UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

U. A. C. P. y P.

ESQUEMA DE CIFRADO Y DESCIFRADO MEDIANTE EL EMPLEO DE SISTEMAS CAÓTICOS CONTINUOS

T E S I S

QUE PARA OBTENER EL TÍTULO DE MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A :

MIGUEL SANTIAGO SUÁREZ CASTAÑÓN

Asesor de Tesis.

Dr. Carlos Fernando Aguilar Ibañez

MÉXICO, D.F. ABRIL DE 2001

Indice.

Resumen	3
1 Introducción	4
2 Conceptos básicos de Criptografía y planteamiento del problema	6
2.1 Introducción	6
2.2 Planteamiento del problema	6
2.3 Seguridad en cómputo: Fundamentos	7
2.3.1 Metas de la seguridad en cómputo	7
2.3.2 Criptografía	10
2.3.3 Criptoanálisis	11
2.3.4 Algoritmos de cifrado por sustitución y por transposición	12
2.3.5 Algoritmos de cifrado seguros	15
2.3.6 Protocolos criptográficos	19
2.4 Conclusiones	22
3 Sistemas dinámicos y caos	23
3.1 Introducción	23
3.2. Sistemas no lineales	23
3.2.1 Mapas iterativos	24
3.2.2 Sistemas continuos	27
3.3 Bifurcaciones en sistemas continuos	28
3.3.1 Espacio de fase	28
3.3.2 Bifurcación de punto silla	28
3.3.3 Bifurcación transcrítica	29
3.3.4 Bifurcación de horca	29
3.3.5 Bifurcación de Hopf	29
3.4 Caos en sistemas continuos	29
3.4.1 Condiciones para el caos	31
4 Diseño de observadores no lineales para sistemas caóticos Hamiltonianos	32
4.1 Introducción	32
4.2 Sistemas Hamiltonianos	33
4.3 Ejemplos de sistemas caóticos Hamiltonianos	35
4.4 Diseño de observadores no lineales para un sistema Hamiltoniano	37
4.5 Aplicación: Cifrado y descifrado de información	41
4.6 Ejemplos de observadores no lineales para sistemas Hamiltonianos	42
4.6.1 Sistema de Lorenz	42
4.6.2 Sistema de Chen	42
4.7 Conclusiones	43

5 Instrumentación numérica 45
5.1 Introducción 45
5.2 Simulación numérica empleando SIMNOM 46
5.2.1 Sistema de Lorenz 46
5.2.2 Sistema de Chen 51
5.3 Experimentos numéricos 53
5.3.1 Método Runge-Kutta-Fehlberg 54
5.3.2 Discretización del sistema de Lorenz y su observador empleando el método Runge-Kutta-Fehlberg 56
5.3.3 Resultados obtenidos mediante la implantación del esquema de cifrado propuesto en esta tesis empleando el sistema de Lorenz 60
6 Conclusiones 67
6.1 Alcances y logros 67
6.2 Trabajo futuro 68
Bibliografía 69
Apéndice A: Estabilidad de Lyapunov y observabilidad 71
Estabilidad de Lyapunov 71
Estabilidad 71
Funciones de Lyapunov 73
Observabilidad 74
Observadores de Luenberger 76
Apéndice B: Código de los prototipos que implantan el esquema de cifrado que se propone en esta tesis, empleando el sistema de Lorenz y el observador pasivo de sus estados 77
B.1 Cifrador 77
B.2 Descifrador 85

Resumen.

La presente tesis propone la utilización de los sistemas no lineales caóticos y de los observadores pasivos de sus estados, para el cifrado y descifrado, respectivamente, de información; en particular, el sistema conocido como sistema de Lorenz [Lorenz,1963]. El proceso de cifrado se realiza mediante la generación de una llave del mismo tamaño que el tamaño de la información, es decir, que ambas estén representadas con igual número de bits. El proceso de descifrado se lleva a cabo mediante la reconstrucción de la llave utilizada para el cifrado, para luego emplearla en la operación inversa a la usada en el cifrado y así recuperar la información original. Las trayectorias de los estados del sistema empleado constituyen la llave de cifrado; para reconstruir la llave y descifrar la información, se utiliza el observador pasivo de estados del sistema. Se muestran varios ejemplos de cifrado/descifrado de distintos tipos de información (texto e imágenes), generados por los programas escritos en el lenguaje de programación C, que implementan el esquema que se sustenta.

1 Introducción.

El concepto de sistema de cómputo normalmente se refiere a una colección de recursos de cómputo que interactúan unos con otros, así como con usuarios, programadores y operadores; estos sistemas, en la mayoría de los casos requieren de aplicaciones para poder funcionar y llevar a cabo un sinnúmero de tareas. El comportamiento en conjunto del sistema de cómputo y las aplicaciones, normalmente deben de satisfacer un cierto número de requerimientos; el grado en que éstos son logrados nos permite establecer la confiabilidad y desempeño del sistema. Casi siempre la confiabilidad y el desempeño es medido por la respuesta a la información externa que se proporciona como entrada y que debe de permanecer dentro de algún rango establecido como aceptable. La información de entrada que sale del rango, puede ocasionar que el sistema falle y se vuelva ineficiente. Sin embargo, existen aplicaciones en las que la noción de confiabilidad y desempeño anteriores no es adecuada; en estos casos el rango de la información de entrada es establecido por un adversario o enemigo, cuyo objetivo es transtornar el adecuado funcionamiento del sistema; no obstante, en la mayoría de los casos, a pesar de la entrada proporcionada por el enemigo, el sistema debe de continuar funcionando.

Ejemplos de sistemas que deben de sobreponerse al ataque de un enemigo existen muchos, tales como los sistemas utilizados para la transferencia de fondos financieros, que deben de evitar que se realicen transacciones fraudulentas por alguna persona deshonesta, como mover fondos de una cuenta ajena a la propia, sin el consentimiento del dueño legítimo de los fondos.

En conjunto, las acciones realizadas y los recursos empleados para preservar la integridad del sistema de cómputo, en los términos que aquí se indican, se conoce como SEGURIDAD EN CÓMPUTO. La seguridad de un sistema de cómputo es la habilidad de éste para lograr algunos requerimientos, a pesar de las acciones de algún enemigo, quizá bien informado y con medios adecuados para corromper el sistema. Los requerimentos pueden incluir, mantener secretos, mantener la integridad de la información, prevenir accesos no autorizados a los recursos del sistema, proveer los servicios del sistema permanentemente, etc.

Las metas de la seguridad en cómputo se pueden resumir en tres: confidencialidad, integridad y disponibilidad. La confidencialidad asegura la protección de información privada, restringe el acceso a los usuarios sólo a la información que les ha sido autorizado, controla quien puede hacer uso del sistema y de sus recursos; la integridad establece que el sistema no debe corromper la información o permitir modificaciones accidentales o no autorizadas a ésta; y la disponibilidad se refiere a que el sistema de cómputo debe permanecer funcionando eficientemente y que sea capaz de recuperarse rápido y por completo en caso de que algún desastre ocurra.

Las metas de la seguridad pueden ser medianamente logradas por un buen número de herramientas ya existentes. Sin embargo, la Criptografía es hasta la fecha la herramienta más poderosa con que se cuenta, por lo que la investigación en esta área ha sido central en el desarrollo de la seguridad. La gran mayoría de las herramientas criptográficas existente y que se emplean comúnmente, se encuentran sumamente restringidas por las políticas de seguridad nacional de los países en donde se desarrollan, por lo que la importación de éstas a nuestro país es limitada, y dada la importancia y utilidad que estas herramientas tiene en el mundo en que vivimos, se considera que las propuestas y desarrollos que se hagan podrán contribuir en algo para disminuir la dependencia que en esta área se tiene hasta la fecha.

La propuesta de esta tesis es un esquema de cifrado y descifrado de llave privada de información, basado en los modelos matemáticos de sistemas dinámicos no lineales caóticos.

El trabajo se organiza de la siguiente manera:

En el capítulo 1, se da el planteamiento del problema que se pretende resolver y se introducen los conceptos básicos de Criptografía que permiten contextualizar este trabajo.

En el capítulo 2, se definen los conceptos básicos de sistemas no lineales, de caos y la relación que éstos tienen, a partir de la cual se describen los circuitos caóticos que se emplean en la solución al problema que se plantea.

En el capítulo 3, se describe el método de sincronización para sistemas Hamiltonianos y se determinan los sistemas observadores para varios sistemas no lineales, entre ellos los observadores de los sistemas de Lorenz y de Chen.

En el capítulo 4, se presentan las simulaciones hechas en SIMNOM de los sistemas de Lorenz y Chen y los observadores de cada uno de ellos.

En el capítulo 5, se plasma la implantación del esquema de cifrado/descifrado, mediante el empleo del sistema de Lorenz y su sistema observador, así como los resultados obtenidos.

Se incluyen los apéndices correspondientes a los capítulos 3 y 5. El primero, se refiere a la teoría de la estabilidad de Liapunov y observabilidad en sistemas dinámicos; el segundo, contiene el código de los programas empleados en el capítulo 5.

2 Conceptos básicos de Criptografía y planteamiento del problema.

2.1 Introducción.

Como ya se mencionó el objeto de esta tesis es proponer un esquema de cifrado y descifrado de información, por lo que en este capítulo se describen los conceptos fundamentales y más relevantes de seguridad en cómputo, que permitirán al lector contar con un marco teórico que servirá como referencia de los principales algoritmos criptográficos, y así poder establecer la importancia que éstos tiene en el área de seguridad en cómputo. Además ubica en este contexto el problema objeto de estudio de este trabajo.

2.2 Planteamiento del problema.

El cifrado y descifrado de información, conocido comúnmente como Criptografía, es la herramienta fundamental en seguridad en cómputo, ya que hasta hoy, es el único medio conocido para mantener en secreto información sensible con un grado importante de confianza. Los algoritmos criptográficos se dividen en algoritmos de llave privada y algoritmos de llave pública. Ambos tipos basan su grado de seguridad en la dificultad para obtener las llaves usadas en el cifrado y/o descifrado. De hecho el conocimiento público del algoritmo empleado es obligado, y aún así éste debe seguir siendo seguro. El problema fundamental que se debe de resolver al proponer un algoritmo criptográfico es la forma en que se obtienen la o las llaves para el cifrado y descifrado, buscando que un enemigo no pueda generarlas, a pesar de conocer, como ya se mencionó, el algoritmo y teniendo a su disposición la información que se haya cifrado. En este trabajo se propone un esquema criptográfico de llave privada que genera la llave de cifrado/descifrado a partir de las trayectorias de los estados de un sistema no lineal caótico (que puede hacerse público), con la particularidad de que ésta **será mismo tamaño que el tamaño de la información, es decir, que ambas estén representadas con igual número de bits; esta característica proporciona un grado absoluto de seguridad, según los resultados que C. Shannon obtuvo en su trabajo sobre Teoría de la Información (ver [Shannon, 1993]).

El esquema que aquí se presenta fue verificado mediante dos programas escritos en el lenguaje de programación C (se agregan en el apéndice B); el primero de ellos implanta el sistema de Lorenz para el cifrado, el segundo es el observador pasivo de los estados del sistema de Lorenz, utilizado para el descifrado. En ambos casos, se utilizó el método numérico Runge-Kutta-Fehlberg (RKF) (ver [Gerald, Wheatley, 1994]), para su discretización.

2.3 Seguridad en cómputo: Fundamentos.

Los recursos de cómputo y la información son considerados en muchas instituciones como uno de sus bienes de más valor. Los componentes de hardware pueden variar en costo, desde unos miles de pesos hasta varios millones, sin embargo, la información generada, almacenada y transmitida mediante la utilización del equipo de cómputo puede tener un valor que rebasa cientos de veces al del hardware. Es por eso que a la información debe de dársele un cuidado especial, ya que ésta puede representar el trabajo de muchos meses y de un gran número de personas, puede significar la subsistencia de alguna compañía, o incluso puede ser ilegal tener acceso a ciertos datos que pondrían en ventaja desleal a un competidor en un cierto ámbito.

2.3.1 Metas de la seguridad en cómputo.

La seguridad en cómputo trata de mantener tres características en el sistema: confidencialidad, integridad y disponibilidad.

Confidencialidad:

Asegura la protección de información privada, restringe el acceso a los usuarios sólo a la información que les ha sido autorizado, controla quién puede hacer uso del sistema y de sus recursos.

Integridad:

Establece que el sistema no debe corromper la información o permitir modificaciones accidentales o no autorizadas a ésta.

Disponibilidad:

Se refiere a que el sistema de cómputo debe permanecer funcionado eficientemente y que sea capaz de recuperarse rápido y por completo en caso de que algún desastre ocurra.

Vulnerabilidades y Amenazas: Estos dos tópicos deben ser considerados siempre que se trate con la seguridad de un sistema de cómputo.

Vulnerabilidad:

Se refiere a lugares o eventos en los que bajo ciertas circunstancias el sistema es susceptible a un ataque. Los lugares o eventos en los que comúnmente un sistema puede ser vulnerable son:

- Los edificios y las habitaciones en las que se encuentra el sistema.
- Desastres naturales y ambientales (incendios, terremotos, perdida de la corriente eléctrica, polvo, humedad, temperatura,etc.).
- Fallas en hardware y software.
- Los medios de almacenamiento pueden ser fácilmente robados o destruidos.
- Radiaciones eléctricas y electromagnéticas.
- El personal que administra y hace uso del sistema de computo.

Amenaza:

Son las circunstancias que potencialmente pueden ocasionar algún daño o la perdida del sistema de cómputo o alguno de sus componentes.

Existen cuatro tipos de amenazas:

Interrupción:

Cuando un recurso del sistema de cómputo se pierde, no está disponible o se vuelve inútil. Una interrupción puede darse por la destrucción mal intencionada del hardware, por el borrado de algún programa o archivo de datos, etc.

Intercepción:

Cuando una persona, programa o sistema de cómputo no autorizado obtiene acceso al sistema, por ejemplo el acceso no autorizado a una red.

Modificación:

Cuando no sólo se obtiene acceso al sistema, sino que también se modifica la información. Por ejemplo, alguien puede obtener acceso a una base de datos y modificar algunos de los registros.

Fabricación:

Cuando algún intruso falsifica objetos en un sistema de cómputo, por ejemplo, insertar registros falsos en una base de datos o enviar un mensaje haciéndose pasar por otra persona.

El hardware, el software, los datos, los medios de almacenamiento y la gente son susceptibles a las amenazas, en cualquiera de los cuatro tipos mencionados.

Controles: Para preservar la confidencialidad, integridad y disponibilidad del sistema de cómputo, es necesario establecer controles. En algunas ocasiones los controles permiten evitar ataques, en otras únicamente permiten advertir que un ataque se ha realizado.

Encripción o Cifrado:

El cifrado de la información es la herramienta más poderosa con que se cuenta en seguridad, y su objetivo es transformar los datos de tal forma que pierdan todo significado para cualquier observador extraño, nulificando la posibilidad de utilizar algún mensaje interceptado, fabricar información o modificarla.

Controles de software:

Los programas por sí mismos son la segunda herramienta en importancia en seguridad. Los programas pueden establecer restricciones de seguridad, tales como limitaciones en el acceso a los recursos del sistema.

Los controles de software pueden usar herramientas como componentes de hardware, cifrado, etc. Los controles de software afectan directamente a los usuarios, por lo que deben de ser diseñados con mucho cuidado.

Controles de hardware:

Se han creado una serie de dispositivos especiales para seguridad, como tarjetas inteligentes para cifrado, que limitan el acceso a ciertos recursos, candados que evitan la extracción de los dispositivos de almacenamiento, etc.

Políticas y educación:

En la práctica se ha observado que la falta de ciertas políticas, como la modificación de las palabras de acceso (passwords) periódicamente y la falta de educación del personal genera una baja importante en el nivel la seguridad.

Controles físicos:

Uno de los controles más económicos y eficientes son los físicos. Estos controles incluyen candados en las puertas, guardias de seguridad, respaldos de la información y del software más importante.

2.3.2 Criptografía.

La Criptografía es la herramienta más poderosa e importante en seguridad. A continuación se dan algunas definiciones importantes que serán utilizadas a lo largo de la discución.

Texto plano: El texto legible y comprensible por cualquier persona.

Texto cifrado: La forma ininteligible del texto plano.

Criptografía: Arte o ciencia que busca formas para ocultarle el contenido de un mensaje a un enemigo.

Criptoanálisis: Arte o ciencia que trata de obtener el texto plano a partir del texto cifrado.

Criptología: Rama de las matemáticas que incluye a la Criptografía y al Criptoanálisis.

Algoritmo criptográfico o cifrador: Es una función matemática para cifrar y descifrar.

Llave: La seguridad de un sistema de cifrado no debe depender del algoritmo utilizado, sino de la llave utilizada para cifrar algún mensaje. Ésta puede ser cualquier valor dentro de algún rango de valores posibles.

Espacio de llaves: Es el rango de valores posible de donde se puede escoger una llave.

Notación utilizada: El texto plano se denota con M, el texto cifrado con C, la función de cifrado con E, la función de descifrado con D y la llave utilizada para cifrar/descifrar con K_n.

La función E opera sobre M y produce C:

$$C = E(M),$$

la función D opera sobre C y produce M:

$$M = D(C), \ o \ M = D(E(M)).$$

Los algoritmos que utilizan una sola llave conocidos como de llave privada o simétricos para cifrado y descifrado se denotan como:

$$M = D_k(E_k(M)).$$

Los algoritmos que utilizan una llave para cifrado (k_1) y otra para descifrado (k_2) conocidos como de llave pública o asimétricos, se denotan como:

$$M = D_{k_2}(E_{k_1}(M))$$

2.3.3 Criptoanálisis.

Es la ciencia que se ocupa de recuperar el texto plano de un mensaje sin tener acceso a la llave de cifrado. Un Criptoanálisis exitoso puede recuperar la llave o el texto plano. También puede encontrar debilidades en el algoritmo utilizado para el cifrado.

El Criptoanálisis se puede dividir en cuatro ataques diferentes:

Ataque de sólo texto cifrado:

El criptoanalista tiene texto cifrado de muchos mensajes, todos ellos cifrados con el mismo algoritmo.

Dado:

$$C_1 = E_k(M_1), C_2 = E_k(M_2), C_3 = E_k(M_3), ..., C_n = E_k(M_n),$$

deducir: $M_1, M_2, M_3, ..., M_n$;o K (llave); o un algoritmo para inferir M_{i+1}, de $C_{i+1} = E_k(M_{i+1})$

Ataque de texto plano conocido:

El criptoanalista tiene acceso además del texto cifrado de varios mensajes, al texto plano correspondiente a los mensajes cifrados.

