Diseño y construccion de un puente grua*

Nicolas Pachon

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License †

DISEÑO Y CONSTRUCCIÓN DE UN PUENTE GRUA.

ALIRIO BAUTISTA VILLAMIL, DIEGO BERMUDEZ ROMERO, JAIRO CUELLAR SABOGAL, CRISTIAN LOZANO VILLA, JUAN PABLO MOYANO QUIÑONES, NICOLAS PACHON BERMUDEZ, CAMILO RAMIREZ RINCON, RAMON ROMERO PINEDA, DANIEL URIBE POLANIA, CAMILO ZAPATA OCAMPO

1

ABSTRACT: Bridge cranes comprise another important class of cranes in which the pulley system is suspended from a trolley that moves on tracks along one or two horizontal beams, called the bridge, that are supported at both ends. In most cases, the bridge itself can move along a pair of parallel rails, so that the crane can serve a large rectangular area. A circular space can be served by a rotary bridge crane, in which one end of the overhead beam is supported by a central pivot while the other end moves on a circular rail on the periphery of the area.The overhead traveling crane, a bridge crane for which the rails are mounted above the level of the ground or floor, has the advantage of causing no obstruction of the working area. Overhead traveling cranes are commonly used indoors, where their rails can be attached to the columns that support the roof. If the construction of overhead rails is impracticable, the ends of the bridge can be attached to upright towers that move on rails at the ground level; such cranes are called gantry, or goliath, cranes.

PALABRAS CLAVE: Puente grúa, rhinoceros, motor, visual basic, transistor, relé.

1.1 INTRODUCCIÓN

A nivel industrial se utiliza ampliamente el puente grúa, el cual es un equipo que ofrece levantar y trasladar cualquier material o equipo con gran facilidad. Es un tipo de aparato de elevación compuesto por una viga, simple o doble, biapoyada sobre dos carriles elevados sobre unos postes, dispuestos sobre una estructura resistente. El movimiento longitudinal se lleva a cabo mediante la translación de la viga principal o puente a través de los carriles elevados. En la práctica totalidad de los casos, la rodadura es por ruedas metálicas sobre carriles también metálicos.

El movimiento transversal se realiza mediante el desplazamiento de un polipasto o carro sobre uno o dos carriles dispuestos sobre la viga principal.

El movimiento vertical se ejecuta a través del mecanismo de elevación: polipasto o carro.

*Version 1.1: Oct 24, 2008 11:04 pm -0500

†http://creativecommons.org/licenses/by/2.0/

1.2 OBJETIVO

Diseñar e implementar un puente grúa, con interfaz a computador para su control, que sea capaz de mover un objeto dentro de un área determinada.

1.3 DISEÑO

1.3.1 3.1 DISEÑO ESTRUCTURAL

El diseño del puente grúa se realizo con ayuda del software RHINOCEROS y Auto CAD.

