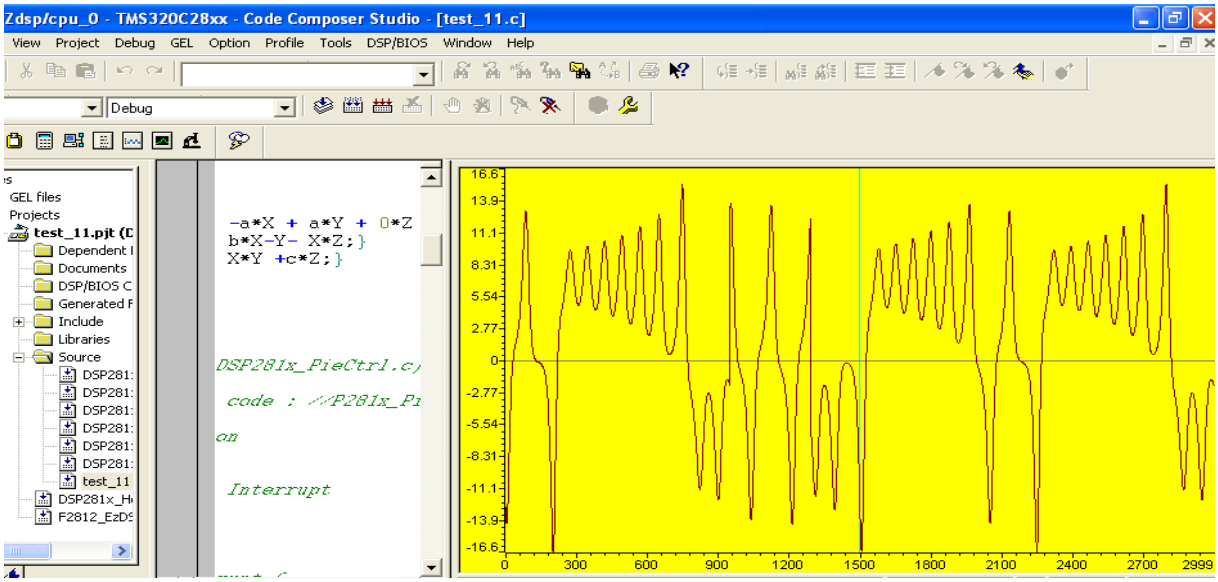


Figure (4.33) : Signal X(t)

