

Automatización de una Grúa Torre Mediante Lógica Neurodifusa

José C. Chahuara Q., Guillermo Tejada M.

Facultad de Ingeniería Electrónica y Eléctrica, Universidad Nacional Mayor de San Marcos, Lima, Perú

RESUMEN: El artículo describe el funcionamiento de una grúa torre mediante algoritmos computacionales basados en la lógica neurodifusa, los cuales han sido probados en un simulador, desde donde se han obtenido resultados que han sido comparados con los obtenidos por otros investigadores quienes han aplicado otras técnicas de automatización.

Palabras Claves: Grúa torre, Control, Neuro-Difuso, Difuso, Redes neuronales, Realimentación de Ganancias, Dual de Anti-Oscilación y Posición, Realimentación Retrasada.

I. INTRODUCCIÓN

Por ser la grúa torre una máquina muy compleja de características multivariable y no lineal, el presente estudio propone controlar su funcionamiento mediante herramientas computacionales de lógica Neuro-Difusa, los datos experimentales relativos al funcionamiento de la grúa han sido extraídos de la operación manual de un simulador implementado en Matlab [1], de esta forma los movimientos realizados por la grúa, según los comandos del usuario para alcanzar su objetivo, han quedado numéricamente registrados en una base de datos. Los datos experimentales han sido utilizados para diseñar los algoritmos del controlador Neuro-Difuso realizado con el sistema de desarrollo fuzzyTECH. Los resultados de la simulación del funcionamiento automático de la grúa, con los algoritmos Neuro Difusos, han sido comparados y han demostrado en algunos casos ser superiores a los obtenidos por Omar Hanafy y Amjed Al-Mousa de la universidad Virginia Tech, quienes realizaron investigaciones con las técnicas de: Realimentación de Ganancias, Dual de Anti-Oscilación y Posición, Realimentación Retrasada y Difuso, probados sobre un

simulador equivalente al nuestro.

II. DESCRIPCIÓN DE PARÁMETROS

Una grúa torre es un sistema de múltiples grados de libertad. Como se muestra en la figura 1, sobre la torre se sitúa una pluma horizontal de movimiento rotacional sobre la cual hay un carro de movimiento radial y un cable de movimiento vertical sometido a variaciones de longitud, en cuyo extremo péndula una carga. La posición espacial de la carga, con respecto a un origen de coordenadas situado en la intersección de los ejes que atraviesan longitudinalmente la torre y la pluma de la grúa, está determinada por cinco variables: el ángulo rotacional $\varphi(t)$, la posición radial $r(t)$, la longitud del cable $L(t)$ y los ángulos de oscilación $\theta(t)$ y $\phi(t)$.

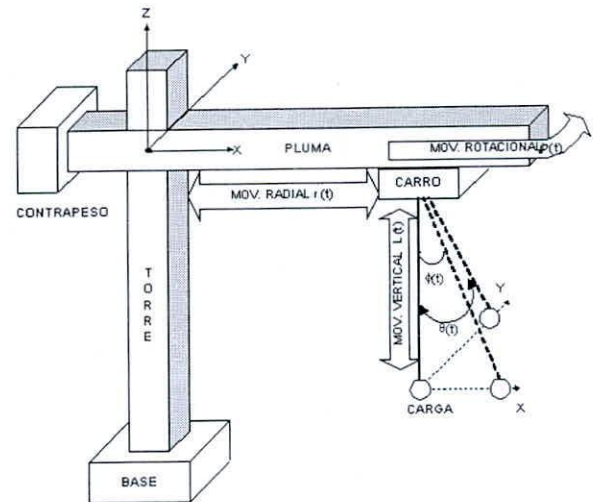


Fig. 1. Estructura y movimientos en una grúa torre

Los cambios en los valores de las variables $\varphi(t)$, $r(t)$ y $L(t)$, producen variaciones en los ángulos de oscilación $\phi(t)$ y $\theta(t)$. El ángulo $\phi(t)$ es el ángulo que traza el cable con el eje Z sobre el plano XZ y el ángulo $\theta(t)$ es el ángulo que traza el cable con el eje Z sobre el plano YZ . El objetivo de un operador humano

o controlador automático, es mover la carga rápidamente hacia su posición de destino manteniendo pequeños los ángulos de oscilación $\theta(t)$ y $\phi(t)$.

