

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL
INSTITUTO POLITÉCNICO NACIONAL
DEPARTAMENTO DE CONTROL AUTOMÁTICO

**Estabilidad de entrada-estado (ISS) para
identificación con redes neuronales dinámicas**

TESIS QUE PRESENTA EL:
Ing. Alejandro Cruz Sandoval

PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS

EN LA ESPECIALIDAD DE
CONTROL AUTOMÁTICO

DIRECTOR DE TESIS:
Dr. Wen Yu Liu

México, D.F., junio del 2003.

Índice general

1. Introducción	1
1.1. Motivación del tema de tesis	2
1.2. Planteamiento del problema	4
1.3. Artículos publicados	5
2. Redes Neuronales e Identificación	7
2.1. Unidades de proceso: La neurona artificial	8
2.1.1. Estado de activación de cada neurona	11
2.1.2. Función de salida o transferencia	12
2.2. Topología de redes neuronales	14
2.2.1. Arquitectura de redes neuronales recurrentes	15
2.2.2. Red neuronal recurrente en espacio de estado	25
2.2.3. Red neuronal recurrente entrenable	26
2.3. Redes neuronales para Identificación	29
2.3.1. Identificación de sistemas	31
2.3.2. Identificación de sistemas no-lineales	32
2.3.3. El modelo de estado-salida	32
2.3.4. El modelo NARMA	33
2.3.5. Modelo de Estados Recurrentes	33
3. Teoría de Estabilidad de Entrada-Estado (ISS)	35

3.1. Teoría de Estabilidad (Fundamentos)	36
3.1.1. Conceptos Básicos	39
3.2. ISS y funciones de Lyapunov-ISS	40
3.3. Teoría ISS en Tiempo Continuo	46
3.4. Teoría ISS en tiempo discreto	53
3.4.1. Notación y definiciones	53
3.4.2. Ecuaciones en el espacio de estado	54
4. ISS para identificación con redes neuronales en tiempo continuo	59
4.1. Neuro Identificación vía Técnica ISS	61
4.2. Identificación de sistemas con Redes Neuronales mono-capa	62
4.3. Identificación de sistemas con Redes Neuronales Multicapas	68
4.4. Resultado de Simulaciones	76
4.4.1. Simulaciones y Gráficas	77
4.5. Comentarios y Conclusiones:	97
5. ISS para identificación con redes neuronales en tiempo discreto	101
5.1. Identificación de sistemas con Redes Neuronales Recurrentes mono-capa en tiempo discreto	103
5.2. Identificación de sistemas con Redes Neuronales Recurrentes Multicapas en Tiempo-Discreto	108
5.3. Resultado de Simulaciones	113
5.3.1. Simulaciones y Gráficas	115
5.4. Comentarios y conclusiones:	124
6. Conclusiones Generales	129
7. References	131

Índice de figuras

2.1. Estructura de una neurona biológica.	9
2.2. Modelo no lineal de una j-ésima neurona.	10
2.3. Funciones de activación.	14
2.4. Red Neuronal con una sola capa de neuronas.	16
2.5. Modelo de una Red Neuronal de Hopfield.	18
2.6. Estructura de una red multicapa.	20
2.7. Red Neuronal con propagación hacia atrás.	21
2.8. Modelo NARX.	23
2.9. Modelo en Espacio de Estados.	24
2.10. Red Neuronal Recurrente en Espacio de Estados.	26
2.11. Diagrama a bloques de la RNRE.	27
3.1. a) y b) Asintóticamente estables, c) Inestable, d) Estable.	38
3.2. Diagrama a bloques de un sistema de control lineal en tiempo discreto.	56
4.1. Presión en el motor.	78
4.2. Velocidad en el motor.	79
4.3. Presión en el motor.	79
4.4. Velocidad en el motor.	80
4.5. Presión en el motor.	80
4.6. Velocidad en el motor.	81
4.7. Presión en el motor.	82

4.8. Velocidad en el motor.	82
4.9. Presión en el motor.	83
4.10. Velocidad en el motor.	83
4.11. Presión en el motor.	84
4.12. Velocidad en el motor.	84
4.13. Presión en el motor.	86
4.14. Velocidad en el motor.	87
4.15. Presión en el motor.	87
4.16. Velocidad en el motor.	88
4.17. Presión en el motor.	88
4.18. Velocidad en el motor.	89
4.19. Presión en el motor.	90
4.20. Velocidad del motor.	90
4.21. Presión en el motor.	91
4.22. Velocidad en el motor.	91
4.23. Presión en el motor.	92
4.24. Velocidad en el motor.	92
4.25. Error de presión al variar V	93
4.26. Error de la velocidad al variar V	94
4.27. Error de presión al variar V	94
4.28. Error de la velocidad al variar V	95
4.29. Error de presión con redes monocapa y multicapa.	95
4.30. Error de la velocidad con redes monocapa y multicapa.	96
4.31. Error de presión con redes monocapa y multicapa.	96
4.32. Error de la velocidad con redes monocapa y multicapa.	97
4.33. Diagrama Simulink para simulación con redes monocapa.	98
4.34. Diagrama Simulink para simulación con redes multicapa.	99
5.1. Esquema de identificación.	103

5.2. Red Neuronal Recurrente Multicapa en tiempo discreto.	109
5.3. Sistema dinámico caótico de Lorenz.	114
5.4. Resultados para $A=\text{diag}[0.8, 0.8, 0.8]$	116
5.5. Resultados para $A=\text{diag}[1.5, 1.5, 1.5]$	117
5.6. Resultados para $A=\text{diag}[2.2, 2.2, 2.2]$	118
5.7. Resultados obtenidos para $\mu_0=1.8$	119
5.8. Resultados obtenidos para $\mu_0=2.4$	120
5.9. Resultados para $\mu_0=2.8$	121
5.10. Resultados para $M=6$	123
5.11. Resultados para $M=10$	123
5.12. Resultados para $M=20$	124
5.13. Resultados para $\beta=4$	125
5.14. Resultados para $\beta=8$	126
5.15. Resultados para $\beta=10$	127

Capítulo 1

Introducción

La creciente demanda de tecnología en nuestra sociedad requiere nuevos enfoques para los problemas actuales de control e identificación de sistemas cada vez más complejos. *Las Redes Neuronales Artificiales (RNA)* con su masivo paralelismo y su capacidad para aprender prometen mejores soluciones, al menos para algunos problemas. En este momento, la comunidad de control ha puesto su atención en las *RNA* y se pregunta si pueden ser usadas para obtener mejores soluciones para viejos problemas o tal vez soluciones a problemas que se han resistido a nuestros mejores esfuerzos.

El uso de las *RNA* en control e identificación puede verse como un paso natural en la evolución de la teoría de control automático. Viendo hacia atrás la evolución del control se ha visto motivada por tres aspectos fundamentales: la necesidad de tratar con sistemas complejos, la necesidad de cumplir con la creciente demanda de nuevos modelos de control y la necesidad de obtener esos modelos con el menor conocimiento de la planta y el medio ambiente, es decir, la necesidad de controlar sistemas bajo condiciones de incertidumbre. La necesidad de controlar sistemas dinámicos complejos bajo incertidumbre ha conducido a la reevaluación de los métodos convencionales de control, y por lo tanto la necesidad de nuevos métodos es obvia. Esto también ha conducido a un concepto mas general de control, uno que incluye decisión, planeación y aprendizaje, los cuales son cualidades necesarias cuando se desea un sistema con autonomía. En vista de esto, no es de sorprender que la comunidad de

control este muy activa buscando nuevas ideas para resolver de forma efectiva los problemas de control de la sociedad moderna. La necesidad motiva y favorece la creatividad y realización de elementos para suprimirla, y en control no ha sido la excepción, ya que por ejemplo, en el *siglo III* se diseñó un reloj de agua con un mecanismo de retroalimentación, el primer artefacto con retroalimentación en la historia. Así el uso de las *RNA* en control es un paso bastante natural en su evolución. Tales técnicas de *Redes Neuronales Artificiales* parecen ofrecer un mejor entendimiento y quizás la solución de algunos de los problemas de control mas difíciles.

Es claro que las *RNA* pueden ser aceptadas y usadas si ellas resuelven problemas que han sido previamente imposibles o muy difíciles de resolver. Estas pueden ser rechazadas y ser solo producto de la moda, como lo han sido inventos tales como la máquina de escribir eléctrica entre otros, si no promueven su utilidad. El reto es entonces encontrar el mejor camino para utilizar completamente esta nueva y poderosa herramienta en identificación y control de sistemas no lineales complejos.

1.1. Motivación del tema de tesis

Recientes resultados muestran que la técnica de *redes neuronales artificiales* al parecer pueden ser muy efectivas para identificar diversas clases de sistemas no lineales complejos cuando no se tiene información completa del modelo o sistema en cuestión. Por otra parte el enfoque o aproximación de Lyapunov puede ser usado directamente para obtener algoritmos de entrenamiento robusto para redes neuronales en tiempo continuo.

Por lo que respecta a las *redes neuronales en tiempo discreto* estas resultan ser más convenientes para aplicaciones en tiempo real. Ya han sido estudiadas dos tipos de estabilidad para redes neuronales en tiempo discreto. La estabilidad del modelo de redes neuronales puede encontrarse en [3] y [16]. La estabilidad de algoritmos de aprendizaje fué discutida por [7] y [12]. En [12] podemos asumir que las *redes neuronales artificiales* pueden representar del punto de vista teórico, exactamente sistemas no lineales, y concluir que el algoritmo de *backpropagation* o retroalimentación garantiza convergencia exacta.

Es bien conocido que algoritmos normales de identificación son estables para plantas ideales [5]. En presencia de perturbaciones ó dinámicas no modeladas, estos procedimientos adaptivos pueden fácilmente conducir a inestabilidades. La falta o carencia de robustez de parámetros de identificación fué demostrada en [2].

También diversas técnicas de modificación robusta fueron propuestas en [5]. La actualización de peso de neuro-identificación es un tipo de identificación de parámetros; el algoritmo de gradiente normal es estable cuando el modelo de la red neuronal puede competir con la planta no lineal exactamente [12]. Generalmente, tenemos que hacer algunas modificaciones al algoritmo del gradiente normal o backpropagation tal que el proceso de aprendizaje sea estable. Por ejemplo, en [17] algunas fuertes restricciones fueron adicionadas en la ley de aprendizaje, en [16] la dinámica de backpropagation o retroalimentación fue modificada con retriaciones de estabilidad NLq . Otro método generalizado es para usar técnicas de modificación robusta para control adaptable robusto [5] en neuro-identificación. [9] aplicando modificación- σ , [6] usando modificación de regla- δ , y [13] usando la región de *zona-muerta* en los algoritmos de ajuste de pesos. Otra motivación es probar que la ley del gradiente normal y el algoritmo de backpropagation o retroalimentación sin modificación robusta son L_∞ estables.

La *neuro-identificación* se enfoca en el sentido de la aproximación de una caja negra. Todas la incertidumbres pueden ser consideradas como parte de la caja negra, es decir, las dinámicas no modeladas están dentro del modelo, no como incertidumbres estructuradas. Por lo tanto, las técnicas de robustez comúnmente usadas no son necesarias. Por otra parte usando teoría de pasividad, probaremos exitósamente que para redes neuronales recurrentes en tiempo continuo, el algoritmo del gradiente descendente sin modificación robusta serán estables y robustas para cualquier incertidumbre acotada [17], y para identificación en tiempo continuo serán también robústamente estables [18]. Sin embargo, la redes neuronales en tiempo discreto tendrán propiedades similares? En esta tesis, daremos una respuesta a esta pregunta. Podemos contar con una ligera ventaja con respecto a nuestro interés de investigación y que es el siguiente: la identificación sin modificación robusta vía *redes neuronales artificiales* en tiempo discreto no ha sido todavía establecida o documentada en la literarura

especializada.

1.2. Planteamiento del problema

Dada una clase de sistemas no lineales de los cuales solo conocemos su información de *entrada-salida* y *entrada-estado*, con las consecuencias que esto implica: *incertidumbres paramétricas, dinámicas no modeladas, perturbaciones externas o incertidumbres mixtas, ruido en mediciones, etc.* El problema estriba en lograr controlar este tipo de sistemas no lineales, y por controlar se entiende que partiendo de un estado inicial, es posible alcanzar todo el espacio de estados, usando entradas apropiadas de control u [60], si ese espacio de estados es \mathbb{R}^n .

Las clases de sistemas no lineales dinámicos a ser tratados en este trabajo de tesis, son descritos en [20] y están representados por las siguientes ecuaciones:

Modelo I

$$y_{k+1_p} = \sum \alpha_i y_{k+i_p} + g [u_k, u_{k-1}, \dots, u_{k-m+1}] \quad (1.1)$$

Modelo II:

$$y_{k+1_p} = f [y_{k_p}, y_{k-1_p}, \dots, y_{k-n+1_p}] + \sum \beta_i u_{k-i} \quad (1.2)$$

Modelo III:

$$y_{k+1_p} = f [y_{k_p}, y_{k-1_p}, \dots, y_{k-n+1_p}] + g [u_k, u_{k-1}, \dots, u_{k-m+1}] \quad (1.3)$$

Modelo IV:

$$y_{k+1_p} = f [y_{k_p}, y_{k-1_p}, \dots, y_{k-n+1_p}; u_k, u_{k-1}, \dots, u_{k-m+1}] \quad (1.4)$$

Donde y es el vector de estado, n es la dimensión del mismo y u es el control.

Los cuatro modelos presentados anteriormente son: *no lineales en el control, no lineales en el estado, no lineales tanto en el estado como en el control, y finalmente no lineales en el estado y control respectivamente.*

1.3. Artículos publicados

Alejandro Cruz Sandoval and Wen Yu, Input-to-State Stability Approach to Discrete-Time Neural Networks, *International Symposium on Robotics and Automation*, p.p.99-104, ISRA'02, Toluca, Mexico, 2002.

Capítulo 2

Redes Neuronales e Identificación

La historia de las redes neuronales artificiales comenzaría con el científico aragonés Santiago Ramón y Cajal, descubridor de la estructura del sistema nervioso. A finales del siglo XIX la teoría reticularista, que sostenía que el sistema nervioso estaba formado por una red continua de fibras nerviosas, era la creencia extendida. Sin embargo, tras años de trabajo aplicando y perfeccionando la técnica de tinción de Golgi, en 1888 Ramón y Cajal demostró que el sistema nervioso en realidad estaba compuesto por una red de células individuales, las neuronas, ampliamente interconectadas entre si. Pero no solo observó al microscopio los pequeños espacios vacíos que separaban una neuronas de otras, sino que también estableció que la información fluye en la neurona desde las dendritas hacia el axón, atravesando el soma. Se estima que el sistema nervioso contiene alrededor de cien mil millones de neuronas. Vistas al microscopio, este tipo de células puede presentarse en múltiples formas, aunque muchas de ellas presentan un aspecto similar muy peculiar (figura 2.1), con un cuerpo celular o soma (de entre 10 y 80 micras de longitud), del que surge un denso árbol de ramificaciones (árbol dendrítico) compuesto por dendritas, y del cual parte una fibra tubular denominada axón (cuya longitud varía desde las 100 micras hasta el metro en el caso de las neuronas motoras), que también se ramifica en su extremo final para conectar con otras neuronas. Desde un punto de vista funcional, las neuronas constituyen procesadores de información sencillos. Como todo sistema de este tipo, poseen un canal de entrada de información, las

dendritas, un órgano de cómputo, el soma, y un canal de salida, el axón. En las interneuronas el axón envía la información a otras neuronas, mientras que en las neuronas motoras lo hace directamente al músculo. Existe un tercer tipo de neuronas, las receptoras o sensoras, que en vez de recibir la información de otras neuronas, la reciben directamente del exterior (tal sucede, por ejemplo, en los conos y bastones de la retina). Una neurona es una célula viva y, como tal contiene los mismos elementos que forman parte de todas las células biológicas. Una de las características que diferencian a las neuronas del resto de las células vivas, es su capacidad de comunicarse.

La unión entre dos neuronas se denomina sinapsis. En el tipo de sinapsis más común no existe un contacto físico entre las neuronas, sino que estas permanecen separadas por un pequeño vacío de unas 0.2 micras.

El conglomerado masivo de estas células dará paso a estructuras o redes que son parte fundamental y vital del adecuado funcionamiento de un sistema biológico.

La teoría y modelado de redes neuronales artificiales está inspirada en la estructura y funcionamiento de los sistemas nerviosos, donde la neurona es el elemento fundamental. Las redes neuronales artificiales son modelos que intentan imitar el comportamiento del cerebro humano. La adecuada elección de las características y una conveniente estructura, son el procedimiento convencional para construir redes capaces de realizar una determinada tarea. Y es posible utilizar estos modelos neuronales artificiales, para implementarlos adecuadamente en la realización de determinadas actividades y en la solución de problemas que resultarían más difíciles de resolver por medio de otra técnica o imposibles de solucionar, y por lo tanto las redes neuronales artificiales resultan ser instrumentos o técnicas de procesamiento bastante efectivas.

2.1. Unidades de proceso: La neurona artificial

Los sistemas neuronales artificiales tratan de imitar la estructura hardware del sistema nervioso, con la intención de construir sistemas de procesamiento de la información paralelos,

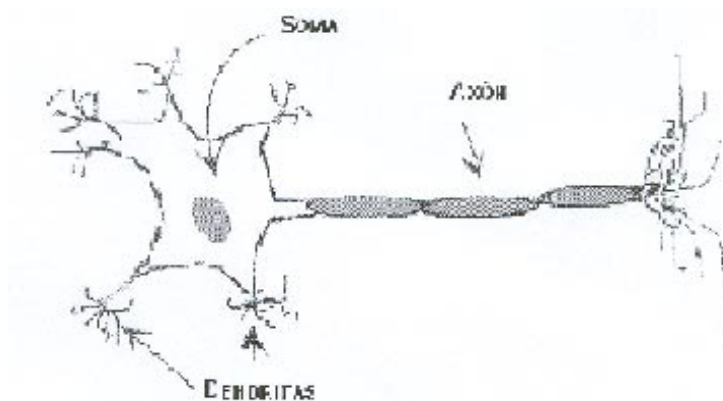


Figura 2.1: Estructura de una neurona biológica.

distribuidos y adaptativos, que puedan presentar un cierto comportamiento "inteligente", por otra parte es necesario establecer los que es una red neuronal artificial y estas son algunas de las definiciones más típicas:

"Sistema de computación hecho de un gran número de elementos simples, elementos de proceso muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas." [Hecht-Niesen 88_a]. "Las redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico." [Kohonen 88_c].

Las redes neuronales artificiales son una tecnología computacional emergente que puede utilizarse en un gran número y variedad de aplicaciones, tanto comerciales como militares. Se pueden desarrollar redes neuronales en un período de tiempo razonable y pueden realizar tareas concretas mejor que otras tecnologías convencionales, incluyendo los sistemas expertos.

Se denomina **procesador elemental o neurona** a un dispositivo simple de cálculo que, a partir de un vector de entrada procedente del exterior o de otras neuronas, proporciona una única respuesta o salida.

Si se tienen n unidades (neuronas), es posible ordenarlas arbitrariamente y designar la

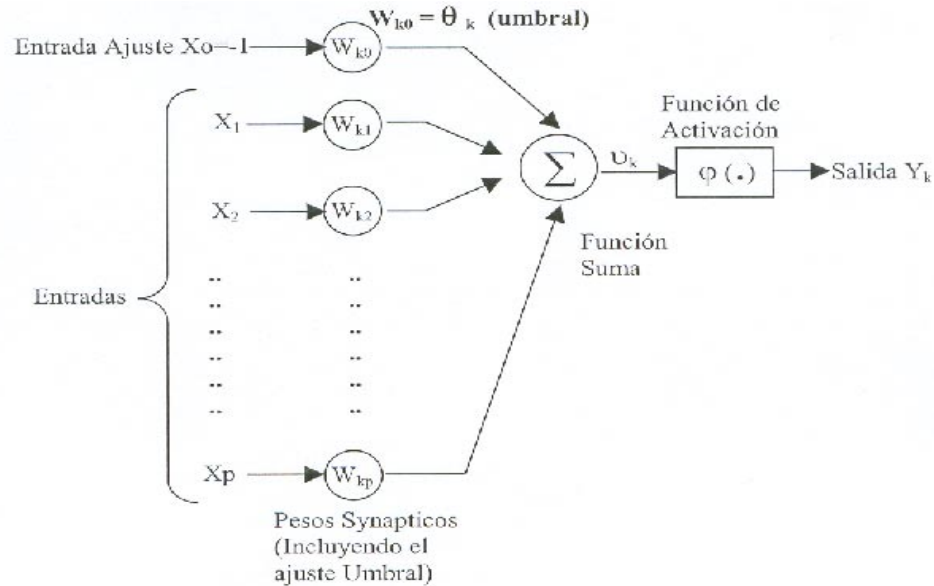


Figura 2.2: Modelo no lineal de una j -ésima neurona.

j - ésima unidad como U_j . Su trabajo es simple y único, y consiste en recibir las entradas de las células vecinas y calcular un valor de salida, el cual es enviado a todas las células restantes (figura 2.2). En cualquier sistema que se esté modelando, es útil caracterizar tres tipos de unidades: *entradas*, *salidas* y *ocultas*. Las unidades de entrada reciben señales desde el entorno; estas entradas (que son a la vez entradas a la red) pueden ser señales provenientes de sensores o de otros sectores del sistema. Las unidades de salida envían la señal fuera del sistema (salidas de red); estas señales pueden controlar directamente potencias u otros sistemas. Las unidades ocultas son aquellas cuyas entradas y salidas se encuentran dentro del sistema, es decir, no tienen contacto con el exterior.

Se conoce como capa o nivel a un conjunto de neuronas cuyas entradas provienen de la misma fuente (que puede ser otra capa de neuronas) y cuyas salidas se dirigen al mismo destino (que puede ser otra capa de neuronas).

2.1.1. Estado de activación de cada neurona

Adicionalmente al conjunto de unidades, la representación necesita los estados del sistema en un tiempo t . Esto se especifica por un vector de n números reales $N(t)$, que representa el estado de activación del conjunto de unidades de procesamiento. Cada elemento del vector representa la activación de una unidad en el tiempo t . La activación de una unidad u_i en el tiempo t se designa por $x_i(t)$; es decir:

$$N(t) = (x_1(t), a_2(t), \dots, a_i(t), \dots, x_n(t)) \quad (2.1)$$

El procesamiento que realiza la red se ve como la evolución de un patrón de activación en el conjunto de unidades que lo componen a través del tiempo. Todas las neuronas que componen la red se hallan en cierto estado. En una visión simplificada, es posible decir que hay dos posibles estados, *reposo* y *excitado*, a los que denominaremos globalmente *estados de activación*, y a cada uno de los cuales se le asigna un valor. Los valores de activación pueden ser continuos o discretos. Además pueden ser limitados o ilimitados. Si son discretos, suelen tomar un conjunto pequeño de valores o bien valores binarios. En notación binaria, un estado activo se indicará por un 1, y se caracteriza por la emisión de un impulso por parte de la neurona (potencial de activación), mientras que un estado inactivo se indicará por un 0, y significará que la neurona está en reposo.

Finalmente, es necesario saber qué criterios o reglas siguen las neuronas para alcanzar tales estados de activación. En principio, esto va a depender de dos factores: a) Por un lado, puesto que las propiedades macroscópicas de las redes neuronales no son producto de actuación de elementos individuales, sino del conjunto como un todo, es necesario tener idea del mecanismo de interacción entre las neuronas. El estado de activación estará fuertemente influenciado por tales interacciones, ya que el efecto que producirá una neurona sobre otra será proporcional a la fuerza, peso o magnitud de la conexión entre ambas. b) Por otro lado, la señal que envía cada una de las neuronas a sus vecinas dependerá de su propio estado de activación.

2.1.2. Función de salida o transferencia

Entre las unidades o neuronas que forman una red neuronal artificial existe un conjunto de conexiones que unen unas a otras. Cada unidad transmite señales a aquellas que están conectadas con su salida. Asociada con cada unidad u_i hay una función de salida $f_i(x_i(t))$, que transforma el estado actual de activación $x_i(t)$ en una señal de salida $y_i(t)$; es decir:

$$y_i(t) = f_i(x_i(t)) \quad (2.2)$$

El vector que contiene las salidas de todas las neuronas en un instante t es:

$$Y(t) = (f_1(x_1(t)), f_2(x_2(t)), \dots, f_i(x_i(t)), \dots, f_n(x_n(t))) \quad (2.3)$$

En algunos modelos, esta salida es igual al nivel de activación de la unidad, en cuyo caso la función f_i es la función identidad, $f_i(x_i(t)) = x_i(t)$. A menudo, f_i es de tipo sigmoideal, y suele ser la misma para todas las unidades. Existen cuatro funciones de transferencia típicas que determinan distintos tipos de neuronas: la función escalón, la función lineal y mixta, la función sigmoideal y la función gaussiana.

La función escalón o umbral únicamente se utiliza cuando las salidas de la red son binarias (dos posibles valores). La salida de una neurona se activa sólo cuando el estado de activación es mayor o igual que cierto valor umbral (offset-la función puede estar desplazada sobre los ejes). La función lineal o identidad equivale a no aplicar función de salida. Se utiliza muy poco. Las funciones mixtas y sigmoideal son las más apropiadas cuando queremos como salida información analógica. Las funciones matemáticas que describen estas funciones se indican a continuación.

2.3.1 Función de activación escalón

La forma más fácil de definir la activación de una neurona es considerar que ésta es binaria. La función de transferencia escalón se asocia a neuronas binarias en las cuales cuando la suma

de las entradas es mayor o igual que el umbral de la neurona, la activación es 1; si es menor, la activación es 0 (o -1).

La figura (2.3 (a)) nos representa la gráfica de una función escalón, y cuya representación matemática esta dada por la siguiente ecuación:

$$f(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ -1 & \text{si } x < 0 \end{cases} \quad (2.4)$$

2.3.2 Función de activación saturación

La figura (2.3 (b)), muestra la gráfica de la función de activación saturación. La ecuación siguiente da la representación matemática de la función:

$$f(x) = \begin{cases} -1 & \text{si } x < -c \\ 1 & \text{si } x > c \\ ax & \text{en otro caso} \end{cases} \quad (2.5)$$

2.3.3 Función de activación sigmoideal

Es una función continua, la figura (2.3 (c)), muestra la gráfica de una función sigmoideal por lo tanto el espacio de los estados de activación es un intervalo del eje real. Es de hecho la función más común en la construcción de Redes Neuronales Artificiales. Esta definida como una función estrictamente creciente que exhibe un balance entre una conducta lineal y no lineal y es definida por la siguiente ecuación, donde p es la pendiente de la función de activación:

$$f(x) = \frac{1}{1 + \exp(-px)} \quad (2.6)$$

2.3.4 Función de activación tangente hiperbólica

Al igual que la sigmoide, es una de las principales funciones en la construcción de las Redes Neuronales Artificiales especialmente como ya se ha mencionado en tareas de identificación y

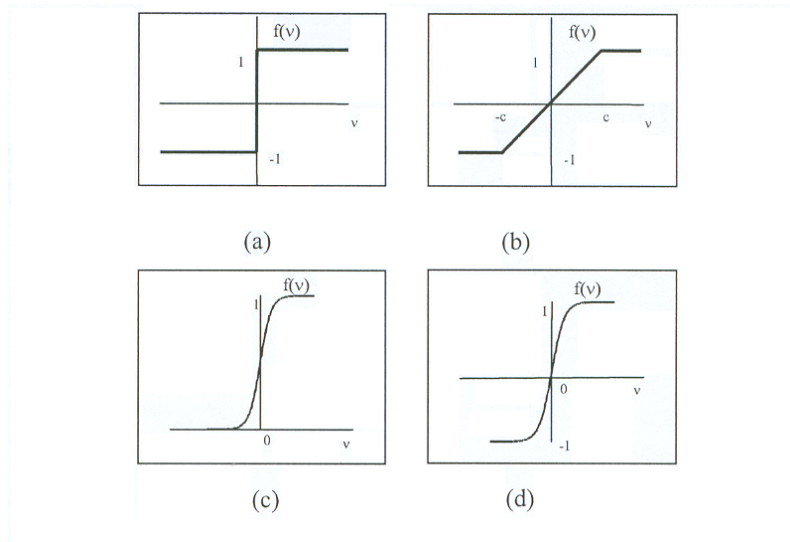


Figura 2.3: Funciones de activación.

control , la figura (2.3 (d)) representa la gráfica de esta función. La representación matemática de tal función está dada por la siguiente ecuación, donde p es la pendiente de la función de activación:

$$f(x) = \frac{1 - \exp(-px)}{1 + \exp(-px)} \quad (2.7)$$

2.2. Topología de redes neuronales

Las conexiones que unen a las neuronas que forman una red neuronal artificial tienen asociado un peso, que es el que hace que la red adquiera conocimiento. Consideremos y_i como el valor de salida de una neurona i en un instante dado. Una neurona recibe un conjunto de señales que le dan información del estado de activación de todas las neuronas con las que se encuentra conectada. Cada conexión (sinapsis) entre la neurona i y la neurona j se pondera por medio de un peso w_{ij} . Normalmente, como simplificación, se considera que

el efecto de cada señal es aditivo, de tal forma que la entrada neta que recibe una neurona x_j es la suma del producto de cada señal individual por el valor de la sinapsis que conecta ambas neuronas. Esta regla muestra el procedimiento a seguir para combinar los valores de entrada a una unidad con los pesos de los valores que llegan a esa unidad y es conocida como regla de propagación hacia adelante.

Es muy común utilizarse una matriz W con todos los pesos w_{ji} que reflejan la influencia que sobre la neurona j tiene la neurona i . W es un conjunto de elementos positivos, negativos o nulos. Si w_{ji} es positivo, indica que la interacción entre las neuronas i y j es excitadora, es decir, siempre que la neurona i este activada, la neurona j recibirá una señal de i que tenderá a activarla. Si w_{ji} es negativo, la sinapsis será inhibitoria. En este caso, si i está activada, enviará una señal a j que tenderá a desactivar a ésta. Finalmente, si $w_{ji} = 0$, se supone que no hay conexión entre ambas neuronas.

Existen cuatro aspectos que caracterizan una red neuronal: su topología, el mecanismo de aprendizaje, tipo de asociación realizada entre la información de entrada y salida, y por último, la forma de representación de estas informaciones. La topología o arquitectura de las redes neuronales consiste en la organización y disposición de las neuronas en la red formando capas o agrupaciones de neuronas más o menos alejadas de la entrada y salida de la red. En este sentido, los parámetros fundamentales de la red son: el número de capas, el número de neuronas, el grado de conectividad y el tipo de conexión entre neuronas.

Cuando se realiza una clasificación de las redes en términos topológicos, se suele distinguir entre las redes con una sola capa o nivel de neuronas y las redes con múltiples capas (2, 3, etc.).

2.2.1. Arquitectura de redes neuronales recurrentes

Las redes neuronales se pueden clasificar en dos grupos debido a su estructura, red neuronal *feed-forward* (*RNFF*) hacia adelante y *redes neuronales recurrentes* (*RNR*) o redes dinámicas. En donde la característica primordial de las redes neuronales *feed-forward* es que no existe conexión hacia atrás ni autorecurrente, ni lateral y se utilizan principalmente en

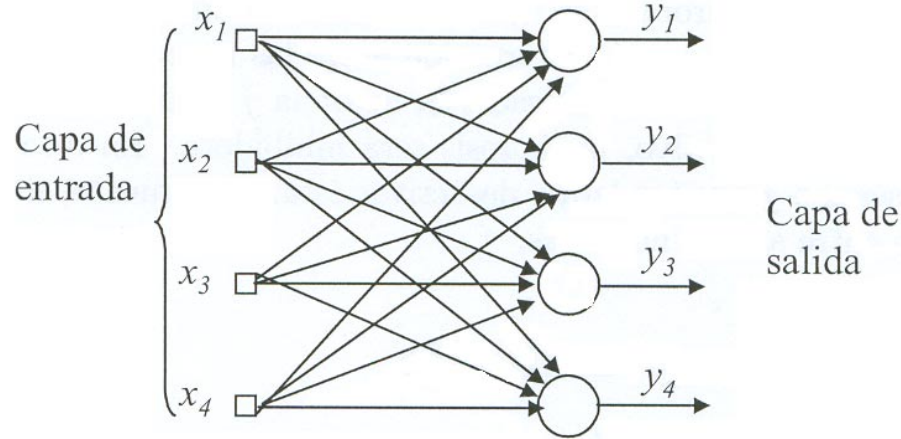


Figura 2.4: Red Neuronal con una sola capa de neuronas.

tareas de reconocimiento o clasificación de patrones [23].

En tanto que las *redes neuronales recurrentes (RNR)*, son aquellas donde la información circula tanto hacia adelante como hacia atrás durante el funcionamiento de la red.

Redes monocapa (1 capa)

En las redes monocapa, como la red de **HOPFIELD** y la red **BRAIN-STATE-IN-A-BOX**, se establecen conexiones laterales entre las neuronas que pertenecen a la única capa que constituye la red. En su forma más simple, un red neuronal tiene una capa de entrada y una capa de salida. La figura 2.4 muestra la gráfica de esta red, para el caso de cuatro nodos en la capa de entrada y cuatro en la capa de salida. Esta red es de una sola capa, por que ningún cálculo es realizado en la capa de entrada.

Las redes monocapa se utilizan típicamente en áreas relacionadas con lo que se conoce como autoasociación; por ejemplo, para regenerar informaciones de entrada que se presentan a la red incompletas o distorsionadas.

El modelo de Hopfield Sin duda, uno de los principales responsables del desarrollo que ha experimentado el campo de la computación neuronal ha sido *J.J. Hopfield* físico norteamericano, quién construyó un modelo de red [Hopfield 82] con el número suficiente de simplificaciones como para poder extraer analíticamente información sobre las características relevantes del sistema, conservando las ideas fundamentales de las redes construidas en el pasado y presentando una serie de funciones básicas de sistemas neuronales reales. Además, Hopfield supo establecer un paralelismo lo cual ha permitido aplicar todo un conjunto de técnicas bien conocidas en este campo y, con ello, producir un avance en la comprensión del funcionamiento de las redes neuronales. Con su aportación, Hopfield redescubrió el mundo casi olvidado de las redes autoasociativas, caracterizadas por una nueva arquitectura y un nuevo funcionamiento, a las que se tuvo que añadir otro tipo de reglas de aprendizaje. Las consecuencias fueron redes con un comportamiento diferente a las diseñadas con estructura *feedforward* (*ADALINE/MADALINE, PERCEPTRON,...*). El modelo de Hopfield como el de la figura 2.5, consiste en una *red monocapa* con N neuronas cuyos valores de salida son binarios: 0/1 ó -1/+1. En la versión original del modelo (DH: Discrete Hopfield) las funciones de activación de las neuronas eran del tipo escalón. Se trataba, por tanto, de una red discreta, con entradas y salidas binarias; sin embargo, posteriormente Hopfield desarrolló una versión continua con entradas y salidas analógicas [Hopfield 84], utilizando neuronas con funciones de activación tipo sigmoideal (CH: Continuous Hopfield), como las que se emplearon en esta tesis. Una de las características del modelo de Hopfield, es que se trata de una *red autoasociativa*. Así, varias informaciones (patrones) diferentes pueden ser almacenadas en la red, como si de una memoria se tratase, durante la etapa de aprendizaje. Posteriormente, si se presenta a la entrada alguna de las informaciones almacenadas, la red evoluciona hasta estabilizarse, ofreciendo entonces en la salida la información almacenada, que coincide con la presentada en la entrada. Si, por el contrario, la información de entrada no coincide con ninguna de las almacenadas, por estar distorsionada o incompleta, la red evoluciona generando como salida la más parecida.

La red Hopfield tiene un mecanismo de aprendizaje *off-line*. Por tanto, existe una etapa de aprendizaje y otra de funcionamiento de la red. En la etapa de aprendizaje se fijan los

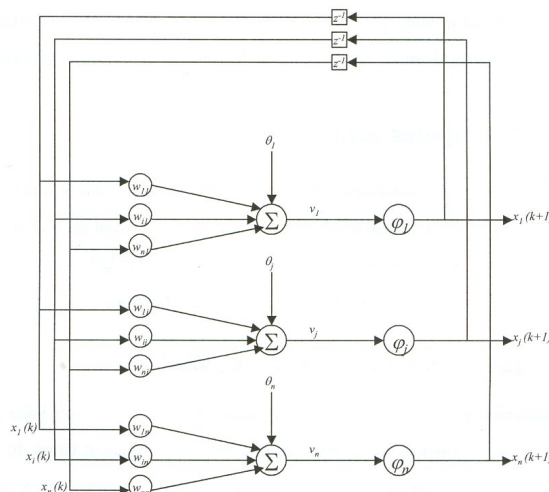


Figura 2.5: Modelo de una Red Neuronal de Hopfield.

valores de los pesos en función de las informaciones que se pretende que *memorice* o almacene la red. Una vez establecidos, la red entra en funcionamiento.

Esta red utiliza un aprendizaje no supervisado de tipo *hebbiano*, de tal forma que el peso de una conexión entre una neurona i y otra j se obtiene mediante el producto de los componentes i –ésimo y j –ésimo del vector que representa la información o patrón que debe almacenar. La elección de esta regla de aprendizaje por Hopfield fue, entre otras razones, debido a que asegura la estabilidad de la red, es decir, la convergencia hacia una respuesta estable cuando se presenta una información de entrada.

Muchas de las investigaciones acerca de la estabilidad de las redes se basan en el establecimiento de una función, denominada *función de energía de la red*, para representar los posibles estados (puntos de equilibrio) de la red. De hecho, una de las causas por la que se considera a Hopfield responsable de impulsar el desarrollo en el campo de las redes neuronales, es precisamente el haber aplicado modelos matemáticos como éste, lo cual constituyó la base de posteriores trabajos sobre redes neuronales.