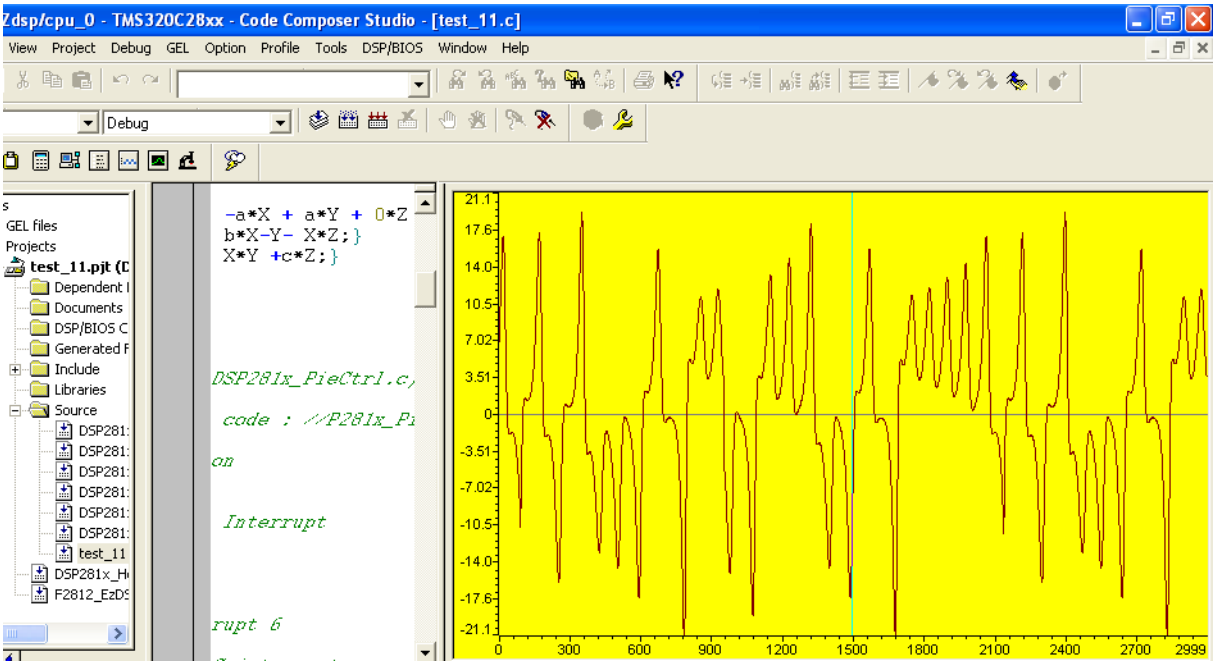


Figure (4.34) : Signal Y(t)

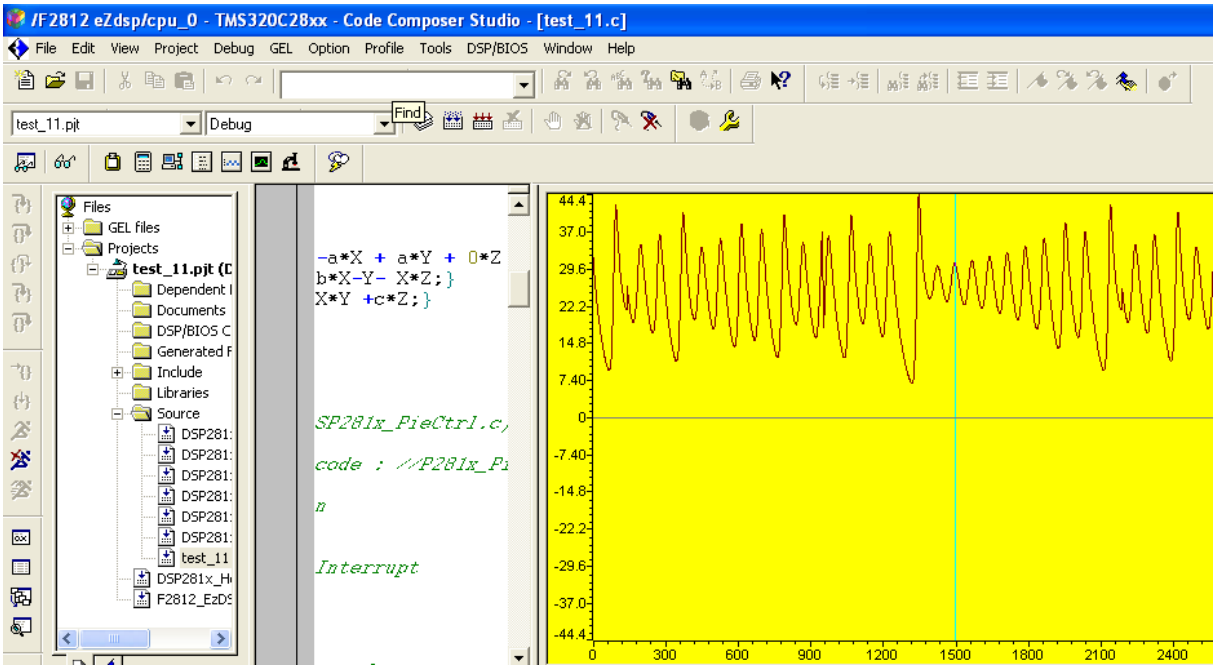


Figure (4.35) : Signal Z(t)

Les figures [(4.36), (4.37), (4.38)] représentent les résultats de l'implantation de la du système de Lu sur le Code Composer Studio (CCS v3.1).