Dado:

$$M_1, C_1 = E_k(M_1), M_2, C_2 = E_k(M_2), ..., M_i, C_i = E_k(M_i),$$

deducir: K (llave); o un algoritmo para inferir M_{i-1},de $C_{i+1} = E_k(M_{i-1})$

Ataque de texto plano escogido:

El criptoanalista además de tener el texto cifrado y el texto plano asociado de muchos mensajes, puede seleccionar el texto plano que se ha cifrado. El trabajo consiste en deducir la o las llaves usadas para cifrar los mensajes o un algoritmo para descifrar cualquier mensaje nuevo cifrado con la misma o las mismas llaves.

Dado:

$$M_1, C_1 = E_k(M_1), M_2, C_2 = E_k(M_2), ..., M_i, C_i = E_k(M_i),$$

donde el criptoanalista seleccionó $M_1, M_2, ..., M_i$, deducir: K (llave);
o un algoritmo para inferir M_{i+1},de $C_{i+1} = E_k(M_{i+1})$

Ataque de texto plano escogido adaptativo:

Este es un caso especial del ataque de texto plano escogido. El criptoanalista puede modificar su selección basado en los resultados de cifrados previos. En este ataque, el criptoanalista puede seleccionar un bloque pequeño de texto, cifrarlo y en base al resultado escoger otro bloque y repetir el proceso.

2.3.4 Algoritmos de cifrado por sustitución y por transposición.

Algoritmos de sustitución monoalfabéticos. Estos algoritmos únicamente tienen un valor histórico, y se basan en una propiedad descrita por C. Shannon (ver [Shannon, 1993]) en su trabajo sobre Teoría de la Información, que busca romper la relación entre los caracteres del texto plano y los del texto cifrado, mediante la permutación de los caracteres del texto plano. Un ejemplo clásico de este algoritmo se llama *Cifrador del César*, en el que cada caracter del texto es sustituido por otro que se encuentra tres posiciones adelante en el orden establecido por el alfabeto:

$$\pi(l) = (l + 3) mod 26$$

Existen varias opciones para mejorar este cifrador, una de ellas utiliza una llave para desplazar el albafeto cierto número de posiciones, por ejemplo si tenemos la llave:

$k = key$, entonces el alfabeto queda como sigue:
$abcdefghijklmnopqrstuvwxyz$
$keyabcdfghijlmnopqrstuvwxz$

si $M = hola$, entonces $C = fnjk$

También es posible crear la llave seleccionando caracteres del alfabeto de la siguiente manera: tomar los caracteres de las posiciones $1, 3, 5, 7$ y 9:

La permutación entonces será:

$$\pi(l) = (l * 3) mod 26$$

Algoritmos de sustitución polialfabética. La debilidad del cifrado por sustitución monoalfabético radica en que la frecuencia de los caracteres refleja la distribución del alfabeto subyacente. El uso de dos o más alfabetos permite aplanar la frecuencia de distribución y hace el cifrado criptográficamente más seguro.

La idea es mezclar distribuciones de frecuencias de letras altas con frecuencias de letras bajas. Si t se cifra algunas veces como a y otras como b y si x se cifra algunas veces como b y otras como t, la frecuencia alta de t se mezcla con la frecuencia baja de x, para obtener una frecuencia más moderada de a y b.

Si tenemos dos alfabetos, uno para las letras en posición *par* y otro para las letras en posición *impar*; por ejemplo para las letras impares tenemos la permutación:

$$\pi(l) = (l * 3) mod 26$$

y para las letras pares la permutación:

$$\pi(l) = ((l * 5) + 13) mod 26$$

el texto plano *hola mundo* se cifraría como *vfhnkjncq*, el texto plano *acción* se cifraría como *axgbqa* y se observa que la *cc* de acción se cifra como *xg*.

Como se alternan los alfabetos, la mitad de las letras cifradas son del mismo alfabeto y la otra mitad de un alfabeto diferente, el cifrado polialfabético 'achata' las frecuencias de distribución del texto plano considerablemente.

Tabla de Vigenère:

El uso de dos alfabetos o permutaciones incrementa la posibilidad de reducir las frecuencias de distribución en el texto cifrado, usando $3, 4, 5$ *o* 6 alfabetos se incrementa esta posibilidad.

El enfoque anterior se puede extender a 26 permutaciones, lo que incrementa al máximo la posibilidad de achatar las frecuencias de distribución del texto cifrado, pero dificulta el cifrado y descifrado, por lo que podemos agregar una llave $k_1 k_2 k_3 ... k_n$ y utilizarla cuantas veces sea necesario para cifrar todo el texto.

La tabla de 26 permutaciones se llama tabla de *Vigenère*, que contiene 26 posibles permutaciones del alfabeto. Cada uno de los 26 alfabetos se obtiene de correr una posición a la derecha todos los caracteres con respecto al alfabeto anterior.

Para cifrar el texto tomamos el carácter que esta en el cruce del renglón P_i con la columna K_i de la tabla:

	0								25
	a	b	c	d	e	...	x	y	z
A	a	b	c	d	e	...	x	y	z
B	b	c	d	e	f	...	y	z	a
C	c	d	e	f	g	...	z	a	b
:	:	:	:	:	:	...	:	:	:
:	:	:	:	:	:	...	:	:	:
X	x	y	z	a	b	...	u	v	w
Y	y	z	a	b	c	...	v	w	x
Z	z	a	b	c	d	...	w	x	y

Figura 2.1 Tabla de Vigenère

Algoritmo de cifrado por transposición. Estos algoritmos se basan en la propiedad de difusión descrita por C. Shannon (ver [Shannon, 1993]) en su trabajo de Teoría de la Información, que busca esparcir la información del texto plano en el texto cifrado.

El cifrado consiste en un reordenamiento de las letras que forman el texto plano, a diferencia de la sustitución en donde las letras del texto plano se cambian por otras.

La transposición trata de romper los patrones establecidos en las letras del mensaje, debidos al propio idioma en que está escrito el texto.

La transposición también es una permutación.

Un ejemplo simple de un cifrado por transposición consiste en escribir el mensaje en forma de columnas, estableciendo el número de columnas, y obtener el texto cifrado de la matriz transpuesta del texto plano:

Si el texto plano es *hola amigos míos*, y definimos una matriz de cuatro columnas:

$$\begin{matrix} h & o & l & a \\ a & m & i & g \\ o & s & m & í \\ o & s & x & x \end{matrix}$$

El texto cifrado es: *haooomsslimxagíx*.

2.3.5 Algoritmos de cifrado seguros.

Sistemas de cifrado de llave pública (o asimétricos). En 1976 Diffie y Hellman (ver [Pfleeger, 1996], [Schneier, 1996]) propusieron el modelo de llave pública. Este modelo utiliza una llave para el cifrado, llamada llave pública y otra para el descifrado, llamada llave privada.

Existen muchos algoritmos de llave pública, pero hasta ahora el único que ha permanecido seguro es el conocido como *RSA* (ver[Pfleeger, 1996], [Schneier, 1996]).

Algoritmo de cifrado de llave pública RSA

Este algoritmo debe su nombre a las iniciales de los nombres de sus tres inventores Rivest, Shamir y Adelman.

Este algoritmo está basado en resultados de teoría de números combinado con el problema de determinar los factores primos de un número compuesto. Este algoritmo requiere de aritmética modular.

Descripción del algoritmo.

i) Se usan dos llaves , una para cifrar e y otra para descifrar d.

e $\rightarrow$ llave pública,
d $\rightarrow$ llave privada.

Si P es el texto plano y C el texto cifrado, entonces:

$$C = P^e \ mod \ n; P = C^d \ mod \ n$$

Por la simetría de la aritmética modular el cifrado y descifrado son mutuamente inversos y conmutativos:

$$P = C^d \ mod \ n = P^{e^d} mod \ n = P^{d^e} mod \ n,$$

y las llaves son intercambiables.

ii) Selección de las llaves:

El par de llaves está formado por los enteros (e, n) para cifrar y los enteros (d, n) para descifrar.

ii.1) Seleccionar n. Debe ser grande y producto de dos primos p y q, también grandes (de por lo menos 100 bits).

ii.2) Se selecciona un entero e, primo relativo de $(p-1)*(q-1)$. Para garantizar que e sea primo relativo de $(p-1)*(q-1)$, seleccionar un primo que sea mayor que $(p-1)$ y $(q-1)$.

ii.3) Seleccionar d, tal que:

$$e * d \text{ sea equivalente a 1 módulo } (p-1) * (q-1),$$
$$\text{es decir } d = e^{-1} \; mod \; (p-1) * (q-1)$$

iii) Fundamentos matemáticos:

La función $\phi(n)$ de Euler (ver[Pfleeger, 1996], [Schneier, 1996]) es el número de primos relativos (enteros positivos) menores que n. Si p es primo:

$$\phi(p) = (p-1)$$

Si $n = p * q$, p y q son primos:

$$\phi(n) = (p-1) * (q-1)$$

Euler y Fermat (ver[Pfleeger, 1996], [Schneier, 1996]) demostraron que:

$$x^{\phi(n)} \; mod \; n = 1 \; mod \; n,$$

para cualquier entero x si n y x son primos relativos.

Supongamos que ciframos un texto plano P, tal que:

$$E(P) = P^e,$$

debemos de asegurarnos que se puede recuperar el mensaje. El valor de e se selecciona de tal forma que podamos recuperar el inverso fácilmente.

Como e y d son inversos modulo $\phi(n)$:

$$e * d \; mod \; \phi(n) = 1$$

Sistema de cifrado de llave privada (o simétrico): Los sistemas de cifrado de llave privada o simétricos utilizan la misma llave para cifrar y para descifrar. El más popular y más utilizado es el conocido como *Encripción Estandar de Datos* ó por sus siglas en inglés DES (*Data Encription Estandar*) (ver[Pfleeger, 1996], [Schneier, 1996]). Este algoritmo fue inventado en los laboratorios de IBM en 1976 y se desarrolló a partir de otro sistema de llave privada llamado *Lucifer* (1974).

DES.

Este algoritmo mezcla las técnicas de sustitución y transposición.

El proceso consta de 16 iteraciones, cada una con una sustitución al inicio y una permutación al final. El texto se cifra en bloques de 64 bits. Este algoritmo sólo usa aritmética estándar y operaciones lógicas con números de hasta 64 bits, lo que facilita su instrumentación en hardware y en software.

Funcionamiento:

Cifrado:

i). Se divide el primer bloque de 64 bits
en dos mitades, izquierda y derecha,
ii). se revuelve (permutación) cada mitad independientemente y
iii). se combina la llave con la mitad derecha al sumarse (XOR) y se intercambian las mitades.
iv). El proceso se repite 16 veces.

Descifrado:

El proceso de descifrado es idéntico, pero la llave se invierte.

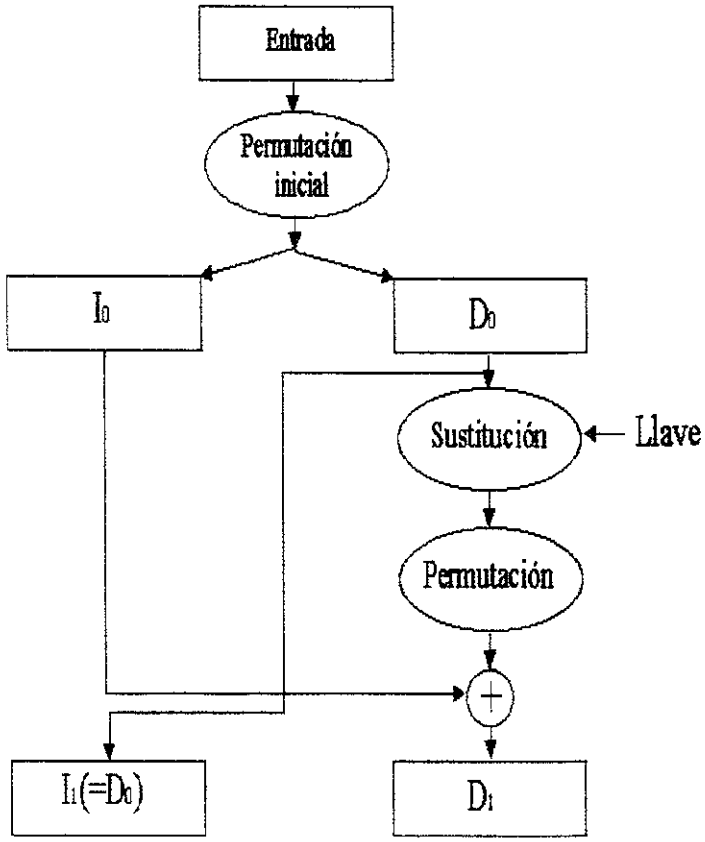

Figura 2.2 Iteración en DES, ésta se repite 16 veces

2.3.6 PROTOCOLOS CRIPTOGRÁFICOS:

El objetivo principal de la Criptografía es resolver problemas. La Criptografía resuelve problemas que involucran mantener secretos, autentificación, integridad y gente deshonesta, entre otros. La utilización de los diferentes criptosistemas por sí solos no resuelven todos los problemas de seguridad, o lo hacen parcialmente. Además, su utilización debe de apegarse a un protocolo perfectamente establecido, de acuerdo al problema que se quiere resolver.

Protocolos. Un protocolo es una serie de pasos, que incluyen dos o más partes (posiblemente personas), y están diseñados para lograr una tarea. Un protocolo debe de satisfacer las siguientes características:

- El número de pasos debe ser finito.

- Todos los involucrados en el protocolo deben de conocer cada uno de los pasos por adelantado.

- Todos los que participan en el protocolo deben de estar de acuerdo en seguirlo.

- El protocolo debe de estar libre de ambigüedades.

- El protocolo debe ser completo; debe de haber una acción específica para cada situación.

Existen tres tipos de protocolos:

Protocolos arbitrados:

En este tipo de protocolos participa un árbitro en el que las partes involucradas tiene total confianza en éste y ayuda a llevar a cabo la tarea para la cual fue diseñado el protocolo.

Protocolos adjudicados:

Este tipo de protocolos permite resolver una disputa. En este caso el protocolo es análogo a la función de un tribunal, en el que se dirime una disputa mediante la intervención de un juez.

Protocolos auto-ejecutados:

Este tipo de protocolos garantiza la equidad entre las partes. No es necesaria la intervención de un tercero para completar la tarea para la cual fue diseñado el protocolo.

Ejemplo: Comunicación usando Criptografía Privada o Simétrica:

El siguiente protocolo permite la comunicación segura entre dos personas, utilizando algún algoritmo de llave privada:

1. Alicia y Bob acuerdan el criptosistema que usarán.
2. Alicia y Bob acuerdan la llave que utilizarán.
3. Alicia toma un texto plano, lo cifra usando el algoritmo y la llave que acordó con Bob.

4. Alicia envía el texto cifrado a Bob.
5. Bob descifra el mensaje cifrado que Alicia le envió con el algoritmo y la llave que ambos acordaron y lee el mensaje.

Desventajas:

1. La distribución de llaves debe ser en secreto.
2. Si una llave es comprometida, entonces un tercero puede descifrar el mensaje e incluso puede hacerse pasar por alguno de los participantes legítimos.
3. Si queremos tener comunicación secreta con más de una persona, entonces necesitaremos $\frac{n(n-1)}{2}$ llaves para n personas.

Ejemplo: Comunicación usando Criptografía Pública a Asimétrica.

La comunicación mediante sistemas simétricos tiene la desventaja de la distribución de las llaves, entre otras; este problema se puede resolver fácilmente utilizando algún criptosistema de llave pública.

Protocolo:

1. Alicia y Bob acuerdan cuál criptosistema utilizarán.
2. Bob envía a Alicia su llave pública.
3. Alicia cifra sus mensajes con la llave pública de Bob y envía el texto cifrado a Bob.
4. Bob descifra el texto con su llave privada.

2.4 Conclusiones.

Satisfacer las metas de la seguridad en cómputo, requiere fundamentalmente de mecanismos que permitan ocultar información a aquellas personas que por diversas razones no deben de tener acceso a ésta. Estos mecanismos están cimentados en la Criptografía, y como se describió en este capítulo, los algoritmos criptográficos son el núcleo de esta área. La importancia de este capítulo radica en la descripción de los dos clases de algoritmos con que se cuenta y poder ubicar adecuadamente el esquema que se propone en este trabajo.

3 Sistemas dinámicos y caos.

3.1 Introducción.

El mecanismo de cifrado que se propone en este trabajo requiere de un medio para generar llaves de la misma longitud que el mensaje a ser cifrado y que la posibilidad de que éstas sean generadas por un atacante sea mínima o nula, ya que como todo algoritmo de cifrado, su fortaleza reside en la dificultad de obtener o generar la llave utilizada para cifrar el mensaje. Para la generación de las llaves aprovecharé la naturaleza caótica casi aletoria de algunos sistemas dinámicos no lineales continuos, cuyas soluciones serán utilizadas como llaves de cifrado y descifrado; para el proceso de descifrado se utilizará el sistema observador correspondiente al sistema utilizado para el cifrado, que también es un sistema dinámico no lineal continuo.

En este capítulo se darán las definiciones y los conceptos teóricos que se requieren para entender qué es un sistema dinámico no lineal, qué es el caos y la relación entre ambos. El capítulo siguiente trata la forma en que se construye un observado a partir de la forma canónica llamada Hamiltoniana de un sistema de ecuaciones diferenciales de primer grado y primer orden.

3.2 Sistemas no lineales.

El sistema de ecuaciones diferenciales que representa un sistema lineal (de primer orden) tiene la forma:

$$\frac{dx}{dt} = c_1 x + c_2,$$

donde c_1 y c_2 son constantes y x y $\frac{dx}{dt}$ sólo aparecen a la primera potencia. Si c_1 y c_2 fueran funciones de t, aún así el sistema es lineal. Lo que determina que un sistema sea lineal o no lineal es la forma en que se presentan las variables dependientes y sus derivadas. Sea el sistema:

$$\frac{dx}{dt} = (1 - x)x,$$

no lineal (de primer orden), porque la variable dependiente x, aparece como x^2.

Los sistemas no lineales pueden dividirse en mapas discretos y en sistemas continuos. Un mapa discreto muy conocido es el modelo de generaciones que es discreto en el tiempo; si en una población (de bacterias, de personas, etc.) aparece una generación nueva cada 10 años, entonces, entre cada generación no hay nada, no existe un cambio en el tamaño de la población en el intervalo de diez años, entre una generación y otra. Los sistemas continuos, son continuos en el tiempo; estos sistemas se representan por una o más ecuaciones diferenciales, y pueden evolucionar en todo instante de tiempo.