La función de energía de una red de Hopfield discreta tiene la siguiente forma:

$$E = -\frac{1}{2} \sum_{i=1}^N \left(\sum_{\substack{j=1 \\ i \neq j}}^N w_{ij} y_i y_j + \sum_{i=j}^N \theta_i y_j \right)$$

donde: w_{ij} es el peso de la conexión entre las neuronas i y j .

y_i valor de salida de la neurona i .

y_j valor de salida de la neurona j .

θ_i Umbral de la función de activación de la neurona i .

Cuando se presenta a la entrada de la red una nueva información, esta evoluciona hasta alcanzar un mínimo de la función de energía, generando una salida estable.

Redes multicapa

Las redes multicapa son aquellas que disponen de conjuntos de neuronas agrupadas en varios (2, 3, etc.) niveles o capas. En estos casos, una forma para distinguir la capa a la que pertenece una neurona, consistiría en fijarse en el origen de las señales que recibe a la entrada y el destino de la señal de salida. Normalmente, todas las neuronas de una capa reciben señales de entrada de otra capa anterior, más cercana a la entrada de la red, y envían las señales de salida a una capa posterior, más cercana a la salida de la red. A estas conexiones se les denomina conexiones hacia adelante o *feedforward*.

Sin embargo, en un gran número de estas redes también existe la posibilidad de conectar las salidas de las neuronas de capas posteriores, a estas conexiones se les denomina conexiones hacia atrás o *feedback*. Estas dos posibilidades permiten distinguir entre dos tipos de redes con múltiples capas: las redes con conexiones hacia adelante o redes *feedforward* y las redes que disponen de conexiones tanto hacia adelante como hacia atrás o redes *feedforward-feedback*.

Redes con conexiones hacia adelante (feedforward)

En las redes *feedforward*, todas las señales neuronales se propagan hacia adelante a través de las capas de la red. No existen conexiones hacia atrás (ninguna salida de neuronas de una capa i se aplica a la entrada de neuronas de capas $i - 1$, $i - 2$), y normalmente tampoco

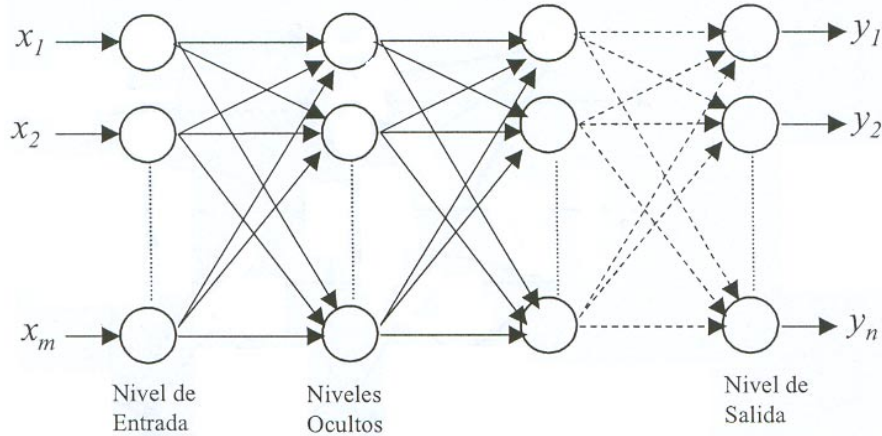


Figura 2.6: Estructura de una red multicapa.

autorrecurrentes (salida de una neurona aplicada a la entrada de neuronas de la misma capa), excepto en el caso de los modelos de red propuestos por **Kohonen** denominados **Learning Vector Quantizer (LVQ)** y **Topology Preserving Map (TPM)**, en las que existen unas conexiones implícitas muy particulares entre las neuronas de la capa de salida.

Las redes *feedforward* más conocidas son : Perceptrón, Adaline, Madaline, Linear Adaptive Memory (LAM), Drive-Reinforcement, Backpropagation. Todas ellas son especialmente útiles en aplicaciones de reconocimiento o clasificación de patrones. La figura 2.6 indica una red *feedforward*.

Redes con conexiones hacia adelante y hacia atrás (feedforward-feedback)

En este tipo de redes circula información tanto hacia adelante (forward) como hacia atrás (backward) durante el funcionamiento de la red. Para que esto sea posible, existen conexiones *feedforward* y conexiones *feedback* entre las neuronas. La figura 2.7 muestra una red de este tipo.

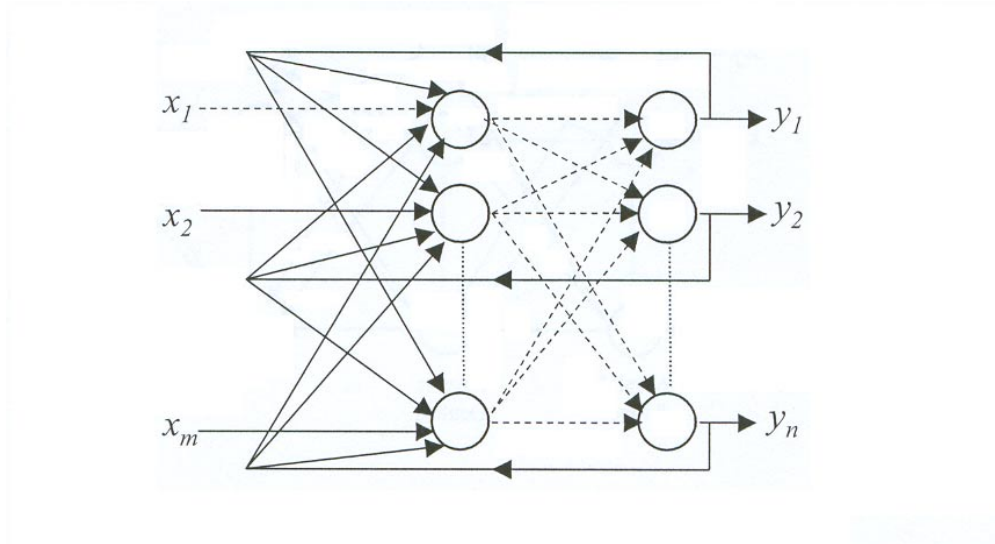


Figura 2.7: Red Neuronal con propagación hacia atrás.

En general, excepto el caso particular de las redes Cognitron y Neocognitron, suelen ser bicapa (dos capas), existiendo por tanto dos conjuntos de pesos: los correspondientes a las conexiones *feedforward* de la primera capa (capa de entrada) hacia la segunda (capa de salida) y los de las conexiones *feedback* de la segunda a la primera. Los valores de los pesos de estos dos tipos de conexiones no tienen porque coincidir, siendo diferentes en la mayor parte de los casos.

Este tipo de estructura (bicapa) es particularmente adecuada para realizar una asociación de una información o patrón de entrada (en la primera capa) con otra información o patrón de salida en la segunda capa (lo cual se conoce como heteroasociación), aunque también pueden ser utilizadas para la clasificación de patrones.

Algunas redes de este tipo tiene un funcionamiento basado en lo que se conoce como *resonancia*, de tal forma que las informaciones en la primera y segundas capas interactúan entre sí hasta que alcanzan un estado estable. Este funcionamiento permite un mejor acceso a las informaciones almacenadas en la red.

Los dos modelos de red *feedforward-feedback* de dos capas más conocidos son la red *Art* (*Adaptive Resonance Theory*) y la red *BAM* (*Bidirectional Associative Memory*).

Modelo entrada-salida recurrente

La figura 2.8 muestra la arquitectura de una red recurrente genérica que sigue naturalmente del perceptrón multicapa. El modelo tiene solo una entrada la cual se aplica a una red de retardo (memoria) de q unidades. Tiene una sola salida que es retroalimentada a la entrada por medio de otra red de retardo también de q unidades. El contenido de estas dos redes de retardo (memorias) es usado para alimentar la capa de entrada del perceptrón multicapa. El valor presente del modelo de entrada es denotado por u_n y el valor correspondiente de la salida del modelo y_{n+1} ; esto es, la salida esta adelantada a la entrada por una unidad de tiempo. Por lo tanto, el vector de señales aplicado a la capa de entrada del perceptrón multicapa consiste de una ventana de datos como sigue:

* Valores presentes y pasados de la entrada, llamados $u_n, u_{n-1}, \dots, u_{n-q+1}$, los cuales representan entradas exógenas originadas desde fuera de la red.

* Valores retrasados de la salida, $y_n, y_{n-1}, \dots, y_{n-q+1}$ en los cuales la salida del modelo y_{n+1} es regresada.

Entonces la red recurrente de la figura 2.8 se le refiere como modelo NARX (modelo autoregresivo no lineal con entradas exógenas). El comportamiento dinámico del modelo NARX es descrito por la siguiente ecuación:

$$y_k = f(y_n, y_{n-1}, \dots, y_{n-q+1}, u_n, u_{n-1}, \dots, u_{n-q+1})$$

donde F es una función no lineal de sus argumentos. Note en el dibujo que, las dos redes de retardo (memoria) son de tamaño q , pero esto en general no es así.

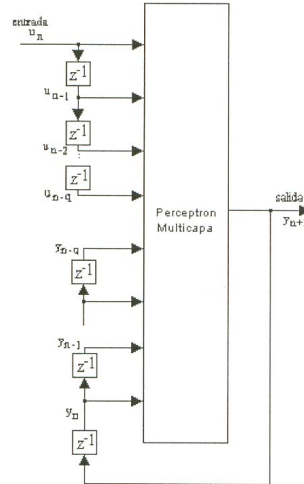


Figura 2.8: Modelo NARX.

Modelo en espacio de estado

La figura 2.9 muestra el diagrama a bloques de otra red neuronal recurrente. Aquí las neuronas de la capa oculta definen el estado de la red. La salida de la capa oculta es retroalimentada a la capa de entrada por medio de una red de retardo, por lo tanto la capa de entrada consiste en una concatenación de nodos de retroalimentación y nodos fuente. La red está conectada al medio externo vía los nodos fuente. El número de unidades de retardo usadas para alimentar la salida de la capa oculta a la capa de entrada determina el orden del modelo, la red de retardo va multiplicada por una matriz identidad dependiendo de la dimensión del estado. Sea el vector de entrada u_n de dimensión $m \times 1$, y el vector de estados x_k de dimensión $q \times 1$ que es la salida de la capa oculta en el tiempo discreto k . El comportamiento dinámico de este modelo está dado por el par de ecuaciones siguiente:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) \\ y_k &= Cx_k \end{aligned} \tag{2.8}$$

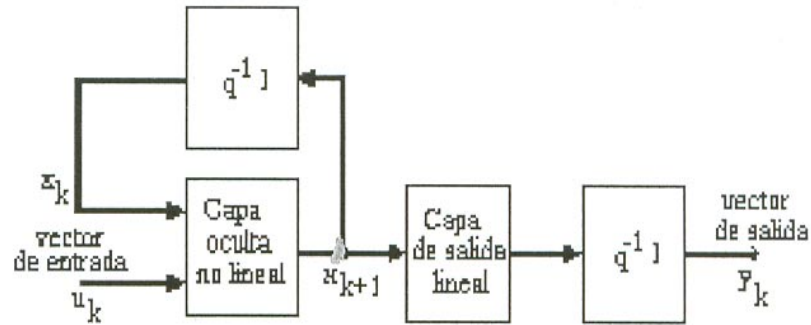


Figura 2.9: Modelo en Espacio de Estados.

donde $f(\cdot, \cdot)$ es una función no lineal la cual caracteriza la capa oculta, C es una matriz de pesos sinápticos la cual caracteriza la capa de salida. Esto es, tenemos una capa oculta no lineal y una capa de salida lineal. Una arquitectura similar al modelo de espacio de estados, es la red neuronal de ([45], Elman (1990)), a excepción de que la capa de salida también es no lineal y no hay red de retardo de la salida.

Perceptrón multicapa recurrente

La tercera arquitectura considerada aquí es el Perceptrón Multicapa Recurrente (RMLP) introducida por Puskorius en 1996, [45]. Básicamente tiene una o dos capas ocultas, y además cada capa tiene retroalimentación alrededor de ella. Denotamos el vector x_{I_k} como la salida de la primera capa oculta, x_{II_k} la salida de la segunda capa oculta, y así. Sea el vector x_{0_k} como la salida de la capa de salida. Entonces el comportamiento dinámico de RMLP es descrito por el siguiente sistema de ecuaciones:

$$\begin{aligned} x_{I_{k+1}} &= \sigma_I(x_{I_k}, u_k) \\ x_{II_{k+1}} &= \sigma_{II}(x_{II_k}, x_{I_{k+1}}) \end{aligned} \quad (2.9)$$

·
·
·

$$x_{0_{k+1}} = \sigma_0(x_{0_k}, x_{P_{k+1}})$$

donde $\sigma_I, \sigma_{II}, \dots$ son las funciones de activación que caracterizan la primera capa oculta, la segunda y así sucesivamente, y σ_0 es la función lineal de la capa de salida del RMLP, y P denota el número de capas ocultas en la red.

2.2.2. Red neuronal recurrente en espacio de estado

La noción de estado juega un papel vital en la formulación matemática de un sistema dinámico. El estado de un sistema dinámico es formalmente definido como un conjunto de cantidades que resumen toda la información acerca del comportamiento pasado del sistema el cual es necesitado para unívocamente describir su comportamiento futuro, excepto por los efectos externos los cuales provienen de la aplicación de una entrada. Sea el vector de entrada u_n de dimensión $m \times 1$, y el vector de salida y_n de dimensión $p \times 1$ en el tiempo discreto k . En términos matemáticos el comportamiento dinámico del sistema, si se asume libre de ruido, es descrito por el siguiente par de ecuaciones no lineales:

$$x_{n+1} = \sigma(W_a x_n + W_b u_n) \tag{2.10}$$

$$y_n = C x_n \tag{2.11}$$

donde $W_a \in \mathbb{R}^{q \times q}$, $W_b \in \mathbb{R}^{q \times (m+1)}$, $C \in \mathbb{R}^{q \times q}$, $\sigma : \mathbb{R}^q \rightarrow \mathbb{R}^q$ diagonal.

Para la interpretación de las matrices W_a , W_b , C y la función no lineal σ , diremos:

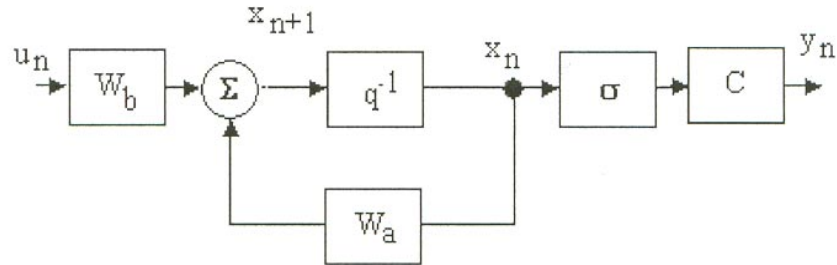


Figura 2.10: Red Neuronal Recurrente en Espacio de Estados.

* La matriz W_a representa los pesos sinápticos de las neuronas en la capa oculta que están conectadas a los nodos de retroalimentación en la capa de entrada a través de un retardo. La matriz W_b representa los pesos sinápticos de esa capa oculta que están conectados a nodos fuente en la capa de entrada, se asume que el umbral son absorbidos por la matriz W_b para las neuronas de la capa oculta, esto es la matriz W_b tiene la dimensión de la entrada más uno que representa el umbral.

* La matriz C representa los pesos sinápticos de las p neuronas lineales en la capa de salida, las cuales están conectadas a la capa oculta.

* La función σ , representa la función de activación sigmoide de la capa oculta.

Esta es una propiedad importante de una red neuronal, descrita en el modelo de espacio de estado de este tipo figura 2.10, que pueden aproximar una amplia clase de sistemas no lineales dinámicos,[64].

2.2.3. Red neuronal recurrente entrenable

La topología de la RNRE ([24], [25], Baruch, y otros, 1996) es dada en las siguientes ecuaciones:

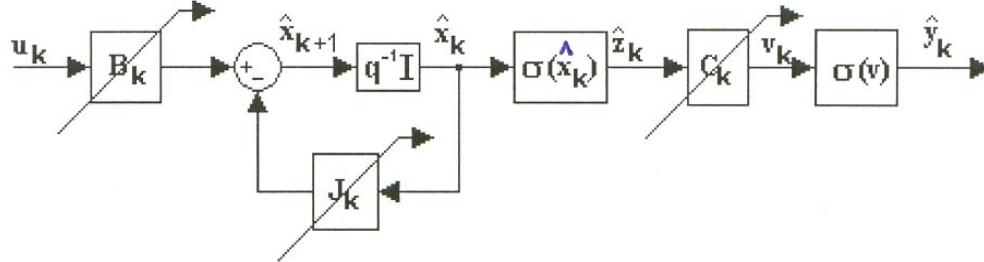


Figura 2.11: Diagrama a bloques de la RNRE.

$$\begin{aligned}
 \hat{x}_{k+1} &= J_k \hat{x}_k + B_k u_k \\
 \hat{z}_k &= \sigma(\hat{x}_k) \\
 \hat{y}_k &= \sigma(C_k \hat{z}_k)
 \end{aligned} \tag{2.12}$$

La topología de la RNRE es dada en la figura 2.11:

$J_k \in \mathbb{R}^{n \times n}$ es una matriz Canónica de Jordan con valores propios en el círculo unitario.

$$J_k = \text{block_diag}(J_{k_i}) \tag{2.13}$$

$$|J_{ii}| < 1 \tag{2.14}$$

Donde (Y, X, U) son la salida, el estado y la entrada respectivamente y de dimensiones l, n, m ; J_k es una matriz diagonal a bloques, las matrices B_k y C_k de pesos son de dimensión $(n \times m)$ y $(l \times m)$ y $\Gamma(\cdot)$, $\Phi(\cdot)$ son funciones de activación vector valuadas, con elementos: saturación, sigmoides o tangentes hiperbólicas.

Considere la RNRE descrita por 2.12:

$x_k \in \mathbb{R}^n$ es el vector de estados en el tiempo k ,

$\sigma : \mathbb{R}^n \rightarrow \mathbb{R}^m$ es el mapeo $L_\Omega(C^2)$ diferentes funciones de activación en las capas ocultas y de salida.

Observación 1 El mapeo σ se asume usualmente de estructura diagonal con elementos sigmoidales acotados σ_{ii} , esto es,

$$\sigma_{ii} := \frac{a_i}{1 + \exp -b_i^T \hat{x}} - c_i \quad (i = \text{mín} \{n, m\})$$

$$\bar{\sigma} := \sup_{\hat{x}_k \in \mathbb{R}^n} \|\sigma(x)\|,$$

lo cual obviamente pertenece a la clase de funciones $L_\Omega(C^2)$, [53].

Dado que la red neuronal propuesta tiene una estructura de sistema lineal variante en el tiempo con ello podemos ver las propiedades de controlabilidad, observabilidad e identificabilidad del sistema, [22, 23], aunque para el caso en que las funciones de activación sean matrices identidad las nociones de controlabilidad y observabilidad las podemos obtener de la teoría clásica de los sistemas lineales [61].

Uno de los objetivos de este trabajo es realizar un profundo análisis topológico de estos modelos que dada la estructura de Jordan ya de por sí presenta restricciones en los pesos que se encuentran en el lazo de retroalimentación, ya que con ello se preserva la estabilidad, aunque con el análisis que se hace en el siguiente capítulo, demostramos que no necesariamente es así.

La matriz J contribuye, pero la estabilidad y convergencia no sólo dependen de esta matriz, de hecho en el capítulo siguiente (Controlabilidad y Observabilidad) veremos que también depende del tipo de función de activación, así como de las matrices de entrada y de salida, de los bias, además depende de los tipos de incertidumbres presentadas (externas) ó presentes en el modelo (dinámicas no modeladas) o incertidumbres mixtas.

Otra ventaja además de las propiedades de controlabilidad y observabilidad que a continuación se verán, es que este tipo de estructura, es determinado por los bloques B y C , que corresponden a las de la matriz J , con ello podemos evaluar la capacidad de aprendizaje de la Red Neuronal Recurrente (RNR) con lo cual se convierte en lo que ahora llamare-

mos red neuronal recurrente entrenable (RNRE). La principal ventaja del modelo canónico de Jordan de la RNR discreta, es de ser un modelo neuronal híbrido universal con una o dos capas feedforward y una capa oculta recurrente, donde la matriz de los pesos J es una matriz diagonal por bloques, de manera que el número de parámetros es mínimo y el funcionamiento es completamente paralelo. La arquitectura de la RNR descrita puede servir como un predictor-estimador de un paso hacia adelante. Otra cosa que hay que notar es que la RNR es globalmente no lineal, pero lineal localmente, de manera que las matrices J_k, B_k, C_k generadas durante el aprendizaje determinan el comportamiento de toda la red. Además, el modelo de red neuronal recurrente (RNR) es robusta debido a que las leyes de adaptación de pesos es dinámica.

2.3. Redes neuronales para Identificación

Las redes neuronales y en particular las redes neuronales recurrentes poseen una gran capacidad en la aproximación de funciones, por lo tanto su aplicabilidad ha sido extensa en el campo de la identificación y el control [33], [17], [15], [11], [60], entre muchos otros. El proceso de identificación es muy importante en el control, porque es ahí donde podemos obtener información del proceso a controlar, es decir, podemos entre otras cosas conocer los parámetros de la estructura de una planta u obtener un modelo que se comporte igual a esta. Existen procesos muy complicados como los biológicos, químicos o económicos en los que tener un modelo exacto es casi imposible, en estos casos la planta o proceso es tomado como una caja negra donde únicamente se conocen los datos de entrada y los datos de salida.

Mientras que la complejidad del modelo requerido depende del propósito del modelado e identificación. Por ejemplo, es necesario enfatizar las necesidades básicas científicas de modelos cuantitativos, cuyos propósitos retornarán cierta capacidad para predecir nuevos fenómenos. La predicción o pronóstico son áreas estrechamente relacionadas para modelado e identificación; mientras los intentos para predecir estados futuros de un sistema son limitados por la precisión del modelo usado y el rango de correlación de procesos aleatorios que afectan al sistema. En este contexto, es importante poder representar acciones externas

y perturbaciones externas, extrayendo y usando el conocimiento de características estadísticas de variables aleatorias, mientras exista por regla general, poca teoría o posibilidades prácticas para determinar tales características por adelantado.

Los análisis de sistemas de control y diseño proporcionan un rico campo para la aplicación de modelado e identificación. Un mecanismo de control es aquel que sensa el error de control, es decir, la diferencia entre los estados actuales y los deseados, y entonces se inicia una serie de procesos y acciones, que en cambio producen efectos para contrarestar y minimizar el error de control. Este principio de control introduce el importante concepto de retroalimentación, que para sistemas lineales puede ser ilustrado en términos de la función de transferencia. Dentro de este contexto también permite hacer predicciones cuantitativas a características cruciales concernientes a sistemas de control tales como las condiciones de estabilidad y el desarrollo de comportamientos oscilatorios.

La identificación es un proceso de suma importancia e interés para el control de procesos, ya que de ahí podemos obtener información del comportamiento del sistema, es decir, podemos entre otras cosas conocer los parámetros de la estructura de una planta y así poder obtener un modelo que se comporte de manera muy semejante a esta.

La caracterización e identificación son problemas fundamentales en la teoría de sistemas. El problema de caracterización es el concerniente con la representación matemática de un sistema, un modelo de un sistema es expresado como un operador P de un espacio de entrada \mathcal{U} a un espacio de salida \mathcal{Y} y el objetivo es caracterizar la clase \mathcal{P} a la cual P pertenece. Dada la clase \mathcal{P} y el hecho que $P \in \mathcal{P}$, el problema de identificación es determinar la clase $\hat{\mathcal{P}} \subset \mathcal{P}$ tal \hat{P} aproxime a \mathcal{Y} en algún sentido deseado. En sistemas estáticos, los espacios \mathcal{U} y \mathcal{Y} son subconjuntos de \mathbb{R}^n y \mathbb{R}^m , respectivamente, mientras que en sistemas dinámicos generalmente se asume que son funciones de *Lebesgue* integrables en el intervalo $[0, T]$ ó $[0, \infty)$. En ambos casos, el operador P es definido implícitamente por el par de entrada-salida. La elección de la clase de los modelos de identificación $\hat{\mathcal{P}}$, así también como el método específico usado para determinar \hat{P} , depende de una gran variedad de factores que están relacionados con la precisión deseada, así como también con el manejo analítico.

2.3.1. Identificación de sistemas

El problema de reconocimiento de patrones es un ejemplo típico de identificación de sistemas estáticos. En estos problemas, conjuntos compactos $U_i \subset \mathbb{R}^n$ son mapeados a elementos $y_i \in \mathbb{R}^m$; ($i = 1, 2, \dots$) en el espacio de salida por una función de decisión P . Los elementos de U_i denotan los vectores de patrones correspondientes a la clase y_i . En sistemas dinámicos, el operador P , que representa a una planta dada, es definido implícitamente por el par de entrada-salida de las funciones del tiempo $u(t), y(t), t \in [0, T]$. En ambos casos el objetivo es determinar \hat{P} tal que :

$$\|y - \hat{y}\| = \|P(u) - \hat{P}(u)\| \leq \epsilon, \quad u \in \mathcal{U} \quad (2.15)$$

para algún $\epsilon > 0$ deseado y con una norma apropiadamente definida en el espacio de salida (denotada por $\|(\cdot)\|$). En la ecuación (2.15) $\hat{P}(u) = \hat{y}$ denota la salida del modelo de identificación, de ahí que $\hat{y} - y \triangleq e$ es el error de salida generado por \hat{P} y la salida observada y .

En el campo de los sistemas lineales invariantes en el tiempo con parámetros desconocidos la obtención de modelos de identificación ha sido bastante bien estudiada, por ejemplo:

Sea

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \quad (2.16)$$

una planta SISO controlable y observable, la matriz A y los vectores B y C pueden ser elegidos de tal forma que la ecuación (2.15) puede ser escrita como, [21]:

$$y_p(k+1) = \sum \alpha_i y(k-i) + \sum \beta_j u(k-j) \quad (2.17)$$

donde α_i y β_j son parámetros constantes desconocidos. Esto implica que la salida al tiempo $k+1$ es una combinación de sus entradas y salidas pasadas.

2.3.2. Identificación de sistemas no-lineales

Como la continuación de un trabajo publicado previamente [11], desarrollamos el análisis de identificación de sistemas no-lineales por medio de redes neuronales dinámicas. Consideramos la siguiente estructura para la red neuronal:

$$\begin{aligned} \dot{x}_{nn} &= A_{nn}x_{nn} + B_{nn}\sigma(x_{nn}) + C_{nn}\phi(x_{nn})\gamma(u_{nn}) & (2.18) \\ x_{nn} &\in \mathbb{R}^n, u_{nn} \in \mathbb{R}^m, \sigma(x_{nn}) \in \mathbb{R}^n, \phi(x_{nn})\gamma(u_{nn}) \in \mathbb{R}^n \\ A_{nn}, B_{nn} &\in \mathbb{R}^{n \times n}, C_{nn} \in \mathbb{R}^{n \times n} \end{aligned}$$

donde: x_{nn} es el estado de la red neuronal, u_{nn} es la entrada de la red neuronal, A_{nn} es la matriz de estado de retroalimentación, B_{nn} es la matriz no lineal de retroalimentación, C_{nn} es la matriz de entrada, (\cdot) , (\cdot) son funciones sigmoideas, (\cdot) es una función no lineal acotada, y nn son niveles de los parámetros de la red neuronal. El sistema no-lineal es definido como:

$$\dot{x} = f(x, u, t), \quad x \in \mathbb{R}^n, u \in \mathbb{R}^m \quad (2.19)$$

donde: x es el estado del sistema, u es la entrada, f es una función vectorial valuada, y t es el tiempo.

2.3.3. El modelo de estado-salida

Es bien conocido en la teoría de sistemas que el modelo de *estado-salida*, el cual relaciona los estados pasados y presentes, puede representar una pequeña clase de sistemas no lineales dinámicos.

El modelo estado-salida está dado por:

$$\begin{aligned} x_k &= f(x_{p_{k-1}}, u_{k-1}) \\ y_k &= g(x_{p_k}, u_k) \end{aligned} \quad (2.20)$$

donde u_k es la entrada, x_k es el estado del modelo, y_k es la salida del modelo, x_{p_k} estado de la planta, y_{p_k} es la salida de la planta, y k es el tiempo discreto. Las funciones $f(\cdot)$, $g(\cdot)$

son estáticas y por lo tanto pueden ser aproximadas usando redes neuronales feedforward (hacia adelante). Si todos los estados de la planta son medibles junto con su salida, entonces el problema de encontrar $f(\cdot), g(\cdot)$ mediante aprendizaje puede ser desacoplado.

2.3.4. El modelo NARMA

Aún a pesar de que el modelo estado-salida en general, es bastante real pensar que no todos los estados de las plantas son medibles. En este caso cuando una extensión del modelo ARMA (Auto Regresive Moving Average) es usada y se llama "Nonlinear ARMA model", el cual predice la salida presente como una función de las entradas y salidas pasadas:

$$y_k = f(y_{p_{k-1}}, y_{p_{k-2}}, \dots, y_{p_{k-n}}, u_k, u_{k-1}, \dots, u_{k-m}, W_{f_k}) \quad (2.21)$$

Donde y_k es la salida del modelo, y_{p_k} es la salida de la planta, W_{f_k} son los pesos de la red neuronal aunque comunmente llamado vector de coeficientes (se use o no una red neuronal), $y(y_{p_{k-1}}, y_{p_{k-2}}, \dots, y_{p_{k-n}}, u_k, u_{k-1}, \dots, u_{k-m})$ se llama vector regresor, y k es el tiempo discreto. Debido a que no hay recurrencia, ya que la salida del modelo depende solamente de la historia de la salida de la planta es por ello que se le llama modelo estático, y se le cita en la literatura predictor o modelo serie-paralelo.

La capacidad de representación del modelo *NARMA* es solo un subconjunto del modelo estado-salida. Esto es debido al hecho de que el modelo *NARMA* no almacena ninguna información del estado (memoria) y utiliza sólo entradas y salidas pasadas para reconstruir los estados. Si esa reconstrucción no es posible entonces el modelo *NARMA* no será capaz de representar el sistema. Más precisamente, el modelo *NARMA* trabaja solo cuando la función $g(\cdot)$ de $y_k = g(x_{p_k}, u_k)$ es invertible con respecto a los estados x_k . Cuando $g(\cdot)$ no es invertible, entonces hay una ambigüedad en reconstruir los estados a partir de las salidas.

2.3.5. Modelo de Estados Recurrentes

Esta es la versión recurrente del modelo *estado-salida*. Este modelo puede representar cualquier sistema dinámico no lineal, pero aprendiendo con la regla delta generalizada (back-

propagation o retroalimentación), la cual es debida a la recursión. Explícitamente mostrando la parametrización de las no linealidades $f(\cdot)$ y $g(\cdot)$ en términos de sus respectivos pesos de la red neuronal W_f, W_g , por lo que el modelo recurrente de la red neuronal se puede escribir como:

$$\begin{aligned}x_k &= f(x_{k-1}, u_{k-1}, W_{f_k}) \\y_k &= g(x_k, u_k, W_{g_k})\end{aligned}\tag{2.22}$$

En este punto debemos señalar que si solo la información de salida esta disponible, entonces la dinámica y las no linealidades del sistema no siempre pueden ser desacopladas. Esto nos fuerza a tener algún tipo de recurrencia en nuestros modelos.

Capítulo 3

Teoría de Estabilidad de Entrada-Estado (ISS)

La ”*Estabilidad de Entrada – Estado (ISS)*” es otro elegante enfoque para analizar estabilidad comparado con el método de Lyapunov. Puede conducir a conclusiones en general sobre estabilidad usando solo características de entrada y estado. Usaremos el enfoque o aproximación de *Estabilidad de Entrada-Estado (ISS)* para obtener algunas nuevas leyes de aprendizaje que no requieran modificaciones robustas.

Por otra parte pero dentro del mismo contexto, la identificación y el control de sistemas, son temas que han adquirido gran importancia y atención en ingeniería, especialmente en lo que a aspectos de carácter no lineal se refiere. Debido a la complejidad y variedad de sistemas no lineales, esto es, sistemas con diferentes tipos de no linealidades e incertidumbres paramétricas, no paramétricas o mixtas, dinámicas no modeladas, perturbaciones externas, etc., no existe una sola metodología para resolver los problemas de estimación-identificación y control para ciertas clases de sistemas. Por lo que en esta tesis, se planea utilizar las técnicas de redes neuronales con el enfoque o aproximación de *Estabilidad de Entrada – Estado* para tratar este tipo de sistemas dinámicos. Se utilizarán las técnicas tanto en tiempo-continuo como en tiempo-discreto; aunque los resultados existentes son en su mayoría para sistemas en tiempo continuo el diseño de identificadores y controladores en tiempo-discreto

tienen la importante ventaja de que casi directamente se pueden implementar en forma digital en modernos microcontroladores y sistemas de cómputo, aunque el diseño en tiempo discreto es bastante más complejo que el correspondiente en tiempo continuo. Esto se ve en la complejidad de las pruebas matemáticas de estabilidad y convergencia.

3.1. Teoría de Estabilidad (Fundamentos)

En el lenguaje común, la palabra estable (*del latín stabilis*) significa constante, permanente. Como concepto físico se introdujo en la mecánica, donde se utilizó (y se utiliza) aplicado sobre todo a una posición de equilibrio de una partícula, cuerpo o sistema mecánico. Tal posición de equilibrio se denomina estable si el cuerpo o partícula retorna a esa posición original después de moverlo separándolo ligeramente de la misma.

Así, se nos enseña en la escuela elemental que un cuerpo con forma de libro suficientemente delgado, tiene una posición "más estable" si se coloca sobre la cubierta que si se para sobre el canto. Esta conclusión se relaciona con el llamado Principio de Torricelli (aunque en realidad es anterior a él), una versión del cual podría ser: "En todo sistema de cuerpos sólidos en equilibrio (estable) el centro de gravedad ocupa la posición relativa más baja posible". Si un punto de un cuerpo rígido está fijo (piénsese por ejemplo en un péndulo) y el cuerpo está sujeto a la acción de la gravedad, se puede dar, en concordancia con el principio de Torricelli, el siguiente criterio para la estabilidad de los equilibrios: hay estabilidad si el centro de gravedad queda por debajo del punto fijo. Naturalmente, el criterio anterior no se aplica si no existen puntos fijos; tal es el caso de una bola que se desplaza sobre una superficie plana horizontal. Sin embargo, para una bola que se desplaza sobre una superficie alabeada, con puntos de equilibrio, funciona una variante del principio de torricelli, que puede enunciarse así: si cualquier desplazamiento pequeño a partir de la posición de equilibrio hace que el centro de gravedad suba, el equilibrio es estable (figura 3.1). La estabilidad es una noción que describe si un sistema es capaz de seguir el comando de entrada, o en general, si dicho sistema es útil. En una forma rigurosa, un sistema se dice inestable si sus salidas salen de control. Al diseñar un sistema de control, debemos ser capaces de predecir su

comportamiento dinámico a partir del conocimiento de los componentes. La característica más importante del comportamiento dinámico de un sistema de control es la estabilidad absoluta, es decir, si el sistema es estable o inestable. Un sistema de control está en equilibrio si, en ausencia de cualquier perturbación o entrada, la salida permanece en el mismo estado. Un sistema de control lineal e invariante en el tiempo es estable si la salida termina por regresar a su estado de equilibrio cuando el sistema está sometido o influenciado a una condición inicial. En realidad, la salida de un sistema físico puede aumentar hasta un cierto grado, pero puede estar limitada por detenciones mecánicas o el sistema puede colapsarse o volverse no lineal después de que la salida excede cierta magnitud, por lo cual ya no se aplican las ecuaciones diferenciales lineales. Entre los comportamientos importantes del sistema (aparte de la estabilidad absoluta) que deben recibir una cuidadosa consideración están la estabilidad relativa y el error en estado estable. Dado que un sistema de control físico implica un almacenamiento de energía, la salida del sistema, cuando éste está sujeto a una entrada, no sucede a la entrada de inmediato, sino que recibe una respuesta transitoria de un sistema de control práctico con frecuencia exhibe oscilaciones amortiguadas antes de alcanzar un estado estable. Si la salida de un sistema en estado estable no coincide exactamente con la entrada, se dice que el sistema tiene un error en estado estable. Este error indica la precisión del sistema. Al analizar un sistema de control, debemos examinar el comportamiento de la respuesta transitoria y el comportamiento en estado estable. En un sistema de control determinado, por lo general, el aspecto cuya determinación es más importante es la estabilidad. Si el sistema es lineal e invariante con el tiempo, existen varios criterios de estabilidad. Entre ellos están el criterio de estabilidad de *Nyquist* y el criterio de estabilidad de *Routh*. Sin embargo, si el sistema es no lineal o lineal pero variante con el tiempo, tales criterios de estabilidad no se aplican por resultar muy difíciles o imposibles de aplicar. El análisis de estabilidad de Lyapunov es un método que se aplica para encontrar respuestas a las preguntas sobre la estabilidad de los sistemas no lineales.

