

## SIMULADOR ANIMADO DE GRÚA TORRE PARA APLICACIONES DE INVESTIGACIÓN

José C. Chahuara Quispe, Guillermo Tejada Muñoz

*Facultad de Ingeniería Electrónica, Universidad Nacional Mayor de San Marcos*

**RESUMEN:** Las grúas torre son sistemas físicos no-lineales de tres grados de libertad. Debido a sus características físicas y su complejidad, son de gran interés para los investigadores del área de automatización. Por este motivo, el artículo describe el diseño e implementación de un simulador animado en 3D de grúa torre realizado con herramientas de Matlab, que puede ser operado manualmente o automáticamente por medio de algoritmos de control. El modelo ha sido sometido a prueba controlándola con algoritmos desarrollados en la Universidad de Virginia Tech.

**SUMMARY:** The crane towers are no-linear physical systems of three degrees of freedom. Due to its physical characteristics and complexity, they are interesting for the investigators of the automation area. For this reason, this paper describes the implementation of a simulator animated in 3D of crane tower made with MATLAB tools, which can be operated manually through icons or automatically through control algorithms. The simulator has been put on test when it has been controlled with algorithms developed at Virginia Tech University.

**Palabras Clave:** Grúa torre, modelamiento, Matlab, Simulink, ecuaciones de movimiento de Lagrange, simulación.

### I. INTRODUCCIÓN

La grúa torre es una máquina de gran importancia en la industria de la construcción, ya que es la encargada del transporte de materiales como acero y cemento dentro del área de trabajo. Debido a su uso extendido y al gran interés en mejorar su desempeño, se han realizado un

gran número de estudios acerca de sistemas que permitan el control automático de estas máquinas.<sup>1</sup>

Para poder experimentar con un prototipo virtual de grúa torre real lo más confiable posible, en donde se pueda realizar pruebas con diferentes algoritmos de control, el presente trabajo presenta el fundamento teórico y el proceso de diseño de un simulador de grúa torre 3D confiable, que permita la visualización de su funcionamiento.

El simulador permite un funcionamiento en modo manual y automático. En el modo manual el usuario, a través de una consola de botones, puede realizar el proceso de transporte de la carga tal como lo haría un operador de grúa, se registra la información numérica de las variables involucradas en una base de datos para su posterior análisis y utilización. En el modo automático, el funcionamiento de la grúa depende de algún algoritmo de control que puede ser incluido en el simulador.

La confiabilidad del modelo ha sido comprobada al aplicar sobre él algoritmos de control desarrollados por Hanafy<sup>1</sup> y Al-Mousa,<sup>2</sup> obteniéndose que los resultados generados por el simulador son iguales a los obtenidos por estos investigadores.

### II. DISEÑO DEL SIMULADOR 3D

El proceso de diseño se ha realizado en 4 pasos:

- Modelamiento matemático de la grúa.
- Implementación del modelo en Simulink.
- Animación.
- Diseño de la interfaz gráfica de usuario.

## 2.1 Modelamiento matemático de la grúa

### 2.1.1 Movimientos y coordenadas

Una grúa torre es un sistema de tres grados de libertad que consiste en una torre que soporta en su parte superior una pluma horizontal de movimiento rotacional sobre la cual hay un carro de movimiento radial y un cable, sometido a variaciones de longitud, de movimiento vertical. De esta manera, la carga puede moverse hacia cualquier posición dentro del espacio delimitado por los movimientos de la grúa. Los cambios en los valores de las variables  $\varphi(t)$ ,  $r(t)$  y  $L(t)$  que se muestran en la figura 1, producen oscilaciones en la carga representadas por los ángulos  $\phi(t)$  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  $YZ$  como se muestra en las figuras 2 y 3. El objetivo del operador humano o sistema de automatización, es mover la carga que pende del punto  $Q$  rápidamente y manteniendo pequeños los ángulos  $\theta(t)$  y  $\phi(t)$ .

De lo visto anteriormente, se concluye que el sistema de grúa torre tiene 5 parámetros que determinan la ubicación espacial de la carga suspendida:  $\varphi(t)$ ,  $r(t)$ ,  $L(t)$ ,  $\theta(t)$  y  $\phi(t)$ .