III. TRABAJOS PREVIOS

Se ha realizado un análisis de trabajos previos de investigadores que han propuesto otras técnicas para automatizar una grúa torre; sus resultados han servido como referencia para compararlos con los obtenidos en esta investigación.

En [2] Omar Hanafy presentó las técnicas: a) Realimentación de Ganancias y b) Anti Oscilación y Posición. Para (a) diseñó dos tipos de algoritmos, uno cuando la realimentación de las variables es total y otro cuando es parcial. Para (b) utilizó un controlador clásico Proporcional Derivativo (PD) para la posición y un Control de Retraso de Tiempo y Control Difuso para la oscilación.

En [3] Amjed Al-Mousa presentó las técnicas: a) Realimentación Retrasada y b) Controlador Difuso. Para (a) realimentó los estados de la grúa retrasados un valor Δt en el tiempo. Para (b) utilizó un controlador radial y un controlador rotacional con dos bases de reglas difusas, el primero para controlar la posición, y el segundo para controlar la oscilación, ambas respuestas son luego combinadas como una suma ponderada para obtener la señal de control.

IV. AUTOMATIZACIÓN NEURO-DIFUSO

En la figura 2, se observa al sistema de automatización, que contiene los algoritmos neurodifusos, y al simulador de la grúa, el cual fue descrito en [1]. El sistema consta de tres Bases de Reglas que controlan el movimiento radial (BRr I, BRr II, BRr III) y otras similares que controlan el movimiento rotacional (BR γ I, BR γ II, BR γ III).

Las entradas de BRr I y BR γ I son las distancias a las posiciones de destino radial y rotacional (d_r y d_γ) y los ángulos de oscilación producidos (ϕ y θ); las salidas son las aceleraciones a_{rr} y $a_{r\gamma}$. Para el caso de los BRr II y BR γ II, las entradas son las velocidades del carro y la pluma (d_r' y d_γ') y las velocidades angulares de oscilación (ϕ' y θ'); las salidas son aceleraciones a_{or} y $a_{o\gamma}$. Las bases de reglas BRr III y BR γ III combinan las aceleraciones a_{rr} , a_{or} y $a_{r\gamma}$, $a_{o\gamma}$ para obtener las aceleraciones de control radial r'' y rotacional γ'' respectivamente.

Las limitaciones de FuzzyTECH no permiten tener más de 343 reglas por cada base de reglas, por lo cual no fue posible reemplazar las tres bases de reglas para

el movimiento radial (BRr I, BRr II, BRr III) y las tres bases de reglas para el movimiento rotacional (BR γ I, BR γ II y BR γ III) por una sola en cada caso, ya que hubiese sido necesario almacenar 16807 reglas por base. El número elevado de reglas es consecuencia de que se tienen siete funciones de pertenencia difusas para cada una de las variables de entrada (ϕ , d_r , ϕ' , d_r' y θ , d_γ , θ' , d_γ') y otras siete funciones de pertenencia difusas para cada una de las variables de salida (r'' y γ'').

De esta manera, antes del entrenamiento, cada BR contiene 343 reglas, en algunas se les asignó un peso de 1 y al resto un peso de 0.

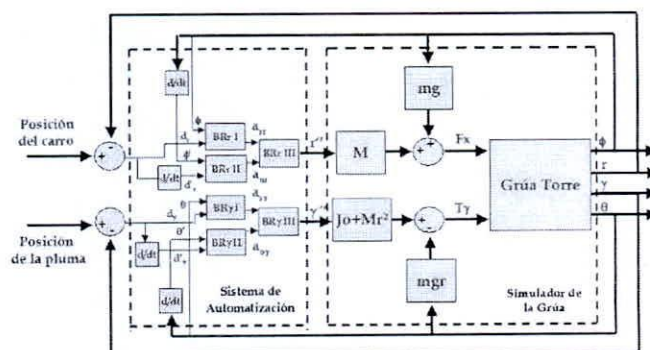


Fig. 2. Sistema de automatización de la grúa

El desarrollo del sistema Neuro-Difuso ha seguido los siguientes pasos: recolección de datos experimentales, selección de los datos más representativos, diseño de una base de reglas obvias, entrenamiento y optimización del sistema.