Aunque el análisis de estabilidad de Lyapunov es muy útil y poderoso para abordar los problemas de estabilidad de los sistemas no lineales, no es sencillo determinar la estabilidad de muchos sistemas no lineales. El ingenio y la experiencia son muy importantes para resolver

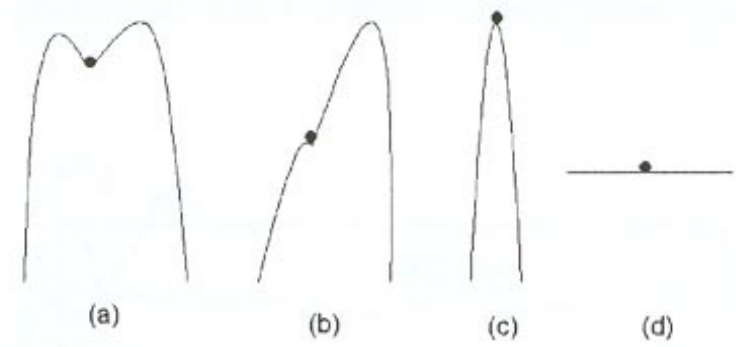


Figura 3.1: a) y b) Asintóticamente estables, c) Inestable, d) Estable.

problemas no lineales.

Muchos eminentes matemáticos y físicos del siglo XIX se ocuparon de cuestiones de estabilidad, entre otros, Lagrange, Kelvin, Routh, Shukovskii y Poncaré. No obstante, un tratamiento general de la estabilidad del movimiento (en su forma clásica) completamente satisfactorio sólo fué dado por A. M. Lyapunov en 1892, en su famosa monografía "*Problema general de la estabilidad del movimiento*", que fue su tesis doctoral [86]. Es muy importante en el trabajo de Lyapunov la propia definición de estabilidad, que se introduce por primera vez con rigor matemático, y que es mucho más amplia que el concepto que se utilizaba en mecánica; no se refiere ya al movimiento de un cuerpo material, sino en general, a una ecuación diferencial. Además no se trata de la estabilidad de un equilibrio sino la de una solución cualquiera de la ecuación. (Los puntos de equilibrio o estacionarios de la ecuación son soluciones particulares de la misma, y representan los puntos de equilibrio físico del cuerpo cuyo movimiento se describe).

Después de los conceptos de estabilidad y estabilidad asintótica introducidos por Lyapunov, aparecen los de estabilidad uniforme, asintótica uniforme, global, equiasintótica, exponencial, cuasi-asintótica y muchas otras, así como sus combinaciones. Por ejemplo, la estabil-

idad asintótica uniforme global o la cuasi-uniformemente asintótica. Además, se consideran la estabilidad según Poisson y según Lagrange (esta última no es más que la acotación), también con diversas variantes. Para analizar los movimientos periódicos se introduce la estabilidad orbital (Poincaré). Téngase en cuenta que estas definiciones corresponden a cierta *clase* de estabilidad: la estabilidad respecto a perturbaciones instantáneas. En cuanto a perturbaciones permanentes o de acción continuada (Doboshin) existen la estabilidad total, la integral, la estabilidad en media, etc. También se han introducido la estabilidad eventual, la estabilidad práctica, los distintos tipos de estabilidad en variaciones, la estabilidad absoluta (de uso en las ecuaciones que describen procesos controlables), la estabilidad de conjuntos, la parcial, la relativa, la estabilidad condicional, los conceptos en espacios L_p , etc.

Como se mencionó anteriormente en 1892, A.M. Lyapunov presentó dos métodos (llamados el primero y el segundo) para determinar la estabilidad de los sistemas dinámicos descritos mediante ecuaciones diferenciales ordinarias. El primer método se compone de todos los procedimientos en los cuales se usa la forma explícita de la solución de las ecuaciones diferenciales para el análisis. En cambio, el segundo método no requiere de las soluciones de las ecuaciones diferenciales. Es decir, mediante el segundo método de Lyapunov, se determina la estabilidad de un sistema sin resolver las ecuaciones de estado. Esto ofrece una gran ventaja porque, por lo general, es muy difícil despejar las ecuaciones de estado no lineales y/o variantes en el tiempo. Aunque el segundo método de Lyapunov, cuando se aplica al análisis de estabilidad de los sistemas no lineales requiere de mucha experiencia e ingenio, contesta a la pregunta de la estabilidad de los sistemas no lineales cuando otros métodos fracasan.

3.1.1. Conceptos Básicos

Sea la función $f : \Omega \times I \rightarrow \mathbb{R}^n$, donde $I \subset (a, +\infty)$ y $a \in \mathbb{R} \cup \{-\infty\}$. $\Omega \subseteq \mathbb{R}^n$, es un conjunto abierto conexo. Sea

$$\dot{x} = f(x, t) \tag{3.1}$$

un sistema de ecuaciones diferenciales ordinarias (EDO), donde:

$$\dot{x} = \frac{dx}{dt}, \quad f$$

es continua y suficientemente suave, de modo que se pueda asegurar la existencia y la unicidad de la solución del problema de Cauchy por cada punto (x_0, t_0) . Eventualmente puede ser $\Omega = \mathbb{R}^n$; la variable t se denomina *tiempo*. Es conocida la siguiente propiedad, llamada continuidad integral, que será muy utilizada. Como se podrá apreciar más adelante, es algo más débil que la estabilidad. *La propiedad es la siguiente:* Si se satisfacen condiciones suficientes para la existencia y unicidad de las soluciones del problema de Cauchy, entonces, para cualquier solución $x^*(t)$ del sistema (3.1), definida para $t \geq 0$:

Dados $\epsilon > 0$ y $T \geq t_0, \exists \delta > 0$ tal que si $x(t)$ es cualquier otra solución y $|x(t_0) - x^*(t_0)| < \delta$, entonces $|x(t) - x^*(t)| < \epsilon$ para $t \in [0, T]$.

Definición 1 Sea $x(t)$ una solución de (3.1). Se dice que es estable (según Lyapunov) en $t = t_0$ si dado un $\epsilon > 0$, existe $\delta > 0$ tal que si $y(t)$ es cualquier otra solución con $|x(t_0) - y(t_0)| < \delta$, entonces $|x(t) - y(t)| < \epsilon$ para $t \geq t_0$. En caso contrario, $x(t)$ es inestable.

Denotamos por $x(t) \equiv x(t, t_0, x_0)$ la solución que satisface $x(t_0, t_0, x_0) = x_0$. Si t_0 es conocido, pondremos más simplemente $x(t, t_0, x_0) = x(t, x_0)$, y también $x(t, x_0) \equiv x(t)$ si se conoce x_0 sin ambigüedad.

Consideremos el operador $L : \Omega \rightarrow C([t_0, +\infty) \times \mathbb{R}^n)$, definido mediante $L(x_0) = x(t, x_0)$. En el espacio de llegada se utiliza la métrica: $d(x(t), y(t)) = \sup\{|x(t) - y(t)| : t \geq t_0\}$ que es la métrica de convergencia uniforme respecto de $t \geq t_0$. La estabilidad de la solución $x(t)$ no es más que la continuidad del operador L en el punto $x(t_0) = x_0$.

3.2. ISS y funciones de Lyapunov-ISS

El objetivo fundamental de esta tesis es estudiar las propiedades de estabilidad de Entrada-Estado (ISS), para sistemas no lineales, tanto en tiempo-continuo como en tiempo-

discreto. Y el principal interés, es entender la dependencia de las trayectorias de estado sobre la magnitud de entradas por los sistemas del siguiente tipo:

$$x(k+1) = f(x(k), u(k)) \quad (3.20)$$

$$y(k) = h[x(k)]$$

Definición 2 *El sistema (3.20) es globalmente estable de entrada-estado (ISS) si existe una \mathcal{KL} -función $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ y una \mathbf{K} -función γ tal que, para cada entrada $u \in l_{\infty}^m$ y cada $\xi \in \mathbb{R}^n$, afirmando que :*

$$\|x(k, \xi, u)\| \leq \beta(\|\xi\|, k) + \gamma(\|u\|) \quad (3.21)$$

para cada $k \in \mathbb{Z}$. Observando que, por causalidad la misma definición resultaría si se reemplazara (3.21) por

$$\|x(k, \xi, u)\| \leq \beta(\|\xi\|, k) + \gamma(\|u_{[k-1]}\|) \quad (3.22)$$

para cada $k \in \mathbb{Z}_+ - \{0\}$. Recordando que $u_{[k-1]}$ denota el truncamiento de u en $k-1$. De la ec. (3.22) puede verse que la propiedad de ISS implica que el sistema de entrada 0, $x(k+1) = f(x(k), 0)$ es globalmente asintóticamente estable (GAS) y que (3.20) es convergente a la entrada a la entrada convergente de estado, es decir, para cada trayectoria $x(k, \xi, u)$ va a 0 cuando $k \rightarrow \infty$. Sin embargo, la conversión no es verdadera. Un ejemplo simple para considerar es $x(k+1) = \frac{1}{2}(1 + \sin u(k)x(k))$. No es difícil ver que el sistema es de entrada-0 GAS, y satisface la propiedad de convergencia de estado-convergencia de entrada. Pero el sistema no es ISS, porque para la función de entrada constante $u(k) = \frac{\pi}{2}$, la trayectoria $x(k, \xi, u)$ es idénticamente ξ , que viola cualquier estimación del tipo (3.21).

Definición 3 *Una función continua se llama una función de Lyapunov-ISS para el sistema (3.20) si las afirmaciones siguientes:*

1. Existen \mathbf{K}_{∞} -funciones α_1, α_2 tal que

$$\alpha_1(|\xi|) \leq V(\xi) \leq \alpha_2(|\xi|), \quad \forall \xi \in \mathbb{R}^n \quad (3.23)$$

2. Existe una **\mathcal{K}_∞ -función** α_3 y una **\mathcal{K} -función** σ tal que

$$V(f(\xi, \mu)) - V(\xi) \leq -\alpha_3(|\xi|) + \sigma(|\mu|), \quad \forall \xi \in \mathbb{R}^n, \forall \mu \in \mathbb{R}^n \quad (3.24)$$

Propiedad 1 Como en el caso en tiempo continuo. La propiedad 2 en la definición anterior es equivalente a la propiedad siguiente:

Existe alguna **\mathcal{K}_∞ -función** α_4 y alguna **\mathcal{K} -función** χ tal que

$$\{|\xi| \geq \chi(|\mu|)\} \implies \{V(f(\xi, \mu)) - V(\xi) \leq -\alpha_4(|\xi|)\} \quad (3.25)$$

Debe mencionarse que estos resultados es una propiedad equivalente si la función α_4 en (3.25) es simplemente necesario que sea continua y definida positiva. Ver Jiang & Wang (2001) para una prueba detallada.

Como una ilustración simple de estas nociones, nos profundizaremos un poco (3.20) para sistemas lineales en tiempo discreto:

$$x(k+1) = Ax(k) + Bu(k) \quad (3.26)$$

donde A es una matriz de *Schur*, es decir, los valores propios de A son estrictamente localizados dentro del disco unitario. Para semejante matriz, hay constantes $c > 0$ y $0 \leq \sigma < 1$ tal que $|A^k| \leq c\sigma^k$ (cf. LaSalle, 1986, Chapter 5). De (3.26), tenemos:

$$x(k+1) = A^{k+1}x(0) + \sum A^{k-j}Bu(j) \quad (3.27)$$

de que la propiedad ISS (3.21) sigue con:

$$\beta(r, k) = c\sigma^k r, \quad \gamma(r) = \sum c\sigma^j |B|r = \frac{c|B|r}{1-\sigma} \quad (3.28)$$

Más adelante, se mostrará que el sistema (3.26) tendrá una función de Lyapunov-ISS cuadrática. Dada una matriz Q simétrica, sea $P > 0$ la única solución a la ecuación matricial : $A^T P A - A = -Q$.

Considérese la función definida positiva y radialmente ilimitada:

$$V(x) = x^T P x \quad (3.29)$$

que satisface la propiedad (3.23) con $\alpha_1(r) = \lambda_{\min}(P)r^2$ y $\alpha_2(r) = \lambda_{\max}(P)r^2$.

Mostrando el cómputo directo:

$$V(x(k+1)) - V(x(k)) = -x^T(k) Q x(k) + 2x^T(k) A^T P B u(k) + u^T(k) B^T P B u(k) \quad (3.30)$$

Entonces, completando cuadrados, la propiedad (3.24) afirma o sostiene que:

$$\alpha_3(r) = \frac{1}{2} \lambda_{\min}(Q) r^2, \quad \sigma(r) = \left(\frac{2|A^T P B|^2}{\lambda_{\min}(Q)} + |B^T P B|^2 \right) r^2$$

Por consiguiente, V definida en (3.29) es una función de Lyapunov-ISS para (3.26).

Claramente, si V es una función de Lyapunov-ISS para (3.20), entonces V es una función de Lyapunov para el sistema de entrada-0 $x(k+1) = f(x(k), 0)$. Como en la teoría clásica de estabilidad de Lyapunov, podemos probar que un sistema es ISS si y sólo si admite una función de Lyapunov-ISS.

Lemma 1 *Si el sistema (3.20) admite una función de Lyapunov-ISS, entonces es ISS.*

Demostración. *Asumir que el sistema (3.20) admite una función de Lyapunov-ISS V . Sea α_i ($i = 1, 2, 3$) y σ como en (3.23) y (3.24).*

Primero observe que (3.24) puede volverse a escribir como:

$$V(f(\xi, \mu)) - V(\xi) \leq -\alpha_4(V(\xi)) + \sigma(|\mu|) \quad (3.31)$$

para todos los ξ y μ , donde $\alpha_4 = \alpha_3 \circ \alpha_2^{-1}$. Sin pérdida de generalidad, consideramos que $Id - \alpha_4 \in \mathbf{K}$ (cf. Lemma B1). Fijar un punto $\xi \in \mathbb{R}^n$ y escoger una entrada u . Sea

$x(k)$ la correspondiente trayectoria $x(k, \xi, u)$ de (3.20). Sea ρ cualquier \mathcal{K}_∞ -función tal que $Id - \rho \in \mathcal{K}_\infty$.

Considerar el conjunto definido por:

$$D = \{\xi : V(\xi) \leq b\}$$

donde $b = \alpha_4^{-1} \circ \rho^{-1} \circ \sigma(\|u\|)$. ■

Claim 2 Si existe algún $k_0 \in \mathbb{Z}_+$ tal que $x(k_0) \in D$, entonces $x(k) \in D$ para toda $k \geq k_0$.

La principal preocupación de esta sección es para entender algunos conceptos de *Estabilidad de Entrada-Estado (ISS)*. Considere el siguiente sistema no lineal en tiempo discreto

$$x(k+1) = f[x(k), u(k)], \quad y(k) = h[x(k)]$$

donde $u(k) \in \mathfrak{R}^m$ es el vector de entrada, $x(k) \in \mathfrak{R}^n$ es el vector de estado, y $y(k) \in \mathfrak{R}^l$ es el vector de salida. f y h son funciones suaves no lineales en general $f, h \in C^\infty$. Ahora es necesario recordar las siguientes definiciones.

Definición 4 Un sistema (3.20) se dice ser globalmente estable de entrada-estado si existe una \mathcal{K} -función $\gamma(\cdot)$ (continua y estrictamente creciente $\gamma(0) = 0$) y \mathcal{KL} -función $\beta(\cdot)$ (\mathcal{K} -función y $\lim_{s_k \rightarrow \infty} \beta(s_k) = 0$), tal que, para cada $u \in L_\infty$ ($\sup\{\|u(k)\|\} < \infty$) y cada estado inicial $x^0 \in \mathfrak{R}^n$, es sostenible que

$$\|x(k, x^0, u(k))\| \leq \beta(\|x^0\|, k) + \gamma(\|u(k)\|)$$

Definición 5 Una función suave $V : \mathfrak{R}^n \rightarrow \mathfrak{R} \geq 0$ es llamada una función suave de Lyapunov-ISS para el sistema (3.20) si: (a) existe una \mathcal{K}_∞ -función $\alpha_1(\cdot)$ y $\alpha_2(\cdot)$ (\mathcal{K} -función y $\lim_{s_k \rightarrow \infty} \beta(s_k) = \infty$) tal que

$$\alpha_1(s) \leq V(s) \leq \alpha_2(s), \quad \forall s \in \mathfrak{R}^n$$

(b) Existe una \mathcal{K}_∞ -función $\alpha_3(\cdot)$ y una \mathcal{K} -función $\alpha_4(\cdot)$ tal que

$$V_{k+1} - V_k \leq -\alpha_3(\|x(k)\|) + \alpha_4(\|u(k)\|), \quad \text{para toda } x(k) \in \mathfrak{R}^n, u(k) \in \mathfrak{R}^m$$

Teorema 3 *Para un sistema no lineal en tiempo discreto, los siguientes son equivalentes cite:Jiang*

- *Es estable de entrada-estado (ISS).*
- *Es robustamente estable.*
- *Admite una función suave de Lyapunov-ISS .*

Property. Si un sistema no lineal es *estable de entrada-estado (ISS)*, el comportamiento del sistema permanecerá acotado cuando las entradas son acotadas.

De (3.20) tenemos

$$\begin{aligned} y(k) &= h[x(k)] := F_1[x(k)], & y(k+1) &= h[f[x(k), u(k)]] := F_2[x(k), u(k)] \\ y(k+n-1) &:= F_n[x(k), u(k), u(k+1), \dots, u(k+n-2)] \end{aligned} \quad (3.32)$$

Denotando $Y(k) = [y(k), y(k+1), \dots, y(k+n-1)]^T$, $U(k) = [u(k), u(k+1), \dots, u(k+n-2)]^T$, así que $Y(k) = F[x(k), U(k)]$, $F = [F_1 \dots F_n]^T$. Si $\frac{\partial Y}{\partial x}$ es no singular en $x = 0$, $U = 0$, (3.32) puede ser expresada como $x(k+1) = g[Y(k+1), U(k+1)]$. Esto conduce al modelo NARMA

$$y(k) = h[x(k)] = \Phi[y(k-1), y(k-2), \dots, u(k-1), u(k-2), \dots] = \Phi[X(k)]$$

donde

$$X(k) = [y(k-1), y(k-2), \dots, u(k-d), u(k-d-1), \dots]^T$$

$\Phi(\cdot)$ es una ecuación en diferencias no lineal desconocida que representa la planta dinámica, $u(k)$ y $y(k)$ son escalares de entrada y salida medibles, d es el tiempo de retardo. Uno puede ver que la *Definición 1,2* y el *Teorema 1* no dependen sobre la expresión exacta de sistemas no lineales. En esta tesis, aplicaremos ISS al modelo NARMA (4.1).

Definición 6 *El grado relativo de un sistema no lineal en tiempo discreto : Sea f_e denota $f[x(k), 0]$, f_e^i la composición iterada de tiempos- i . \sum tendrá un grado relativo r en (x, u) identificando si existe una vecindad del estado de equilibrio $(0,0)$ para que las siguientes*

condiciones sean satisfechas : $\frac{\partial(h \circ f_e^k \circ f)}{\partial u} = 0$ para $0 \leq k \leq r - 2$, $\frac{\partial(h \circ f_e^{r-1} \circ f)}{\partial u} \neq 0$. Si el retardo entre la entrada y la salida esta definida como d , Σ es un sistema analítico , tenemos $r = d$ siempre que r esta bien definido.

Definición 7 *Dinámica cero de un sistema no lineal en tiempo discreto: Se tiene Σ de grado relativo r en $(0, 0)$. Entonces existe un cambio analítico de coordenadas $z = \Phi(x)$ con $\Phi(0) = 0$ tal que en coordenadas $-z$, el sistema se tornará normal para Σ_n*

$$\begin{aligned} z_i(k+1) &= z_{i+1}(k), \quad i \leq r-1 \\ z_r(k+1) &= F(z(k), \eta(k), u(k)) \\ \eta(k+1) &= D(z(k), \eta(k), u(k)) \\ y(k) &= z_1(k) \end{aligned}$$

donde $z(k) = [z_1(k) \cdots z_r(k)]$, $\eta(k) = [z_{r+1}(k) \cdots z_n(k)]$. Dado un estado inicial $x(0)$, asumimos que podemos encontrar un control u tal que la salida es cero. Así que $\eta(k+1) = D(0, \eta(k), u(k))$, $u(k)$ es solamente una función de $\eta(k)$, como $u(k) = g(\eta(k))$. El sistema dinámico autónomo

$$\eta(k+1) = D(0, \eta(k), g(\eta(k)))$$

es llamado *dinámica cero del sistema Σ_n* .

3.3. Teoría ISS en Tiempo Continuo

Consideremos el sistema como un modelo de Espacio-Estado de la forma general:

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), u(t)) \\ y(t) &= h(t, x(t), u(t)) \end{aligned} \tag{3.3}$$

donde u, y & x son la entrada, salida y variables de estado, respectivamente.

En el enfoque de espacio de estado, pondremos mucho énfasis en el comportamiento de las variables de estado. Esto es verdad para la mayor parte de nuestro análisis, por ahora, para resolver la ecuación de estado, y la mayoría del tiempo sin una presencia explícita

de una entrada externa u . Un enfoque alternativo para el modelo matemático de sistemas dinámicos es el enfoque Entrada-Salida. Un modelo Entrada-Estado relaciona la salida del sistema directamente con la entrada, sin conocimiento interno de la estructura del sistema que es representada por la ecuación de Estado. El sistema es visto como una caja negra que puede ser accesada solo a través de sus terminales de entrada y salida.

Del sistema (3.3) $f : [0, \infty) \times D \times D_u \rightarrow \mathbb{R}^n$ es seccionalmente continua en t y localmente *Lipschitz* en x & u , $D \subset \mathbb{R}^n$ es un dominio que contiene a $x = 0$, & $D_u \subset \mathbb{R}^n$ es un dominio que contiene a $u = 0$. La entrada $u(t)$ es una función acotada y seccionalmente continua de t para todo $t \geq 0$.

Supongamos que el sistema no forzado:

$$\dot{x}(t) = f(t, x(t), 0) \quad (3.4)$$

tiene un punto de equilibrio uniformemente asintóticamente estable (*UAE*) en el origen $x = 0$.

Una forma de analizar el comportamiento Entrada-Estado del sistema forzado (3.3), es considerar a (3.3) como una perturbación del sistema no forzado (3.4), donde el término de perturbación es:

$$f(t, x, u) - f(t, x, 0) \quad (3.5)$$

Esto lo vamos a realizar para la propiedad de *Estabilidad de Entrada-Estado (Input-to-state stability, ISS)* definida a continuación.

Definición 8 (*Estabilidad de Entrada-Estado*). *El sistema (3.3) es localmente Estable de Entrada-Estado si existe una función β de clase \mathbf{KL} , una función γ de clase \mathbf{K} , y constantes positivas K_1 & K_2 tales que, para cualquier estado inicial $x(t_0)$ con $\|x(t_0)\| < K_1$ y cualquier entrada $u(t)$ con $\sup_{t \geq t_0} \|u(t)\| < K_2$, la solución $x(t)$ existe y satisface:*

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0) + \gamma\left(\sup_{t_0 \leq \tau \leq t} \|u(\tau)\|\right) \quad (3.6)$$

para todo $t \geq t_0 \geq 0$. Es *Estable de Entrada-Estado* si $D = \mathbb{R}^n$, $D_u = \mathbb{R}^m$, y la desigualdad o ec. (3.6) se satisface para cualquier estado inicial $x(t_0)$ y cualquier entrada acotada $u(t)$.

La desigualdad (3.6) garantiza que para cualquier entrada acotada $u(t)$, el estado $x(t)$ se va a mantener acotado. Más aún, cuando t aumenta, el estado $x(t)$ va a tener una cota final que es una función clase \mathbf{K} de

$\sup_{t > t_0} \|u(t)\|$. Puede probarse que si $u(t)$ converge a cero cuando $t \rightarrow \infty$, también lo hace $x(t)$. Como, con $u(t) = 0$, (3.3) se reduce a :

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0) \quad (3.7)$$

La *estabilidad de entrada-estado* (ISS) local implica que el origen del sistema no forzado es uniformemente asintóticamente estable (*UAE*), mientras que ISS implica que es uniformemente globalmente asintóticamente estable (*GUAE*).

El siguiente teorema de Lyapunov da una condición suficiente para estabilidad de entrada-estado.

Teorema. Sea $D = \{x \in \mathbb{R}^n \mid \|x\| < r\}$, $D_u = \{u \in \mathbb{R}^m \mid \|u\| < r_u\}$, & $f : [0, \infty) \times D \times D_u \rightarrow \mathbb{R}^n$

seccionalmente continua en t y localmente Lipschitz en x y u . Sea $V : [0, \infty) \times D \rightarrow \mathbb{R}$ una función continuamente diferenciable tal que:

$$\alpha_1(\|x\|) \leq V(t, x) \leq \alpha_2(\|x\|) \quad (3.8)$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x, u) \leq -\alpha_3(\|x\|)$$

$\forall \|x\| \geq \rho(\|u\|) > 0, \quad \forall (t, x, u) \in [0, \infty) \times D \times D_u$, donde α_1, α_2 & β son funciones clase \mathbf{K} . Entonces el sistema (3.3) es localmente ISS con $\gamma = \alpha_1^{-1} \circ \alpha_2 \circ \rho$, $K_1 = \alpha_2^{-1}(\alpha_1(r))$ & $K_2 = \rho^{-1}(\min\{K_1, \rho(r_u)\})$. Más aún, si $D = \mathbb{R}^n, D_u = \mathbb{R}^m$, & α_1 es clase K_∞ , entonces (3.3) es ISS con $\gamma = \alpha_1^{-1} \circ \alpha_2 \circ \rho$.

Los siguientes lemas son consecuencia de los teoremas conversos de Lyapunov:

Lemma 4 I.1 *Supongamos que en algún entorno de $(x, u) = (0, 0)$, la función $f(t, x(t), u(t))$ es continuamente diferenciable y las matrices jacobianas $\left[\frac{\partial f}{\partial x}\right]$ y $\left[\frac{\partial f}{\partial u}\right]$ están acotadas, uniformemente en t . Si el sistema no forzado de la ec. (3.4) tiene un punto de equilibrio PE UAE en el origen $x = 0$, entonces el sistema (3.4) es localmente ISS.*

Demostración. Por el teorema (converso de Lyapunov), el sistema no forzado (3.4) tiene una función de Lyapunov $V(t, x)$ que satisface las desigualdades (3.6) en algún entorno acotado de $x = 0$. Como $\left[\frac{\partial f}{\partial u}\right]$ está acotada, el término de perturbación satisface:

$$\|f(t, x(t), u(t)) - f(t, x, 0)\| \leq L\|u\|, \quad L > 0$$

para todo $t \geq t_0$ y todo (x, u) en algún entorno acotado de $(x, u) = (0, 0)$. Puede verificarse que $V(t, x)$ satisface las condiciones del teorema en algún entorno acotado de $(x, u) = (0, 0)$. Para sistemas estacionarios, las hipótesis del lema I.1 de que los jacobianos estén acotados se satisfacen trivialmente si $f(x, u)$ es continuamente diferenciable. Por lo tanto, para sistemas estacionarios el lema dice que si $f(x, u)$ es continuamente diferenciable y el origen de (3.4) es asintóticamente estable AE, entonces (3.3) es localmente ISS. ■

Lemma 5 *Supongamos que $f(t, x(t), u(t))$ es continuamente diferenciable y globalmente Lipschitz en (x, u) , uniformemente en t . Si el sistema no forzado (3.4) tiene un punto de equilibrio PE globalmente exponencialmente estable en el origen $x = 0$, entonces el sistema (3.3) es ISS.*

Demostración. Por el teorema (converso de Lyapunov), el sistema no forzado (3.4) tiene una función de Lyapunov $V(t, x)$ que satisface las desigualdades (3.6) globalmente. ■

Como f es globalmente Lipschitz en (x, u) , el término de perturbación satisface a (3.4) para todo $t \geq t_0$ y todo (x, u) . Puede verificarse que $V(t, x)$ satisface las condiciones del lema I.1 globalmente.

Si el origen del sistema no forzado (3.4) es GUAE pero no globalmente exponencialmente estable, el sistema (3.3) no es necesariamente ISS incluso cuando f es globalmente Lipschitz en (x, u) .

El sistema no lineal está definido como:

$$\dot{x} = f(x, u, t), x \in \mathbb{R}^n, u \in \mathbb{R}^m \quad (3.9)$$

donde: x es el estado del sistema, u es la entrada, f es una función vectorial estimada, y t es el tiempo, con A_{nn} Hurwitz, $f(0) = 0$, y $(\cdot), (\cdot), (u)$ globalmente Lipschitz:

$$\begin{aligned} \|\sigma(\Delta_x - x) - \sigma(x)\| &\leq L_\sigma \|\Delta_x\| \\ \|\phi(\Delta_x - x) - \phi(x)\| &\leq L_\phi \|\Delta_x\| \\ \|\Gamma(\Delta_u - u) - \Gamma(u)\| &\leq L_\Gamma \|\Delta_u\| \\ \forall x, \Delta_x \in \mathbb{R}^n, \forall u, \Delta_u \in \mathbb{R}^m \end{aligned} \quad (3.10)$$

Siguiendo el mismo procedimiento como en [17], es trivial demostrar que (2.18) presenta al menos un punto de equilibrio [18].

Dando x^* como un punto de equilibrio de (2.18) con $u = 0$, definimos $y = x - x^*$ entonces:

$$\dot{y} = \dot{x} = A_{nn}x + B_{nn}(x) + C_{nn}(x)(u) \quad (3.11)$$

Sumando y sustrayendo $B_{nn}(x^*) + C_{nn}(x^*)(u)$ a (3.11), y tomando en cuenta que $y = x - x^*$, obtenemos:

$$\begin{aligned} \dot{y} &= A_{nn}y + B_{nn}[(y + x^*) - (x^*)] \\ &\quad + C_{nn}[(y + x^*) - (x^*)](u) \\ &\quad + C_{nn}(x^*)[(u) - (0)] \end{aligned} \quad (3.12)$$

Una vez definido este modelo, procedemos con el análisis ISS; usamos las siguientes definiciones.

El siguiente teorema tipo Lyapunov da una condición suficiente para ISS.

Definición 9 (*Función de clase K_∞*)[19].

(Función de clase \mathbf{K}_∞)

Una función continua $f : [0, a) \rightarrow [0, \infty)$ se dice que pertenece a la clase \mathbf{K} si es estrictamente creciente y $f(0) = 0$; y se dice que pertenece a la clase \mathbf{K}_∞ si $a = \infty$ y $\lim_{r \rightarrow \infty} f(r) = \infty$.

Definición 10 (Función de Lyapunov ISS)[7].

Una función $V(x) : \mathbb{R}^n \rightarrow \mathbb{R}^+$ es definida como de Lyapunov-ISS si existe (r) , (r) es una función de clase \mathbf{K}_∞ tal que: $\dot{V}(x) \leq -(\|x\|) + (\|u\|)$, $\Delta x \in \mathbb{R}^n, \forall u \in \mathbb{R}^m$.

Teorema 6 *El sistema (3.10), o el sistema equivalente (2.18) es ISS si:*

$$\|B_{nn}\|^2 < \frac{-3\|P\|}{L^2\|P\|}, \|P\| < \frac{1}{3}, \quad \in \mathbb{R}^+, P = P^T > 0 \quad (3.13)$$

donde P es solución de la ecuación de Lyapunov $A_{nn}^T P + P A_{nn} = -I$. Tal solución unica existe porque A_{nn} es Hurwitz.

Demostración. Consideremos la siguiente función:

$$V(y) = y^T P y, \quad P = P^T > 0 \quad (3.14)$$

Su tiempo deriva hacia las trayectorias de (3.11) es dado como:

$$\begin{aligned} \dot{V} &= -y^T y + 2y^T P B_{nn} [(y + x^*) - (x^*)] \\ &\quad + 2y^T P C_{nn} [(y + x^*) - (x^*)] (u) \\ &\quad + 2y^T P C_{nn} (x^*) [(u) - (0)] \end{aligned} \quad (3.15)$$

■

Analizamos la segunda parte derecha, tercer y cuarto término de (3.15) y usando la desigualdad matricial [11]: $X^T Y + Y^T X \leq X^T X + Y^T Y$, con $X = P$. Tomando en cuenta que $V(x)$ es globalmente Lipschitz, entonces es uniformemente acotado, i.e:

$$\| (y + x^*) - (x^*) \| \leq K, \quad \forall y \in \mathbb{R}^n, \text{ finalmente obtenemos:}$$

$$\begin{aligned}
V &\leq (-3\|P\| - L^2\|P\|\|B_{nn}\|^2)\|y\|^2 + \\
&(K^2L^2\|P\|\|C_{nn}\|^2 + L^2\|P\|\|C_{nn}\|^2\|(x^*)\|^2)\|u\|^2
\end{aligned} \tag{3.16}$$

Definiendo las siguientes funciones:

$$\begin{aligned}
(r) &= (-3\|P\| - L^2\|P\|\|B_{nn}\|^2)r^2 \\
(r) &= (K^2L^2\|P\|\|C_{nn}\|^2 + L^2\|P\|\|C_{nn}\|^2\|(x^*)\|^2)r^2 \\
r &\in \mathbb{R}
\end{aligned} \tag{3.17}$$

entonces:

$$\dot{V} \leq (\|x\|) + (\|u\|) \tag{3.18}$$

V será una función de Lyapunov-ISS, si $x(\cdot)$ y $u(\cdot)$ son funciones de clase \mathbf{K}_∞ . Como se definió $x(\cdot)$ satisface esta condición; por lo tanto, para el sistema (3.11)(equivalentemente a (2.18)) es ISS, es necesario que:

$$(-3\|\mathbf{P}\| - L^2\|\mathbf{P}\|\|\mathbf{B}_{nn}\|^2) > 0 \tag{3.19}$$

que implica que:

$$\|\mathbf{B}_{nn}\|^2 < \frac{-3\|\mathbf{P}\|}{L^2\|\mathbf{P}\|}, \quad \|\mathbf{P}\| < \frac{1}{3} \tag{3.20}$$

Corolario 7 *Para el punto de equilibrio $y = 0$ de (3.11) (equivalentemente a $x = x^*$ de (2.18)) es estable asintóticamente global (GAS), entonces (3.20) ha sido comprobada.*

Demostración. Se sigue directamente de (3.18), con $u = 0$. ■

La demostración detallada es discutida en [18,20].

Propiedad 2 *Los resultados de ISS para redes neuronales dinámicas es muy importante; permite utilizar este tipo de redes neuronales para nuevos esquemas de control no-lineal.*

3.4. Teoría ISS en tiempo discreto

La noción de Estabilidad de Entrada de Estado (ISS) para sistemas de control no lineal fue propuesta por: (Sontag en 1989 y 1990) y ha sido usada en el análisis de estabilidad y síntesis de control de sistemas no lineales por diversos investigadores. Básicamente, *ISS* da una cota cuantitativa de trayectorias de estado en términos de la magnitud de la entrada de control y sus condiciones iniciales. Mostraremos que muchos resultados *ISS* para sistemas no lineales en tiempo continuo puede ser extendido al caso en tiempo discreto. A través de estas extensiones, probaremos que varias características equivalentes de la condición *ISS* propuesta en (Sontag y Wang, 1996) también son retenidas para sistemas no lineales en tiempo discreto. También se muestra que los teoremas *ISS* de ganancia pequeña son extendibles a la posición en tiempo discreto. Como siempre, nuevos fenómenos surgen en la extensión de tiempo continuo a tiempo discreto. Para sistemas afines en tiempo continuo, estabilización continua implica estabilización *ISS* por medio del cambio de retroalimentación de estado $u = K(x) + v$. Para sistemas afines en tiempo discreto o sistemas no afines, una transformación de retroalimentación más compleja de la forma $u = K_1(x) + K_2(x)v$ es en general requerida. La construcción de los términos de retroalimentación K_1 y K_2 resulta no trivial; sin embargo, mostraremos que ellos pueden ser explícitamente obtenidos, para una clase de retroalimentación de sistemas linealizables.

3.4.1. Notación y definiciones

Se usará \mathbb{Z}_+ para denotar el conjunto de todos los enteros no negativos. Para cualquier número real positivo r , $[r]$ denota el entero más grande que es menor o igual a r . Para cualquier x en \mathbb{R}^n , x^T es su transpuesta y $|x|$ es la norma euclidiana. Para una matriz $A_{n \times m}$, $|A|$ significa la norma de la matriz inducida.

Para cualquier función $\phi : \mathbb{Z}_+ \rightarrow \mathbb{R}^n$, se denota $\|\phi\| = \sup\{|\phi(k)| : k \in \mathbb{Z}_+\} \leq \infty$. Para el caso cuando ϕ esta acotado, esta es la norma standard l_∞ . Para cualquier $k \in \mathbb{Z}_+$ y cualquier función $\phi : \mathbb{Z}_+ \rightarrow \mathbb{R}^n$, $\phi_{[k]}$ denota el truncamiento de ϕ en k , es decir, $\phi_{[k]}(j) = \phi(j)$ si $j \leq k$, y $\phi_{[k]}(j) = 0$ si $j > k$.

Denotamos $\phi^k := \phi - \phi_{[k]}$. Denotamos id como la función identidad de \mathbb{R} sobre \mathbb{R} , y usamos $\gamma_1 \circ \gamma_2$ para denotar la composición de dos funciones γ_1 y γ_2 que van de \mathbb{R} a \mathbb{R} .

Los controles o entradas son funciones $u : \mathbb{Z}_+ \rightarrow \mathbb{R}^m$. Para un sistema dado, con frecuencia consideramos el mismo sistema pero con controles restringidos para tomar valores en el mismo subconjunto $\Omega \subset \mathbb{R}^m$, usamos M_Ω para denotar el conjunto de controles tomando valores en Ω .

Para cada $\epsilon \in \mathbb{R}^m$ y cada entrada u , denotamos por $x(\cdot, \epsilon, u)$ la trayectoria del sistema (1) con estado inicial $x(0) = \epsilon$ y la entrada u .

