

# Generación de señales de referencia de posición para el desplazamiento de una grúa viajera de dos grados de libertad

I. Torres-Rodríguez<sup>1</sup>  
P. Huerta-González<sup>1</sup>  
J. Rodríguez-Rivas<sup>2</sup>

<sup>1</sup>Ingeniería en Control y Automatización (ICA),  
Escuela Superior de Ingeniería Mecánica y Eléctrica (ESIME),  
Instituto Politécnico Nacional (IPN), UPALM,  
Edif. 3, 1er piso, Col. Lindavista, México, DF.

<sup>2</sup>Sección de Estudios de Posgrado e Investigación (SEPI),  
Escuela Superior de Ingeniería Mecánica y Eléctrica (ESIME),  
Instituto Politécnico Nacional (IPN), UPALM,  
Ed. Z4, 1er piso, Col. Lindavista, México, DF.  
MÉXICO.

Correo electrónico: ictorres@ipn.mx  
phuerta@ipn.mx  
jrodriguezr@ipn.mx

Recibido el 1 de septiembre de 2008; aceptado el 10 de febrero de 2009.

## 1. Resumen

En este trabajo se utiliza una red neuronal artificial de tres capas, la cual se entrena usando el algoritmo de retropropagación de Levenberg-Marquardt para obtener el tiempo final o de llegada ( $t_f$ ) del polinomio de Bézier de quinto orden y generar una señal suave y continua, la cual será utilizada como señal de referencia de posición de una grúa viajera. En su estructura básica, una curva de Bézier queda totalmente definida por cuatro puntos característicos, los puntos inicial y final de la curva (nodos) y dos puntos de control (o manejadores), invisibles en el gráfico final, que delimitan su forma. Así también la curva de Bézier puede definirse en cualquier intervalo  $[t_0, t_f]$ . De esta manera, esta señal de referencia de posición, puede emplearse para cualquier tipo de control siendo indistinto el motor a utilizar, el cual es el que acciona a la grúa.

**Palabras clave:** polinomio de Bézier, red neuronal artificial, grúa viajera.

## 2. Abstract (Position Reference Signal Generation used in a Crane Motion Control with Two Degrees of Freedom)

This paper presents the generation of the position reference signal used in a crane control system. The reference signal was generated by a three-layer artificial neural network, which was trained offline using the Levenberg-Marquardt algorithm. Attention is focused on computing the final time or arrived time ( $t_f$ ) based on the solution of the five-order Bézier polynomial, which permits to get a smoothing and continuing signal that may be used as position reference signal in a crane control system. The Bézier curve can be defined by four characteristic points: the ending points or nodes, and two control or drive points. This position reference signal may be used in whatever control system and drive.

**Key words:** Bézier polynomial, artificial neural network, crane.

## 3. Introducción

El control de la grúa viajera se divide en dos áreas principales, la planeación del movimiento o señal de referencia [1] y el control del movimiento, las cuales logran que la grúa viajera siga un camino predeterminado. En este artículo se enfoca solamente la etapa de planeación del movimiento, ya que la etapa de control se tratará en otro artículo.

Es necesario que la grúa viajera siga una señal de referencia determinada que permita evaluar la precisión con que funciona el lazo de control de posición, tanto en régimen transitorio como en régimen permanente. Esta evaluación se realiza midiendo la desviación existente entre la señal de referencia dada al motor y el giro producido en él. Cuanto menor sea esta desviación, más se ajustará el movimiento de la grúa viajera al que ha sido definido por la trayectoria de referencia.

## 4. Curva de Bézier

Para trazar una curva suave desde un punto inicial hasta un punto final, y que ésta sea afectada en su trayectoria por un conjunto de puntos que describen un polígono de apoyo,

entonces el problema puede resolverse con las curvas de Bézier [2], [3]. Por lo tanto, la trayectoria debe satisfacer los siguientes requisitos:

- Deben hacer pasar al extremo por los puntos definidos por el usuario (inicio y término).
- Deben satisfacer los requisitos cinemáticos y dinámicos que imponen los motores, como son: no sobrepasar la velocidad o aceleración máximas, el par admisible, etcétera.
- Deben ser funciones temporales, al menos derivables una vez para que el movimiento o transición sea suave en todo momento.

Pierre Bézier se ocupó de determinar un método que permitiera unir dos puntos con una curva, que fuera de grado conocido además de controlarla y adecuarla con el simple hecho de ajustar los puntos de apoyo de la misma y que su solución permitiera un fácil trazado.

La principal aplicación de las curvas y superficies de Bézier se encuentran en el área de la aerodinámica ya que permiten construir modelos de estudio de una manera sencilla, sin embargo, su aplicación se extiende a diferentes áreas de la ingeniería y la arquitectura.

#### 4.1. Construcción de la curva de Bézier

La idea de definir geoméricamente las formas no es compleja: basta recordar algunas nociones de geometría plana para saber que un punto del plano puede definirse por coordenadas.