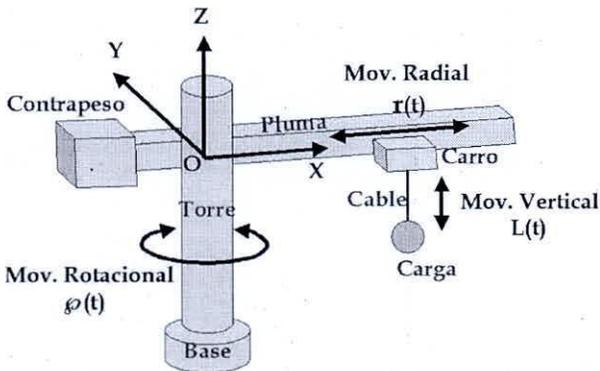


Figura 1. Estructura simplificada de una grúa torre

### 2.1.2 Ecuaciones de movimiento

Para derivar las ecuaciones de movimiento de la grúa torre se sigue el enfoque de las ecuaciones de movimiento de Lagrange, basadas en el principio de conservación de la energía. Para esto, es necesario definir: las coordenadas generalizadas; las fuerzas  $F_x$ ,  $T_\varphi$  y  $F_L$ , responsables de los cambios en los valores de  $\varphi(t)$ ,  $r(t)$ ,  $L(t)$ , y el Lagrangiano.<sup>3</sup> Los resultados se presenten mediante las ecuaciones del 1 al 4.

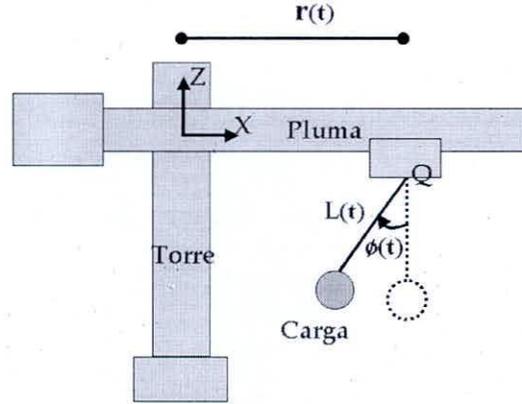


Figura 2. Vista lateral del modelo de grúa torre

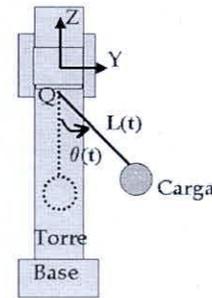


Figura 3. Vista frontal del modelo de grúa torre

$$F_x = mr''(t) - 2mL'(t)\cos\theta(t)\cos\phi(t)\phi'(t) + 2mL(t)\sin\theta(t)\cos\phi(t)\theta'(t)\phi'(t) + mL(t)\cos\theta(t)\sin\phi(t)\phi'^2(t) - mL(t)\cos\theta(t)\cos\phi(t)\phi''(t) + Mr''(t) + 2mL'(t)\sin\phi(t)\sin\theta(t)\theta'(t) + mL(t)\cos\theta(t)\sin\phi(t)\theta'^2(t) - m\varphi'^2(t)r(t) + mL(t)\sin\theta(t)\sin\phi(t)\theta''(t) - mL''(t)\cos\theta(t)\sin\phi(t) - mL(t)\sin\theta(t)\varphi''(t) - M\varphi'^2(t)r(t) - 2mL'(t)\sin\theta(t)\varphi'(t) - 2mL(t)\cos\theta(t)\varphi'(t)\theta'(t) + m\cos\theta(t)\sin\phi(t)L(t)\varphi'^2(t) + b1r'(t) \quad (1)$$

$$0 = L(t)\cos^2\theta(t)\phi''(t) + g\cos\theta(t)\sin\phi(t) - L(t)\cos^2\theta(t)\cos\phi(t)\sin\phi(t)\varphi'^2(t) + \cos\phi(t)\cos\theta(t)r(t)\varphi'^2(t) - \cos\theta(t)\cos\phi(t)r''(t) - 2L(t)\sin\theta(t)\cos\theta(t)\phi'(t)\theta'(t) + 2L'(t)\sin\theta(t)\cos\theta(t)\cos\phi(t)\varphi'(t) + 2L'(t)\cos^2\theta(t)\phi'(t) + L(t)\cos\theta(t)\cos\phi(t)\sin\theta(t)\varphi''(t) + 2L(t)\cos^2\theta(t)\cos\phi(t)\theta'(t)\varphi'(t) \quad (2)$$

$$T_\varphi = -mL(t)\sin\theta(t)r''(t) + 2m\cos\theta(t)L(t)L'(t)\sin\theta(t)\cos\phi(t)\phi'(t) - 2mL^2(t)\sin^2\theta(t)\cos\phi(t)\phi'(t)\theta'(t)$$