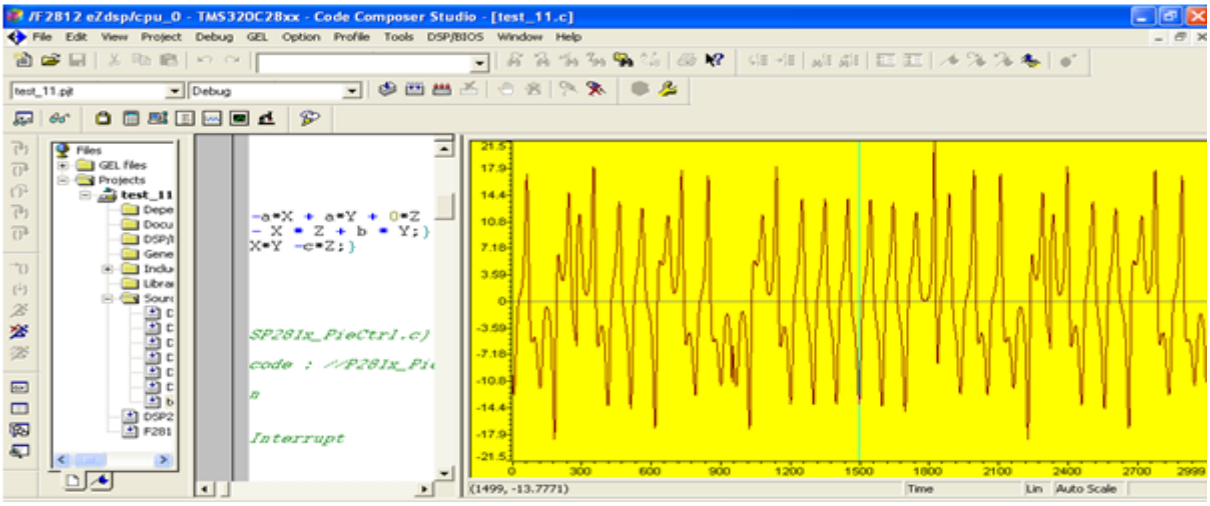


Figure (4.36) : Signal X(t)

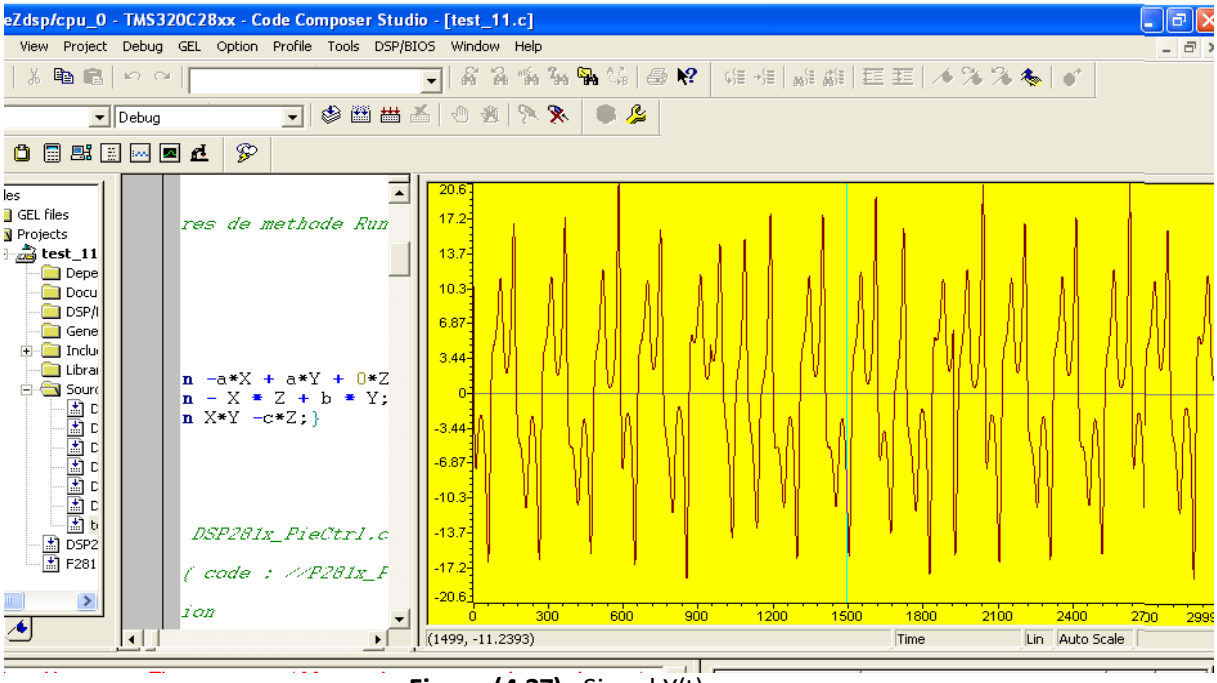


Figure (4.37) : Signal Y(t)