Básicamente, la curva de Bézier se define por cuatro puntos, los puntos inicial y final de la curva (nodos) y dos puntos de control (o manejadores), invisibles en el gráfico final, que definen su forma. La curva pasa forzosamente sólo por el primer y el último punto y no por los demás [2]. Al polígono que se forma al unir secuencialmente los puntos de control se le llama polígono de apoyo o de control. Las curvas de Bézier son una forma alternativa de representación de una curva polinomial.

El método de Bézier utiliza como funciones base, o de forma a los polinomios de Bernstein, cuya expresión es:

$$B_i^n(u) = \binom{n}{i} (i-u)^{n-i} \cdot u^i \quad (1)$$

donde  $\binom{n}{i} = \frac{n!}{i!(n-i)!}$  y  $n$  es el grado del polinomio.

El parámetro está definido entre  $[0, 1]$ , y el número de puntos de control es  $n + 1$ .

La ecuación del polinomio de Bézier es:

$$b(u) = \sum_{i=0}^n P_i B_i(u) \quad u \in [0,1] \quad (2)$$

donde:

- $u$  representa el punto que se quiere obtener en la curva.
- $i$  es el índice que representa cada uno de los puntos de control que definen la curva, el cual va de 0 a  $n$ .
- $P_i$  representa los puntos de control.
- $B_i$  función polinomial base de Bernstein.

En los polinomios de Bézier, todas las funciones de forma son nulas, salvó la primera y la última que valen uno para  $u = 0$  y  $u = 1$ . Esto implica que en los puntos extremos, es decir, al principio y final de la curva sólo influye uno de los puntos de control, y que en los extremos la curva pasa por el primer y último punto de control [4], [5].

Aunque sólo se haya definido la curva en el intervalo  $[0, 1]$ , es posible utilizar otros intervalos. Sólo hace falta transformar el intervalo por una aplicación afín para que la parametrización esté definida en el intervalo  $[a,b]$  [4].

$$t = \frac{u-a}{b-a} \quad u \text{ en } [a,b] \quad (3)$$

De modo que  $u = a$  en  $t = t_0 = 0$  y  $u = b$  en  $t = t_f = 1$ .

Ahora se puede decir que el punto inicial  $a$  ocurre en  $t_0$  y el punto final  $b$  en  $t_f$ . Esto nos da una relación entre el tiempo total del movimiento y la forma del polinomio generado.

#### 4.2. Polinomio de quinto grado

Con base en las propiedades del polinomio de Bézier y como se necesita que el desplazamiento de la grúa viajera sea suave, pueden emplearse polinomios de diferentes grados, pero para este trabajo se utiliza un polinomio de quinto grado tomado de la referencia [5], el cual tiene la forma:

$$b(t) = \left[ \frac{1}{(t_f-t_0)^5} \right] \left[ (t_f-t)^5 b_0 + 2(t_f-t_0)(t_f-t)^4 b_1 + 3(t-t_0)^2(t_f-t)^3 b_2 + 4(t-t_0)^3(t_f-t)^2 b_3 + 5(t-t_0)^4(t_f-t) b_4 + (t-t_0)^5 b_5 \right] \quad (4)$$

Con la ecuación (4) se pueden generar curvas en un intervalo de tiempo cualquiera y su geometría depende de la ubicación de los puntos de partida y llegada.

Para este trabajo, el valor del tiempo final o de llegada ( $t_f$ ) se determinó usando una red neuronal artificial, ya que ésta proporciona el tiempo apropiado para que la grúa viajera se mueva con una oscilación de la carga limitada en el rango del 3.0% de su apertura total, tomando en cuenta para su entrenamiento diversos valores en la masa de la carga, longitud del cable y distancias a recorrer.

## 5. Obtención del tiempo final de desplazamiento por medio de una red neuronal artificial

En este apartado se presenta un método para calcular el tiempo final,  $t_f$ , de desplazamiento que es utilizado por el polinomio de Bézier de quinto grado (ecuación 4), para obtener la señal de referencia de posición con la finalidad de que la grúa se traslade de un punto a otro manteniendo una oscilación limitada en el rango del 3.0% de su apertura total en la carga. Para obtener el tiempo  $t_f$  se utiliza una red neuronal artificial con tres capas, la cual se entrena usando el algoritmo de retropropagación de Levenberg-Marquardt. Este tipo de red es de aprendizaje supervisado que emplea un ciclo de propagación-adaptación.

La razón principal para utilizar una red neuronal artificial, es que se explota su capacidad para asociar la salida con los valores que un operario debe introducir para el funcionamiento de la grúa [6].

Con las redes neuronales artificiales se busca la solución de problemas complejos, no como una secuencia de pasos, sino como la evolución de unos sistemas de computación inspirados en el cerebro humano [7].

### 5.1. Estructura de la red neuronal artificial para la determinación del tiempo final de desplazamiento

En esta red se utilizan como patrones de entrada los valores proporcionados por el operador de la grúa, los cuales son: la distancia que la grúa recorrerá, la masa de la carga y la longitud del malacate. De esta manera el patrón de salida será el tiempo final de desplazamiento ( $t_f$ ).