A. Recolección de datos experimentales

La recolección de datos experimentales ha consistido en registrar los valores de posición, velocidad y aceleración que generó el simulador [1] mientras era manejado manualmente a través de la consola que se muestra en la figura 3. En el archivo *entrenadat.xls* se registraron 2814 muestras para el movimiento radial y una cifra similar para el movimiento rotacional.

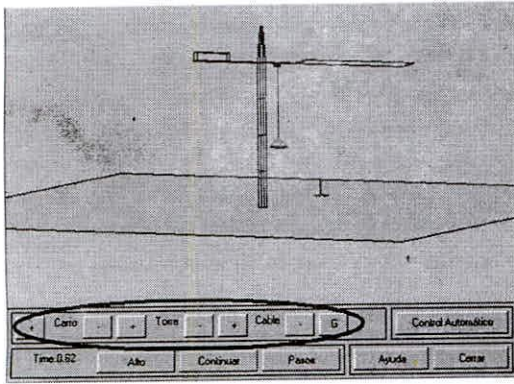


Fig. 3. Simulador en operación manual para la recolección de datos

B. Selección de los datos más representativos (clustering)

Antes de entrenar el sistema de control Neuro-Difuso, fue necesario seleccionar los datos más representativos del total que fueron extraídos durante la operación manual del simulador. Este procedimiento se realizó utilizando el algoritmo de agrupamiento de datos contenido en uno de los módulos (*Clustering Module*) de fuzzyTECH; el cual se basa en la distancia geométrica entre los datos para eliminar los redundantes y no consistentes [4], [5].

Este procedimiento permitió reducir los datos de 2814 a 112 para el movimientos radial y de 2814 a 51 para el movimiento rotacional, permitiendo que aumenten las probabilidades de convergencia al entrenar el sistema Neuro-Difuso y que disminuya el tiempo de entrenamiento.

El entrenamiento realizado sólo con datos relevantes, permite obtener un conjunto mínimo de reglas lógicas, cada una de ellas asociadas a un peso numérico que define su importancia para el funcionamiento deseado del sistema.

C. Diseño de la base de reglas obvias

En las tablas I, II y III se representan las bases de reglas, cada variable de las filas y columnas poseen cada una siete funciones de pertenencia (PL, PM, PS, ZE, NS, NM, NL) y cada celda (intersección de cada fila con columna) tienen siete posibles respuestas deseadas (PL, PM, PS, ZE, NS, NM, NL), esto da un total de 343 (7*7*7) reglas en cada caso con un peso asociado por defecto igual a cero. Sin embargo, para dar las condiciones iniciales del entrenamiento, se han considerado las reglas obvias, es decir aquellas cuyas salidas puedan predecirse siguiendo un criterio lógico de funcionamiento. Así, por ejemplo, en la tabla I, es lógico considerar como regla obvia aquella que enuncie

que “si el carro se encuentra lejos de la posición de destino (PL) y el ángulo de oscilación es grande y positivo (PL) entonces la aceleración debe ser nula (ZE)”, pues de lo contrario el ángulo de oscilación se incrementaría.

A todas las reglas obvias se les asignó un peso inicial de 1, mientras las demás permanecieron en su valor 0 por defecto. En las tablas, se muestran dentro de cada celda sólo las reglas cuyas respuestas son obvias. Sin embargo, después del entrenamiento los pesos de las reglas (0 y 1) designados previamente se alteraron de acuerdo a su contribución real para el alcance de la respuesta deseada del sistema.

