

ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

ESTUDIO DE ALTERNATIVAS DE CONTROL Y SIMULACION
DINAMICA APLICADA AL SISTEMA DEL PENDULO
INVERTIDO MEDIANTE UN PROGRAMA PARA WINDOWS

TESIS PREVIA A LA OBTENCION DEL TITULO DE INGENIERO EN
ELECTRONICA Y CONTROL

LIGIA ELIZABETH GARCES ALBAN

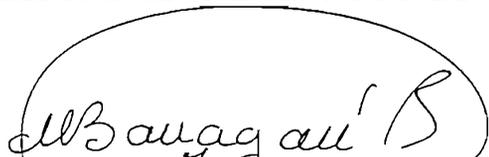
NELSON IVAN GARRIDO DUTARI

Quito, Enero de 1998

CERTIFICACION

CERTIFICO QUE EL PRESENTE TRABAJO FUE
DESARROLLADO INTEGRAMENTE POR LOS
SEÑORES:

LIGIA ELIZABETH GARCES ALBAN
NELSON IVAN GARRIDO DUTARI


INGENIERO M^{ARCO} BARRAGAN
DIRECTOR DE TESIS

DEDICATORIA

A NUESTROS PADRES POR SU AMOR Y DEDICACION.
A NUESTROS HERMANOS.

AGRADECIMIENTOS

AL INGENIERO MARCO BARRAGAN POR SU
ACERTADA DIRECCION EN ESTE TRABAJO.

INDICE DE CONTENIDOS

CAPITULO I: INTRODUCCION

1.1.- Generalidades.....	1
1.2.- Importancia del Tema	2
1.3.- Contenido.....	3

CAPITULO II: ANALISIS MATEMATICO

2.1.- Descripción del Sistema del Péndulo Invertido	5
2.2.- Obtención de los Modelos Matemáticos.....	6
2.3.- Posibles Alternativas de Control.....	17

CAPITULO III: COMPENSACIONES

3.1.- Compensación por Redes y Análisis de Resultados	33
3.2.- Acciones de Control y Análisis de Resultados.	64
3.3.- Realimentación de Estado y Análisis de Resultados	92
3.4.- Control Fuzzy y Análisis de Resultados.	115
3.5.-Comparación Inicial de las Técnicas de Control Aplicadas.....	146

CAPITULO IV: DESARROLLO DEL PROGRAMA

4.1.- Simulación Dinámica.....	149
4.2.- Algoritmos para el Cálculo del Péndulo Invertido.....	151
4.2.1.- Diagramas de Flujo de la Simulación Dinámica.....	155
4.2.2.- Diagramas de Flujo de las Alternativas de Control	159

CAPITULO V: RESULTADOS Y CONCLUSIONES

5.1.- Resultados de la Simulación Dinámica.....	179
5.2.-Comparación de los Resultados Analíticos.....	181
5.3.- Conclusiones.....	184

BIBLIOGRAFIA.....	186
--------------------------	------------

ANEXOS

A.-Nomenclatura.....	189
B.-Manual de Usuario.....	193
C.-Listado del Programa.....	198

CAPITULO I: INTRODUCCION

1.1.- Generalidades.

Gracias al desarrollo de la computación y la informática, el computador se ha convertido en un instrumento indispensable en las diferentes actividades del quehacer humano, y así ha permitido un desarrollo más rápido y eficaz en el campo de la ingeniería.

De los diferentes Sistemas Reales que tienen inestabilidad inherente, se ha escogido el problema del Péndulo Invertido, sobre el cual se ha realizado una simulación elaborada en el lenguaje Visual Basic para Windows en 16-bit, y así poder cuantificar todas las variables asociadas con el problema del Péndulo Invertido y poder comparar las técnicas de control clásicas (redes, acciones de control), modernas (realimentación de estado) y control Fuzzy dando al estudiante una alternativa diferente sobre la estabilización de sistemas.

Para ejecutar el programa desarrollado es necesario que el computador tenga instalado el entorno Windows (recomendable 8 Mby de memoria RAM), y un monitor preferiblemente VGA a color o superior, el cual permitirá visualizar de mejor manera las gráficas. El programa para el presente trabajo fue desarrollado en un computador 486SX con 8 Mby de memoria RAM.

El programa desarrollado permite la simulación dinámica del péndulo invertido, aplicando los siguientes controles:

- Red de Compensación de Adelanto de Fase
- Acciones de Control PD
- Acciones de Control PID
- Control por Realimentación de Estado
- Control Fuzzy

1.2.- Importancia del Tema

En el área de Control, el computador se ha convertido en una herramienta valiosa, tanto para la enseñanza de las teorías como para la industria, obteniendo con ello diferentes alternativas para el control cada vez más seguras, estables y robustas.

Es así, como la Simulación Dinámica de un sistema real por computadora resulta económicamente ventajoso, por costos operativos, tiempo de realización del prototipo y cambios en las condiciones del sistema, permitiendo el ensayo de diferentes sistemas de control, y compensaciones que hagan cumplir las especificaciones deseadas; además se tiene una idea clara de cómo reacciona la planta frente a las alternativas del control planteadas, permitiendo realizar un análisis comparativo con los resultados obtenidos, constituyendo así un trabajo relativamente completo que servirá de respaldo en los laboratorios.

Por otro lado, el control por Lógica Fuzzy, permite la solución de sistemas que antes eran irresolubles con las teorías de control clásicas, pues estas requieren conocer el modelo matemático de la planta (lo que involucra un conocimiento profundo de ésta); en cambio para el control Fuzzy se requiere conocer la información cualitativa de la planta, permitiendo así la reducción de la complejidad de los proyectos.

Es decir las técnicas de modelaje y control Fuzzy utilizan la información cualitativa, tomando en cuenta la falta de exactitud del conocimiento de la planta.

En el programa desarrollado, se utilizó el lenguaje Visual Basic debido a que éste es un lenguaje de programación orientado a eventos, y contiene herramientas que permiten obtener un máximo provecho del entorno gráfico Windows para crear potentes aplicaciones con rapidez.

1.3.- Contenido

A continuación se da una breve resumen del contenido de cada uno de los capítulos del presente trabajo:

En el capítulo II se describe el Sistema del Péndulo Invertido, el proceso para la obtención del modelo matemático y adicionalmente se presentan las posibles alternativas de control con sus respectivos diagramas de bloques.

En el capítulo III se encuentra el desarrollo matemático, análisis y comparación de los resultados gráficos de cada una de las compensaciones de control que estabilizan al sistema, las cuales son:

- Compensación por Red de Adelanto de Fase en la cual sus parámetros son encontrados por el Método de la Bisectriz y el Método del Cero Arbitrario.
- Acciones de Control, de las cuales se desarrollaron el Control Proporcional Derivativo y el Control Proporcional Integral Derivativo.
- Realimentación de Estado.
- Control Fuzzy

En el capítulo IV se describe cada uno de los elementos que conforman la pantalla de la simulación dinámica, luego se presentan los diagramas de flujo del programa en general, de la simulación dinámica y de cada una de las alternativas de control desarrolladas en el capítulo III.

En el capítulo V se presentan el análisis y comparación de resultados obtenidos con cada uno de los controles y sus conclusiones generales.

Finalmente, como anexos se presentan, un listado de la nomenclatura que se utilizó en el desarrollo de la tesis, el manual del usuario del programa desarrollado y el listado del código fuente del mismo.

CAPITULO II: ANALISIS MATEMATICO

2.1.- Descripción del Sistema del Péndulo Invertido.

El problema planteado consiste básicamente, en una varilla que se encuentra apoyada en su extremo inferior de forma articulada sobre una plataforma móvil (carro), la cual se puede mover en dos direcciones (derecha o izquierda). El extremo superior de la varilla queda libre, lo que produce la inestabilidad en su posición vertical (ver Fig. 2.1.1).

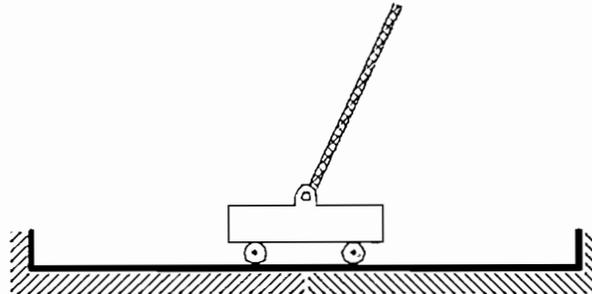


Fig.2.1.1.- Esquema del Péndulo Invertido

Se requiere entonces variar la velocidad del carro, para equilibrar las fuerzas inerciales de la varilla libre y llevarla a la posición de equilibrio angular (perpendicular a la pista).

El control se basa en calcular la fuerza externa que se ha de aplicar al carro para equilibrar la varilla ó péndulo, y para esto se observa que:

∴

- a) Si el péndulo tiende a caer a la derecha el carro se debe mover a la derecha, aplicándole una fuerza en esa dirección (que se definirá como positiva).
- b) Si el péndulo tiende a caer a la izquierda el carro se debe mover a la izquierda, aplicándole una fuerza en esa dirección (que será entonces negativa).

2.2.- Obtención de los Modelos Matemáticos.

Para obtener el modelo matemático, se ha de tomar en cuenta por separado el movimiento del péndulo y el movimiento del carrito considerando las fuerzas que actúan sobre ellos.

A los ejes de referencia se los ubicará de la siguiente manera: el Eje "Y" en el centro de la pista y el Eje "X" en el extremo inferior de la varilla. (ver Fig.2.2.1)

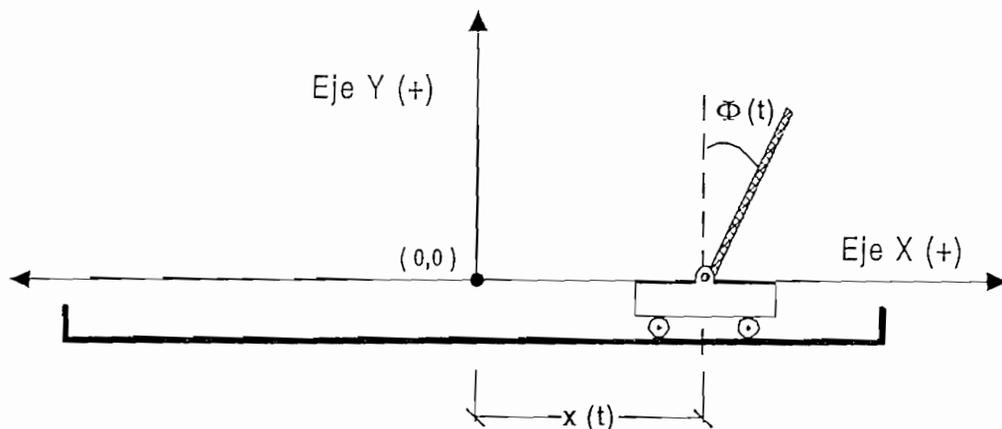


Fig.2.2.1.- Ubicación del Sistema de Referencia, usado para la Obtención del Modelo Matemático del Péndulo Invertido

Donde se define:

$x(t)$: desplazamiento desde el centro de la pista hasta el centro del carrito, se considera como positivo el desplazamiento hacia la derecha.

$\Phi(t)$: ángulo formado entre el péndulo y la vertical considerando como positivo los ángulos cuyo desplazamiento se realicen en sentido horario:

De esta manera las ecuaciones diferenciales que describen la dinámica del sistema, se encuentran del siguiente modo:

a) Movimiento de la varilla.