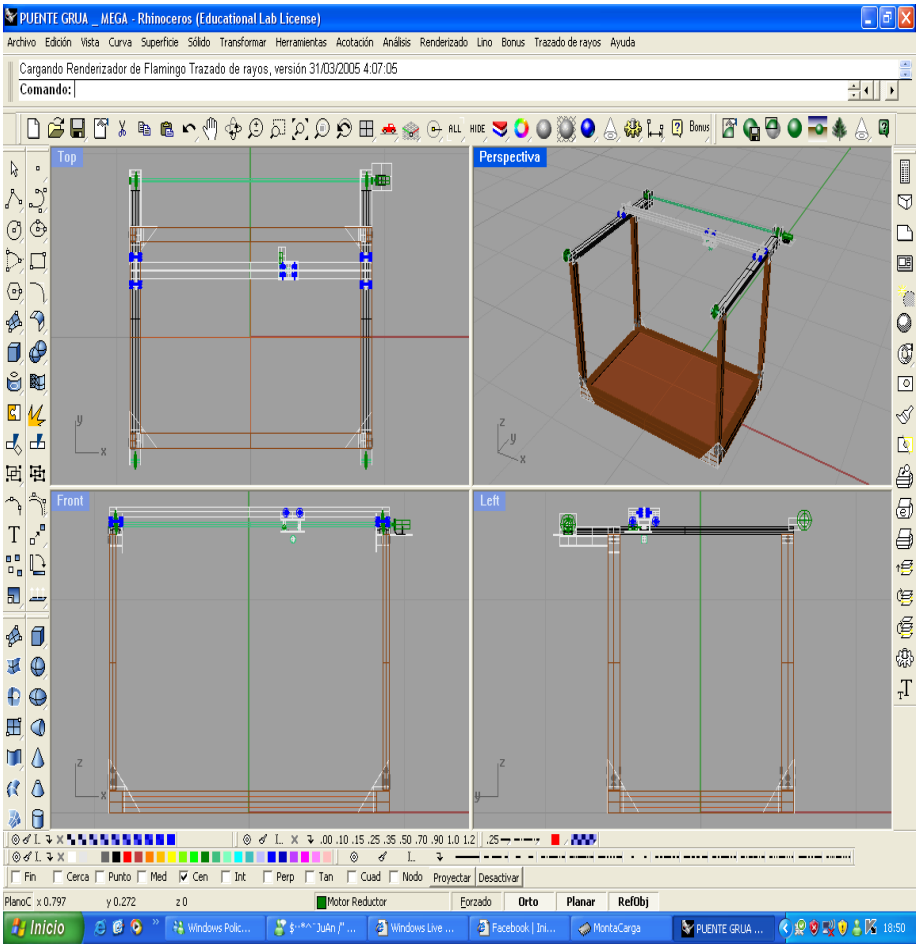

Figure 1

El diseño consta de una estructura rectangular la cual soporta la carga del brazo. El puente grúa tiene un movimiento lineal en el plano. El sistema consta de dos móviles que agarran la carga para poder transportarla y de un gancho el cual la levanta.

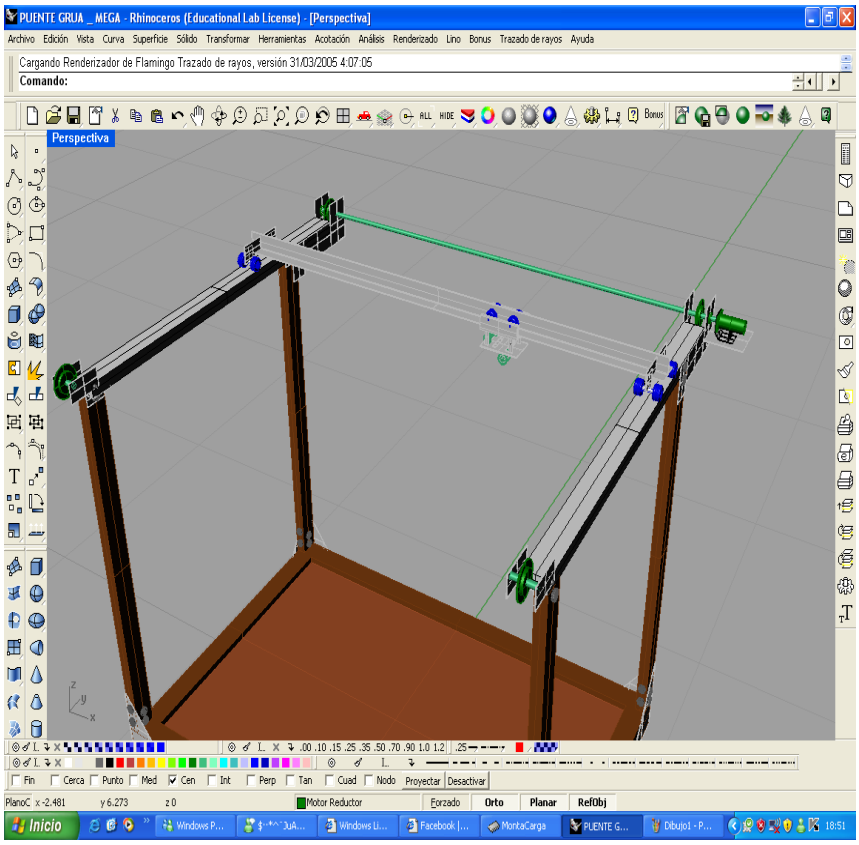

Figure 2

Para el movimiento lineal se utilizaron motores de corriente continua que fueron ensamblados sobre un riel que permite el movimiento del gancho. El "carro" que soporta y transporta la carga consta de dos motores, uno para levantar la carga y otro para el movimiento "en las dimensiones X y Z".

