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

LAROUI Samir

&

REBHI Ahmed

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

Thème

Génération Numérique d'un Signal chaotique

Proposé par : Mr. FERDJOUNI Abdelaziz

Année Universitaire 2012-2013

ملخص:

يتمثل هذا العمل في توليد إشارة فوضوية يقدمها نظام روسلر.

أولا يتم تحليل النظام، ثم حله 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 لهذا النظام على بطاقة DSP TMS320F2812 لمشاهدة الإشارات في الوقت الحقيقي.

كلمات المفاتيح : إشارة الفوضى؛ نظام روسلر؛ طريقة رونج كوتا؛ بطاقة DSP.

Résumé :

Ce travail consiste à générer un signal chaotique fourni par le système de Rössler. D'abord ce dernier est analysé, ensuite, sa résolution effectuée à l'aide de la méthode numérique de Runge-Kutta d'ordre 4. La simulation du système Rössler est effectuée à l'aide de Simulink, ensuite à l'aide d'un programme Matlab. Les signaux de sortie sont visualisés et comparés. Enfin, une implémentation digitale de ce système est réalisée sur la carte DSP TMS320F2812 pour voir les signaux eu temps réel.

Mots clés : signal chaotique; système de Rössler; méthode de Runge Kutta; la carte DSP.

Abstract:

This work is to generate a chaotic signal from the Rössler system. First it is analyzed, then the resolution is made using the numerical method of Runge Kutta 4. Rössler system simulation is performed using Simulink, then using a Matlab program. The output signals are displayed and compared. Finally, a digital implementation of the system is performed on the DSP TMS320F2812 card to see signals in real time.

Keywords: chaotic signal; Rössler system; method of Range-Kutta; DSP card.

Dédicace

Je dédie ce travail à l'âme de mon cher père et à ma mère auxquels je ne pourrai jamais exprimer ma gratitude par quelques mots, que sans eux je ne serais jamais ce que je suis aujourd'hui et que je ne pourrai jamais leur rendre ce que mon offert d'amour, d'affection et de soutien.

Je le dédie à mes frères et mes sœurs

Je le dédie aussi à mes amis Dalal, Sid Ali, Amine et tous les amis de master 2 TISE ainsi que mon binôme AHMED qui ont toujours été là quand j'avais besoin.

A la fin, à tous ceux que j'ai n'est pas cité.

LAROUI SAMIR

Introduction générale

L'analyse des systèmes dynamiques linéaires et non linéaires, a fait l'objet d'une importante quantité de travaux de recherche menée par des chercheurs de différents domaines. Certains 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 est imprévisible à long terme : c'est le comportement chaotique. L'analyse de tels systèmes permet de déterminer dans quelles conditions le fonctionnement en mode chaotique se produit. Les applications des systèmes chaotiques sont multiples, l'une des importantes d'entre elles est la cryptographie ou la transmission sécurisée des données [1].

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, dichotome, etc...), et ensuite implémenter le programme sur une carte DSP ou FPGA.

L'objet de notre travail est de générer des signaux chaotiques à l'aide du système de Rössler. On commence par un travail d'analyse et de simulation par la solution à l'aide de la méthode Runge Kutta d'ordre 4. Un programme sous Matlab est écrit et qui montre la validité de la solution proposée. Ensuite, un autre programme en langage C, destiné à une implémentation sur la carte DSP TMS320F2812, est proposé.

Ce mémoire est organisé en quatre chapitres, comme suit:

Le premier chapitre de ce mémoire, est consacré à la présentation générale des systèmes dynamiques et chaotiques. Nous donnons quelques notions importantes de base nécessaire à la compréhension de la théorie du chaos.

Dans le deuxième chapitre, nous présentons quelques généralités sur la résolution des systèmes dynamiques par les méthodes numériques. Parmi ces méthodes, nous appliquons la méthode de Runge Kutta d'ordre 4 à l'exemple de système de Rössler.

Les caractéristiques de la carte DSP TMS320F2812 sont données dans le troisième 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 sur la carte DSP est présentée.

Dans le quatrième chapitre nous donnons des résultats et des commentaires de la simulation du système de Rössler sur logiciel Matlab et langage C, ainsi que les résultats obtenus par l'implémentation, afin de comparer ces résultats avec ceux de la simulation théoriques.
Finalement nous terminons par une conclusion générale sur l'ensemble de cette étude.

Liste des figures

Liste des tableaux

Listes des acronymes et abréviations

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: registre 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.

TxCON: timer x Control register.

Remerciements

Nous tenons tout d'abord à remercier Allah le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, nous tenons à remercier notre promoteur :

Mr. FERDJOUNI ABDELAZIZ Les conseils qu'il nous a prodigué, la patience, la confiance qu'il nous a témoignés ont été déterminants dans la réalisation de notre travail de recherche.

Nos chaleureux remerciements à nos familles, qui nous ont soutenus par leur Amour, leurs encouragements et leur intérêt tout au long de nos études.

Nous tenons à remercier Mr. BLAIFI SID ALI qui prépare un doctorat en électronique à Université de Médéa pour son orientation et les conseils dans ce travail.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Table des matières

Chapitre 1

Systèmes dynamiques et chaos

Chapitre 2

Résolutions des systèmes dynamiques chaotiques par la méthode

Runge-Kutta d'ordre 4

Chapitre 3

Implémentation du système chaotique sur la carte DSP

Bibliographie

[1] Georges Kaddoum :'Contributions à l'amélioration des systèmes de communication multi-utilisateurs par chaos : synchronisation et analyse des performances', Thèse de doctorat d'université de Toulouse, France, 2008.

[2] Huyên Dang-Vu, Claudine Delcarte :'Bifurcations et chaos : Introduction à la dynamique contemporaine avec des programmes en Pascal, Fortan et Mathématica', Ellipses, Paru en 09/2000.

[3] Ikhlef Ameur :'Contrôle chaotification ethyperchaotification des systèmes dynamiques', Thèse de magister d'Université de Mentouri Constantine, Algérie, 2007.

[4] Menacar Tidjani : 'Synchronisation des Systèmes Dynamiques Chaotiques à Dérivées Fractionnaires', Thèse de magistère en mathématiques d'Université Mentouri Constantine, Algérie, 2009.

[5] Ibtissem TALBI, Systèmes dynamiques non linéaires et phénomènes de chaos (Application à la Cryptographie), Thèse de magister d'Université Mentouri de Constantine, Algérie, 2010.

[6] Abdelkrim Boukabou : ' Méthodes de contrôle des systèmes chaotiques d'ordre élevé et leur application pour la synchronisation: Contribution à l'élaboration de nouvelles approches', Thèse de doctorat d'Université de Constantine, Algérie, 2006.

[7] Zeraoulia Elhadj : ' Étude de quelques types de systèmes chaotiques: généralisation d'un modèle issu du modèle de Chen', Thèse de doctorat en mathématiques d'Université de Mentouri 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] André NIETTO, César SAUVAGE, Xavier MORENO, Camille PIALLAT : ' méthodes d'intégration d'équations différentielles ' Polytech Nice, France, 2012.

[10] www.umc.edu.dz/vf/images/cours/regulation/chap1_equadiff.pdf

[11] http://www.math.u-bordeaux1.fr/~jgillibe/enseignement/MHT204_chap4.pdf

[12] Saïd Mammar : ' MT31-méthodes numériques, institut universitaire professionnalisé d'Evry',1999 .

[13] vcorbex.pagesperso-orange.fr/terminale/physique/activites/Euler.pdf

[14] Julien Fontchastagner : ' Analyse 3 Introduction aux Équations Différentielles ', Université de Nancy, France, 2007.

[15] MAOUGAL Abdelaziz : ' Optimisation d'un écoulement de couche limite par la méthode E-G-M Entropy Generation Minimisation', Thèse de doctorat d'Université de Constantine, Algérie, 2010.

[16] www.ac-nancy-metz.fr/enseign/physique/divers/.../Schw.../RK4_02.pdf

[17] http://www.physique.usherbrooke.ca/pages/sites/default/files/admin/PHQ405_ipad

[18] MESSAOUDA KHAMMAR : ' Etude d'un jet en spray d'une solution chimique sur un substrat chaud destiné à l'élaboration des couches minces', Thèse de magister d'Université de Constantine, Algérie, 2010.

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

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

[21] 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.

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, l'analyse, la géométrie, la théorie de mesure et probabilités, et qui s'efforce d'étudier les propriétés d'un système dynamique. La nature de cette étude diffère suivant le système dynamique étudié.

Un système dynamique est dit chaotique si une portion « significative » de son espace des phases présente la caractéristique suivante: Le phénomène de sensibilité aux conditions initiales. La présence de cette propriété entraîne un comportement extrêmement désordonné qualifié à juste titre de « chaotique ».

Les systèmes chaotiques s'opposent notamment aux systèmes intégrables de la mécanique classique, qui furent longtemps les symboles d'une régularité toute puissante en physique théorique.

L'objectif de ce chapitre est d'étudier les systèmes dynamiques et chaotiques, nous donnons quelques notions de base de la théorie de chaos.

1.2 Les systèmes dynamiques

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

1.2.1 Système dynamique à temps discret

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

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

Ou $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ 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 [1].

v : Vecteur du paramètre.

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

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].

Le système de Rössler [2] est donné par les équations suivantes:

$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + z\,x - z\,c \end{cases} \quad (1.4)$$

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 l'équation (1.4) sont choisis de la manière suivante : a = 2.25; b = 0.04; c = 2.5 avec $(x_0, y_{0,} z_0)$ = (0.25, 0.45, 0.85)

Ce type de système est dit autonome car l'équation (1.4) n'a pas de dépendance explicite par rapport au temps t . La figure 1.1 est un exemple de trajectoire du système de Rössler.

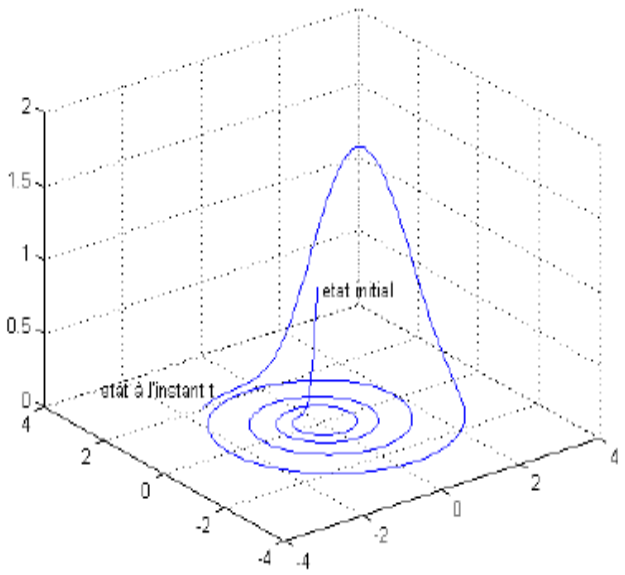


Figure 1.1. Une trajectoire pour le système de Rössler.

1.4 Section de Poincaré

Henri Poincaré a apporté une contribution très utile pour l'étude des systèmes chaotiques. Parmi ces contributions on trouve les sections de Poincaré. Réaliser une section de Poincaré revient à couper la trajectoire dans l'espace des phases. L'application de Poincaré est un outil mathématique simple permettant de transformer un système dynamique continu en un système dynamique discret. Cette transformation se fait par une réduction d'une unité de l'ordre du système.

Les mathématiciens ont démontré que les propriétés du système sont conservées après la réalisation d'une section de Poincaré intelligemment choisie [4].

Définition

Soit un système dynamique définit par l'équation suivante :

$$\dot{x}(t) = f\big(x(t)\big),\ x(0) = x_0 \tag{1.5}$$

Avec $f : \mathbb{R}^{n} \longmapsto \mathbb{R}^{n}$, $n > 2$ et soit Ω un ensemble de $\mathbb{R}^{2}$, l'intersection du plan Ω et la trajectoire du système (1.6) nous permet de définir une fonction G comme suit

$$\mathrm{G} : \mathrm{U} \subset \Omega \rightarrow \Omega$$

$$x(\mathrm{t}) \rightarrow \phi(x(t), \tau)$$

Avec τ désigne le temps qu'il faut pour que les trajectoires $x(\mathrm{t})$démarrent de U pour arriver à Ω. La fonction G est appelée la fonction de premier retour [4].

Remarque

Dans les systèmes à temps discret, pour une variable d'état x, l'application de premier retour consiste plus simplement à associer $x(\mathrm{t})$ à $x(\mathrm{t}-1)$.

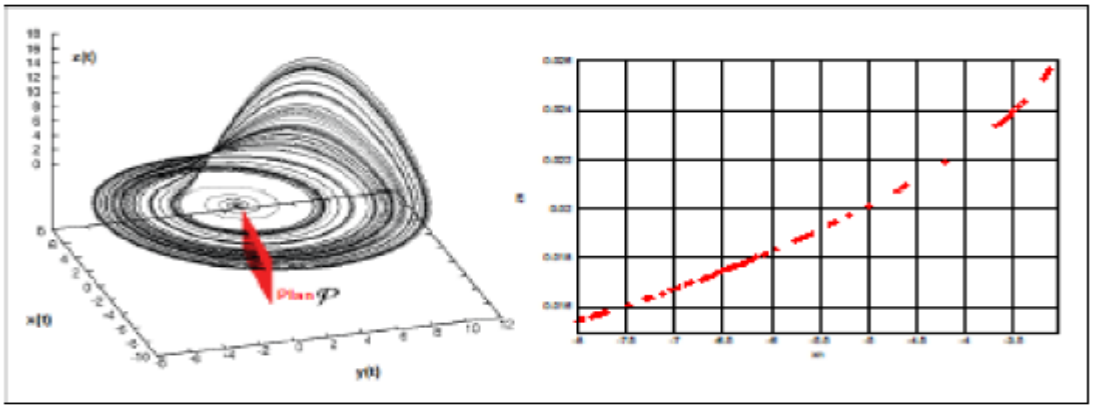


Figure 1.2. Intersection de la trajectoire de l'attracteur de Rössler avec un plan P

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] :

$$\dot{\mathrm{V}} = \sum_{\mathrm{j}=1}^{\mathrm{n}} \frac{\partial \mathrm{V}}{\partial x_{\mathrm{j}}} \dot{x}_{\mathrm{j}} = \sum_{\mathrm{j}=1}^{\mathrm{n}} \frac{\partial \mathrm{V}}{\partial x_{\mathrm{j}}} f_i(x) \tag{1.6}$$

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.