Así, la red neuronal artificial tiene tres entradas y una salida, donde  $x$  es la posición (metros),  $m$  la masa de la carga (kilogramos),  $l$  la longitud del cable (metros) y  $t_f$  el tiempo final de desplazamiento (segundos). El número de capas ocultas y el número de neuronas en cada capa, dependerá de los resultados que se obtengan en el entrenamiento.

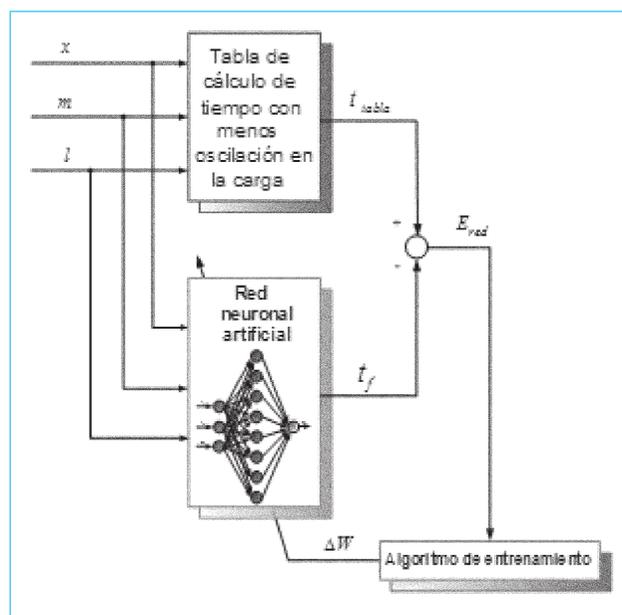


Fig. 1. Fase de entrenamiento.

Cualquier cambio que se realice en los patrones de entrenamiento exige una codificación diferente del vector de entrada y a su vez cambian las condiciones generales de la red, pero el proceso de entrenamiento sigue siendo igual [8]. Los datos de entrada a la red neuronal artificial deben normalizarse entre [0, 1] y para evitar problemas con la saturación de la neurona de salida los datos se normalizan en el rango de [0, 1] [6], [9], [10].

### 5.2. Generación de datos y entrenamiento de la red neuronal artificial

En la figura 1 se muestra la forma en que se entrenó la red neuronal artificial, el cual se llevó a cabo por medio del algoritmo de entrenamiento de retropropagación [9], [10]. Para generar los datos de la tabla del tiempo final que se usa en el entrenamiento de la red neuronal, se realizaron diversas simulaciones del sistema de la grúa viajera, utilizando diferentes distancias, longitudes del cable y cargas de la grúa viajera, tomando como base que la oscilación de la carga estuviera limitada en el rango del 3.0% de su apertura total, es decir 0.078 rad que equivale a una apertura en la parte horizontal de 15 cm para una longitud de 3 m.

Para los datos de entrada se estableció un rango de variación, lo cual depende de la forma en la cual la grúa tiene que operar, lo anterior es para obtener el tiempo final. Por lo

tanto, se establece una variación de la masa desde 0 hasta 1 500 kg que corresponde a la carga que la grúa debe transportar. En las distancias de desplazamiento de la carga, se considera que ésta puede desplazarse hasta 200 m. Finalmente, se varía la longitud del malacate, considerando que su longitud máxima es de 3 m, de acuerdo a esto, el malacate solamente puede disminuir en su longitud y se toma que su longitud mínima es de 1 m. Con los datos obtenidos de estas simulaciones, se realiza el entrenamiento de la red neuronal artificial para encontrar el tiempo final de desplazamiento de la grúa viajera con el rango de oscilación especificado.

## 6. Simulación en Matlab para realizar el entrenamiento de la red neuronal artificial

### 6.1. Capa de entradas

Los valores de entrada a la red neuronal se agruparon en el vector de tres entradas,  $p$ . Estas entradas son los valores proporcionados por el operador, que son la distancia, la masa y la longitud del malacate. Para cada una de las entradas se generó un vector fila, lo anterior se muestra a continuación (tabla 1) en donde se emplea la codificación para el archivo de MATLAB [9], [10].

Finalmente el vector de entradas  $p$  se muestra en la tabla 2.

La red neuronal artificial se entrenó con 1 200 valores en cada una de las entradas, cuyos valores fueron normalizados; de tal forma que el mayor valor de entrada a la red esté en el rango de [0 1]:

$$x_{(normalizado)} = \frac{\text{distancia de recorrido (m)}}{200\text{m}} \quad (5)$$

**Tabla 1.** Generación de las entradas de la red neuronal.

| Entrada | Código en MATLAB    |
|---------|---------------------|
| $x$     | $\text{distancia};$ |
| $m$     | $\text{carga};$     |
| $l$     | $\text{cable};$     |

**Tabla 2.** Vector de entradas.

| Código en MATLAB                                      |
|---|
| $p = [\text{distancia}; \text{carga}; \text{cable}];$ |

**Tabla 3.** Vector de salidas.

| Salida             | Código              |
|--------------------|---------------------|
| $t_{\text{tabla}}$ | $t_{\text{tabla}};$ |

**Tabla 4.** Creación de una red con una capa oculta.

| Código en MATLAB   |
|--|
| $\text{net} = \text{newff}([\text{minmax}(p)], [8 \ 1], \{\text{'tansig'}, \text{'purelin'}\}, \text{'trainlm'});$ |

$$m_{(normalizado)} = \frac{\text{masa de la carga (kg)}}{1\ 500\ \text{kg}} \quad (6)$$

$$l_{(normalizado)} = \frac{\text{longitud del malacate (m)}}{3\ \text{m}} \quad (7)$$

### 6.2. Capa de salidas

El valor de la tabla del tiempo final (tabla 3), es representado por  $t_{\text{tabla}}$  [9], [10].