Para el análisis se considera que la varilla es homogénea y delgada, de longitud total $2L$ y por tanto con su centro de gravedad ubicado en la mitad de su longitud. (ver Fig.2.2.2)

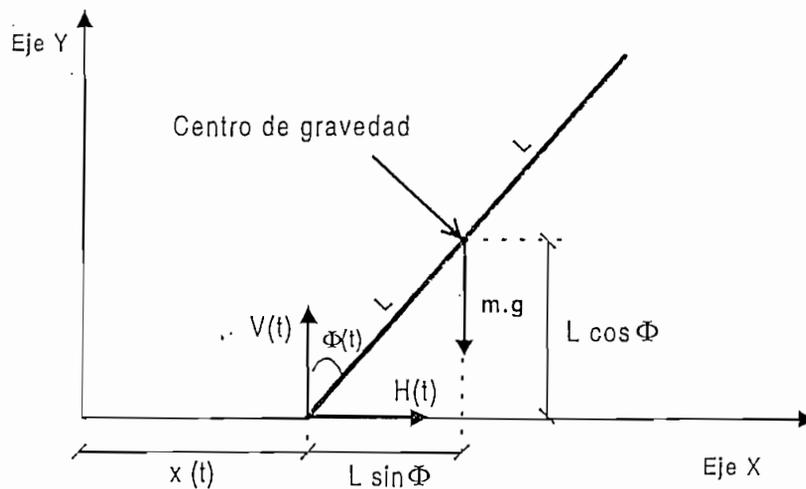


Fig. 2.2.2 Diagrama del Cuerpo Libre de la Varilla

Las fuerzas consideradas aplicadas a la varilla son:

- Peso de la varilla (mg) en el centro de gravedad.
- Reacción horizontal $H(t)$ debida al punto de contacto varilla-carrito.
- Reacción vertical $V(t)$ debida al punto de contacto varilla-carrito.

a.1) Aplicando sumatoria de fuerzas en el Eje X:

$$\sum F_x = ma_x$$

$$H(t) = ma_x$$

Donde la aceleración en los Ejes "X" e "Y" (a_x y a_y), se obtienen de la segunda derivada de la ubicación del centro de gravedad del péndulo.

$$a_x = \frac{\partial^2}{\partial t^2} (x(t) + L \sin \Phi(t))$$

se sabe que:

$$\frac{\partial^2}{\partial t^2} \sin \Phi(t) = \frac{\partial}{\partial t} \left[\cos \Phi(t) \left(\frac{\partial \Phi(t)}{\partial t} \right) \right] = -\sin \Phi(t) \left(\frac{\partial \Phi(t)}{\partial t} \right)^2 + \cos \Phi(t) \frac{\partial^2 \Phi(t)}{\partial t^2}$$

reemplazando:

$$a_x = \frac{\partial^2 x(t)}{\partial t^2} - L \sin \Phi(t) \left(\frac{\partial \Phi(t)}{\partial t} \right)^2 + L \cos \Phi(t) \frac{\partial^2 \Phi(t)}{\partial t^2}$$

entonces la ecuación de la sumatoria de fuerzas queda:

$$H(t) = m \frac{\partial^2 x(t)}{\partial t^2} + mL \cos \Phi(t) \frac{\partial^2 \Phi(t)}{\partial t^2} - mL \sin \Phi(t) \left(\frac{\partial \Phi(t)}{\partial t} \right)^2 \quad (\text{Ec.2.2.1})$$

a.2) Aplicando sumatoria de fuerzas en el Eje Y:

$$\sum F_y = m a_y$$

$$V(t) - mg = ma_y \quad (\text{Ec.2.2.2})$$

la aceleración queda definida por:

$$a_y = \frac{\partial^2}{\partial t^2}(L \cos \Phi(t)) = \frac{\partial}{\partial t} \left(-L \operatorname{sen} \Phi(t) \frac{\partial \Phi(t)}{\partial t} \right)$$

$$a_y = -L \cos \Phi(t) \left(\frac{\partial \Phi(t)}{\partial t} \right)^2 - L \operatorname{sen} \Phi(t) \frac{\partial^2 \Phi(t)}{\partial t^2}$$

reemplazando la ecuación anterior en la Ec.2.2.2 se tiene:

$$V(t) = mg - mL \operatorname{sen} \Phi(t) \frac{\partial^2 \Phi(t)}{\partial t^2} - mL \cos \Phi(t) \left(\frac{\partial \Phi(t)}{\partial t} \right)^2 \quad (\text{Ec.2.2.3})$$

a.3) Aplicando sumatoria de torques con respecto al centro de gravedad de la varilla

$$\sum \tau = J\alpha$$

donde:

J = es el momento de inercia del péndulo

α = es la aceleración angular del péndulo

τ = es el torque

la aceleración angular se la obtiene de la segunda derivada del ángulo:

$$\alpha = \frac{\partial^2 \Phi(t)}{\partial t^2}$$

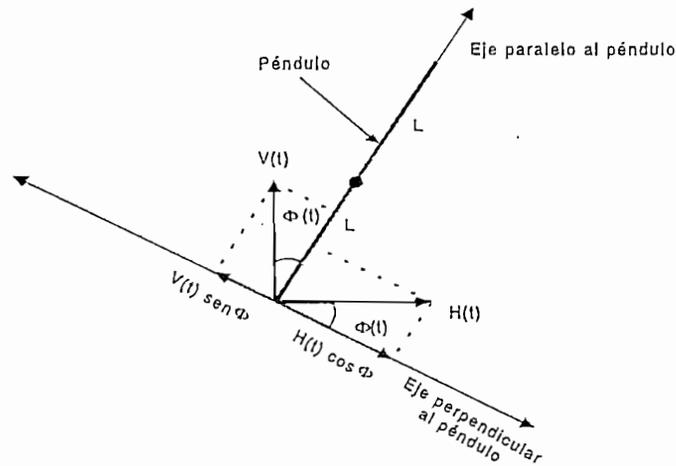


Fig.2.2.3.- Fuerzas que Ejercen Torque con Respecto al Centro de Gravedad del Péndulo

La ecuación de sumatoria de torques asumiendo como positivo los torques que actúan en sentido horario queda (ver Fig.2.2.3):

$$LV(t) \operatorname{sen} \Phi(t) - LH(t) \cos \Phi(t) = J \frac{\partial^2}{\partial t^2} \Phi(t) \quad (\text{Ec.2.2.4})$$

reemplazando las Ecs. 2.2.1 y 2.2.3 en Ec.2.2.4 se obtiene:

$$J \frac{\partial^2 \Phi(t)}{\partial t^2} = L \operatorname{sen} \Phi(t) \left[mg - mL \operatorname{sen} \Phi(t) \frac{\partial^2 \Phi(t)}{\partial t^2} - mL \cos \Phi(t) \left(\frac{\partial \Phi(t)}{\partial t} \right)^2 \right] \dots$$

$$\dots - L \cos \Phi(t) \left[m \frac{\partial^2 x}{\partial t^2} + mL \cos \Phi(t) \frac{\partial^2 \Phi(t)}{\partial t^2} - mL \operatorname{sen} \Phi(t) \left(\frac{\partial \Phi(t)}{\partial t} \right)^2 \right]$$

se conoce que:

$$(\operatorname{sen}^2 \Phi(t) + \cos^2 \Phi(t)) = 1$$

resolviendo se tiene:

$$(J + mL^2) \frac{\partial^2 \Phi(t)}{\partial t^2} - mgL \sin \Phi(t) + mL \cos \Phi(t) \frac{\partial^2 x}{\partial t^2} = 0$$

se reemplaza :

$$L' = \frac{J + mL^2}{mL}$$

y

$$J = \frac{mL^2}{3}$$

donde:

L' : es la longitud efectiva del péndulo y es igual a la longitud del péndulo equivalente con toda su masa concentrada en un punto, al extremo superior de la varilla.

J : es el Momento de Inercia (por tablas de momentos de inercia) de una varilla delgada y homogénea de longitud $2L$ y cuyo centro de masa está ubicado a una distancia L de su extremo inferior¹.

entonces queda:

$$\frac{\partial^2 \Phi(t)}{\partial t^2} - \frac{g \sin \Phi(t)}{L'} + \frac{\cos \Phi(t)}{L'} \frac{\partial^2 x}{\partial t^2} = 0 \quad (\text{Ec.2.2.5})$$

La ecuación 2.2.5, describe el movimiento del péndulo invertido, esta no es lineal pues existe en ella las funciones trigonométricas de seno y coseno, por lo que más adelante se la linealizará.

¹ OGATA Katsuhiko, "Ingeniería De Control Moderna", 1980, pag. 303.

b) Movimiento del carrito

Las fuerzas que actúan son (ver Fig.2.2.4) :

$u(t)$: fuerza que actúa sobre el carro para mantener la varilla vertical.

N : normal del carro debido al contacto entre carro-pista

Mg : peso del carro

$V(t)$: reacción vertical debido al punto de contacto varilla-carro

$H(t)$: reacción horizontal debido al punto de contacto varilla-carro

$x(t)$: posición del centro del carro respecto al eje de referencia

La fuerza de rozamiento se la desprecia para simplificar el modelo.

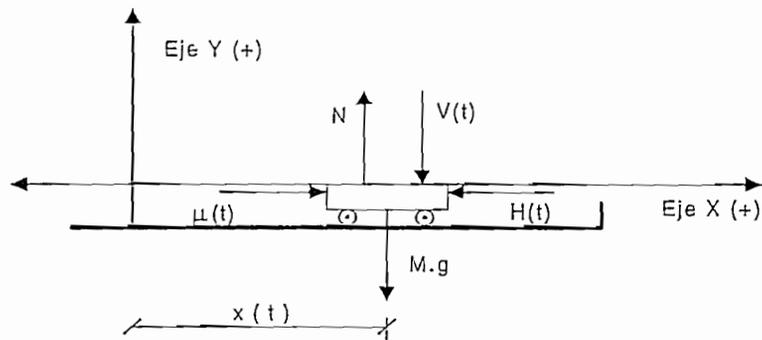


Fig.2.2.4.- Diagrama del Cuerpo Libre del Carro

b.1) Sumatoria de fuerzas en el Eje X:

$$\sum F_x = M a_c$$

$$\mu(t) - H(t) = M a_c$$

donde a_c es la aceleración del carro y es igual a:

$$a_c = \frac{\partial^2 x(t)}{\partial t^2}$$

reemplazando queda:

$$u(t) - H(t) = M \frac{\partial^2 x(t)}{\partial t^2} \quad (\text{Ec.2.2.6})$$

b.2) Sumatoria de fuerzas en el Eje Y:

$$\sum F_y = 0$$

$$N - Mg - V(t) = 0 \quad (\text{Ec.2.2.7})$$

se iguala a cero porque el carrito solamente se desplaza en el eje horizontal.

c) Linealización de las Ecs.2.2.5 y 2.2.6:

Por series de Taylor se sabe que:

$$\text{sen } \Phi = \Phi - \frac{\Phi^3}{3!} + \frac{\Phi^5}{5!} - \frac{\Phi^7}{7!} \dots$$

$$\text{cos } \Phi = 1 - \frac{\Phi^2}{2!} + \frac{\Phi^4}{4!} - \frac{\Phi^6}{6!} \dots$$

si se asume que Φ es pequeño entonces se puede decir que:

$$\text{sen } \Phi \approx \Phi \quad (\text{Ec.2.2.8})$$

$$\text{cos } \Phi \approx 1 \quad (\text{Ec.2.2.9})$$

como se ve en la Fig.2.2.5:

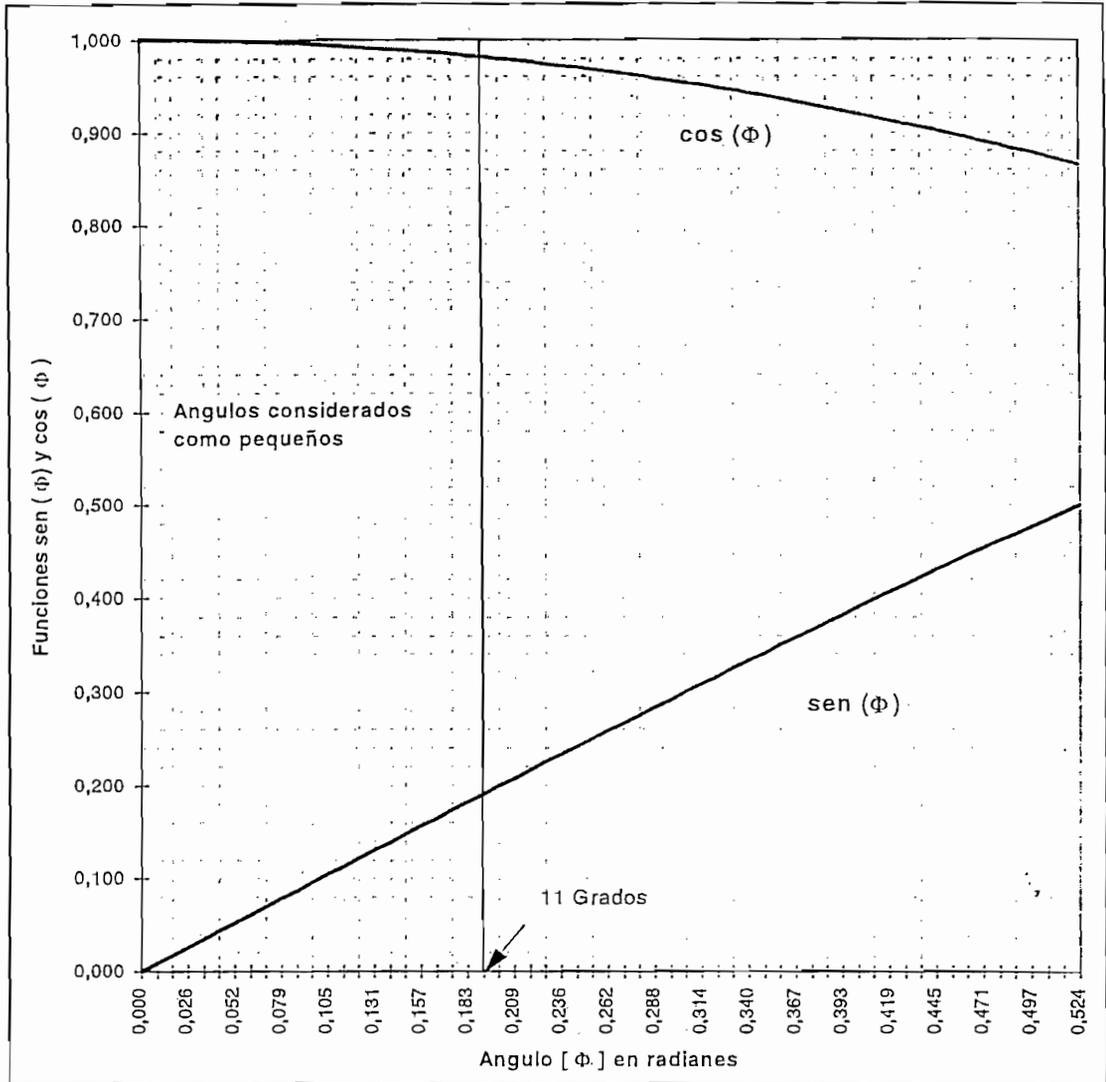


Fig.2.2.5.- Curva de las Funciones Seno y Coseno Vs Angulo en Radianes.

Para linealizar la Ec.2.2.5, se reemplazan las aproximaciones de seno y coseno en radianes, obteniendo:

$$\frac{\partial^2 \Phi(t)}{\partial t^2} - \frac{g}{L'} \Phi(t) + \frac{1}{L'} \frac{\partial^2 x}{\partial t^2} = 0 \quad (\text{Ec.2.2.10})$$

Ahora, se asume $M \gg m$ (masa del carro mucho mayor que masa del péndulo, por lo menos 10 veces mayor). Así $H(t)$ puede ser despreciada en la Ec.2.2.6 quedando reducida a la siguiente expresión:

$$M \frac{\partial^2 x(t)}{\partial t^2} = u(t) \quad (\text{Ec.2.2.11})$$

Las Ecs.2.2.10 y 2.2.11 describen la dinámica del sistema linealizado.

Para hallar la Función de Transferencia se aplica Laplace a las Ecs.2.2.10 y 2.2.11 y se asumen condiciones iniciales igual a cero:

de la Ec.2.2.11 se obtiene:

$$Ms^2 X(s) = U(s) \quad (\text{Ec.2.2.12})$$

Considerando que:

$U(s)$: variable de entrada (la fuerza)

$X(s)$: variable de salida (desplazamiento del carrito)

se tiene la Función de Transferencia:

$$\frac{X(s)}{U(s)} = \frac{1}{Ms^2} \quad (\text{Ec.2.2.13})$$

y su diagrama de bloques es:

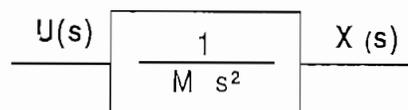


Fig.2.2.6.- Diagrama de Bloques de la Función de Transferencia de la Posición del Carro y la Fuerza

de la Ec.2.2.10 se obtiene:

$$s^2\Phi(s) - \frac{g}{L'}\Phi(s) + \frac{s^2 X(s)}{L'} = 0 \quad (\text{Ec.2.2.14})$$

ahora considerando que:

$U(s)$: variable de entrada (la fuerza)

$\Phi(s)$: variable de salida (ángulo entre la vertical y la varilla)

y despejando $X(s)$ de la Ec.2.2.12 para remplazarla en la Ec.2.2.14 se obtiene:

$$s^2\Phi(s) - \frac{g}{L'}\Phi(s) + \frac{s^2 U(s)}{L' Ms^2} = 0$$

simplificando:

$$\Phi(s) \left[s^2 - \frac{g}{L'} \right] = -\frac{U(s)}{L' M}$$

$$\frac{\Phi(s)}{U(s)} = -\frac{1}{ML' \left[s^2 - \frac{g}{L'} \right]}$$

entonces la función de transferencia es:

$$\frac{\Phi(s)}{-U(s)} = \frac{1}{ML' \left[s^2 - \frac{g}{L'} \right]} \quad (\text{Ec.2.2.15})$$

y su diagrama de bloques es:

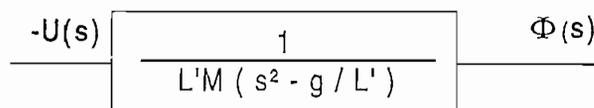


Fig.2.2.7 Diagrama de Bloques de la Función de Transferencia del Angulo y la Fuerza

Por lo tanto el sistema será inestable pues, para cualquier valor de ganancia siempre existirá un polo de lazo cerrado en el semiplano derecho de "s" como se ve en el Lugar Geométrico de las raíces del sistema sin compensar (Fig.2.2.8).

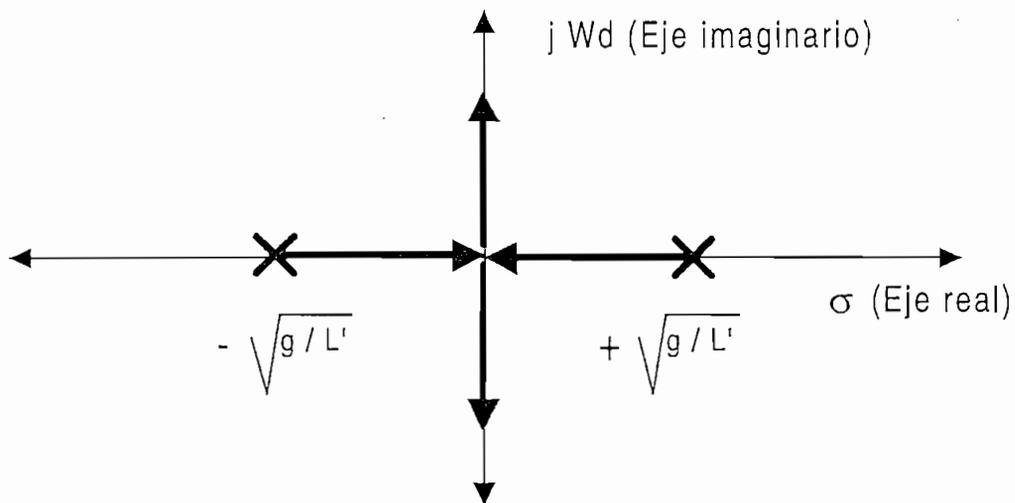


Fig.2.2.8.- Lugar Geométrico de las Raíces del Sistema sin Compensar

2.3.- Posibles Alternativas de Control

En el sistema del Péndulo Invertido se ensayarán diferentes alternativas de control, como son: las Acciones de Control, Compensación por Redes, Control por Realimentación de Estado y el Control Fuzzy; esta última es una técnica que hoy se encuentra en la vanguardia del diseño pues, el sistema de control actúa en base a una serie de reglas intuitivas o derivadas de la experiencia.

A continuación se analizará cada una de las técnicas de control mencionadas, indicando cuales lograrán estabilizar al péndulo para su posterior desarrollo en el Capítulo III.

a) Acción de Control Proporcional

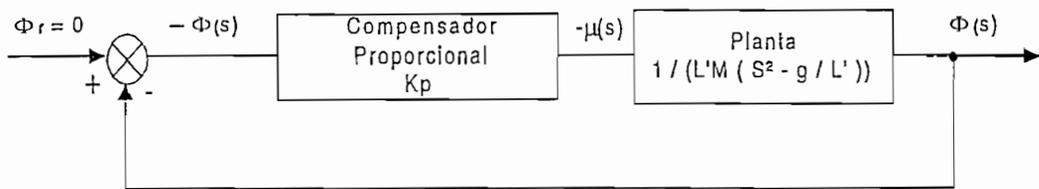


Fig.2.3.1.- Diagrama de Bloques del Control Proporcional

Este control simplemente hace un ajuste de ganancia como se puede ver en la Fig.2.3.1.

Al analizar el L.G.R. del sistema sin compensar (Fig.2.2.8) se deduce que ningún valor de ganancia (K_p) logrará que el sistema se estabilice, pues en ningún caso todas las raíces de lazo cerrado del sistema compensado estarán en el semiplano izquierdo del plano complejo.

b) Acción de Control Derivativo

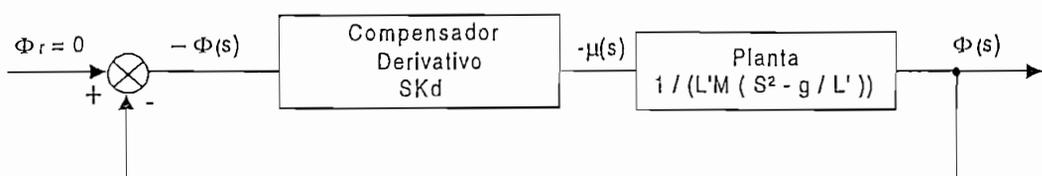


Fig.2.3.2.- Diagrama de Bloques del Control Derivativo

Esta acción de control como se observa en la Fig.2.3.2 aumenta un cero en el origen, manteniendo así un ramal del L.G.R. en el semiplano derecho (ver Fig.2.3.3), por lo que tampoco se logra estabilizar al sistema.