❖ **Exemple de fonctions de Lyapunov**

L'équation différentielle de système est :

$$m\,\ddot{x} + k\,(x + x^3) = 0 \tag{1.7}$$

Ou, en posant $\dot{x} = y$

$$\dot{x} = y\,,\, \dot{y} = -\frac{k}{m}(x + x^3) \tag{1.8}$$

Etudions la stabilité du système (1.7) au point $(x, y) = (0,0)$. L'énergie totale d'un système est :

$$E(x, y) = \frac{my^2}{2} + k\,(\frac{x^2}{2} + \frac{x^4}{4}) \tag{1.9}$$

$E(x, y)$ est aussi une fonction de Lyapunov pour (1.8), car $E(0,0) = 0$

Et $E(x, y) > 0$ pour $(x, y) \neq (0,0)$

Et $\dot{E} = my\dot{y} + k(x + x^3)\dot{x} = -ky(x + x^3) + k(x + x^3)y \equiv 0$ (1.10)

Donc le point fixe $(x, y) = (0,0)$ est stable, d'après le théorème de Lyapunov [2].

1.6 Théorie du Chaos

1.6.1 Définition du chaos

En générale il n'y a pas de définition rigoureuse du chaos, On peut observer le phénomène du chaos dans plusieurs domaines, mais comment le formaliser ? La réponse est négative car jusqu'à l'heure actuelle, il n'existe pas une théorie générale qui donne une explication ou une caractérisation finale de ce phénomène. Tout ce que nous pouvons dire est qu'il existe plusieurs critères physiques qui permettent de confirmer qu'un système est chaotique [5].

1.6.2 Historique du chaos

La théorie du chaos, très récemment développée, n'a pas bien été acceptée au début: Elle représentait un grand changement de modèle, une rupture épistémologique dans la science de la fin du XIX ^{éme} siècle [6].

En 1890 : Le Roi Oscar II de Suède donne un prix au premier chercheur qui pourrait déterminer et résoudre le problème des orbites des corps célestes et ainsi prouver la stabilité du système solaire. Jusqu'à ce jour, le problème n'a pas été résolu.

En 1963 : Edward Lorenz découvre le premier système chaotique dans la météo ou encore appelé attracteur étrange.

En 1975 : Tien-Yien Li et James A. Yorke ont présenté pour la première fois le terme "Chaos" dans un article appelé "Period three implies 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.

1.6.3 Application du chaos

- **Contrôle** : Première application du chaos est le contrôle du comportement irrégulier dans les circuits et les systèmes.
- **Synchronisation** : Communication sécurisée, cryptage, radio.
- **Traitement d'information** : Traitement d'information Codage, décodage et stockage d'information dans des systèmes chaotiques, tel que les éléments de mémoires et les circuits, Reconnaissance de forme.
- **Prédiction à court terme** : Les maladies contagieuses, température, économie [6].

1.6.4 Domaine d'application du chaos

- **Ingénierie** : 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 (EEG), prédiction et contrôle d'activité irrégulière du cœur.
- **Management et finance** : Prévisions économiques, analyse financière, et prévision du marché [6].

1.6.5 Caractéristique du chaos

Caractériser le chaos est un sujet vaste, et les tests les plus utiles pour ce type de comportement sont les suivants [6]:

- Les mouvements chaotiques sont plus compliqués que stationnaire, périodique ou quasi-périodique, et ils ont des formes très complexes, appelés attracteurs étranges.
- Sensibilité aux conditions initiales : La plupart des systèmes chaotiques montrent la sensibilité aux conditions initiales.
- La non-linéarité : Si le système est linéaire, il ne peut pas être chaotique.
- Le déterminisme : Un système chaotique a des règles fondamentales déterministes (plutôt que probabilistes).
- 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).

1.7 Les attracteurs

1.7.1 Définitions d'un attracteur

En général, un attracteur est défini comme une sous-partie fermée de l'espace des phases qui attire toutes les autres orbites vers elle [2].

Définition : Soit A un ensemble compact, fermé de l'espace des phases. On suppose que A est un ensemble invariant ($\phi_t(A) =$ A pour tout t), l'ensemble A est un attracteur si :

- Pour tout voisinage U de A, il existe un voisinage V de A tel que toute solution $x\,(t,x_0) = \phi_t(x_0)$ restera dans U si $x_0 \in$ V
- $\bigcap_{t\geq 0} \phi_t(V)$ = A.
- Il existe une orbite dense dans A.

1.7.2 Les différents types d'attracteurs

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

a Attracteurs réguliers

Les attracteurs réguliers caractérisent l'évolution de systèmes non chaotiques, et peuvent être de trois sortes [4] :

1. **Les points fixes**

C'est le cas le plus courant, et le plus simple d'attracteur, dans lequel le système évolue vers un état de repos (point).Notons que seuls les puits peuvent être des attracteurs. Les autres types de points fixes ont en effet toujours au moins une "voie de sortie" : à chaque valeur propre du Jacobien de partie réelle positive est associé un vecteur propre qui pointe dans une direction où la trajectoire de phase s'éloigne du point fixe.

2. **Le cycle limite périodique**

Il peut arriver que la trajectoire de phase se referme sur elle-même. L'évolution temporelle est alors cyclique, le système présentant des oscillations permanentes. Dans un système physique dissipatif, cela exige la présence d'un terme de forçage dans les équations qui vient compenser en moyenne les pertes par dissipation.

3. **Le cycle limite pseudo périodique**

C'est presque un cas particulier du précédent. Le système présente au moins deux périodes simultanées dont le rapport est irrationnel. La trajectoire de phase ne se referme pas sur elle-même, mais s'enroule sur une variété de dimension 2.

b L'attracteur étrange ou chaotique

Le terme attracteur étrange est introduit pour la première fois par Ruelle et Takens (1971), pour appeler un ensemble limite d'un système dynamique qui n'est pas une variété et par suite il n'est pas un point fixe, un cycle limite, un tore invariant ou autre. La notion de l'attracteur étrange indique la nature du modèle qui est un objet mathématique bien défini [7].

Définition

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:

$$\begin{cases} \frac{dx}{dt} = a(y - x) \\ \frac{dy}{dt} = bx - y - xz \\ \frac{dz}{dt} = xy - cz \end{cases} \quad (1.11)$$

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

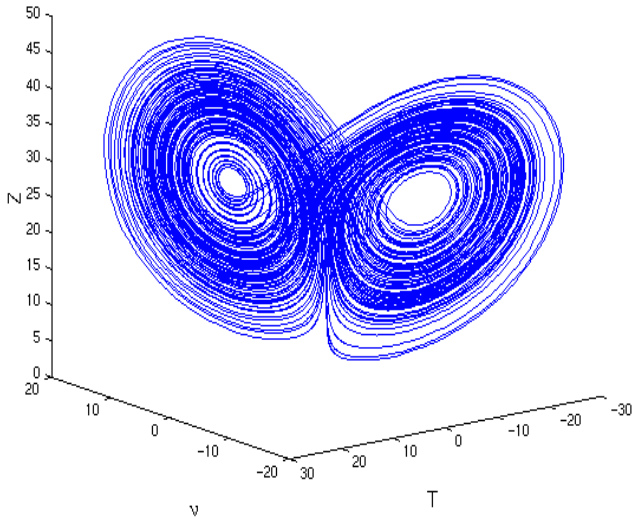


Figure 1.3. Attracteur de Lorenz : (a, b, c) = (10, 28, $\frac{8}{3}$).

- **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:

$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + xz - cz \end{cases} \quad (1.12)$$

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 [8].

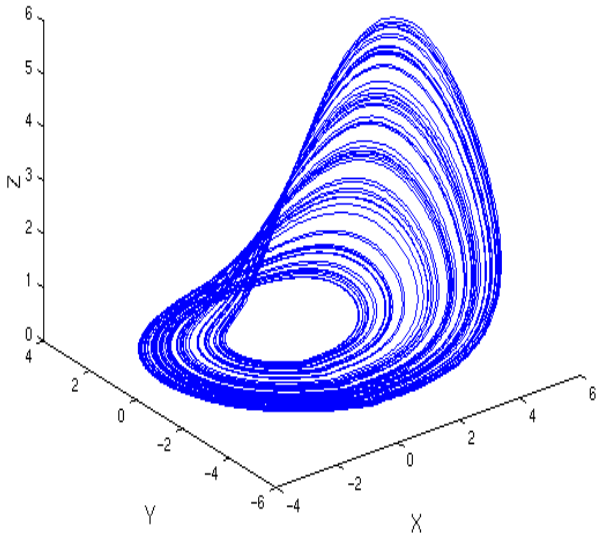


Figure 1.4. Attracteur de Rössler : (a, b, c) = (0.398, 2, 4).

- **L'attracteur de Hénon**

Le système de Hénon est un modèle proposé en 1976 par le mathématicien Michel Hénon. Ce modèle vise à rendre compte de certaines propriétés d'une section de Poincaré (définie plus loin) de l'attracteur de Lorenz. Il s'agit d'un système qui introduit des itérations dans le plan. Ces itérations sont définies par les relations suivantes.

$$\begin{cases} x_{k+1} = a - x_k^2 + by_k \\ y_{k+1} = x_k \end{cases} \quad (1.13)$$

Dans cet exposé, on prendra pour conditions initiales $(X_0, Y_0) = (1, 0)$, conditions initiales contenues dans le bassin d'attraction présenté plus loin. De plus, on prendra les valeurs suivantes: $a = 1.4$ et $b = 0.3$. Ces valeurs furent proposées par Michel Hénon et permettent d'observer un comportement chaotique [8].

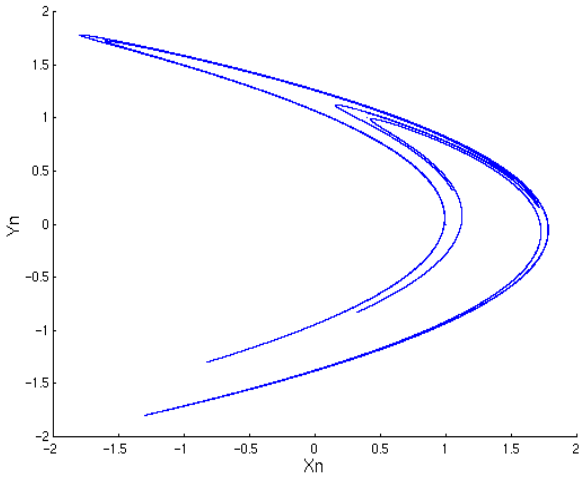


Figure 1.5. Attracteur de Hénon : (a, b) = (1.4, 0.3)

1.7.3 Exposant de Lyapunov

L'exposant de Lyapunov sert à mesurer le degré de stabilité d'un système. Un système sensible à de très petites variations de la condition initiale aura un exposant positif (système chaotique). En revanche, l'exposant est négatif si le système n'est pas sensible à des petites variations des conditions initiales, les trajectoires se rapprochent et on perd donc l'information sur les conditions initiales. Un système de dimension *n* possède *n* exposants de Lyapunov qui mesurent le taux de divergence suivant un des axes de l'espace de phase. L'apparition du chaos exige l'existence d'un exposant positif selon au moins un axe, tout en rendant compte que la somme des exposant est négative (respectivement nulle) pour les systèmes dissipatifs (respectivement conservatifs).

Un exposant de Lyapunov positif (respectivement négatif) selon une direction indique, qu'une divergence entre deux trajectoires voisines augmente (respectivement diminue) exponentiellement avec le temps [3].

1.7.4 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 (1.14)

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 est utilisée pour désigner dans un sens large, toute modification qualitative du comportement d'un système dynamique suite à la variation de l'un de ses paramètres. 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).

4 -Bifurcation de Hopf (sous-critique ou sur-critique).

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].

1.7.5 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 [3].