3.2.1 Mapas iterativos.

Los mapas iterativos son los sistemas no lineales más simples y se llaman así por la forma en que se generan. Inicialmente se da un valor a la función y el resultado se utiliza para volver a evaluar a la función, y este procedimiento se repite. Los valores de la función eventualmente convergen en un punto fijo (como es el caso de la función matemática coseno), al que se le conoce como punto fijo estable; los valores que toma la función antes de llegar al punto fijo estable se llaman órbitas. El conjunto de Mandelbrot es un ejemplo de un mapa iterativo pero en el plano de los números complejos.

Otro ejemplo muy conocido de mapas iterativos es la *ecuación logística*, que es un modelo del crecimiento de una población que no tiene enemigos naturales y su crecimiento está limitado por la cantidad de comida disponible:

$$x_{n+1} = rx_n(1 - x_n)$$

donde $0 \leq r \leq 4$ (la cantidad de alimento). La x inicial debe estar en el rango $0 < x < 1$. Si hacemos $r = 2.9$ y tenemos una población muy pequeña (origen), ésta crecerá hasta cierto tamaño (punto estable fijo). El origen se llama *propulsor* y el punto fijo estable se llama *atractor*.

La siguiente figura muestra la serie de tiempo de la órbita para $r = 2.9$.

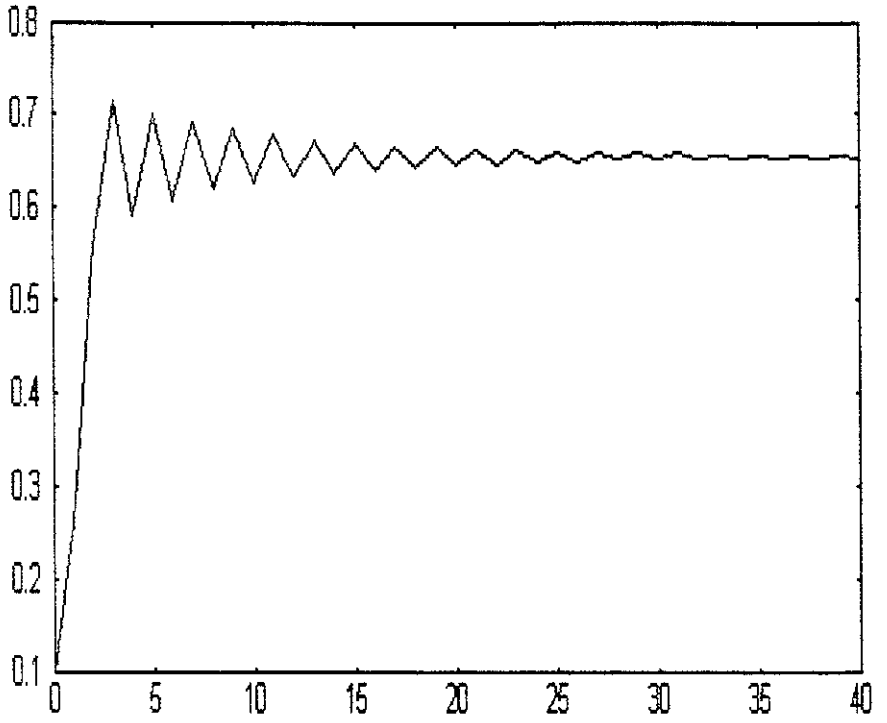

Figura 3.1 Órbita de la ecuación logística, para $r = 2.9$

Un aspecto muy interesante de los sistemas no lineales, es el cambio de su comportamiento al modificar alguno de los parámetros. Considere por ejemplo, si $r = 3$, la órbita no converge en un punto fijo, es decir, el punto fijo se desestabiliza y el sistema se cicla entre dos puntos. Este comportamiento se llama ciclo estable (en este caso de dos puntos) y en nuestro ejemplo se puede interpretar como el efecto que tiene un incremento en el alimento disponible, que hace que la población crezca más rápido, pero debido a ésto, el alimento no es suficiente, por lo que parte de la población muere, habiendo una reducción de individuos, que nuevamente tienen más alimento y vuelve a crecer la población, repitiéndose el ciclo.

La siguiente figura muestra la serie de tiempo de la órbita para $r = 3$.

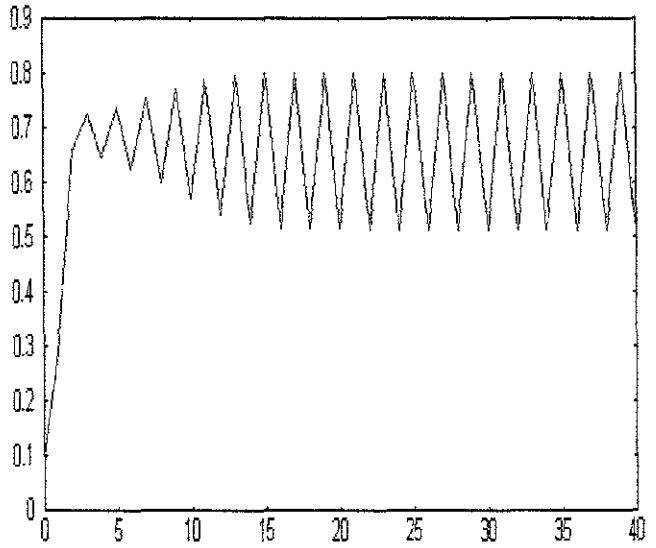

Figura 3.2 Órbita de la ecuación logística, para $r = 3.0$

Si incrementamos el valor de r, por ejemplo, $r = 3.5$ y $r = 3.565$, obtendremos un ciclo estable de 4 puntos y un ciclo estable de 8 puntos, respectivamente. Al hacer $r = 3.9$, obtenemos la siguiente figura (o telaraña):

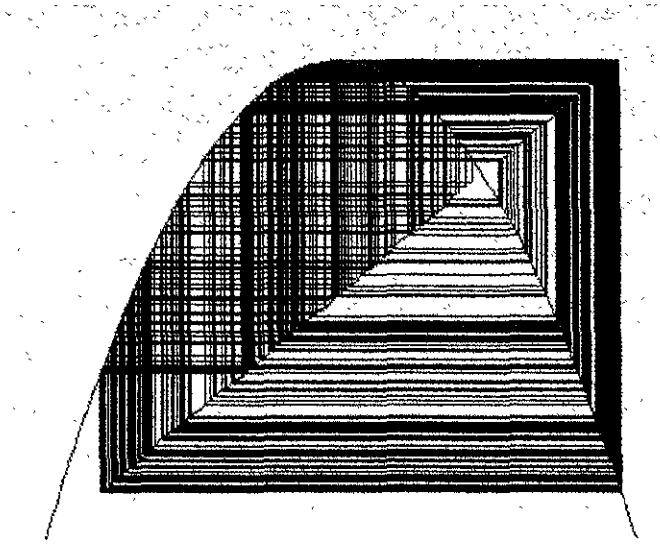

Figura 3.3 Telaraña de la ecuación logística para $r = 3.9$

La figura 3.3 se construye tomando el punto x_1 de la curva logística (que aparece en color rojo en la figura); de este punto se traza una línea horizontal hasta la recta $y = x$ (que aparece en azul en la figura) y de ahí se traza una línea vertical hasta la curva logística. El procedimineto anterior se repite hasta que se tiene un punto fijo.
El comportamiento del sistema no lineal para $r = 3.9$ se llama *caos*. Si graficamos los ciclos estables como funciones de r, éstos se verán como bifurcaciones en un ciclo el doble de largo, como se muestra en la siguiente figura.

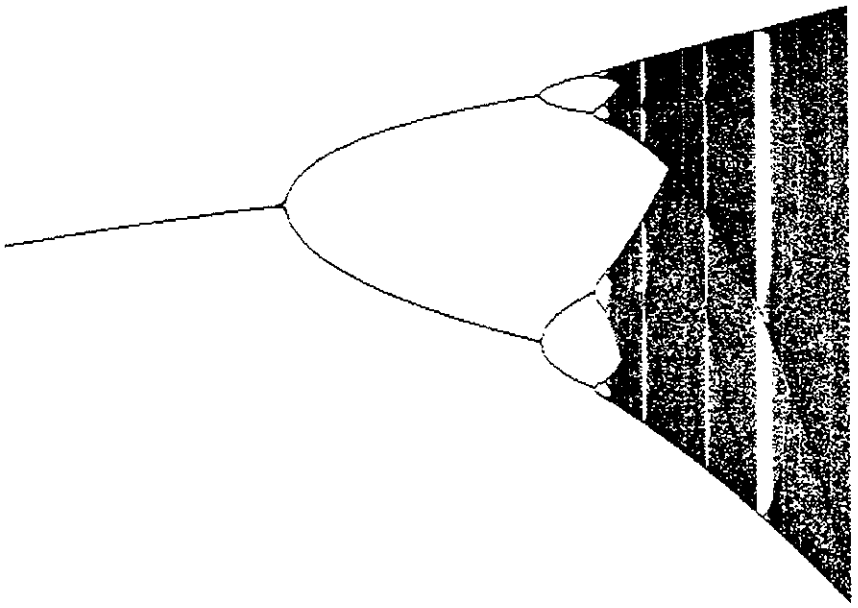

Figura 3.4 Bifurcación de la ecuación logística para $r = 3.9$

Este sistema no lineal, se vuelve caótico a partir de $r = 3.5699$, aunque existen algunos intervalos en los que el comportamiento es periódico.

3.2.2 Sistemas continuos.

Los sistemas no lineales continuos, a diferencia de los mapas discretos, pueden evolucionar para cualquier punto, y se representan mediante ecuaciones diferenciales. Los sistemas lineales generalmente pueden ser resueltos mediante técnicas matemáticas analíticas, mientras que, los sistemas no lineales generalmente tienen que ser resueltos mediante métodos numéricos, tales como los métodos de Euler y Runge-Kutta-Fehlberg (que se tratarán adelante).

3.3 Bifurcaciones en sistemas continuos.

A continuación se dará una serie de definiciones que son de gran utilidad para caracterizar la naturaleza caótica de ciertos sistemas no lineales.

Muchos sistemas no lineales, pueden tener un comportamiento que cambia al modificar alguno o algunos de los parámetros. Supongamos que tenemos un sistema con un parámetro r, y que para algún valor de éste, el sistema tiene un punto fijo; si el valor de r cambia de tal forma que el punto fijo se vuelve inestable, entonces tendremos una bifurcación, a partir del punto en el que el punto fijo se vuelve inestable. Existen varios tipos de bifurcaciones; a continuación se da una breve descripción de cada uno de ellos.

3.3.1 Espacio de fase.

Un espacio de fase es un espacio que tiene todos los aspectos de la dinámica de un sistema en sus ejes, por lo que permiten observar la dinámica del sistema en su conjunto. Normalmente los espacios de fase son de más de tres dimensiones. En un sistema de segundo orden que representa un objeto en movimiento, el espacio de fase puede estar formado por dos ejes, la posición y la velocidad. Un método gráfico muy útil consiste en representar en un eje perpendicular a la línea de fases el valor de la derivada de x, que indica el sentido en que varía x a lo largo de la referida línea de fases. Esta construcción muestra de manera muy simple los puntos fijos del sistema, como aquellos en los que la curva correspondiente a la velocidad cruza el eje x (la velocidad es 0). Cuando la curva se encuentra por debajo del eje x (es decir, cuando la velocidad es negativa), la partícula se mueve hacia la izquierda (hacia valores de x cada vez menores). Por el contrario, cuando la curva de la velocidad está por encima del eje x (velocidad positiva), la partícula se mueve hacia la derecha (hacia x mayores). Con esta información se puede analizar la estabilidad de los puntos fijos: si la partícula se mueve hacia el punto fijo a ambos lados de éste, dicho punto fijo es estable. Por el contrario, si la partícula se mueve en la dirección contraria al punto fijo a ambos lados de éste, el punto fijo es inestable.

3.3.2 Biburcación de punto silla.

La bifurcación de punto silla es el tipo más simple de bifurcación. Con ella se crean y destruyen puntos fijos. Una bifurcación de punto silla puede ocurrir en sistemas que no tienen puntos fijos. A medida que un parámetro del sistema varía, dos puntos fijos aparecen en la bifurcación, uno estable y el otro inestable.

3.3.3 Bifurcación transcrítica.

Una bifurcación transcrítica no crea ni destruye puntos fijos. Por el contrario, a medida que varía un parámetro, dos puntos fijos se encuentran e intercambian su estabilidad. Es decir, el punto fijo estable se hace inestable en el punto de bifurcación, y el punto inestable se hace estable.

3.3.4 Bifurcación de horca.

La bifurcación de horca es una bifurcación simétrica, y por ello se observa en muchos sistemas que tienen simetría entre una parte positiva y otra negativa. Las bifurcaciones de horca consisten en que un único punto fijo se bifurca en tres, uno de los cuales tiene la misma estabilidad que el original y los otros dos la contraria.

Las bifurcaciones de horca pueden clasificarse en dos grupos importantes: subcríticas y supercríticas. En una bifurcación subcrítica dos puntos fijos inestables y uno estable colapsan en un punto fijo inestable. En una bifurcación supercrítica un punto fijo estable se bifurca en dos puntos fijos estables y uno inestable.

3.3.5 Bifurcación de Hopf.

Las bifurcaciones de Hopf ocurren en osciladores no lineales. En ellas, un punto fijo se bifurca en un ciclo, o bien un ciclo colapsa en un punto fijo.

3.4 Caos en sistemas continuos.

Muchas ecuaciones no lineales son modelos matemáticos de sistemas físicos que son muy difíciles de analizar: sistemas muy inestables, o de comportamiento muy complejo. El modelo matemático suele ser muy complicado, a veces imposible de resolver de forma exacta. El estudio de estos sistemas se realiza mediante un nuevo campo de las matemáticas, llamado *Teoría del Caos.*

Los sistemas no lineales caóticos tienen dos características fundamentales. Primero, las oscilaciones son persistentemente irregulares y nunca se establecen en un patrón regular. La otra característica es su sensible dependencia a las condiciones iniciales. Esto no significa por sí solo que un sistema sea caótico, pero todos los sistemas caóticos tienen esta propiedad. Considere por ejemplo un péndulo simple. Para oscilaciones pequeñas, el movimiento del péndulo

puede determinarse de manera muy simple mediante la función seno. Si el péndulo empezara su movimiento desde un conjunto de condiciones iniciales muchas veces, su comportamiento sería a grandes rasgos el mismo para todas las pruebas.

Se ha hecho evidente que el comportamiento caótico es típico de los sistemas no lineales, pero la no linearidad no implica en todos los casos caos. Un ejemplo muy conocido de un sistema no lineal que no es caótico es el oscilador de van der Pol, que se define como:

$$\frac{\partial^2 x}{\partial t^2} + x + \epsilon(x^2 - 1)\frac{\partial x}{\partial t} = 0$$

Epsilon es una constante de la no linealidad del sistema. Si $\epsilon = 0$, el sistema es un oscilador lineal armónico. A medida que ε crece, el sistema se hace más no lineal. La siguiente figura muestra el comportamiento del oscilador para $\epsilon = 10$:

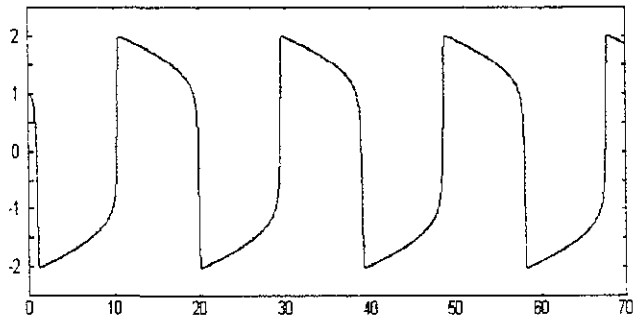

Figura 3.5 Comportamiento del oscilador de van del Pol, $\varepsilon = 10$

Si a este sistema no se le perturba inicialmente, entonces nada ocurre, de lo contrario, eventualmente oscilará con un patrón particular, con una amplitud y una frecuencia independientes de las condiciones iniciales.

Los sistemas caóticos no tienen este comportamiento. Como ya se dijo, cualquier pequeño cambio en las condiciones iniciales puede provocar en ellos un comportamiento completamente diferente. Parecen evolucionar de forma errática, y cualquier pequeña perturbación puede cambiar todo el sistema drásticamente.

3.4.1 Condiciones para el caos.

Para determinar si un sistema no lineal es caótico, es necesario que el espacio de fase asociado al sistema sea de por lo menos tres dimensiones, es decir, el caos no se presenta en sistemas de dos dimensiones.

Lo anterior se establece de un resultado llamado *Teorema de Poncairé-Bendixon* (ver [Acheson,1997]) que establece lo siguiente:

Considere el conjunto de ecuaciones diferenciales definidas como:

$$\frac{dx}{dt} = f(x,y),$$

$$\frac{dy}{dt} = g(x,y)$$

bajo las dos siguientes suposiciones:

i) Existen los puntos de equilibrio $f(x,y) = 0$ y $g(x,y) = 0$.

ii) Existe un camino de fase que inicia en algún punto y no puede dejar cierta región acotada del plano (x,y)

Entonces el Teorema de Poincaré-Bendixon afirma que solamente hay tres posibilidades:

a) Terminar en un punto de equilibrio, o

b) regresar al punto origen, formando un camino cerrado, o

c) aproximarse a un ciclo límite.

De ésto se desprende que un sistema de dos dimensiones no puede ser caótico. Se enfatiza que, para que un sistema exhiba caos, debe de ser de por lo menos de tres dimesiones, sin embargo, el hecho de que el sistema sea de tres dimensiones no implica siempre que sea caótico.

4 Diseño de observadores no lineales para sistemas caóticos Hamiltonianos.

4.1 Introducción.

Los sistemas caóticos oscilatorios han tenido un gran impacto en la Física, Biología, Ingeniería en Comunicaciones, Teoría del Control y Ciencias Atmosféricas. Como ejemplos se pueden citar los tres volúmenes especiales editados por una de las revistas más importantes en el área de Ingeniería Eléctrica y Comunicaciones (ver Special Issue [1997; 1993 and 1997]). En estas revistas se reunieron una gran colección de artículos relacionados con el estudio del caos y sus diversas aplicaciones a la ingeniería. Por otra parte, se pueden mencionar la sorprendente cantidad de libros y referencias relacionadas con el estudio del caos de los sistemas dinámicos, que han sido editados en las últimas dos décadas (ver [Holden ,1986, Mira 1987, Ott *et al.* 1994, Fradkov y Pogromsky, 1988 y Chen 1997]).