ESFUERZOS ADMISIBLES(MPa)

GRUPO	Fb	Ft	Fc	Fp	Fv
A	21	14.5	14.5	4	1.5
B	15	10.5	11	2.8	1.2
C	10	7.5	8	1.5	0.8

Figure 3

MODULOS DE ELASTICIDAD LONGITUDINAL (MPa)

GRUPO	Emín	Eprom
A	9500	13000
B	7500	10000
C	5500	9000

Figure 4

1.4 3.2 DISEÑO ELECTRONICO

Una etapa de potencia se utiliza para acoplar dispositivos que utilicen altas corrientes eléctricas, con circuitos de bajas corrientes. En el proyecto, se utilizan para accionar los relés que activan y cambian la dirección de giro de los motores.

El diodo que está en paralelo con el relé y en polaridad opuesta, comúnmente llamado "diodo volante", se utiliza para cancelar la contracorriente que se produce en la bobina del relé, para así proteger el transistor.

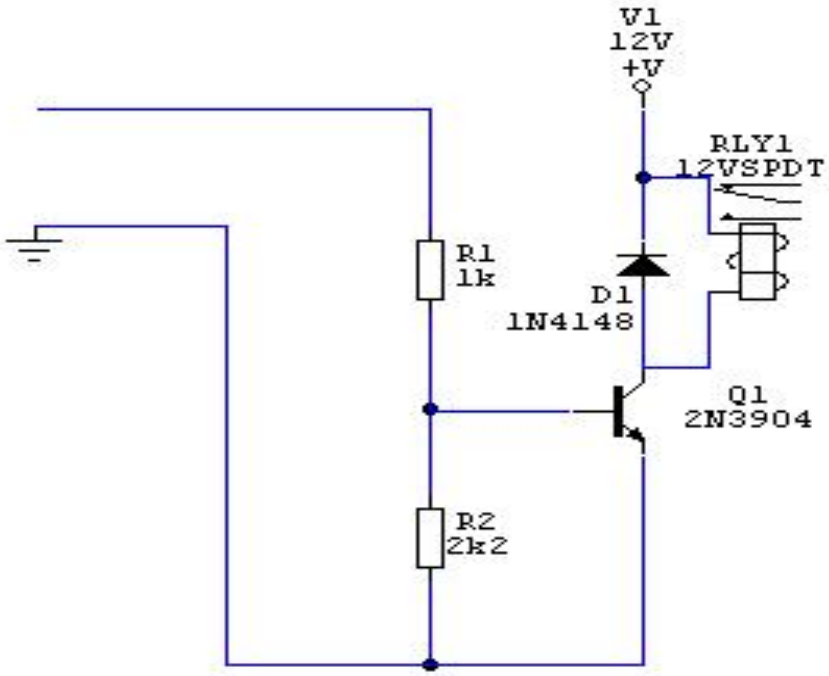

Figure 5

La resistencia interna del relé es de aproximadamente 300 Ω. Por la ley de Ohm, se puede conocer el valor de la corriente que pasa a través de él, que viene a ser la corriente de colector (

$$I_C$$

Figure 6

) del transistor:

$$V = I \times R$$

Figure 7

$$I = \frac{V}{R} = \frac{12v}{300\Omega} = 0.04\ A = 40\ mA$$

Figure 8

Como la corriente del colector del transistor es 40 mA, y su ganancia (hFE) es de 200, se puede saber la corriente de base:

$$I_C = h_{FE} \times I_B$$

Figure 9

$$I_B = \frac{I_C}{h_{FE}} = \frac{40\ mA}{200} = 0{,}5\ mA$$

Figure 10

Sabiendo que el voltaje de la base del transistor corresponde a la diferencia entre un uno (1) lógico de estándar TTL (5v) y el voltaje entre la base y el emisor, se puede conocer el valor de la resistencia para proteger el transistor, por medio de la ley de Ohm:

$$V = I \times R$$

Figure 11

$$R = \frac{V}{I} = \frac{5v}{0{,}5mA} = 10\ K\Omega$$

Figure 12

Se utilizó el circuito integrado 74HC245, es un transceptor octal con búferes no inversores de salida 3-State. Estos búferes se diferencian de los comunes y corrientes, porque tienen un terminal adicional llamado Enable, el cual puede desconectar la compuerta de su salida, encontrándose en un estado de alta impedancia eléctrica, conocido como High-Z.

Como los búferes no son inversores, la entrada es igual a la salida si el Enable está en 1

ENTRADA	ENABLE	SALIDA
0	1	0
1	1	1
X	0	HI-Z

Table 1

X= no importa el estado que tome esta entrada

Debido a que las salidas del puerto paralelo no expulsan la suficiente corriente para activar la bobina del relé, es necesario elaborar una entrada de potencia el cual consta de un transistor NPN de baja ganancia, un diodo que sirve para eliminar el ruido proveniente del campo eléctrico magnético de la bobina, la resistencia sirve para proteger el transistor y polarizarlo.