$$\begin{aligned}
 & - m \cos\theta(t) \sin\phi(t) L^2(t) \sin\theta(t) \dot{\phi}^2(t) + m \cos\theta(t) \\
 & L^2(t) \sin\theta(t) \cos\phi(t) \ddot{\phi}(t) - \\
 & 2mL(t)L'(t)\sin\phi(t)\dot{\theta}'(t) - mL^2(t)\sin\phi(t)\ddot{\theta}'(t) + \\
 & m\sin^2\theta(t)L^2(t)\ddot{\varphi}''(t) + \\
 & 2m\dot{\varphi}'(t)L(t)L'(t)\sin^2\theta(t) + mL''(t)\sin\theta(t)r(t) + \\
 & 2m\cos\theta(t)L'(t)r(t)\dot{\theta}'(t) - m\sin\theta(t)r(t)L(t)\dot{\theta}^2(t) \\
 & + m\cos\theta(t)r(t)L(t)\ddot{\theta}''(t) + m\dot{\varphi}''(t)r^2(t) + \\
 & 2m\dot{\varphi}'(t)r(t)r'(t) - \\
 & 2m\dot{\varphi}'(t)r'(t)L(t)\cos\theta(t)\sin\phi(t) - \\
 & 2m\dot{\varphi}''(t)r(t)L(t)\cos\theta(t)\sin\phi(t) - \\
 & 2m\dot{\varphi}'(t)r(t)L'(t) \\
 & \cos\theta(t)\sin\phi(t) + \\
 & 2m\dot{\varphi}'(t)r(t)L(t)\sin\theta(t)\sin\phi(t)\dot{\theta}'(t) - \\
 & 2m\dot{\varphi}'(t)r(t)L(t)\cos\theta(t)\cos\phi(t)\dot{\phi}'(t) + \\
 & m\dot{\varphi}''(t)L^2(t)\cos^2\theta(t)\sin^2\phi(t) + \\
 & 2m\dot{\varphi}'(t)L(t)L'(t)\cos^2\theta(t)\sin^2\phi(t) + \\
 & 2Mr(t)r'(t)\dot{\varphi}'(t) + \\
 & Mr^2(t)\ddot{\varphi}''(t) + J_0\ddot{\varphi}''(t) + mL^2(t)\dot{\varphi}'(t) \\
 & \cos^2\phi(t)\sin 2\theta(t)\dot{\theta}'(t) + mL^2(t)\dot{\varphi}'(t)\cos^2\theta(t) \\
 & \sin 2\phi(t)\dot{\phi}'(t) + b_2\dot{\varphi}'(t)
 \end{aligned} \tag{3}$$

$$\begin{aligned}
 0 = & r''(t) \sin\theta(t) \sin\phi(t) + 2L'(t)\dot{\theta}'(t) + L(t)\ddot{\theta}''(t) - \\
 & 2L'(t)\sin\phi(t)\dot{\varphi}'(t) - L(t)\sin\phi(t)\ddot{\varphi}''(t) - \\
 & L(t)\cos\phi(t)\dot{\varphi}'(t)\dot{\phi}'(t) + r(t)\cos\theta(t)\dot{\varphi}''(t) + 2r'(t) \\
 & \cos\theta(t)\dot{\varphi}'(t) + L(t)\cos\theta(t)\sin\theta(t)\dot{\phi}^2(t) + \\
 & L(t)\cos\phi(t)\sin^2\theta(t)\dot{\varphi}'(t)\dot{\phi}'(t) - \\
 & L(t)\cos^2\theta(t)\cos\phi(t)\dot{\varphi}'(t)\dot{\phi}'(t) - L(t) \\
 & \cos\theta(t)\sin\theta(t)\dot{\varphi}^2(t) - r(t)\sin\theta(t)\sin\phi(t) \\
 & \dot{\varphi}^2(t) + L(t)\cos\theta(t)\sin\theta(t)\sin^2\phi(t)\dot{\varphi}^2(t) + g \\
 & \cos\phi(t)\sin\theta(t)
 \end{aligned} \tag{4}$$

En la figura 4, se muestra el sistema de elevación de cargas, que representa un sistema independiente del movimiento del carro y de la pluma. Se incluye el coeficiente de fricción viscosa  $b_3$  y el peso longitudinal del cable  $K_{mas}$ .

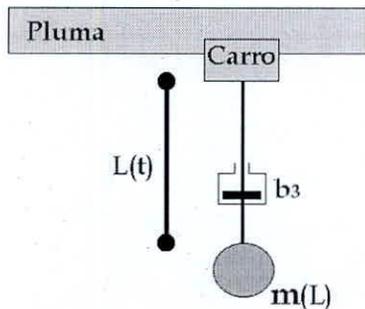


Figura 4. Modelo del sistema de elevación

La aplicación del enfoque de ecuaciones de movimiento de Lagrange produce la ecuación 5 que al igual que las anteriores es de característica no-lineal.

$$F_1 = K_{mas}L'^2(t) + mL''(t) - m(t)g + b_3L'(t) \tag{5}$$

### 2.2 Implementación del modelo en Simulink

El sistema de grúa puede ser representado por el diagrama de la figura 5.