Exemple 1:

Le diagramme de bifurcation du système de Rössler écrit par l'équation (1.4) [3] :

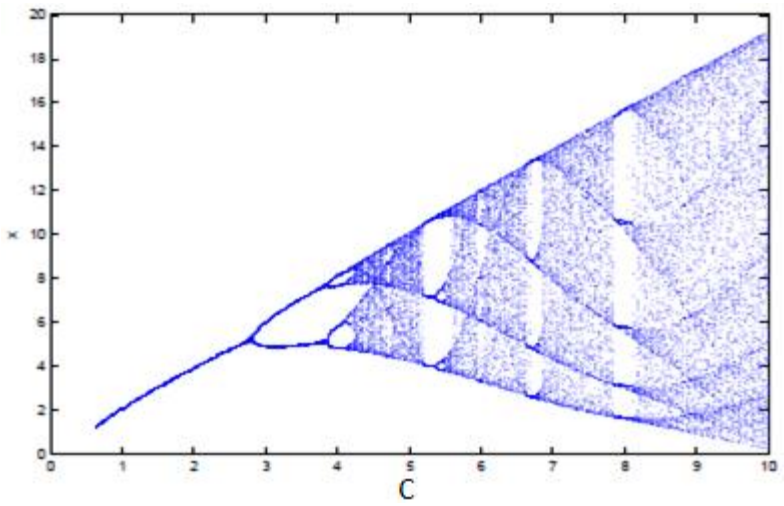


Figure 1.6. Diagramme de bifurcation du système de Rössler

Exemple 2:

Le diagramme de bifurcation du système de Hénon écrit par l'équation (1.13) [3] :

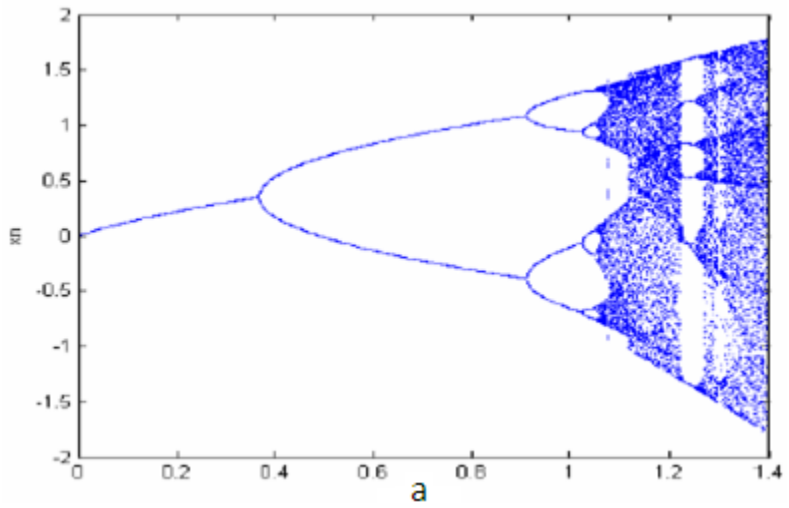


Figure 1.7. Diagramme de bifurcation du système de Hénon

1.8 Conclusion

Dans ce chapitre nous anons présenté quelques notions et définitions de base de systèmes dynamiques et chaos, telles que les différents types de système dynamique, les systèmes autonomes et non autonomes, section de Poincaré, fonctions de Lyapunov.

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

Chapitre 2 Résolutions des systèmes dynamiques chaotiques par la méthode de Runge-Kutta

2.1 Introduction

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 telles que (la méthode d'Euler, la méthode de Newmark la méthode de Runge Kutta, la méthode de dichotomie ....) [9].

Dans ce chapitre, nous présentons quelques généralités sur la résolution des systèmes dynamiques par les méthodes numériques. Parmi ces méthodes, nous appliquons la méthode de Runge Kutta d'ordre 4 à l'exemple du système de Rössler.

2.2 Les Equations Différentielles

Les équations différentielles peuvent être partagées en deux classes principales : les équations différentielles linéaires et les équations différentielles non linéaires [10].

2.2.1 Les équations différentielles linéaires

Une équation différentielle linéaire se présente comme une relation entre une ou plusieurs fonctions inconnues et leurs dérivées, de la forme suivante :

$$a_n * y^{(n)} + \cdots + a_2 * y^{(2)} + a_1 * y^{(1)} + a_0 * y = b \quad (2.1)$$

Où a_n, a_2, a_1, a_0, b sont des fonctions numériques continues.

2.2.2 Les équations différentielles non linéaires

Contient des termes non linéaires en $y(t)$ et/ou en ses dérivées L'exemple suivant représentent une équation différentielle non linéaire.

$$\dot{y} = \frac{dy}{dt} = f(x, u, t) \quad (2.2)$$

f : Fonction quelconque.

Dans ce chapitre on s'intéressera, particulièrement, aux équations différentielles non linéaires. Par ailleurs, il faut noter également, que l'on dispose des méthodes pour résoudre analytiquement les équations différentielles non linéaires [10].

2.3 Méthodes numérique de Résolutions des équations différentielles non linéaires

2.3.1 la méthode de Taylor

La formule de Taylor, du nom du mathématicien Brook Taylor qui l'´établit en 1715, permet l'approximation d'une fonction plusieurs fois dérivable au voisinage d'un point par un polynôme dont les coefficients dépendent uniquement des dérivées de la fonction en ce point. La première ´étape est la formule suivant :

$$f(x_0 + h) = f(x_0) + hf^{(1)}(x_0) + \mathcal{E}(h) \quad (2.3)$$

Qui montre que, si f est d´dérivable, alors f est approchée par un polynôme de degré 1 (une droite). On se pose la question du comment faire pour augmenter le degré.

Soient I un intervalle de $\mathcal{R}$, x_0 un point intérieur à I , et $f:\mathrm{I} \longrightarrow \mathcal{R}$ On dit qu'une fonction est de classe C^{n} sur I si elle est n fois dérivable sur I, et si sa dérivée n -ieme est continue sur I. Pour tout $h \in \mathcal{R}$ tel que $(x_0 + h) \in \mathrm{I}$ on peut écrire :

$$f(x_0 + h) = f(x_0) + hf^{(1)}(x_0) + \frac{h^2}{2!}f^{(2)}(x_0) + \cdots + \frac{h^n}{n!}f^{(n)}(x_0) + h^n\mathcal{E}(h) \qquad (2.4)$$

Où $\mathcal{E}(h)$ est une fonction qui tend vers 0 quand h tend vers 0 [11].

2.3.2 Méthode de Dichotomie (Bissection)

Le mot dichotomie (dicho = deux, tomie =coupe), Cette méthode repose sur le constat que si $f(x)$ est continue et que le produit $f(a) * f(b) < 0$. Alors la fonction f s'annule au moins une fois sur l'intervalle $[a, b]$.

Les différentes étapes de la méthode peuvent être résumées comme suit [12] :

1. Choisir un intervalle $[x_0 = a, x_1 = b]$ tel que$f(a) * f(b) < 0$.
2. Calculer la valeur de la fonction au point $x_2 = \frac{a+b}{2}$.
3. On retient comme nouvel intervalle $[x_1, x_2]$ ou $[x_2, x_1]$.
4. Répéter les étapes 2 et 3 jusqu'à l'obtention de la précision désirée, c'est-à-dire jusqu'à ce que $|f(x_2)| < \epsilon$, où $|x_{k+1} - x_k| < \epsilon$, ϵ étant la précision désirée.

Après k itérations, la précision sur la racine s vaut :

$$|x_k - s| < \frac{b-a}{2^{k+1}} \qquad (2.5)$$

Il est donc possible d'estimer à l'avance le nombre d'itérations nécessaires pour approcher s avec une précision donnée. Si on désire une tolérance τ sur la racine. Alors l'inégalité précédente donne le nombre minimal d'itérations n qui doit vérifier $\frac{b-a}{2^{k+1}} < \tau$.

Donc :

$$k > \frac{\log(b-a)-log2\tau}{log2} \tag{2.6}$$

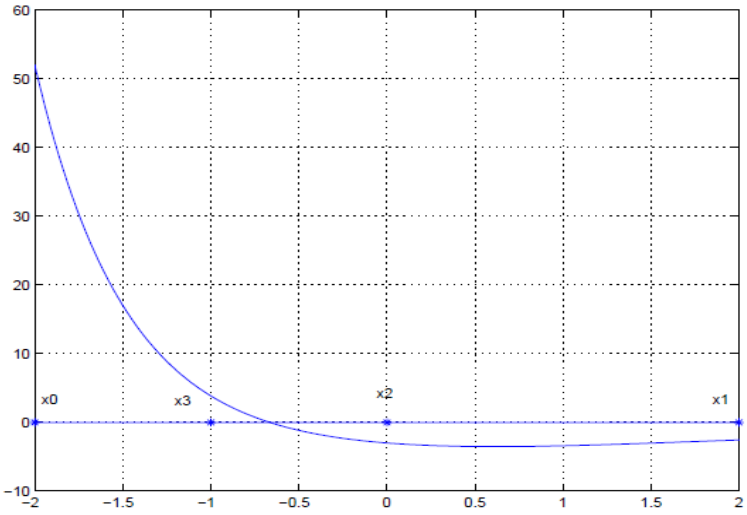


Figure 2.1 la méthode de dichotome.

2.3.3 La méthode d'éluer

Leonhard Paul Euler (1707-1783) : mathématicien et physicien Suisse, a trouvé une approximation simple de la dérivée.

Le problème est de déterminer une fonction dont on connaît l'expression de la dérivée $f^{(1)}(x)$ et la valeur initiale $f(x_0) = y_0$.

Soit $y = f(x)$ la fonction considérée (supposée continue et dérivable) La valeur de la dérivée au point x_0 est obtenue par :

$$f^{(1)}(x_0) = lim_{h\to 0} \frac{f(x_0+h)-f(x_0)}{h} \tag{2.7}$$

On déduit en première approximation dans le cas où h reste petit la valeur de $f(x_0 + h)$

A partir de la valeur de $f(x_0)$ et sa dérivée au point x_0 :

$$f(x_0 + h) = f(x_0) + h * f^{(1)}(x_0) \tag{2.8}$$

En posant $x_1 = x_0 + \mathrm{h}$, on obtient :

$$y_1 = \mathrm{f}(x_1) = \mathrm{f}(\mathrm{x}_0) + h * f^{(1)}(x_0) \tag{2.9}$$

De même

$$f(x_1 + h) = \mathrm{f}(\mathrm{x}_1) + h * f^{(1)}(x_1) \tag{2.10}$$

En posant $x_2 = x_1 + \mathrm{h}$, on obtient :

$$y_2 = \mathrm{f}(x_2) = \mathrm{f}(\mathrm{x}_1) + h * f^{(1)}(x_1) \tag{2.11}$$

En généralisant, on obtient :

$$y_n = \mathrm{f}(\mathrm{x}_\mathrm{n}) = \mathrm{f}(\mathrm{x}_{\mathrm{n}-1}) + h * f^{(1)}(x_{n-1}) \tag{2.12}$$

Pour h voisin de 0, les points $M_n(\mathrm{x}_\mathrm{n}, \mathrm{y}_\mathrm{n})$ sont proches de la courbe de la primitive f de $f^{(1)}$ vérifiant la condition initiale [13].

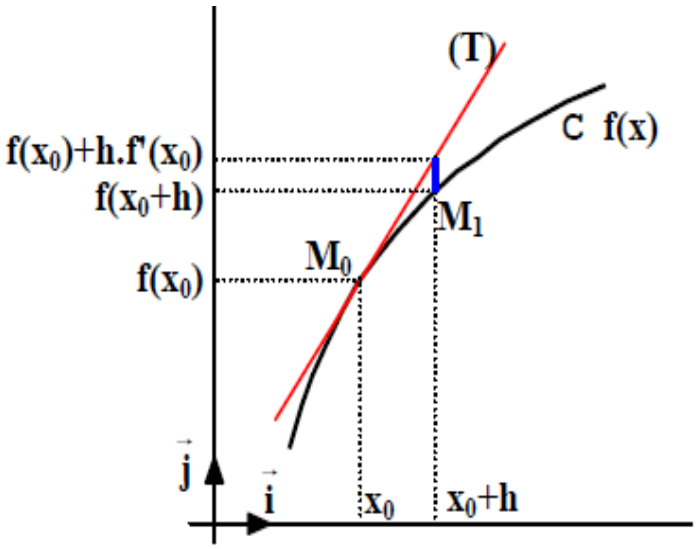


Figure 2.2. La méthode d'éluer

Soit C : la représentation graphique de la fonctionf(x).

(T) : est la tangente de $\mathrm{f}(\mathrm{x}_0)$ au point M_0

M_0 : obtenir par $y(M_1) = f(x_1) = f(x_0 + h)$

Il est donc nécessaire que l'incrément h soit le plus petit possible pour que la fonction calculée soit la plus proche possible de la primitive recherchée.

2.3.4 La méthode de Runge kutta

Carl David Tolmé Runge (1856- 1927) et Martin Wilhelm Kutta (1867- 1944) : mathématiciens allemands ont développé une des méthodes de résolution numérique pour les équations différentielles les plus utilisées, 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) [9] :

- Runge-Kutta d'ordre 1 (RK-1),
- Runge-Kutta d'ordre 2 (RK-2),
- Runge-Kutta d'ordre 3 (RK-3),
- Runge-Kutta d'ordre 4 (RK-4),

a Principe de la méthode

Soit l'équation différentielle du premier ordre

$$y^{(1)} = \mathcal{F}(\mathrm{x},\mathrm{y}) \tag{2.13}$$

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

Un développement en série de Taylor de l'équation (2.1) donne [14] :

$$\mathrm{y}_{n+1} = \mathrm{y}_n + h * y_n^{(1)} + \frac{h^2}{2!} * y_n^{(2)} + \frac{h^3}{3!} * y_n^{(3)} + \frac{h^4}{4!} * y_n^{(4)} + \cdots + \frac{h^n}{n!} * y_n^{(n)} + O(h^{n+1}) \tag{2.14}$$

Ou

$\mathrm{y}_n = y(x_n)$ et $\mathrm{y}_{n+1} = y(x_{n+1}) = (x_n + h)$

On va ensuite exprimer les dérivées $y^{(1)}(\mathrm{x})$ et $y^{(2)}(\mathrm{x})$ en fonction de $\mathcal{F}$(x, y) et de ses dérivées partielles par rapport à x et y.