1) *Determinación de Reglas Obvias para BRr I:* las entradas del bloque son las variables de distancia a la posición deseada del carro (d_r) y el ángulo de oscilación producido (ϕ). Como salida se tiene la aceleración a_{rr} . Para cada variable se tienen las siguientes funciones de pertenencia:

$$d_r = \{PL, PM, PS, ZE, NS, NM, NL\}$$

$$\phi = \{PL, PM, PS, ZE, NS, NM, NL\}$$

$$a_{rr} = \{PL, PM, PS, ZE, NS, NM, NL\}$$

donde, P: Positivo, N: Negativo, L: Largo, M: Mediano, S: Pequeño, ZE: Cero, d_r : distancia a la posición de destino, ϕ : ángulo de oscilación, a_{rr} : aceleración.

TABLA I
REGLAS OBIVAS DEL CARRO UTILIZANDO ϕ Y d_r

		ϕ						
		PL	PM	PS	ZE	NS	NM	NL
d_r	PL	ZE	PS	PS	PM	PM	PL	PL
	PM	NS	ZE	PS	PS	PM	PM	PL
	PS	NS	NS	ZE	PS	PS	PM	PM
	ZE	NM	NS	NS	ZE	PS	PS	PM
	NS	NM	NM	NS	NS	ZE	PS	PS
	NM	NL	NM	NM	NS	NS	ZE	PS
	NL	NL	NL	NM	NM	NS	NS	ZE

2) *Determinación de Reglas Obvias para BRr II:* en este caso las entradas son la velocidad del actuador (d_r') y la velocidad angular (ϕ') y la salida la aceleración a_{or} . Para cada variable se tienen las siguientes funciones de pertenencia:

$$d_r' = \{PL, PM, PS, ZE, NS, NM, NL\}$$

$$\phi' = \{PL, PM, PS, ZE, NS, NM, NL\}$$

$$a_{or} = \{PL, PM, PS, ZE, NS, NM, NL\}$$

donde, P: Positivo, N: Negativo, L: Largo, M: Mediano, S: Pequeño, ZE: Cero, dr': Velocidad radial, ϕ' : Velocidad angular, a_{or} : aceleración.

La tabla II, muestra el conjunto de reglas obvias que fueron consideradas para BRr II.

TABLA II
REGLAS OBIAS DEL CARRO UTILIZANDO DR' Y ϕ'

		ϕ'						
		PL	PM	PS	ZE	NS	NM	NL
dr'	PL	ZE	NS	NS	NM	NM	NL	NL
	PM	PS	ZE	NS	NS	NM	NM	NL
	PS	PS	PS	ZE	NS	NS	NM	NM
	ZE	PM	PS	PS	ZE	NS	NS	NM
	NS	PM	PM	PS	PS	ZE	NS	NS
	NM	PL	PM	PM	PS	PS	ZE	NS
	NL	PL	PL	PM	PM	PS	PS	ZE

3) *Determinación de Reglas Obvias para BRr III:* o la base de reglas BRr III hace las veces de combinador de las dos aceleraciones (a_{rr} y a_{or}) originadas por las bases de reglas BRr I y BRr II. Su salida es la aceleración radial del carro de la grúa (r''). En este caso no se necesitan más de 13 reglas obvias para modelar el combinador como se muestra en la tabla III. Las otras 330 reglas mantuvieron su peso por defecto de 0.

TABLA III
REGLAS OBIAS DEL COMBINADOR DE a_{or} Y a_{rr}

		a_{rr}						
		PL	PM	PS	ZE	NS	NM	NL
a_{or}	PL	PL						ZE
	PM		PM				ZE	
	PS			PS		ZE		
	ZE				ZE			
	NS			ZE		NS		
	NM		ZE				NM	
	NL	ZE						NL

Este mismo procedimiento se ha seguido con el resto de bases de reglas del controlador Neuro-Difuso.

D. Fusificación, Inferencia y Defusificación

En la etapa de Fusificación se utilizaron las funciones de pertenencia de forma geométrica triangular. Como se muestra en la figura 4, son siete conjuntos difusos (PL, PM, PS, ZE, NS, NM y NL) por variable.