Enfocaremos nuestra atención en el mecanismo de cifrado y descifrado de señales mediante el uso de un circuito caótico oscilatorio. Este proceso es realizado mediante un *circuito emisor* : este circuito genera la señal emitida, S_E, la que se cifra por medio de un *circuito enmascarador.* Este sistema crea un conjunto de n estados caóticos, $C_i\ i = 1, 2, ..n$.Uno de los estados caóticos, $C_k\ k = 1, 2, ..n$, se mezcla con la señal emitida S_E por medio de una simple operación aritmética:

$$M_{C_k} = KC_k \mp S_E$$

donde M_C representa el mensaje trasmitido, K es un factor de escalamiento diseñado de tal forma que la señal caótica C_k sea lo suficientemente grande comparada con la señal emitida S_E. Finalmente, en el otro lado se encuentra el *circuito receptor* o *descifrador* el cual es capaz de reconstruir en forma aproximada la señal emitida S_E, a partir de las señales caóticas recibidas: M_C y C_i donde $i \neq k. \wedge i = 1, 2, ..n$. Este proceso recibe generalmente el nombre de "Sincronización de Circuitos Caóticos" (ver [Nijmeijer and Mareels, 1997], [Pecora and Carrol, 1995], [Wu and Chua, 1993] and [Willems *et al*, 1998]), el circuito receptor está diseñado por medio de un observador de estados, *i.e.* en base a los estados trasmitidos se recontruye la señal original.

Cabe mencionar que la mayoría de los esquemas propuestos anteriormente, fueron realizados bajo un marco de investigación puramente teórico y académico. Algunos de éstos fueron realizados mediante el desarrollo de un experimento en tiempo real, logrando una eficiencia en la recuperación de la señal trasmitida

de un setenta y cinco a noventa porciento de la información total (ver [Cuomo y Oppenheim , 1993]). Esta eficiencia es lo suficientemente buena en el área de comunicaciones y en el procesamiento digital de imágenes. Pero, es muy poco confiable para poder ser usado en el proceso de cifrado de un texto o en el manejo de información de un banco.

Por estas razones, en base al esquema de diseño de observadores propuesto por [Sira Ramírez y Cruz, 2000], se desarrolla una metodología para poder cifrar y descifrar cualquier tipo de información digital. Esto se logró mediante la implantación digital de los sistemas caóticos con sus respectivos observadores, usando el método de Runge-Kutta-Fehlberg (ver[Gerald, Wheatley, 1994]).

Este capítulo está organizado en cuatro secciones, en la primera se presenta una breve introducción; en la segunda se introduce el concepto de Sistema Hamiltoniano y se mencionan algunas de sus propiedades; en la tercera sección se presenta una metodología para diseñar observadores no lineales para un Sistema Hamiltoniano; en la cuarta sección se propone una aplicación para el cifrado y descifrado de información; finalmente se presentan las conclusiones del capítulo.

4.2 Sistemas Hamiltonianos.

Se dice que un sistema dinámico es Hamiltoniano, si sus respectivas ecuaciones diferenciales pueden ser obtenidas mediante el uso de las ecuaciones de *Euler-Lagrange* o *Hamilton*. Generalmente, para ser modelados, primero se obtienen o proponen las funciones de energía con sus respectivas restricciones, después se procede a obtener el conjunto de ecuaciones diferenciales que describen el comportamiento del sistema (ver [Ortega et al, 1999]).

Considere el siguiente sistema suave no lineal[1], descrito a continuación:

$$\dot{x} = \overline{J}(x)\frac{\partial H}{\partial x} + S(x)\frac{\partial H}{\partial x}, \quad x \in R^n \tag{1}$$

donde $H(x)$ es una función suave de energía que es definida positiva para toda $x \in R^n$. Generalmente $H(x)$ es una función cuadrática que puede expresarse como:

$$H(x) = \frac{1}{2}x^T M x \tag{2}$$

[1] Función suave, es una función que es continua y derivable.

donde M es una matriz simétrica definida positiva. La mayoría de las ocasiones, M es una matriz función del estado x *i.e.* $M = M(x)$. En está sección se asumirá que M es una matriz constante. En este caso claramente se tiene:

$$\frac{\partial H}{\partial x} = Mx. \tag{3}$$

La matriz $\overline{J}(x)$, puede ser expresada como la suma de dos matrices *i.e.* $\overline{J}(x) = J + J_D(x)$ y cumple con la siguiente propiedad[2]:

$$J_D(x) + J_D^T(x) = 0; \quad \text{y} \quad J + J^T = 0. \tag{4}$$

La matriz $S(x)$, puede ser escrita como la suma de dos matrices $P(x)$ y $N(x)$; donde $P(x)$ es una matriz simétrica definida positiva y $Q(x)$ es una matriz simétrica definida negativa, por lo que:

$$S(x) \triangleq P(x) + N(x) = S^T(x) \tag{5}$$

En consecuencia de las propiedades (4) y (5) claramente tenemos:

$$\frac{\partial H}{\partial x}^T J(x) \frac{\partial H}{\partial x} = 0; \frac{\partial H}{\partial x}^T P(x) \frac{\partial H}{\partial x} \geq 0; \frac{\partial H}{\partial x}^T N(x) \frac{\partial H}{\partial x} \leq 0. \tag{6}$$

Algunas ocasiones, cuando se desea diseñar un observador o cuando se desea estudiar el comportamiento de sistemas que exhiben un comportamiento caótico, se puede describir al sistema dinámico (1), como sigue:

$$\dot{x} = \overline{J}(x) \frac{\partial H}{\partial x} + S(x) \frac{\partial H}{\partial x} + F(x), \tag{7}$$

[2] La propiedad recibe el nombre de matriz antisimétrica.

donde $F(x)$ representa el vector desestabilizador, que generalmente representa la parte de la dinámica no lineal que produce caos. Esto se debe básicamente a que el vector $F(x)$ es lo suficientemente grande para poder crear caos. Por otro lado, el vector desestabilizador $F(x)$ nos sugiere cuáles estados se deberán tomar como salidas del sistema caótico (7). Por ejemplo, sea $F(x) = F(x_1)$ *i.e.* sólo es función del primer estado x_1,entonces nos conviene tomar como señal de salida o trasmisión a la variable x_1.

A continuación se dan una serie de ejemplos, en donde se mostrará que varios sistemas caóticos pueden ser escritos como se muestra en (7).

4.3 Ejemplos de sistemas caóticos Hamiltonianos.

En esta sección presentaremos una serie de ejemplos que están relacionados con la forma canónica de la ecuación diferencial (7). Se pondrá especial atención al sistema de Lorenz, ya que este sistema caótico será usado para cifrar y descifrar información digital (ver siguiente capítulo).

Ejemplo 1: Considere el sistema de Lorenz (ver [Lorenz, 1963]):

$$\dot{x}_1 = \sigma(x_2 - x_1)$$

$$x_2 = rx_1 - x_2 - x_1x_3$$

$$\dot{x}_3 = x_1x_2 - bx_3 \tag{8}$$

fácilmente puede ser expresado como un sistema Hamiltoniano. Se propone como función Hamiltoniana de energía, la siguiente función cuadrática:

$$H(x) = \frac{1}{2}(\frac{x_1^2}{\sigma} + x_2^2 + x_3^2), \tag{9}$$

donde M=diagonal$\{\frac{1}{\sigma}, 1, 1\}$ y $\frac{\partial H}{\partial x}^T = (\frac{1}{\sigma}x_1, x_2, x_3)$. Finalmente, nótese que (8) puede ser escrita en la forma de la ecuación (7), donde las matrices $J(x)$, $S(x)$ y $F(x)$, están definidas como:

$$\overline{J}(x) = \begin{bmatrix} 0 & \sigma/2 & 0 \\ -\sigma/2 & 0 & -x_1 \\ 0 & x_1 & 0 \end{bmatrix}; \quad F(x) = \begin{bmatrix} 0 \\ rx_1 \\ 0 \end{bmatrix} \tag{10}$$

$$S(x) = \begin{bmatrix} -\sigma^2 & \sigma/2 & 0 \\ \sigma/2 & -1 & 0 \\ 0 & 0 & -b \end{bmatrix}; \tag{11}$$

Ejemplo 2: Considere el sistema caótico de Chen (ver [Sira Ramírez y C. Cruz, 1999]):

$$\begin{aligned} \dot{x}_1 &= a(x_2 - x_1) \\ x_2 &= (c-a)x_1 + cx_2 - x_1x_3 \\ \dot{x}_3 &= x_1x_2 - bx_3 \end{aligned} \tag{12}$$

tomando la función Hamiltoniana de energía como sigue:

$$H(x) = \frac{1}{2}(x_1^2 + x_2^2 + x_3^2) \tag{13}$$

donde M=diagonal$\{1, 1, 1\}$ y $\frac{\partial H}{\partial x}^T = (x_1, x_2, x_3)$. Expresando (12) como en (7), se tiene que $J(x)$, $S(x)$ y $F(x)$, están definidas como:

$$\overline{J}(x) = \begin{bmatrix} 0 & a - c/2 & 0 \\ -a + c/2 & 0 & -x_1 \\ 0 & x_1 & 0 \end{bmatrix}; F(x) = \begin{bmatrix} x_2(c/2) \\ x_1(c/2) + cx_2 \\ 0 \end{bmatrix}; \tag{14}$$

$$S(x) = \begin{bmatrix} -a & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -b \end{bmatrix} \tag{15}$$

Ejemplo 3: Considere el sistema caótico conocido como sistema de Rössler (ver [Acheson D., 1997]):

$$\begin{aligned} \dot{x}_1 &= -x_2 - x_3; \\ \dot{x}_2 &= x_1 + ax_2; \\ \dot{x}_3 &= b + x_3(x_1 - c); \end{aligned} \tag{16}$$

se tomará la siguiente función Hamiltoniana de energía, como sigue:

$$H(x) = \frac{1}{2}(x_1^2 + x_2^2 + x_3^2) \tag{17}$$

donde M=diagonal$\{1,1,1\}$ y $\frac{\partial H}{\partial x}^T = (x_1, x_2, x_3)$. Finalmente, se expresa (16) como en (7), donde $J(x)$, $S(x)$ y $F(x)$, están definidas como siguen:

$$\overline{J}(x) = \begin{bmatrix} 0 & -1 & -1/2 \\ 1 & 0 & 0 \\ 1/2 & 0 & 0 \end{bmatrix}; \; F(x) = \begin{bmatrix} 0 \\ 0 \\ b + x_1 x_3 \end{bmatrix}$$

$$S(x) = \begin{bmatrix} 0 & 0 & -1/2 \\ c/2 & a & 0 \\ -1/2 & 0 & -c \end{bmatrix} \tag{18}$$

4.4 Diseño de observadores no lineales para un sistema Hamiltoniano.

En esta sección se diseña un observador asintótico de tipo Luenberger (ver [Apéndice A]) para poder estimar los estados en función de las salidas. Para esto se divide el mecanismo de diseño en tres pasos:

i) Se factoriza el sistema caótico como un sistema de Hamilton.

ii) Se propone un observador de estado con su respectivo vector de ganancias, y más adelante se hace el análisis de estabilidad.

iii) Finalmente se propone un teorema el cual nos da condiciones necesarias sobre el vector de ganancias para poder garantizar que el error converge exponencialmente a cero.

Primer Paso: Se escribe el sistema caótico como un Sistema Hamiltoniano:

Considere el siguiente sistema:

$$\dot{x} = J_D(y)\frac{\partial H}{\partial x} + (J+S)\frac{\partial H}{\partial x} + F(y);$$

$$y = C\frac{\partial H}{\partial x}, \quad y \in R\ , C \in R^{1\times n}. \tag{19}$$

Donde S es una matriz simétrica constante no necesariamente con signo definido. $J_D(y)$ y J son matrices antisimétricas (ver 4). La variable escalar y, es considerada como la salida del sistema. C es una matriz con un renglón constante.

Segundo Paso: Se propone el observador:

Sea $\widehat{x}$ el vector estimado correspondiente al estado x, y considere la función de energía $H(x) = x^T Mx/2$ relacionada con el sistema (19). Sea $\widehat{y}$ el vector de salida estimada correspondiente a y, el cual es función del vector de estado estimado $\widehat{x}$.

Se propone el siguiente observador para el sistema no lineal (19), dado como sigue:

$$\dot{\widehat{x}} = J_D(y)\frac{\partial H}{\partial \widehat{x}} + (J+S)\frac{\partial H}{\partial \widehat{x}} + F(y) + K(y-\widehat{y});$$

$$\widehat{y} = C\frac{\partial H}{\partial \widehat{x}}. \tag{20}$$

donde K es un vector constante, conocido como vector de ganancia del observador. Claramente por (3), se tiene $M\widehat{x} = \partial H/\partial \widehat{x}$.

Tercer Paso: Análisis de estabilidad:

Se define el error e, como $e = x - \widehat{x}$ y se calcula la diferencia entre la primera ecuación de (19) y la primera ecuación de (??), así tenemos que:

$$\dot{e} = J_D(y)\frac{\partial H}{\partial e} + (J + S - KC)\frac{\partial H}{\partial e};$$

$$e_y = C\frac{\partial H}{\partial e}, \tag{21}$$

donde:

$$\frac{\partial H(e)}{\partial e} \triangleq \frac{\partial H}{\partial x} - \frac{\partial H}{\partial \widehat{x}} = M(x - \widehat{x}) = Me. \tag{22}$$

Se procede a realizar el análisis de estabilidad.

Se propone la función Hamiltoniana como una función de Lyapunov (ver [Apéndice A]):

$$H(e) = \frac{1}{2}e^T Me.$$

nótese que $H(e)$ es definida positiva para toda e, ya que la matriz M es definida positiva.

Se procede a calcular la derivada de la función Hamiltoniana a lo largo de las trayectorias de (21), se tiene lo siguiente:

$$\frac{d}{dt}H(e) = e^T Me + e^T M\dot{e};$$

$$= \frac{\partial H}{\partial e}^T \left\{J_D^T(y) + (J^T + S^T - C^T K^T)\right\} Me$$

$$+e^T M \left\{J_D(y) + (J + S - KC)\right\} \frac{\partial H}{\partial e}; \tag{23}$$

de (22) se obtiene:

$$\frac{\partial H}{\partial e} = Me, \tag{24}$$

Por otro lado de (24) y (4) se puede simplificar (23), como sigue:

$$\frac{d}{dt}H(e) = \frac{\partial H}{\partial e}^T \left[J_D^T(y) + J^T + S^T - C^T K^T + J_D(y) + J + S - KC)\right] \frac{\partial H}{\partial e}$$

$$= \frac{\partial H}{\partial e}^T \left(S - \frac{1}{2}(C^T K^T + KC)\right) \frac{\partial H}{\partial e};$$

aplicando la desigualdad de Ostrosky a la última expresión [Ruth, 1995][3], se obtiene la siguiente desigualdad:

$$\frac{d}{dt}H(e) \leq -k\frac{\partial H}{\partial e}^T \frac{\partial H}{\partial e} \leq -k \left\|e\right\|^2 \lambda_{\min}\left[M^2\right] \tag{25}$$

donde el vector de ganancia K se escoge de tal forma que[4]:

$$k \triangleq \frac{1}{2}\lambda_{\min}[C^T K^T + KC] - \lambda_{\max}[S] > 0. \tag{26}$$

Finalmente, resumiremos los pasos anteriores en la siguiente proposición:

Teorema principal: *El vector de estado x del sistema no lineal (19) converge exponencialmente al estado estimado $\widehat{x}$ del observador (21), si el vector de ganancia K es escogido de tal forma que la constante k definida en (26) sea estrictamente positiva.*

---

[3] Sea P una matriz simétrica entonces $\lambda_{\min}[P]\left\|x\right\|^2 \leq x^T P x \leq \lambda_{\max}[P]\left\|x\right\|^2$.
donde $\lambda_{\min}[P]$ denota el eigenvalor más pequeño
y $\lambda_{\max}[P]$ denota el eigenvalor más grande.

[4] Note que las matrices M, S y $\frac{1}{2}(C^T K^T + KC)$ son matrices simétricas.

4.5 Aplicación: Cifrado y Descifrado de Información

En esta sección se aplica el teorema anterior para cifrar y descifrar señales digitales en forma eficiente, para lo que se propone el siguiente algoritmo:

1) Dado el sistema caótico, se procede a escribirlo en su forma canónica Hamiltoniana. De esa forma se identifica el vector desestabilizador $F(x)$.

Si el vector desestabilizador es función no lineal de un estado *i.e.* x_k donde $k \in \{1,2,..,n\}$, entonces se toma la salida $y = x_k$. Si el vector $F(x)$ con $x \in R^n$ es una función no lineal de dos estados *i.e.* $F(x_k, x_j)$ donde, j, $k \in \{1,2,..,n\}$ entonces se toman dos salidas $y_1 = x_k$ y $y_2 = x_j$. Y así sucesivamente se establecen las salidas en función de las variables del vector desestabilizador. Si el vector desestabilizador depende de los n estados entonces no es posible construir el observador, ya que necesitaríamos conocer el vector de estado. Finalmente si $F(x) = 0$, entonces podemos escoger cualquier estado.

2) Se propone el siguiente esquema de cifrado. Supóngase que la salida $y = x_1$ y que deseamos cifrar los mensaje w_1 y w_2, entonces se emplean las señales caóticas $\{y\ , m_1 = x_2 + \lambda_1 w_1\ ,\ m_2 = x_3 + \lambda_2 w_2\}$ donde λ_1 y λ_2 son factores de escalamiento propuestos de la siguiente forma:

$$\max_{w_1} |\lambda_1 w_1| << \max |x_2| \quad \text{y} \quad \max_{w_2} |\lambda_2 w_2| << \max |x_3| \,.$$

3) Una vez que se tienen bien identificadas las matrices $\overline{J}(x), S, F(x)$ y C, se propone el observador como en (??). Si la matriz S es definida negativa entonces no es necesario incluir la inyección en el error de la salida *i.e* . $K = 0$. En caso contrario se encuentra K de tal forma que k sea estrictamente positiva [ver (26)].

4) El observador recibe las señales caóticas $\{y, m_1, m_2\}$. Y en función de la salida y, estima los estados $\widehat{x}_2$ y $\widehat{x}_3$. Así se propone el siguiente descifrador:

$$\widehat{w}_1 = \frac{1}{\lambda_1}(m_1 - \widehat{x}_2); \quad \widehat{w}_2 = \frac{1}{\lambda_2}(m_2 - \widehat{x}_3),$$

A partir de un tiempo pequeño podemos garantizar que $\widehat{w}_1 = w_1$ y $\widehat{w}_2 = w_2$. El tiempo t, depende básicamente de dos factores:

i) Si las condiciones iniciales del sistema caótico (19) son cercanas a las condiciones iniciales del observador (21), entonces t es pequeño, en caso contrario necesita más tiempo para descifrar el mensaje original.

ii) Dependiendo de la magnitud del vector de ganancia K, se tiene lo siguiente: si la magnitud de K es grande entonces t es pequeño de otra forma t es más grande.