On sait que : $y^{(1)} = \mathcal{F}(x, y)$

$$y^{(2)}(\mathrm{x}) = \frac{d}{dx}[\mathcal{F}(x, y(x))] = \frac{d\,\mathcal{F}(x,y(x))}{dx} + \frac{d\,\mathcal{F}(x,y(x))}{dy} * \frac{d\,(y(x))}{dx} \quad (2.15)$$

$$y^{(2)}(\mathrm{x}) = \frac{d}{dx}[\mathcal{F}(x, y(x))] = \frac{d\,\mathcal{F}(x,y)}{dx} + \frac{d\,\mathcal{F}(x,y)}{dy} * \frac{d\,(y(x))}{dx} \quad (2.16)$$

$$y^{(2)}(\mathrm{x}) = \frac{d}{dx}[\mathcal{F}(x, y(x))] = \frac{d\,\mathcal{F}(x,y)}{dx} + \frac{d\,\mathcal{F}(x,y)}{dy} * \mathcal{F}(x, y(x))] \quad (2.17)$$

On substitue alors ces deux expressions dans le développement de Taylor précédent d'ordre 2 et on obtient :

$$\mathrm{y}_{n+1} = \mathrm{y}_n + h * \mathcal{F}(\mathrm{x}, \mathrm{y}) + \frac{h^2}{2!} * [\frac{d\,\mathcal{F}(x,y)}{dx} + \frac{d\,\mathcal{F}(x,y)}{dy} * \mathcal{F}(x, y(x))] + O(h^3) \quad (2.18)$$

Runge et Kutta ont l'idée d'écrire une combinaison linéaire de deux valeurs de la fonction $\mathcal{F}$ pour exprimer y_{n+1} :

$$\mathrm{y}_{n+1} = \mathrm{y}_n + \mathrm{A} * h * k_1 + \mathrm{B} * h * k_2 \quad (2.19)$$

$$\begin{cases} k_1 = \mathcal{F}(x, y(x)) \\ k_2 = \mathcal{F}(x + P * h\,, y(x) + Q * h * k_1\,) \end{cases} \quad (2.20)$$

Ils ont ensuite appliqué à k_2 le développement de Taylor (à l'ordre 1) d'une fonction de deux variables :

$$k_2 = \mathcal{F}(x, y) + P * h * \frac{d\,\mathcal{F}(x,y)}{dx} + Q * h * k_1 * \frac{d\,\mathcal{F}(x,y)}{dy} + O\,(h) \quad (2.21)$$

$$k_2 = \mathcal{F}(x, y) + P * h * \frac{d\,\mathcal{F}(x,y)}{dx} + Q * h * \frac{d\,\mathcal{F}(x,y)}{dy} * \mathcal{F}(x, y(x)) + O\,(h) \quad (2.22)$$

En reportant les expressions de k_1 et k_2 dans (2.10)

$$\mathrm{y}_{n+1} = \mathrm{y}_n + (\mathrm{A} + B) * h * \mathcal{F}(x,y) + B * P * h^2 * \frac{d\,\mathcal{F}(x,y)}{dx} + B * Q * h^2 * \frac{d\,\mathcal{F}(x,y)}{dy} * \mathcal{F}\big(x, y(x)\big) + O(h) \quad (2.23)$$

En identifiant terme à terme les deux développements faisant intervenir les dérivées partielles, on trouve :

$$\begin{cases} A + B = 1 \\ B * P = \frac{1}{2} \\ B * Q = \frac{1}{2} \end{cases} \quad (2.24)$$

Ainsi, avec des valeurs de A, B, P et Q vérifiant les relations ci-dessus, la solution de (2.9) fournie par la méthode de Runge-Kutta d'ordre 2 aura la même approximation que celle fournie par le développement en série de Taylor.

Puisqu'on n'a que 3 équations pour 4 inconnues, on peut choisir arbitrairement un des coefficients. On prend alors A = 0, et ainsi B = 1, et P = Q = 1/2. Dans ce cas, le schéma correspondant est :

- Pour Runge kutta d'ordre 2 :

$$\begin{cases} k_1 = \mathcal{F}(x,y) \\ k_2 = \mathcal{F}\left(x + \frac{h}{2}, y + \frac{h}{2} * k_1\right) \end{cases} \quad (2.25)$$

- Pour Runge kutta d'ordre 4 :

$$\begin{cases} k_1 = \mathcal{F}(x,y) \\ k_2 = \mathcal{F}\left(x + \frac{h}{2}, y + \frac{h}{2} * k_1\right) \\ k_3 = \mathcal{F}\left(x + \frac{h}{2}, y + \frac{h}{2} * k_2\right) \\ k_4 = \mathcal{F}(x + h, y + h * k_3) \end{cases} \quad (2.26)$$

Pour une approximation du type :

$$\mathrm{y}_{n+1} = \mathrm{y}_n + \frac{1}{6} * (k_1 + 2 * k_2 + 2 * k_3 + k_4) \quad (2.27)$$

A partir de la valeur initial $\mathrm{y}(\mathrm{x}_0) = y_0$, nous déduirons de pas en pas $y_1, y_2, y_3 \ldots \mathrm{y}_n$.

L'erreur systématique de troncature est de l'ordre de h^5, Où

h : Le pas de discrétisation en x (pas de calcul).

k_1 : est la pente au début de l'intervalle.

k_2 : est la pente au milieu de l'intervalle, en utilisant la pente k_1 pour calculer la valeur de y.

k_3 : est de nouveau la pente au milieu de l'intervalle, mais obtenue cette fois en utilisant la pente k_2 pour calculer y.

k_4 : est la pente à la fin de l'intervalle, avec la valeur de y calculée en utilisant k_3.

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

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'ordre de h^4 [15].

Remarque : Par fois la méthode RK2 est aussi appelée « méthode de la tangente améliorée» ou encore « méthode d'Euler modifiée ».

La méthode de Runge Kutta d'ordre 4 définit deux suites [16] :

1. Une première qui permet de définir les valeurs de x.
    - Le terme initial : x_0.
    - La relation de récurrence : $x_{n+1} = x_n + h$.
2. une deuxième qui permet d'évaluer les valeurs de y.
    - Le terme initial : y_0.
    - La relation de récurrence : $y_{n+1} = y_n + \frac{1}{6} * (k_1 + 2 * k_2 + 2 * k_3 + k_4)$

b 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 [17].

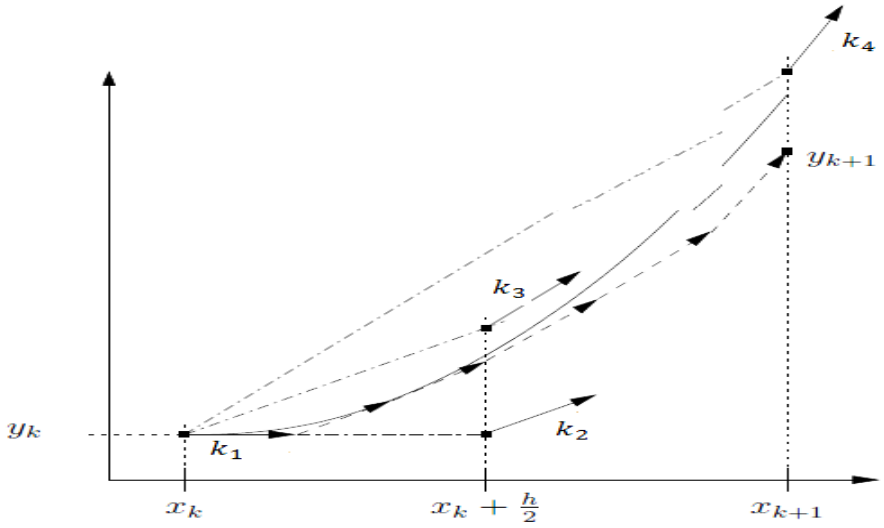


Figure 2.3. *Schéma de Runge Kutta d'ordre 4*

c Exemple de système des équations différentielles non linaire

Nous nous proposons d'étudier le système dynamique non linaire de Rössler composé par le système d'équations différentielles non linéaires suivant:

$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + z\,x - z\,c \end{cases} \quad (2.28)$$

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 = 0.2; b = 0.2; c = 5.7 avec $(x_0, y_{0,} z_0)$ = (0, 0, 0).

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 :

$$\begin{cases} k_{1x} = h * dx(x, y, z) \\ k_{1y} = h * dy(x, y, z) \\ k_{1z} = h * dz(x, y, z) \end{cases} \quad (2.29)$$

Pour k_2 :

$$\begin{cases} k_{2x} = h * dx(x + 0.5 * k_{1x}, y + 0.5 * k_{1y}, z + 0.5 * k_{1z}) \\ k_{2y} = h * dy(x + 0.5 * k_{1x}, y + 0.5 * k_{1y}, z + 0.5 * k_{1z}) \\ k_{2z} = h * dz(x + 0.5 * k_{1x}, y + 0.5 * k_{1y}, z + 0.5 * k_{1z}) \end{cases} \quad (2.30)$$

Pour k_3 :

$$\begin{cases} k_{3x} = h * dx(x + 0.5 * k_{2x}, y + 0.5 * k_{2y}, z + 0.5 * k_{2z}) \\ k_{3y} = h * dy(x + 0.5 * k_{2x}, y + 0.5 * k_{2y}, z + 0.5 * k_{2z}) \\ k_{3z} = h * dz(x + 0.5 * k_{2x}, y + 0.5 * k_{2y}, z + 0.5 * k_{2z}) \end{cases} \quad (2.31)$$

Pour k_4 :

$$\begin{cases} k_{4x} = h * dx(x + k_{3x}, y + k_{3y}, z + k_{3z}) \\ k_{4y} = h * dy(x + k_{3x}, y + k_{3y}, z + k_{3z}) \\ k_{4z} = h * dz(x + k_{3x}, y + k_{3y}, z + k_{3z}) \end{cases} \quad (2.32)$$

Donc :

$$x_{i+1} = x + \frac{(k_{1x} + 2 * k_{2x} + 2 * k_{3x} + k_{4x})}{6} \quad (2.33)$$

$$y_{i+1} = y + \frac{(k_{1y} + 2 * k_{2y} + 2 * k_{3y} + k_{4y})}{6} \quad (2.34)$$

$$z_{i+1} = z + \frac{(k_{1z} + 2 * k_{2z} + 2 * k_{3z} + k_{4z})}{6} \quad (2.35)$$

Application numérique : avec les conditions initiales a = 0.2; b = 0.2; c = 5.7.

$(x_0, y_{0,} z_0)$ = (0, 0, 0). On obtient les résultats suivants :

i	x_i	y_i	z_i
0	1.0e-003 * -0.0082	1.0e-005 *-0.0025	0.0020
1	1.0e-003 *-0.0355	1.0e-005 *-0.0212	0.0038
2	1.0e-003 *-0.0807	1.0e-005 *-0.0747	0.0055
3	1.0e-003 *-0.1430	1.0e-005 *-0.1809	0.0072
4	1.0e-003 *-0.2214	1.0e-005 *-0.3563	0.0087

Tableau 2.1. Les résultats de la méthode RK4 appliqué sur le système de Rössler.

La figure 2.4 ci-dessous montre l'attracteur du système de Rössler.

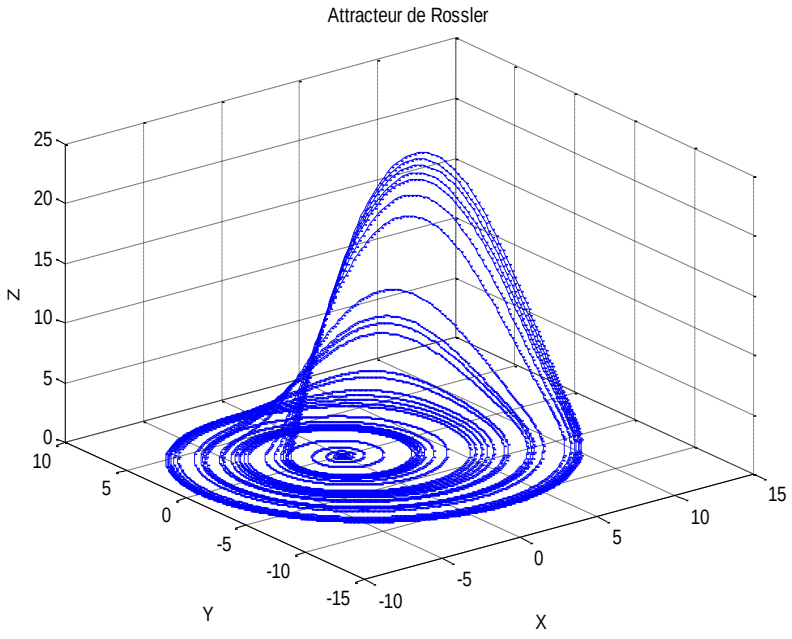


Figure 2.4. Attracteur de Rössler.

2.4 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 dans un espace 3 dimensions.

La figure 2.2 présente l'organigramme de la simulation du système de Rössler par la méthode de Runge Kutta d'ordre 4.