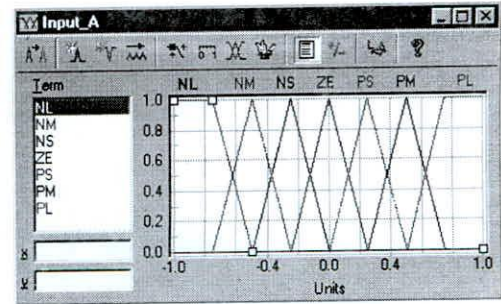


Fig. 4. Funciones de pertenencia de la variable lingüística d_r' (distancia radial)

En cuanto a los operadores de inferencia, se seleccionaron el MIN para la agregación de las entradas, el PROD para la composición y el BSUM y MAX para la agregación de salida.

En la etapa de defusificación, así como en el caso de la fusificación, se utilizaron las funciones de pertenencia de forma geométrica triangular que muestra la figura 5, como se puede apreciar, también consta de 7 conjuntos difusos en cada caso (r'' y γ'').

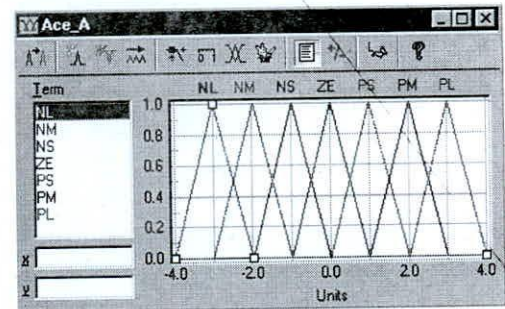


Fig. 5. Términos de la variable lingüística r'' (aceleración radial)

El algoritmo de defusificación utilizado para obtener las señales de control r'' y γ'' fue el centro del máximo (CoM – Center of Maximum), por ser el más utilizado en aplicaciones de control.

E. Entrenamiento y Optimización

Para entrenar el sistema se utilizaron el módulo Neurofuzzy de fuzzyTECH y los datos más relevantes del funcionamiento de la grúa, obtenidos utilizando la técnica de agrupamiento de datos de fuzzyTECH [4], [5]. El método de entrenamiento utilizado fue el Random; la selección de datos fue aleatoria, debido a la amplitud del tamaño de las muestras; las condiciones de convergencia fueron: desviación máxima entre [50% - 1%] con un factor de 0.75 y desviación promedio de 0.1%. La figura 6, presenta algunas reglas de BRr III, después del entrenamiento.

Spreadsheet Rule Editor - RB5				
#	IF	THEN	DoS	Acc_A
	Output_A	Output_B		
41	NS	ZE	0.50	ZE
42	NS	ZE	0.02	PS
43	NS	ZE	0.07	PM
44	NS	ZE	0.09	PL
45	NS	PS	0.14	NL
46	NS	PS	0.02	NM
47	NS	PS	1.00	ZE
48	NS	PS	0.19	PS
49	NS	PS	0.08	PL
50	NS	PM	0.16	NL
51	NS	PM	0.06	NS

Fig. 6. Algunas reglas generadas mediante el proceso de entrenamiento

La figura 7, muestra el proceso de entrenamiento y optimización que permite sintonizar el funcionamiento del controlador a través de la modificación manual de algunos parámetros del sistema Neuro-Difuso, tales como los pesos de las reglas y los rangos de los conjuntos difusos, hasta que el sistema alcance el desempeño deseado.