Figura 5. Modelo del sistema de grúa torre

Las fuerzas  $F_x$ ,  $F_1$  y el torque  $T\varphi$  son las entradas del sistema y producen como salida la posición de la carga representada por las variables  $\varphi(t)$ ,  $r(t)$ ,  $L(t)$ ,  $\theta(t)$  y  $\phi(t)$ . La figura 6, muestra el modelo del sistema realizado en la herramienta Simulink de Matlab, que implementa el diagrama de la figura 5. Los bloques que representan una ecuación, generan una señal o permiten la visualización de respuestas, están unidos por flechas que indican el flujo de la información.

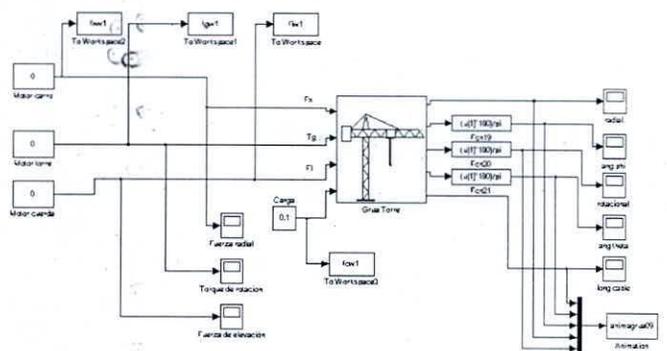


Figura 6. Implementación del modelo en Simulink

Los parámetros especificados para realizar la simulación fueron los siguientes:

- Método de integración ODE 45 (recomendado para sistemas pendulares).
- Paso inicial, auto.
- Tamaño del paso máximo: 0.1.
- Tolerancia o error relativo: 1e-6.

2.3 Animación

A través de la animación de la grúa torre es posible visualizar los movimientos del sistema dentro de un entorno virtual; la animación consiste de un conjunto de objetos descritos como formas sencillas (rectángulos, círculos, etc.), una colección de datos de los puntos que constituyen una imagen, y un conjunto de instrucciones computacionales que describen los cambios de cada uno de los objetos en el tiempo; se pueden incluir cambios en la locación, orientación, geometría, o incluso el color. Los elementos geométricos básicos de una grúa torre son:

- La torre.
- La pluma.
- El contrapeso.
- El carro.
- El cable.

Estos parámetros fueron determinados después de un proceso iterativo de ajuste hasta obtener una ejecución fluida de la simulación.

Para poder simular el proceso de transporte de cargas, se ha agregado una carga y una marca que señale el punto de destino. En las figuras numeradas del 7 al 11, se presentan respectivamente las vistas (a) frontal, (b) lateral y (c) desde un plano superior, para los casos de: la torre; la pluma y el contrapeso; el carro y la cuerda; la carga y el señalador de destino de transporte de la carga de la grúa.

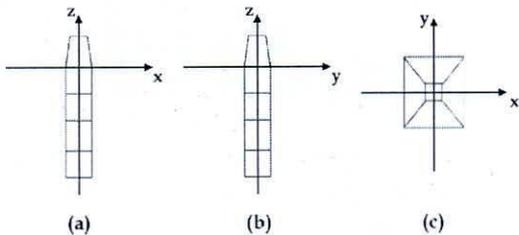


Figura 7 – Vistas de la torre

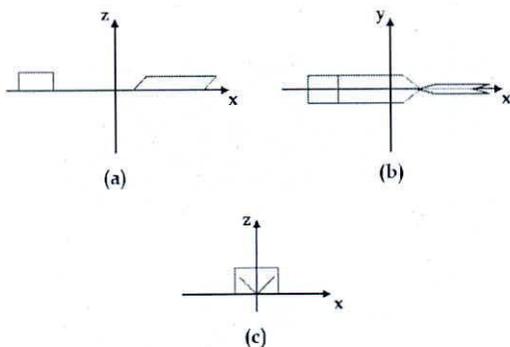


Figura 8. Vistas de la pluma y el contrapeso

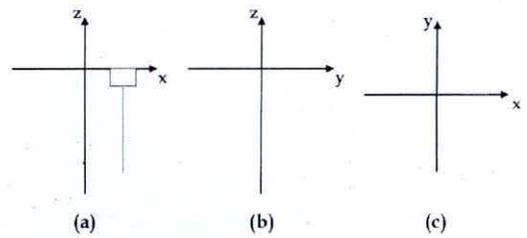


Figura 9. Vistas del carro y de la cuerda

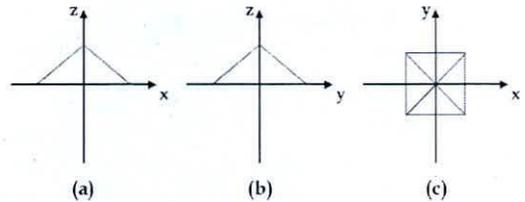


Figura 10. Vistas de la carga.