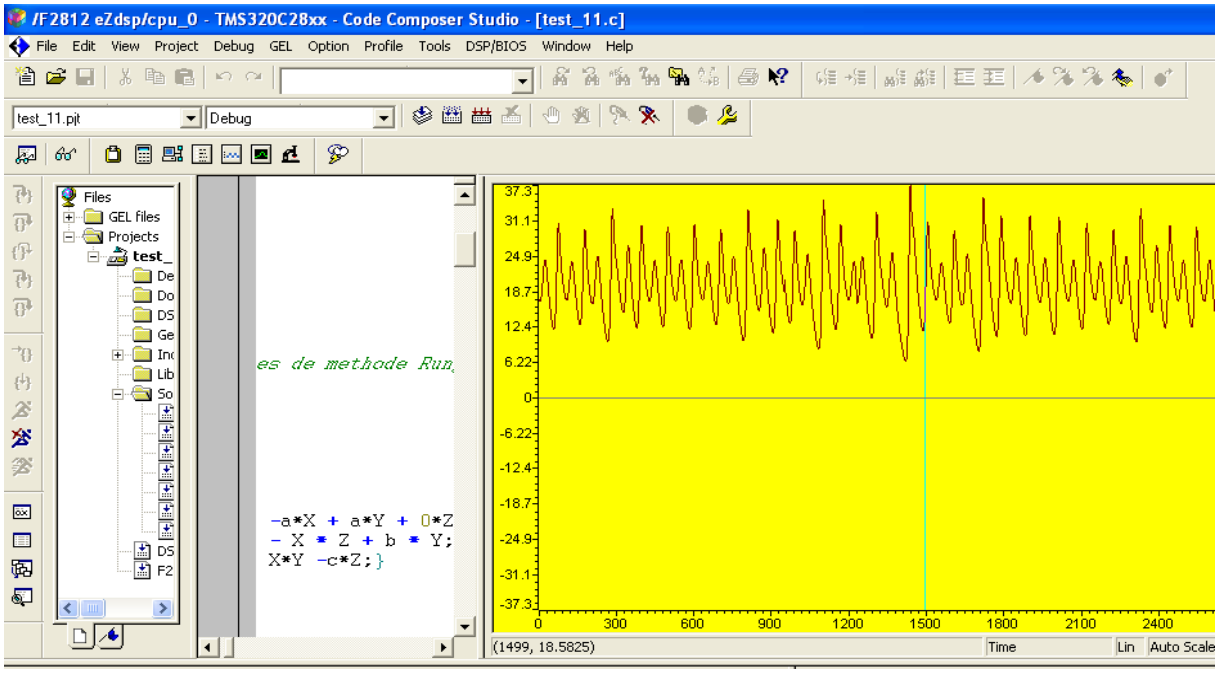


Figure (4.38) : Signal Z(t)

4.5 Commentaires :

- Les figures [(4.3) (4.4) (4.5) (4.6) (4.7) (4.8) (4.9) (4.10)] représentent les résultats de la simulation sous Matlab du système de Rössler et les figures [(4.30) (4.31) (4.32)] représentent les résultats expérimentaux du système de Rössler sur la carte DSP.

Telle que : X, Y, Z sont les variables d'états, et a=0.2, b=0.2, c=5.7 sont des valeurs réels et les conditions initiales de système ont été choisis de la manière suivante $(x_0, y_{0,} z_0)$ = (0, 0, 0).

- Les figures [(4.12) (4.13) (4.14) (4.15) (4.16) (4.17) (4.18) (4.19)] représentent les résultats de la simulation sous Matlab du système de Lorenz et les figures [(4.33) (4.34) (4.35)] représentent les résultats expérimentaux du système de Lorenz sur la carte DSP.

Telle que : X, Y, Z sont des variables d'états, et a=36, b=28, c=-8/3 sont des valeurs réels et les conditions initiales de système ont été choisis de la manière suivante $(x_0, y_{0,} z_0)$ = (0.05, 1, 3).

- Les figures [(4.21) (4.22) (4.23) (4.24) (4.25) (4.26) (4.27) (4.28)] représentent les résultats de la simulation sous Matlab du système de Lu et les figures [(4.36)

(4.37) (4.38)] représentent les résultats expérimentaux du système de Lu sur la carte DSP.