Para evitar problemas con la saturación de la neurona de salida, los datos del tiempo final se normalizan, para que el valor máximo que proporcione la salida este en el rango de [0 1].

$$t_{\text{tabla}(normalizado)} = \frac{\text{tiempo de recorrido (s)}}{60\ \text{s}} \quad (8)$$

### 6.3. Entrenamiento de la red neuronal artificial

El código de entrenamiento para esta red, es desarrollado en la herramienta de redes neuronales de MATLAB. La red es creada mediante el comando *newff* para creación de redes multicapa de realimentación conocida como de retropropagación [9], [10]. Por lo tanto, se crea una red de tres capas (tabla 4). En la primera capa se tiene un vector de entradas con tres elementos, en la segunda se tiene una capa oculta con ocho neuronas y en la capa de salida hay una neurona. La función de transferencia en la capa oculta es la tan-sigmoidal y en la capa de salida la función de transferencia es lineal. El algoritmo de entrenamiento es *trainlm* (Levenberg-Marquardt).

En la tabla 5 se proporcionan las instrucciones para obtener el entrenamiento de la red para 4 000 iteraciones, la cual se simula con los valores obtenidos en el entrenamiento.

**Tabla 5.** Entrenamiento de la red neuronal artificial.

**Código en MATLAB**

```
net = train (net, p, t_tabla1)
y = sim (net, p)
```

**Tabla 6.** Parámetros para realizar el entrenamiento.

**Código en MATLAB**

```
net.trainParam.show = 50;
net.trainParam.lr = 0.0001;
net.trainParam.epochs = 4000;
net.trainParam.goal = 1e-9;
net.trainParam.max_fail = 5;
net.trainParam.mem_reduc = 1;
net.trainParam.min_grad = 1e-12;
net.trainParam.time = inf;
```

**Tabla 7.** Creación de una red con dos capas ocultas.

**Código en MATLAB**

```
net = newff ([minmax(p)], [7 7 1], {'tansig',
'tansig', 'purelin'}, 'trainlm');
```

**Tabla 8.** Características de la red con dos capas ocultas.

| Núm. capa | Tipo     | Núm. neuronas | Función de transferencia |
|-----------|----------|---------------|--------------------------|
| 1         | entradas | 3             |                          |
| 2         | oculta   | 7             | tan-sigmoidal            |
| 3         | oculta   | 7             | tan-sigmoidal            |
| 4         | salida   | 1             | lineal                   |

**Tabla 9.** Creación de una red con tres capas ocultas.

**Código en MATLAB**

```
net = newff ([minmax(p)], [10 20 10 1], {'tansig',
'tansig', 'tansig', 'purelin'}, 'trainlm');
```

**Tabla 10.** Características de la red con tres capas ocultas.

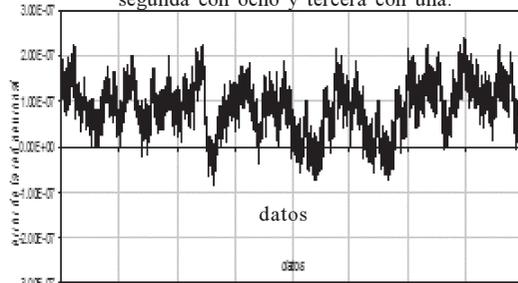
| Núm. capa | Tipo     | Núm. neuronas | Función de transferencia |
|-----------|----------|---------------|--------------------------|
| 1         | entradas | 3             |                          |
| 2         | oculta   | 10            | tan-sigmoidal            |
| 3         | oculta   | 20            | tan-sigmoidal            |
| 4         | oculta   | 10            | tan-sigmoidal            |
| 5         | salida   | 1             | lineal                   |

Los valores de iniciación de la matriz de pesos se generaron de manera aleatoria y los parámetros que determinan el entrenamiento se fijaron en los valores que se muestran en la tabla 6.

Para validar la estructura de la red neuronal anterior, se sigue el mismo procedimiento que en el caso anterior y se crean dos estructuras de redes neuronales artificiales (tablas 7 y 9), cuyas características se muestran en las tablas 8 y 10.