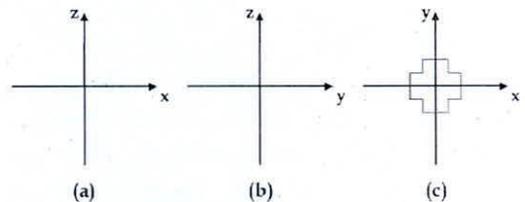


Figura 11. Vistas del señalador de destino

Cada imagen de la animación de la grúa torre contiene todos los elementos descritos líneas arriba como se muestra en la figura 12, el punto de inicio y final para el transporte de la carga son escogidos por el usuario a través del ingreso de las coordenadas de inicio y final de transporte de la carga correspondientes en el código del programa.

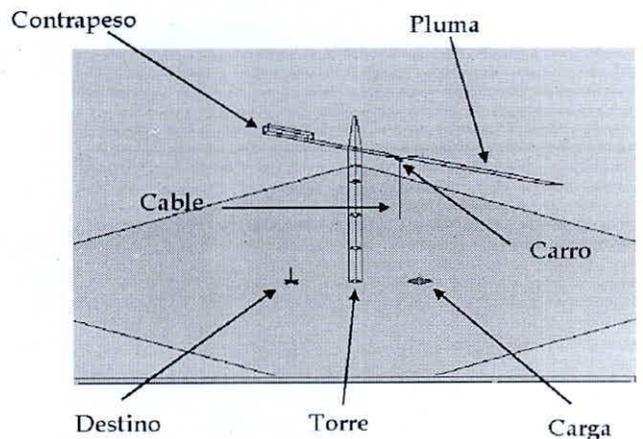


Figura 12. Animación del simulador 3D

Para dibujar cada una de las figuras que conforman una imagen en Matlab, se necesita que los datos de la ubicación de los puntos de cada figura geométrica estén agrupados en matrices. Por ejemplo, si se desea dibujar el cuadrado de la figura 13, se debe considerar los vértices del mismo como los datos que determinan su ubicación. Con los datos se forman dos vectores con sus elementos ordenados de la misma forma como si se dibujara el cuadrado utilizando un lápiz.

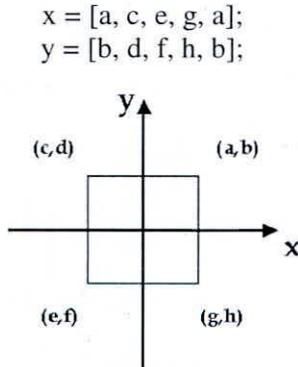


Figura 13. Dibujo geométrico en Matlab.

El cuadrado es dibujado cuando se ingresa la siguiente orden:

```
cuadrado = line(x,y,'color',[0 0 0]);
```

la orden "line" indica que se debe trazar una línea que pase por los puntos (x,y) y que esta línea sea de color negro ([0, 0, 0]).

En el caso de la torre de la grúa, en lugar de literales como a,b,c, etc., se utilizaron las siguientes variables para describir sus dimensiones: grua.torre.ancho, grua.torre.fondo, grua.torre.alto. Se debe tener en cuenta que la torre es una reunión de formas geométricas en sí misma, por lo que debe ser dibujada por partes, utilizando los descriptores mencionados. Así para la parte inferior de la torre se tiene:

```
x = [-grua.torre.fondo/2 -grua.torre.fondo/2
      grua.torre.fondo/2 grua.torre.fondo/2 -
      grua.torre.fondo/2];
y = [-grua.torre.ancho/2 grua.torre.ancho/2
      grua.torre.ancho/2 -grua.torre.ancho/2 -
      grua.torre.ancho/2];
z = [-0.8*grua.torre.alto -0.8*grua.torre.alto -
      0.8*grua.torre.alto -0.8*grua.torre.alto -
      0.8*grua.torre.alto];
hti = line(x,y,z,'color',[0 0 0],'erasemode','none');
hold on;
```

Para la parte superior de la torre:

```
hts = line(x/4,y/4,z+grua.torre.alto,'color',[0 0
0],'erasemode','none');
```