Telle que : X, Y, Z sont les variables d'états, et a=36, b=20, c=3 sont des valeurs réels et les conditions initiales de système ont été choisis de la manière suivante $(x_0, y_{0,} z_0)$ = (6, 4, 2).

Remarque :

Les résultats obtenus sur MATLAB sont meilleurs que ceux obtenus par l'implémentation sur la carte DSP. Cette différence revient à la précision avec laquelle les calculs sont faits par le DSP F2812 qui est un micro-processeur à virgule fixe 32 bits.

4.6 Conclusion

Les résultats obtenus après la simulation du système de Rössler et Lorenz et Lu sur logiciel Matlab pour l'étude théorique ainsi que l'implémentation sur la carte DSP TMS320F2812 concernant l'étude pratique on constate après avoir fait une analyse comparative que les résultats obtenus sont acceptables et relativement proches de ceux donnés par la simulation.

Conclusion générale

Ce travail consiste à réalise un système de transmission de donnée basé sur la synchronisation de deux systèmes chaotiques, Ces systèmes résulte de l'inclusion de la théorie de chaos, et nous avons analysé la synchronisation des systèmes chaotiques à base d'observateur a mode glissant et leur applications pour la transmission d'information.

Dans le premier chapitre de ce travail nous avons évoqué d'abord quelque notion sur les systèmes dynamiques, par la suite nous avons présenté les systèmes chaotiques et leurs caractéristiques, Et dans le deuxième chapitre nous avons donnée l'analyse d'observabilité et synthèse d'observateur, leurs définitions, d'abord ce chapitre consacré à l'étude du phénomène de la synchronisation, Ensuite le troisième chapitre basé sur deux parties la première partie est de la résolution des systèmes chaotiques par la méthode de Range Kutta et la deuxième partie est consacrée à l'implémentation de cette méthode sur la carte DSP F2812.

Après cela nous avons finalisé notre travail par des résultats de simulation numérique sous Matlab puis de Simulink et d'implémentation sur la carte Dsp.

Nous concluons que :

- L'analyse des systèmes chaotiques (Rössler, Lorenz, Lu) par la méthode de Runge-Kutta donne des résultats analogues avec plusieurs approches différentes.
- La réalisation de ces systèmes, nous a donné un aspect pratique à l'étude théorique acquise.

- On peut récupérer les signaux des systèmes chaotique par l'observateur a mode glissant.

Donc ce travail consiste a la synchronisation des systemes chaotiques par observateur a mode glissant et la réalisation des systèmes chaotique par l'implémentation sur la carte Dsp.

Bibliographie

[1] :IbtissamTalbi « Systhèmes Dynamiques non linéaires et phénomènes de chaos,29 .06 .2010.université Montouri de constantine ».

[2] : Georges Kaddoum « Contributions à l'amélioration des systèmes de communication multi-utilisateurs par chaos» Thèse de doctorat d'université de Toulouse, France, 2008.

[3] :Ali Zemouche « l'observation de l'état des systèmes dynamiques non linéaires ».Thèse de Docteurat de l'Université Louis Pasteur Strasbourg , soutenue publiquement le 30 mars 2007.

[4] : MEGHERBI_OUERDIA« Etude et réalisation d'un système sécurisé à base de système chaotique », mémoire de magister le 10/10/2013) Université Mouloud Mammeri Tizi-Ouzou.

[5]: Devaney, R.L. [1987] "An Introduction to Chaotic Dynamical Systems". (Addison-Wesley,New York)

[6] :Li, T.Y. & Jorke, J.A. [1975] "Period three implies chaos", American Mathematical Monthly, 82, pp. 481—485.

[7] : Abdelkrim Boukabou :« Méthodes de contrôle des systèmes chaotiques d'ordre élevé». Thèse de doctorat d'Université de Constantine, Algérie, 2006.

[8] : Mammeri Mohammed« La stabilité Structurelle des Difféomorphismes Quadratiques en Dimension 2 ». Thèse de magister d'Université kasdi merbah – Ouargla, Algérie, 2011.

[9] :Ikhlef Ameur « Contrôle chaotification et hyperchaotification des systèmes dynamiques ».Thèse de magister d'Université de Mentouri Constantine, Algérie, 2007.