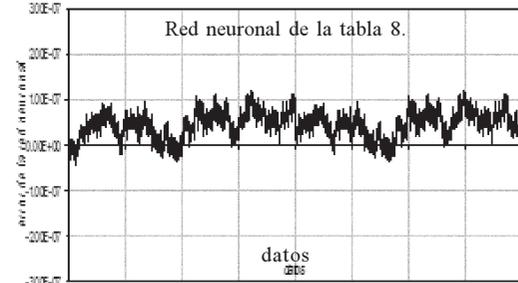
Después de crear las redes anteriores, y de haber obtenido el error entre el tiempo final de desplazamiento de referencia y la que proporciona la red neuronal artificial durante el entrenamiento, se grafican los errores de las tres redes (figura 2), en donde se observa que el error en la red neuronal descrita en la tabla 4 (figura 2c) es inferior que las redes descritas en las tablas 8 y 10 (figuras 2a y 2b).

Red neuronal con tres capas: primera capa con tres neuronas, segunda con ocho y tercera con una.



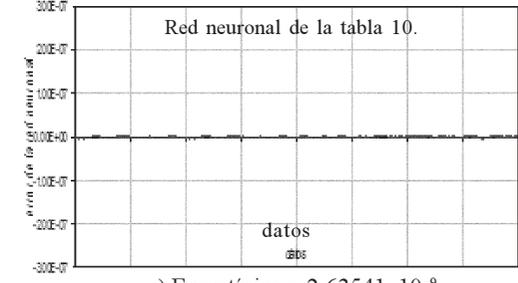
a) Error típico =  $2.449 \times 10^{-7}$

Red neuronal de la tabla 8.



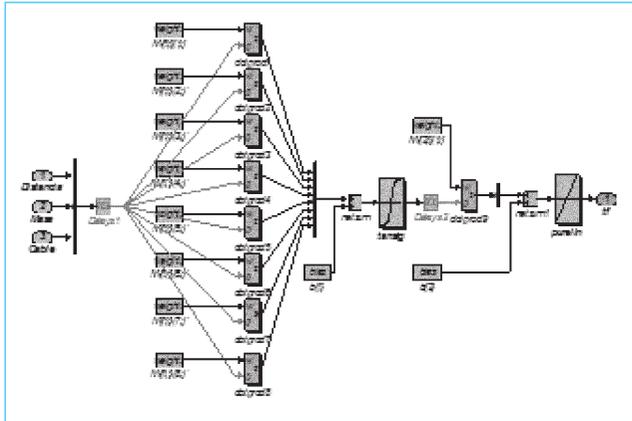
b) Error típico =  $1.134 \times 10^{-7}$

Red neuronal de la tabla 10.



c) Error típico =  $2.63541 \times 10^{-9}$

**Fig. 2.** Error entre el tiempo final de desplazamiento de referencia y la que proporciona la red neuronal.



**Fig. 3.** Configuración de la red neuronal artificial para la obtención del tiempo final de desplazamiento utilizando el software de SIMULINK.

De acuerdo a los resultados proporcionados por la red neuronal artificial de tres capas (primera capa con tres neuronas de entrada, segunda capa con ocho neuronas y tercera capa con una neurona de salida), se obtiene la configuración de la red (figura 3) por medio de la siguiente instrucción `gensim(net, 1)`.

#### 6.4. Obtención de la trayectoria de posición de referencia

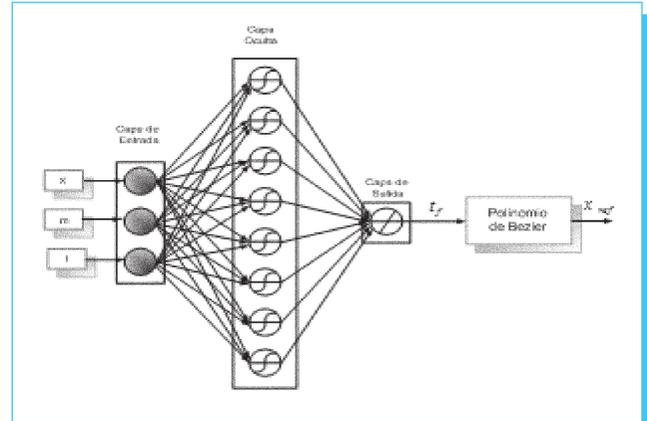
Para obtener la señal de referencia de posición  $x_{ref}$  se toma en cuenta el valor inicial de partida  $y_0$ , el valor final de llegada  $y_f$  así como la ecuación (4) que es el polinomio de Bézier de quinto orden. En esta ecuación se considera que el tiempo inicial  $t_0 = 0$  y el tiempo final de llegada de la grúa viajera  $t_f$  es determinado por la red neuronal artificial entrenada en la sección anterior. Establecidas las condiciones anteriores, se utiliza la ecuación (9) que es una expresión simplificada de la ecuación (4) tomada de la referencia [5]:

$$x_{ref} = y_0 + (y_f - y_0) \left[ 6 \left[ \frac{t}{t_f} \right]^5 - 15 \left[ \frac{t}{t_f} \right]^4 + 10 \left[ \frac{t}{t_f} \right]^3 \right] \quad (9)$$

A partir de esta ecuación se obtiene la señal de referencia de posición (figura 4) y es la señal que utiliza el controlador de posición.

### 7. Resultados de la simulación

Para verificar el desempeño de la red neuronal artificial, la cual se emplea para calcular el tiempo final del polinomio de Bézier, que a su vez se utiliza para obtener la señal de referencia de posición, se consideraron diferentes condiciones de operación de la grúa, tanto para variaciones de masa como de distancia.