Para el lado lateral izquierdo de la torre:

```
x = [-grua.torre.fondo/2 grua.torre.fondo/2
      grua.torre.fondo/2 grua.torre.fondo/8 -
      grua.torre.fondo/8];
y = [-grua.torre.ancho/2 -grua.torre.ancho/2 -
      grua.torre.ancho/2 -grua.torre.ancho/8 -
      grua.torre.ancho/8];
z = [-0.8*grua.torre.alto -0.8*grua.torre.alto 0 -
      0.8*grua.torre.alto+grua.torre.alto -
      0.8*grua.torre.alto+grua.torre.alto];
htli = line(x,y,z,'color',[0 0 0],'erasemode','none');
```

Para el lado lateral derecho de la torre:

```
htld = line(-x,-y,z,'color',[0 0 0],'erasemode','none');
```

Para la parte frontal de la torre:

```
x = [-grua.torre.fondo/2 -grua.torre.fondo/2 -
      grua.torre.fondo/2 -grua.torre.fondo/8 -
      grua.torre.fondo/8];
y = [grua.torre.ancho/2 -grua.torre.ancho/2 -
      grua.torre.ancho/2 -grua.torre.ancho/8
      grua.torre.ancho/8];
z = [-0.8*grua.torre.alto -0.8*grua.torre.alto 0 -
      0.8*grua.torre.alto+grua.torre.alto -
      0.8*grua.torre.alto+grua.torre.alto];
htf = line(x,y,z,'color',[0 0 0],'erasemode','none');
```

Para la parte trasera:

```
htt = line(-x,-y,z,'color',[0 0 0],'erasemode','none');
```

Las variables hti, hts, htli, htld, htf y htt son los "handlers" de la torre y determinan su posición en cada instante de tiempo. Se repite el mismo procedimiento con cada uno de los elementos de la grúa torre. Así, toda la información acerca de la posición de la grúa torre se encuentra contenida en estructuras matriciales de puntos. Si se desea modificar la posición de los elementos de la grúa torre, se deben actualizar los datos almacenados en los "handlers" de cada uno de los elementos que conforman la grúa.

## 2.4 Interfaz gráfica de usuario

El GUI (Graphical User Interface) o interfaz gráfica de usuario, se muestra en la figura 14, contiene los contro-

les que permiten la operación manual o automática del simulador.

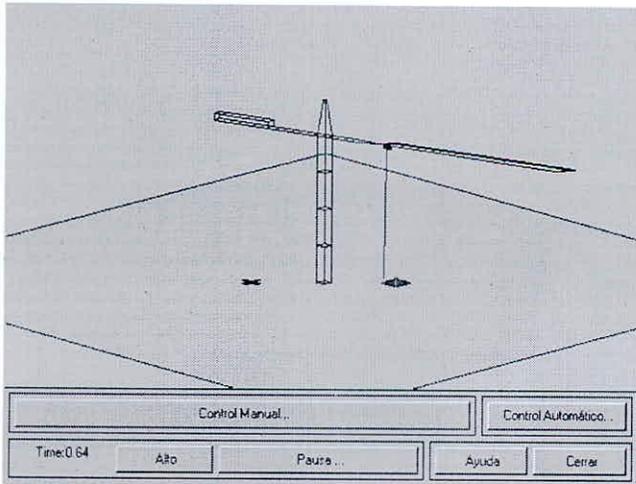


Figura 14. Interfaz Gráfica de usuario del simulador

En el modo de funcionamiento automático la grúa es manejada con un programa que contiene algoritmos de control; mientras que a través del modo manual, el usuario puede operar el simulador presionando los botones, mostrados en la figura 15, que permiten realizar los movimientos del carro, la torre, el cable y enganche de la carga. De acuerdo a las diferentes posiciones de la grúa, las magnitudes que toman las variables son almacenadas en una base de datos. El GUI permite controlar la simulación a través de los botones de inicio, alto y pausa de la simulación. La Figura 16, muestra los botones que permiten que la simulación sea realizada por pasos (Pasos) o continuamente (Continuar).

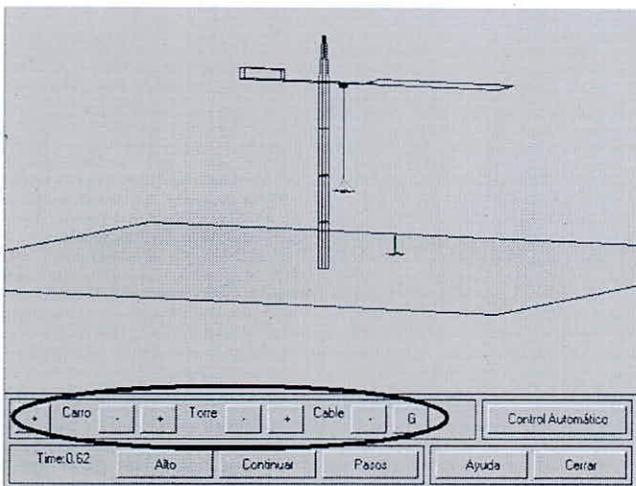


Figura 15. Botones que permiten la operación manual

La figura 17, muestra el menú que permite seleccionar la perspectiva visual de la simulación.