[10] :RUELLE, D et TAKENS, F « On the nature of turbulence Commun Math Phys 1971

[11]Y.S. Tang, A.I. Mess and L.O. Chua, Synchronization and chaos, IEEE Transaction on circuit and systems, vol. 30, pp-1-2, 1983.

[12] L.M. Pecora and T.L Carroll, "Synchronization in chaotic systems ,physicals Review and Letters,pp.821-824,1990".

[13]: L.M. Pecora and T.L Carroll," Synchronized chaotic signal and systems ,proceeding IEEE International conference of Acoustics , Speech ,and signal ICASSP'92,Minneapolis,MN ,USA,pp.137-140,1992".

[14]:A. Fradkor, A.Y .Pogromsky," introduction to control of oscillations and chaos .world scientific, singapor, serisesA,vol.35,1998".

[15] :E. Ott, C. Grebogi and J.A. Yorke." Controlling chaos, physical Review Letters, pp.821-824, 1990".

[16]:M .Chen.D. Zhou and Y.Shang "A.new observer_based synchronization Scheme for private communication ,chaos,Solitons and fractals , 30,pp.1025_1030,2005".

[17]:A. fradkov, H.Nijmeyer and A.Markov," adaptive observer _based synchronization for communication ,International journal of Bifurcation and choas,pp.2807_2813.2000 " .

[18]:K.M Cuomo and A.V. Oppenheim" circuit implementation of synchronized chaos with application to communication , IEEE Transactions an circuit and systems , vol.40?,PP.626_633,1993.

[19]:K.M. Cuomo ,A.Y. Oppenheim and S.H. Strogatz "synchronization of Lorenz base d chaotic circuits with application to communications ,physics ,review and letters,vol.71,pp.65_68,1993" . **[20]A.V. Oppenheim, G.W-WORNELLI,S.H. Isabelle and K.M."** Cuomo ,signal processing in the context of chaotic signals ,Speech ,and signal Processing ICASSP'92,San Francisco, USA ,pp.117-120,1992".

[21]L. Koracev, chaos _base d Glyptography ,a brief overview ,IEEE Transaction an circuits and systems ,pp. 6_21,July 2001.

[30] : BOUTAT BADDAS « Analyse des singularités d'observabilité et de détectabilité '' application a la synchronisation des circuits électroniques chaotique ». Thèse de doctorat 2002, université de Cergy pontoise.

[31] : Mihai Bogdan Luca. « Apports du chaos et des estimateurs d'états pour la transmission sécurisée de l'information ».

[32] : palançon robert « Systèmes linéaires continus et invariant »

[33] : ODEN jérérny « Le chaos dans les systèmes dynamiques »5 juillet 2007.

[34] : Ali Zamouche « Sur l'observation de l'état des systèmes dynamiques non linéaires ». Thèse de Docteur.

[35] : Rafael Marti´nez-Guerra , Wen Yu, Enrique Cisneros-Saldan. « A new model-free sliding observer to synchronization problem».

[37]: Pr. Mohamed DJEMAI. « Analyse, commande &observation des Systems non linéaires».

[38] :G. Baudoin et F.Virolieu « Les DSP Famille TMS320C54x : Développement d'applications', Dunod, Paris, France, 2000 ».

[39] :BLAIFI Sid-Ali,NAMANE Abdenour« Etude et implémentation de la commande DTC sur une carte DSP basée sur un régulateur flou appliquée à la machine asynchrone', thèse de master Université Saad Dahlab de BLIDA, Algérie, 2012 ».

[40] : BRAHMI Abdelghani et HAMOUDI Ahmed « Implémentation du filtre Kalman sur DSP pour l'estimation des grandeurs rotoriques.Université Saad Dahlab de BLIDA, Algérie, 2012 ».

[41] :KAHLANE Abdelwahid Hamza« Processeur de traitement de signal (DSP) : Contrôle des onduleurs photovoltaïques',CDER 2014 ».

[42] :a. Lallouani,«Dé bruitage d'un signal de la parole corrompu par un bruit colore en utilisant la transformée en ondelettes et implantations sur un processeur de traitement numérique des signaux», Pour un grade de maitrise, Université du Québec, septembre 2004.

[43] : http://www.physique.usherbrooke.ca/pages/sites /files/admin/PHQ4 05_ipad