**Fig. 4.** Arquitectura de la red neuronal de realimentación.

En cada uno de los casos se graficaron los siguientes valores con respecto a la grúa viajera:

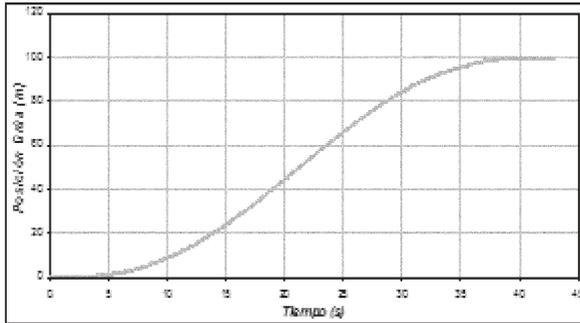
- Posición lineal de referencia,  $x$  en m.
- Velocidad lineal de referencia,  $\dot{x}$  en m/s.
- Error de posición en m.
- Oscilación en la carga,  $\theta$  en rad.

#### 7.1. Operación de la grúa viajera con una masa de 800 kg, distancia de 100 m y longitud de 3 m

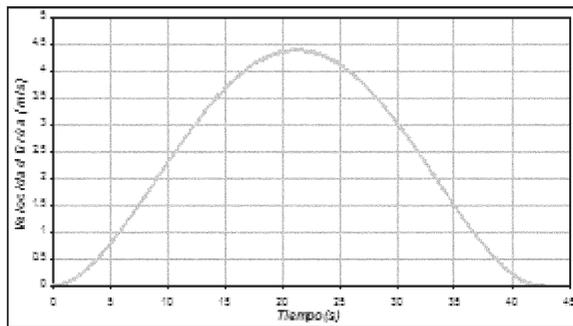
Como se observa en la figura 5a, la posición que describe la grúa es de forma suave, dicha posición no debe tener cambios bruscos ya que la carga puede oscilar. La velocidad con la cual se desplaza la grúa viajera para cubrir la distancia de 100 m en 42.7 s se muestra en la figura 5b. Se observa que la velocidad se va incrementando hasta llegar a su valor máximo de 4.47 m/s en 22 s, y después va disminuyendo hasta llegar a la posición de 100 m con una velocidad nula. El error de la posición se define como la diferencia entre la posición de referencia y la posición real a la cual llega la grúa viajera; en la figura 6, se muestra que el error máximo es de 0.059 m y se presenta a la mitad del desplazamiento. También se muestra que el error de posición final es cero.

La figura 7 muestra la oscilación que presenta la carga al realizarse el desplazamiento. En este sistema, la energía potencial es mínima cuando la carga se encuentra en la parte inferior.

Las oscilaciones de la carga durante el desplazamiento de la grúa se encuentran en los siguientes valores: en  $+0.036$  rad y en  $-0.034$  rad, equivalentes a tener  $2.06^\circ$  y  $-1.95^\circ$  respectivamente.



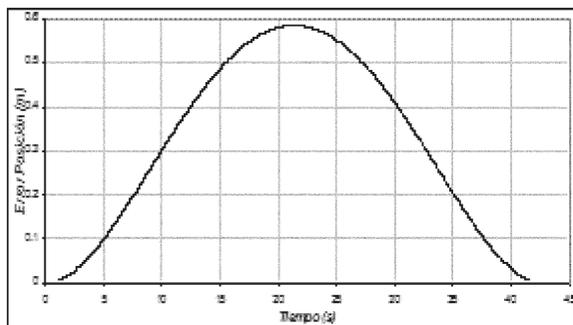
a) Posición lineal de referencia,  $x$  en m.



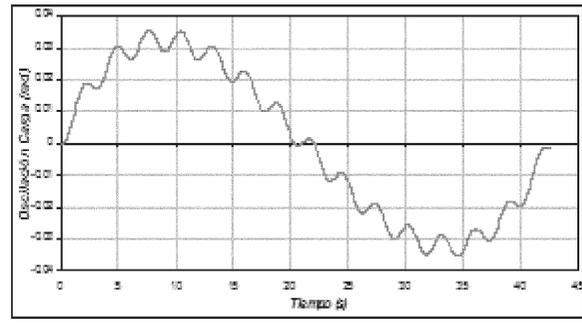
b) Velocidad lineal de referencia,  $x$  en m/s.

**Fig. 5.** Posición y velocidad lineal de la grúa viajera con una masa de 800 kg, distancia de 100 m y longitud de 3 m.

Tomado en cuenta la figura 8 y la ecuación (10), se obtiene para este caso a  $d_1 = 0.071$  m,  $d_2 = 0.068$  m y donde  $d_H = 0.139$  m, concluyendo que la oscilación se mantiene en el rango especificado.