Una función $\gamma : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ es una función **K-función** si es continua, estrictamente creciente y $\gamma = 0$, que aumenta y $\gamma(0) = 0$, es una **K_∞-función** si es una **K-función** y también el $\gamma(s) \rightarrow \infty$ cuando $s \rightarrow \infty$, y es una función definida positiva si $\gamma(s) > 0$ para todos los $s > 0$, y $\gamma(0) = 0$. Una función $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ es una **KL-función** si para cada fijo $t \geq 0$, la función $\beta(\cdot, t)$ es una **K-función**, y para cada fijo $s \geq 0$, la función $\beta(s, \cdot)$ es decreciente y $\beta(s, t) \rightarrow 0$ cuando $t \rightarrow \infty$.

Es importante entender la dependencia de las trayectorias de estado sobre la magnitud de entradas por los sistemas del siguiente tipo:

$$\begin{aligned} x(k+1) &= f[x(k), u(k)] \\ y(k) &= h[x(k)] \end{aligned} \tag{3.41}$$

donde las entradas $u(\cdot)$ son funciones de \mathbb{Z}_+ a \mathbb{R}^m . Se asume que $f(0, 0) = 0$, es decir, $\epsilon = 0$ es un equilibrio del sistema a la entrada 0.

3.4.2. Ecuaciones en el espacio de estado

Para sistemas lineales o no lineales en tiempo discreto variantes en el tiempo, la ecuación de estado se puede expresar como:

$$x(k+1) = f[x(k), u(k), k] \tag{3.42}$$

y la ecuación de salida como:

$$y(k) = g[x(k), u(k), k] \quad (3.43)$$

Para los sistemas lineales en tiempo discreto variantes en el tiempo, la ecuación de estado y la ecuación de salida se puede simplificar a:

$$x(k+1) = G(k)x(k) + H(k)u(k)$$

$$y(k) = C(k)x(k) + D(k)u(k)$$

Si el sistema es invariante en el tiempo, entonces la dos últimas ecuaciones se pueden simplificar a:

$$x(k+1) = Gx(k) + Hu(k) \quad (3.44)$$

$$y(k) = Cx(k) + Du(k) \quad (3.45)$$

Del mismo modo que en el caso de tiempo discreto, los sistemas en tiempo continuo (lineal o no lineal) se pueden representar mediante la siguiente ecuación de estado y la siguiente ecuación de salida:

$$\begin{aligned} \dot{x}(t) &= f[x(t), u(t), t] \\ y(t) &= g[x(t), u(t), t] \end{aligned}$$

En la figura 3.2 se muestra la representación en diagrama a bloques del sistema de control en tiempo discreto definido por las ecuaciones (3.44) y (3.45)

Considérese el sistema en tiempo discreto descrito por:

$$\begin{aligned} & y(k) + a_1y(k-1) + a_2y(k-2) + \dots + a_ny(k-n) \\ = & \quad \quad \quad b_0u(k) + b_1u(k-1) + \dots + b_nu(k-n) \end{aligned} \quad (3.46)$$

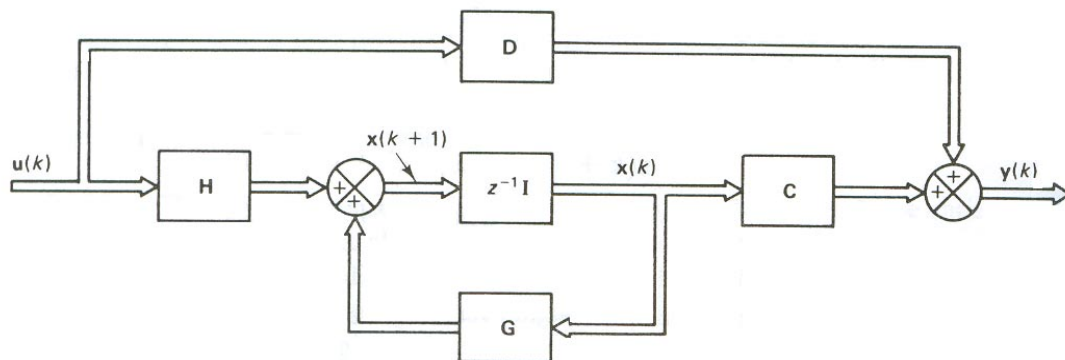


Figura 3.2: Diagrama a bloques de un sistema de control lineal en tiempo discreto.

donde $u(k)$ es la entrada e $y(k)$ es la salida del sistema en el instante de muestreo k .

La ecuación anterior se puede escribir en la forma de la función de transferencia pulso como:

$$\begin{aligned} \frac{Y(Z)}{U(Z)} &= \frac{b_0 + b_1 Z^{-1} + \dots + b_n Z^{-n}}{1 + a_1 Z^{-1} + \dots + a_n Z^{-n}} \\ \frac{Y(Z)}{U(Z)} &= \frac{b_0 Z^n + b_1 Z^{n-1} + \dots + b_n}{Z^n + a_1 Z^{n-1} + \dots + a_n} \end{aligned} \quad (3.47)$$

La principal preocupación es entender algunos conceptos de *Estabilidad de Entrada-Estado (ISS)*. Considerese el siguiente sistema no lineal en tiempo discreto:

$$\begin{aligned} x(k+1) &= f[x(k), u(k)] \\ y(k) &= h[x(k)] \end{aligned} \quad (3.48)$$

donde $u(k) \in \mathbb{R}^m$ es el vector de entrada, $x(k) \in \mathbb{R}^n$ es el vector de estado, $y(k) \in \mathbb{R}^l$ es el vector de salida, f & h son funciones en general suaves no lineales, $f, h \in \mathbb{C}^\infty$.

Asumimos que $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ es continua, y $h : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ es continua.

Fuimos motivados por la noción de *Estabilidad de Entrada-Estado* correspondiente, que fue propuesta originalmente por Sontag (1989,1990) para sistemas no lineales en tiempo

continuo. La propiedad de ISS tiene relación con la continuidad de trayectorias de estado en los estados iniciales y las entradas. Estrictamente hablando, un sistema es ISS si cada trayectoria de estado que corresponde a un control limitado o acotado permanecerá acotado, y la trayectoria tenderá a ser pequeña si la señal de entrada es pequeña no importando que estado inicial es. Nuestro interés en sistemas no lineales (3.42) en tiempo discreto (o en diferencias) tienen su propio interés y han encontrado aplicaciones en varios campos. Por ejemplo la teoría de estabilidad de diferentes sistemas fue recientemente usado para diseñar leyes de control de estabilización para sistemas no lineales en tiempo discreto. Uno de los principales resultados en este trabajo proporciona una relación de la equivalencia entre *ISS* y la función de *Lyapunov-ISS*. Como en el contexto en tiempo continuo, mostramos que para sistemas en tiempo discreto (3.44) puede ser interpretados *ISS* si y solo si es globalmente estabilizable vía retroalimentación. Sin embargo, nuevos fenómenos surgen en la extensión de tiempo continuo a tiempo discreto. Para sistemas afines en tiempo continuo, la utilización continua implica estabilización *ISS* por medio de cambio de retroalimentación de estado $u = k(x) + \nu$. También se ha mostrado que los teoremas de ganancia pequeña *ISS* son extensibles al caso en tiempo discreto.

Capítulo 4

ISS para identificación con redes neuronales en tiempo continuo

Muchas aplicaciones muestran que la neuro identificación ha emergido como una herramienta efectiva para sistemas no lineales desconocidos. Este enfoque o aproximación de modelo libre usa las versátiles características de redes neuronales, porque la falta de modelos las hace ideales para la obtención de resultados teóricos sobre estabilidad y realización de neuro-identificadores . Es muy importante para los ingenieros asegurar la estabilidad de neuro-identificadores en teoría antes de aplicarlos en sistemas reales. Dos tipos de estabilidad para neuro-identificadores han sido estudiados. La estabilidad de neuro-identificadores puede ser encontrada en [16] y [19]. La estabilidad de algoritmos de aprendizaje fué discutida por [15] y [27]. Enfatizaremos en esta tesis sobre la obtención o derivación novedosa de algoritmos de aprendizaje estable del neuro-identificador multicapas.

La estabilidad asintótica global (GAS) de redes neuronales dinámicas fue desarrollada durante la pasada década. La estabilidad diagonal [6] y semi-definida negativa [7] de la interconexión matricial o de matrices puede hacer que el neuro-circuito electrónico de Hopfield-Tank sea globalmente asintóticamente estable (GAS); y es que la estabilidad de neuro circuitos fue establecida por el concepto de estabilidad diagonal. En 1986, J. Hopfield y D. Tank, presentaron una nueva forma de desarrollar conversores análogos-digitales

basada en plantear el diseño como la solución de un problema de optimización en términos de la función energía de una red neuronal de Hopfield. Los perceptrones multicapas y las redes neuronales recurrentes pueden ser relacionados con los sistemas de *Lur* ; la estabilidad absoluta fue desarrollada por [14] y [8] . Como el análisis de Lyapunov es adecuado para redes neuronales dinámicas , las redes de señal de capa fueron discutidas en [11] y [18] , redes de alto orden y redes multicapa puede ser encontrada en [27] y [9] . El método de *Estabilidad de Entrada-Estado (ISS)* [13] es otra efectiva herramienta para redes neuronales dinámicas, [12] concluyendo que las redes neuronales recurrentes son ISS y GAS con entrada cero si los pesos son suficientemente pequeños.

La estabilidad de los algoritmos de aprendizaje puede ser derivada analizando la identificación o seguimiento de los errores de las redes neuronales. En [5] se estudiaron las condiciones de estabilidad de las leyes actualizadas cuando el perceptrón multicapas son usados para identificación y control de un sistema no-lineal . En [15] la retroalimentación dinámica fue modificada con restricciones o limitaciones de estabilidad NL_q . Puesto que las redes neuronales no pueden competir exactamente con sistemas no lineales desconocidos, algunas modificaciones robustas [4] serán aplicadas sobre el gradiente normal o el algoritmo back-propagation o retroalimentación [5], [11], [14], [18].

El enfoque o aproximación de pasividad puede conducir con sistemas no lineales definidos no satisfactoriamente, generalmente por medio límites de sector, y ofrecer soluciones elegantes para la prueba de estabilidad absoluta. Puede conducir a conclusiones generales sobre la estabilidad usando solo características de entrada-salida. Las propiedades de pasividad de redes neuronales multicapas estáticas fueron examinadas en [2] . Por medio del análisis del interconectado de modelos de error, ellos derivaron las relaciones entre la pasividad y lazo cerrado estable. La técnica de pasividad puede también ser aplicada sobre redes neuronales dinámicas. Las propiedades de pasividad de redes neuronales dinámicas puede ser encontrada en [19]. Esta técnica fué también extendida para neuro-identificadores en el caso de capa simple [20]. Concluyendo que los algoritmos de aprendizaje comunmente usados con modificaciones robustas tales como zona-muerta [5] y modificación- σ [11] no son necesarios.

En esta tesis, extenderemos nuestros resultados previos de redes neuronales dinámicas de capa simple [19][20] para el caso multicapas. Para lo mejor de nuestro conocimiento, el análisis en lazo abierto basado en el método de pasividad para redes neuronales dinámicas multicapa no ha sido todavía establecido en la literatura especializada. La teoría de pasividad es aplicada para analizar la estabilidad de neuro-identificadores dinámicos multicapas. Mostramos que la ley de aprendizaje como el backpropagation (retroalimentación) puede hacer la identificación del error de estabilidad, estabilidad asintótica y estabilidad de entrada-estado. Las simulaciones de identificación de la velocidad en ralentí de un motor fuel injection de un vehículo parado o en reposo, mostrará la efectividad del algoritmo propuesto en esta tesis.

4.1. Neuro Identificación vía Técnica ISS

Consideremos una clase de sistemas no lineales descritos por:

$$\begin{aligned}\dot{x}_t &= f(x_t, u_t) \\ y_t &= h(x_t, u_t)\end{aligned}\tag{4.1}$$

donde $x_t \in \mathfrak{R}^n$ es el estado, $u_t \in \mathfrak{R}^m$ es el vector de entrada, $y_t \in \mathfrak{R}^m$ es el vector de salida. $f : \mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}^n$ es localmente Lipschitz, $h : \mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}^m$ es continua. Siguiendo el trabajo de algunos autores como *Byrnes*, ahora es necesario recordar algunas propiedades de pasividad además de propiedades de estabilidad para sistemas pasivos.

Definición 11 *Un sistema (4.1) se dice pasivo si existe una C^r función no-negativa $S(x_t) : \mathfrak{R}^n \rightarrow \mathfrak{R}$, llamada función de almacenamiento, tal que, para toda u_t , todas las condiciones iniciales x^0 y todas las $t \geq 0$ la siguiente desigualdad sostiene o afirma que*

$$\dot{S}(x_t) \leq u_t^T y_t - \varepsilon u_t^T u_t - \delta y_t^T y_t - \rho \psi(x_t)\tag{4.2}$$

donde ε , δ y ρ son constantes no negativas, $\psi(x_t)$ es una función semidefinida positiva de x_t tal que $\psi(0) = 0$. $\rho\psi(x_t)$ es llamado índice o tasa de disipación de estado.

Sin embargo, el sistema se dice ser estrictamente pasivo si existe una función definida positiva $V(x_t) : \mathfrak{R}^n \rightarrow \mathfrak{R}$ tal que $\dot{S}(x_t) \leq u_t^T y_t - V(x_t)$

Property 1. Si la función de almacenamiento $S(x_t)$ es diferenciable y el sistema dinámico es pasivo, la función de almacenamiento $S(x_t)$ satisface $\dot{S}(x_t) \leq u_t^T y_t$.

Definición 12 Un sistema (4.1) se dice ser globalmente estable de entrada-estado si existe una \mathcal{K} -función $\gamma(s)$ (continua y estrictamente creciente $\gamma(0) = 0$) y \mathcal{KL} -función $\beta(s, t)$ (\mathcal{K} -función y para cada fijo $s_0 \geq 0$, $\lim_{t \rightarrow \infty} \beta(s_0, t) = 0$), tal que, para cada $u_t \in L_\infty$ ($\|u(t)\|_\infty < \infty$) y cada estado inicial $x^0 \in \mathfrak{R}^n$, se afirma que

$$\|x(t, x^0, u_t)\| \leq \beta(\|x^0\|, t) + \gamma(\|u_t\|_\infty) \quad (4.3)$$

para cada $t \geq 0$.

Property 2. Si un sistema es estable de entrada-estado, el comportamiento del sistema permanecerá acotado cuando las entradas son acotadas.

4.2. Identificación de sistemas con Redes Neuronales mono-capa

La identificación de un sistema no lineal esta dado por

$$\dot{x}_t = f(x_t, u_t, t), \quad x_t \in \mathfrak{R}^n, u_t \in \mathfrak{R}^m \quad (4.4)$$

Construimos la siguiente red neuronal para identificación

$$\dot{\hat{x}}_t = A\hat{x}_t + W_{1,t}\sigma(\hat{x}_t) + W_{2,t}\phi(\hat{x}_t)\gamma(u_t) \quad (4.5)$$

donde $\hat{x}_t \in \mathfrak{R}^n$ es el estado de la red neuronal, $A \in \mathfrak{R}^{n \times n}$ es una matriz estable. $W_{1,t} \in \mathfrak{R}^{n \times n}$, $W_{2,t} \in \mathfrak{R}^{n \times n}$ son matrices de pesos de las redes neuronales. La función vectorial $\sigma(x_t) \in \mathfrak{R}^n$ es considerada de n -dimensión con los elementos monótonamente crecientes. La función

matricial $\phi(\cdot)$ es considerada $\mathfrak{R}^{n \times m}$ diagonal: $\phi(\hat{x}_t) = \text{diag}(\phi_1(\hat{x}_1) \cdots \phi_n(\hat{x}_n))$. $\gamma(u_t) \in \mathfrak{R}^m$. La función $\gamma(\cdot)$ es seleccionada como $\|\gamma(u_t)\|^2 \leq \bar{u}$. La presentación típica de los elementos $\sigma_i(\cdot)$ y $\phi_i(\cdot)$ son funciones sigmoidales, i.e.

$$\sigma_i(x_i) = \frac{a_i}{1 + e^{-b_i x_i}} - c_i$$

Propiedad 3 *Las redes neuronales dinámicas han sido discutidas por muchos autores, por ejemplo [11], [27], [9] y [18]. Se puede ver que el modelo de Hopfield es el caso especial de estas redes con $A = \text{diag}\{a_i\}$, $a_i := -1/R_i C_i$, $R_i > 0$ y $C_i > 0$. R_i y C_i son la resistencia y capacitancia en el i –ésimo nodo de la red respectivamente.*

Definamos el error de identificación como

$$\Delta_t = \hat{x}_t - x_t$$

Porque $\sigma(\cdot)$ y $\phi(\cdot)$ son elegidas como funciones sigmoidales, claramente satisfacen la siguiente propiedad de *Lipschitz*

$$\tilde{\sigma}^T \Lambda_1 \tilde{\sigma} \leq \Delta_t^T D_\sigma \Delta_t, \quad \left(\tilde{\phi}_t \gamma(u_t)\right)^T \Lambda_2 \left(\tilde{\phi}_t \gamma(u_t)\right) \leq \bar{u} \Delta_t^T D_\phi \Delta_t, \quad (4.6)$$

donde $\tilde{W}_{1,t} = W_{1,t} - W_1^*$, $\tilde{W}_{2,t} = W_{2,t} - W_2^*$, $\tilde{\sigma} = \sigma(\hat{x}_t) - \sigma(x_t)$, $\tilde{\phi} = \phi(\hat{x}_t) - \phi(x_t)$, Λ_1 , Λ_2 , D_σ y D_ϕ son constantes positivas conocidas, no lineales.

Generalmente, una red neuronal dinámica (4.5) no puede seguir el sistema no lineal exactamente. El sistema no lineal puede ser escrito como

$$\dot{x}_t = Ax_t + W_1^* \sigma(x_t) + W_2^* \phi(x_t) \gamma(u_t) - \tilde{f}_t \quad (4.7)$$

donde W_1^* y W_2^* son matrices desconocidas acotadas

$$W_1^* \Lambda_1^{-1} W_1^{*T} \leq \bar{W}_1, \quad W_2^* \Lambda_2^{-1} W_2^{*T} \leq \bar{W}_2 \quad (4.8)$$

\bar{W}_1 y \bar{W}_2 son matrices conocidas previamente, la función vectorial \tilde{f}_t puede ser considerada como error modelado y perturbaciones. Las redes neuronales multicapas dinámicas se obtienen de la siguiente manera:

$$\dot{\Delta}_t = A \Delta_t + \tilde{W}_{1,t} \sigma(\hat{x}_t) + \tilde{W}_{2,t} \phi(\hat{x}_t) \gamma(u_t) + W_1^* \tilde{\sigma} + W_1^* \tilde{\phi} \gamma(u_t) + \tilde{f}_t \quad (4.9)$$

Si definimos

$$R = \overline{W}_1 + \overline{W}_2, \quad Q = D_\sigma + \overline{u}D_\phi + Q_0 \quad (4.10)$$

y las matrices A y Q_0 son seleccionadas para satisfacer las siguientes condiciones:

- (1) El par $(A, R^{1/2})$ es controlable, el par $(Q^{1/2}, A)$ es observable,
- (2) la condición de frecuencia local [17] satisface

$$A^T R^{-1} A - Q \geq \frac{1}{4} [A^T R^{-1} - R^{-1} A] R [A^T R^{-1} - R^{-1} A]^T \quad (4.11)$$

entonces la siguiente suposición puede ser establecida:

A1: Existe una matriz estable A y una matriz estrictamente definida positiva Q_0 tal que la ecuación matricial de Riccati

$$A^T P + PA + PRP + Q = 0 \quad (4.12)$$

tendrá una solución positiva $P = P^T > 0$.

Estas condiciones son fácilmente realizadas si seleccionamos A como una matriz diagonal estable. El siguiente teorema establece el procedimiento de aprendizaje del neuro-identificador.

Teorema 8 *Sujeto a la proposición A1 es satisfecha, si los pesos $W_{1,t}$ y $W_{2,t}$ son actualizados como*

$$\begin{aligned} \dot{W}_{1,t} &= -K_1 P \Delta_t \sigma^T(\hat{x}_t) \\ \dot{W}_{2,t} &= -K_2 P \phi(\hat{x}_t) \gamma(u_t) \Delta_t^T \end{aligned} \quad (4.13)$$

donde P es la solución de la ecuación de Riccati, entonces el error de identificación dinámico es estrictamente pasivo para el error modelado \tilde{f}_t para el error de identificación $2P\Delta_t$

Demostración. Seleccionando una función de Lyapunov (función de almacenamiento) como

$$S_t = \Delta_t^T P \Delta_t + tr \left\{ \widetilde{W}_{1,t}^T K_1^{-1} \widetilde{W}_{1,t} \right\} + tr \left\{ \widetilde{W}_{2,t}^T K_2^{-1} \widetilde{W}_{2,t} \right\} \quad (4.14)$$

donde $P \in \mathfrak{R}^{n \times n}$ es una matriz definida positiva. De acuerdo a (4.14), la derivada es

$$\begin{aligned} \dot{S}_t = & \Delta_t^T (PA + A^T P) \Delta_t + 2\Delta_t^T P \widetilde{W}_{1,t} \sigma(\widehat{x}_t) + 2\Delta_t^T P \widetilde{W}_{2,t} \phi(\widehat{x}_t) \gamma(u_t) + 2\Delta_t^T P \widetilde{f}_t \\ & + 2\Delta_t^T P \left[W_1^* \widetilde{\sigma} + W_1^* \widetilde{\phi} \gamma(u_t) \right] + 2tr \left\{ \dot{\widetilde{W}}_{1,t}^T K_1^{-1} \widetilde{W}_{1,t} \right\} + 2tr \left\{ \dot{\widetilde{W}}_{2,t}^T K_2^{-1} \widetilde{W}_{2,t} \right\} \end{aligned} \quad (4.15)$$

De modo que $\Delta_t^T P W_1^* \widetilde{\sigma}_t$ es un escalar , usando (4.12) y la desigualdad matricial

$$X^T Y + (X^T Y)^T \leq X^T \Lambda^{-1} X + Y^T \Lambda Y \quad (4.16)$$

donde $X, Y, \Lambda \in \mathfrak{R}^{n \times k}$ son cualesquiera matrices, Λ es cualquier matriz definida positiva, obtenemos

$$\begin{aligned} 2\Delta_t^T P W_1^* \widetilde{\sigma}_t & \leq \Delta_t^T P W_1^* \Lambda_1^{-1} W_1^{*T} P \Delta_t + \widetilde{\sigma}_t^T \Lambda_1 \widetilde{\sigma}_t \leq \Delta_t^T (P \overline{W}_1 P + D_\sigma) \Delta_t \\ 2\Delta_t^T P W_2^* \widetilde{\phi}_t \gamma(u_t) & \leq \Delta_t^T (P \overline{W}_2 P + \overline{u} D_\phi) \Delta_t \end{aligned} \quad (4.17)$$

Así que tenemos

$$\begin{aligned} \dot{S}_t & \leq \Delta_t^T \left[PA + A^T P + P (\overline{W}_1 + \overline{W}_2) P + (D_\sigma + \overline{u} D_\phi + Q_0) \right] \Delta_t \\ & + 2tr \left\{ \dot{\widetilde{W}}_{1,t}^T K_1^{-1} \widetilde{W}_{1,t} \right\} + 2\Delta_t^T P \widetilde{W}_{1,t} \sigma(\widehat{x}_t) + 2\Delta_t^T P \widetilde{f}_t \\ & + 2tr \left\{ \dot{\widetilde{W}}_{2,t}^T K_2^{-1} \widetilde{W}_{2,t} \right\} + 2\Delta_t^T P \widetilde{W}_{2,t} \phi(\widehat{x}_t) \gamma(u_t) - \Delta_t^T Q_0 \Delta_t \end{aligned}$$

De tal modo que $\dot{\widetilde{W}}_{1,t} = \dot{W}_{1,t}$, si usamos la ley actualizada como en (4.16) y **A1**, tenemos

$$\dot{S}_t \leq -\Delta_t^T Q_0 \Delta_t + 2\Delta_t^T P \widetilde{f}_t \quad (4.18)$$

De la Definición 1, si definimos la entrada como \widetilde{f}_t y la salida como $2P\Delta_t$, entonces el sistema es estrictamente pasivo con $V_t = \Delta_t^T Q_0 \Delta_t \geq 0$. ■

Propiedad 4 *De modo que la tasa de actualización es $K_i P$ ($i = 1, 2$), y K_i puede ser seleccionada como cualquier matriz positiva, el proceso de aprendizaje de la red neuronal dinámica esta libre de la solución de la ecuación de Riccati.*

Corolario 9 Si solo presenta parámetros inciertos ($\tilde{f}_t = 0$), entonces la ley actualizada puede hacer la identificación del error asintóticamente estable ,

$$\lim_{t \rightarrow \infty} \Delta_t = 0, \quad W_{1,t} \in L_\infty, \quad W_{2,t} \in L_\infty, \quad \lim_{t \rightarrow \infty} \dot{W}_{1,t} = 0, \quad \lim_{t \rightarrow \infty} \dot{W}_{2,t} = 0 \quad (4.19)$$

Demostración. De tal modo que el error de identificación dinámico es pasivo y $\tilde{f}_t = 0$, de la *Propiedad 1* la función de almacenamiento $S(x_t)$ satisface

$$\dot{S}(x_t) \leq \tilde{f}_t^T 2P\Delta_t = 0$$

El definido positivo $S(x_t)$ implica Δ_t , $W_{1,t}$ y $W_{2,t}$ son acotados. Del la ecuación de error $\dot{\Delta}_t \in L_\infty$

$$\dot{S} \leq -\Delta_t^T Q_0 \Delta_t \leq 0$$

Integrando ambos lados

$$\int_0^\infty \|\Delta_t\|_{Q_0} \leq S_0 - S_\infty < \infty$$

Así que $\Delta_t \in L_2 \cap L_\infty$, usando el *Lemma de Barlat* [4] tenemos del teorema 1. De tal modo que u_t , $\sigma(\hat{x}_t)$, $\phi(\hat{x}_t)$ y P son acotados , $\lim_{t \rightarrow \infty} \dot{W}_{1,t} = 0$ y $\lim_{t \rightarrow \infty} \dot{W}_{2,t} = 0$. ■

Propiedad 5 Para el caso del modelo combinado , el análisis de Lyapunov puede alcanzar los mismos resultados (ver *Corollary 1* y *Corollary 2* en [18]). Pero en presencia del error modelado, daremos un resultado absolutamente diferente : el gradiente del algoritmo descendente es también robusto respecto a la perturbación y dinámica no modelada acotada.

Teorema 10 Si suponemos A1 es satisfecha y existe una matriz definida positiva Λ_f tal que

$$\lambda_{\max}(P\Lambda_f P) \leq \lambda_{\min}(Q_0) \quad (4.20)$$

entonces la ley actualizada puede hacer la dinámica del neuro-identificador estable de entrada-estado (ISS).

Demostración. En vista de la desigualdad matricial ,

$$2\Delta_t^T P \tilde{f}_t \leq \Delta_t^T P \Lambda_f P \Delta_t + \tilde{f}_t^T \Lambda_f^{-1} \tilde{f}_t \quad (4.21)$$

puede ser representada como

$$\begin{aligned} \dot{S}_t &\leq -\Delta_t^T Q_0 \Delta_t + 2\Delta_t^T P \tilde{f}_t \leq -\lambda_{\min}(Q_0) \|\Delta_t\|^2 + \Delta_t^T P \Lambda_f P \Delta_t + \tilde{f}_t^T \Lambda_f^{-1} \tilde{f}_t \\ &\leq -\alpha_{\|\Delta_t\|} \|\Delta_t\| + \beta_{\|\tilde{f}_t\|} \|\tilde{f}_t\| \end{aligned}$$

donde $\alpha_{\|\Delta_t\|} := [\lambda_{\min}(Q_0) - \lambda_{\max}(P \Lambda_f P)] \|\Delta_t\|$, $\beta_{\|\tilde{f}_t\|} := \lambda_{\max}(\Lambda_f^{-1}) \|\tilde{f}_t\|$. Podemos seleccionar una matriz definida positiva Λ_f tal que (4.20) esta establecida . Así que α y β son \mathcal{K}_∞ funciones, S_t es una función de Lyapunov-ISS .Usando el siguiente Teorema 1 de [13], la dinámica del error de identificación es estable de entrada-estado (ISS). ■

Propiedad 6 *Primero, podemos elegir A y Q_0 tal que la ecuación de Riccati tendrá una solución positiva P . Entonces Λ_f puede ser encontrada de acuerdo a la condición. De tal modo que es correcta para cualquier matriz definida positiva , puede ser establecida si Λ_f es seleccionada como una suficientemente pequeña matriz constante. La condición no tendrá efectos sobre la dinámica de las redes y sus entrenamientos.*

Corolario 11 *Si el error modelado \tilde{f}_t es acotado , entonces la ley actualizada puede hacer el procedimiento de identificación estable*

$$\Delta_t \in L_\infty, \quad W_{1,t} \in L_\infty, \quad W_{2,t} \in L_\infty$$

Demostración. De la Propiedad 2 conocemos que por medio de la *estabilidad de entrada-estado* que el comportamiento de la red neuronal dinámica permanecerá acotada cuando sus entradas estan acotadas. ■

Propiedad 7 *De modo que el estado y las variables de salida son físicamente acotadas, el error modelado \tilde{f}_t puede ser considerado acotado también(ver por ejemplo [5][9][11]). A diferencia de la leyes adaptivas robustas , tales como zona-muerta [9] modificación- σ [27], el límite superior del error modelado no es necesario para el entrenamiento de la red.*

Propiedad 8 *Es bien conocido que estructuras con incertidumbres causarán parámetros sin sentido para control adaptivo , así que se podrá usar modificación robusta para hacer estable*

la identificación del error [4] . Los métodos adaptivos robustos pueden ser extendidos para neuro-identificación directamente [5][9] [11] .Pero el método de neuro-identificación es un tipo de "caja negra", no se necesita información de la estructura y todas las incertidumbres están dentro la caja negra. Aunque los algoritmos adaptivos robustos son adecuados para neuro-identificación , ellos no lo simplifican. Por medio de técnica de pasividad, exitosamente probaremos nuestra conclusión: el gradiente del algoritmo descendente para los pesos es robusto con respecto a cualquier incertidumbre acotada.

4.3. Identificación de sistemas con Redes Neuronales Multicapas

Construyendo la siguiente red neuronal dinámica:

$$\dot{\hat{x}}_t = A\hat{x}_t + W_{1,t}\sigma(V_{1,t}\hat{x}_t) + W_{2,t}\phi(V_{2,t}\hat{x}_t)\pi(u_t) \quad (4.22)$$

donde $\hat{x}_t \in \mathfrak{R}^n$ es el estado de la red neuronal, $A \in \mathfrak{R}^{n \times n}$ es una matriz estable. $W_{1,t} \in \mathfrak{R}^{n \times m}$, $W_{2,t} \in \mathfrak{R}^{n \times m}$ son los pesos de las matrices de las capas de salida, $V_{1,t} \in \mathfrak{R}^{m \times n}$, $V_{2,t} \in \mathfrak{R}^{m \times n}$ son matrices de peso de las capas ocultas. El campo vectorial $\sigma(x_t) : \mathfrak{R}^n \rightarrow \mathfrak{R}^m$ es considerado que tiene los elementos monótonamente crecientes. La función $\phi(\cdot)$ es mapeada de $\mathfrak{R}^m \rightarrow \mathfrak{R}^{m \times m}$. $\pi(u_t) \in \mathfrak{R}^m$, es seleccionada como función de saturación: $\|\pi(u_t)\|^2 \leq \bar{u}$. La presentación típica de los elementos $\sigma_i(\cdot)$ y $\phi_{ii}(\cdot)$ son funciones sigmoideas

$$\sigma_i(x_{i,t}) = a_i / (1 + e^{-b_i x_{i,t}}) - c_i$$

Propiedad 9 Las redes neuronales han sido discutidas por muchos autores , por ejemplo [11], [27],[9] y [18]. Puede verse que el modelo de Hopfield es un caso especial de estas redes con $A = \text{diag}\{a_i\}$, $a_i := -1/R_i C_i$, $R_i > 0$ y $C_i > 0$. R_i y C_i son la resistencia y capacitancia en el i -ésimo nodo de la red respectivamente .

Generalmente, el sistema no-lineal (4.1) puede ser representado de la manera siguiente

$$\dot{x}_t = Ax_t + W_1^* \sigma(V_1^0 x_t) + W_2^* \phi(V_2^0 x_t) \pi(u_t) - \tilde{f}_t(V_1^0, V_2^0) \quad (4.23)$$

donde W_1^* y W_2^* son matrices óptimas que pueden minimizar el error de modelado \tilde{f}_t , están acotadas como

$$W_1^* \Lambda_1^{-1} W_1^{*T} \leq \overline{W}_1, \quad W_2^* \Lambda_2^{-1} W_2^{*T} \leq \overline{W}_2$$

V_1^0 y V_2^0 son previamente matrices dadas que son obtenidas de aprendizaje *off-line*. Se define el error de identificación como $\Delta_t = \hat{x}_t - x_t$, $\tilde{\sigma}_t = \sigma(V_1^0 \hat{x}_t) - \sigma(V_1^0 x_t)$, $\tilde{\phi}_t = \phi(V_2^0 \hat{x}_t) \pi(u_t) - \phi(V_2^0 x_t) \pi(u_t)$, $\tilde{\sigma}'_t = \sigma(V_{1,t} \hat{x}_t) - \sigma(V_1^0 \hat{x}_t)$, $\tilde{\phi}'_t = \phi(V_{2,t} \hat{x}_t) \pi(u_t) - \phi(V_2^0 \hat{x}_t) \pi(u_t)$, $\tilde{V}_{1,t} = V_{1,t} - V_1^0$, $\tilde{V}_{2,t} = V_{2,t} - V_2^0$, $\tilde{W}_{1,t} = W_{1,t} - W_1^*$, $\tilde{W}_{2,t} = W_{2,t} - W_2^*$. Porque $\sigma(\cdot)$ y $\phi(\cdot)$ son elegidos como funciones sigmoideas, claramente satisfacen la condición de *Lipschitz*,

$$\begin{aligned} \tilde{\sigma}_t^T \Lambda_1 \tilde{\sigma}_t &\leq \Delta_t^T \Lambda_\sigma \Delta_t, & \tilde{\phi}_t^T \Lambda_2 \tilde{\phi}_t &\leq \bar{u} \Delta_t^T \Lambda_\phi \Delta_t \\ \tilde{\sigma}'_t &= D_\sigma \tilde{V}_{1,t} \hat{x}_t + \nu_\sigma, & \tilde{\phi}'_t &= D_\phi \tilde{V}_{2,t} \hat{x}_t + \nu_\phi \end{aligned}$$

donde

$$\begin{aligned} D_\sigma &= \frac{\partial \sigma^T(Z)}{\partial Z} \Big|_{Z=V_{1,t} \hat{x}_t}, & \|\nu_\sigma\|_{\Lambda_1}^2 &\leq l_1 \left\| \tilde{V}_{1,t} \hat{x}_t \right\|_{\Lambda_1}^2 \\ D_\phi &= \frac{\partial [\phi(Z) \pi(u_t)]^T}{\partial Z} \Big|_{Z=V_{2,t} \hat{x}_t}, & \|\nu_\phi\|_{\Lambda_2}^2 &\leq l_2 \left\| \tilde{V}_{2,t} \hat{x}_t \right\|_{\Lambda_2}^2 \end{aligned}$$

$l_1 > 0$, $l_2 > 0$, Λ_1 , Λ_2 , Λ_σ y Λ_ϕ son matrices definidas positivas. La dinámica del error de identificación es obtenida de (4.22) y (4.23)

$$\begin{aligned} \dot{\Delta}_t &= A \Delta_t + \tilde{W}_{1,t} \sigma(V_{1,t} \hat{x}_t) + \tilde{W}_{2,t} \phi(V_{2,t} \hat{x}_t) \pi(u_t) \\ &\quad + W_1^* \tilde{\sigma}_t + W_1^* \tilde{\sigma}'_t + W_2^* \tilde{\phi}_t + W_2^* \tilde{\phi}'_t + \tilde{f}_t(V_1^0, V_2^0) \end{aligned}$$

De donde conocemos si las matrices A , R y Q satisfacen las siguientes condiciones:

- (a) el par $(A, R^{1/2})$ es controlable, el par $(Q^{1/2}, A)$ es observable,
- (b) *condición de frecuencia local*

$$A^T R^{-1} A - Q \geq \frac{1}{4} [A^T R^{-1} - R^{-1} A] R [A^T R^{-1} - R^{-1} A]^T$$

la ecuación matricial de Riccati

$$A^T P + PA + PRP + Q = 0$$

tendrá una solución positiva $P = P^T > 0$.

Si A es una matriz diagonal estable, R y Q son matrices definidas positivas, las condiciones (a) y (b) son fácilmente satisfechas. Ahora definimos

$$R = 2\overline{W}_1 + 2\overline{W}_2, \quad Q = \Lambda_\sigma + \overline{u}\Lambda_\phi + Q_0$$

Asumimos la siguiente consideración que puede ser establecida.

A1: Existe una matriz estable A y una matriz definida estrictamente positiva Q_0 tal que (4.12) tendrá una solución positiva.

El siguiente teorema da un procedimiento de aprendizaje estable de neuro identificador.