4.6 Ejemplos de observadores no lineales para sistemas Hamiltonianos.

4.6.1 Sistema de Lorenz.

Considere nuevamente el sistema de Lorenz del ejemplo 1 (ver [8 a 11]). En este caso como la matriz S es simétrica, definida negativa, entonces todos sus eigenvalores son negativos, y no necesitamos inyectar la salida, es decir, $K = 0$. Como el vector desestabilizante $F(x)$ depende del estado x_1, entonces tomamos la salida $y = x_1 = [\sigma, 0, 0]\frac{\partial K}{\partial x}$. Así el observador queda de la siguiente manera:

$$\begin{bmatrix} \dot{\widehat{x}}_1 \\ \dot{\widehat{x}}_2 \\ \dot{\widehat{x}}_3 \end{bmatrix} = \begin{bmatrix} 0 & \sigma/2 & 0 \\ -\sigma/2 & 0 & -y \\ 0 & y & 0 \end{bmatrix} \frac{\partial H}{\partial \widehat{x}} + \begin{bmatrix} -\sigma^2 & \sigma/2 & 0 \\ \sigma/2 & -1 & 0 \\ 0 & 0 & -b \end{bmatrix} \frac{\partial H}{\partial \widehat{x}} + \begin{bmatrix} 0 \\ ry \\ 0 \end{bmatrix}. \tag{27}$$

Restando (8) y (27) para el caso $y = x_1$, obtenemos la ecuación del error:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 0 & \sigma/2 & 0 \\ -\sigma/2 & 0 & -y \\ 0 & y & 0 \end{bmatrix} \frac{\partial H}{\partial e} + \begin{bmatrix} -\sigma^2 & \sigma/2 & 0 \\ \sigma/2 & -1 & 0 \\ 0 & 0 & -b \end{bmatrix} \frac{\partial H}{\partial e} \tag{28}$$

Recuérdese que $e^T = (e_1, e_2, e_3)$ y $\frac{\partial H}{\partial e}^T = (\frac{1}{\sigma}e_1, e_2, e_3)$. Claramente la ecuación (28) es exponencialmente estable, según los resultados del teorema principal.

4.6.2 Sistema de Chen:

Considere nuevamente el sistema de Chen del ejemplo 2 [ver (12 a 14)]. En este caso como la matriz S es no definida negativa, ya que el parámetro $c > 0$, entonces es necesario inyectar la salida, es decir, $K \neq 0$. Como el vector desestabilizante $F(x) = 0$ no depende del estado x, entonces podemos tomar el estado $y_1 = x_1$ y $y_2 = x_2$. Así, el observador queda de la siguiente manera:

$$\begin{bmatrix} \dot{\widehat{x}}_1 \\ \dot{\widehat{x}}_2 \\ \dot{\widehat{x}}_3 \end{bmatrix} = \begin{bmatrix} 0 & a-c/2 & 0 \\ -a+c/2 & 0 & -x_1 \\ 0 & x_1 & 0 \end{bmatrix} \frac{\partial H}{\partial \widehat{x}} + \begin{bmatrix} -a & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -b \end{bmatrix} \frac{\partial H}{\partial \widehat{x}} \quad (29)$$
$$+ \begin{bmatrix} x_2(c/2) \\ x_1(c/2)+cx_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ K \\ 0 \end{bmatrix} (x_2 - \widehat{x}_2).$$

Restando (12) y (29), obtenemos la ecuación del error:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 0 & a-c/2 & 0 \\ -a+c/2 & 0 & -x_1 \\ 0 & x_1 & 0 \end{bmatrix} \frac{\partial H}{\partial e} \quad (30)$$
$$+ \begin{bmatrix} -a & 0 & 0 \\ 0 & -K & 0 \\ 0 & 0 & -b \end{bmatrix} \frac{\partial H}{\partial e}$$

Para poder garantizar que la ecuación del error (30) sea exponencialmente estable, se debe garantizar que la matriz:

$$W = \begin{bmatrix} -a & 0 & 0 \\ 0 & -K & 0 \\ 0 & 0 & -b \end{bmatrix}$$

sea definida negativa, lo cual se puede garantizar para toda $K > 0$.

4.7 Conclusiones

En este capítulo se presentó una metodología para cifrar y descifrar cualquier tipo de información representada en forma digital, basada en la naturaleza caótica, casi aleatoria de cierto tipo de circuitos oscilatorios, como los que se describieron en las secciones (4.3) a (4.6) (ver ejemplos [Lorenz y Chen]).

Básicamente, este algoritmo puede resumirse de la siguiente manera:

i) Se mezcla la señal que se desea cifrar con una variable del sistema caótico. Esta variable se escoge de tal manera que pueda ser reconstruida a través de una o varias salidas procedentes del circuito caótico emisor.

ii) El mecanismo de descifrado de la señal, puede ser realizado casi en forma inmediata, si las condiciones iniciales del *circuito cifrador* y del *circuito descifrador u observador*, están cercanas. Si las condiciones iniciales de ambos circuitos están muy alejadas es conveniente aumentar la ganancia del observador y de esta forma optimizar el tiempo de recuperación de la o las señales.

Fundamentalmente para poder cifrar y descifrar se propone el uso y diseño de un observador asintótico. Este diseño está basado en las propiedades de los sistemas Hamiltonianos [ver (Sistemas Hamiltonianos)], cuya principal característica es que pueden factorizarse como la suma de una *matriz antisimétrica* y una *matriz simétrica* más un vector desestabilizante [ver (4 y 7)]. En base a esta estructura se propone un observador, que puede ser considerado como una pseudo-copia del sistema caótico original [ver (??)]. Este observador incluye a la *salida del emisor* como variable dependiente del observador y al error de las salidas entre ambos sistemas *i.e. salida del emisor y salida del observador*. La prueba del error que existe entre el estado original y el estado estimado fue hecha en base a la teoría de Lyapunov. Se propone una función de Lyapunov que es el Hamiltoniano del *sistema caótico original*, y se prueba que la derivada respecto al tiempo de la función Hamiltoniana alrededor de las trayectorias de la ecuación del error [ver (21)], puede ser definida negativa [ver derivada de la función de energía (25)], para ciertos vectores de ganancia K. De esta forma se garantiza que el error entre el estado emitido y el estimado convergen exponencialmente a cero.

5 Instrumentación numérica.

5.1 Introducción.

En este capítulo se presenta un desarrollo numérico diseñado en base a los resultados previos, en particular a lo mencionado en el capítulo 3. Para demostrar la propuesta que se hizo, se desarrollan las simulaciones (empleando SIMNOM) de los sistemas de Lorenz y Chen, con sus respectivos observadores de estado, diseñados en base a las propiedades pasivas de los sistemas Hamiltonianos. Después se presenta un experimento numérico, mediante la implantación de dos programas escritos en lenguaje de programación C, que permiten corroborar la posibilidad de un criptosistema como el que se propuso.

Para obtener las soluciones numéricas de los sistemas de ecuaciones diferenciales que modelan los sistemas caóticos (emisor) y observadores (receptor), se instrumentó el método conocido como Runge-Kutta-Feldberg (RKF) (ver[Gerald y Wheatley, 1994]), bajo las siguientes suposiciones:

i) Cualquier sistema caótico continuo puede ser aproximado a un sistema dinámico discreto mediante el método RKF, *i.e.*, existe un paso de integración lo suficientemente pequeño, tal que, el sistema discreto exhibe un comportamiento caótico en la solución numérica.

ii) El error introducido por el método RKF y el correspondiente al manejo del punto flotante en la computadora empleada, para la generación de la señal o las señales de cifrado, es reproducido de manera indéntica en el proceso de observación para la generación de la o las señales de descifrado.

iii) El experimento asume que cualquier problema en la transmisión de la información, es resuelto por la plataforma y los mecanismos de comunicación subyacentes a la plataforma de software empleada (como ejemplo se menciona internet).

En base a las suposiciones anteriores se desarrolló un experimento en el que se propone cifrar y descifrar una imagen y un texto. Los resultados que adelante se proporcionan, nos permiten demostrar que sí es posible elaborar un sistema de cifrado y descifrado de información empleando sistemas no lineales caóticos.

En este capítulo se muestran las simulaciones numéricas correspondientes al sistema de Lorenz y del sistema de Chen, elaboradas con la herramienta SIMNOM: y el desarrollo del experimento para cifrar y descifrar una señal digital empleando el sistema de Lorenz y el observador pasivo de sus estados. ambos en su forma discreta aproximada.

5.2 Simulación numérica empleando SIMNOM.

5.2.1 Sistema de Lorenz.

Considere el sistema de Lorenz[5]:

$$\begin{aligned} \dot{x}_1 &= 10(x_2 - x_1) \\ \dot{x}_2 &= 28x_1 - x_2 - x_1x_3 \\ \dot{x}_3 &= x_1x_2 - \tfrac{8}{3}x_3 \\ y &= x_1 \end{aligned} \tag{31}$$

con condiciones iniciales: $x_1 = 5, x_2 = -2$ y $x_3 = -5$; y considere el observador pasivo de sus estados:

$$\begin{aligned} \dot{\widehat{x}}_1 &= 10(x_1 - \widehat{x}_2) \\ \dot{\widehat{x}}_2 &= -\widehat{x}_2 - x_1\widehat{x}_3 + 28x_1 \\ \dot{\widehat{x}}_3 &= x_1\widehat{x}_2 - \tfrac{8}{3}\widehat{x}_3 \end{aligned} \tag{32}$$

con condiciones iniciales: $\widehat{x}_1 = -10$, $\widehat{x}_2 = 5$ y $\widehat{x}_3 = 1$.
Con la herramienta de simulación SIMNOM se simularon simultáneamente los sistemas (31) y (32).
Las gráficas de las trayectorias correspondientes a los estados del sistema (31) (x_1, x_2, x_3) y los estados del observador (32) $(\widehat{x}_1, \widehat{x}_2, \widehat{x}_3)$ se muestran en las siguientes figuras:

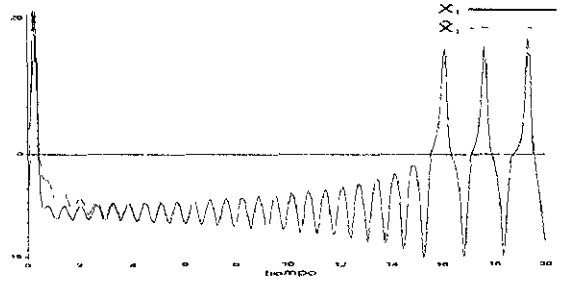

Fig. 5.1a x_1 y $\widehat{x}_1$

[5]Se eligió el sistema caótico de Lorenz y su observador respectivo para la implantación del prototipo del esquema de cifrado/descifrado que se propone, y se presenta en la siguiente sección

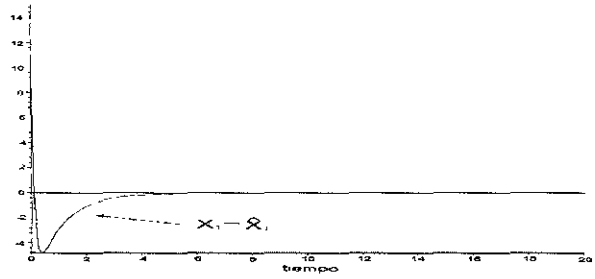

Fig. 5.1b $x_1 - \widehat{x}_1$

Las figuras 5.1a y 5.1b, muestran en una gráfica el comportamiento de los estados x_1y $\widehat{x}_1$ y el error $x_1 - \widehat{x}_1$, respectivamente. Es importante señalar que $x_1$es la salida del sistema emisor y que el observador reconstruye los estados x_2 y x_3 a partir de la salida $y = x_1$. Se puede observar de las figuras 5.1a y 5.1b, que a pesar de que el error inicial entre los estados es $x_1 - \widehat{x}_1 = 20$, a partir de un tiempo $t = 5$ segundos, es del orden de 10^{-2} y decrece exponencialmente hasta un rango de 10^{-4} a partir de $t = 10$ segundos.

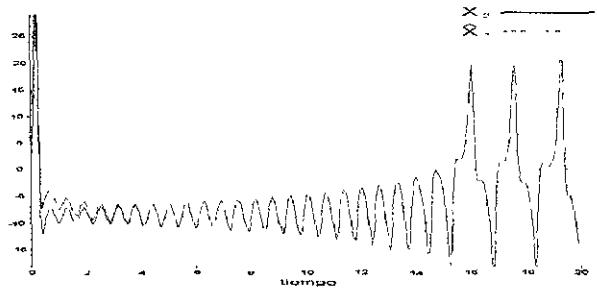

Fig. 5.2a x_2 y $\widehat{x}_2$

A continuación se muestra el código empleado en la simulación realizada en SIMNOM para obtener las gráficas de las figuras 5.1a, 5.1b, 5.2a, 5.2b, 5.3a y 5.3b.:

```
CONTINUOUS SYSTEM LORENZ
    STATE x1 x2 x3 y1 y2 y3
    DER dx1 dx2 dx3 dy1 dy2 dy3
    TIME t
    x1:5
    x2:-2
    x3:-5
    y1:-10
    y2:5
    y3:1
    s=10
    r=28
    b=8/3
/* Sistema de Lorenz */
    dx1=s*(x2-x1)
    dx2=r*x1-x2-x1*x3
    dx3=x1*x2-b*x3
/* Fin del sistema de Lorenz */
/* Observador */
    dy1=s*(y2-y1)
    dy2=r*x1-y2-x1*x3
    dy3=x1*y2-b*y3
/* Fin del observador */
    e1=x1-y1
    e2=x2-y2
    e3=x3-y3
END
```

5.2.2 Sistema de Chen.

Considere el sistema de Chen:

$$\begin{aligned} \dot{x}_1 &= 35(x_2 - x_1) \\ \dot{x}_2 &= -7x_1 - x_1x_3 + 28x_2 \\ \dot{x}_3 &= x_1x_2 - bx_3 \end{aligned} \tag{33}$$

con condiciones iniciales: $x_1 = 1, x_2 = 2$ y $x_3 = 3$; y considere el observador pasivo de sus estados:

$$\begin{aligned} \dot{\widehat{x}}_1 &= \widehat{x}_2(a - \tfrac{c}{2}) - a\widehat{x}_1 + x_2(\tfrac{c}{2}) \\ \dot{\widehat{x}}_2 &= (-35 + 14)\widehat{x}_1 - x_1\widehat{x}_3 + 28x_2 + 14x_1 + (x_2 - \widehat{x}_2) \\ \dot{\widehat{x}}_3 &= x_1\widehat{x}_2 - 3\widehat{x}_3 \end{aligned} \tag{34}$$

con condiciones iniciales: $\widehat{x}_1 = 10$, $\widehat{x}_2 = -20$ y $\widehat{x}_3 = 20$ y $k = 57.4$.

Las gráficas de las trayectorias correspondientes a los estados de los sistemas (33) y (34) obtenidas en la simulación numérica realizada en SIMNOM se muestran en las siguientes figuras:

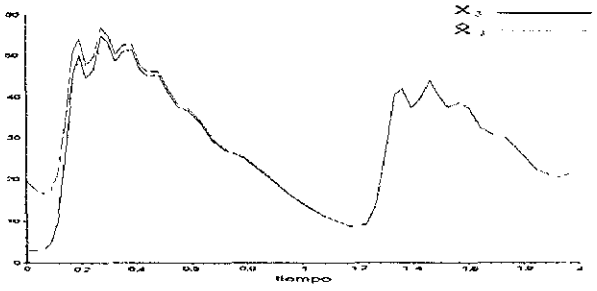

Fig. 5.4a x_3y $\widehat{x}_3$

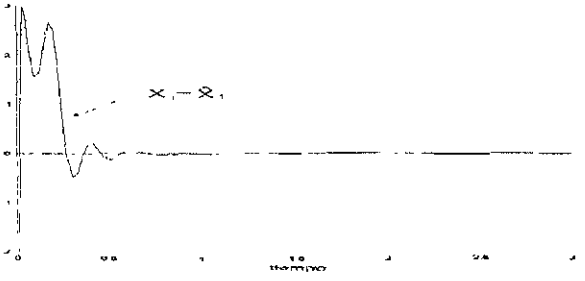

Fig. 5.4b $x_1 - \widehat{x}_1$

Las figuras 5.4a y 5.4b, muestran las trayectorias x_1y $\widehat{x}_1$y el error $x_1 - \widehat{x}_1$, respectivamente.

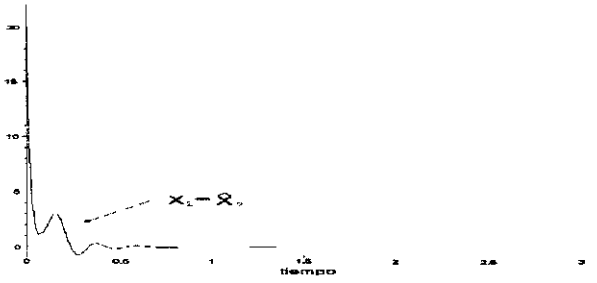

Fig. 5.5 $x_2 - \widehat{x}_2$

La figura 5.5, muestra el error $x_2 - \widehat{x}_2$.

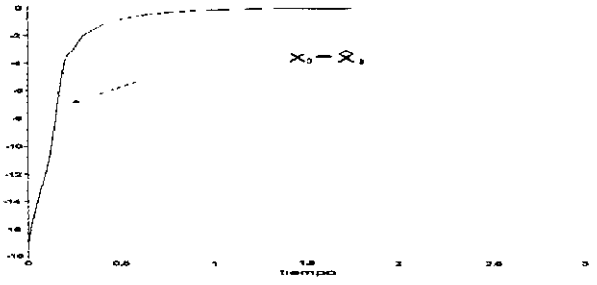

Fig. 5.6 $x_3 - \widehat{x}_3$

La figura 5.6 muestra el error $x_3 - \widehat{x}_3$.