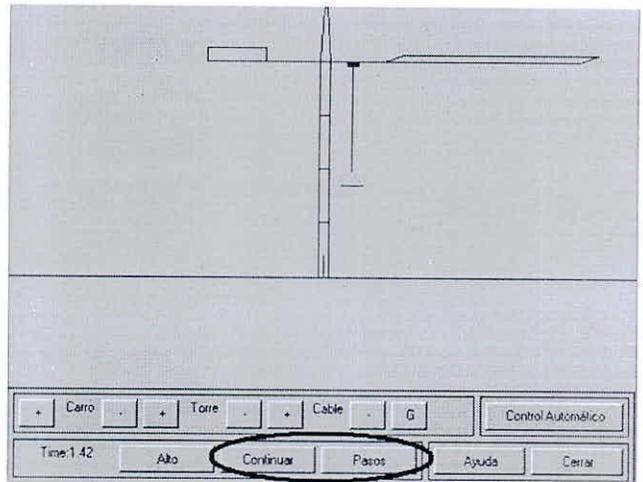


Figura 16. Botones de pausa de la simulación

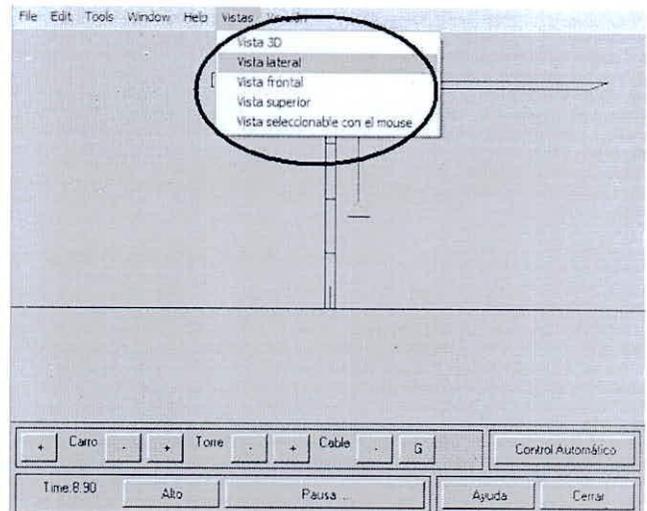


Figura 17. Menú de selección de perspectiva visual

### III. VALIDACIÓN DEL SIMULADOR

Para probar la confiabilidad del simulador se ha controlado su funcionamiento con los algoritmos que los investigadores de la universidad Virginia Tech, Omar Hanafy y Amjed Al-Mousa, desarrollaron en sus tesis. La validez del modelo quedó demostrada al obtener resultados similares.

Como se puede apreciar en la figura 18, las curvas resultantes de la aplicación del controlador de realimentación parcial de estados de Hanafy sobre el simulador 3D (derecha) son iguales a las que publicó este investigador en su tesis (izquierda).

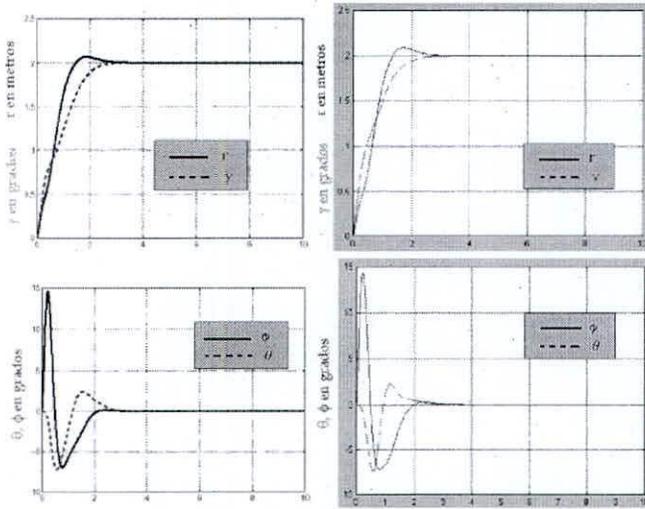


Figura 18. Comparación con los algoritmos de realimentación parcial de Hanafy.

La figura 19, presenta algunas de las curvas obtenidas cuando se aplican los controladores PD-Delay (H DELA) y PD-difuso de Hanafy sobre el simulador 3D (derecha) y aquellas que fueron publicadas en el volumen de tesis de este investigador (izquierda).