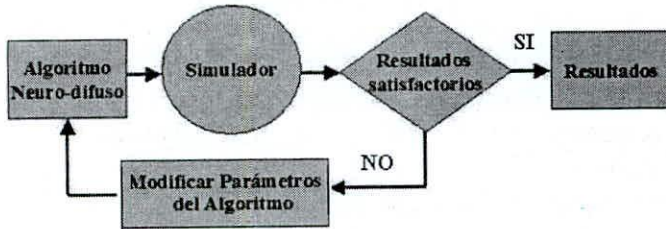


Fig. 7. Proceso de entrenamiento y optimización

V. RESULTADOS

A continuación se presenta la comparación de los resultados obtenidos con el controlador Neuro-Difuso y los obtenidos por los investigadores Hanafy y Al-Mousa. Las figuras 8 a 10 muestran los casos para el movimiento radial (sólo se desplaza el carro), rotacional (sólo se desplaza la pluma) y compuesto (se desplazan el carro y la pluma al mismo tiempo) respectivamente. Dentro de cada grafico, los resultados del algoritmo Neuro-Difuso se muestran con una línea sólida, mediante la abreviatura NF. Los resultados de los tres algoritmos desarrollados por Hanafy se muestran con líneas punteadas, con las abreviaturas:

- H FULL (Realimentación de Ganancias con Total de Variables).
- H DELA (Anti Oscilación de Realimentación Retrasada).
- H FUZZ (Anti Oscilación PD Difuso).

Los resultados de los dos algoritmos desarrollados por Al-Mousa se muestran también con líneas punteadas, con las abreviaturas:

- A FUZZ (Difuso).
- A DELA (Realimentación retrasada).

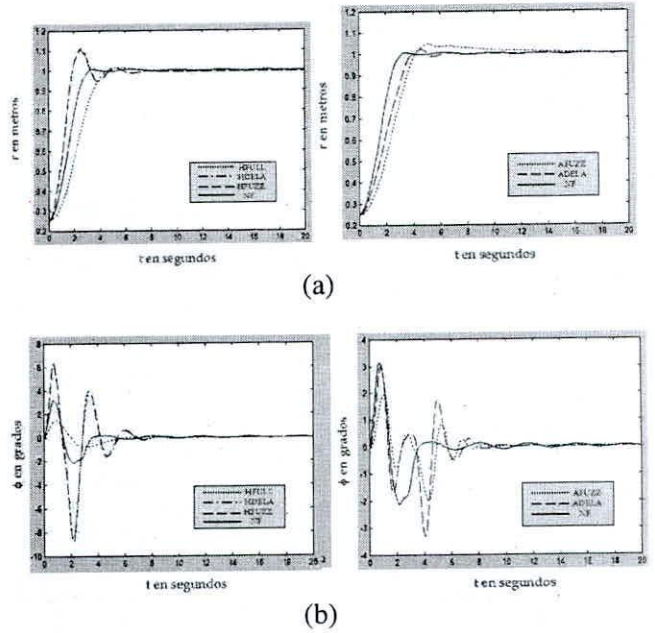
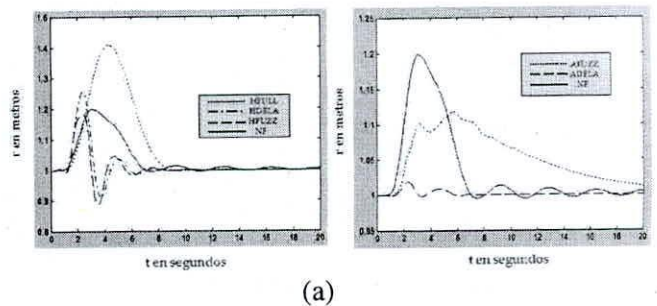


Fig. 8. Curvas de respuesta del movimiento radial. (a) Posición del carro (b) Oscilación de ϕ



(a)