**Fig. 6.** Error de posición en m para una masa de 800 kg, distancia de 100 m y longitud de 3 m.



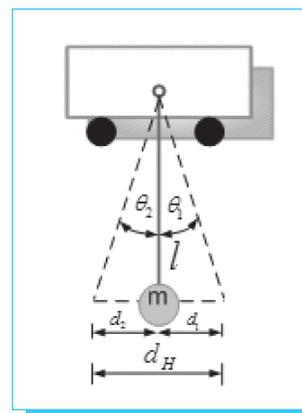
**Fig. 7.** Oscilación de la carga para una masa de 800 kg, distancia de 100 m y longitud de 3 m.

$$\begin{aligned} \tan \theta_1 &= d_1 / l \\ \tan \theta_2 &= d_2 / l \end{aligned} \quad (10)$$

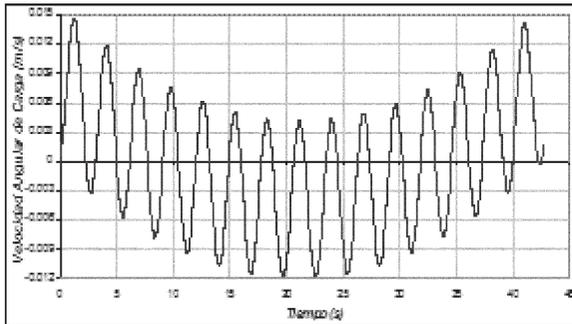
De la respuesta del sistema se observa que cuando la grúa avanza la carga tiende a permanecer en el mismo punto debido a su inercia, presentándose la oscilación hacia el sentido opuesto cuando la grúa frena.

La derivada de la oscilación de la carga se muestra en la figura 9. En ésta se aprecia que dicha velocidad es pequeña.

A continuación se muestra la gráfica de oscilación de la carga que se obtuvo tomando en cuenta las mismas condiciones de distancia y de masa, pero con un tiempo diferente al obtenido por la red neuronal artificial. Como se observa en la figura 7 el



**Fig. 8.** Representación esquemática para calcular el desplazamiento horizontal.



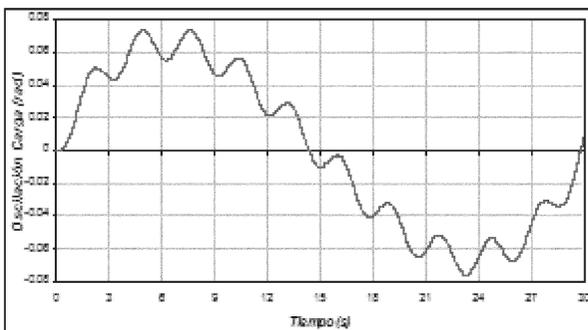
**Fig. 9.** Velocidad angular de la carga para una masa de 800 kg, distancia de 100 m y longitud de 3 m.

desplazamiento horizontal total es de 0.139 m, mientras que el desplazamiento que se obtiene en la figura 10 es de 0.304 m.

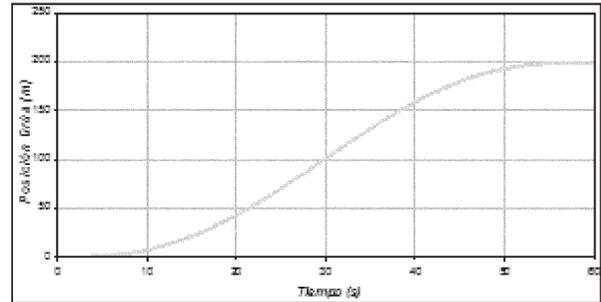
### 7.2. Operación de la grúa viajera con una masa de 1 500 kg, distancia de 200 m y longitud de 3 m

En la figura 11a, se muestra la posición que describe la grúa, la cual es de forma suave para evitar las oscilaciones. La oscilación generada se observa en la figura 11d.

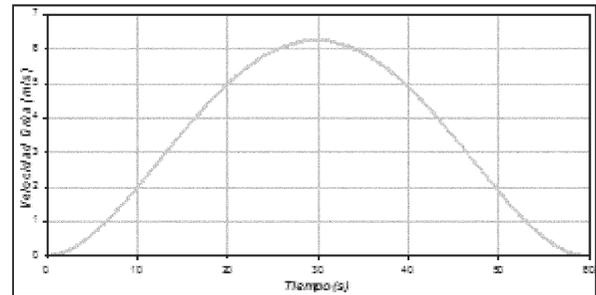
La velocidad con la cual se desplaza la grúa viajera para cubrir la distancia de 200 m en 60 s se muestra en la figura 11(b). En ésta se observa que la velocidad se va incrementando hasta llegar a su valor máximo de 6.37 m/s en 30 s, y después va disminuyendo hasta llegar a la posición de 200 m con una velocidad nula.



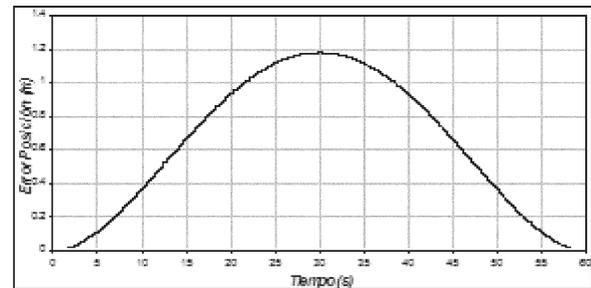
**Fig. 10.** Oscilación de la carga para una masa de 800 kg, distancia de 100 m, longitud de 3 m y tiempo de 30 s.