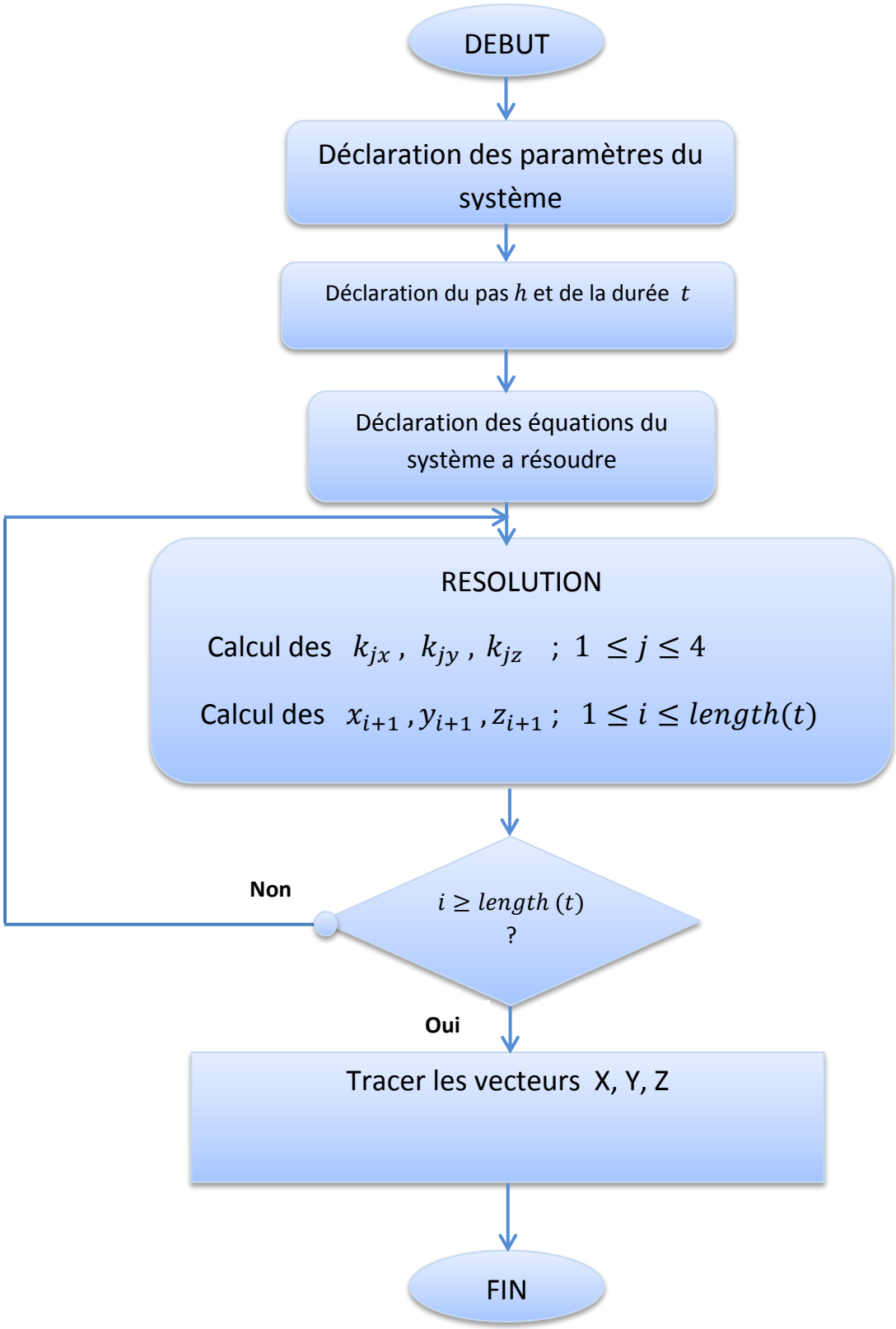


Figure 2.2. Organigramme de la méthode RK4 appliqué sur le système de Rössler

2.5 Les avantages d'utilisation des méthodes Runge Kutta

Les formules de Runge Kutta sont parmi les plus utilisées car elles ont les avantages suivants [18] :

- elles sont faciles à programmer.
- elles sont en générale stables, en tout cas pour les fonctions courantes de la physique.
- la largeur du pas peut être modifiée sans difficultés.
- on peut adapter le pas pour obtenir une précision souhaitée.
- surtout, elles « démarrent » toutes seules : la connaissance des valeurs initiales suffit à intégrer l'équation différentielle.
- la précision des résultats numériques obtenus.

2.6 Les inconvénients d'utilisation des méthodes Runge Kutta

L'utilisation de la méthode de Runge–Kutta pose un problème concerne sur le temps d'exécution (ou de calcul). [18]:

2.7 Conclusion

Dans ce chapitre, nous avons rappelé quelques notions relatives aux équations différentielles. Différentes méthodes numériques pour résoudre les systèmes dynamiques non linéaires ont été données. Ainsi que la simulation de la méthode sur Matlab.

Dans le chapitre suivant nous allons présenter une implémentation de cette méthode dans la carte DSP, afin de comparer les résultats de l'implémentation avec celle de la simulation.

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 au système de Rössler 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 Domaine d'application de DSP

Les applications des DSP sont utilisé dans nombreuses domaines voici quelques exemples des applications les plus courantes suivants [19]:

- **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 Présentation des DSP

Un DSP est un type particulier de microprocesseur. Il se caractérise par le fait qu'il intègre un ensemble de fonctions spéciales. Ces fonctions sont destinées à le rendre particulièrement performant dans le domaine du traitement numérique du signal.

Comme un microprocesseur classique, un DSP est mis en œuvre en lui associant de la mémoire (RAM, ROM) et des périphériques. Il se présente donc généralement sous la forme d'un microcontrôleur intégrant, selon les marques et les gammes des constructeurs, de la mémoire, des timers, des ports séries synchrones rapides, des contrôleurs DMA, des ports d'E/S divers [19].

3.4 Architecture des DSP

L'architecture d'un DSP est un élément important qui conditionne directement ses performances. Il existe deux types de structures fondamentales dites Von Neuman et Harvard [20].

3.4.1 Structure de Von Neuman

La structure de Von Neuman consiste à mettre les données et le programme dans la même zone mémoire, et donc utiliser un seul bus de données et d'adresses pour les données et le programme ; on ne peut lire, en un seul cycle d'horloge, qu'une donnée ou une instruction [20].

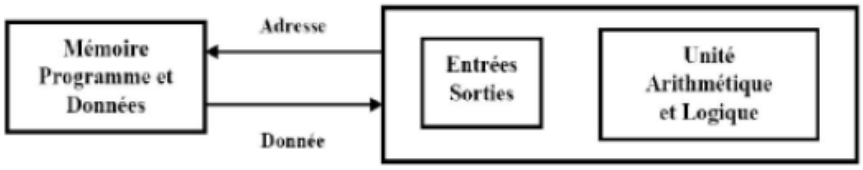


Figure 3.1. Principe de l'architecture de Von Neuman

3.4.2 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 [20].

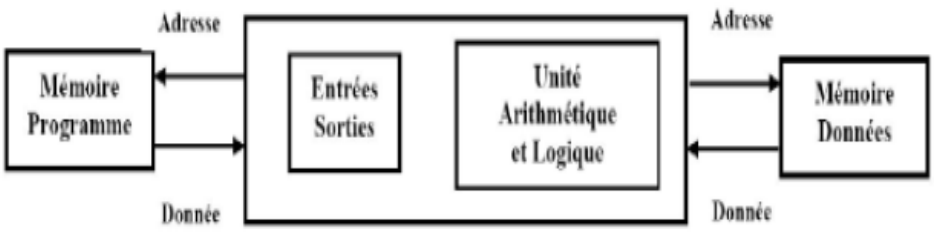


Figure 3.2. Principe de l'architecture de Harvard

3.5 les formats des données utilisés dans les DSP

La classification la plus générale des DSP est selon le format, format virgule fixe et format virgule flottante. Les DSP sont conçus pour travailler avec l'un de ces formats. Toutefois, ils peuvent faire des calculs avec un deuxième format mais d'une façon moins efficace [21].

3.5.1 Les DSP à virgules fixes

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 bits 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 bits forts, s'il n'y a plus de registres libres lors d'une étape de calcul [21].

3.5.2 Les DSP à virgule flottante

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

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

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

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) [21].

3.6 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 [21].

3.6.1 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 [21].
L'architecture générale est donnée par la figure 2.3 ci-dessous :

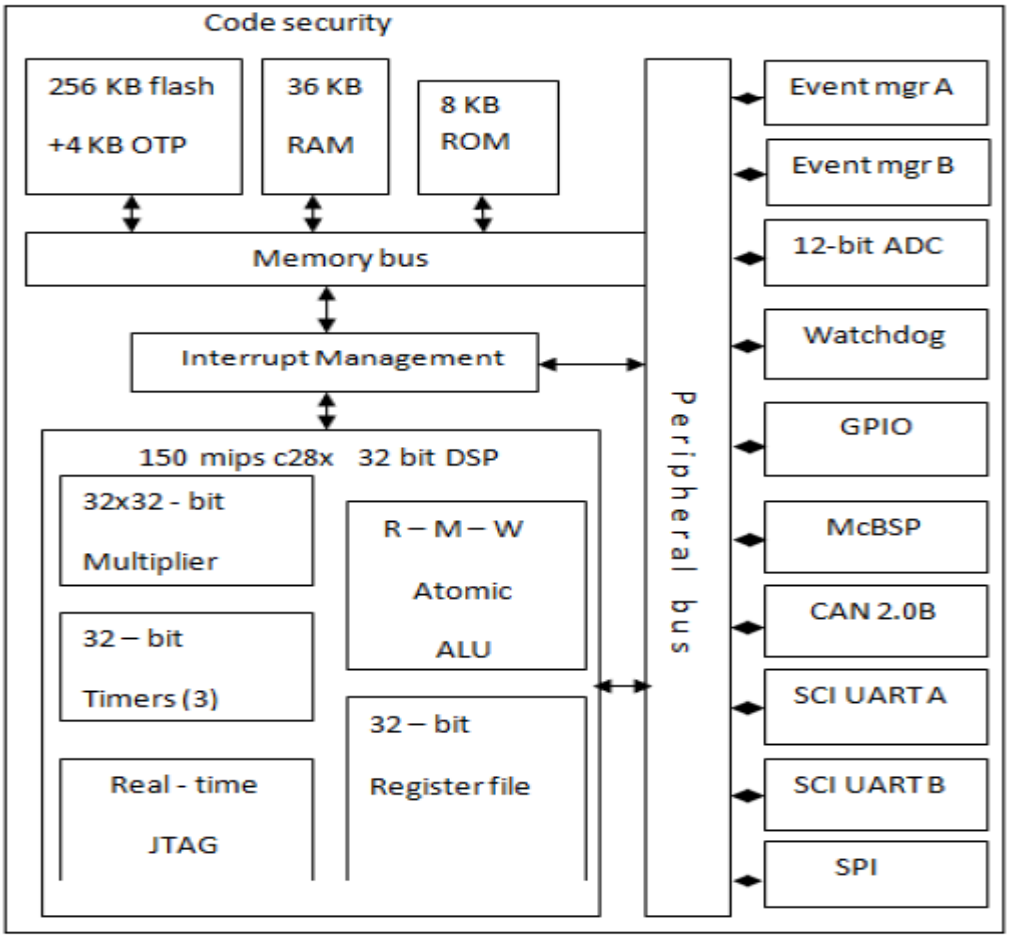


Figure 3.3. Schéma fonctionnel du DSP (TMS320F2812)

3.6.2 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) [21].

3.7 Description du processeur

3.7.1 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 [21].

3.7.2 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 [19].

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 [19].

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 [19].

d Chien de Grade (Watchdog)

Le TMS320F2812 possède un chien de garde. L'utilisateur doit régulièrement réinitialiser dans son programme le compteur du chien de garde, afin que ce dernier ne génère une réinitialisation du processeur. Le chien de garde peut être désactivé si nécessaire [19].

e 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 [19].

Le schéma synoptique de ce module est donné par la figure ci-dessous.

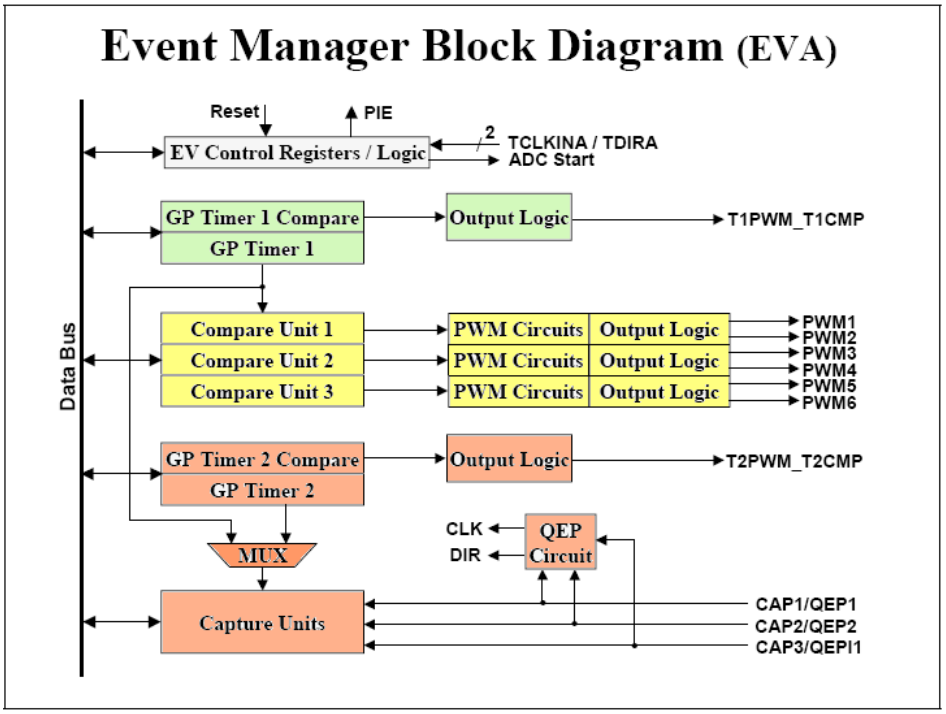


Figure 3.4. Schéma synoptique du module EVA

On peut voir que ce module est constitué par les éléments suivants [19] :

1. Deux compteurs 16 bits appelés respectivement Timer1 et Timer2 et qui sont différents de ceux de la CPU.
2. Avec chaque compteur on a associé une unité de comparaison permettant ainsi de générer des signaux PWM sur les sorties T1PWM/T1CMP et T2PWM/T2CMP.
3. unités de comparaison « Compare unit1 », « compare unit2 » et « compare unit3 » qui permettent de générer 3 signaux PWM indépendants PWM1, PWM3 et PWM5.
4. Les sorties PWM2, PWM3 et PWM5 peuvent générer des signaux complémentaires par rapport aux précédents.
5. Le module peut aussi recevoir deux signaux externes qui sont un signal d'horloge externe (TCLKINA) et la direction de comptage (TDIRA). Il peut aussi renvoyer un signal de début de conversion.
6. Les Registres associés aux Timers x (x=1,2) du module EVA sont les suivants :
   - GPTCONA: General Purpose Timer Control register A.
   - TxCNT: Timer x Counter register.
   - TxCMPR: Timer x Compare register buffer.
   - TxPR: Timer x period regsiter buffer.
   - TxCON: timer x Control register.

f La configuration des ports GPIO

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

Le C28x est 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, F et 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, F et G) [21].