A continuación se muestra el código empleado en la simulación realizada para obtener las gráficas de las figuras 5.4a, 5.4b, 5.5 y 5.6:

```
CONTINUOUS SYSTEM CHEN
    STATE x1 x2 x3 e1 e2 e3
    DER dx1 dx2 dx3 de1 de2 de3
    TIME t
    x1:1
    x2:2
    x3:3
    e1:10
    e2:-20
    e3:20
    a=35
    b=3
    c=28
    k=0
/* Sistema de Chen */
    dx1=a*(x2-x1)
    dx2=(c-a)*x1-x1*x3+c*x2
    dx3=x1*x2-b*x3
/* Fin del sistema de Chen */
/* Observador */
    de1=e2*(a-c/2)-a*e1+x2*c/2
    de2=(-a+c/2)*e1-x1*e3+c*x2+x1*c/2+(x2-e2)
    de3=x1*e2-b*e3
/* Fin del observador */
    err1=x1-e1
    err2=x2-e2
    err3=x3-e3
END
```

5.3 Experimentos numéricos.

Este apartado está dedicado a la implantación en el lenguaje de programación C del sistema de Lorenz y el observador pasivo de sus estados. En principio, se obtiene una aproximación discreta del sistema de Lorenz y su observador, empleando el método RKF, que se describe a continuación; posteriormente se presentan los resultados obtenidos al emplear la implantación del esquema de cifrado y descifrado.

5.3.1 Método Runge-Kutta-Fehlberg.

La solución de una ecuación diferencial es una función que satisface a la ecuación y satisface también algunas condiciones iniciales de dicha función. Al resolver una ecuación diferencial analíticamente, se encuentra una solución general que contiene constantes arbitrarias que se instancian para satisfacer ciertas condiciones iniciales. Sin embargo, los métodos analíticos están limitados a ciertas formas especiales de las ecuaciones, lo que limita la solución de gran parte de éstas. Aquellas ecuaciones que no pueden ser resueltas analíticamente, pueden ser resueltas mediante métodos numéricos, que no están limitados a formas especiales de las ecuaciones, pero independientemente del método empleado, siempre se tendrá un error en la precisión de la solución y si las condiciones inciales cambian, el cálculo de la solución deberá de volverse a realizar.

Uno de los métodos numéricos más sencillos es el conocido como método de Euler. Este método es empleado para resolver ecuaciones de primer orden, tomando únicamente los dos primeros términos de la serie de Taylor y un paso de integración h suficientemente pequeño. El método de Euler se expresa como el siguiente algoritmo:

$$y(x_0+h) = y(x_0) + y'(x_0) + \frac{y''(\xi)h^2}{2},$$

$$x_0 < \xi < x_0 + h \tag{35}$$

En (35), el valor de $y(x_0)$ está dado por la condición inicial y $y'(x_0)$ se evalúa de $f(x_0, y_0)$ que está dado por la ecuación diferencial, $\frac{dy}{dx} = f(x,y)$. Este método se emplea iterativamente, avanzando hacia la solución $x = x_0 + 2h$ después de que $y(x_0+h)$ se ha calculado, y así sucesivamente. Empleando una notación de subíndices y expresando el error como una relación de orden, la ecuación (35) se puede reescribir como:

$$y_{n+1} = y_n + hy'_n + O(h^2) \tag{36}$$

El método de Euler tiene una precisión proporcional a h^2, por lo que, si se quiere reducir el error es necesario emplear una h muy pequeña. Este método tiene una variante llamada método modificado de Euler, que permite una reducción en el error a $O(h)$.

El método de Euler sirve como una introducción a un grupo de métodos llamado Runge-Kutta, que emplean más términos de la expansión de la serie de Taylor. En cierto sentido el método de Euler es considerado como un Runge-Kutta de segundo orden, ya que toma los dos primeros términos de Taylor.

Para entender como se desarrollan los métodos Runge-Kutta, haré una derivación del método de segundo orden. En este caso, el incremento de y es el promedio de dos estimaciones del cambio de y cuando x se incrementa en h y que nombraremos como k_1 y k_2. Entonces para la ecuación:

$$\frac{dy}{dx} = f(x, y)$$

el método Runge-Kutta de segundo orden que la resuelve es:

$$\begin{aligned} y_{n+1} &= y_n + ak_1 + bk_2, \\ k_1 &= hf(x_n, y_n), \\ k_2 &= hf(x_n + \alpha h, y_n + \beta k_1), \end{aligned}$$

El método Runge-Kutta de cuarto orden se obtiene de manera similar al de segundo orden, y tiene una mayor precisión. El siguiente algoritmo permite resolver numéricamente una ecuación diferencial empleando el método Runge-Kutta de cuarto orden:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1)$$

$$k_3 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2)$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

El método Runge-Kutta-Fehlberg emplea un Runge-Kutta de cuarto orden y uno de quinto para moverse de (x_n, y_n) a (x_{n+1}, y_{n+1}), y el error se estima como la diferencia de las dos y calculadas en $x = x_{n+1}$, y es de orden $O(h^5)$. El siguiente algoritmo permite resolver numéricamente una ecuación diferencial empleando el método Runge-Kutta-Fehlberg:

$$k_1 = hf(x_n, y_n),$$

$$k_2 = hf(x_n + \frac{h}{4}, y_n + \frac{k_1}{4})$$

$$k_3 = hf(x_n + \frac{3h}{8}, y_n + \frac{3k_1}{32} + \frac{9k_2}{32})$$

$$k_4 = hf(x_n + \frac{12h}{13}, y_n + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197})$$

$$k_5 = hf(x_n + h, y_n + \frac{439k_1}{216} - 8k_2 + \frac{3680k_3}{513} - \frac{845k_4}{4104})$$

$$k_6 = hf(x_n + \frac{h}{2}, y_n - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40})$$

$$y_{n+1} = y_n + (\frac{16k_1}{135} + \frac{6656k_3}{12825} + \frac{28561k_4}{56430} - \frac{9k_5}{50} + \frac{2k_6}{55}) \tag{37}$$

5.3.2 Discretización del sistema de Lorenz y su observador empleando el método Runge-Kutta-Fehlberg.

Considere el sistema de Lorenz:

$$\begin{aligned} x_1 &= 10(x_2 - x_1) \\ x_2 &= 28x_1 - x_2 - x_1x_3 \\ \dot{x}_3 &= x_1x_2 - \tfrac{8}{3}x_3; \\ y &= x_1 \end{aligned} \tag{38}$$

y considere el observador pasivo de sus estados:

$$\begin{aligned} \widehat{x}_1 &= 10(\widehat{x}_1 - \widehat{x}_2); \\ \dot{\widehat{x}}_2 &= -\widehat{x}_2 - x_1\widehat{x}_3 + 28x_1; \\ \widehat{x}_3 &= x_1\widehat{x}_2 - \tfrac{8}{3}\widehat{x}_3; \end{aligned} \tag{39}$$

Empleando el algoritmo que implanta el método Runge-Kutta-Fehlberg (37), el sistema de ecuaciones (38) se puede discretizar como sigue:

Discretización de la primera ecuación:

$$k_1 = h(10(x_2 - x_1)),$$

$$k_2 = h(10((x_2 + \frac{k_1}{4}) - (x1 + \frac{k_1}{4}))),$$

$$k_3 = h(10((x_2 + \frac{3k_1}{32} + \frac{9k_2}{32}) - (x_1 + \frac{3k_1}{32} + \frac{9k_2}{32}))),$$

$$k_4 = h(10((x_2 + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197}) - (x_2 + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197}))),$$

$$k_5 = h(10((x_2 + \frac{439k_1}{216} - 8k_2 + \frac{368k_3}{513} - \frac{845k_4}{4104}) - (x_1 + \frac{439k_1}{216} - 8k_2 + \frac{3680k_3}{513} - \frac{845k_4}{4104}))),$$

$$k6 = h(10((x_2 - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40}) - (x_1 - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40})))$$

$$y_{n+1} = y_n + (\frac{16k_1}{135} + 6656k_3 + \frac{28561k_4}{56430} - \frac{9k_5}{50} + \frac{2k_6}{55})$$

Discretización de la segunda ecuación:

$$k1 = h * (28x_1 - x_2 - x_1x_3),$$

$$k_2 = h * (28(x_1 + \frac{k_1}{4}) - (x_2 + \frac{k_1}{4}) - (x_1 + \frac{k_1}{4})(x_3 + \frac{k_1}{4})),$$

$$k_3 = h(28(x_1 + \frac{3k_1}{32} + \frac{9k_2}{32}) - (x_2 + \frac{3k_1}{32} + \frac{9k_2}{32}) - (x_1 + \frac{3k_1}{32} + \frac{9k_2}{32})(x_3 + \frac{3k_1}{32} + \frac{9k_2}{32})),$$

$$k_4 = h(28(x_1 + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197}) - (x_2 + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197})$$

$$-(x_1 + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197})(x_3 + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197})),$$

$$k_5 = h(28(x_1 + \frac{439k_1}{216} - 8k_2 + \frac{3680k_3}{513} - \frac{845k_4}{4104}) - (x_2 + \frac{439k_1}{216} - 8k_2 + \frac{3680k_3}{513} - \frac{845k_4}{4104})$$

$$-(x_1 + \frac{1439k_1}{216} - 8k_2 + \frac{3680k_3}{513} - \frac{845k_4}{4104})(x_3 + \frac{439k_1}{216} - 8k_2 + \frac{3680k_3}{513} - \frac{845k_4}{4104})),$$

$$k6 = h(28(x_1 - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40}) - (x_2 - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40})$$

$$-(x_1 - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40})(x_3 - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40})),$$

$$y_{n+1} = y_n + (\frac{16k_1}{135} + 6656k_3 + \frac{28561k_4}{56430} - \frac{9k_5}{50} + \frac{2k_6}{55}))$$

Discretización de la tercera ecuación:

$$k_1 = h(x_1x_2 - \frac{8}{3}x_3),$$

$$k_2 = h((x_1 + \frac{k_1}{4})(x_2 + \frac{k_1}{4}) - \frac{8}{3}(x_3 + \frac{k_1}{4})),$$

$$k_3 = h((x_1 + \frac{3k_1}{32} + \frac{9k_2}{32})(x_2 + \frac{3k_1}{32} + \frac{9k_2}{32}) - \frac{8}{3}(x_3 + \frac{3k_1}{32} + \frac{9k_2}{32})),$$

$$k_4 = h((x_1 + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197})(x_2 + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197}),$$

$$-\frac{8}{3}(x_3 + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197}))$$

$$k_5 = h((x_1 + \frac{439k_1}{216} - 8k_2 + \frac{3680k_3}{513} - \frac{845k_4}{4104}) * (x_2 + \frac{439k_1}{216} - 8k_2 + \frac{3680k_3}{513} - \frac{845k_4}{4104})$$

$$-\frac{8}{3}(x_3 + \frac{439k_1}{216} - 8k_2 + \frac{3680k_3}{513} - \frac{845k_4}{4104})),$$

$$k_6 = h((x_1 - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40})(x_2 - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40})$$

$$-\frac{8}{3}(x_3 - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40}))$$

$$y_{n+1} = y_n + (\frac{16k_1}{135} + 6656k_3 + \frac{28561k_4}{56430} - \frac{9k_5}{50} + \frac{2k_6}{55})$$

El sistema de ecuaciones que representa al observador (39) se discretiza de manera análoga a la discretización del sistema de Lorenz (38).

5.3.3 Resultados obtenidos mediante la implantación del esquema de cifrado propuesto en esta tesis empleando el sistema de Lorenz.

Para corroborar los resultados obtenidos en la simulación hecha en SIMNOM, se implantó un prototipo (ver [Apéndice B]) en el lenguaje de programación C del esquema de cifrado que se propuso, empleando el sistema de Lorenz discretizado que se presentó en la sección anterior. Las pruebas se realizaron cifrando una imagen (archivo bmp de 24 bits) dos veces: la primera, sumando el estado x_2 para el cifrado, con un escalamiento de 100 elegido arbitrariamente; la segunda, sumando el estado x_3 para el cifrado, con un escalamiento de 200 elegido arbitrariamente. El proceso de descifrado se realizó restando los estados estimados $\widehat{x}_2$(multiplicado por 100) y $\widehat{x}_3$(multiplicado por 200), para el primero y segundo cifrado respectivamente.

Para poder hacer uso de este esquema de cifrado/descifrado es necesario que el emisor y el receptor acuerden los siguientes datos:

1) Los valores de los parámetros asignados a los sistemas de cifrado y descifrado (σ, r y b, para el caso del sistema de Lorenz).

2) El o los estados (x_2 y x_3 para el caso del sistema de Lorenz) y el escalamiento que se emplearán para el cifrado/descifrado.

3) El valor del paso de integración h.

4) El valor arbitrario de una constante $k\prime$, tal que los valores del estado o estados empleados en el cifrado/descifrado, sean tomados cada $t = k\prime h$.

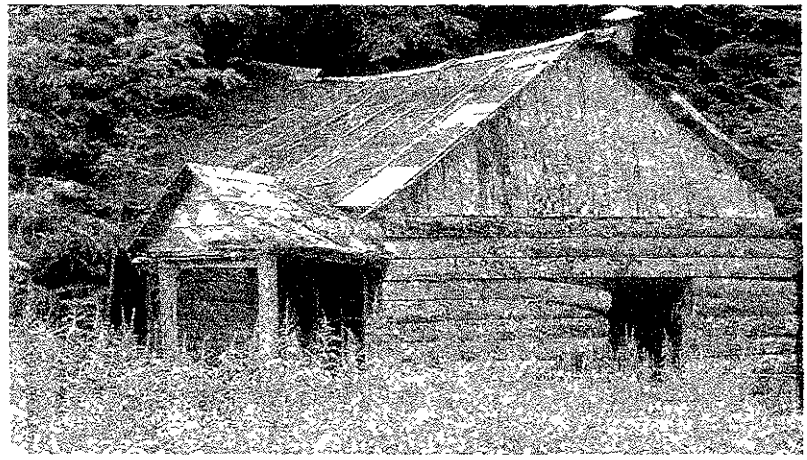

Figura 5.7 Imagen empleada para realizar
las pruebas de cifrado y descifrado

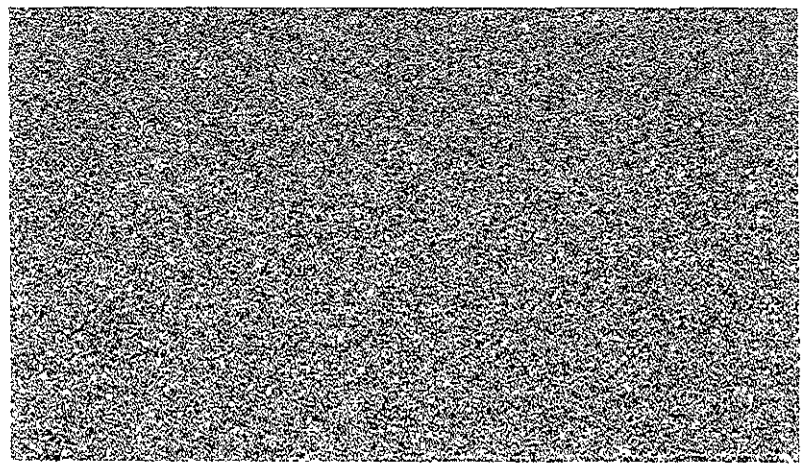

Figura 5.8 Imagen obtenida al aplicar el proceso de cifrado,
empleando x_2

Figura 5.9 Imagen obtenida al aplicar el proceso de descifrado, empleando $\widehat{x}_2$

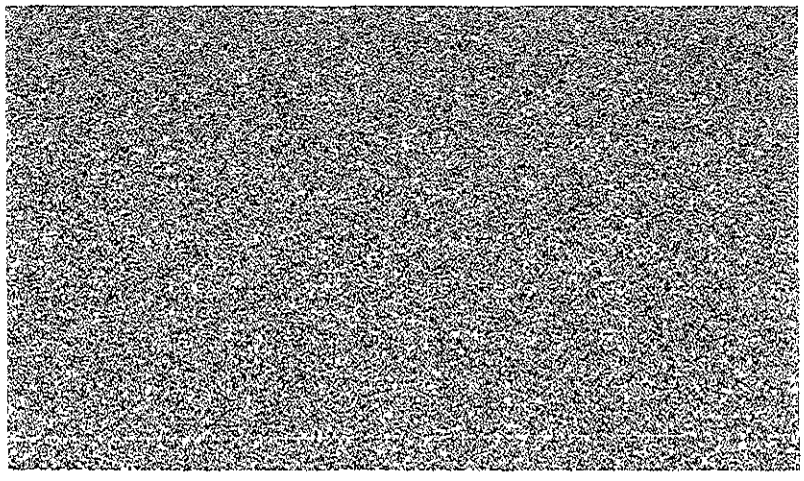

Figura 5.10 Imagen obtenida al aplicar el proceso de cifrado, empleando x_3

Finalmente se muestra un texto correspondiente a un fragmento del código fuente del programa que implanta el prototipo para el cifrado, elegido arbitrariamente y que fue cifrado empleando x_2 y el texto recuperado con el programa que implanta el prototipo para el descifrado.

Texto cifrado:

©-P1
#Ç f7Uû□"DÎ."□¤‰ö
PQ
8[^QZf
##—"##
e
_a
[N_Q^bMP[¶YYYYYYppz□]Z]"ä##
5ZUOU[
OU^OaU`[
OM[`ûü¨pz□]Z]Zī¨□ýXQ
Pd##P[aNXQ#
P[aNXQ#
P[□ý¬æ|pï¨□ý¬æy<]"P[aNXQ
P
P[aý¬æ|pï¨□ý¬æ|pï¨□NXQ#
P[a^"¡ïù□öìNXQ#'—"ï¨□ý¬æpïÔŸxï¨□ý¬æ#
P[aN"¡ïù□öì^"¡#
P[aN¬æ|pï¨□ý¬æy<]Z]ZY##
2U"ù‰¤£%ol¤□öù‰M[`UO¨pppppâèÆµó]ZY□zp5Z
UOUöù‰¤£%ol¤□öùõ^_Q^bñïï¨pèÆµóµó##)ý¬æpïQ##Pöì^"¡ïù□öì^"¡#
P¨□ý¬#…â##)# #…âï¨□ýXQ#ù□öì^?%M_1.□öì^XQpïæ'š##)# #…â##)ý¬æ#
Pöì^"%PD^"TÅ?¡ïùP[□ý¬#…â#Î?z"}…â##)ý¬æ#
□öìÅ?%M_1.^"T^"Q
PæŸš##?z"}D8¦Î)# #¦pP[ì^"%PD$/5",
%PD□öaN¬æí‹%"%"!GB8^?Åâ#û¶üaU□öD""3%26!$/-Dù
#□µó%"!®ÝoµŒŒ‰S}<â#¶?)dûæ%U□…µ×ZTâ•¡%Vùepæ&KU"î¸E#
ï#?èŠE #¡^DOUBŠ 8}&&¨^§ø^/¥ód]$#\g
¥ó,%ñ'µ"|‰UBø¸d¸ôôüR#Ï…Î?#¬Q#ù^/¥Zd—«k-z##eSý
£?^
#Ô

Texto descifrado:

```
/* Inicio circuito caotico */
double dx1(double, double, double, double);
double dx2(double, double, double, double, double, double);
double dx3(double, double, double, double, double);
/* Fin circuito caotico */
/* Inicio circuito observado */
double de1(double, double, double, double, double, double);
double de2(double, double, double, double, double, double, double, double);
double de3(double, double, double, double, double);
/* Fin circuito observador */
/* Errores er1=x1-e1, er2=x2-e2 y er3=x3-e3 */
double error1(double, double);
double error2(double, double);
double error3(double, double);
/* Fin erorres er1, er2 y er3 */
```

6 Conclusiones.

Las conclusiones finales de este trabajo se dividen en dos secciones: en la primera se mencionan los alcances y logros; en la segunda se discuten los posibles trabajos futuros.