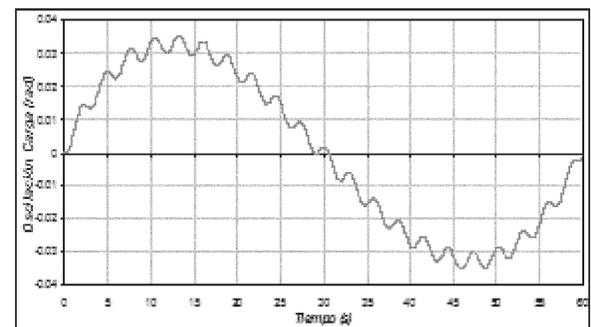
a) Posición lineal de referencia,  $x$  en m.



b) Velocidad lineal de referencia,  $x$  en m/s.



c) Error de posición en m.



d) Oscilación en la carga,  $\theta$  en rad.

**Fig. 11.** Operación de la grúa viajera con una masa de 1500 kg, distancia de 200 m y longitud de 3 m.

En cuanto al error de posición que se presenta en la grúa (figura 11c), se observa que a la mitad del desplazamiento se encuentra

el error máximo de 1.17 m. Asimismo, la grúa llega a su posición final con un error de posición de cero.

## 8. Conclusiones

Con base en los resultados de las simulaciones, se puede observar, que el uso del polinomio de Bézier para la planificación de trayectorias de la grúa viajera es adecuado para realizar el desplazamiento de la grúa y a su vez reduce la oscilación de la carga a valores establecidos.

También se demuestra la eficacia de la red neuronal artificial en la determinación del tiempo final del desplazamiento de la grúa. Esto se debe a que la red neuronal artificial logra asociar y generalizar los datos de entrada de manera satisfactoria, logrando buenos resultados, los cuales pueden ser aplicables en un futuro en maquinaria real.

## 9. Referencias

- [1] H. H. Lee, «A New Motion-Planning Scheme for Overhead Cranes with High-Speed Hoisting», *Journal of Dynamic System, Measurement, and Control*, Vol. 126, 359-364, 2004.
- [2] Gerald Farin, *Curves and Surfaces for Computer Aided Geometric Design, A practical Guide*. Second Edition. Academic Press, Inc. 2001.
- [3] Donald Hearn y M. Pauline Baker, *Gráficos por computadora con OPENGL*, 3ed., Ed. Prentice Hall, México 2006.
- [4] Kenneth I Joy, *Bézier Curves, Visualization and Graphics*, Group University of California, 2000.
- [5] Marco Paluszny, Hartmut Prautzsch y Wolfgang Boehm. *Metodos de Bezier y B-splines*. Springer Verlag Berlin. 2002.
- [6] Martin del Brio, Bonifacio y Sanz Molina, Alfredo *Redes Neuronales y Sistemas Borrosos*, 3ª ed., Ra-Ma Editorial, 2006.
- [7] W. Yu and X. Li, «System Identification using adjustable RBF Neural Network with Stable Learning Algorithms», *Advances in Neural Networks*, ISSN 2004, Springer-Verlang, Lecture Notes in Computer Science, Vol. 3174, 212-217, 2004.
- [8] 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.
- [9] Torres Rodríguez, Ivone C. "Control de un Eje para una Grúa Viajera". Tesis de Maestría en Ingeniería Eléctrica, SEPI-ESIME-IPN, 2006.
- [10] Huerta González, P. F., Rodríguez Rivas, J. J. y Torres Rodríguez, I. C. "Estimación de la resistencia del rotor usando una red neuronal artificial en el control vectorial indirecto del motor de inducción", June 2008, *IEEE Latin America Transactions*, Vol. 6, No. 2.

### Anexo 1. Especificaciones de la grúa viajera.

|                                      |          |
|--------------------------------------|----------|
| Masa máxima de la carga              | 1 500 kg |
| Desplazamiento máximo                | 200 m    |
| Masa de la grúa                      | 500 kg   |
| Longitud máximo del cable (malacate) | 3 m      |
| Radio de las ruedas                  | 10 cm    |
| caja de engrane                      | 3:1      |

### Anexo 2. Especificaciones del motor de inducción.

|                        |                           |                         |
|------------------------|---------------------------|-------------------------|
| 5 HP, 3 $\Phi$         | 220 V, 60 Hz              | 4 polos, 1780 rpm       |
| $r_s = 0.1062 \Omega$  | $X_m = 5.834 \Omega$      | $X_r = 0.2145 \Omega$   |
| $r'_s = 0.0764 \Omega$ | $X'_{ls} = 0.2145 \Omega$ | $J = 2.8 \text{ kgm}^2$ |

# Redalyc

Red de revistas científicas de  
América Latina, el Caribe,  
España y Portugal

[www.redalyc.uaemex.mx](http://www.redalyc.uaemex.mx)