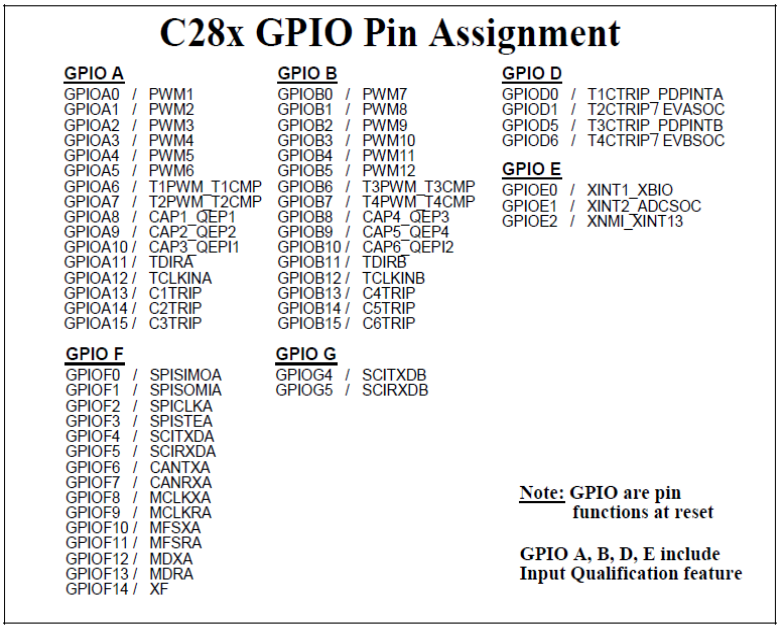
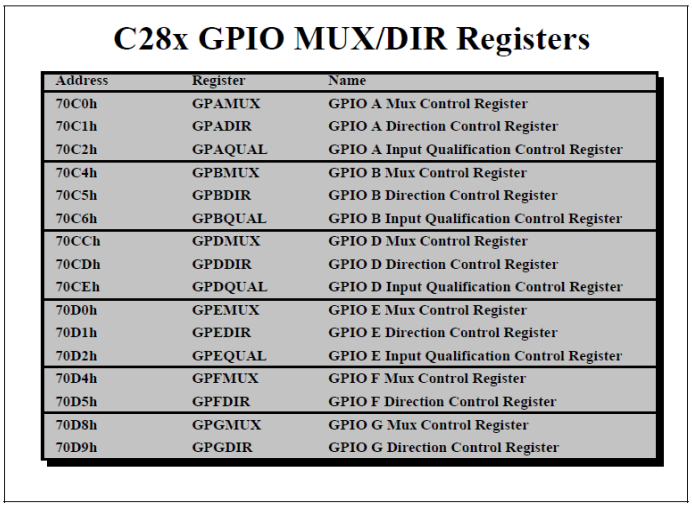


Figure 3.5. Les entrées/sorties de GPIO

g. 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 3.6. Les registres GPXMUX de GPIO

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

- Si GPXMUX.bit=0, alors la broche est configurée en mode E/S.
- Si GPXMUX.bit=1, alors la broche est 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 [21].

- SiGPxDIR.bit=0, alors la broche est configurée en tant qu'entrée.
- SiGPxDIR.bit=1, alors la broche est 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 3.7. 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. Écrit 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.

3.8 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 [21].

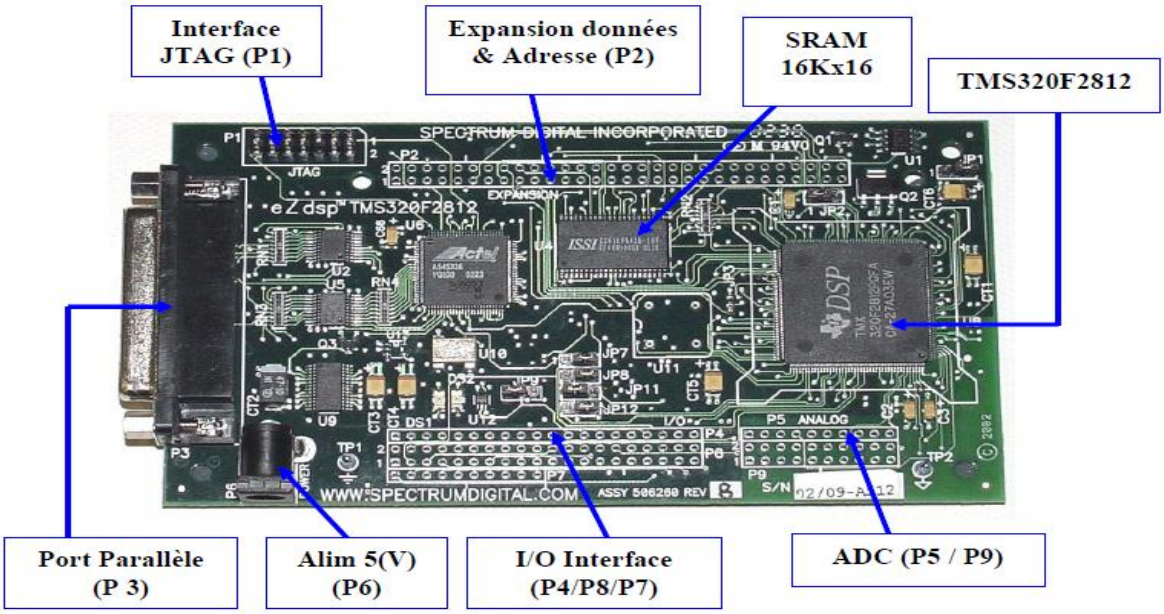


Figure 3.8. Les connecteurs sur la carte DSP

Les caractéristiques principales du DSK eZdsp F2812 :

- TMS320F2812 DSP opérant à 150MHZ.
- 128K words Flash ROM.
- 64K words on-chip-board RAM.
- IEEE 1149.1 Controller JTAG.
- Alimentation de 5V.
- Expansion de connexion.
- IEEE 1149.Q JTAG émulateur.

Le module est équipé de 5 connecteurs qui sont [19]:

- **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.
- **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 analogique-numériques.
- **P6 :** Connecteur pour l'alimentation en 5V DC de la carte.

3.9 Logiciel ''Code Composer Studio''

Code Composer Studio (CCS) est un environnement intégré de développement de code pour les DSP de Texas Instrument. Il est fourni en standard avec la carte de développement pour le DSP.CCS fournit plusieurs outils pour faciliter la construction et la mise au point des programmes de DSP. Il comprend un éditeur de code source, un compilateur de langage C/C++, un assembleur de code relocalisation, un éditeur de liens, et un environnement d'exécution qui permet de télécharger un programme exécutable sur une carte cible, de l'exécuter et de le déboguer au besoin. CCS comprend aussi des outils qui permettent l'analyse en temps réel d'un programme en cours d'exécution et des résultats produits. Il fournit aussi un environnement de gestion de fichiers qui facilite la construction et la mise au point des programmes [21].
La figure ci-dessous montre la fenêtre de l'outil de développement CCS.

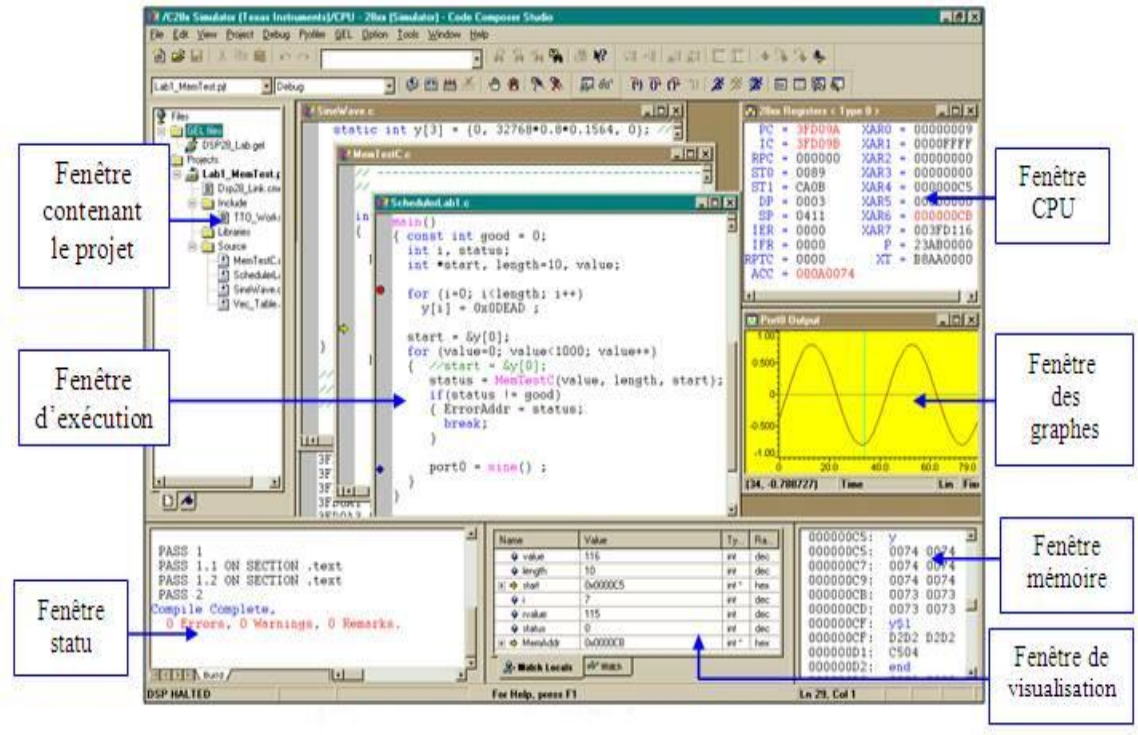


Figure 3.9. La fenêtre de l'outil de développement CCS

3.10 Implémentation de la méthode RK4 appliquée au système de Rössler sur la carte DSP

L'implémentation de la méthode RK4 appliquée au système de Rössler 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 consiste à utiliser un convertisseur numérique analogique pour visualiser les signaux sur l'oscilloscope.

On résume l'algorithme de l'implémentation de la méthode RK4 appliquée au système de Rössler dans le DSP par l'organigramme suivant :

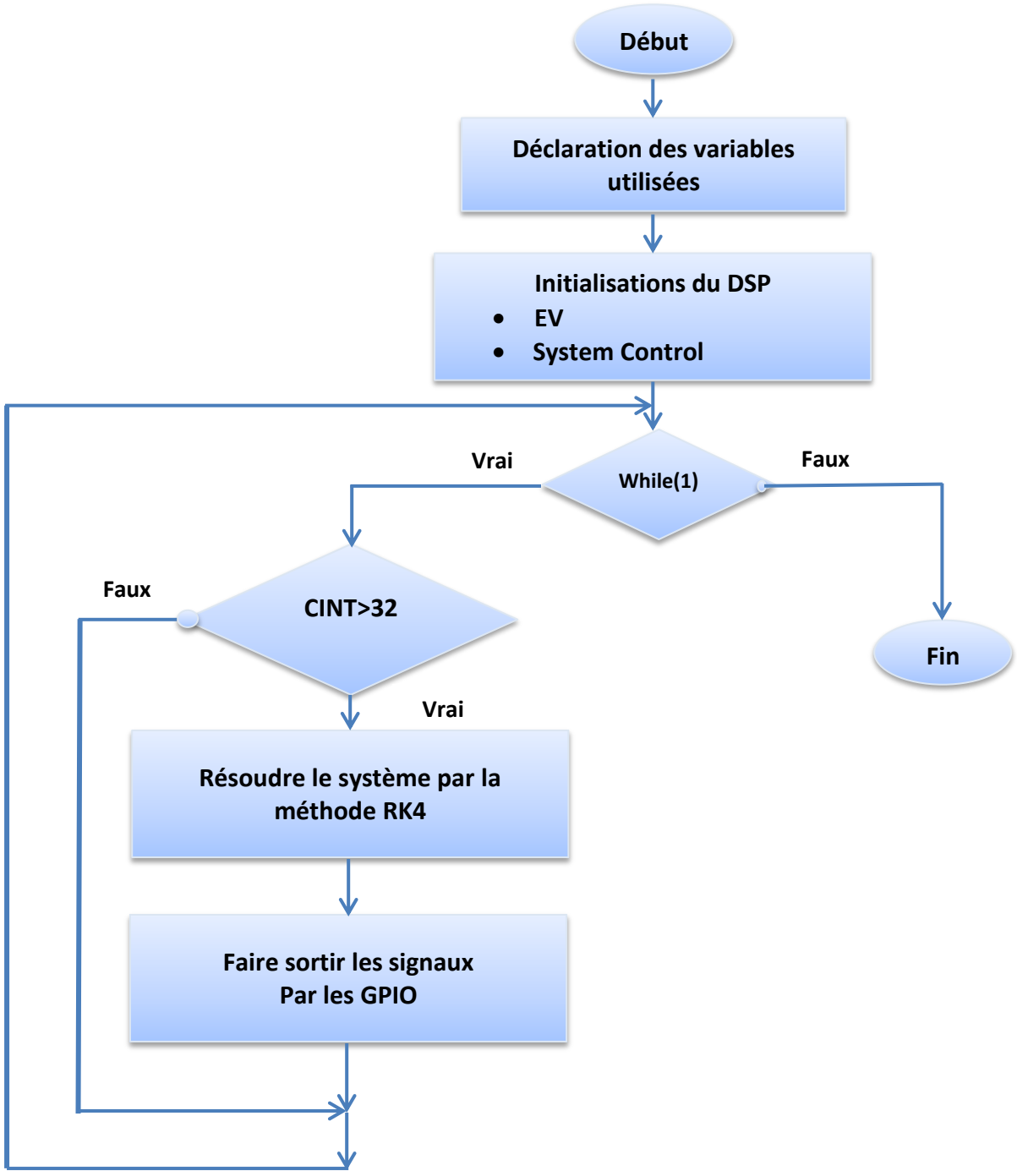


Figure 3.10. L'organigramme de la réalisation de l'implémentation

- **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.