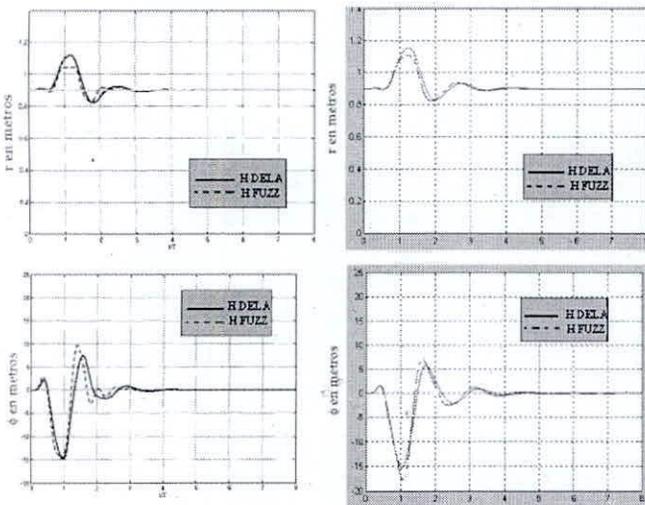


Figura 19. Comparación con los algoritmos dual de Hanafy.

La figura 20, presenta los resultados comparativos al aplicar los controladores de realimentación retrasada en el tiempo (A DELA) y difuso (A FUZZ) diseñados por Al-Mousa sobre el simulador 3D (derecha), y las que publicó en su volumen de tesis (izquierda).

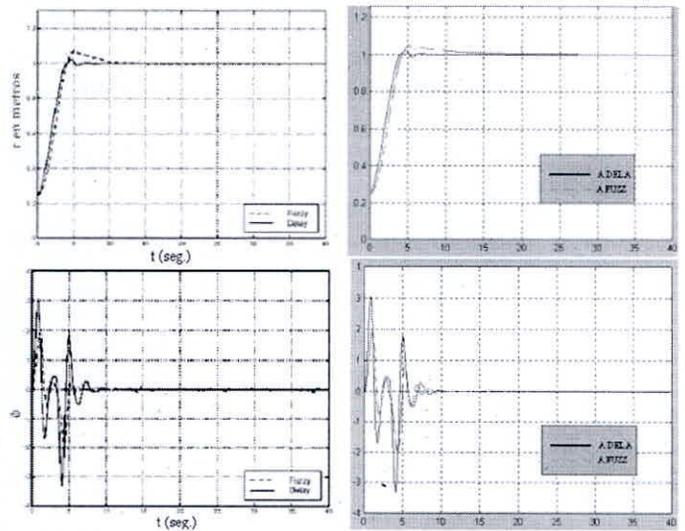


Figura 20. Comparación con los algoritmos de Al-Mousa.

#### IV. CONCLUSIONES

La falta de prototipos reales de grúa para la realización de investigaciones en el área de control automático, se puede superar utilizando herramientas software que permitan la implementación de un modelo matemático para lograr la visualización de su desempeño dentro de un entorno virtual ya sea funcionando manualmente por medio de botones de control o automáticamente desde un bloque que contenga algoritmos de control. La validez del simulador es necesaria si se requiere resultados confiables.

Al aplicar los algoritmos de Hanafy y Al-Mousa para controlar el simulador los resultados son casi exactamente iguales a los publicados por estos investigadores en sus tesis (figuras 18, 19 y 20), las pequeñas diferencias se deben a la falta de información acerca de los parámetros de simulación que se utilizaron en esos trabajos de investigación (métodos de integración, tolerancias, etc.); de estos resultados se concluye que el simulador es confiable.

#### REFERENCIAS

1. O. Hanafy (2003). Control of Gantry and Tower Cranes. Phd Thesis. Department of Mechanics Engineering, Virginia Tech – Virginia.
2. A. Al-Mousa, A. (2000). Control of Rotary Cranes Using Fuzzy Logic and Time Delayed Position Feedback Control. M.Sc. Thesis. Department of Electrical Engineering, Virginia Tech – Virginia.

3. K. Ogata, K. (1998). Dinámica de Sistemas. Prentice Hall.
4. R. Ghigliaza, P. Holmes (2002). On the dynamics of cranes, or spherical pendula with moving supports. International Journal of Non-Linear Mechanics. <http://www.elsevier.com/locate/ijnonlinmec>
5. M. Alonso, E. Finn (1986). Física Volumen I. Addison-Wesley Iberoamericana
6. E. Larradé, A. Miravete (1996). Grúas. Publidisa.