Teorema 12 Si los pesos $W_{1,t}$, $W_{2,t}$, $V_{1,t}$ y $V_{2,t}$ son actualizados como

$$\begin{aligned} \dot{W}_{1,t} &= -K_1 P \sigma(V_{1,t} \hat{x}_t) \Delta_t^T + K_1 P D_\sigma (V_{1,t} - V_1^0) \hat{x}_t \Delta_t^T \\ \dot{W}_{2,t} &= -K_2 P \phi(V_{2,t} \hat{x}_t) \pi(u_t) \Delta_t^T + K_2 P D_\phi (V_{2,t} - V_2^0) \hat{x}_t \pi(u_t) \Delta_t^T \\ \dot{V}_{1,t} &= -K_3 P W_{1,t} D_\sigma \Delta_t \hat{x}_t^T - \frac{1}{2} K_3 \Lambda_1 (V_{1,t} - V_1^0) \hat{x}_t \hat{x}_t^T \\ \dot{V}_{2,t} &= -K_4 P W_{2,t} D_\phi \Delta_t \hat{x}_t^T - \frac{1}{2} K_4 \Lambda_2 (V_{2,t} - V_2^0) \hat{x}_t \hat{x}_t^T \end{aligned}$$

donde P es la solución de la ecuación de Riccati, entonces la dinámica del error de identificación es estrictamente pasivo de $\tilde{f}_t (V_1^0, V_2^0)$ para la identificación del error $2P\Delta_t$

Demostración. Seleccionando una función de Lyapunov (función de almacenamiento) como

$$S_t = \Delta_t^T P \Delta_t + tr \left\{ \tilde{W}_{1,t}^T K_1^{-1} \tilde{W}_{1,t} \right\} + tr \left\{ \tilde{W}_{2,t}^T K_2^{-1} \tilde{W}_{2,t} \right\} + tr \left\{ \tilde{V}_{1,t}^T K_3^{-1} \tilde{V}_{1,t} \right\} + tr \left\{ \tilde{V}_{2,t}^T K_4^{-1} \tilde{V}_{2,t} \right\} \quad (4.24)$$

donde $P \in \mathfrak{R}^{n \times n}$ es una matriz definida positiva. De acuerdo a, la derivada de es

$$\begin{aligned} \dot{S}_t &= \Delta_t^T (PA + A^T P) \Delta_t + 2\Delta_t^T P \tilde{f}_t (V_1^0, V_2^0) \\ &+ 2\Delta_t^T P \tilde{W}_{1,t} \sigma(V_{1,t} \hat{x}_t) + 2\Delta_t^T P \tilde{W}_{2,t} \phi(V_{2,t} \hat{x}_t) \pi(u_t) + 2tr \left\{ \dot{\tilde{W}}_{1,t}^T K_1^{-1} \tilde{W}_{1,t} \right\} + 2tr \left\{ \dot{\tilde{W}}_{2,t}^T K_2^{-1} \tilde{W}_{2,t} \right\} \\ &+ 2\Delta_t^T P \left[W_1^* \tilde{\sigma}_t + W_1^* \tilde{\sigma}'_t + W_2^* \tilde{\phi}_t + W_2^* \tilde{\phi}'_t \right] + 2tr \left\{ \dot{\tilde{V}}_{1,t}^T K_3^{-1} \tilde{V}_{1,t} \right\} + 2tr \left\{ \dot{\tilde{V}}_{2,t}^T K_4^{-1} \tilde{V}_{2,t} \right\} \end{aligned}$$

De modo que $2\Delta_t^T PW_1^* \tilde{\sigma}_t$ es ecalar , usando(4.12) y la desigualdad matricial

$$X^T Y + (X^T Y)^T \leq X^T \Lambda^{-1} X + Y^T \Lambda Y \quad (4.25)$$

donde $X, Y, \Lambda \in \mathfrak{R}^{n \times k}$ son matrices , $\Lambda = \Lambda^T > 0$, obtenemos

$$2\Delta_t^T PW_1^* \tilde{\sigma}_t \leq \Delta_t^T PW_1^* \Lambda_1^{-1} W_1^{*T} P \Delta_t + \tilde{\sigma}_t^T \Lambda_1 \tilde{\sigma}_t \leq \Delta_t^T (P\bar{W}_1 P + D_\sigma) \Delta_t \quad (4.26)$$

Similarmente tenemos $2\Delta_t^T PW_2^* \tilde{\phi}_t \leq \Delta_t^T (P\bar{W}_2 P + \bar{u}D_\phi) \Delta_t$,

$$\begin{aligned} 2\Delta_t^T PW_1^* \tilde{\sigma}'_t &= 2\Delta_t^T PW_1^* D_\sigma \tilde{V}_{1,t} \hat{x}_t + 2\Delta_t^T PW_1^* \nu_\sigma \\ &\leq 2\Delta_t^T P (W_{1,t} - \tilde{W}_{1,t}) D_\sigma \tilde{V}_{1,t} \hat{x}_t + \Delta_t^T P \bar{W}_1 P \Delta_t + l_1 \hat{x}_t^T \tilde{V}_{1,t}^T \Lambda_1 \tilde{V}_{1,t} \hat{x}_t \end{aligned}$$

También

$$2\Delta_t^T PW_2^* \tilde{\phi}'_t \leq 2\Delta_t^T P (W_{2,t} - \tilde{W}_{2,t}) D_\phi \tilde{V}_{2,t} \hat{x}_t + \Delta_t^T P \bar{W}_2 P \Delta_t + l_2 \hat{x}_t^T \tilde{V}_{2,t}^T \Lambda_2 \tilde{V}_{2,t} \hat{x}_t$$

Así que obtenemos

$$\begin{aligned} \dot{S}_t &\leq \Delta_t^T [PA + A^T P + P (2\bar{W}_1 + 2\bar{W}_2) P + (D_\sigma + \bar{u}D_\phi + Q_0)] \Delta_t - \Delta_t^T Q_0 \Delta_t + 2\Delta_t^T P \tilde{f}_t \\ &+ 2tr \left\{ \begin{matrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{matrix} \right. \left. \begin{matrix} T \\ T \\ T \\ T \\ T \\ T \\ T \\ T \end{matrix} \right\} \\ &+ 2\Delta_t^T P \tilde{W}_{1,t} \left(\sigma(V_{1,t} \hat{x}_t) - D_\sigma \tilde{V}_{1,t} \hat{x}_t \right) \\ &+ 2\Delta_t^T P \tilde{W}_{2,t} \left(\phi(V_{2,t} \hat{x}_t) - D_\phi \tilde{V}_{2,t} \hat{x}_t \right) \pi(u_t) \\ &+ 2\Delta_t^T PW_{1,t} D_\sigma \tilde{V}_{1,t} \hat{x}_t + l_1 \hat{x}_t^T \tilde{V}_{1,t}^T \Lambda_1 \tilde{V}_{1,t} \hat{x}_t \\ &+ 2\Delta_t^T PW_{2,t} D_\phi \tilde{V}_{2,t} \hat{x}_t + l_2 \hat{x}_t^T \tilde{V}_{2,t}^T \Lambda_2 \tilde{V}_{2,t} \hat{x}_t \end{aligned}$$

Si usamos la ley actualizada y **A1**, tenemos

$$\dot{S}_t \leq -\Delta_t^T Q_0 \Delta_t + 2\Delta_t^T P \tilde{f}_t (V_1^0, V_2^0) \quad (4.27)$$

De la *Definición 1*, si definimos $\tilde{f}_t (V_1^0, V_2^0)$ como entrada, y $2P\Delta_t$ como salida, entonces el sistema es estrictamente pasivo con $V_t = \Delta_t^T Q_0 \Delta_t \geq 0$. ■

Propiedad 10 Puesto que la ganancia actualizada es $K_i P$ ($i = 1 \dots 4$) y K_i puede ser cualquier matriz positiva, el proceso de aprendizaje de la red neuronal dinámica no depende de la solución P de la ecuación de Riccati. Puesto que la consideración **A1** es para seleccionar A tal que tendrá una solución positiva. R esta relacionada para límites superiores de matrices óptimas desconocidas W_1^* y W_2^* , asumimos que conocemos los límites superiores. Q es libre para escoger porque de Q_0 . Para la ecuación matricial podemos cambiar A , R y Q tal que P es positiva, así que casi siempre es posible para satisfacer **A1**.

Propiedad 11 $W_{1,t} D_\sigma \Delta_t$ es el error de retroalimentación para la capa oculta, \hat{x}_t^T es la entrada de la capa oculta; $\sigma(V_{1,t} \hat{x}_t)$ es la entrada para la capa de salida, así que las primeras partes $K_1 P \sigma(V_{1,t} \hat{x}_t) \Delta_t^T$ y $-K_3 P W_{1,t} D_\sigma \Delta_t \hat{x}_t^T$ son la misma que el esquema de retroalimentación para perceptrones multicapa. Las segundas partes son usadas para asegurar las propiedades de pasividad del error de identificación.

Corolario 13 Si las redes neuronales pueden competir con plantas no-lineales exactamente, i.e., solo para parámetros inciertos presentes ($\tilde{f}_t = 0$), entonces la ley actualizada como puede hacer el error de identificación asintóticamente estable,

$$\lim_{t \rightarrow \infty} \Delta_t = 0$$

Demostración. Puesto que la dinámica del error de identificación es pasivo, de la Propiedad 1 la función de almacenamiento $S(x_t)$ satisface

$$\dot{S}(x_t) \leq \tilde{f}_t^T 2P \Delta_t = 0$$

La función definida positiva $S(x_t)$ implica Δ_t , $W_{1,t}$ y $W_{2,t}$ son acotadas. De la ecuación del error $\dot{\Delta}_t \in L_\infty$

$$\dot{S}(x_t) \leq -\Delta_t^T Q_0 \Delta_t \leq 0$$

Integrando ambos lados

$$\int_0^\infty \|\Delta_t\|_{Q_0} \leq S_0 - S_\infty < \infty$$

Así que $\Delta_t \in L_2 \cap L_\infty$, usando el *Lemma de Barbalat* tenemos del teorema 1. ■

Teorema 14 Usando la ley actualizada como, la dinámica del neuro-identificador es estable de entrada-estado (ISS).

Demostración. En vista de la desigualdad matricial ,

$$2\Delta_t^T P \tilde{f}_t \leq \Delta_t^T P \Lambda_f P \Delta_t + \tilde{f}_t^T \Lambda_f^{-1} \tilde{f}_t$$

puede ser representada como

$$\begin{aligned} \dot{S}_t &= -\Delta_t^T Q_0 \Delta_t + 2\Delta_t^T P \tilde{f}_t \leq -\lambda_{\min}(Q_0) \|\Delta_t\|^2 + \Delta_t^T P \Lambda_f P \Delta_t + \tilde{f}_t^T \Lambda_f^{-1} \tilde{f}_t \\ &\leq -\alpha_{\|\Delta_t\|} \|\Delta_t\| + \beta_{\|\tilde{f}_t\|} \|\tilde{f}_t\| \end{aligned}$$

donde $\alpha_{\|\Delta_t\|} := [\lambda_{\min}(Q_0) - \lambda_{\max}(P \Lambda_f P)] \|\Delta_t\|$, $\beta_{\|\tilde{f}_t\|} := \lambda_{\max}(\Lambda_f^{-1}) \|\tilde{f}_t\|$. Podemos seleccionar una matriz definida positiva Λ_f tal que

$$\lambda_{\max}(P \Lambda_f P) \leq \lambda_{\min}(Q_0)$$

Así que α y β son \mathcal{K}_∞ -funciones, S_t es una función de *Lyapunov-ISS*. Usando el Teorema 1 de [13], la dinámica del error de identificación es estable de entrada-estado. ■

Corolario 15 Si el error modelado \tilde{f}_t es acotado, entonces la ley actualizada puede hacer el procedimiento de identificación estable

$$\Delta_t \in L_\infty, \quad W_{1,t} \in L_\infty, \quad W_{2,t} \in L_\infty$$

Demostración. De la propiedad 2 conocemos que la estabilidad de entrada-estado implica que el estado de la red neuronal dinámica permanecerá acotada cuando su entrada es acotada. ■

Propiedad 12 Es bien conocido que estructuras con incertidumbres causarán parámetros sin sentido para control adaptivo, así que se podrá usar modificación robusta para hacer estable la identificación. Los métodos adaptivos robustos pueden ser extendidos para neuro-identificación directamente. Pero el método de neuro-identificación es un tipo de caja negra,

no se necesita información de la estructura y todas las incertidumbres están dentro la caja negra . Aunque los algoritmos adaptivos robustos son adecuados para neuro-identificación , ellos no lo simplifican. Por medio de técnica de pasividad, exitosamente probaremos nuestra conclusión: cómo el algoritmo de retroalimentación o backpropagation es robusto con respecto a todos los tipos de incertidumbres acotadas para neuro-identificación multicapas.

La condición *ISS* puede ser establecida si Λ_f es seleccionada como una matriz pequeña. De tal modo que el estado y las variables de salida son físicamente acotadas , el error modelado \tilde{f}_t puede ser considerado acotado también (ver, por ejemplo [5][9][11]).

Para el caso del modelo que combina bien, como el análisis de Lyapunov [18] puede alcanzar el mismo resultado como en el *Corolario 1*. Pero en el caso del error modelado ($\tilde{f}_t \neq 0$), los términos de modificación robusta serán adicionados en la ley actualizada para asegurar la estabilidad [5][9] [11]. La modificación robusta normalmente depende los límites superiores del error modelado \tilde{f}_t . Diferente a las leyes adaptivas robustas, tales como zona muerta [9] y modificación $-\sigma$ [27], las leyes actualizadas no necesitan los límites superiores de las incertidumbres.

Teorema 16 *Si el error modelado $\tilde{f}_t(V_1^0, V_2^0)$ esta acotado como $\tilde{f}_t^T \Lambda_f \tilde{f}_t \leq \bar{\eta}(V_1^0, V_2^0)$, P es la solución de la ecuación de Riccati (4.12) con*

$$R = 2\bar{W}_1 + 2\bar{W}_2 + \Lambda_f, \quad Q = D_\sigma + \bar{u}D_\phi + Q_0$$

entonces la ley actualizada:

$$\begin{aligned} W_{k+1} &= W_k - \eta_k e(k) \sigma' V_k^T x(k) \\ V_{k+1} &= V_k - \eta_k e(k) \sigma^T [w_k(k) x(k)] \end{aligned} \tag{4.28}$$

puede hacer que el error de identificación converja a

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \int_0^T \|\Delta_t\|_{Q_0}^2 dt \leq \bar{\eta}(V_1^0, V_2^0)$$

Demostración. Se define una función de Lyapunov como, en vista de la desigualdad matricial

$$2\Delta_t^T P \tilde{f}_t \leq \Delta_t^T P \Lambda_f P \Delta_t + \tilde{f}_t^T \Lambda_f^{-1} \tilde{f}_t \leq \Delta_t^T P \Lambda_f P \Delta_t + \bar{\eta} (V_1^0, V_2^0)$$

Usando la ley actualizada , la derivada de la función de Lyapunov es

$$\dot{S}_t \leq \Delta_t^T [PA + A^T P + P (2\bar{W}_1 + 2\bar{W}_2 + \Lambda_f) P + (D_\sigma + \bar{u}D_\phi + Q_0)] \Delta_t - \Delta_t^T Q_0 \Delta_t + \bar{\eta}$$

Del error modelado tenemos

$$\dot{S}_t \leq -\Delta_t^T Q_0 \Delta_t + \bar{\eta} (V_1^0, V_2^0) \quad (4.29)$$

Integrando de 0 hasta T produce

$$V_T - V_0 \leq -\int_0^T \Delta_t^T Q_0 \Delta_t dt + \bar{\eta} T$$

Así que

$$\int_0^T \Delta_t^T Q_0 \Delta_t dt \leq V_0 - V_T + \bar{\eta} T \leq V_0 + \bar{\eta} T$$

es establecido ■

Propiedad 13 *El error de identificación convergerá al radio de la esfera o bola de límites acotados de \tilde{f}_t , y esto esta influenciado por las matrices conocidas previamente V_1^0 y V_2^0 . El teorema 12 muestra que V_1^0 y V_2^0 no influyen en la propiedad de estabilidad , podemos seleccionar cualesquiera valores para V_1^0 y V_2^0 en primera. Del teorema 12 conocemos el algoritmo (4.28) puede hacer el error de identificación convergente . V_1^0 y V_2^0 pueden ser seleccionados por los siguientes pasos off-line :*

1. Comenzar de cualesquiera valores iniciales para V_1^0 y V_2^0
2. Hacer identificación on-line con V_1^0 y V_2^0
3. Sean $V_{1,t}$ y $V_{2,t}$ como nuevas condiciones iniciales, i.e., $V_1^0 = V_{1,t}, V_2^0 = V_{2,t}$

4. Si el error de identificación decrece , repetir los procesos de identificación, ir a 2. De lo contrario parar la identificación off-line , ahora $V_{1,t}$ y $V_{2,t}$ son los valores finales para V_1^0 y V_2^0 .

Después obtenemos estos finales V_1^0 y V_2^0 , podemos empezar la identificación on-line con ellos.

Propiedad 14 *Puesto que la velocidad o índice de actualización en el algoritmo de aprendizaje es $K_i P$ ($i = 1 \dots 4$), y K_i puede ser seleccionada como cualquier matriz positiva ,el proceso de aprendizaje de la red neuronal dinámica es libre de las soluciones de las dos ecuaciones de Riccati. Estas dos ecuaciones de Riccati son propuestas para probar los resultados de estabilidad. Cuando usamos la ley actualizada para los pesos, solo necesitamos seleccionar buenos índices o velocidades de actualización $k_i = K_i P$.*

4.4. Resultado de Simulaciones

En esta sección se mostrarán las simulaciones para un sistema no lineal en tiempo continuo concerniente a un motor fuel injection, en el cual se tomarán como variables de interés a medir o sensar aunque de manera indirecta, la presión de la mezcla aire-combustible y la velocidad del motor en ralentí. Ambas mediciones son consideradas cuando el vehículo, se encuentra en reposo o en punto muerto. Para la realización de las simulaciones, se hará uso del lenguaje de programación Matlab-Simulink.

La operación del motor en ralentí de un vehículo en reposo o en punto muerto es un proceso de carácter no lineal que no es en absoluto su rango óptimo. Asumimos que la existencia de perturbaciones en la planta o el sistema, tales como el comportamiento del aire en la etapa de compresión , el cambio de neutral a transmisión automática, en aplicación y liberación de cargas eléctricas, y dirección de potencia interna, no son directamente medidos. El sistema del modelo del motor dinámico en dos entradas y dos salidas [10]:

$$\begin{aligned} \dot{P} &= k_P (\dot{m}_{ai} - \dot{m}_{ao}), & \dot{N} &= k_N (T_i - T_L) \\ \dot{m}_{ai} &= (1 + k_{m1}\theta + k_{m2}\theta^2) g(P), & \dot{m}_{ao} &= -k_{m3}N - k_{m4}P + k_{m5}NP + k_{m6}NP^2 \end{aligned}$$

Los parámetros del modelo del motor de 1.6 litros, motor de 4 cilindros fuel injection son:

$$g(P) = \begin{cases} 1 & P < 50,6625 \\ 0,0197\sqrt{101,325P - P^2} & P \geq 50,6625 \end{cases}$$

$$T_i = -39,22 + 325024m_{ao} - 0,0112\delta^2 + 0,635\delta + \frac{2\pi}{60} (0,0216 + 0,000675\delta) N - \left(\frac{2\pi}{60}\right)^2 0,000102N^2$$

$$T_L = \left(\frac{N}{263,17}\right)^2 + T_d, \quad m_{ao} = \dot{m}_{ao}(t - \tau)/(120N)$$

$$k_P = 42,40, \quad k_N = 54,26, \quad k_{m1} = 0,907, \quad k_{m2} = 0,0998$$

$$k_{m3} = 0,0005968, \quad k_{m4} = 0,0005341, \quad k_{m5} = 0,000001757, \quad \tau = 45/N$$

La salida del sistema con colector de presión P (kPa) y velocidad de motor N (rpm). Las entradas de control son ángulo de la válvula de admisión θ (grado) y la chispa de avance o encendido δ (grado). Las perturbaciones que actúan en el motor en la forma de torque de accesorio no medido T_d (N-m). La variable \dot{m}_{ai} y \dot{m}_{ao} se refieren a la masa de aire que fluye dentro y fuera del colector. m_{ao} es la masa de aire en el cilindro. El parámetro τ es un tiempo de retardo de transporte dinámico. La función $g(P)$ es un colector de presión de la función de influencia. T_i es el torque desarrollado internamente en el motor, T_L es el torque de carga. Si definimos $x = (P, N)^T$, $u = (\theta, \delta)^T$, entonces la dinámica de la velocidad del motor en ralentí del vehículo parado o en reposo es

$$\dot{x}_t = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} f_1(x, u) \\ f_2(x, u) \end{pmatrix}. f_1 \text{ y } f_2 \text{ son consideradas ser desconocidas y solo } x \text{ y } u \text{ son}$$

medibles. Para realizar la simulación, seleccionamos la entrada como $\delta = 30 \sin \frac{t}{2}$, θ es una onda diente de sierra con amplitud de 10, a una frecuencia de $\frac{1}{2}Hz$, T_d es una onda cuadrada con amplitud de 20, a una frecuencia de $\frac{1}{4}Hz$.

4.4.1. Simulaciones y Gráficas

(A) Iniciaremos la simulación con una red neuronal monocapa de la siguiente forma:

Se compara la red neuronal con la red monocapa como sigue:

$$\hat{x}_t = A\hat{x}_t + W_{1,t}\sigma(\hat{x}_t) + W_{2,t}\phi(\hat{x}_t)\pi(u_t)$$

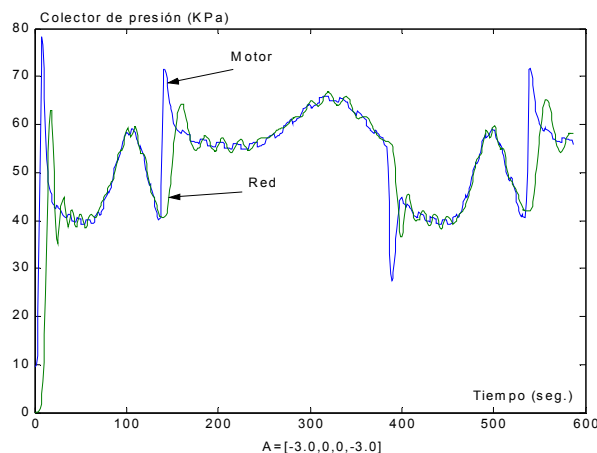


Figura 4.1: Presión en el motor.

$$A = \begin{bmatrix} -0,1 & 0 \\ 0 & -0,1 \end{bmatrix} \quad x_0 = [0, 0]^T$$

$$W_{1,0} = W_{2,0} = \begin{bmatrix} 1 & 10 & 10 \\ 1 & 0 & 0 \end{bmatrix} \quad K_1 P = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Observaremos el comportamiento de las simulaciones, tanto para la etapa del colector de presión como la que corresponde a la velocidad del motor, en las que se podrán distinguir las señales tanto del sistema como el de la red neuronal, y cuya finalidad es establecer las ventajas de utilizar redes neuronales en el proceso de identificación de sistemas dinámicos complejos desconocidos o en los que son difíciles de medir ciertos parámetros. En este primer paso, se harán variaciones de los parámetros de la matriz A para analizar cuáles son las simulaciones más óptimas y adecuadas para nuestro diagnóstico.

De estas simulaciones podemos observar que las que corresponden a las figuras 4.1 y 4.2 son las que mejor estabilidad y comportamiento presentan, es decir, la red neuronal se mantiene muy semejante y aprende la respuesta emitida por la planta lo que indica que el seguimiento de la trayectoria es bastante bueno, mantenido la estabilidad que se necesita para poder concluir, que la convergencia y seguimiento de la red neuronal es bastante aceptable.

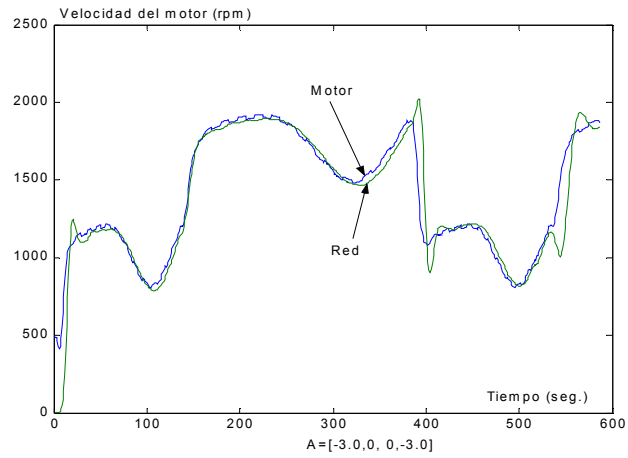


Figura 4.2: Velocidad en el motor.

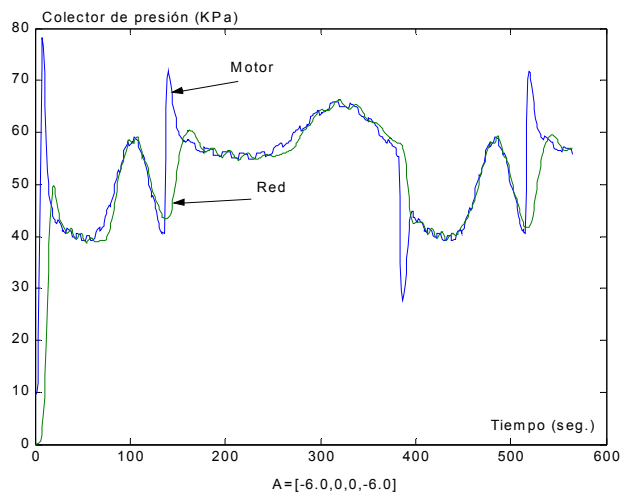


Figura 4.3: Presión en el motor.

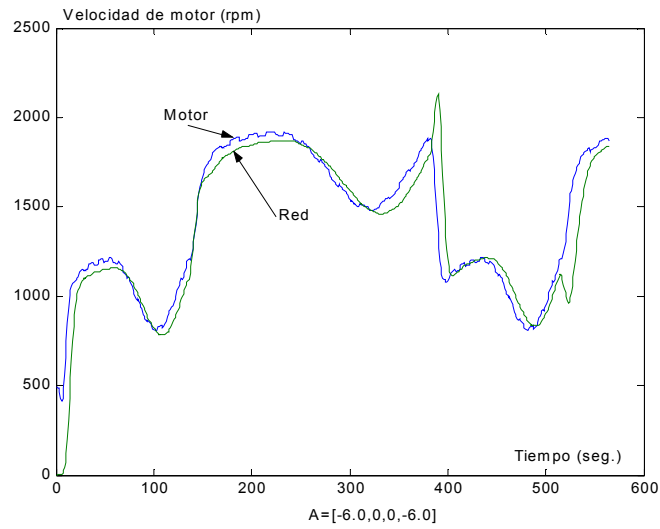


Figura 4.4: Velocidad en el motor.

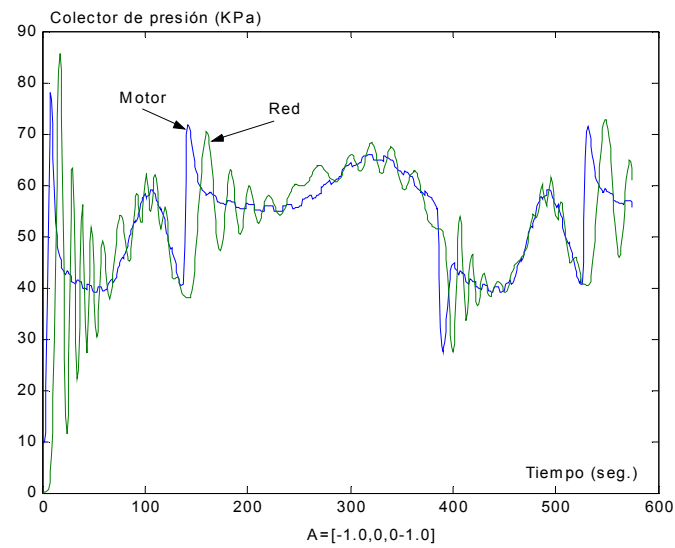


Figura 4.5: Presión en el motor.

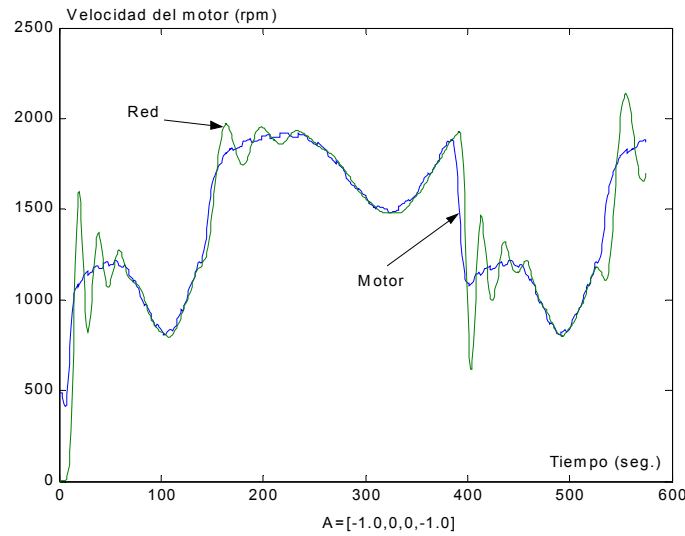


Figura 4.6: Velocidad en el motor.

A continuación se muestran estas mismas gráficas solo que ahora se procede a variar las condiciones iniciales para la matriz de pesos W :

Observando detenidamente las gráficas anteriores, es posible constatar que las variaciones no son muy significativas, pero si se pueden notar ciertas variaciones, quedando las figuras 4.7 y 4.8, como las más adecuadas para el proceso de identificación, ya que presentan mayor estabilidad y convergencia de las trayectorias de la red neuronal con respecto a la señal del sistema no lineal dinámico, pudiendo concluir que la efectividad de la red neuronal mono capa utilizada aquí, es muy positiva en el proceso de identificación.

(B) Ahora procederemos a realizar semejantes simulaciones al mismo sistema pero utilizando redes neuronales multicapa, para analizar y comprobar la mayor versatilidad y capacidad que tienen sobre las redes neuronales monocapa.

Seleccionamos la red neuronal dinámica como

$$\dot{\hat{x}}_t = A\hat{x}_t + W_{1,t}\sigma(V_{1,t}\hat{x}_t) + W_{2,t}\phi(V_{2,t}\hat{x}_t)\pi(u_t)$$

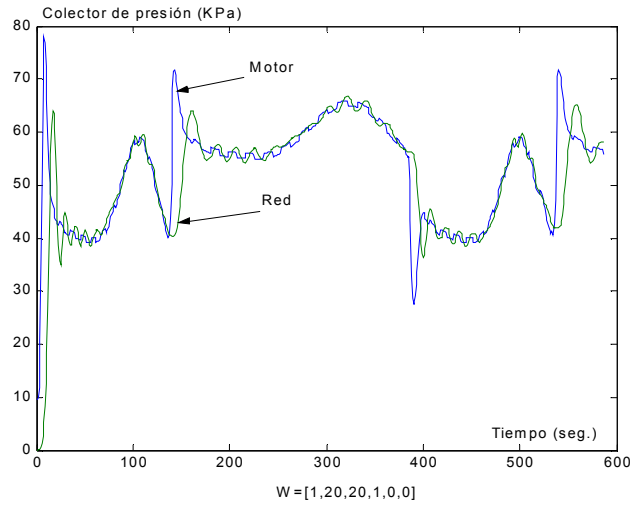


Figura 4.7: Presión en el motor.

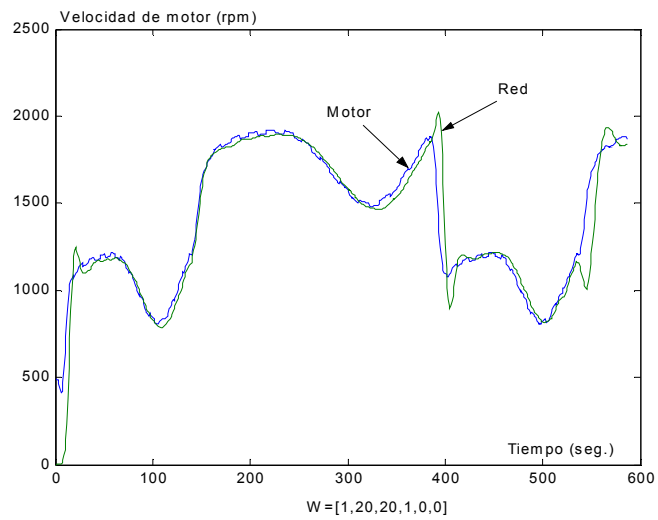


Figura 4.8: Velocidad en el motor.

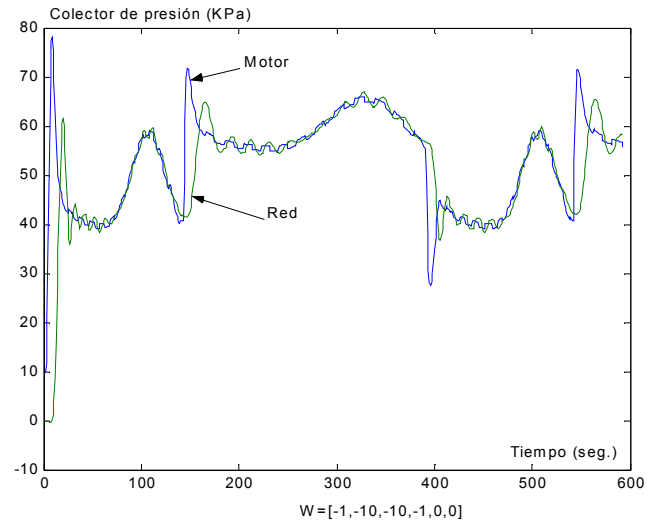


Figura 4.9: Presión en el motor.

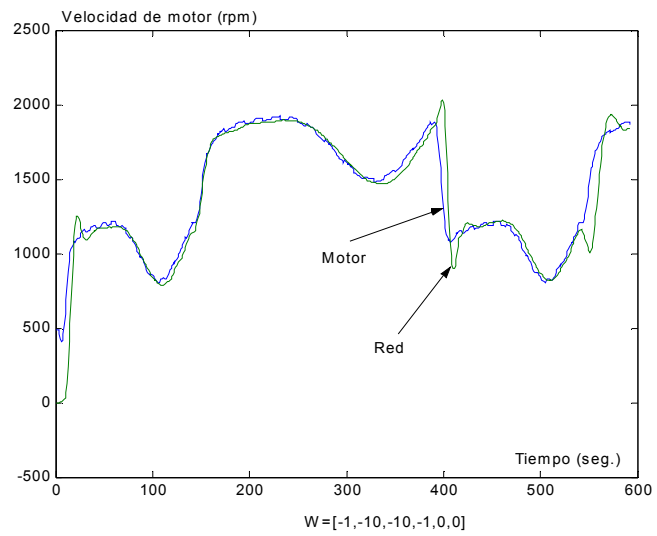


Figura 4.10: Velocidad en el motor.

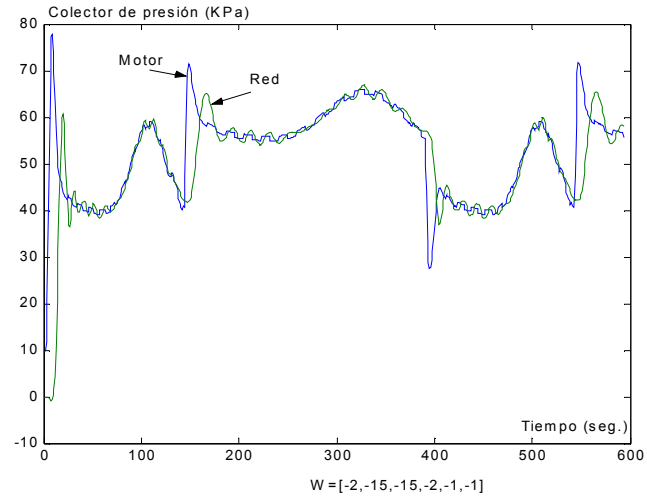


Figura 4.11: Presión en el motor.

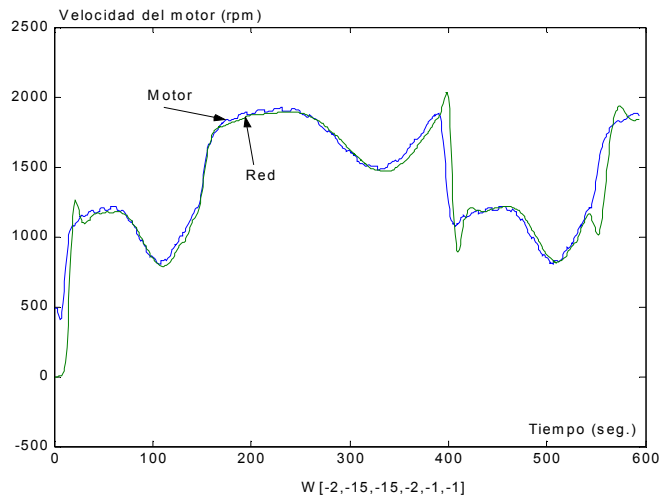


Figura 4.12: Velocidad en el motor.

donde $A = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix}$, $\hat{x}_0 = [0, 0]^T$, $W_{1,t}$ y $W_{2,t} \in \mathfrak{R}^{2 \times 3}$, $V_{1,t}$ y $V_{2,t} \in \mathfrak{R}^{3 \times 2}$. Las funciones sigmoideas son $\sigma(x_i) = \frac{2}{1+e^{-2x_i}} - 0,5$, $\phi(x_i) = \frac{0,2}{1+e^{-0,2x_i}} - 0,05$. $\pi(u_t) = u_t$. $D_\sigma = \text{diag}[D_{\sigma_1}, D_{\sigma_2}, D_{\sigma_3}]$, $D_\phi = \text{diag}[D_{\phi_1}, D_{\phi_2}, D_{\phi_3}]$, $u_3 = 0$,

$$D_{\sigma_i} = \frac{4e^{-2Z_{1,i}}}{(1 + e^{-2Z_{1,i}})^2}, \quad Z_{1,i} = (V_{1,t}\hat{x})_i$$

$$D_{\sigma_i} = \frac{0,04e^{-0,2Z_{2,i}}}{(1 + e^{-0,2Z_{2,i}})^2} u_i, \quad Z_{2,i} = (V_{2,t}\hat{x})_i$$

Seleccionamos $K_1P = K_2P = K_3P = K_4P = 2I$.