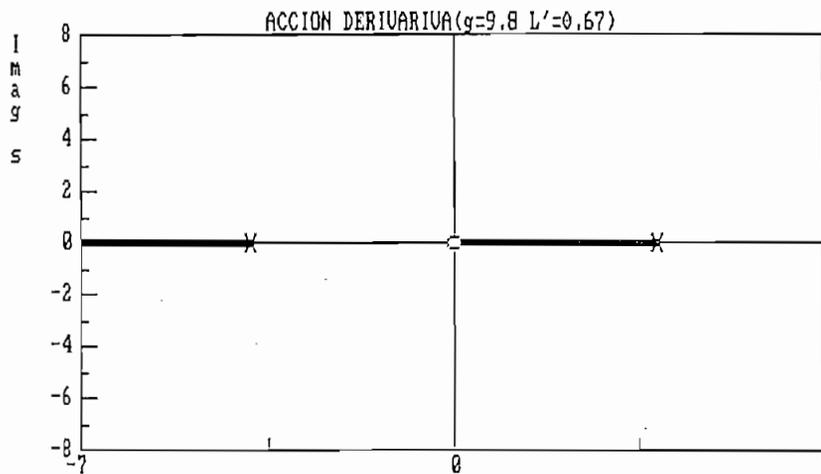


Fig.2.3.3.- L.G.R. del Sistema Compensado por el Control Derivativo

c) Acción de Control Integral

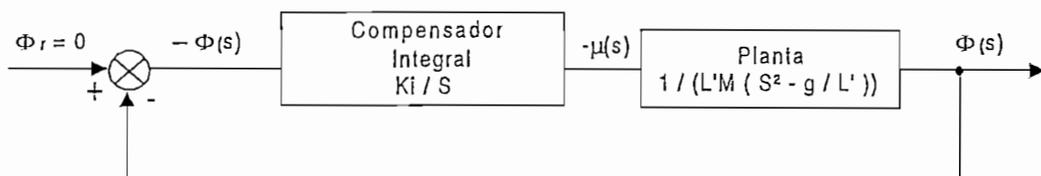


Fig.2.3.4.- Diagrama de Bloques del Control Integral

Esta acción de control aumenta un polo en el origen (Fig.2.3.4), aumentando a su vez las ramas que tienden al infinito.

Observando la Fig.2.3.5, el L.G.R. tiene ramales en el semiplano derecho, con lo cual el sistema se sigue manteniendo inestable.

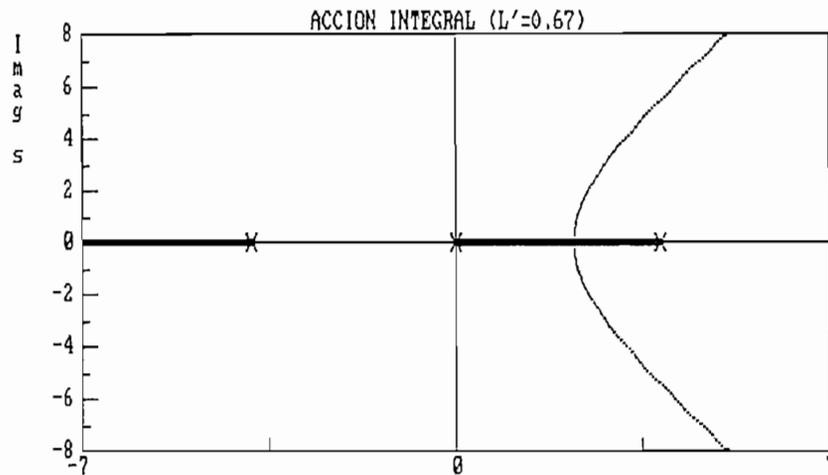


Fig.2.3.5.- L.G.R. del Sistema Compensado por el Control Integral

d) Acción de Control Proporcional Integral

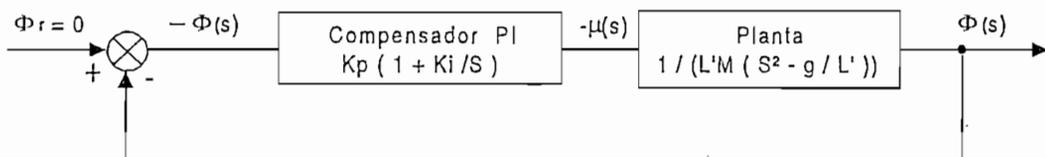


Fig.2.3.6.- Diagrama de Bloques del Control Proporcional Integral

Como se puede observar, esta acción de control añade un polo en el Eje (jw) (Fig.2.3.6), lo cual desplaza el L.G.R. hacia la derecha, impidiendo de esta manera estabilizar el sistema, para cualquier valor de K_i .

Adicionalmente se aumenta un cero en el semiplano izquierdo, por lo que el número de ramales que tienden al infinito siguen siendo dos (Fig.2.3.7).

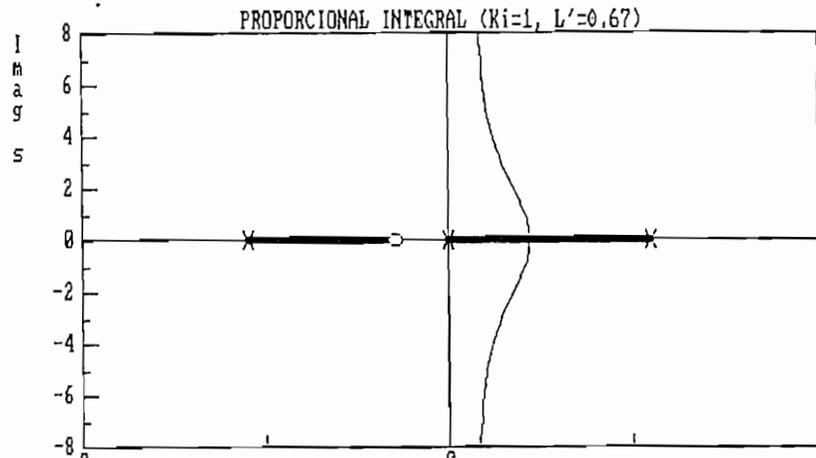


Fig.2.3.7.- L.G.R. del Sistema Compensado por el Control Proporcional Integral

e). Acción de Control Proporcional Derivativo

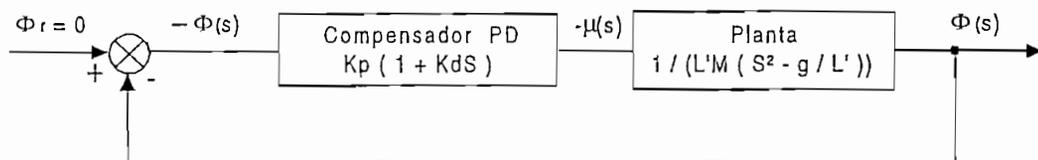


Fig.2.3.8.- Diagrama de Bloques del Control Proporcional Derivativo

Este tipo de acción de control añade un cero ubicado en el semiplano izquierdo a la planta (Fig.2.3.8), con lo que el L.G.R. del sistema compensado tiende a desplazarse hacia la izquierda logrando estabilizar el sistema.

e.1) Análisis del L.G.R.:**e.1.1) Número de ramas que tienden al infinito:**

$$m - n = 2 - 1 = 1$$

donde:

m: es el número de polos

n: es el número de ceros

e.1.2) Determinación de las asíntotas:

$$q = 0, 1, 2, \dots, m-n-1$$

$$q = 0$$

$$\Phi_A = \frac{(2q+1)}{m-n} 180$$

$$\Phi_A = 180^\circ$$

e.1.3) Centro de gravedad:

$$\sigma_A = \frac{\sum \text{polos} - \sum \text{ceros}}{m-n}$$

$$\sigma_A = \frac{1}{Kd}$$

e.1.4) Ecuación característica del sistema compensado:

$$1 + GH(s) = 0$$

$$1 + \frac{KpKd \left(s + \frac{1}{Kd} \right)}{ML' \left(s^2 - \frac{g}{L'} \right)} = 0$$

$$ML' \left(s^2 - \frac{g}{L'} \right) + KpKd \left(s + \frac{1}{Kd} \right) = 0$$

$$ML's^2 + K_p K_d s + (K_p - Mg) = 0 \quad (\text{Ec.2.3.1})$$

e.1.5) Estabilidad crítica:

Por el método de Routh Hurwitz aplicado a la Ec.2.3.1 se llega a determinar el límite de la estabilidad:

$$K_p = gM$$

entonces para valores de K_p mayores que gM , se asegura que las raíces del lazo cerrado estén a la izquierda del semiplano complejo, y por ende la estabilidad del sistema.

El cero del compensador se lo puede ubicar bien a la izquierda (ver Fig.2.3.9), o bien a la derecha (ver Fig.2.3.10) del polo ubicado en el semiplano izquierdo.

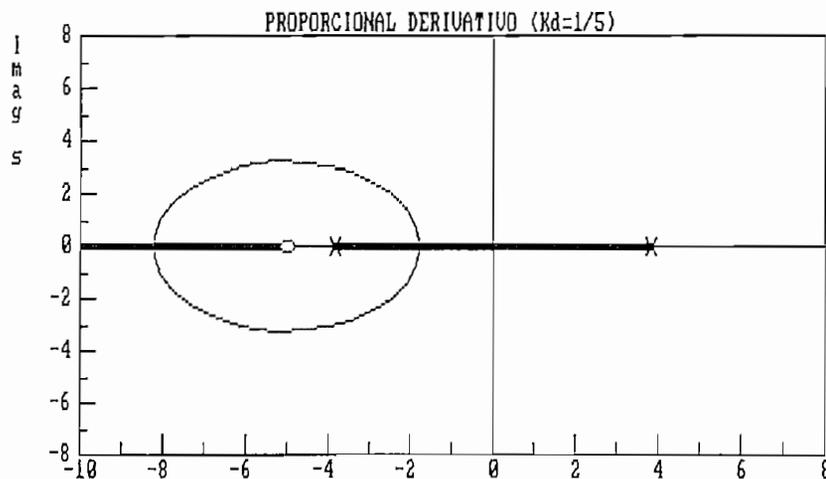


Fig.2.3.9.- L.G.R. del Sistema Compensado por el Control Proporcional Derivativo con el Cero Ubicado a la Izquierda del Polo