1.5 SOFTWARE

En la parte que se relaciona con el software, el lenguaje de programación utilizado para la realización del programa fue Visual Basic, en el cual, mediante un archivo *.dll, que permite el control del puerto paralelo LPT1 o DB25, se realiza un modulo utilizando el mismo archivo, el cual define los valores de programación, que deben ser utilizados para su correcto funcionamiento.

Después se crea un formulario, en el cual ubicamos los componentes necesarios, en este caso botones principalmente, junto con elementos de verificación visual, garantizando una apariencia agradable.

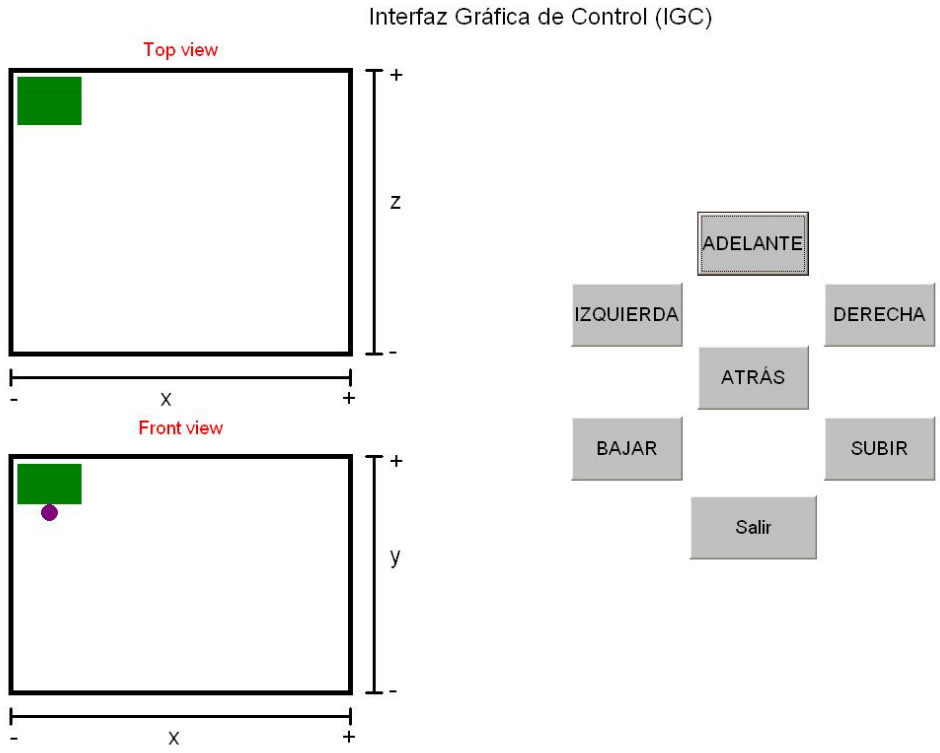

Figure 13

1.6 REFERENCIAS

[1] Creus Sole, Antonio, "Instrumentación Industrial"; Marcombo, 5 edición, España, 1993.

[2] H. Khalil, "Nonlinear Systems", 2nd. ed., Prentice Hall, NJ, pp. 50-56, 1996.

[3] Francis. B. A. and W. M. Wonham, "The internal model principle of control theory", Automatica. Vol. 12. pp. 457-465. 1976.

[4] Gómez M., Luis Gerardo; Estructuras metálicas, puente-grua; Pontificia Universidad Javeriana. Facultad de Ingeniería Civil; 1981

[5] Dorf, Richard Carl 1933; Circuitos eléctricos; Alfaomega Grupo Editor; 2006

[6] Margarit, Joan 1938-; Cálculo de estructuras; Visor Libros; 2005

[7] Mosquera Feijóo, Juan Carlos; Curso práctico de resistencia de materiales; Colegio de Ingenieros de Caminos, Canales y Puertos; 2000

[7] Reyes Fonseca, José Orlando; Pautas para la presentación

escrita de trabajos según las normas Icontec; Pontificia
Universidad Javeriana; 2002.