Escogemos los valores iniciales $V_1^0 = V_2^0 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$, usando la ley de aprendizaje

podemos obtener $V_{1,20} = \begin{bmatrix} 0 & 0,94 \\ 0,45 & 0,25 \\ -0,44 & 0,49 \end{bmatrix}$, $V_{2,20} = \begin{bmatrix} 0,25 & 0,83 \\ 0,68 & 0,36 \\ 0,29 & 0,52 \end{bmatrix}$ después de 20s. Ahora

los usamos como los nuevos valores V_1^0 y V_2^0 , entrenando la red neuronal 30s con ley de

aprendizaje, tenemos $V_{1,30} = \begin{bmatrix} 0,001 & 0,94 \\ 0,47 & 0,23 \\ -0,45 & 0,43 \end{bmatrix}$, $V_{2,30} = \begin{bmatrix} 0,25 & 0,80 \\ 0,71 & 0,35 \\ 0,28 & 0,48 \end{bmatrix}$. Podemos ver que

$V_{1,t}$ y $V_{2,t}$ no cambian mucho, así que detenemos nuestro procedimiento de búsqueda para V_1^0 y V_2^0 , usamos $V_{1,30}$ y $V_{2,30}$ como los nuevos V_1^0 y V_2^0 . Los resultados de identificación son mostrados en

$$A = A = \begin{bmatrix} -0,1 & 0 \\ 0 & -0,1 \end{bmatrix}$$

$$x_0 = [0, 0]^T$$

$$W_{1,0} = W_{2,0} = \begin{bmatrix} 1 & 10 & 10 \\ 1 & 0 & 0 \end{bmatrix},$$

$$V_1^0 = V_2^0 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

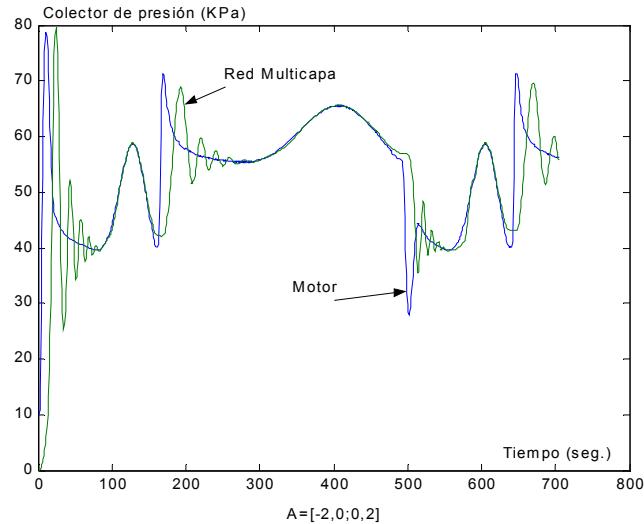


Figura 4.13: Presión en el motor.

$$K_1P = K_2P = K_3P = K_4P = 2I = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Realizamos nuestras simulaciones para el mismo sistema complejo y tomando en consideración las mismas señales a la salida, es decir, la presión en el motor así como su velocidad y los resultados de las simulaciones utilizando redes neuronales multicapa al variar parámetros, en este caso, la matriz A son los siguientes:

Observando cuidadosamente las gráficas anteriores, podemos concluir, que las que corresponden a las figuras 4.13 y 4.14, son las más adecuadas para el proceso de identificación, por la estabilidad y convergencia que presentan las redes neuronales multicapa con respecto a la señal del sistema. Pudiéndose notar que el seguimiento que presentan las redes, es bastante aceptable, sufriendo algunas variaciones en el inicio del proceso, esto por los transitorios que se presentan, sin embargo, esto nos indica que la identificación llevada a cabo por las redes neuronales, es bastante buena.

A continuación se muestran las mismas gráficas, pero solo que ahora se han agregado

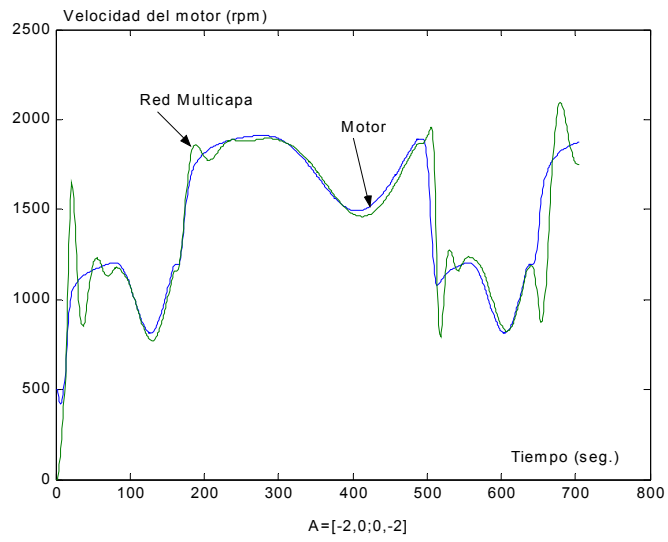


Figura 4.14: Velocidad en el motor.

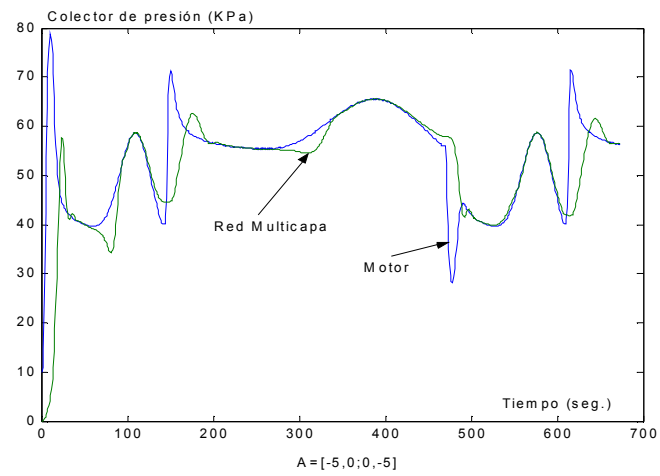


Figura 4.15: Presión en el motor.

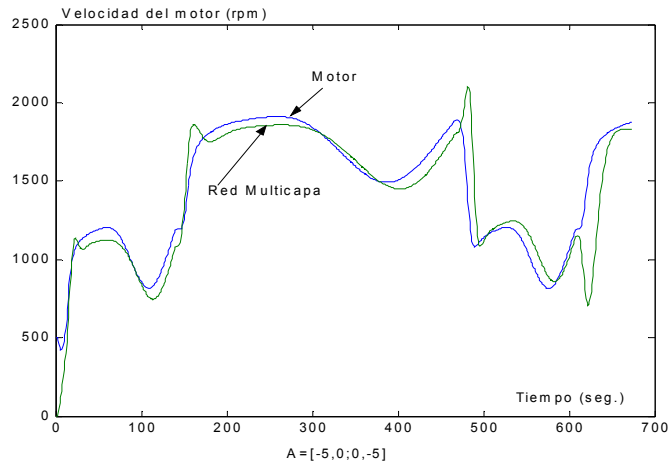


Figura 4.16: Velocidad en el motor.

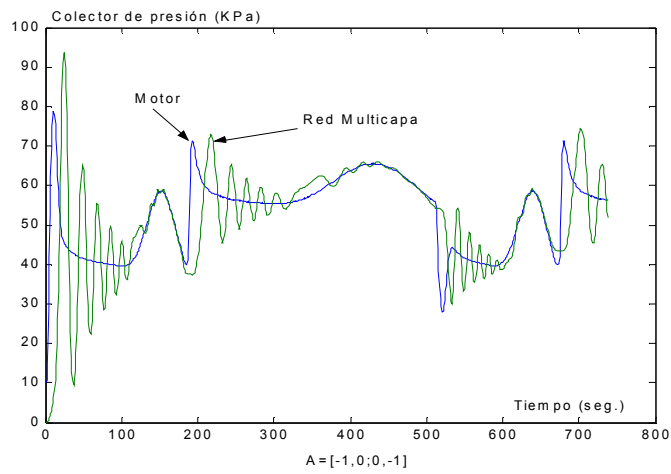


Figura 4.17: Presión en el motor.

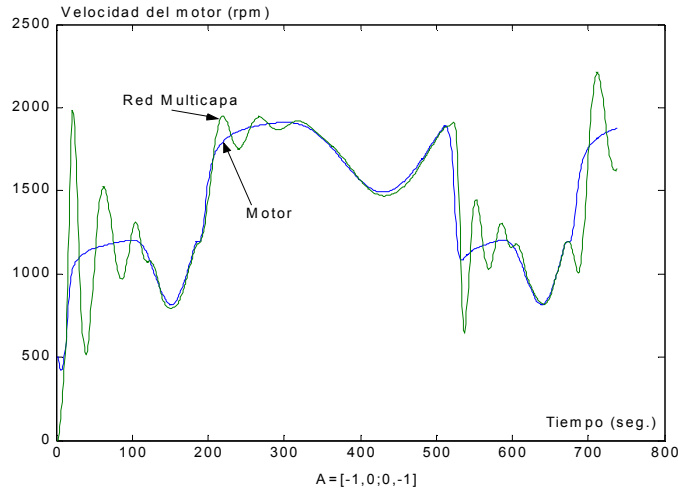


Figura 4.18: Velocidad en el motor.

fuentes para generar ruido o introduciendo ruido a las redes multicapa y observar su comportamiento:

Con base a los resultados obtenidos en nuestras simulaciones anteriores, se puede observar que dependiendo de la magnitud o amplitud de nuestra señal espuria así como la frecuencia a la que es inyectada, se presentan variaciones en cuanto a la amplitud de ambas señales, es decir, se separan aunque por lo que respecta a las gráficas de las figuras 4.19 y 4.20, el seguimiento de la trayectoria de las redes multicapa con respecto a la del sistema, es bastante bueno, por lo que el proceso de identificación es muy significativo. Aunque es necesario considerar, que al suministrar ruido con mayor amplitud y frecuencia, se ocasionará inestabilidad en el proceso de identificación, como se puede observar en las gráficas 4.23 y 4.24.

Las figuras 4.25, 4.26, 4.27 y 4.28 muestran como V_1^0 y V_2^0 influyen en el error de identificación, los valores iniciales corresponden a $V_1^0 = V_2^0 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$, mientras que los

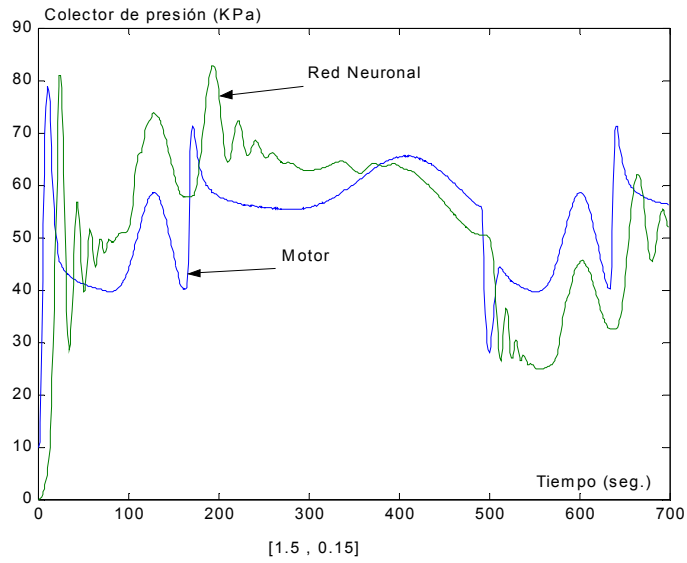


Figura 4.19: Presión en el motor.

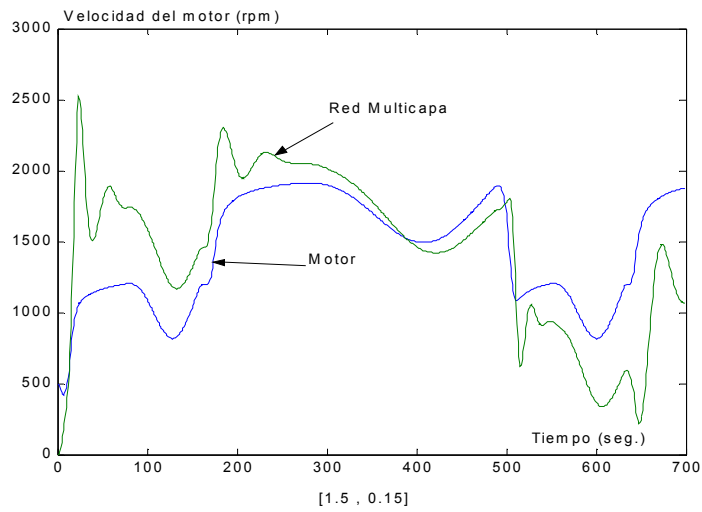


Figura 4.20: Velocidad del motor.

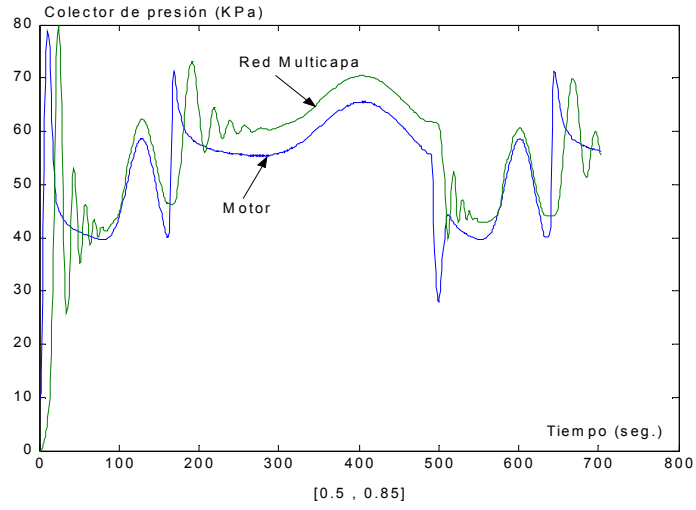


Figura 4.21: Presión en el motor.

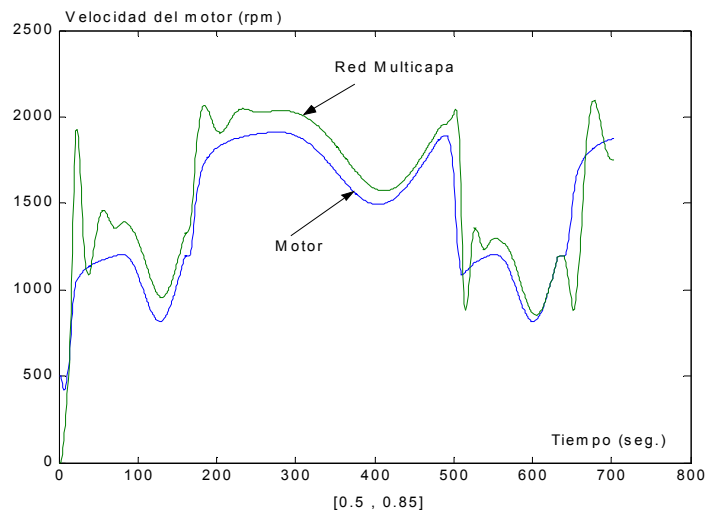


Figura 4.22: Velocidad en el motor.

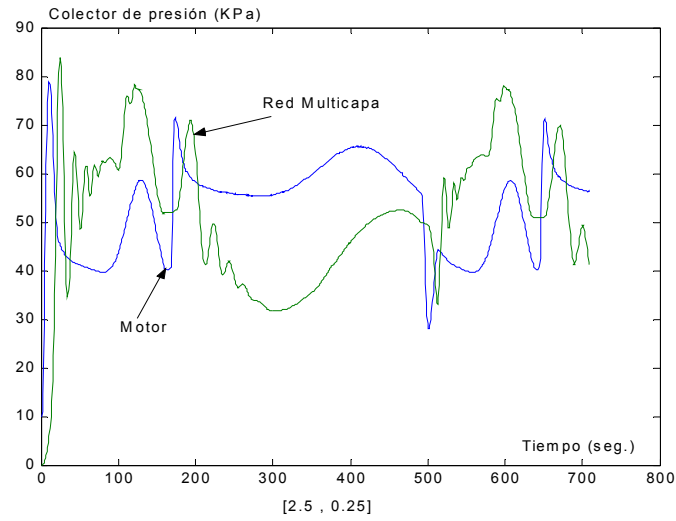


Figura 4.23: Presión en el motor.

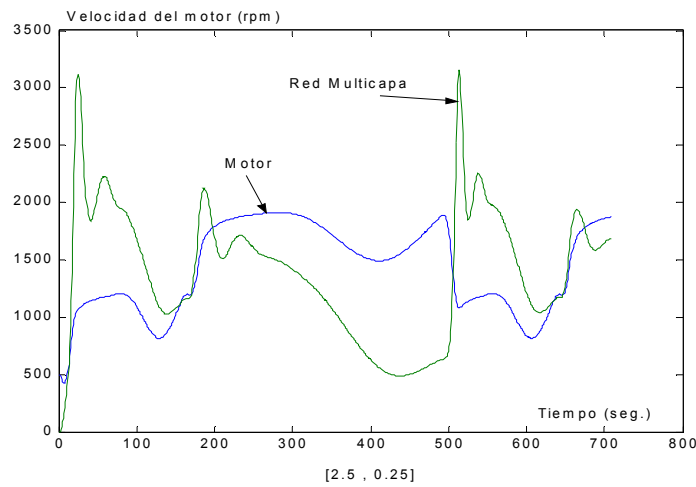


Figura 4.24: Velocidad en el motor.

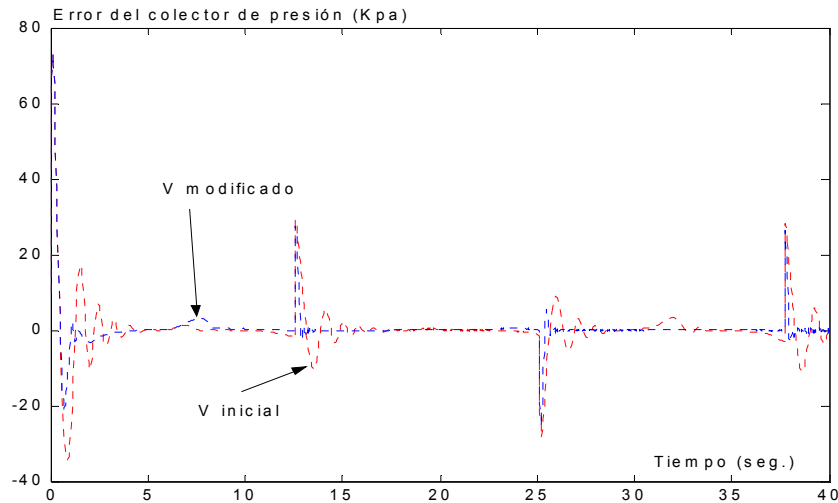


Figura 4.25: Error de presión al variar V .

valores modificados son el error de identificación con $V_1^0 = V_{1,30}$ y $V_2^0 = V_{2,30}$. Así podemos ver que adecuadamente V_1^0 y V_2^0 pueden ser encontradas por el procedimiento de entrenamiento o aprendizaje.

(C) Comparación entre redes mono-capa y multi-capa.

Ahora se comparan la red neuronal con la red mono-capa como sigue

$$\dot{\hat{x}}_t = A\hat{x}_t + W_{1,t}\sigma(\hat{x}_t) + W_{2,t}\phi(\hat{x}_t)\pi(u_t)$$

todas las condiciones son las mismas que las redes neuronales multicapa, la ley de aprendizaje es usada en [20]. Las figuras 4.29, 4.30, 4.31 y 4.32 muestran la compensación del error de identificación con redes neuronales mono-capa y multi-capa.

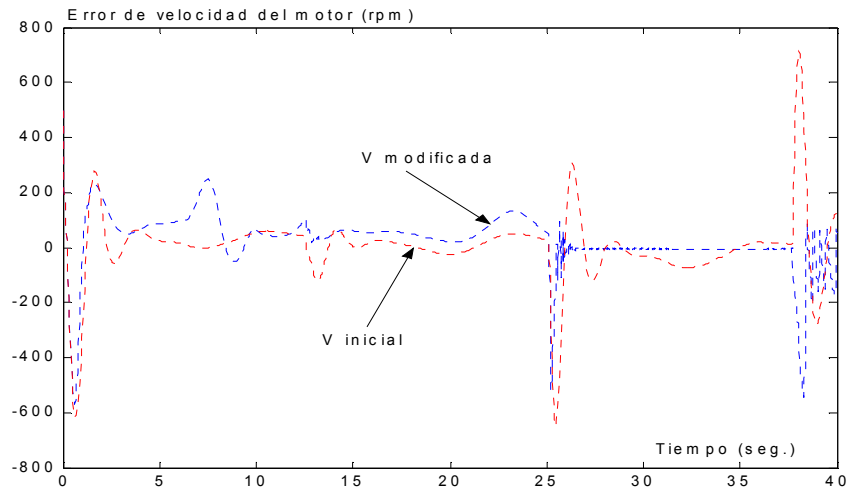


Figura 4.26: Error de la velocidad al variar V.

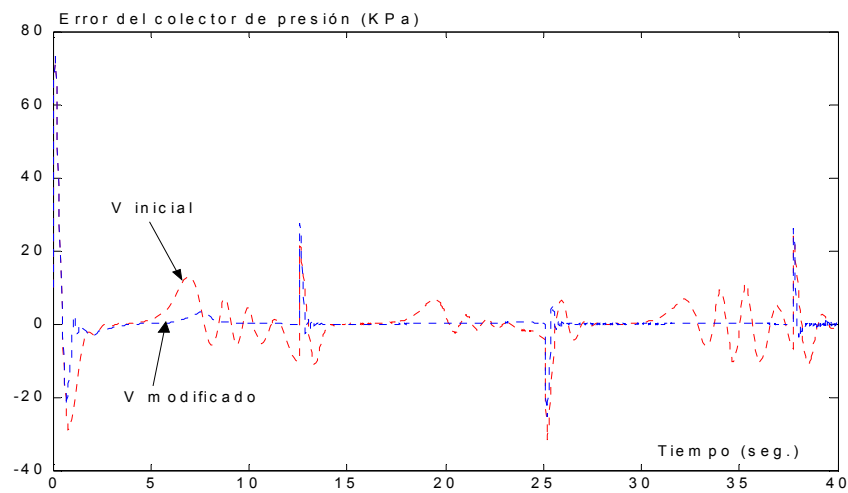


Figura 4.27: Error de presión al variar V.

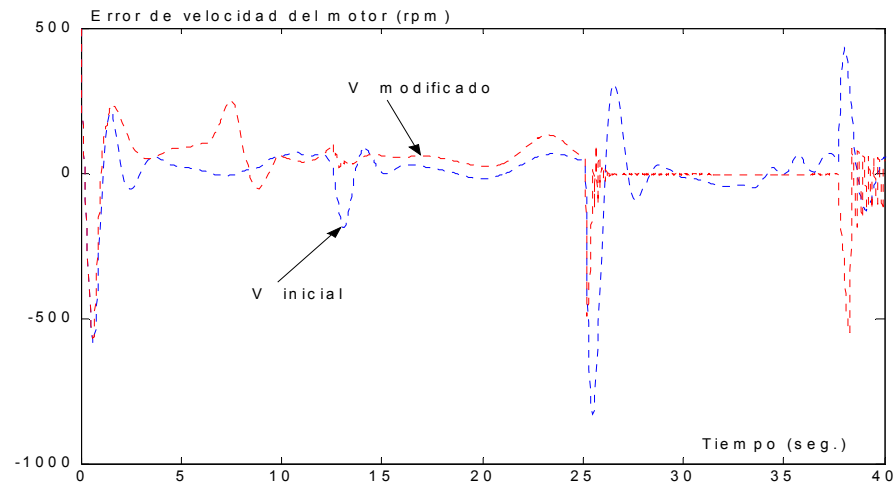


Figura 4.28: Error de la velocidad al variar V.

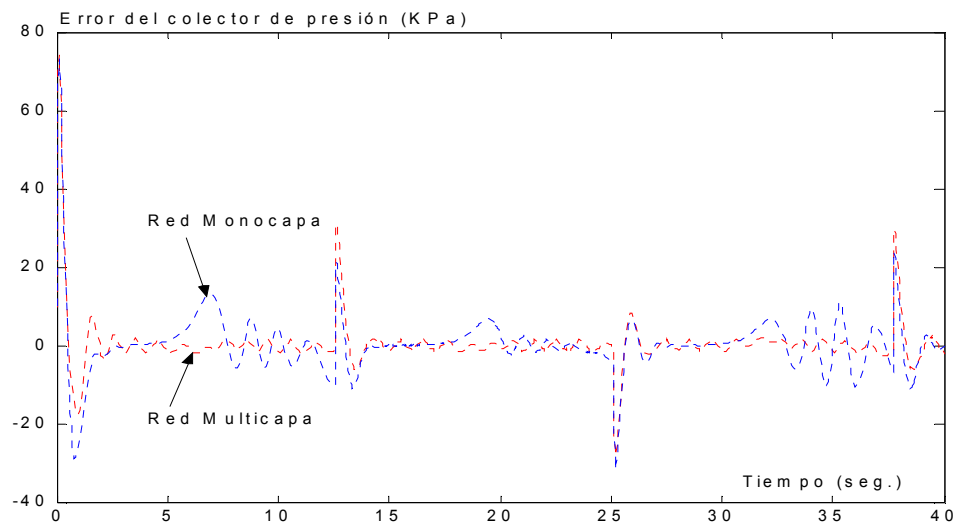


Figura 4.29: Error de presión con redes monocapa y multicapa.

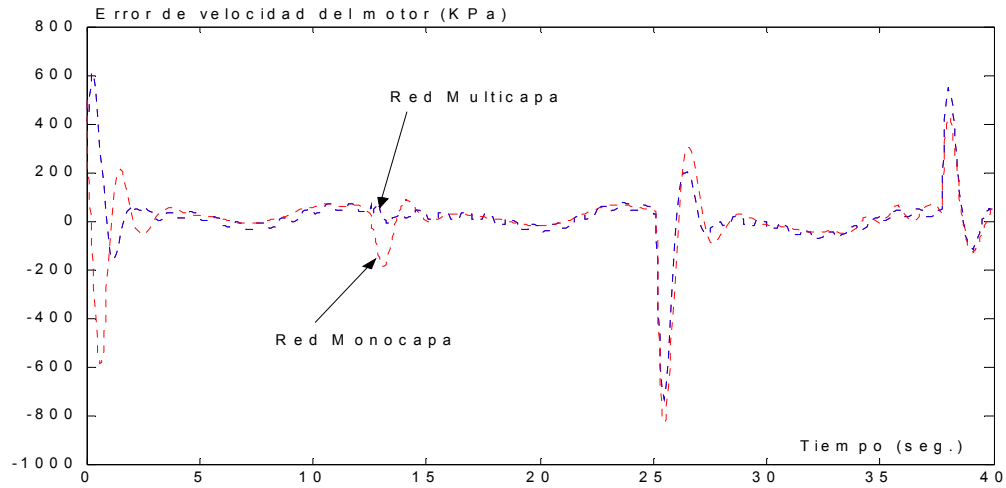


Figura 4.30: Error de la velocidad con redes monocapa y multicapa.

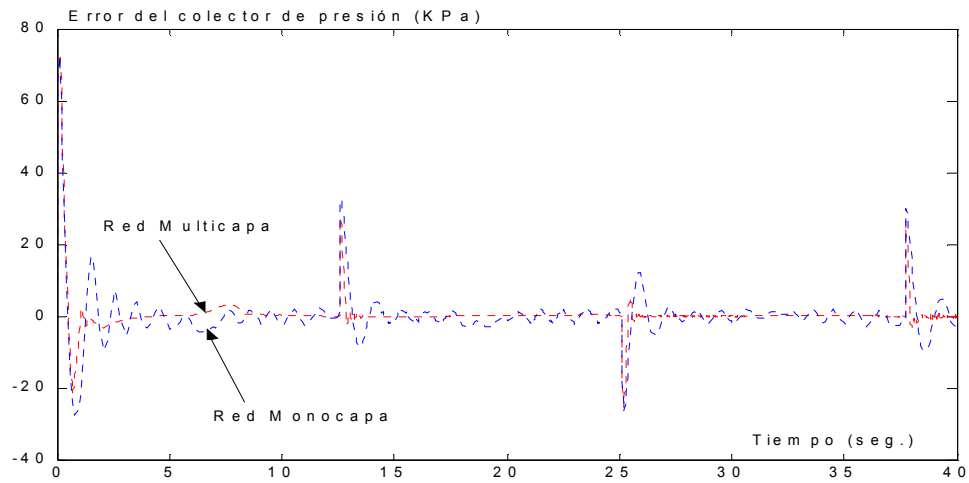


Figura 4.31: Error de presión con redes monocapa y multicapa.

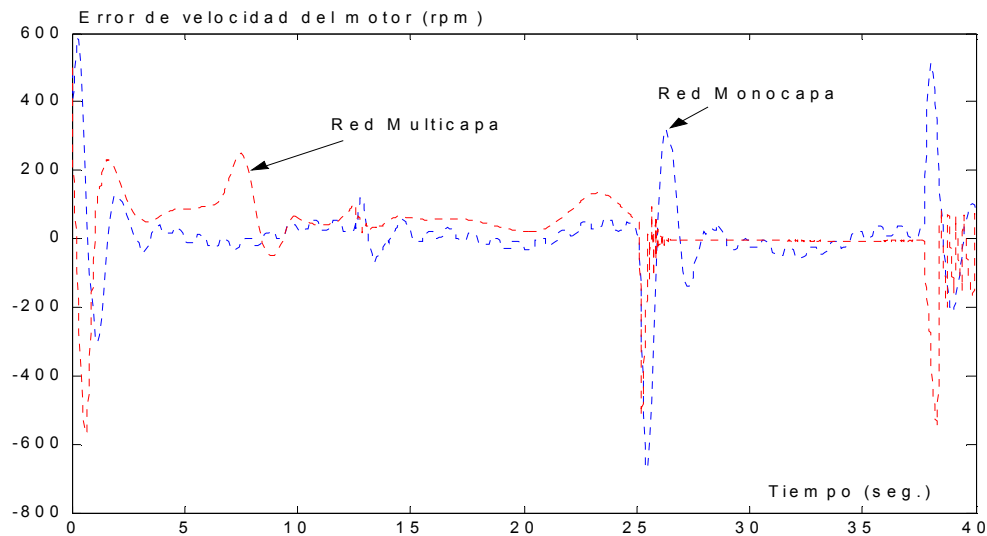


Figura 4.32: Error de la velocidad con redes monocapa y multicapa.

4.5. Comentarios y Conclusiones:

Para la simulación del sistema, se usaron programas en matlab-simulink tanto para redes neuronales monocapa como para redes neuronales multicapa, tal y como se muestra en los diagramas en simulink de las figuras 4.33 y 4.34, en las que se puede observar los bloques que interactúan para llevar a cabo las simulaciones en la identificación del sistema. Pudiéndose observar la mayor complejidad que presentan los modelos para las redes multicapa. La razón de haber utilizado estos diagramas, fue la necesidad de simplificar las simulaciones del que consideramos un sistema no lineal en tiempo continuo, para poder llevar a cabo, satisfactoriamente el proceso de identificación y que en base al comportamiento de las señales obtenidas a la salida de los modelos neuronales, podemos constatar, como buenos los resultados obtenidos.

* Se puede ver claramente que las redes neuronales dinámicas multicapa son más versátiles y eficientes que las redes neuronales monocapa, y que resultan ser bastante robustas con

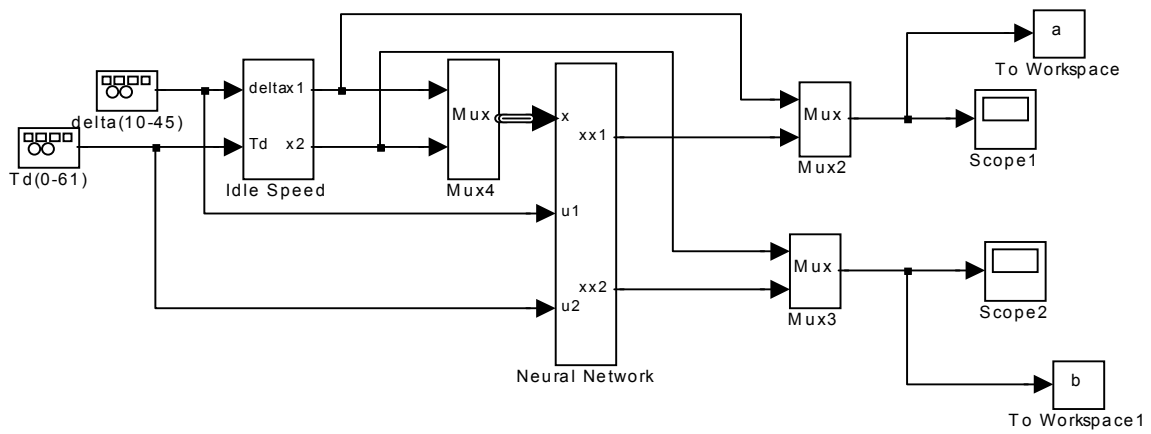


Figura 4.33: Diagrama Simulink para simulación con redes monocapa.

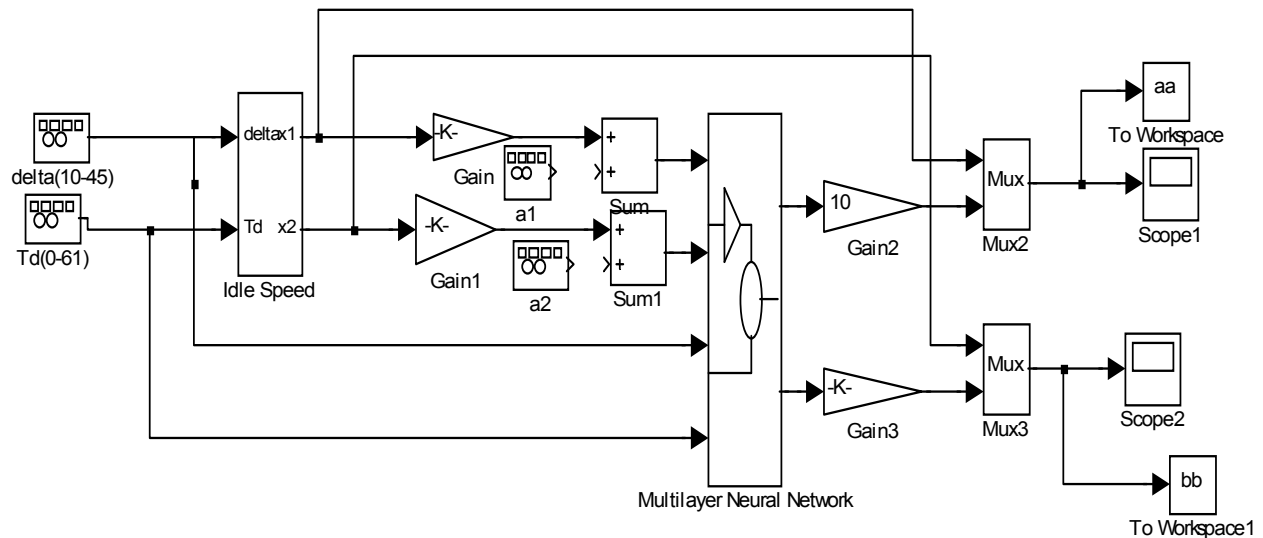


Figura 4.34: Diagrama Simulink para simulación con redes multicapa.

respecto a incertidumbres acotadas.

* En el proceso de identificación, es posible introducir diferentes tipo de señales al sistema modelado, para poder observar cuáles son las respuestas que entrega el sistema modelado, pudiéndose observar claramente, que las señales que entregan las redes, son muy similares entre si, sin que afecte demasiado, la información de entrada que se necesita para emitir una conclusión.

* Lo indicado anteriormente puede ser usado como una validación del tipo de información obtenida, debido a la variedad de señales que se utilizaron; pudiendo concluir, que el sistema permanecerá estable si las señales introducidas y obtenidas se mantienen dentro de una rango de acotamiento, es decir, que el sistema es *estable de entrada-estado*, si las señales de entrada y de salida permanecen acotadas, logrando con esto que el sistema sea estable.