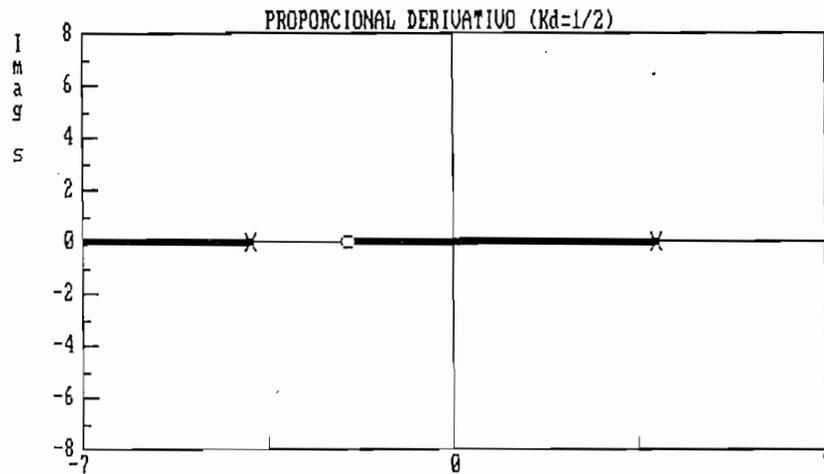


Fig.2.3.10.- L.G.R. del Sistema Compensado por el Control Proporcional Derivativo con el Cero Ubicado a la Derecha del Polo

f) Acción de Control Proporcional Integral Derivativo

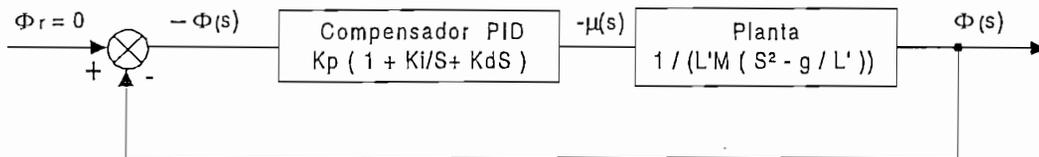


Fig.2.3.11.- Diagrama de Bloques del Control Proporcional Integral Derivativo

Esta acción de control añade dos ceros ubicados en el semiplano izquierdo y un polo ubicado en el origen, como se ve en la Fig.2.3.11

f.1) Análisis del L.G.R.:**f.1.1) Número de ramas que tienden al infinito:**

$$m - n = 3 - 2 = 1$$

donde:

m: es el número de polos

n: es el número de ceros

f.1.2) Determinación de las asíntotas:

$$q = 0, 1, 2, \dots, m - n - 1$$

$$q = 0$$

$$\Phi_A = \frac{(2q + 1)}{m - n} 180$$

$$\Phi_A = 180^\circ$$

f.1.3) Centro de gravedad:

$$\sigma_A = \frac{\sum \text{polos} - \sum \text{ceros}}{m - n}$$

$$\sigma_A = \frac{1}{Kd}$$

f.1.4) Ecuación característica del sistema compensado:

$$1 + GH(s) = 0$$

$$1 + \frac{Kp}{s} (Kds^2 + s + K_i) \left(\frac{1}{ML' \left(s^2 - \frac{g}{L'} \right)} \right) = 0$$

$$ML's \left(s^2 - \frac{g}{L'} \right) + Kp(Kds^2 + s + K_i) = 0$$

$$ML's^3 + KpKds^2 + (Kp - Mg)s + KpK_i = 0 \quad (\text{Ec.2.3.2})$$

En la Fig.2.3.12 se han seleccionado dos ceros complejos conjugados, de la función de transferencia compensada para el gráfico del Lugar Geométrico de la Raíces.

f.1.5) Estabilidad relativa

Por el método de Routh Hurwitz aplicado a la Ec.2.3.2 se llega a determinar cuales valores de K_p van a dar raíces ubicadas en el semiplano izquierdo para que el sistema sea estable:

$$K_p > \frac{ML'K_i + MgKd}{Kd}$$

Hay una gran variedad de métodos para obtener los parámetros del compensador P.I.D. En el programa desarrollado en el presente trabajo se da la libertad de seleccionar cualquier conjunto de valores para observar sus resultados.

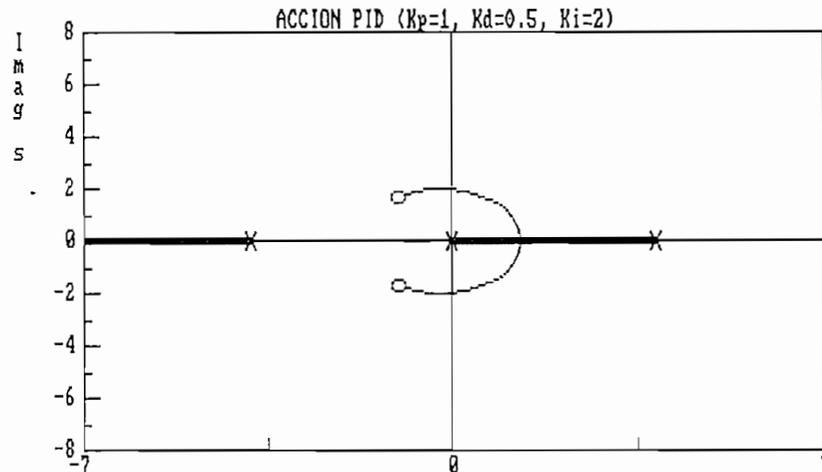


Fig.2.3.12.- L.G.R. del Sistema Compensado por el Control Proporcional Integral Derivativo

g) Compensación por Red de Adelanto de Fase

Esta compensación introduce un polo y un cero ubicados en el semiplano izquierdo (ver Fig.2.3.13), estando siempre el cero ubicado a la derecha del polo.

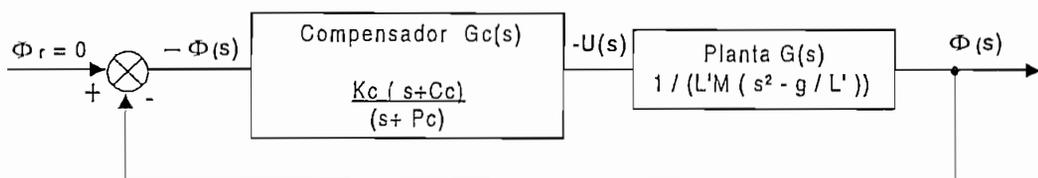


Fig.2.3.13.- Diagrama de Bloques de la Red de Adelanto y Atraso de fase

Aplicando la condición de ángulo de la planta compensada con referencia al polo deseado, se tiene que cumplir:

$$\phi|Gc|_{p_{d1}} + \phi|G|_{p_{d1}} = -180 \quad (\text{Ec.2.3.3})$$

y observando el L.G.R. de la planta sin compensar (ver Fig.2.3.14) se tiene que:

$$\alpha_1 + \alpha_2 < -180$$

$$\phi|G|_{P_{D1}} < -180$$

Por lo tanto, para cumplir con la condición de la Ec.2.3.3 se tiene que la fase aportada por la red es positiva:

$$\phi|Gc|_{P_{D1}} > 0 \quad (\text{Ec.2.3.4})$$

Lo cual es característica de esta compensación, pudiéndose compensar con un ángulo máximo de 60° debido a limitaciones físicas de la red.

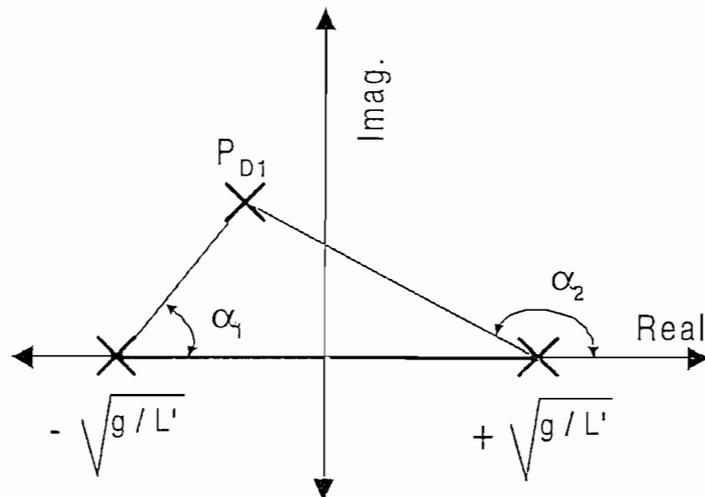


Fig.2.3.14.- Angulos que Forman los Polos de la Planta sin Compensar con el Polo Deseado

La Fig.2.3.15 muestra el L.G.R. con esta compensación.

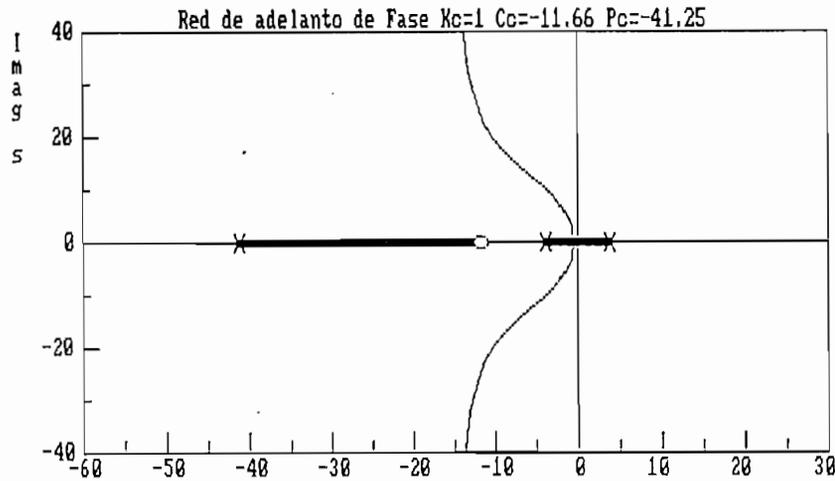


Fig.2.3.15.- L.G.R. del Sistema Compensado por Red de Adelanto de Fase

h) Compensación por Red de Atraso

Esta compensación introduce un polo y un cero ubicados en el semiplano izquierdo (ver Fig.2.3.13), estando siempre el polo ubicado a la derecha del cero, por lo que, la fase aportada por el compensador será negativa.

Tomando referencia a la Ec.2.3.4 se observa que esta red no conseguirá estabilizar al sistema.

i) Control por Realimentación de Estado

Una buena alternativa para controlar el sistema, es con la realimentación de estado, en donde la planta se la representa con una descripción a variables de estado, cuyo diagrama de bloques es el siguiente:

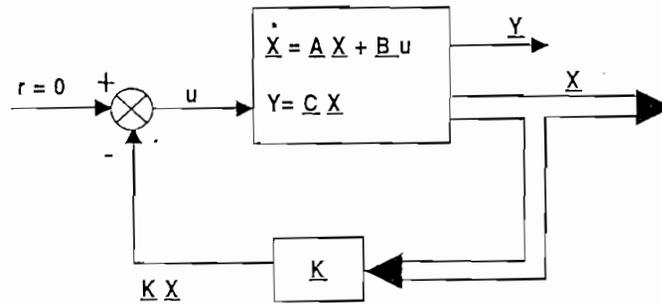


Fig.2.3.16.- Diagrama de Bloques de Realimentación de Estado

Con este control, se puede conseguir un sistema muy estable, pues el L.G.R. queda ubicado totalmente en el semiplano izquierdo, para problemas de mínima fase.

La señal de realimentación es una combinación lineal del vector de estado como puede verse en la Fig.2.3.16.

Para el diseño se asume la ubicación adecuada de las raíces para que cumplan con las especificaciones deseadas.

En este control se trabaja usualmente con múltiples entradas y salidas, aumentando así la complejidad pero mejorando los resultados obtenidos.

j) Control Fuzzy

También llamado control “Difuso” o “Borroso”, a diferencia de los otros controles mencionados busca antes que modelar el sistema

matemáticamente, describirlo mediante las formas de toma de decisión sobre el proceso, permitiendo que el sistema de control actúe más adecuadamente al seguir una serie de reglas intuitivas y derivadas de la experiencia, permitiendo de esta manera la reducción de la complejidad de proyectos y su implementación; además con esta forma de control se puede dar solución a problemas intratables con técnicas clásicas.

El Control Fuzzy emplea la lógica difusa, basada en la teoría de conjuntos difusos, donde un elemento no necesariamente pertenece o no pertenece a un conjunto, y en consecuencia hay un continuo de "grados de pertenencia".