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

$$1000\ HZ \longrightarrow 1\ ms \longrightarrow 32 \text{ interruptions}$$

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 :

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

3.11 Conclusion

Dans ce chapitre nous avons présenté quelques généralités sur la carte DSP, comme : architecture des DSP, les types de DSP, 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 au système de Rössler 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 le système de Rössler sera faite sur plusieurs étapes. On commence par la simulation à l'aide de Matlab (programme et Simulink). Ensuite, on écrit le programme 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 les résultats théoriques de la simulation et les résultats pratiques de l'implémentation du système de Rössler.

4.2 Résultats de la simulation sur Matlab

4.2.1 la simulation sur Matlab/Programme

Les figures [(4.1) (4.2) (4.3) (4.4) (4.5) (4.6) (4.7)] représentent les résultats de la méthode de Runge Kutta d'ordre 4 appliquée au système de Rössler programmée sur Matlab.

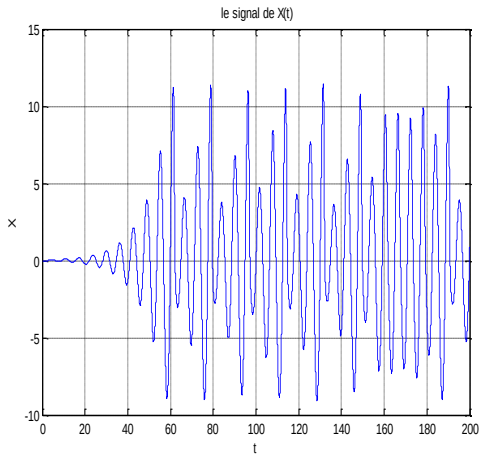


Figure 4.1. Signal X(t).

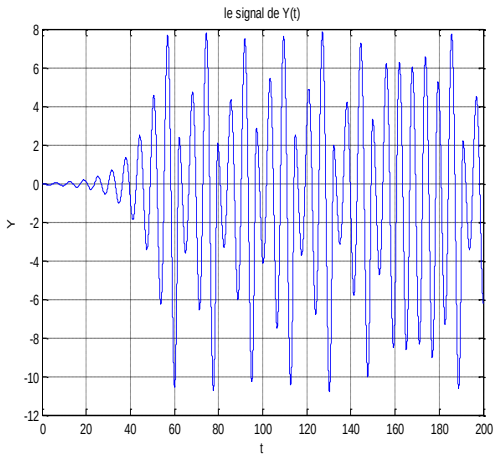


Figure 4.2. Signal Y(t).

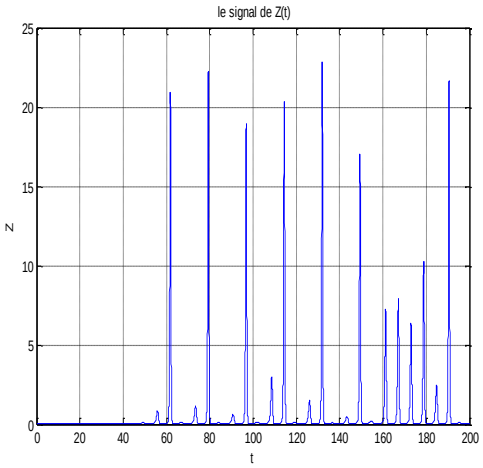


Figure 4.3. Signal Z(t).

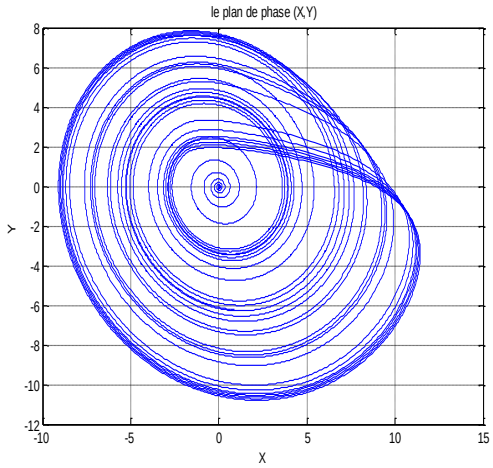


Figure 4.4. Plan de phase X.Y

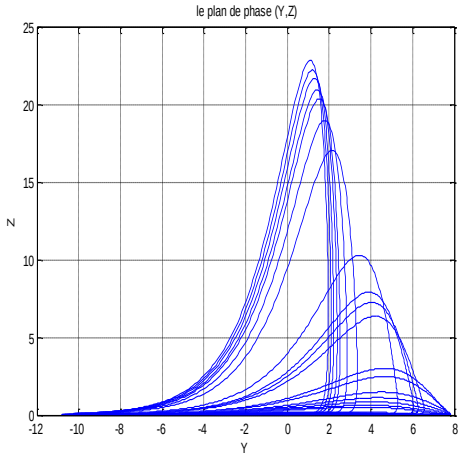


Figure 4.5. Plan de phase Y.Z

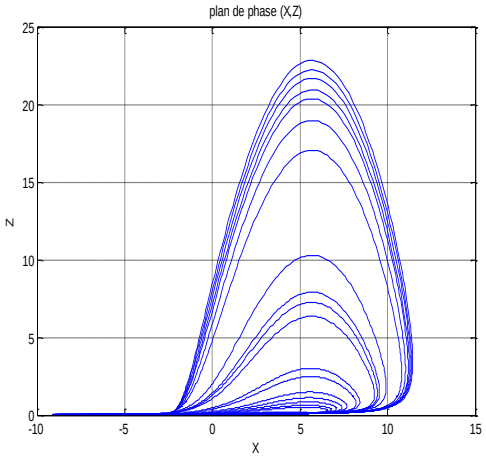


Figure 4.6. Plan de phase X.Z

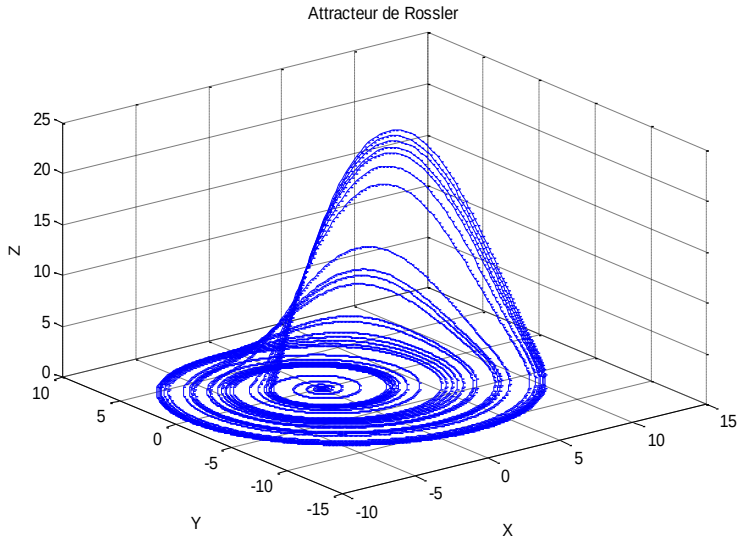


Figure 4.7. Attracteur de Rössler.

4.2.2 la simulation sur Matlab/Simulink

Les figures [(4.8) (4.9) (4.10) (4.11) (4.12) (4.13) (4.14)] représentent les résultats de la Simulation du système de Rössler sur Matlab/Simulink :

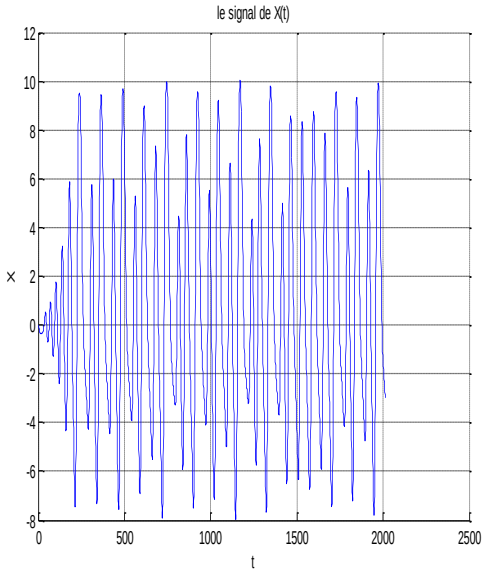


Figure 4.8. Signal X(t).

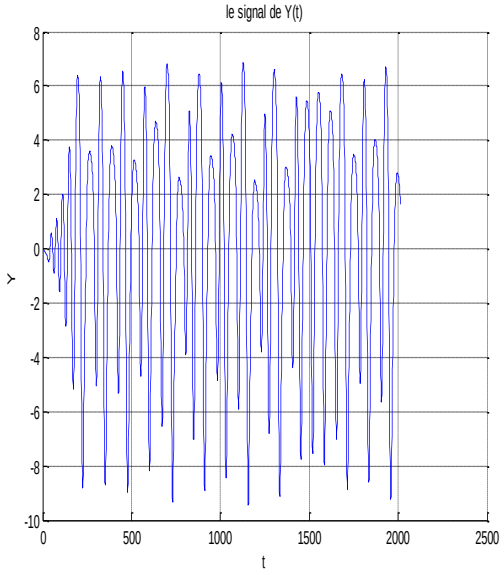


Figure 4.9. Signal Y(t).

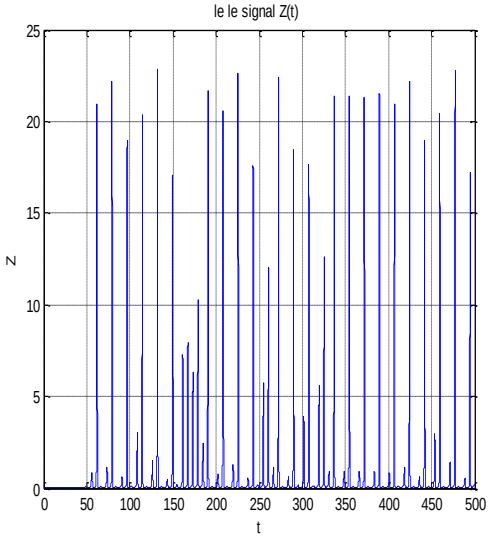


Figure 4.10. Signal Z(t).

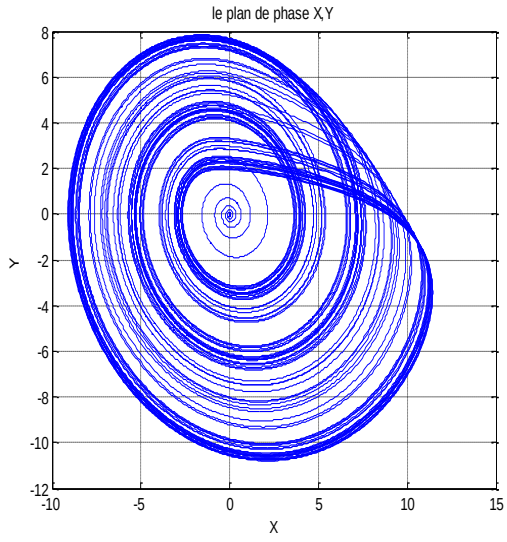


Figure 4.11. Plan de phase X.Y

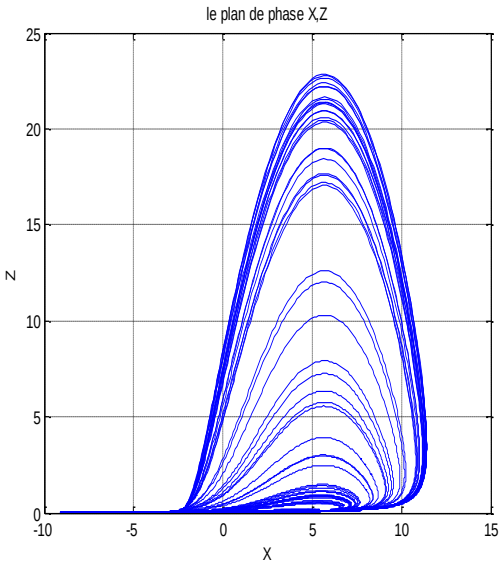


Figure 4.12. Plan de phase X.Z

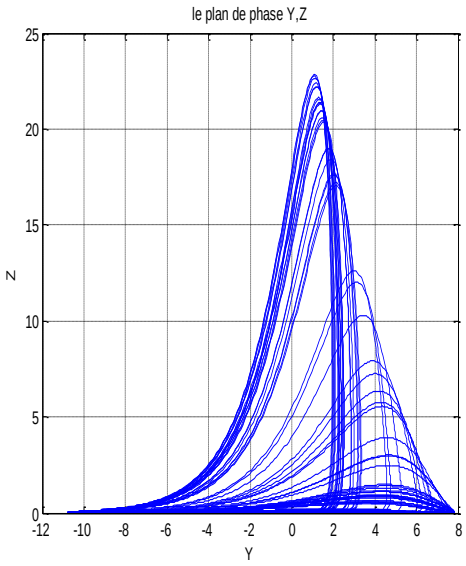


Figure 4.13. Plan de phase Y.Z

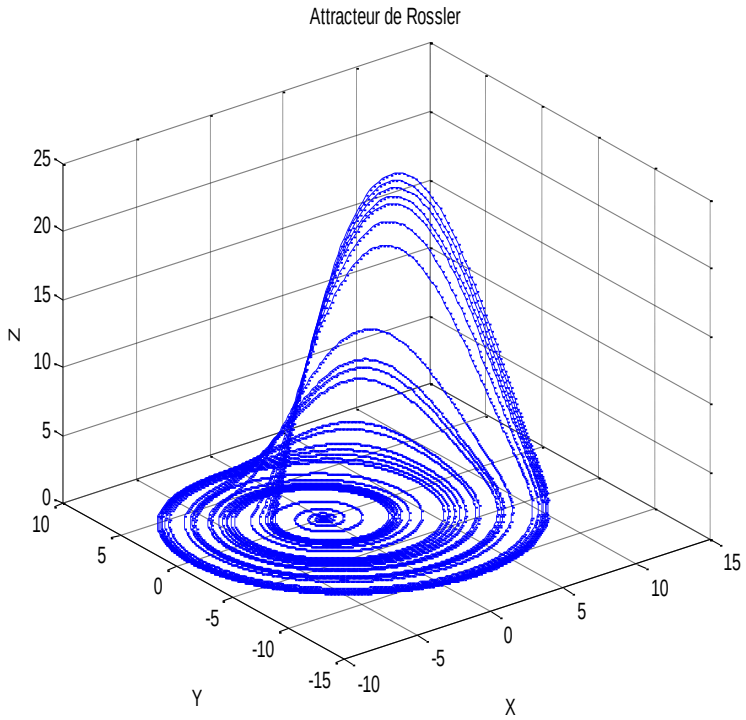


Figure 4.14. Attracteur de Rössler.