Capítulo 5

ISS para identificación con redes neuronales en tiempo discreto

El método de representación de sistemas dinámicos por ecuaciones vectoriales diferenciales o en diferencias esta actualmente bien establecido en teoría de sistemas y en aplicaciones de bastantes clases de sistemas. Por ejemplo, la ecuación diferencial:

$$\begin{aligned} \frac{dx(t)}{dt} \triangleq \dot{x}(t) &= \Phi [x(t), u(t)] & t \in \mathbb{R}^+ \\ y(t) &= \Psi [x(t)] \end{aligned} \quad (5.0)$$

donde: $x(t) \triangleq [x_1(t), x_2(t), \dots, x_n(t)]^T$, $u(t) \triangleq [u_1(t), u_2(t), \dots, u_p(t)]^T$ y $y(t) \triangleq [y_1(t), y_2(t), \dots, y_m(t)]$ representando a un sistema de p entradas m salidas de orden n con $u_i(t)$ representando las entradas, $x_i(t)$ las variables de estado, y $y_i(t)$ las salidas del sistema. Φ y Ψ son mapeos no lineales estáticos definidos como $\Phi : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ y $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}^m$. El vector $x(t)$ denota el estado del sistema al tiempo t y está determinado por el estado al tiempo $t_0 < t$ y la entrada u definida sobre el intervalo $[t_0, t)$. La salida $y(t)$ esta determinada completamente por el estado del sistema en el tiempo t . La ecuación 5.0 es referida como la representación *entrada-estado-salida* del sistema. En este capítulo estaremos involucrados con sistemas en tiempo discreto que pueden ser representados por las ecuaciones en diferencias correspondientes a

la ecuación diferencial dada en (5.0). Estas tomarán la forma de:

$$\begin{aligned}x(k+1) &= \Phi[x(k), u(k)] \\y(k) &= \Psi[x(k)]\end{aligned}\tag{5.1}$$

donde $u(\cdot)$, $x(\cdot)$, y $y(\cdot)$ son sucesiones en tiempo discreto. Como ya se mencionó anteriormente, sin embargo, estos resultados o expresiones pueden ser extendidos a sistemas en tiempo continuo. Si el sistema descrito por (5.1) es considerado ser lineal e invariante en el tiempo, las ecuaciones que gobiernan este comportamiento puede ser expresado como:

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) \\y(k) &= Cx(k)\end{aligned}\tag{5.2}$$

donde $A, B,$ y C son matrices $(n \times n)$, $(n \times p)$ y $(m \times n)$ respectivamente. El sistema es entonces parametrizado por la terna $\{C, A, B\}$. La teoría de sistemas lineales invariantes en el tiempo, cuando $C, A,$ y B son conocidas, esta bien desarrollado y conceptos tales como controlabilidad, estabilidad y observabilidad de tales sistemas han sido estudiados extensamente en las pasadas tres décadas. Métodos para determinar la entrada de control $u(\cdot)$ para optimizar un criterio de realización son también bien conocidos.

Cuando las funciones Φ y Ψ en (5.1), o matrices $A, B,$ y C en (5.2), son desconocidas, el problema de identificación del sistema desconocido surge [12]. Este puede ser formalmente indicado como sigue [1]:

La entrada y salida de un sistema o planta dinámica causal en tiempo discreto e invariante en el tiempo son $u(\cdot)$ y $y_e(\cdot)$, respectivamente, donde $u(\cdot)$ es una función uniformemente acotada en el tiempo. La planta o el sistema se supone ser estable con parametrización conocida pero con valores desconocidos de los parámetros. El objetivo es construir un modelo de identificación adecuado (figura 5.1) que cuando cuando este sujeta a la misma entrada $u(k)$ de la planta, produzca una salida $y_e(\cdot)$ que aproxime $y(k)$ en el sentido descrito por (5.2).

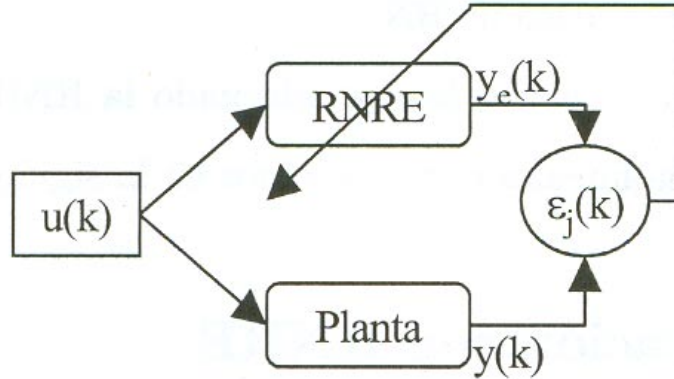


Figura 5.1: Esquema de identificación.

5.1. Identificación de sistemas con Redes Neuronales Recurrentes mono-capa en tiempo discreto

Consideremos una red neuronal recurrente en tiempo discreto en la forma de

$$\beta \hat{x}(k+1) = A \hat{x}(k) + \sigma [W_1(k) x(k)] + \phi [W_2(k) x(k)] U(k) \quad (5.10)$$

donde $\hat{x}(k) \in \mathfrak{R}^n$ representan el estado interno de la red neuronal. La matriz $A \in \mathfrak{R}^{n \times n}$ es una matriz estable que será especificada después. Las matrices $W_1(k), W_2(k) \in \mathfrak{R}^{n \times n}$ son los pesos de la red neuronal. β es una constante positiva $\beta > 1$ que es un parámetro de diseño. $U(k) = [u_1, u_2 \cdots u_m, 0, \cdots, 0]^T \in \mathfrak{R}^n$. $|U(k)|^2 \leq \bar{u}$. $\sigma(\cdot) \in \mathfrak{R}^m$ son funciones vectoriales

sigmoidales , $\phi(\cdot)$ es $\mathfrak{R}^{n \times n}$ una matriz diagonal,

$$\begin{aligned} \sigma [W_1(k) x(k)] &= \left[\sigma_1 \left(\sum_{j=1}^n w_{1j}^1 x_j \right), \sigma_2 \left(\sum_{j=1}^n w_{2j}^1 x_j \right), \dots, \sigma_n \left(\sum_{j=1}^n w_{nj}^1 x_j \right) \right]^T \\ \phi [W_2(k) x(k)] U(k) &= \left[\phi_1 \left(\sum_{j=1}^n w_{1j}^2 x_j \right) u_1, \phi_2 \left(\sum_{j=1}^n w_{2j}^2 x_j \right) u_2, \dots, \phi_n \left(\sum_{j=1}^n w_{nj}^2 x_j \right) u_n \right]^T \end{aligned} \quad (5.11)$$

El sistema no lineal identificado esta representado como (3.20). Asumimos que la planta es estable acotada a la entrada-acotada a la salida (BIBO) , i.e., $x(k)$ y $u(k)$ son acotados . De acuerdo al teorema de Stone-Weierstrass cite: Cybenko, esta función diferenciable general no lineal puede ser escrita como

$$\beta x(k+1) = Ax(k) + \sigma [W_1^* x(k)] + \phi [W_2^* x(k)] U(k) + \mu(k) \quad (5.12)$$

donde W_1^* y W_2^* son pesos constantes que pueden minimizar $\mu(k)$, $\mu(k)$ es el error modelado. De modo que σ y ϕ son funciones acotadas, $\mu(k)$ es acotada como $\mu^2(k) \leq \bar{\mu}$, $\bar{\mu}$ es una constante positiva desconocida. La neuro identificación del error esta definido como

$$e(k) = \hat{x}(k) - x(k)$$

De (5.12) y (5.10)

$$\begin{aligned} \beta e(k+1) &= Ae(k) + \{ \sigma [W_1(k) x(k)] - \sigma [W_1^* x(k)] \} \\ &+ \{ \phi [W_2(k) x(k)] - \phi [W_2^* x(k)] \} U(k) - \mu(k) \end{aligned} \quad (5.13)$$

Usando series de Taylor alrededor de los puntos de $W_1(k) x(k)$

$$\sigma [W_1(k) x(k)] - \sigma [W_1^* x(k)] = \sigma' [W_1(k) x(k)] \widetilde{W}_1(k) x(k) + \varepsilon_1(k)$$

donde $\widetilde{W}_1(k) = W_1(k) - W_1^*$, $\varepsilon_1(k)$ es un error de aproximación de segundo orden . σ' es la derivada de la función de activación no lineal $\sigma(\cdot)$ en el punto de $W_1(k)$. Así que ϕ es una función de activación sigmoideal , $\varepsilon_1(k)$ esta cotada como $\|\varepsilon_1(k)\|^2 \leq \bar{\varepsilon}_1$, $\bar{\varepsilon}_1$ es una constante positiva desconocida. Similarmente

$$\phi [W_2(k) x(k)] U(k) - \phi [W_2^* x(k)] U(k) = u(k) \phi' [W_2(k) x(k)] \widetilde{W}_2(k) x(k) + \varepsilon_2(k)$$

donde $u(k) = \text{diag}(u_i)$, $\widetilde{W}_2(k) = W_2(k) - W_2^*$, $\|\varepsilon_2(k)\|^2 \leq \bar{\varepsilon}_2$, $\bar{\varepsilon}_2$ es una constante positiva desconocida. Así que

$$\beta e(k+1) = Ae(k) + \sigma' \widetilde{W}_1(k) x(k) + u(k) \phi' \widetilde{W}_2(k) x(k) + \zeta(k) \quad (5.14)$$

donde $\zeta(k) = \varepsilon_1(k) + \varepsilon_2(k) - \mu(k)$. El siguiente teorema da un algoritmo de aprendizaje estable para redes neuronales mono-capa en tiempo discreto.

Teorema 17 *Si usamos la red neuronal mono-capa (5.10) para identificar plantas no lineales (4.1 o 3.20) y los valores propios de A están seleccionados como $-1 < \lambda(A) < 0$, el siguiente gradiente de ley actualizada sin modificación robusta puede hacer el error de identificación $e(k)$ acotado (estable en un sentido L_∞)*

$$\begin{aligned} W_1(k+1) &= W_1(k) - \eta(k) \sigma' x(k) e(k)^T \\ W_2(k+1) &= W_2(k) - \eta(k) u(k) \phi' x(k) e^T(k) \end{aligned} \quad (5.15)$$

donde $\eta(k)$ satisface

$$\eta(k) = \begin{cases} \frac{\eta}{1 + \|\sigma' x(k)\|^2 + \|u(k) \phi' x(k)\|^2} & \text{si } \beta \|e(k+1)\| \geq \|e(k)\| \\ 0 & \text{si } \beta \|e(k+1)\| < \|e(k)\| \end{cases}$$

$0 < \eta \leq 1$.

Demostración. Seleccionamos la función de Lyapunov como

$$V(k) = \|\widetilde{W}_1(k)\|^2 + \|\widetilde{W}_2(k)\|^2$$

donde $\|\widetilde{W}_1(k)\|^2 = \sum_{i=1}^n \tilde{w}_1(k)^2 = \text{tr} \left\{ \widetilde{W}_1^T(k) \widetilde{W}_1(k) \right\}$. De la ley actualizada (5.15)

$$\widetilde{W}_1(k+1) = \widetilde{W}_1(k) - \eta(k) \sigma' x(k) e(k)^T$$

Así que,

$$\begin{aligned}
\Delta V(k) &= V(k+1) - V(k) \\
&= \left\| \widetilde{W}_1(k) - \eta(k) \sigma' x(k) e(k)^T \right\|^2 - \left\| \widetilde{W}_1(k) \right\|^2 \\
&+ \left\| \widetilde{W}_2(k) - \eta(k) u(k) \phi' x(k) e^T(k) \right\|^2 - \left\| \widetilde{W}_2(k) \right\|^2 \\
&= \eta^2(k) \|e(k)\|^2 \|\sigma' x(k)\|^2 - 2\eta(k) \left\| \sigma' \widetilde{W}_1(k) x(k) e^T(k) \right\| \\
&+ \eta^2(k) \|e(k)\|^2 \left\| u(k) \phi' x(k) \right\|^2 - 2\eta(k) \left\| u(k) \phi' \widetilde{W}_2(k) x(k) e^T(k) \right\|
\end{aligned}$$

Existe una constante $\beta > 0$, tal que

- Si $\|\beta e(k+1)\| \geq \|e(k)\|$, usando (5.14) y $\eta(k) \geq 0$,
$$\begin{aligned}
&-2\eta(k) \left\| \sigma' \widetilde{W}_1(k) x(k) e^T(k) \right\| - 2\eta(k) \left\| u(k) \phi' \widetilde{W}_2(k) x(k) e^T(k) \right\| \\
&\leq -2\eta(k) \|e^T(k)\| \|\beta e(k+1) - Ae(k) - \zeta(k)\| \\
&= -2\eta(k) \|e^T(k)\| \|\beta e(k+1) - e^T(k) Ae(k) - e^T(k) \zeta(k)\| \\
&\leq -2\eta(k) \|e^T(k)\| \|\beta e(k+1)\| + 2\eta(k) \|e^T(k)\| \|Ae(k) + \zeta(k)\| \\
&\leq -2\eta(k) \|e(k)\|^2 + 2\eta(k) \lambda_{\max}(A) \|e(k)\|^2 + \eta(k) \|e(k)\|^2 + \eta(k) \|\zeta(k)\|^2
\end{aligned}$$

De modo que $0 < \eta \leq 1$,

$$\begin{aligned}
\Delta V(k) &\leq \eta^2(k) \|e(k)\|^2 \|\sigma' x(k)\|^2 + \eta^2(k) \|e(k)\|^2 \left\| u(k) \phi' x(k) \right\|^2 \\
&- \eta(k) \|e(k)\|^2 + 2\eta(k) \lambda_{\max}(A) \|e(k)\|^2 + \eta(k) \|\zeta(k)\|^2 \\
&= -\eta(k) \left[\left(1 - 2\lambda_{\max}(A) \right) - \eta \frac{\|\sigma' x(k)\|^2 + \left\| u(k) \phi' x(k) \right\|^2}{1 + \|\sigma' x(k)\|^2 + \left\| u(k) \phi' x(k) \right\|^2} \right] e^2(k) + \eta \zeta^2(k) \quad (5.16) \\
&\leq -\pi e^2(k) + \eta \zeta^2(k)
\end{aligned}$$

donde $\pi = \frac{\eta}{1 + \kappa} \left[1 - 2\lambda_{\max}(A) - \frac{\kappa}{1 + \kappa} \right]$, $\kappa = \max_k \left(\|\sigma' x(k)\|^2 + \left\| u(k) \phi' x(k) \right\|^2 \right)$. De tal modo que $-1 < \lambda(A) < 0$, $\pi > 0$

$$n \min(\tilde{w}_i^2) \leq V_k \leq n \max(\tilde{w}_i^2)$$

donde $n \times \min(\tilde{w}_i^2)$ y $n \times \max(\tilde{w}_i^2)$ son \mathcal{K}_∞ -funciones, y $\pi e^2(k)$ es una \mathcal{K}_∞ -función, $\eta \zeta^2(k)$ es una \mathcal{K} -función, así que V_k admite una función suave o diferenciable de Lyapunov-ISS como en la

Definición 2. Del Teorema 1, la dinámica del error de identificación es *estable de entrada-estado*. La "ENTRADA" corresponde al segundo término de la línea anterior en (5.16), *i.e.*, el modelado del error $\zeta(k) = \varepsilon_1(k) + \varepsilon_2(k) - \mu(k)$, el "ESTADO" corresponde al primer término de la línea anterior en (ref:dLk1), *i.e.*, el error de identificación $e(k)$. Porque la "ENTRADA" $\zeta(k)$ es acotada y la dinámica es ISS, el "ESTADO" $e(k)$ esta acotado.

· Si $\beta \|e(k+1)\| < \|e(k)\|$, $\Delta V(k) = 0$. $V(k)$ es constante, $W_1(k)$ es constante. De modo que $\|e(k+1)\| < \frac{1}{\beta} \|e(k)\|$, $\frac{1}{\beta} < 1$, $e(k)$ esta acotado. ■

Propiedad 15 (5.5) es el algoritmo del gradiente descendente, cuya tasa de aprendizaje normalizado η_k es variante en el tiempo para asegurar que el proceso de identificación sea estable. Esta ley de aprendizaje es simple para usar, porque no necesitamos tener cuidado acerca de como seleccionar una mejor tasa de aprendizaje para asegurar rápida convergencia y estabilidad. No cualquier información previa es requerida.

Propiedad 16 La condición $\beta \|e(k+1)\| \geq \|e(k)\|$ es zona-muerta. Si β es seleccionada suficientemente grande, la zona-muerta se tornará pequeña.

Propiedad 17 La clases de redes consideradas en este artículo en no lineal en los pesos como en cite:Polycarpou y cite:Song. Debido a que la convergencia de aprendizaje es lenta y sus conjuntos de datos altos, muchas implementaciones prácticas de redes neuronales son lineales en los pesos cite:Gecite:Suikens.

$$\beta \hat{x}(k+1) = A\hat{x}(k) + W_1(k) \sigma[x(k)]$$

En este caso el error de identificación dinámico (5.4) se tornará

$$\beta e(k+1) = Ae(k) + W_k \sigma[x(k)] - W^* \sigma[x(k)] + \mu(k) = Ae(k) + \widetilde{W}_k \sigma[x(k)] + \mu(k)$$

Si usamos la siguiente ley actualizada

$$W_{k+1} = W_k - \eta_k e(k) \sigma^T$$

donde η_k *satisface*

$$\eta_k = \begin{cases} \frac{\eta}{1 + \|\sigma\|^2}, & 0 < \eta \leq 1 \quad \text{si } \beta \|e(k+1)\| \geq \|e(k)\| \\ 0 & \text{si } \beta \|e(k+1)\| < \|e(k)\| \end{cases}$$

podemos probar que el error de identificación $e(k)$ *esta también acotado. Los procedimientos de prueba son los mismos que el Teorema 2.*

5.2. Identificación de sistemas con Redes Neuronales Recurrentes Multicapas en Tiempo-Discreto

Para simplificar los procesos de cálculo, discutimos el siguiente sistema no lineal en tiempo discreto sin entrada de control

$$x(k+1) = f[x(k)] \quad (5.20)$$

La red neuronal recurrente multicapa en tiempo discreto esta representada como

$$\beta \hat{x}(k+1) = A \hat{x}(k) + V_k \sigma [W_k x(k)] \quad (5.21)$$

donde los pesos en la capa de salida son $V_k \in R^{1 \times m}$, los pesos en la capa oculta son $W_k \in R^{m \times n}$, σ es una función vectorial de m - *dimensión* . La presentación típica del elemento $\phi_i(\cdot)$ es una función sigmoideal. La estructura de la red neuronal recurrente multicapa en tiempo discreto se muestra en la figura 5.2. El MLP es un perceptrón multicapa en tiempo discreto.

Si la planta no lineal tiene entradas de control como

$$x(k+1) = f[x(k), u(k)] \quad (5.22)$$

La red neuronal recurrente en tiempo discreto esta en la forma de

$$\beta \hat{x}(k+1) = A \hat{x}(k) + V_{1,k} \sigma [W_{1,k} x(k)] + V_{2,k} \phi [W_{2,k} x(k)] U(k) \quad (5.23)$$

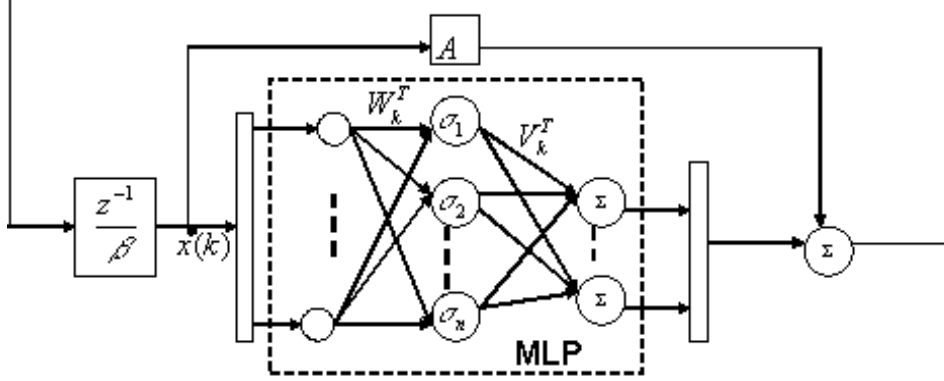


Figura 5.2: Red Neuronal Recurrente Multicapa en tiempo discreto.

La neuro identificación discutida en esta tesis es un tipo de identificación on-line, *i.e.*, usaremos el error de identificación $e(k)$ para entrenar la red neuronal (5.23) on-line tal que $\hat{y}(k)$ puede aproximarse a $y(k)$. El sistema no lineal identificado (5.20) puede ser representado y reescrito como

$$\beta x(k+1) = Ax(k) + V^* \sigma [W^* x(k)] - \mu(k) \quad (5.24)$$

donde V^* y W^* son un conjunto de pesos desconocidos que pueden minimizar el error modelado $\mu(k)$. En este caso las tres variables independientes, una función suave f tendrá la fórmula de Taylor como

$$f(x_1, x_2) = \sum_{k=0}^{l-1} \frac{1}{k!} \left[(x_1 - x_1^0) \frac{\partial}{\partial x_1} + (x_2 - x_2^0) \frac{\partial}{\partial x_2} \right]^k f + R_l$$

donde R_l es el residuo de la fórmula de Taylor. Si dejamos x_1 y x_2 correspondiendo a V^* y W^* , x_1^0 y x_2^0 corresponden a V_k y W_k . Usando series de Taylor alrededor del punto de $W_k X(k)$ y V_k , el error de identificación $e(k) = \hat{x}(k) - x(k)$ puede ser representado como

$$\beta e(k+1) = Ae(k) + \tilde{V}_k \sigma [W_k x(k)] + V_k \sigma' \tilde{W}_k x(k) + \zeta(k) \quad (5.25)$$

donde σ' es la derivada de la función de activación no lineal $\sigma(\cdot)$ en el punto de $W_k X(k)$, $\widetilde{W}_k = W_k - W^*$, $\widetilde{V}_k = V_k - V^*$, $\zeta(k) = R_1 + \mu(k)$, aquí R_1 es un error de aproximación de segundo orden de la serie de Taylor.

En esta tesis estamos solo interesados en identificación en lazo-abierto, entonces podemos asumir que la planta (4.1) es estable acotada, *i.e.*, $x(k)$ en (5.20) esta acotada. Por los límites de la función sigmoidal σ podemos asumir que $\mu(k)$ en (5.24) esta acotada, también R_1 esta acotada. Así que $\zeta(k)$ en (5.25) es acotada. El siguiente teorema da un algoritmo de backpropagation estable para redes neuronales multicapa en tiempo discreto.

Teorema 18 *Si usamos la red neuronal multicapa (5.23) para identificar plantas no lineales (5.20) y A es seleccionada como $-1 < \lambda(A) < 0$, el siguiente gradiente de ley actualizada sin modificación robusta puede hacer el error de identificación $e(k)$ acotado*

$$\begin{aligned} W_{k+1} &= W_k - \eta_k e(k) \sigma' V_k^T x(k) \\ V_{k+1} &= V_k - \eta_k e(k) \sigma^T [W_k(k) x(k)] \end{aligned} \quad (5.26)$$

donde η_k satisface

$$\eta_k = \begin{cases} \frac{\eta}{1 + \|\sigma' V_k^T x(k)\|^2 + \|\sigma\|^2} & \text{si } \beta \|e(k+1)\| \geq \|e(k)\| \\ 0 & \text{si } \beta \|e(k+1)\| < \|e(k)\| \end{cases} \quad (5.27)$$

$0 < \eta \leq 1$. El promedio del error de identificación satisface

$$J = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=1}^T e^2(k) \leq \frac{\eta \bar{\zeta}}{\pi} \quad (5.28)$$

donde $\pi = \frac{\eta}{1 + \kappa} \left[1 - \frac{\kappa}{1 + \kappa} \right] > 0$, $\kappa = \max_k \left(\|\sigma' V_k^T x(k)\|^2 + \|\sigma\|^2 \right)$, $\bar{\zeta} = \max_k [\zeta^2(k)]$

Demstración. Seleccionando una matriz definida positiva L_k como

$$L_k = \left\| \widetilde{W}_k \right\|^2 + \left\| \widetilde{V}_k \right\|^2$$

De la ley actualizada (5.26), tenemos

$$\widetilde{W}_{k+1} = \widetilde{W}_k - \eta_k e(k) \sigma' V_k^T x^T(k), \quad \widetilde{V}_{k+1} = \widetilde{V}_k - \eta_k e(k) \sigma^T$$

De tal modo ϕ' es una matriz diagonal, y usando (5.25) tenemos

$$\begin{aligned} \Delta L_k &= \left\| \widetilde{W}_k - \eta_k e(k) \sigma' V_k^T x^T(k) \right\|^2 + \left\| \widetilde{V}_k - \eta_k e(k) \sigma^T \right\|^2 - \left\| \widetilde{W}_k \right\|^2 - \left\| \widetilde{V}_k \right\|^2 \\ &= \eta_k^2 e^2(k) \left(\left\| \sigma' V_k^T x^T(k) \right\|^2 + \left\| \sigma \right\|^2 \right) - 2\eta_k \|e(k)\| \left\| V_k^T \sigma' \widetilde{W}_k X(k) + \widetilde{V}_k \sigma \right\| \\ &= \eta_k^2 e^2(k) \left(\left\| \sigma' V_k^T x^T(k) \right\|^2 + \left\| \sigma \right\|^2 \right) - 2\eta_k \|e(k)\| [\beta e(k+1) - Ae(k) - \zeta(k)] \end{aligned} \quad (5.29)$$

Usando (5.25) y $\eta(k) \geq 0$, existe una constante $\beta > 0$, tal que

· Si $\|\beta e(k+1)\| \geq \|e(k)\|$

$$\begin{aligned} \Delta L_k &\leq -\eta_k e^2(k) \left[1 - \eta_k \left(\left\| \sigma' V_k^T x^T(k) \right\|^2 + \left\| \sigma \right\|^2 \right) \right] + \eta \zeta^2(k) \\ &\leq -\pi e^2(k) + \eta \zeta^2(k) \end{aligned} \quad (5.30)$$

donde π esta definida en (5.28). Porque

$$n \left[\min(\tilde{w}_i^2) + \min(\tilde{v}_i^2) \right] \leq L_k \leq n \left[\max(\tilde{w}_i^2) + \max(\tilde{v}_i^2) \right]$$

donde $n \left[\min(\tilde{w}_i^2) + \min(\tilde{v}_i^2) \right]$ y $n \left[\max(\tilde{w}_i^2) + \max(\tilde{v}_i^2) \right]$ son \mathcal{K}_∞ -funciones, y $\pi e^2(k)$ es una \mathcal{K}_∞ -función, $\eta \zeta^2(k)$ es una \mathcal{K} -función. De (5.25) y (ref:Lk1) conocemos V_k es la función de $e(k)$ y $\zeta(k)$, de tal modo que L_k admite una función suave o difernciable de Lyapunov-ISS como en la *Definición 2*. Del *Teorema 1*, la dinámica del error de identificación identification es estable de entrada-estado . Porque la "ENTRADA" $\zeta(k)$ esta acotada y la dinámica es ISS, el "ESTADO" $e(k)$ esta acotado.

· Si $\beta \|e(k+1)\| < \|e(k)\|$, $\Delta V(k) = 0$. $L(k)$ es constante, las constantes de W_k y V_k por medio de $e(k)$ esta acotado.

(5.28) puede ser reescrita como

$$\Delta L_k \leq -\pi e^2(k) + \eta \zeta^2(k) \leq \pi e^2(k) + \eta \bar{\zeta} \quad (5.31)$$

Resumiendo (5.30) de 1 hasta T , y usando $L_T > 0$ y L_1 es una constante, obtenemos

$$\begin{aligned} L_T - L_1 &\leq -\pi \sum_{K=1}^T e^2(k) + T\eta \bar{\zeta} \\ \pi \sum_{K=1}^T e^2(k) &\leq L_1 - L_T + T\eta \bar{\zeta} \leq L_1 + T\eta \bar{\zeta} \end{aligned}$$

(5.29) esta establecido. ■

Propiedad 18 *La tasa de aprendizaje normalizado η_k en (5.26) y (5.27) son variantes en el tiempo para asegurar que los procesos de identificación sea estables. Estas ganancias de aprendizaje son fácil de ser decididas, no cualquier información previa es requerida , por ejemplo podemos seleccionar $\eta = 1$. La contradicción en rápida convergencia y aprendizaje estable puede ser evitable. Si seleccionamos η como función de zona-muerta:*

$$\begin{cases} \eta = 0 & \text{si } |e(k)| \leq \bar{\zeta} \\ \eta = \eta_0 & \text{si } |e(k)| > \bar{\zeta} \end{cases}$$

(5.26) es la misma que en cite:Song y cite:YuJJC. Si un término de modificación— σ o término modificado de regla— δ son adicionados en (5.26), se convierte que de cite:Jagannathan o de que cite:Lewis. Pero todos ellos necesitan el límite superior del error modelado $\bar{\zeta}$. Y el error de identificación es aumentado por las modificaciones robustas cite:Ioannou.

Propiedad 19 *Puesto que asumimos que las redes neuronales no pueden competir exactamente con sistemas no lineales ,no podemos hacer que los parámetros (pesos) converjan ,podemos solo forzar la salida de las redes neuronales a seguir la salida de la planta, i.e.,la identificación del error es estable . Aunque los pesos no pueden converger a sus valores óptimos, (5.27) se muestra que el error de identificación convergerá a la esfera o bola de radio $\frac{\eta}{\pi}\bar{\zeta}$. Aún cuando la entrada es excitada persistentemente, el error modelado $\zeta(k)$ no hará que los pesos converjan a sus valores óptimos . Es posible que el error de salida sea convergente, pero los errores de los pesos son muy altos cuando la estructura de la red no está bien definida. Las relaciones del error de salida y los errores de los pesos son mostrados en (5.25). Un caso simple es que usamos on-line en los pesos y las redes neuronales pueden competir exáctamente con la planta no lineal*

$$\begin{aligned} \text{planta: } y &= W^* \sigma [x(k)] \\ \text{redes neuronales: } \hat{y} &= W_k \sigma [x(k)] \\ \text{error de salida: } (y - \hat{y}) &= (W^* - W_k) \sigma [x(k)] \end{aligned}$$

Si $\sigma [x(k)]$ es grande, el error de salida pequeño ($y - \hat{y}$) no hará una buena convergencia del error de los pesos ($W^ - W_k$).*

Propiedad 20 *El ruido (o perturbación) es una cuestión importante en la identificación de sistemas. Existen dos tipos de perturbaciones: externa e interna. La perturbación interna puede ser estimada como una dinámica no modelada en $\mu(k)$. Una perturbación interna acotada no tiene efectos en los resultados teóricos en esta tesis, pero puede aumentar el error de identificación si la perturbación interna se torna grande. Las perturbaciones externas pueden ser estimadas como una medida de ruido, ruido a la entrada, etc. En el punto de la estructura, los ruidos a la entrada son incrementados hacia adelante de un lado a otro de cada capa cite:Brown. Por ejemplo, un ruido $\mu(k)$ es multiplicado por $V_k\sigma [W_k\zeta(k)]$ y alcanza a la salida. La medida del ruido es aumentada debido a la retroalimentación del error de identificación (5.26), por lo tanto los pesos de las redes neuronales son influenciados por el ruido a la salida. Por otra parte la perturbación externa pequeña puede acelerar la velocidad de convergencia de acuerdo a la teoría de excitación persistente cite:Polycarpou, pequeñas perturbaciones en la entrada $u(t)$ o en la salida $y(t)$ puede enriquecer la frecuencia de la señal $X(t)$, esto es bueno para la convergencia de parámetros. En la siguiente simulación podemos ver este punto.*

5.3. Resultado de Simulaciones

En esta sección un típico sistema caótico, el modelo de Lorenz, es elegido para demostrar la capacidad del neuro-identificador. El modelo de Lorenz es usado para la descripción de un fluido por convección especialmente para algunas características de la dinámica atmosférica :YuCS0. El modelo no controlado es dado por las siguientes ecuaciones

$$\begin{aligned}\dot{x}_1 &= -\beta x_1 + x_2 x_3 \\ \dot{x}_2 &= \omega(x_3 - x_2) \\ \dot{x}_3 &= -x_1 x_2 + \rho x_2 - x_3\end{aligned}\tag{5.32}$$

donde x_1 , x_2 y x_3 representan medidas de temperaturas verticales, horizontales y velocidad del fluido, respectivamente. ω , ρ y β son parámetros positivos que representan el número de Prandtl, el número de Rayleigh y el factor geométrico. Si $\rho < 1$, el origen es un equilibrio

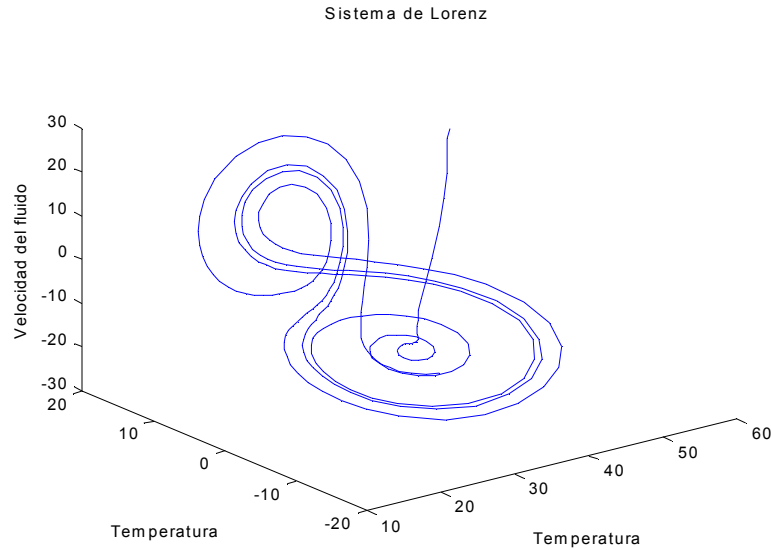


Figura 5.3: Sistema dinámico caótico de Lorenz.

estable. Si

$$1 < \rho \leq \rho^*(\sigma, \beta) := \omega(\omega + \beta + 3) / (\omega - \beta - 1),$$

el sistema tendrá dos puntos de equilibrio estable $(\pm\sqrt{\beta(\rho-1)}, \pm\sqrt{\beta(\rho-1)}, (\rho-1))$ y un equilibrio inestable (el origen). Si $\rho^*(\omega, \beta) < \rho$, los tres puntos de equilibrio se tornarán inestables. Como en el caso comúnmente estudiado, seleccionamos $\omega = 10$ y $\beta = 8/3$ que conduce a $\rho^*(\omega, \beta) = 24,74$. $x_0 = [50, 10, 20]^T$.

Sistema de Lorenz En esta aplicación consideraremos el sistema con $\omega = 10$, $\beta = 8/3$ y $\rho = 28$. La dinámica de este sistema caótico es mostrado en la fig.5.3.