A continuación se presenta el Diagrama de bloques del Control Fuzzy:

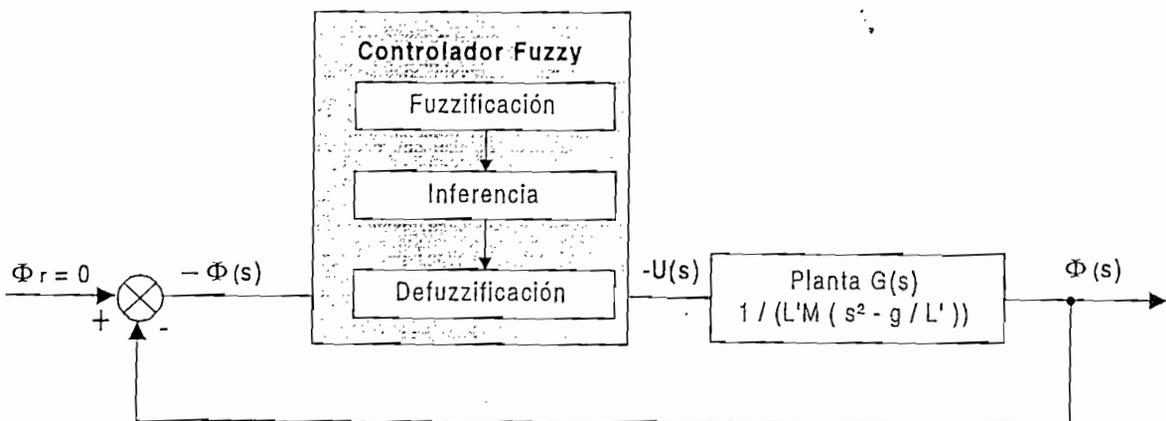


Fig.2.3.17.- Diagrama de Bloques para el Control Fuzzy

El proceso de fuzzificación lo que hace, es traducir las entradas numéricas en variables lingüísticas y determinar el correspondiente grado de pertenencia.

La inferencia, procesa los datos de entrada junto con las reglas dadas en la base de conocimiento, a fin de poder inferir las acciones de control Fuzzy a tomar.

La defuzzificación transforma las acciones fuzzy en acciones (valores numéricos) que puedan ser utilizados para controlar el sistema.

La teoría de control Fuzzy no será tratada en profundidad en esta tesis, pues este ya fue tratado en una tesis previa¹.

¹ MONTOYA Omar, "Desarrollo del modelo flexoBase-C y aplicación al control de Temperatura de una termoselladora mediante Lógica Fuzzy", Tesis de grado, 1996.

CAPITULO III: COMPENSACIONES

3.1.- Compensación por Redes y Análisis de Resultados

La red de compensación esta ubicada en cascada con la planta como se puede ver en la Fig.2.3.13. y su función de transferencia tiene la siguiente estructura:

$$G_c(s) = K_c \frac{s + C_c}{s + P_c} \quad (\text{Ec.3.1.1})$$

Como ya se mencionó en el Capítulo II, la red de compensación del sistema debe ser una red de adelanto de fase.

Para el diseño de sus parámetros se usarán dos técnicas basadas en el lugar geométrico, que son:

- a) Método de la Bisectriz
- b) Método del Cero Arbitrario

Estas técnicas de compensación ofrecen buenos resultados cuando las especificaciones deseadas del sistema se dan en términos de magnitudes en el dominio del tiempo, tales como máximo sobreimpulso (M_p) y tiempo de establecimiento (t_s).

a) Método de la Bisectriz

El procedimiento a seguir para el diseño es el siguiente:

- a.1) A partir del tiempo de establecimiento del sistema y el máximo sobreimpulso requeridos para una adecuada respuesta transitoria del sistema, se encuentran las raíces deseadas, las cuales deben determinar el comportamiento del sistema.

sea:

t_s = Tiempo de establecimiento criterio del 2%.

M_p = Máximo sobreimpulso en tanto por uno.

Ln = Logaritmo natural.

se sabe que¹ :

$$M_p = e^{-\frac{\zeta \pi}{\sqrt{1-\zeta^2}}}$$

despejando de la ecuación anterior, el coeficiente de amortiguamiento (ζ) queda:

$$\zeta = \frac{(Ln M_p)^2}{\pi^2 + (Ln M_p)^2} \quad (\text{Ec.3.1.2})$$

con el tiempo de establecimiento (criterio del 2%) se obtiene la frecuencia natural no amortiguada (ω_n):

$$\omega_n = \frac{4}{\zeta t_s} \quad (\text{Ec.3.1.3})$$

entonces las raíces deseadas son:

¹ OGATA Katsuhiko, "Ingeniería De Control Moderna", 1980, pag. 254.

$$P_{D1,2} = -\zeta W_n \pm jW_n \sqrt{1-\zeta^2} \quad (\text{Ec.3.1.4})$$

si:

$$\sigma = \zeta W_n \quad (\text{Ec.3.1.5})$$

$$Wd = W_n \sqrt{1-\zeta^2} \quad (\text{Ec.3.1.6})$$

reemplazando las Ecs. 3.1.5 y 3.1.6 en la Ec.3.1.4 se tiene:

$$P_{D1,2} = -\sigma \pm jWd \quad (\text{Ec.3.1.7})$$

a.2) Cálculo de la fase que debe aportar el compensador para que el L.G.R. pase por las raíces deseadas.

se define como:

$$\theta_{planta} = \text{Fase del sistema sin compensar}$$

$$\theta_c = \text{Fase que proporciona el compensador}$$

$$\theta_{PD} = \text{Fase de las raíces deseadas}$$

por la condición de fase se tiene:

$$\theta |G_c(s) G_{planta}(s)|_{P_{D1,2}} = -180^\circ$$

entonces:

$$\theta_c + \theta_{planta} = -180$$

despejando θ_c se tiene:

$$\theta_c = -180^\circ - \theta_{planta} \quad (\text{Ec.3.1.8})$$

de la Ec.3.1.7 la fase de la raíz deseada queda definida por:

$$\theta_{PD} = 180^\circ - \arctan\left(\frac{Wd}{\sigma}\right) \quad (\text{Ec.3.1.9})$$

reemplazando la raíz deseada en la Función de Transferencia de lazo abierto de la planta se tiene:

$$G(s)_{planta} = \frac{1}{ML' \left[s^2 - \frac{g}{L'} \right]} \Bigg|_{p_{pl} = -\sigma + jWd}$$

entonces queda:

$$G(s)_{planta} = \frac{1}{ML' \left[(-\sigma + jWd)^2 - \frac{g}{L'} \right]}$$

resolviendo:

$$G(s)_{planta} = \frac{1}{ML' \left[\sigma^2 - Wd^2 - 2j\sigma Wd - \frac{g}{L'} \right]}$$

representando en forma polar la ecuación anterior:

$$G(s)_{planta} = \frac{1}{ML' \left[\sqrt{\left(\sigma^2 - Wd^2 - \frac{g}{L'} \right)^2 + (2\sigma Wd)^2} \left(\text{tg}^{-1} \frac{-2\sigma Wd}{\left(\sigma^2 - Wd^2 - \frac{g}{L'} \right)} \right) \right]} \quad (\text{Ec.3.1.10})$$

así la fase de la planta queda:

$$\theta_{planta} = -\text{tg}^{-1} \frac{-2\sigma Wd}{\left(\sigma^2 - Wd^2 - \frac{g}{L'} \right)} \quad (\text{Ec.3.1.11})$$

analizando la Ec.3.1.11 se debe tomar en cuenta los siguientes casos en el programa:

Primer caso:

Si: $\sigma^2 - Wd^2 - \frac{g}{L'} = 0$

entonces:

$$\theta_{planta} = 90^\circ$$

y

$$\theta_c = -180 - 90 = -270^\circ$$

Este valor de fase no permite encontrar los parámetros del compensador (K_c , P_c y C_c), pues como ya se mencionó en el capítulo anterior la fase del compensador debe ser positiva y no mayor a 60 grados por lo tanto el programa presenta un mensaje indicando que se deben cambiar las especificaciones ingresadas.

Segundo caso :

Si: $\sigma^2 - Wd^2 - \frac{g}{L'} > 0$

$$\theta_{planta} = -\text{tg}^{-1} \frac{-2\sigma Wd}{\left(\sigma^2 - Wd^2 - \frac{g}{L'}\right)}$$

La fase de la planta será positiva y menor a 90 grados

$$\theta_c = -180 + \text{tg}^{-1} \frac{-2\sigma Wd}{\left(\sigma^2 - Wd^2 - \frac{g}{L'}\right)}$$

La fase del compensador resulta ser negativa y menor a -90 grados y al igual que en el caso anterior, este valor de fase no permite encontrar los parámetros del compensar.

Tercer caso:

$$\text{Si: } \sigma^2 - Wd^2 - \frac{g}{L'} < 0$$

entonces:

$$\theta_{planta} = - \left(\text{tg}^{-1} \frac{-2\sigma Wd}{\left(\sigma^2 - Wd^2 - \frac{g}{L'} \right)} + 180 \right)$$

$$\theta_c = -180 + \text{tg}^{-1} \frac{-2\sigma Wd}{\left(\sigma^2 - Wd^2 - \frac{g}{L'} \right)} + 180$$

$$\theta_c = \text{tg}^{-1} \frac{-2\sigma Wd}{\left(\sigma^2 - Wd^2 - \frac{g}{L'} \right)} \quad (\text{Ec.3.1.12})$$

Esta fase es positiva y es por tanto el único caso en el que se podrá encontrar los parámetros del compensador y además que el lugar geométrico pase por las raíces deseadas; adicionalmente se chequea en el programa que este valor sea menor de 60 grados.

a.3) Para la ubicación del polo y cero del compensador, es necesario recordar que en la red de adelanto de fase el cero siempre debe estar localizado a la derecha del polo en el semiplano izquierdo del plano complejo S.

El Método de la Bisectriz ¹ encuentra la ubicación del polo y cero del compensador de manera geométrica como se ve en la Fig.3.1.1:

¹ OGATA Katsuhiko, "Ingeniería De Control Moderna" , 1980, pag. 528-533.

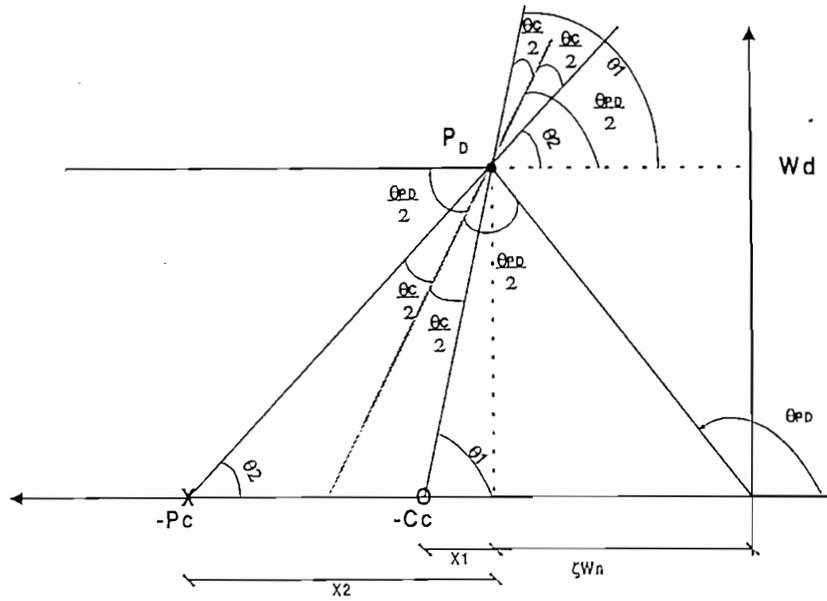


Fig.3.1.1.- Ubicación del Polo y el Cero del Compensador por el Método de la Bisectriz

del gráfico (Fig.3.1.1) los ángulos del polo y cero son:

$$\tan \theta_1 = \frac{Wd}{x_1}$$

$$\tan \theta_2 = \frac{Wd}{x_2}$$

$$\theta_1 = \frac{\theta_c}{2} + \frac{\theta_p}{2}$$

$$\theta_2 = \frac{\theta_p}{2} - \frac{\theta_c}{2}$$

donde el polo y el cero están ubicados a:

$$-P_c = -\zeta W_n - x_2$$

$$-C_c = -\zeta W_n - x_1$$

por lo tanto, quedan definidas la ubicación de las singularidades:

$$P_c = \zeta Wn + \frac{Wd}{\tan\left(\frac{\theta_{P_n} - \theta_c}{2}\right)} \quad (\text{Ec.3.1.13})$$

$$C_c = \zeta Wn + \frac{Wd}{\tan\left(\frac{\theta_{P_n} + \theta_c}{2}\right)} \quad (\text{Ec.3.1.14})$$

con la condición de Módulo se calcula la ganancia del compensador K_c :

$$|G_c(s)G_{PLANTA}(s)| = 1 \quad (\text{Ec.3.1.15})$$

sea el módulo del compensador igual a:

$$|G_c(s)| = \left| K_c \frac{s + C_c}{s + P_c} \right|_{P_n = -\sigma + jWd}$$

reemplazando la raíz deseada se tiene:

$$|G_c(s)| = \left| K_c \frac{\sqrt{(-\sigma + C_c)^2 + Wd^2}}{\sqrt{(-\sigma + P_c)^2 + Wd^2}} \right| \quad (\text{Ec.3.1.16})$$

sustituyendo el módulo de la Ec.3.1.10 y Ec.3.1.16 en Ec.3.1.15 se tiene:

$$\left| K_c \frac{\sqrt{(-\sigma + C_c)^2 + Wd^2}}{\sqrt{(-\sigma + P_c)^2 + Wd^2}} \right| \cdot \left| \frac{1}{ML' \left[\sqrt{\left(\sigma^2 - Wd^2 - \frac{g}{L'} \right)^2 + (2\sigma Wd)^2} \right]} \right| = 1$$

despejando la ecuación anterior:

$$K_c = \frac{\sqrt{(-\sigma + P_c)^2 + Wd^2}}{\sqrt{(-\sigma + C_c)^2 + Wd^2}} \cdot ML' \left[\sqrt{(2\sigma Wd)^2 + \left(\sigma^2 - Wd^2 - \frac{g}{L'} \right)^2} \right] \quad (\text{Ec.3.1.17})$$

b) Método del Cero Arbitrario

Con este método se obtienen de mejor manera los parámetros de la red compensadora, pues al ubicar el cero en una posición adecuada se puede lograr que las raíces deseadas coincidan con los raíces dominantes cumpliendo así con las especificaciones dadas.

Al igual que el Método de la Bisectriz, a partir del máximo sobreimpulso y tiempo de establecimiento, se encuentran las raíces deseadas ($P_{D1,2}$) de lazo cerrado del sistema compensado y la fase que debe proporcionar el compensador (θ_c).

Dada la ubicación del cero de la red compensadora (C_c) en una posición conveniente, se debe calcular la ubicación del polo (P_c), de tal manera que se verifique la fase que proporcionó el compensador.

Evaluando la fase de la F.T. del compensador en la raíz deseada se obtiene:

$$\theta_c = \theta \left| K_c \frac{(s + C_c)}{(s + P_c)} \right|_{-\sigma + jWd}$$

reemplazando:

$$\theta_c = \theta \left| K_c \frac{(-\sigma + jWd + C_c)}{-\sigma + jWd + P_c} \right|$$

se obtiene la fase

$$\theta_c = \tan^{-1} \left(\frac{Wd}{-\sigma + C_c} \right) - \tan^{-1} \left(\frac{Wd}{-\sigma + P_c} \right)$$

luego resolviendo se tiene:

$$\begin{aligned} \tan^{-1}\left(\frac{Wd}{-\sigma + P_c}\right) &= \tan^{-1}\left(\frac{Wd}{-\sigma + C_c}\right) - \theta_c \\ \tan\left[\tan^{-1}\left(\frac{Wd}{-\sigma + P_c}\right)\right] &= \tan\left[\tan^{-1}\left(\frac{Wd}{-\sigma + C_c}\right) - \theta_c\right] \\ Wd &= (-\sigma + P_c)\tan\left[\tan^{-1}\left(\frac{Wd}{-\sigma + C_c}\right) - \theta_c\right] \end{aligned}$$

finalmente despejando se encuentra el polo del compensador:

$$P_c = \sigma + \frac{Wd}{\tan\left[\tan^{-1}\left(\frac{Wd}{-\sigma + C_c}\right) - \theta_c\right]} \quad (\text{Ec.3.1.18})$$

analizando la Ec.3.1.18 se presentan cuatro casos a tomar en cuenta en el programa:

1) Si: $-\sigma + C_c > 0$:

La ubicación del polo queda determinada por la Ec.3.1.18

2) Si: $-\sigma + C_c = 0$:

$$P_c = \sigma + \frac{Wd}{\tan[90 - \theta_c]} \quad (\text{Ec.3.1.19})$$

3) Si: $-\sigma + C_c < 0$:

$$P_c = \sigma + \frac{Wd}{\tan\left[180 + \tan^{-1}\left(\frac{Wd}{-\sigma + C_c}\right) - \theta_c\right]} \quad (\text{Ec.3.1.20})$$

4) *Sí el denominador es cero:*

En este caso, el programa desarrollado muestra un mensaje indicando que se ingrese otros valores para la compensación.

La ganancia del compensador K_c se la obtiene de la misma forma que en el método de la Bisectriz (ver Ec.3.1.17).

Respuesta del sistema considerando condiciones iniciales

Para la simulación dinámica y por ende la obtención de la respuesta del sistema, se deben tomar en cuenta las condiciones iniciales, para lo cual se desarrolla el siguiente proceso:

despejando de la Ec.2.2.11 se obtiene:

$$\frac{\partial^2 x(t)}{\partial t^2} = \frac{\mu(t)}{M}$$

y reemplazando en la Ec.2.2.10:

$$\frac{\partial^2 \Phi}{\partial t^2} - \frac{g}{L'} \Phi(t) + \frac{1}{L' M} \mu(t) = 0 \quad (\text{Ec.3.1.21})$$

despejando se tiene:

$$L' M \frac{\partial^2 \Phi(t)}{\partial t^2} - g M \Phi(t) = -\mu(t) \quad (\text{Ec.3.1.22})$$

derivando la Ec.3.1.22 se tiene:

$$L' M \frac{\partial^3 \Phi(t)}{\partial t^3} - gM \frac{\partial \Phi(t)}{\partial t} = - \frac{\partial u(t)}{\partial t} \quad (\text{Ec.3.1.23})$$

a partir de la función de transferencia del compensador Ec.3.1.1 se tiene:

$$U(s)s + U(s)P_c = K_c s \Phi(s) + K_c C_c \Phi(s)$$

donde el equivalente en el tiempo de la ecuación anterior es:

$$\frac{\partial u(t)}{\partial t} + u(t) P_c = K_c \frac{\partial \Phi(t)}{\partial t} + K_c C_c \Phi(t) \quad (\text{Ec.3.1.24})$$

reemplazando las Ecs.3.1.22 y.3.1.23 en la Ec.3.1.24 e igualando a cero se obtiene:

$$-L' M \frac{\partial^3 \Phi(t)}{\partial t^3} - L' M P_c \frac{\partial^2 \Phi(t)}{\partial t^2} + (gM - K_c) \frac{\partial \Phi(t)}{\partial t} + (g M P_c - K_c C_c) \Phi(t) = 0 \quad (\text{Ec.3.1.25})$$

aplicando Laplace con C.I. ≠ 0 y despejando $\Phi(s)$ se llega a:

$$\Phi(s) = \frac{Fb_2 s^2 + Fb_1 s + Fb_0}{s^3 + s^2 P_c + s \left(\frac{K_c}{L' M} - \frac{g}{L'} \right) + \frac{K_c C_c}{L' M} - \frac{g P_c}{L'}} \quad (\text{Ec.3.1.26})$$

donde:

$$Fb_2 = \Phi(0)$$

$$Fb_1 = \dot{\Phi}(0) P_c + \ddot{\Phi}(0)$$

$$Fb_0 = \ddot{\Phi}(0) + \dot{\Phi}(0) P_c + \frac{\Phi(0) K_c}{L' M} - \frac{\Phi(0) g M}{L' M}$$

evaluando la Ec.3.1.25 en t=0 se tiene:

$$\frac{\partial^2 \Phi(0)}{\partial t^2} = \frac{(-g M P_c + K_c C_c) \Phi(0) + (-gM + K_c) \frac{\partial \Phi(0)}{\partial t}}{-L' M P_c}$$

igualando el denominador de la Ec.3.1.26 a un sistema de la forma:

$$\Phi(s) = \frac{Fb_2 s^2 + Fb_1 s + Fb_0}{[(s + \sigma)^2 + Wd^2][s + y]} \quad (\text{Ec.3.1.27})$$

queda:

$$y = Pc - 2\sigma$$

dividiendo en fracciones parciales la Ec.3.1.27 se tiene:

$$\Phi(s) = \frac{A}{s + y} + \frac{B(s + \sigma)}{(s + \sigma)^2 + Wd^2} + \frac{CWd}{(s + \sigma)^2 + Wd^2}$$

y aplicando la transformada inversa de Laplace:

$$\Phi(t) = Ae^{-y't} + Be^{-\sigma t} \cos Wdt + Ce^{-\sigma t} \sen Wdt \quad (\text{Ec.3.1.28})$$

donde:

$$A = \frac{Fb_2(-y)^2 + Fb_1(-y) + Fb_0}{[(-y + \sigma)^2 + Wd^2]}$$

$$B = Fb_2 - A$$

$$C = \frac{Fb_1 - 2\sigma A - B(\sigma + y)}{Wd}$$

la velocidad angular se obtiene derivando la Ec.3.1.28:

$$\omega(t) = A_1 e^{-y't} + B_1 e^{-\sigma t} \cos Wdt + C_1 e^{-\sigma t} \sen Wdt \quad (\text{Ec.3.1.29})$$

donde:

$$A_1 = -Ay$$

$$B_1 = -\sigma B + CWd$$

$$C_1 = -WdB - \sigma C$$

derivando por segunda vez la Ec.3.1.29 se obtiene:

$$\ddot{\Phi}(t) = A_2 e^{-y't} + B_2 e^{-\sigma t} \cos Wdt + C_2 e^{-\sigma t} \sen Wdt \quad (\text{Ec.3.1.30})$$

donde:

$$A_2 = -A_1 y$$

$$B_2 = -\sigma B_1 + C_1 W d$$

$$C_2 = -W d B_1 - \sigma C_1$$

A continuación se detalla el procedimiento para encontrar las ecuaciones en el tiempo de la posición y velocidad del carrito.