6.1 Alcances y Logros

Se instrumentaron en forma numérica los métodos propuestos en el capítulo 4, que se refiere al diseño de observadores asintóticos y su aplicación al mecanismo de cifrado y descifrado (ver [sec. 4.4 y 4.5]).

Se realizaron algunas simulaciones numéricas en las que se mostró en forma gráfica el comportamiento del vector de estado (producido por el sistema cifrador) y su respectivo vector de estimación de estado (producido por el sistema descifrador). Esto es, en función de la o las variables de salida (sistema cifrador) se pueden reconstruir los estados restantes casi con una precisión del cien por ciento. Este hecho es de gran importancia en el mecanismo de cifrado/descifrado, ya que la información cifrada con el sistema, puede recuperarse con el observador pasivo de sus estados.(ver [sec. 5.2]).

Se implantó un prototipo de cifrado/descifrado propuesto en las sección 4.5 en el lenguaje de programación C, y se utilizó para cifrar y descifrar una imagen (mapa de bits) y un achivo de texto. En ambos casos fué posible recuperar la información (ver [capítulo 5]). De este experimento podemos comentar lo siguiente:

1) Para que el esquema funcione correctamente, los parámetros $\{\sigma, r, b\}$ deben de ser iguales en el sistema cifrador y en el sistema descifrador.

2) No es necesario que las condiciones iniciales de los estados del sistema sean idénticas a las condiciones iniciales de los estados del observador, dado que éste es capaz de recuperar la información total a partir del momento en que el error $x_n - \widehat{x}_n$ se hace cero ; aunque es deseable que la diferencia entre los valores de los estados iniciales del sistema y de su observador sea pequeña o igual a cero.

3) El valor del paso de integración h, debe ser el mismo en ambos sistemas, de lo contrario no es posible estimar los estados del sistema original y por ende no es posible recuperar la información.

4) Se puede definir arbitrariamente una constante $k\prime$, tal que los valores del estado o estados empleados en el cifrado, sean tomados cada $k\prime h$.

5) La seguridad del esquema depende de que los parámetros datos $\{\sigma, r, b, \lambda\}$, h y k/h permanezcan en secreto.

Finalmente, se comenta que se eligió el sistema de Lorenz para el desarrollo de los experimentos, primero porque es completamente observable, cuando se toma como salida del sistema el estado $y = x_1$; segundo, es un sistema que no necesita incluir la ganancia del observador; y tercero, porque los rangos de los parámetros $\{\sigma, r, b\}$ para los cuales el sistema exhibe un comportamiento caótico, es amplio.

6.2 Trabajo Futuro

El programa que se implantó como prototipo para probar los resultados que se obtuvieron durante el desarrollo de esta tesis, puede ser rediseñado y desarrollar una aplicación completa que pueda ser empleada en situaciones reales.

Con respecto al enfoque que se utilizó para el diseño de sistemas observadores asintóticos, cabe mencionar que no es el único, pero fue elegido para esta tesis por su simplicidad y elegancia matemática. Un estudio más a fondo de la teoría de observadores quizá permita diseñar observadores asintóticos más eficientes que los aquí se propusieron y entonces optimizar el tiempo de cifrado/descifrado.

Esta tesis únicamente emplea sistemas caóticos continuos, porque el espacio de estado que contiene a las condiciones iniciales es mas amplio que en los sistemas discretos, además porque no dependen de las condiciones iniciales y ofrecen rangos más amplios para los valores de los parámetros en los cuales el comportamiento de éstos es caótico. Quizá los resultados obtenidos en este trabajo, pudieran ser referencia para un trabajo posterior que involucre técnicas de diseño de observador distintas a las aquí empleada y aplicarse a sistemas caóticos discretos, que en cierto sentido son más adecuados para el manejo de información representada de forma digital.

En lo referente a la Teoría del Caos, en este trabajo únicamente se hace mención de algunos resultados obtenidos previamente y que se requerían como sustento.

Bibliografía

[1] Shannon C. E., Collected Papers: Claude Elmwood Shannon, N.J.A.,Sloane and A. D. Wyner, eds., New York: IEEE Press, (1993).
[2] Gerald C. F., Wheatley P. O., Applied Numerical Analisys, Edit. Addison Wesley, fifth edition (1994).
[3] P. Pfleeger C., Security in computing, Edit. Prentice-Hall, (1996)..
[4] Schneier B., Applied Cryptography, Edit. John Wiley & sons, (1996).
[5] Wayner P., Disappearing Cryptography, Edit. AP Professional, 1996.
[6] Russel D., Gangemi Sr., Computer Security Basics, Edit. O'Reilly & Assoc., (1991).
[7] DeMillo R, et. al., Applied Cryptology, cryptographic protocols, and computer security models, American Mathematical Society,Proceedings of Symposia in Applied Mathematics, vol. 29, (1983).
[8] Acheson D., From Calculus to Chaos, an introduction to dynamics, Edit. Oxford University Press, (1997).
[9] Holden A., Chaos, Princeton University Press, (1986).
[10] T. L. Carroll, and L. Pecora, Synchronizing chaotic circuits IEEE Transactions on Circuits and Systems, vol. 38, (4) (1991), pp. 453-456.
[11] G. Chen, Control and Synchronization of Chaos, a Bibliography, Department of Electrical Engineering. University of Houston, Houston TX (also available via ftp at ftp:uhoop.uh.edu/pub/chaos.tex), 1997.
[12] K. M. Cuomo, A. V. Oppenheim ans S. H. Strogatz, Synchronization of Lorenz-Based Chaotic Circuits with Applications to Communications, IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 40, October (1993), pp. 626-633.
[13] A. Fradkov and A. Yu. Markov, Adaptive Synchronization of Chaotic Systems Based on Speed Gradient Method and Passification, IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, vol. 44, (10), (1997), pp. 905-917.
[14] H. J. C. Huijberts, H. Nijmeijer and R. M. A. Willems, A control perspective on communications using chaotic systems, Proceedings 37th IEEE Conference on Decision and Control, Tampa, Florida, December 16-18 (1998), pp. 1957-1962.
[15] E. N. Lorenz, Deterministic nonperiodic flow, Journal of Atmospheric Science, vol. 20, (1963), pp. 130-141.
[16] F. Mitschke and N. Flüggen, Chaotic behavior of a hybrid optical bistable system without a time delay, Applied Physics B, vol. 35, (1984), pp. 59-64.
[17] H. Nijmeijer and M. Y. Mareels, An Observer Looks at Synchronization, IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, vol. 44 (10), October (1997).
[18] E. Ott, T. Sauer and J. A. Yorke, (Eds.) Coping with Chaos: Analysis of Chaotic Data and the Exploitation of Chaotic Systems, New York: Wiley-Interscience, (1994).
[19] L. M. Pecora and T. L. Carroll, Driving systems with chaotic signals, Physical Review A, vol. 44 (4), pp. 2374-2383.

[20] Special Issue Systems and Control Letters, vol. 31, (1997).
[21] Special Issue, Chaos synchronization and control: theory and applications, IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, vol. 40, (1993).
[22] Special Issue, Chaos synchronization and control: theory and applications, IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, vol. 44, (1997).
[23] C. W. Wu and L. Chua, A simple way to synchronize chaotic systems with applications to secure communication systems Int. J. Biffurcation and Chaos, vol. 3 (6), (1993), pp. 1619-1627.

Apéndice A. Estabilidad de Lyapunov y observadores.

En este apartado se da una introducción de los fundamentos teóricos de los conceptos de estabilidad de Liapunov y de observabilidad. El primero es una herramienta que permite diseñar y encontrar el vector de ganancia del observador para poder garantizar que el error entre el estado del sistema emisor y el estado estimado por el observador converge exponencialmente a cero. El segundo es utilizado en este trabajo para diseñar e implementar un observador de estados, que será empleado como mecanismo de cifrado y descifrado en el esquema de encriptado que se propone.

Estabilidad de Lyapunov:

Estabilidad:

La teoría de la estabilidad juega un papel central en el estudio de los sistemas dinámicos y las ecuaciones diferenciales. En general, se dice que un sistema dinámico es estable si la solución está acotada. Existen diferentes definiciones de estabilidad. Entre las más usadas están: el concepto de estabilidad en el sentido de Lyapunov; estabilidad entrada/salida; y estabilidad en promedio. Para esta tesis el concepto de estabilidad que se usará será el de estabilidad en el sentido de Lyapunov. Existe toda una teoría alrededor del estudio del análisis de estabilidad, que fue iniciada por Vladimir Lyapunov, a finales del siglo XIX , cuya principal característica es el poder garantizar que las soluciones de un sistema de ecuaciones diferenciales lineales y no lineales sean estables sin necesidad de encontrar su solución. Los teoremas de estabilidad de Lyapunov dan condiciones suficientes para garantizar la estabilidad uniforme, la estabilidad asintótica y la estabilidad exponencial, *i.e.*, la primera garantiza que la solución está uniformemente acotada; la segunda garantiza que la solución converge a cero; la tercera, garantiza que la solución decrece exponencialmente a cero.

La estabilidad en el sentido de Lyapunov se apoya básicamente en el concepto de punto de equilibrio, que está relacionado con el punto en el que las derivadas se hacen cero. A continuación se presenta la definición formal de punto de equilibrio, estabilidad e inestabilidad.

Definición 1: Punto de equilibrio: Considere el siguiente sistema no lineal:

$$x(t) = f(x(t)), \tag{40}$$

donde $f(x)$ es una función vectorial suave sobre $\Re^n$, y supóngase que existe $\bar{x} \in \Re^n$ tal que $f(\bar{x}) = 0$. Entonces $\bar{x}$ es un punto de equilibrio de (40) si $x(t_0) = \bar{x}$ entonces $x(t) = \bar{x}$ para toda $t > t_0$.

Como ejemplo, considere el siguiente sistema lineal invariante en el tiempo (autónomo):

$$\dot{x}(t) = Ax(t) \tag{41}$$

el punto de equilibrio satisface la condición;

$$A\bar{x} = 0,$$

que siempre tiene al origen como punto de equilibrio. Si la matriz A es no singular, entonces el origen es un punto de equilibrio único y si la matriz A es singular (no invertible), entonces puede haber otros puntos de equilibrio.

Definición 2: Un punto de equilibrio $\bar{x} = 0$ para el sistema autónomo (40) es,

i) estable, si para cada $\varepsilon > 0$ existe $\delta = \delta(\varepsilon) > 0$ tal que:

$$\|x(0)\| < \delta \Longrightarrow \|x(t)\| < \varepsilon, \text{ para toda } t > 0.$$

ii) asintóticamente estable, si es estable y existe δ tal que:

$$\|x(0)\| < \delta \Longrightarrow \lim_{t \to \infty} x(t) = 0,$$

iii) exponencialmente estable, si existen constantes r, a, $b > 0$ tal que:

$$\|x(t + t_0)\| \leq a\|x(t_0)\| \exp(-bt) \nabla t, t_0 \geq 0, \text{ y } \|x(t_0)\| \leq r,$$

iv) inestable, si no es estable.

A la evolución de los estados de un sistema, se les denomina como las trayectorias de los estados. La estabilidad asintótica requiere que el estado final de la

trayectoria del sistema sea $\bar{x}$, cuando es iniciada cercana a $\bar{x}$. Para demostrar que el origen es estable, se toma un valor de ε cuidadosamente designado, tal que cualquier punto iniciado en una vecindad del origen nunca saldrá de la ε vecindad.

Teorema 1: El punto de equilibrio $\bar{x} = 0$ de (41) es estable si y sólo si todos los eigenvalores de A tiene partes reales no positivas. El punto de equilibrio $\bar{x} = 0$ de (41) es (globalmente) exponencialmente estable si y sólo si todos los eigenvalores de A tiene partes reales negativas [Rugh, 1993].

Funciones de Lyapunov:

La idea básica del método de Lyapunov es buscar una función que cumpla determinadas condiciones bajo las cuales el sistema sea estable, y una de las condiciones que se requieren es que la función decremente continuamente a un mínimo mientras el sistema evoluciona. Una función con estas características se dice que es función de Liapunov.

Definición 3: Una función $V : D \to \Re$, definida sobre una región $D \subset \Re^n$ del espacio de estado y conteniendo a $\bar{x}$, es una función localmente definida positiva si cumple con los siguientes requerimientos:

- V es continuamente diferenciable,

- $V(0) = 0$ y $V(x) > 0$ en $D - \{0\}$.

Será definida positiva si satisface las condiciones anteriores para todo $x \in \Re^n$.

A continuación se presenta el resultado de Lyapunov que permite analizar la estabilidad en los sistemas dinámicos:

Teorema 2: Sea $\bar{x} = 0$ un punto de equilibrio para (40), $D \subset \Re^n$ y $V : D \to \Re$ una función localmente definida positiva y $\dot{V}(x) \leq 0$ en D entonces $\bar{x} = 0$ es estable, más aún, si $\dot{V}(x) < 0$ en $D - \{0\}$, entonces $\bar{x} = 0$ es asintóticamente estable.

Teorema 3: Sea $\bar{x} = 0$ un punto de equilibrio para (40), $V : \Re^n \to \Re$ una función definida positiva y $V(x) < 0$ para todo $x \in \Re^n$ entonces $\bar{x} = 0$ es globalmente(i.e. $D = \Re^n$) asintóticamente estable.

Este resultado es un caso especial de la estabilidad asintótica, se satisface cuando un punto de equilibrio es asintóticamente estable y además si las condiciones

iniciales del sistema se especifican en cualquier punto del espacio de estado, los estados del sistema tienden hacia el punto de equilibrio.

Observabilidad:

Un sistema puede ser observable si a partir de las salidas y entradas del sistema es posible obtener las variables de estado (reconstrucción). A esta característica de los sistemas se le denonima observabilidad y se define como:

Definición 4: Considere el sistema:

$$\begin{array}{c} \dot{x}(t) = f(x) + \sum u_i g_i(x), \\ y(t) = h(x) \end{array} \tag{42}$$

los estados x_0 y x_1,se dicen ser distinguibles, si existe una función de entrada $u(\cdot)$ tal que

$$y(\cdot, x_0, u) \neq y(\cdot, x_1, u),$$

donde $y(\cdot, x_i, u)$, $i = 1, 2$ es la función de salida del sistema (42) correspondiente a la función de entrada $u(\cdot)$ y la condición inicial $x(0) = x_i$. El sistema se dice ser observable localmente en $x(0) \in X$ si existe una vecindad N de x_0 tal que para toda $x \in N$ distintas de x_0 son distinguibles de x_0. Finalmente el sistema se dice observable localmente si éste es observable localmente en cada uno de los $x_0 \in N$.

La observabilidad es una condición necesaria para los observadores. Dado un sistema donde se hace la estimación de las variables de estado, es indispensable saber cuales son las condiciones para la existencia de la observabilidad en estos sistemas.

La observabilidad en los sistemas lineales no depende de las entradas y es una característica de estos sistemas, mientras que en los sistemas no lineales sí depende de las entradas.

Observadores:

Definición 5: Se llama observador asintótico (o reconstructor de estado) de un sistema dinámico

$$\begin{aligned} \dot{x} &= f(x,u), \\ y &= h(x) \end{aligned} \tag{43}$$

a un sistema dinámico donde las entradas se constituyen de vectores de entrada y salida de un sistema a observar, donde el vector de salida que se denota $\widehat{x}$ es el estimado de

$$\begin{aligned} \dot{z} &= \widehat{f}(z(t), y, u), \\ \widehat{x}(t) &= \widehat{n}(z(t), y, u), \end{aligned}$$

tal que

i) $\|e(t)\| = \|\widehat{x}(t) - x(t)\| \to 0$ cuando $t \to \infty$,

ii) si, en $t = t_0$ se tiene $\widehat{x}(t_0)$, entonces para $t \geq t_0$, se tiene $\widehat{x}(t) = x(t)$.

Se puede esquematizar un conjunto sistema-observador como en la siguiente figura:

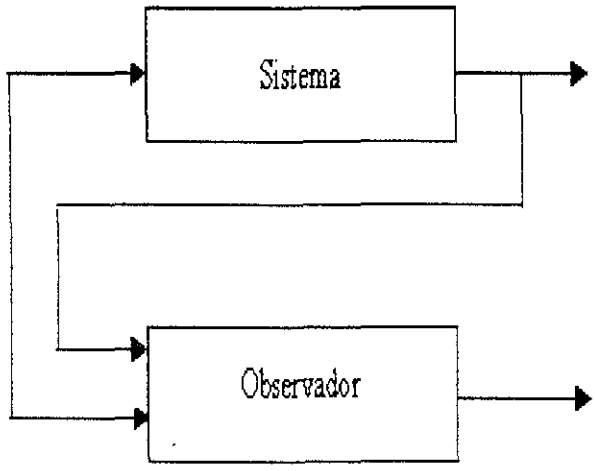

Figura A.1 Esquema Sistema-Observador

Un observador que satisface los 2 puntos de la definición anterior se dice que posee la propiedad (local) de reconstructividad y el sistema se llama observador (local) asintótico.

Observador de Luenberger:

Sea el siguiente sistema lineal invariante en el tiempo observable:

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{44}$$

y cuyo observador trivial es:

$$\dot{z}(t) = Az(t) + Bu(t),$$

donde las entradas $u(t)$ del sistema (38), son los controles que se suministran al sistema y si $z(0) = x(0)$ los dos sistemas serán equivalentes. Dado que:

$$\left[\dot{z}(t) - \dot{x}(t)\right] = A\left[z(t) - x(t)\right],$$

el error en la estimación será cero si alguna de las condiciones sobre la matriz A se cumplen *i.e.*, el sistema será estable si A tiene todos sus eigenvalores negativos. Una desventaja de este observador es que no hay forma de poder actuar en el error y de poder cambiar la rapidez de convergencia.
Un observador de la forma:

$$\dot{z}(t) = Az(t) - K\left[Cz(t) - y(t)\right] + Bu(t),$$

es un sistema $n-$dimensional con vector de estado $z(t)$. Para el sistema (38) este observador es una generalización del observador trivial. Si $z(0) \neq x(0)$ entonces el vector error $e(t) = z(t) - x(t)$ es gobernado por el sistema homogéneo:

$$\dot{e}(t) = \left[A - KC\right] e(t)$$

Para garantizar que el error converge a cero, se escoge el vector K tal que los eigenvalores de la matriz $A - KC$ esten en el semiplano izquierdo del plano complejo, *i.e.* $[\lambda(A - KC)] < 0$.