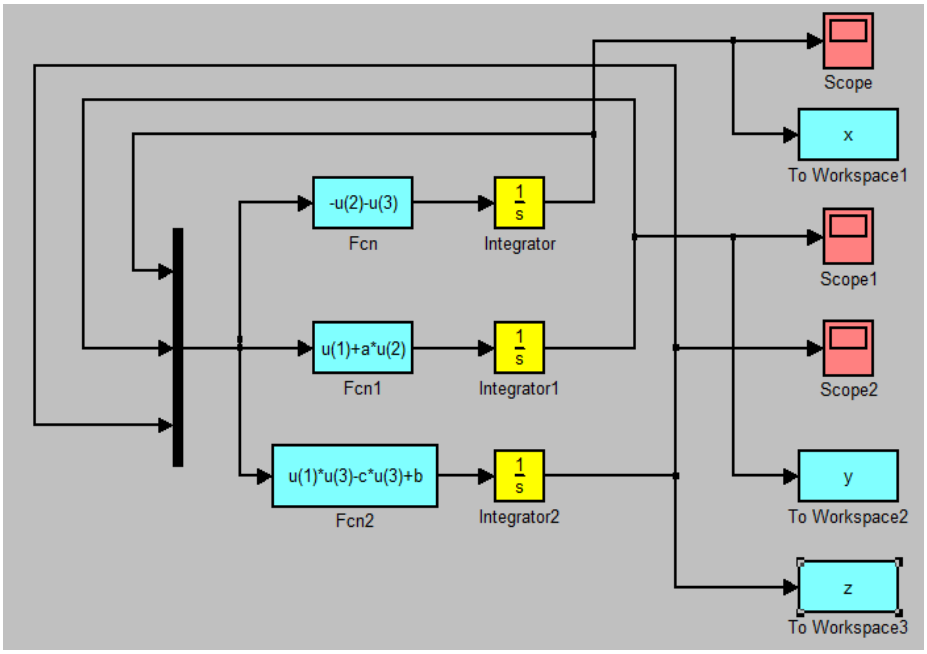


Figure 4.15. Schéma du système de Rössler sur Matlab/Simulink.

4.3 Résultats de l'implémentation

Les figures [(4.14) (4.15) (4.16) représentent les résultats de l'implantation de la du système de Rössler sur le Code Composer Studio (CCS v3.1).

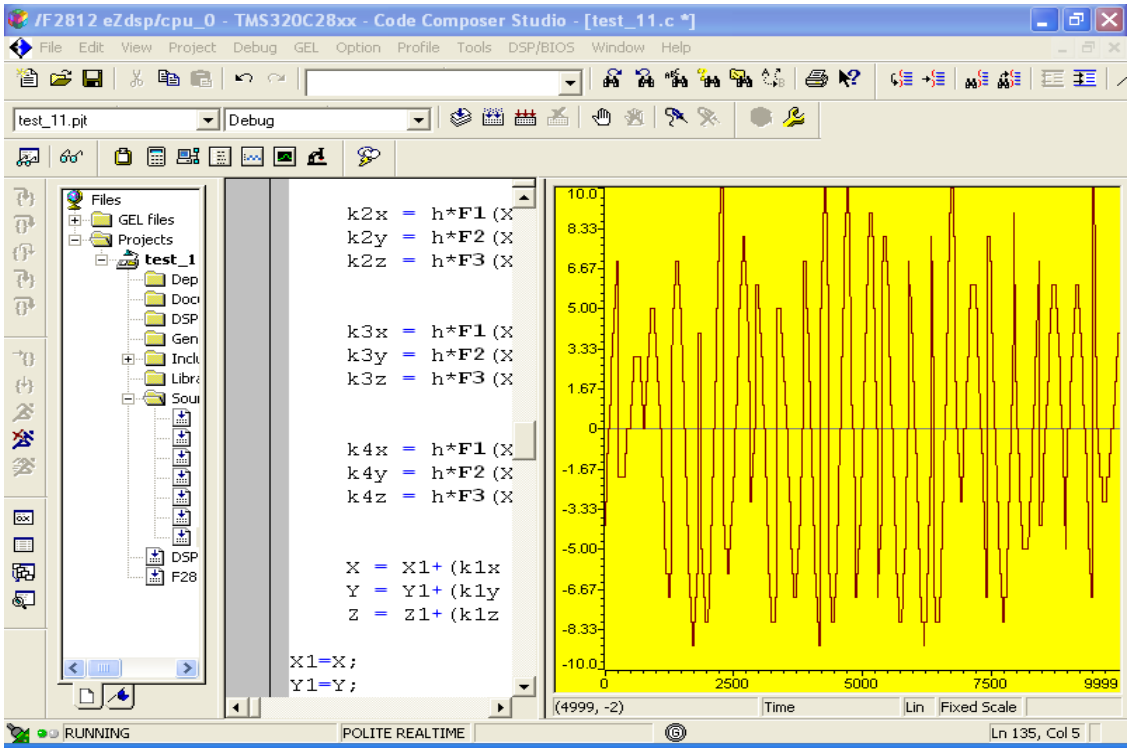


Figure 4.16. Signal X(t).

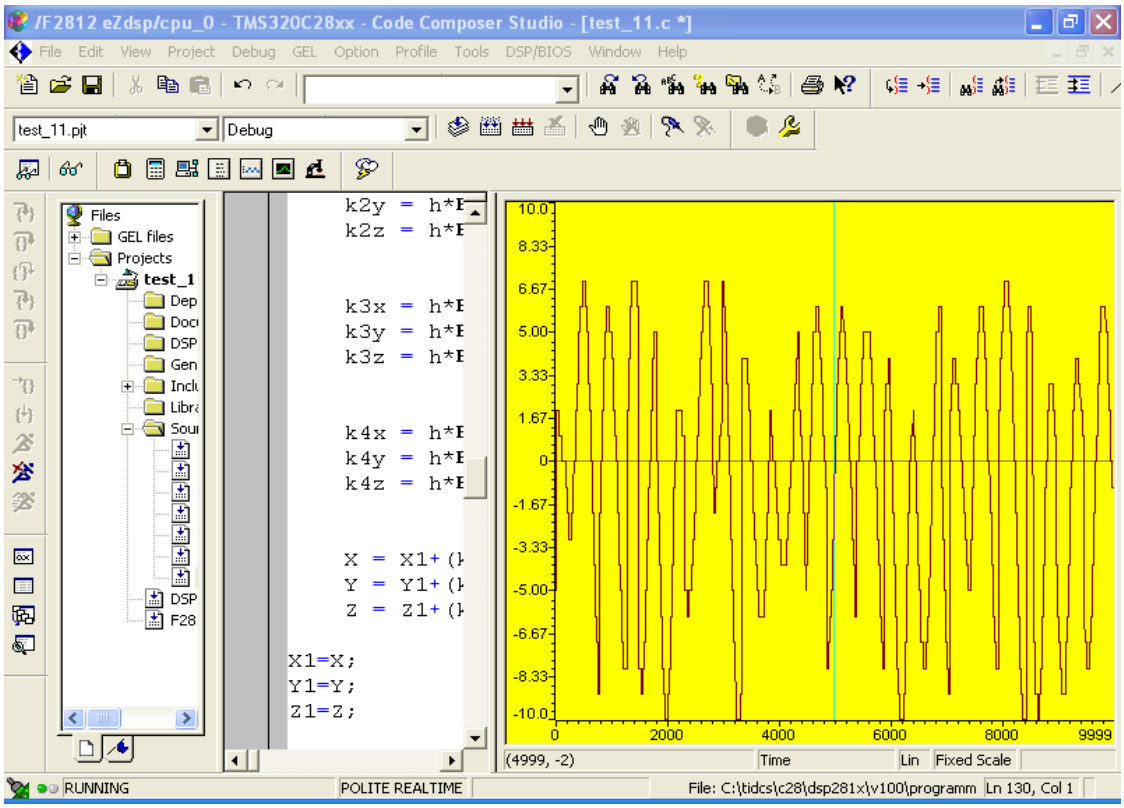


Figure 4.17. Signal Y(t).

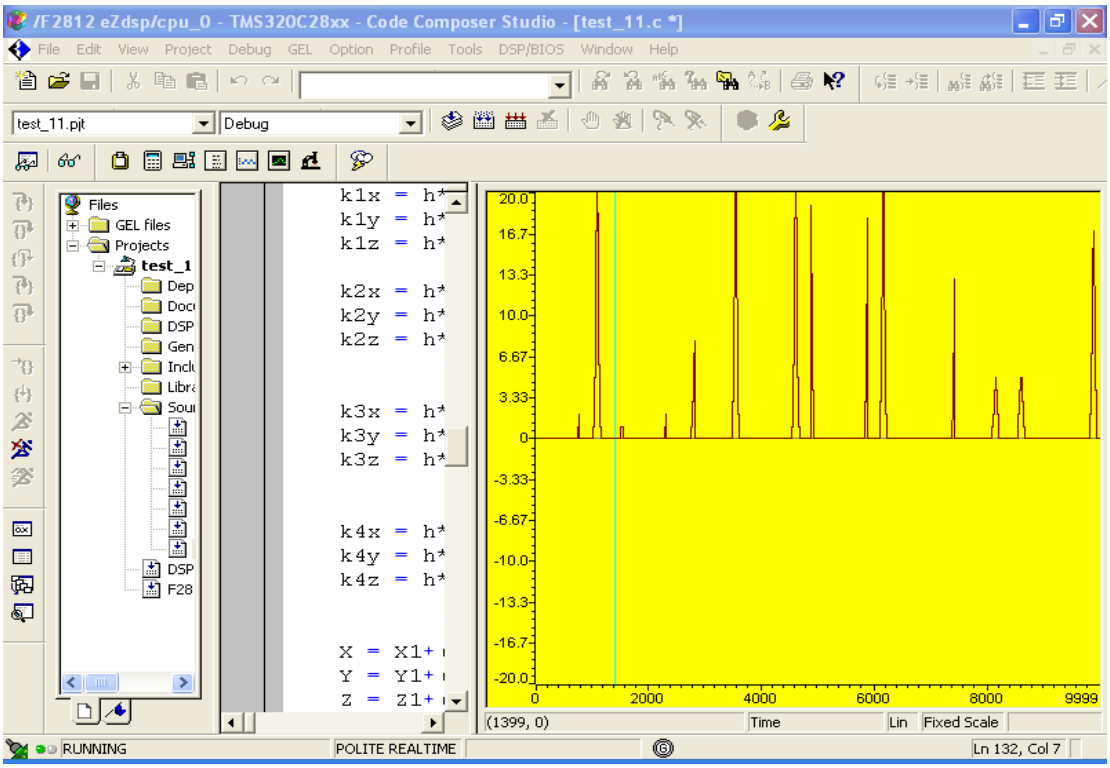


Figure 4.18. Signal Z(t).

4.4 Commentaires

Les figures [(4.1) (4.2) (4.3) (4.4) (4.5) (4.6)…… (4.13)] représentent les résultats de la simulation théorique du système de Rössler et les figures [(4.15) (4.16) (4.17] représentent les résultats expérimentaux du système de Rössler sur la carte DSP.

Telle que : X, Y, Z sont des variables d'états, et a=0.2, b=0.2, c=5.7 sont des valeurs réels.

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.

4.5 Conclusion

Les résultats obtenus après la simulation du système de Rössler 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

Dans ce travail nous avons étudié la génération du signal chaotique fourni par le système de Rössler, en utilisent la méthode numérique de Runge-Kutta d'ordre 4.

Nous avons présenté d'abord les notions de base des systèmes dynamiques et du chaos, ensuite la résolution du système de Rössler à l'aide de la méthode numérique Runge–Kutta d'ordre 4. Et le troisième chapitre est consacré à l'implémentation de cette méthode sur la carte DSP. Après cela, les résultats de la simulation et de l'implémentation sont données.

Nous concluons que :

- L'analyse des systèmes chaotiques (système de Rössler) par la méthode de Runge-Kutta donne des résultats analogues avec plusieurs approches différentes.
- Validation des résultats de simulations.
- La réalisation de ce système, nous a donné un aspect pratique aux études théoriques acquises.
- L'utilisation du DSP, nous a permis d'acquérir de nouvelles connaissances sur les technologies modernes utilisées.

Dédicace

A mes chers parents qui était ma raison de vivre, et qui m'ont bien soutenu pour arriver jusqu'ici.

A mes proches et toute ma famille

A mes camarades de promotion que je n'oublierai jamais les moments passés avec eux.

A tous les gens qui m'aiment sans oublier mon frère et ami et voisin BELERBI ISLEM.

Au bonheur des plus chers.

Je dédie ce modeste travail.

Ahmed