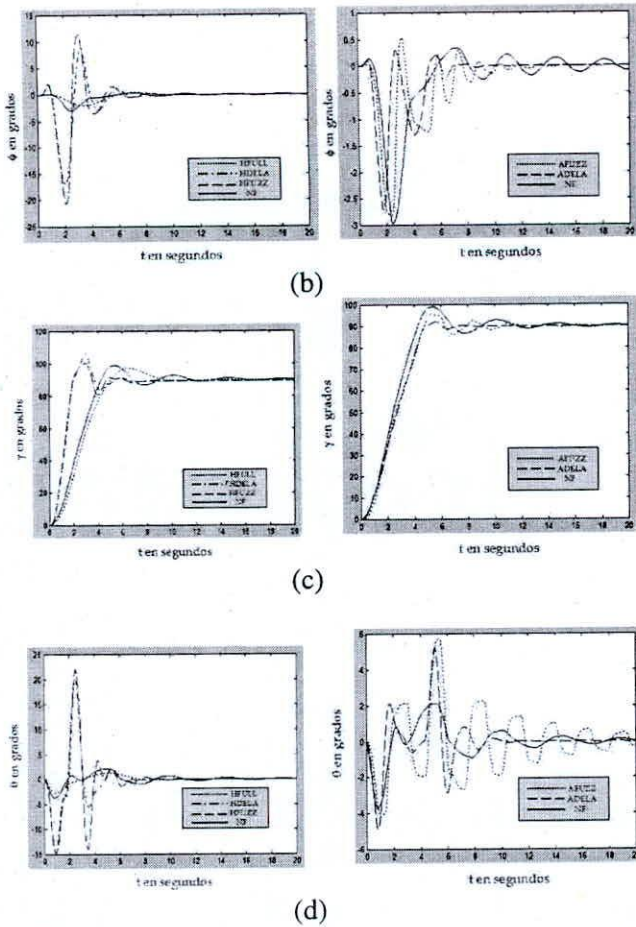


Fig.9. Curvas de respuesta del movimiento rotacional. (a) Desplazamiento de la pluma (b) Oscilación de θ (c) Desplazamiento del carro (d) Oscilación de ϕ

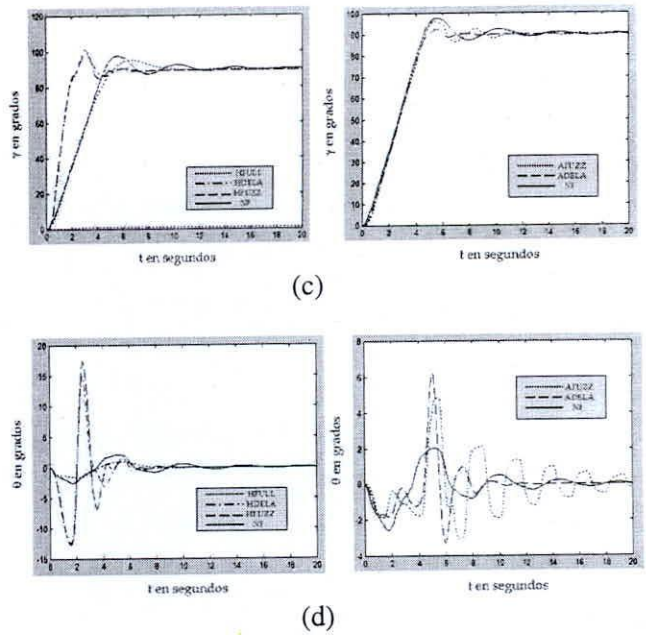
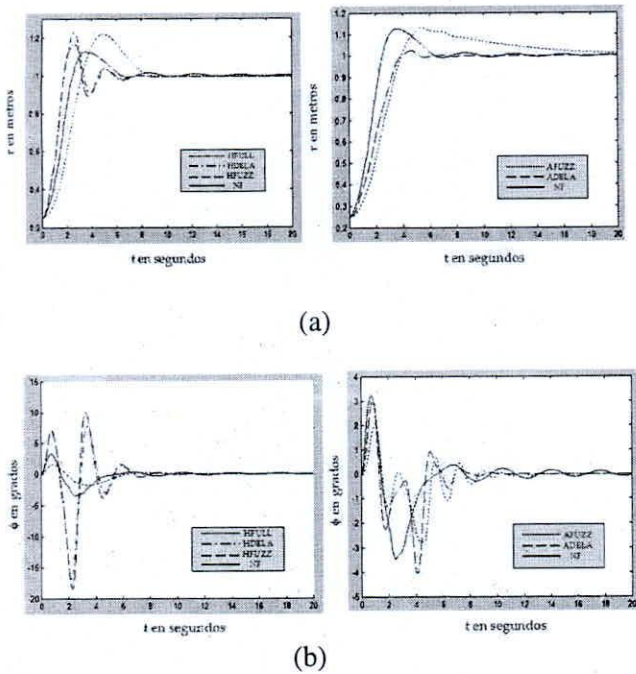