Usaremos la siguiente técnica en diferencias para obtener los estados en tiempo discreto del sistema (5.32). La ecuación de Lorenz (5.32) puede ser escrita como

$$\dot{x} = Ax$$

$$\text{con } x = [x_1, x_2, x_3]^T, A = \begin{bmatrix} -\beta & 0 & x_2 \\ 0 & -\sigma & \sigma \\ -x_2 & \rho & -1 \end{bmatrix}. \text{ Definamos } s_1 = Ax_k, s_2 = A(x_k + s_1),$$

$$s_3 = A\left(x_k + \frac{s_1 + s_2}{4}\right). \text{ Si } \left| \frac{s_1 - 2 * s_3 + s_2}{3} \right| \leq \frac{|x_k|}{1000} \text{ o } \left| \frac{s_1 - 2 * s_3 + s_2}{3} \right| < 1, \text{ entonces}$$

$$x_{k+1} = x_k + \frac{s_1 + 4s_3 + s_2}{6}, k = 0, 1, 2 \dots,$$

5.3.1. Simulaciones y Gráficas

(A) Utilizaremos redes neuronales monocapa para realizar el proceso de identificación variando la matriz A , para que en base a los resultados obtenidos, concluir cuales son los valores más adecuados en el mejor comportamiento de la red monocapa con respecto a la señal del sistema. Primero usamos redes neuronales monocapa (5.10) para identificar a (5.32).

$$\beta = 4$$

$$\eta = < 1$$

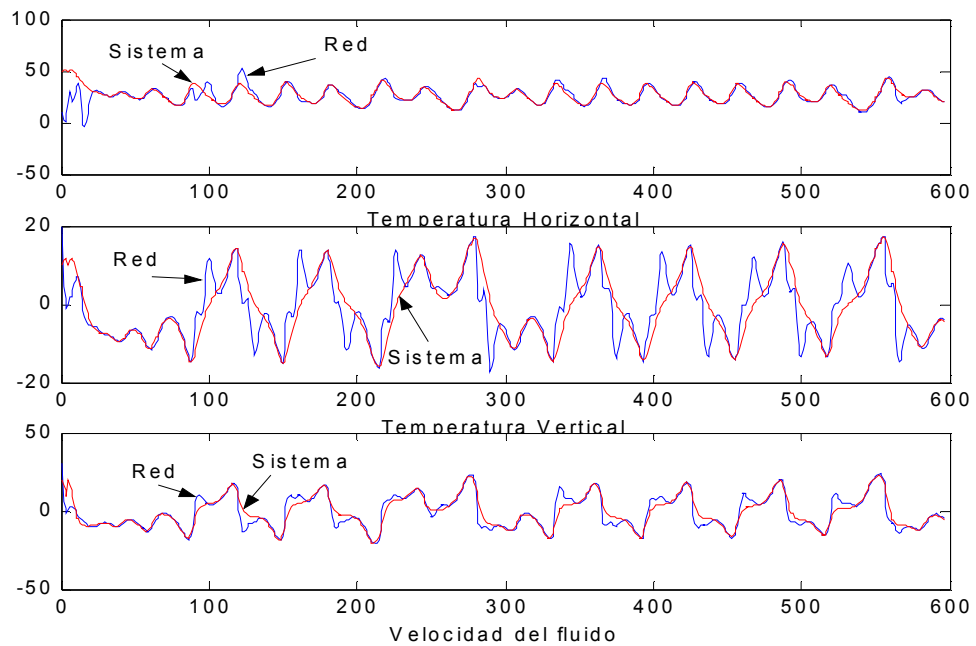
$$W_1(0) = \begin{bmatrix} \text{rand}(0, 1) \\ 1 \\ 1 \end{bmatrix}$$

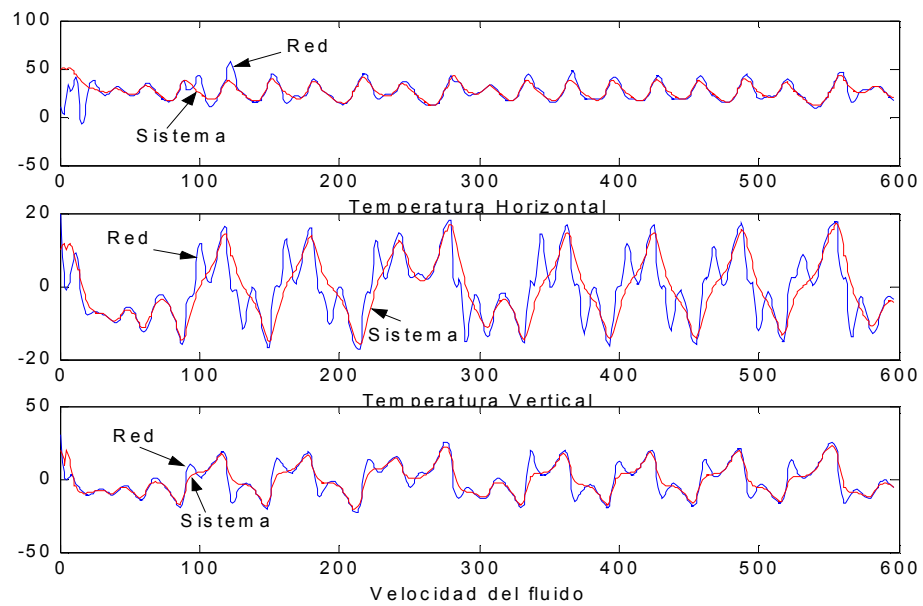
$$\beta \hat{x}(k+1) = A \hat{x}(k) + W_1(k) \sigma[x(k)]$$

donde $\beta = 4$, $\sigma(\cdot) = \tanh(\cdot)$, $\hat{x} = [\hat{x}_1, \hat{x}_2, \hat{x}_3]^T$, $A = \text{diag}[8, 8, 8]$, $W_1(k) \in \mathfrak{R}^{3 \times 3}$. Los elementos de $W_1(0)$ son números aleatorios entre $[0, 1]$. Usamos (5.26) para actualizar el peso

$$W_1(k+1) = W_1(k) - \frac{\eta}{1 + \|\sigma\|^2} e(k) \sigma^T$$

$\eta = 1$. El Teorema 2 da una condición necesaria de η para aprendizaje estable, $\eta \leq 1$. En este ejemplo, encontramos que si $\eta \geq 2,5$, el proceso de aprendizaje se tornará inestable. La neuro-identificación discutida en esta tesis es *on-line*, no estudiaremos la convergencia del peso, cuidaremos acerca del error de identificación $e(k)$. Los pesos no convergen a las mismas constantes o valores óptimos. El tiempo de simulación total es de 600, existen 14 tiempos $\beta \|e(k+1)\| < \|e(k)\|$. Las diferentes identificaciones *on-line* resultantes son mostradas en las siguientes figuras:

Figura 5.4: Resultados para $A=\text{diag}[0.8, 0.8, 0.8]$

Figura 5.5: Resultados para $A=\text{diag}[1.5, 1.5, 1.5]$

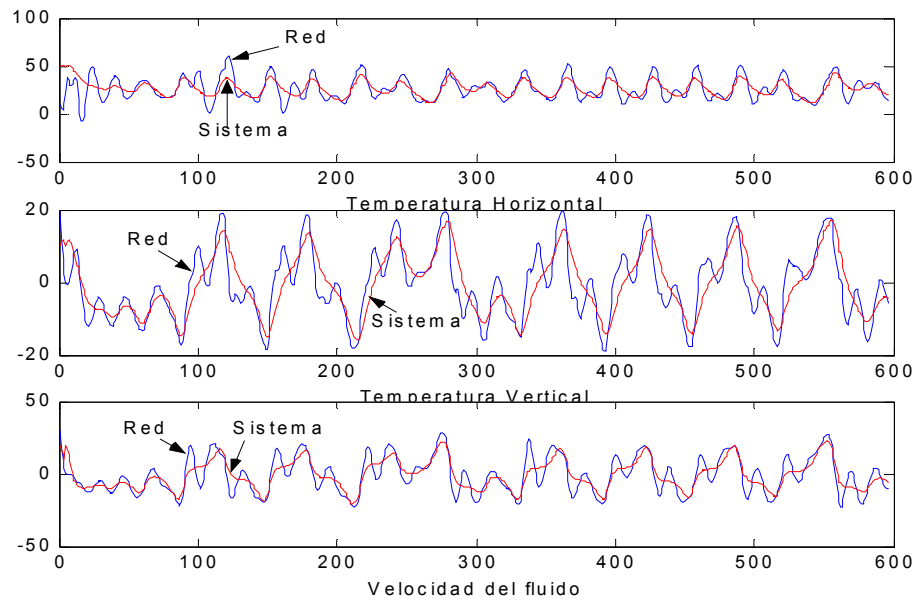


Figura 5.6: Resultados para $A=\text{diag}[2.2, 2.2, 2.2]$

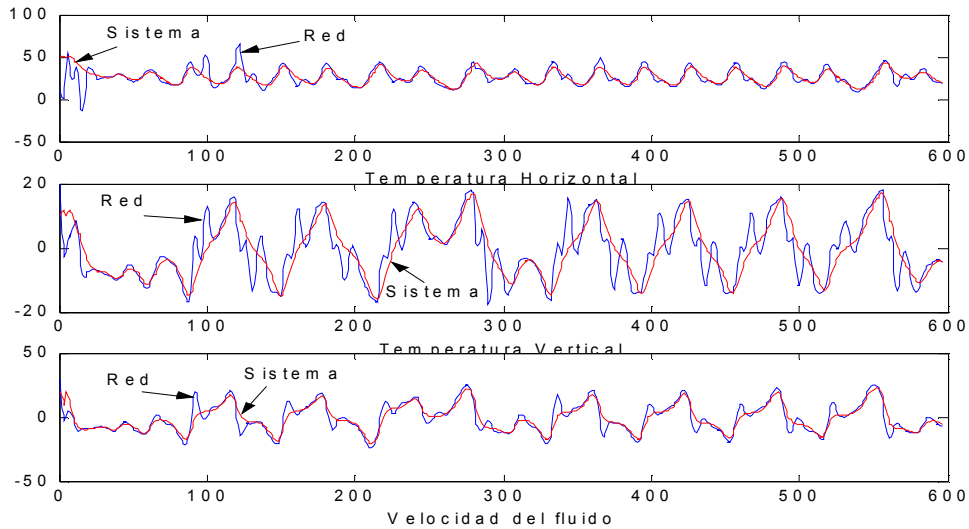
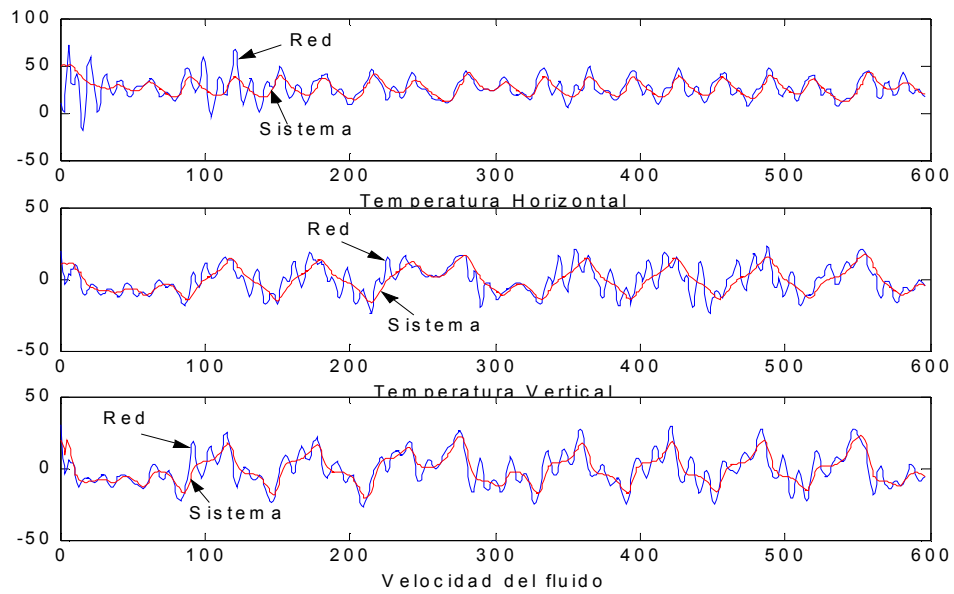


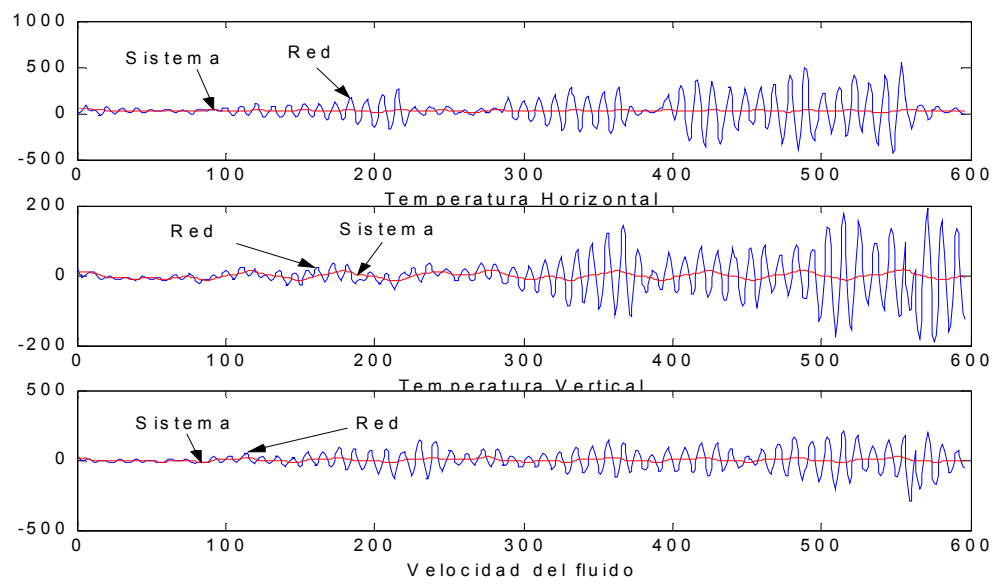
Figura 5.7: Resultados obtenidos para $\mu_0=1.8$

En base a los resultados obtenidos en las anteriores simulaciones, es posible sacar en conclusión, que las más adecuadas y en donde se presenta mayor estabilidad y convergencia de la red neuronal moncoapa, son las que corresponden, a la gráficas de la figura 5.4. En donde es posible ver, mayor restabilidad y buen seguimiento de la red, con respecto al sistema. Mientras que de las gráficas de la figura 5.6, es fácil observar la inestabilidad que presenta la señal de la red neuronal al aumentar los valores de la matriz diagonal A .

A continuación se presentan las mismas simulaciones, pero ahora variando los valores de μ_0 :

De los resultados obtenidos en estas simulaciones, es posible concluir, que al aumentar el valor del parámetro μ_0 la red neuronal va presentando mayor inestabilidad, mientras que al mantener valores acotados, la estabilidad permanecerá bastante aceptable, como se puede ver en las gráficas de la figura 5.7, y al salir de un determinado rango, la inestabilidad se hará presente, tal y como sucede en las gráficas de la figura 5.9, donde se ha alcanzado un estado totalmente inestable.

Figura 5.8: Resultados obtenidos para $\mu_0=2.4$

Figura 5.9: Resultados para $\mu_0=2.8$

(B) En seguida se presentarán, las mismas gráficas, pero ahora con la variación de los valores del parámetro M y utilizando redes neuronales multicapa para el proceso de identificación y para poder observar cuáles son las ventajas, que estas pueden presentar sobre las redes monocapa..

$$\beta = 4$$

$$M = 10, 20, \text{ capas ocultas}$$

$$\eta = < 1$$

$$W_1(0) = \begin{bmatrix} \text{rand}(0, 1) \\ 1 \\ 1 \end{bmatrix}$$

$$V_1(0) = [\text{rand}(0, 1), 1, 1]^T$$

Entonces usaremos redes neuronales multicapa (5.23) para identificar a (5.32).

$$\beta \hat{x}(k+1) = A \hat{x}(k) + V_1 \sigma [W_1(k) x(k)]$$

donde $\beta = 4$, $\sigma(\cdot) = \tanh(\cdot)$, $A = \text{diag}[8, 8, 8]$. La complejidad del modelo es importante en el contexto de identificación de sistemas, que corresponde al número de unidades ocultas del modelo neuronal. En esta simulación trataremos de probar diferentes números de nodos ocultos, encontramos que después de que los nodos ocultos son más de 20, la precisión de identificación no será mejorada mucho. En cite:Narendra, ellos también usaron 20 nodos ocultos para la primera capa oculta . Así que seleccionamos $W_k \in R^{20 \times 3}$, $V_k \in R^{3 \times 20}$. El algoritmo de aprendizaje para W_k y V_k son (5.26) con $\eta_k = \frac{1}{1 + \|\sigma' V_k^T x(k)\|^2 + \|\sigma\|^2}$, $\sigma'(\cdot) = \text{sech}^2(\cdot)$. Las condiciones iniciales para W_k y V_k son números aleatorios, *i.e.*, $W_0 = \text{rand}(\cdot)$, $V_0 = \text{rand}(\cdot)$. La identificación on-line resultante es mostrada en las siguientes figuras. El tiempo de simulación total es de 600, existe 1 tiempo $\beta \|e(k+1)\| < \|e(k)\|$. Comparada con backpropagation recurrente cite:Jin, la tasa de aprendizaje variante en el tiempo η_k en (5.26) es fácil de ser realizado.

Resultados de las simulaciones para redes neuronales multicapa al variar M :

De los anteriores resultados de las simulaciones obtenidos, es posible observar, que al aumentar el número de capas ocultas de la red neuronal multicapa, esta presenta mejores comportamientos y mayor capacidad de estabilidad, con respecto las redes de menor número

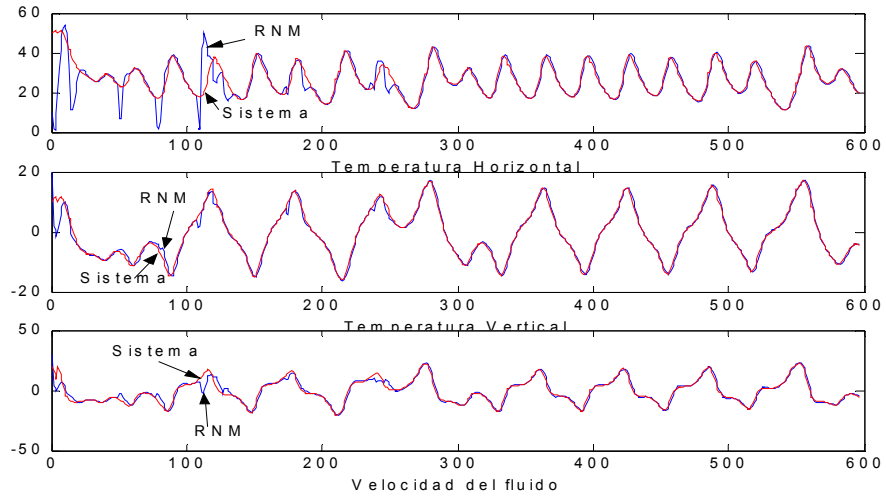


Figura 5.10: Resultados para $M=6$.

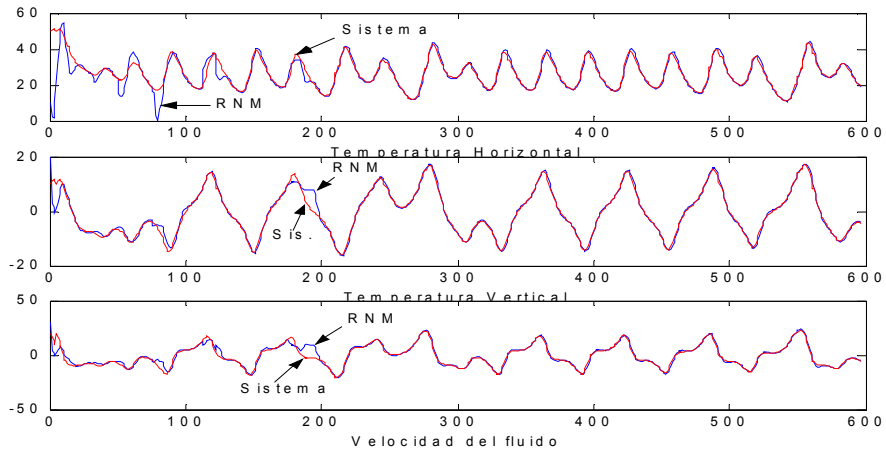


Figura 5.11: Resultados para $M=10$.

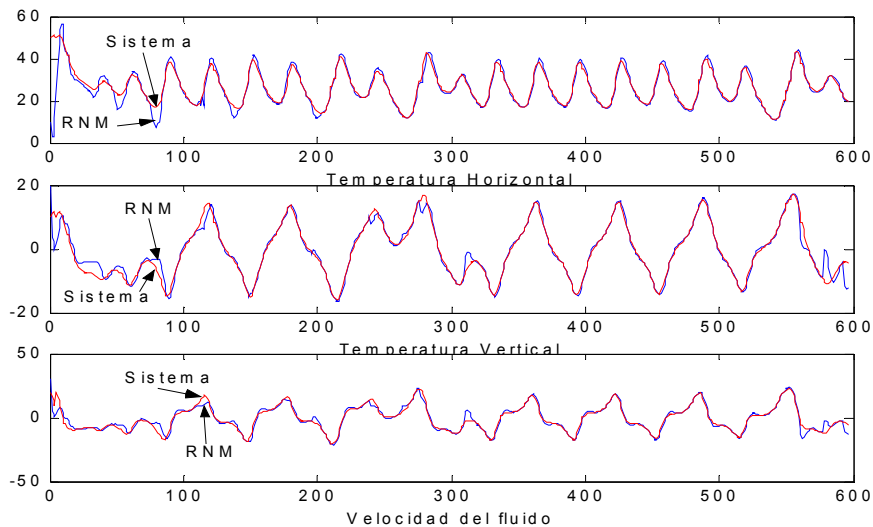


Figura 5.12: Resultados para $M=20$.

de capas, en donde por ejemplo, las gráficas de las figura 5.12, presentan un mejor comportamiento con respecto a las dos anteriores.

A continuación se mostrarán las mismas simulaciones, pero para diferentes valores de β :

Por lo que respecta a los resultados obtenidos en estas simulaciones, es posible concluir, que al incrementar el valor de β , se presenta mayor inestabilidad por parte de las redes multicapa, con respecto a la señal del sistema. Pero hay que tomar en cuenta, que los valores de β , están integrados dentro la matriz A . Entonces el mejor comportamiento de la red mutlicapa, es el presentado en las gráficas de la figura 5.14, donde se puede observar un buen seguimiento de las trayectorias de la red con respecto a la señal del sistema.

5.4. Comentarios y conclusiones:

Para la elaboración de las simulaciones del sistema caótico de Lorenz, se utilizó solo lenguaje matlab sin necesidad de recurrir a los bloques de simulink, porque no fue necesario ya

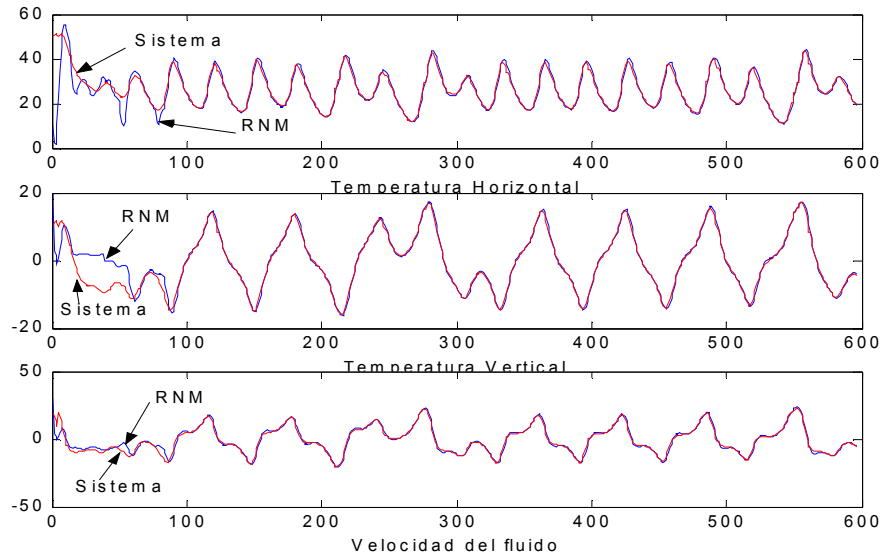


Figura 5.13: Resultados para $\beta=4$.

que solo se necesitaron hacer los programas adecuadamente y así poder hacer las simulaciones pertinentes. Aquí también se utilizaron redes neuronales monocapa y multicapa, para el sistema, que se consideró en tiempo discreto debido a la naturaleza de sus ecuaciones por ser factibles y viables de ser discretizadas para posteriormente, utilizar ecuaciones en diferencias y poder relacionarlas adecuadamente, con redes neuronales discretas.

Generalmente se dice, que una posible técnica de discretizar sistemas no lineales complejos en tiempo continuo, es el de adicionar retenedores de orden cero aunque sea de un punto de vista teórico, sin embargo esto no fue necesario en nuestro sistema, por el hecho de que las ecuaciones que gobiernan o describen el sistema, son factibles de ser discretizadas vía transformada Z.

* Se utilizó el algoritmo backpropagation o del gradiente descendente, para el entrenamiento de las redes, por ser este uno de los paradigmas de aprendizaje más versátiles y efectivos que se utilizan en redes neuronales.

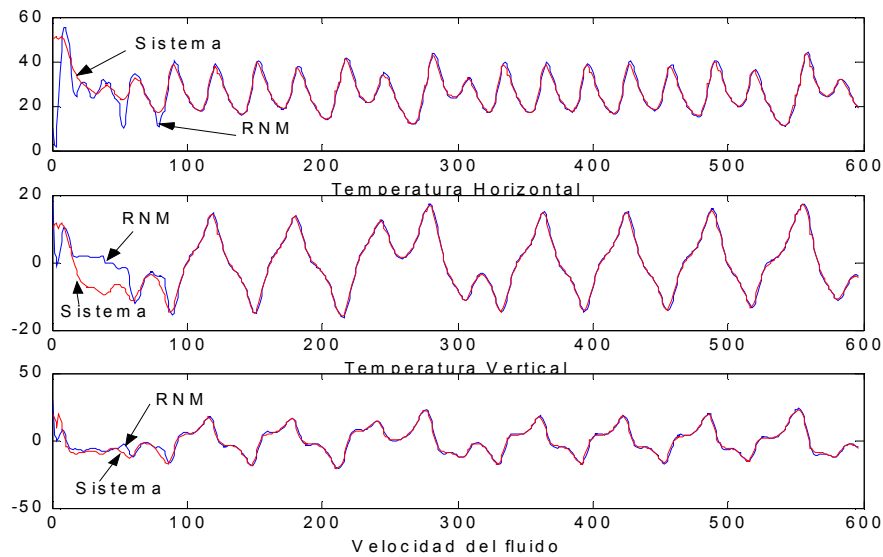
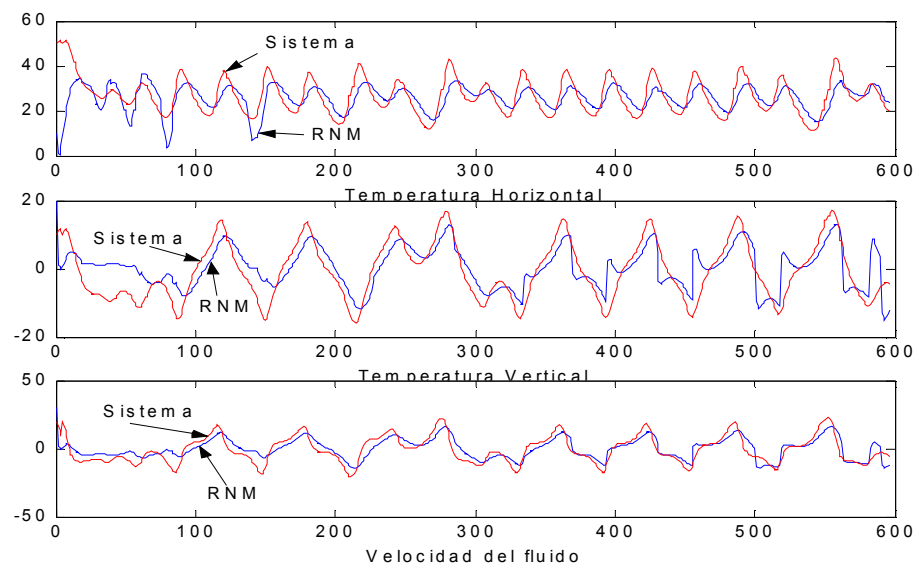


Figura 5.14: Resultados para $\beta=8$.

* El aprendizaje de las redes fue posible realizarlo on-line, es decir, el entrenamiento de las redes se realizó al mismo tiempo que se efectuaba el funcionamiento de las mismas.

* Una vez más es posible observar, que los resultados obtenidos con las redes neuronales multicapa, entregan mejores resultados con respecto a las monocapa y que prácticamente solo se requiere de jugar con los datos de los parámetros pertinentes, para visualizar que las señales entregadas por las redes neuronales, se comportan como deseamos, pudiéndose identificar plenamente el sistema en cuestión y que el enfoque de *estabilidad de entrada-estado*, es una buena técnica para obtener la estabilidad de un sistema.

* En base a los resultados obtenidos es posible concluir, que las técnicas de las redes neuronales artificiales, en el proceso de identificación de sistemas no lineales complejos con el enfoque de *estabilidad de entrada-estado*, nos indicará que para que un sistema sea estable, bastará que las señales de entrada estén acotadas para que las señales a la salida permanezcan acotadas y así se mantenga el estado de estabilidad que tanto se pretende.

Figura 5.15: Resultados para $\beta=10$.

Capítulo 6

Conclusiones Generales

1. Por medio de técnica de pasividad, hemos mostrado algunos nuevos resultados sobre neuro-identificación con redes neuronales dinámicas multicapa. Comparando con otros análisis de estabilidad de neuro-identificaciones, nuestro algoritmo es más simple porque las modificaciones robustas no son aplicadas, así que el algoritmo propuesto en esta tesis es más adecuado para aplicaciones ingenieriles. Exitosamente probamos que para cada evento el algoritmo de aprendizaje de retroalimentación o backpropagation puede garantizar el error de identificación estable robusto.
2. En esta tesis estudiamos y analizamos la identificación de sistemas no lineales por redes neuronales recurrentes multicapa y monocapa en tiempo continuo y tiempo discreto. Usando aproximación o enfoque de estabilidad de entrada-estado (ISS), concluimos que las técnicas de robustez comúnmente usadas para identificación de sistemas, tal como zona-muerta y modificación- σ , no son necesarias para el gradiente de ley descendente y el algoritmo de backpropagation (retroalimentación) cuando usamos redes neuronales recurrentes en tiempo-discreto.
3. Se puede concluir que las técnicas de redes neuronales artificiales multicapa en el proceso de identificación de sistemas no lineales complejos con el enfoque de *Estabilidad de Entrada-Estado*, son bastante efectivas para la realización de tales eventos, sin perder

afinidad con el concepto de estabilidad del punto de vista de Lyapunov.

4. El proceso de identificación es de vital importancia ya que por medio de este, es posible obtener información del sistema que se este analizando, para que en base a los datos obtenidos, sea posible el diseñar elementos o sistemas de control, lo cual puede resultar bastante difícil de realizar, si no se cuenta pleno conocimiento del sistema. Y es que por medio de identificación, se podrá a aspirar a realizar control de procesos de sistemas no lineales complejos.
5. Es también muy importante mencionar la versatilidad y eficacia que tienen las redes neuronales artificiales para aproximar diversas clase de funciones no lineales, lo que las convierte en excelentes alrtnativas para ser usadas en modelos dinámicos, para la posible representación de plantas o sistemas no lineales complejos, y es que un sistema de tales características puede tener varias entradas y varias salidas relacionadas entre si pero que pueden ser abordadas y analizadas por arquitecturas de redes neuronales artificiales recurrentes.
6. Trabajos más adelante estarán enfocados sobre *redes neuronales recurrentes* en tiempo continuo y tiempo-discreto para neuro-control basados en el enfoque o aproximación de *Estabilidad de Entrada-Estado (ISS)*.
7. Y debido a que las técnicas de redes neuronales artificiales han mostrado un buen desempeño en el proceso de identificación es posible su aplicación en control adaptable por medio de esquemas de control directo adaptable y control indirecto adaptable, por lo que utilizando el enfoque (ISS), tanto para sistemas en tiempo continuo como en tiempo discreto, se espera poder continuar con estas tareas de investigación y desarrollo, para extender la agradable capacidad de las redes neuronales artificiales hacía mecanismos de neurocontrol, tanto del punto de vista teórico como práctico.

Capítulo 7

References

Bibliografía

- [1] Byrnes C.I., Isidori A., Willems J.C., Passivity, Feedback Equivalence, and the Global Stabilization of Minimum Phase Nonlinear Systems, *IEEE Trans. Automat. Contr.*, Vol.36, 1228-1240, (1991).
- [2] S.Commuri and F.L.Lewis, CMAC Neural Networks for Control of Nonlinear Dynamical Systems: Structure, Stability and Passivity, *Automatica*, vol.33, no.4, 635-641, 1996.
- [3] J.J.Hopfield, Neurons with grade response have collective computational properties like those of a two-state neurons, *Proc. of the National Academy of Science, USA*, vol. 81, 3088-3092, 1984.
- [4] P.A.Ioannou and J.Sun, *Robust Adaptive Control*, Prentice-Hall, Inc, Upper Saddle River: NJ, 1996
- [5] S.Jagannathan and F.L. Lewis, Identification of Nonlinear Dynamical Systems Using Multilayered Neural Networks, *Automatica*, vol.32, no.12, 1707-1712, 1996.
- [6] K.Kaszakurewics and A.Bhaya, On a Class of Globally Stable Neural Circuits, *IEEE Trans. on Circuit and Systems-I*, Vol. 41, 171-174, 1994.
- [7] M.Forti, S.Manetti and M.Marini, Necessary and Sufficient Condition for Absolute Stability of Neural Networks, *IEEE Trans. on Circuit and Systems-I*, Vol. 41, 491-494, 1994.

- [8] K.Matsouka, Stability Conditions for Nonlinear Continuous Neural Networks with Asymmetric Connection Weights, *Neural Networks*, Vol.5, 495-500, 1992
- [9] A.S.Poznyak, W.Yu, E.N.Sanchez and J.P.Perez, Nonlinear Adaptive Trajectory Tracking Using Dynamic Neural Networks, *IEEE Trans. on Neural Networks*, Vol.10, No.5, 1402-1411, 1999.
- [10] G.V.Puskorius and L.A.Feldkamp, Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks, *IEEE Trans. on Neural Networks*, Vol.5, No.2, 279-297, 1994
- [11] G.A.Rovithakis and M.A.Christodoulou, Adaptive Control of Unknown Plants Using Dynamical Neural Networks, *IEEE Trans. on Syst., Man and Cybern.*, Vol. 24, 400-412, 1994.
- [12] E.N.Sanchez, J.P. Perez, Input-to-State Stability (ISS) Analysis for Dynamic Neural Networks, *IEEE Trans. Circuits and Systems, Part I*, Vol.46 , Nov.11, 1395 -1398, 1999.
- [13] E.D.Sontag and Y.Wang, On Characterization of the Input-to-State Stability Property, *System & Control Letters*, Vol.24, 351-359, 1995
- [14] J.A.K.Suykens, J.Vandewalle and B.De Moor, Lur'e Systems with Multilayer Perceptron and Recurrent Neural Networks; Absolute Stability and Dissipativity, *IEEE Trans. on Automatic Control*, Vol.44, 770-774, 1999.
- [15] J.A.K. Suykens, J. Vandewalle, B. De Moor, NLq Theory: Checking and Imposing Stability of Recurrent Neural Networks for Nonlinear Modelling, *IEEE Transactions on Signal Processing (special issue on neural networks for signal processing)*, Vol.45, No.11, 2682-2691, 1997.
- [16] J.A.K.Suykens, B.De Moor, J.Vandewalle, NLq Theory: A Neural Control Framework with Global Asymptotic Stability Criteria, *Neural Networks*, Vol.10, No.4, 615-637, 1997.

- [17] J.C.Willems, Least Squares Optimal Control and Algebraic Riccati Equations, *IEEE Trans. Automat. Contr.*, vol.16, 621-634, 1971,
- [18] Wen Yu and Alexander S.Poznyak, Indirect Adaptive Control via Parallel Dynamic Neural Networks, *IEE Proceedings - Control Theory and Applications*, Vol.146, No.1, 25-30, 1999.
- [19] Wen Yu and Xiaoou Li, Some Stability Properties of Dynamic Neural Networks, *IEEE Trans. Circuits and Systems, Part I*, Vol.48, No.1, 256-259, 2001.
- [20] Wen Yu and Xiaoou Li, New Results on System Identification with Dynamic Neural Networks, *IEEE Trans. Neural Networks*, Vol.12, No.2, 412-417, 2001
M.Brown, C.J.Harris, *Neurofuzzy adaptive modelling and control*, Prentice Hall, 1994.
- [21] G.Cybenko, Approximation by Superposition of Sigmoidal Activation Function, *Math. Control, Sig Syst*, Vol.2, 303-314, 1989
- [22] B.Egardt, *Stability of Adaptive Controllers*, Lecture Notes in Control and Information Sciences, Vol.20, Springer-Verlag, Berlin, 1979
- [23] Z.Feng and A.N.Michel, Robustness Analysis of a Class of Discrete-Time Systems with Applications to Neural Networks, *Proc. of American Control Conference*, 3479-3483, San Deigo, 1999
- [24] S. S. Ge, T.H. Lee and C.J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*, World Scientific, London, Series in Robotics and Intelligent Systems-Vol. 19, 1998.
- [25] L.Jin ad M.M.Gupta, Stable Dynamic Backpropagation Learning in Recurrent Neural Networks, *IEEE Trans. Neural Networks*, Vol.10, No.6, 1321-1334, 1999.
- [26] Z.P.Jiang and Y.Wang, Input-to-State Stability for Discrete-Time Nonlinear Systems, *Automatica*, Vol.37, No.2, 857-869, 2001.

- [27] E.B.Kosmatopoulos, M.M.Polycarpou, M.A.Christodoulou and P.A.Ioannou, High-Order Neural Network Structures for Identification of Dynamical Systems, *IEEE Trans. on Neural Networks*, Vol.6, No.2, 442-431, 1995
- [28] F.L.Lewis, A.Yesildirek and K.Liu, Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance, *IEEE Trans. Neural Networks*, Vol.7, No.2, 388-399, 1996.
- [29] K.S.Narendra and K.Parthasarathy, Identification and Control of Dynamical Systems Using Neural Networks, *IEEE Trans. Neural Networks*, Vol.1, No.1, 4-27, 1990.
- [30] M.M.Polycarpou and P.A.Ioannou, Learning and Convergence Analysis of Neural-Type Structured Networks, *IEEE Trans. Neural Networks*, Vol.3, No.1, 39-50, 1992
- [31] Q.Song, Robust Training Algorithm of Multilayered Neural Networks for Identification of Nonlinear Dynamic Systems, *IEE Proceedings - Control Theory and Applications*, Vol.145, No.1, 41-46,1998
- [32] Q.Song, J.Xiao and Y.C.Soh, Robust Backpropagation Training Algorithm for Multilayered Neural Tracking Controller, *IEEE Trans. Neural Networks*, Vol.10, No.5, 1133-1141, 1999.
- [33] *Wen Yu, Passive Equivalence of Chaos in Lorenz System, *IEEE Trans. Circuits and Systems, Part I*, Vol.46, No.7, 876-879, 1999
- [34] W.Yu and X. Li, Some Stability Properties of Dynamic Neural Networks, *IEEE Trans. Circuits and Systems, Part I*, Vol.48, No.1, 256-259, 2001.
- [35] W.Yu and X. Li, Some New Results on System Identification with Dynamic Neural Networks, *IEEE Trans. Neural Networks*, Vol.12, No.2, 412-417, 2001.
- [36] W.Yu, A.S. Poznyak and X.Li, Multilayer Dynamic Neural Networks for Nonlinear System On-line Identification, *International Journal of Control*, Vol.74, No.18, 1858-1864,2001.
- A.Albert, *Regression and the Moore-Penrose Pseudoinverse*, Academic Press, 1972.

- [37] J.B.D.Cabrera and K.S.Narendra, Issues in the application of neural networks for tracking based on inverse control, *IEEE Trans. Automatico Control*, Vol.44, No.11, 2007-2027, 1999.
- [38] F.C.Chen and H.K.Khalil, Adaptive control of nonlinear systems using neural networks, *Int.J.Control*, Vol.55,No.6, 1299-1317,1992
- [39] F.C.Chen and H.K.Khalil, Adaptive control of a class of nonlinear discrete-time systems using neural networks, *IEEE Trans. Automatico Control*, Vol.40, No.5, 791-801, 1995.
- [40] k.Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Networks*, Vol.4 No.5,251-257,1991