Apéndice B. Código de los prototipos que implantan el esquema de cifrado que se propone en esta tesis, empleando el sistema de Lorenz y el observador pasivo de sus estados.

B.1 Cifrador.

```
#include<stdio.h>
#include<math.h>
#include<conio.h>


/* Definicion de las funciones para el circuito caotico de Lorenz */
/* y su observador                                                 */

        /* Inicio circuito caotico */

double dx1(double, double, double, double);
double dx2(double, double, double, double, double, double);
double dx3(double, double, double, double, double);

        /* Fin circuito caotico     */

        /* Inicio circuito observado */

double de1(double, double, double, double, double, double);
double de2(double, double, double, double, double, double, double, double);
double de3(double, double, double, double, double);

        /* Fin circuito observador   */

        /* Errores er1=x1-e1, er2=x2-e2 y er3=x3-e3  */

double error1(double, double);
double error2(double, double);
double error3(double, double);

        /* Fin errores er1, er2 y er3   */
```

Cifrador (continuación)

```
main(int argc, char *argv[])
{

        FILE *infp, *outfp;
        char g;
        int c2, t=0,w=0;


/* Condiciones iniciales */

        double x10=3.0, x20=2.0,  x30=3.0;
        double e10=3.0, e20=2.0, e30=3.0;
        double a=16.0,  b=4.0, k=8.0, c=61.0,  h=0.0001;

/*                      */

        double x1n, x2n, x3n;
        double e1n, e2n, e3n;
        double er1n, er2n, er3n;

        x1n=x10; x2n=x20; x3n=x30;
        e1n=e10; e2n=e20; e3n=e30;

/* Manejo de archivos */

        if(argc==1)
            printf("Error\n");
        else
            if((infp=fopen(*++argv,"rb"))==NULL)
                    printf("Imposible abrir %s\n",*argv);
        else
            if((outfp=fopen(*++argv,"wb"))==NULL)
                    printf("Imposible abrir %s\n",*argv);
```

Cifrador (continuación)

```
    for(w=1;w<=80;w++)
      putc(getc(infp), outfp);

     while((c2=getc(infp))!=EOF)
     {
      for(t=1;t<=100;t++)
        {
        x1n=dx1(x10,x20,h,a);
        x2n=dx2(x10,x20,x30,h,a,c);
        x3n=dx3(x10,x20,x30,h,b);

        e1n=de1(x10,h,e10,e20,a,k);
        e2n=de2(x10,x20,e30,h,e10,e20,a,c);
        e3n=de3(x10,h,e20,e30,b);

        er1n=error1(x1n,e1n);
        er2n=error1(x2n,e2n);
        er3n=error1(x3n,e3n);

/*      printf("er1n=%f\ter2n=%f\ter3n=%f\n",er1n,er2n,er3n);
*/
        x10=x1n; x20=x2n; x30=x3n;
        e10=e1n; e20=e2n; e30=e3n;
        }

        putc(c2+ceil(x2n)*100, outfp);
     }
        fclose(infp);
        fclose(outfp);
}
```

Cifrador (continuación)

```
double dx1(double x1, double x2, double h, double a)
{
   double k1,k2,k3,k4,k5,k6;

   k1=h*a*(x2-x1);
   k2=h*a*((x2+k1/4.0)-(x1+k1/4.0));
   k3=h*a*((x2+(3.0*k1+9.0*k2)/32.0)-(x1+(3.0*k1+9.0*k2)/32.0));
   k4=h*a*((x2+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
         -(x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0));
   k5=h*a*((x2+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
         -(x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0));

   k6=h*a*((x2+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
         -(x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0));

   return (x1+(6688.0*k1+28561.0*k4+2052.0*k6)/56430.0+(13312.0*k3-4617.0*k5)/25650.0);
}

double dx2(double x1, double x2, double x3, double h, double a, double c)
{
   double k1,k2,k3,k4,k5,k6;

   k1=h*((c-a)*(x1)-x1*x3-x2);
   k2=h*((c-a)*(x1+k1/4.0)-(x1+k1/4.0)*(x3+k1/4.0)-(x2+k1/4.0));
   k3=h*((c-a)*(x1+(3.0*k1+9.0*k2)/32.0)
               -(x1+(3.0*k1+9.0*k2)/32.0)
               *(x3+(3.0*k1+9.0*k2)/32.0)
            -(x2+(3.0*k1+9.0*k2)/32.0));
   k4=h*((c-a)*(x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
               -(x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
               *(x3+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
            -(x2+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0));
```

Cifrador (continuación)

```
    k5=h*((c-a)*(x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
                -(x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
                *(x3+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
                -(x2+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0));
    k6=h*((c-a)*(x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
                -(x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
                *(x3+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
            -(x2+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0));

    return (x2+(6688.0*k1+28561.0*k4+2052.0*k6)/56430.0+(13312.0*k3-4617.0*k5)/25650.0);
}


double dx3(double x1, double x2, double x3, double h, double b)
{
    double k1,k2,k3,k4,k5,k6;

    k1=h*(x1*x2-b*x3);
    k2=h*((x1+k1/4.0)*(x2+k1/4.0)-b*(x3+k1/4.0));
    k3=h*((x1+(3.0*k1+9.0*k2)/32.0)
          *(x2+(3.0*k1+9.0*k2)/32.0)
        -b*(x3+(3.0*k1+9.0*k2)/32.0));
    k4=h*((x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
          *(x2+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
        -b*(x3+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0));
    k5=h*((x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
          *(x2+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
        -b*(x3+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0));
    k6=h*((x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265 0*k4-5643.0*k5)/20520.0)
          *(x2+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
        -b*(x3+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0));

    return (x3+(6688.0*k1+28561.0*k4+2052.0*k6)/56430.0+(13312.0*k3-4617.0*k5)/25650.0);
}
```

Cifrador (continuación)

```
double de1(double x1, double h, double e1, double e2, double a, double k)
{
   double k1,k2,k3,k4,k5,k6;

   k1=h*(a*(e2-e1)-k*(e1-x1));
   k2=h*(a*((e2+k1/4.0)
            -(e1+k1/4.0))
          -k*((e1+k1/4.0)
            -(x1+k1/4.0)));
   k3=h*(a*((e2+(3.0*k1+9.0*k2)/32.0)
            -(e1+(3.0*k1+9.0*k2)/32.0))
          -k*((e1+(3.0*k1+9.0*k2)/32.0)
            -(x1+(3.0*k1+9.0*k2)/32.0)));
   k4=h*(a*((e2+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
            -(e1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0))
          -k*((e1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
            -(x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)));
   k5=h*(a*((e2+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
            -(e1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0))
          -k*((e1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
            -(x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)));
   k6=h*(a*((e2+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
            -(e1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0))
          -k*((e1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
            -(x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)));

   return (e1+(6688.0*k1+28561.0*k4+2052.0*k6)/56430.0+(13312.0*k3-4617.0*k5)/25650.0);
}
```

Cifrador (continuación)

```
double de2(double x1, double x2, double e3, double h, double e1, double e2, double a, double c)
{
    double k1,k2,k3,k4,k5,k6;

    k1=h*((c-a)*x1-x1*e3-e2);
    k2=h*((c-a)*(x1+k1/4.0)-(x1+k1/4.0)*(e3+k1/4.0)-(e2+k1/4.0));
    k3=h*((c-a)*(x1+(3.0*k1+9.0*k2)/32.0)
                -(x1+(3.0*k1+9.0*k2)/32.0)
                *(e3+(3.0*k1+9.0*k2)/32.0)
            -(e2+(3.0*k1+9.0*k2)/32.0));
    k4=h*((c-a)*(x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
                -(x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
                *(e3+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
            -(e2+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0));
    k5=h*((c-a)*(x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
                -(x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
                *(e3+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
            -(e2+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0));
    k6=h*((c-a)*(x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
                -(x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
                *(e3+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
            -(e2+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0));

    return (e2+(6688.0*k1+28561.0*k4+2052.0*k6)/56430.0+(13312.0*k3-4617.0*k5)/25650.0);
}

double de3(double x1, double h, double e2, double e3, double b)
{
    double k1,k2,k3,k4,k5,k6;

    k1=h*(x1*e2-b*e3);
    k2=h*((x1+k1/4.0)*(e2+k1/4.0)-b*(e3+k1/4.0));
    k3=h*((x1+(3.0*k1+9.0*k2)/32.0)
          *(e2+(3.0*k1+9 0*k2)/32.0)
        -b*(e3+(3.0*k1+9.0*k2)/32.0));
```

Cifrador (continuación)

```
    k4=h*((x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
          *(e2+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
         -b*(e3+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0));
    k5=h*((x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
          *(e2+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
         -b*(e3+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0));
    k6=h*((x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
          *(e2+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
         -b*(e3+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0));

    return (e3+(6688.0*k1+28561.0*k4+2052.0*k6)/56430.0+(13312.0*k3-4617.0*k5)/25650.0);
}

double error1(double x1, double e1)
{
    return x1-e1;
}

double error2(double x2, double e2)
{
    return x2-e2;
}

double error3(double x3, double e3)
{
    return x3-e3;
}
```

B.2 Descifrador.

```
#include<stdio.h>
#include<math.h>
#include<conio.h>


/* Definicion de las funciones para el circuito caotico de Lorenz */
/* y su observador                                                 */

        /* Inicio circuito caotico */

double dx1(double, double, double, double);
double dx2(double, double, double, double, double, double);
double dx3(double, double, double, double, double);

        /* Fin circuito caotico     */

        /* Inicio circuito observado */

double de1(double, double, double, double, double, double);
double de2(double, double, double, double, double, double, double, double);
double de3(double, double, double, double, double);

        /* Fin circuito observador   */

        /* Errores er1=x1-e1, er2=x2-e2 y er3=x3-e3  */

double error1(double, double);
double error2(double, double);
double error3(double, double);

        /* Fin erorres er1, er2 y er3   */
```

Descifrador (continuación)

```
main(int argc, char *argv[])
{

        FILE *infp, *outfp;
        char g;
        int c2, t=0,w=0;


/* Condiciones iniciales */

        double x10=3.0, x20=2.0,  x30=3.0;
        double e10=3.0, e20=2.0, e30=3.0;
        double a=16.0,  b=4.0, k=8.0, c=61.0,  h=0.0001;

/*                              */

        double x1n, x2n, x3n;
        double e1n, e2n, e3n;
        double er1n, er2n, er3n;

        x1n=x10; x2n=x20; x3n=x30;
        e1n=e10; e2n=e20; e3n=e30;

/* Manejo de archivos */

        if(argc==1)
            printf("Error\n");
        else
            if((infp=fopen(*++argv,"rb"))==NULL)
                  printf("Imposible abrir %s\n",*argv);
        else
            if((outfp=fopen(*++argv,"wb"))==NULL)
                  printf("Imposible abrir %s\n",*argv);
```

Descifrador (continuación)

```
for(w=1;w<=80;w++)
 putc(getc(infp), outfp);

while((c2=getc(infp))!=EOF)
{
 for(t=1;t<=100;t++)
   {
    x1n=dx1(x10,x20,h,a);
    x2n=dx2(x10,x20,x30,h,a,c);
    x3n=dx3(x10,x20,x30,h,b);

    e1n=de1(x10,h,e10,e20,a,k);
    e2n=de2(x10,x20,e30,h,e10,e20,a,c);
    e3n=de3(x10,h,e20,e30,b);

    er1n=error1(x1n,e1n);
    er2n=error1(x2n,e2n);
    er3n=error1(x3n,e3n);

/*  printf("er1n=%f\ter2n=%f\ter3n=%f\n",er1n,er2n,er3n);
*/
    x10=x1n; x20=x2n; x30=x3n;
    e10=e1n; e20=e2n; e30=e3n;
    }

    putc(c2-ceil(e2n)*100, outfp);
}
    fclose(infp);
    fclose(outfp);
}
```

Descifrador (continuación)

```
double dx1(double x1, double x2, double h, double a)
{
    double k1,k2,k3,k4,k5,k6;

    k1=h*a*(x2-x1);
    k2=h*a*((x2+k1/4.0)-(x1+k1/4.0));
    k3=h*a*((x2+(3.0*k1+9.0*k2)/32.0)-(x1+(3.0*k1+9.0*k2)/32.0));
    k4=h*a*((x2+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
            -(x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0));
    k5=h*a*((x2+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
            -(x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0));

    k6=h*a*((x2+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
            -(x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0));

    return (x1+(6688.0*k1+28561.0*k4+2052.0*k6)/56430.0+(13312.0*k3-4617.0*k5)/25650.0);
}

double dx2(double x1, double x2, double x3, double h, double a, double c)
{
    double k1,k2,k3,k4,k5,k6;

    k1=h*((c-a)*(x1)-x1*x3-x2);
    k2=h*((c-a)*(x1+k1/4.0)-(x1+k1/4.0)*(x3+k1/4.0)-(x2+k1/4.0));
    k3=h*((c-a)*(x1+(3.0*k1+9.0*k2)/32.0)
                -(x1+(3.0*k1+9.0*k2)/32.0)
                *(x3+(3.0*k1+9.0*k2)/32.0)
            -(x2+(3.0*k1+9.0*k2)/32.0));
    k4=h*((c-a)*(x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
                -(x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
                *(x3+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
            -(x2+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0));
```

Descifrador (continuación)

```
   k5=h*((c-a)*(x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
                -(x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
                *(x3+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
                -(x2+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0));
   k6=h*((c-a)*(x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
                -(x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
                *(x3+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
           -(x2+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0));

   return (x2+(6688.0*k1+28561.0*k4+2052.0*k6)/56430.0+(13312.0*k3-4617.0*k5)/25650.0);
}


double dx3(double x1, double x2, double x3, double h, double b)
{
   double k1,k2,k3,k4,k5,k6;

   k1=h*(x1*x2-b*x3);
   k2=h*((x1+k1/4.0)*(x2+k1/4.0)-b*(x3+k1/4.0));
   k3=h*((x1+(3.0*k1+9.0*k2)/32.0)
        *(x2+(3.0*k1+9.0*k2)/32.0)
       -b*(x3+(3.0*k1+9.0*k2)/32.0));
   k4=h*((x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
        *(x2+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
       -b*(x3+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0));
   k5=h*((x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
        *(x2+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
       -b*(x3+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0));
   k6=h*((x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
        *(x2+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
       -b*(x3+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0));

   return (x3+(6688.0*k1+28561.0*k4+2052.0*k6)/56430.0+(13312.0*k3-4617.0*k5)/25650.0);
}
```

Descifrador (continuación)

```
double de1(double x1, double h, double e1, double e2, double a, double k)
{
   double k1,k2,k3,k4,k5,k6;

   k1=h*(a*(e2-e1)-k*(e1-x1));
   k2=h*(a*((e2+k1/4.0)
            -(e1+k1/4.0))
          -k*((e1+k1/4.0)
            -(x1+k1/4.0)));
   k3=h*(a*((e2+(3.0*k1+9.0*k2)/32.0)
            -(e1+(3.0*k1+9.0*k2)/32.0))
          -k*((e1+(3.0*k1+9.0*k2)/32.0)
            -(x1+(3.0*k1+9.0*k2)/32.0)));
   k4=h*(a*((e2+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
            -(e1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0))
          -k*((e1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
            -(x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)));
   k5=h*(a*((e2+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
            -(e1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0))
          -k*((e1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
            -(x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)));
   k6=h*(a*((e2+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
            -(e1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0))
          -k*((e1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
            -(x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)));

   return (e1+(6688.0*k1+28561.0*k4+2052.0*k6)/56430.0+(13312.0*k3-4617.0*k5)/25650.0),
}
```

Descifrador (continuación)

```
double de2(double x1, double x2, double e3, double h, double e1, double e2, double a, double c)
{
   double k1,k2,k3,k4,k5,k6;

   k1=h*((c-a)*x1-x1*e3-e2);
   k2=h*((c-a)*(x1+k1/4.0)-(x1+k1/4.0)*(e3+k1/4.0)-(e2+k1/4.0));
   k3=h*((c-a)*(x1+(3.0*k1+9.0*k2)/32.0)
              -(x1+(3.0*k1+9.0*k2)/32.0)
              *(e3+(3.0*k1+9.0*k2)/32.0)
         -(e2+(3.0*k1+9.0*k2)/32.0));
   k4=h*((c-a)*(x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
              -(x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
              *(e3+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
         -(e2+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0));
   k5=h*((c-a)*(x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
              -(x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
              *(e3+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
         -(e2+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0));
   k6=h*((c-a)*(x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
              -(x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
              *(e3+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
         -(e2+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0));

   return (e2+(6688.0*k1+28561.0*k4+2052.0*k6)/56430.0+(13312.0*k3-4617.0*k5)/25650.0);
}

double de3(double x1, double h, double e2, double e3, double b)
{
   double k1,k2,k3,k4,k5,k6;

   k1=h*(x1*e2-b*e3);
   k2=h*((x1+k1/4.0)*(e2+k1/4.0)-b*(e3+k1/4.0));
   k3=h*((x1+(3.0*k1+9.0*k2)/32.0)
        *(e2+(3.0*k1+9.0*k2)/32.0)
      -b*(e3+(3 0*k1+9.0*k2)/32.0));
```

Descifrador (continuación)

```
    k4=h*((x1+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
          *(e2+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0)
         -b*(e3+(1932.0*k1-7200.0*k2+7296.0*k3)/2197.0));
    k5=h*((x1+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
          *(e2+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0)
         -b*(e3+(8341.0*k1-32832.0*k2+29440.0*k3-845.0*k4)/4104.0));
    k6=h*((x1+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
          *(e2+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0)
         -b*(e3+(-6080.0*k1+41040.0*k2-28352.0*k3+9265.0*k4-5643.0*k5)/20520.0));

    return (e3+(6688.0*k1+28561.0*k4+2052.0*k6)/56430.0+(13312.0*k3-4617.0*k5)/25650.0);
}

double error1(double x1, double e1)
{
    return x1-e1;
}

double error2(double x2, double e2)
{
    return x2-e2;
}

double error3(double x3, double e3)
{
    return x3-e3;
}
```