Fig.10. Curvas de respuesta del movimiento compuesto. (a) Desplazamiento del carro (b) Oscilación de ϕ (c) Desplazamiento de la pluma (d) Oscilación de θ

VI. CONCLUSIONES

El algoritmo Neuro-Difuso y los algoritmos de Al Mousa siempre mantienen ángulos de oscilación (ϕ y θ) pequeños a diferencia de los algoritmos de Realimentación Retrasada y Difuso de Hanafy que, como se observa en las figuras 9 (b, d) y 10 (b, d), producen ángulos de oscilación hasta de 20° que ponen en riesgo la seguridad de la carga.

En el movimiento radial, cuando el carro se desplaza un metro a lo largo de la pluma, el algoritmo Neuro-Difuso produce un tiempo de establecimiento menor que los algoritmos de Hanafy y Al Mousa (ver figura 8).

En el caso del movimiento rotacional, cuando la pluma rota teniendo como centro la torre, el algoritmo Neuro-Difuso tiene un tiempo de establecimiento mayor que los algoritmos de Hanafy y Al-Mousa (ver figura 9).

En el caso del movimiento compuesto, cuando el carro y la pluma de la grúa tienen movimientos simultáneos, el algoritmo Neuro-Difuso y el algoritmo de retraso de tiempo de Al-Mousa, producen un tiempo de establecimiento para el carro ligeramente menor que los otros algoritmos de Al-Mousa y los de Hanafy. Pero el algoritmo Neuro-Difuso produce un tiempo de establecimiento ligeramente mayor para la pluma que los otros algoritmos (ver figura 10).

Sólo en los casos que se presentan en las figuras 9 (c) y 10 (c), se ha observado que el tiempo de

establecimiento de los algoritmos desarrollados en la universidad *Virginia Tech* son notoriamente menores que el tiempo de establecimiento del algoritmo Neuro-Difuso, pero como se indicó anteriormente, las magnitudes de los ángulos de oscilación que producen ponen en riesgo la seguridad de la carga, lo cual no ocurre con el algoritmo Neuro-Difuso. En consecuencia, en estos casos, el algoritmo Neuro-Difuso sacrifica tiempo de establecimiento para otorgar seguridad a la carga.

El algoritmo Neuro-Difuso ha demostrado en el simulador mucha flexibilidad para mantener el mismo desempeño a pesar que se ha incluido nuevos parámetros físicos como son: fricciones, variaciones en la longitud del cable, y variaciones en la magnitud de la masa de la carga. Esta característica se contrapone a las características de los algoritmos de control clásicos que por estar basados en modelos matemáticos rígidos y complicados la inclusión de nuevos parámetros puede llegar a ser una tarea sumamente complicada.

BIBLIOGRAFÍA

- [1] J. C. Chahuara y G. Tejada, *Simulador Animado de Grúa Torre para Aplicaciones de Investigación, Revista Electrónica – UNMSM, No. 14*, pp. 3-14, dic. 2004.
- [2] O.Hanafy, *Control of Gantry and Tower Cranes. Phd. Thesis*, Department of Mechanics Engineering, Virginia Tech- Virginia, 2003.
- [3] A. Al-Mousa, *Control of Rotary Cranes Using Fuzzy Logic and Time-Delayed Position Feedback Control. M.Sc. Thesis*, Department of Electrical Engineering, Virginia Tech- Virginia, 2000.
- [4] FuzzyTECH 5.3 User's Manual, Inform gmbh, 1999.
- [5] J. M. Buhman, *Data Clustering and Learning. Hand Book Theory and Neural Networks*, MIT Press 2002.