ANEXO CÓDIGO FUENTE

A continuación, se enuncia el código fuente del programa implementado para el control del puente grúa por computador. Este programa fue diseñado en Visual Basic for Applications 6.0. El programa controla la dirección de movimiento de los motores, y muestra su ubicación en tiempo real.

```
Public Out_Port as Integer
Private Sub Command1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
SetPortBit Out_Port, 6 ;
Timer4.Enabled = True ;
End Sub
Private Sub Command1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
ClrPortBit Out_Port, 6 ;
Timer4.Enabled = False
End Sub
Private Sub Command3_Click()
If MsgBox(" ¿¿¿¿Realmente Desea Salir????", 4) = vbYes Then ;
PortOut Out_Port, 0 ;
End ;
End If
End Sub
Private Sub Command4_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
SetPortBit Out_Port, 5 ;
Timer3.Enabled = True ;
End Sub
Private Sub Command4_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
ClrPortBit Out_Port, 5 ;
Timer3.Enabled = False ;
End Sub
Private Sub Command5_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
SetPortBit Out_Port, 2 ;
Timer2.Enabled = True ;
End Sub
Private Sub Command5_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
ClrPortBit Out_Port, 2 ;
Timer2.Enabled = False ;
End Sub
Private Sub Command6_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
SetPortBit Out_Port, 1 ;
Timer1.Enabled = True ;
End Sub
Private Sub Command6_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
ClrPortBit Out_Port, 1 ;
Timer1.Enabled = False ;
End Sub
Private Sub Command7_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
SetPortBit Out_Port, 4 ;
Timer6.Enabled = True ;
End Sub
Private Sub Command7_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
ClrPortBit Out_Port, 4 ;
```

```
Timer6.Enabled = False ;
End Sub
Private Sub Command8_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
SetPortBit Out_Port, 3 ;
Timer5.Enabled = True ;
End Sub
Private Sub Command8_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
ClrPortBit Out_Port, 3 ;
Timer5.Enabled = False ;
End Sub
Private Sub Form_Load()
Out_Port = &H378
Open "c:\puentegrua\position1l.dll" For Input As #1
Input #1, linea$
Shape3.Left = linea$
Close (1)
Open "c:\puentegrua\position1t.dll" For Input As #1
Input #1, linea$
Shape3.Top = linea$
Close (1)
Open "c:\puentegrua\position2l.dll" For Input As #1
Input #1, linea$
Shape4.Left = linea$
Close (1)
Open "c:\puentegrua\position3l.dll" For Input As #1
Input #1, linea$
Shape5.Left = linea$
Close (1)
Open "c:\puentegrua\position3t.dll" For Input As #1
Input #1, linea$
Shape5.Top = linea$
Close (1)
End Sub
Private Sub Timer1_Timer()
Shape3.Left = Shape3.Left + 6
Shape4.Left = Shape4.Left + 6
Shape5.Left = Shape5.Left + 6
Open "c:\puentegrua\position1l.dll" For Output As #1
Print #1, Shape3.Left
Close (1)
Open "c:\puentegrua\position2l.dll" For Output As #1
Print #1, Shape4.Left
Close (1)
Open "c:\puentegrua\position3l.dll" For Output As #1
Print #1, Shape5.Left
Close (1)
End Sub
Private Sub Timer2_Timer()
Shape3.Left = Shape3.Left - 6
Shape4.Left = Shape4.Left - 6
Shape5.Left = Shape5.Left - 6
```

```
Open "c:\puentegrua\position1l.dll" For Output As #1
Print #1, Shape3.Left Close (1)
Open "c:\puentegrua\position2l.dll" For Output As #1
Print #1, Shape4.Left
Close (1)
Open "c:\puentegrua\position3l.dll" For Output As #1
Print #1, Shape5.Left
Close (1)
End Sub
Private Sub Timer3_Timer()
Shape3.Top = Shape3.Top + 6
Open "c:\puentegrua\position1t.dll" For Output As #1
Print #1, Shape3.Top
Close (1)
End Sub
Private Sub Timer4_Timer()
Shape3.Top = Shape3.Top - 6
Open "c:\puentegrua\position1t.dll" For Output As #1
Print #1, Shape3.Top
Close (1)
End Sub
Private Sub Timer5_Timer()
Shape5.Top = Shape5.Top + 6
Open "c:\puentegrua\position3t.dll" For Output As #1
Print #1, Shape5.Top
Close (1)
End Sub
Private Sub Timer6_Timer()
Shape5.Top = Shape5.Top - 6
Open "c:\puentegrua\position3t.dll" For Output As #1
Print #1, Shape5.Top
Close (1)
End Sub
```