como se vio anteriormente la fuerza es igual a:

$$u(t) = M\ddot{x}(t) \quad (\text{Ec.3.1.31})$$

reemplazando la ecuación anterior en la Ec.3.1.22 y dividiendo para $L'M$ se obtiene:

$$\ddot{\Phi}(t) - \frac{g}{L'}\Phi(t) = -\frac{\ddot{x}(t)}{L'} \quad (\text{Ec.3.1.32})$$

despejando:

$$\ddot{x}(t) = -\ddot{\Phi}(t)L' + g\Phi(t) \quad (\text{Ec.3.1.33})$$

integrando la ecuación anterior:

$$\dot{x}(t) = -\dot{\Phi}(t)L' + g\int\Phi(t) + C_1 \quad (\text{Ec.3.1.34})$$

reemplazando en la Ec.3.1.24 la derivada de la Ec.3.1.31 se tiene:

$$\Phi(t) = \frac{\ddot{x}(t)M}{KcC_c} + \frac{P_c M\dot{x}(t)}{KcC_c} - \frac{\dot{\Phi}(t)}{C_c}$$

integrando la ecuación anterior:

$$\int\Phi(t) = \frac{u(t)}{KcC_c} + \frac{P_c M\dot{x}(t)}{KcC_c} - \frac{\Phi(t)}{C_c} \quad (\text{Ec.3.1.35})$$

reemplazando la Ec.3.1.35 en la Ec.3.1.34 se llega:

$$\dot{x}(t) = v_{A1}u(t) + v_{A2}\dot{\Phi}(t) + v_{A3}\Phi(t) + C_1$$

$$v(t) = v_{A1}u(t) + v_{A2}\dot{\Phi}(t) + v_{A3}\Phi(t) + C_1 \quad (\text{Ec.3.1.36})$$

donde C_1 se obtiene evaluando en $t = 0$ la Ec.3.1.36:

$$C_1 = v(0) - v_{A1}u(0) - v_{A2}\dot{\Phi}(0) - v_{A3}\Phi(0)$$

y

$$v_{A1} = \frac{-g}{P_C M g - K c C_C}$$

$$v_{A2} = \frac{K c C_C L'}{P_C M g - K c C_C}$$

$$v_{A3} = \frac{g K c}{P_C M g - K c C_C}$$

integrando por segunda vez la Ec.3.1.36 se obtiene la ecuación de espacio del carrito:

$$x(t) = X_{A1}\dot{x}(t) + X_{A2}\Phi(t) + X_{A3}u(t) + C_1 t + C_2 \quad (\text{Ec.3.1.37})$$

donde C_2 se encuentra evaluando la Ec.3.1.37 en cero:

$$C_2 = x(0) - X_{A1}\dot{x}(0) - X_{A2}\Phi(0) - X_{A3}u(0)$$

y

$$X_{A1} = v_{A1}M + \frac{v_{A3}P_C M}{K c C_C}$$

$$X_{A2} = v_{A2}M - \frac{v_{A3}}{C_C}$$

$$X_{A3} = \frac{v_{A3}}{K c C_C}$$

Finalmente, las ecuaciones descritas a continuación son las que describen la dinámica del sistema compensado considerando $CI \neq 0$:

$$\Phi(t) = Ae^{-\gamma t} + Be^{-\sigma t} \cos Wdt + Ce^{-\sigma t} \operatorname{sen} Wdt \quad (\text{Ec.3.1.28})$$

$$\omega(t) = A_1 e^{-\gamma t} + B_1 e^{-\sigma t} \cos Wdt + C_1 e^{-\sigma t} \operatorname{sen} Wdt \quad (\text{Ec.3.1.29})$$

$$v(t) = v_{A_1} u(t) + v_{A_2} \dot{\Phi}(t) + v_{A_3} \Phi(t) + C_1 \quad (\text{Ec.3.1.36})$$

$$x(t) = X_{A_1} \dot{x}(t) + X_{A_2} \Phi(t) + X_{A_3} u(t) + C_1 t + C_2 \quad (\text{Ec.3.1.37})$$

Análisis de Resultados con el Método de la Bisectriz

En el programa desarrollado, cuando se escoge la compensación por red de adelanto de fase con la técnica de la bisectriz, se ingresan como datos el tiempo de establecimiento ($t_{s\ 2\%}$) y el máximo sobreimpulso (M_p) en tanto por ciento, donde un sistema con estabilidad aceptable tendría un M_p menor al 40% y un tiempo de establecimiento que permita que el control lleve al péndulo a su posición de referencia (ángulo cero) antes que el carrito sobrepase los límites de la pista.

A continuación se presenta un ejemplo corrido en el programa que cumple con las condiciones requeridas:

Datos generales del péndulo:

$$M = 1\text{Kg}$$

$$m = 0,1\text{Kg}$$

$$L = 0,5\text{m}$$

$$g = 9,8\text{m/s}^2$$

$$\text{Longitud de la pista} = 1\text{m}$$

Condiciones Iniciales:

$$\text{Espacio} = 0\text{m}$$

$$\text{Velocidad} = 0\text{m/s}$$

$$\text{Ángulo} = 10\text{grad}$$

$$\text{Vel. Angular} = 0\text{grad/s}$$

Los datos anteriores son generales para todos los controles, los cuales serán usados como ejemplo en todas las compensaciones, a fin hacer más adelante una comparación entre ellas.

Datos ingresados para este control:

$$t_s = 1 \text{ seg}$$

$$M_p = 30\%$$

Para el análisis de los gráficos se define:

- Tiempo de crecimiento: t_r
- Tiempo en alcanzar el primer pico : t_p
- Tiempo de establecimiento criterio del 2 % : t_s
- Tiempo del primer cambio de dirección de la velocidad del carro: t_{cvc}
- Tiempo del primer cambio de dirección de la fuerza: t_{cf}
- Tiempo del primer cambio de dirección de la velocidad angular: t_{cva}
- Valor pico de la primera oscilación: V_p
- Máximo sobreimpulso: M_p
- Fuerza máxima aplicada: f_M
- Espacio recorrido por el carro a t_s del ángulo : X_{t_s}
- Velocidad angular en el primer punto donde el ángulo es cero : $W_{\Phi=0}$
- Angulo en el primer punto donde la velocidad angular es cero : $\Phi_{W=0}$
- Velocidad final con la que el carro queda moviéndose después de estabilizar al péndulo: V_f

Aplicando estos datos al programa se obtienen las siguientes curvas:

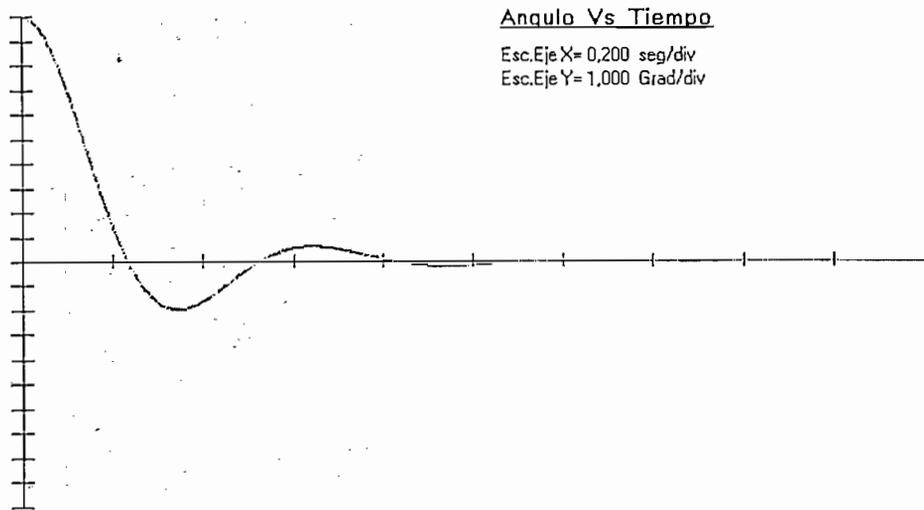


Fig.3.1.2 Gráfica del Angulo Vs. Tiempo por el Método de la Bisectriz

De los gráficos se obtienen algunos datos de interés empleando el cursor:

Datos de la Fig.3.1.2:

$$t_r = 0,236\text{seg}$$

$$t_p = 0,340\text{seg}$$

$$t_s = 0,788\text{seg}$$

$$V_p = - 1,941\text{grad}$$

$$M_p = 19,41\%$$

Como se puede ver, el sistema converge al punto de equilibrio, obteniéndose un máximo sobreimpulso inferior al ingresado como dato, esto se lo puede atribuir principalmente a que se tiene una raíz real no muy alejada de

las raíces complejas conjugadas (ver Fig.3.1.9) por lo que las raíces conjugadas no son completamente raíces dominantes.

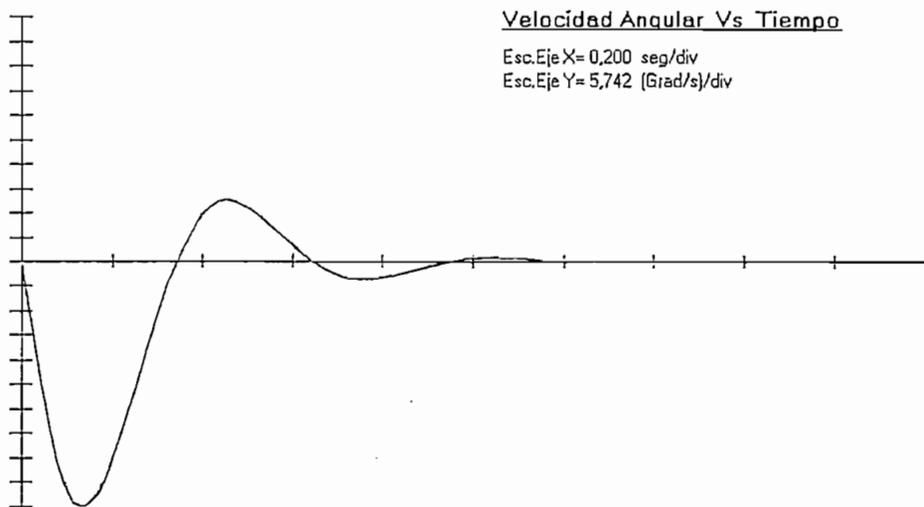


Fig.3.1.3 Gráfica de la Velocidad Angular Vs. Tiempo por el Método de la Bisectriz

Datos de la Fig.3.1.3:

$$t_{cva} = 0,343\text{seg}$$

$$t_p = 0,135\text{seg}$$

$$t_s = 1,118\text{seg}$$

$$V_p = - 57,42\text{grad/s}$$

Como se observa en la figura 3.1.3, el valor de la velocidad angular empieza siendo negativo, como se espera pues, el ángulo inicial del péndulo es positivo y para corregirlo la velocidad debe ser negativa y además, la velocidad angular final es cero, lo que confirma la estabilidad del sistema.

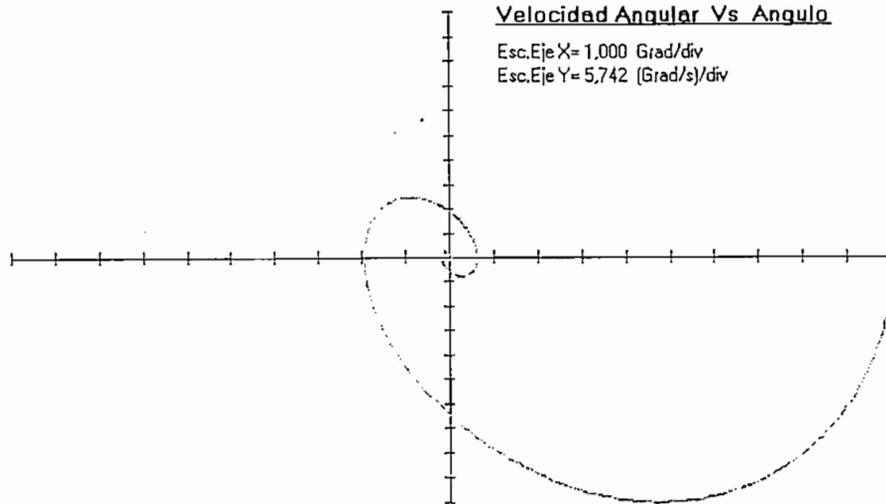


Fig.3.1.4 Gráfica de la Velocidad Angular Vs. Angulo por el Método de la Bisectriz

Datos de la Fig.3.1.4:

$$W_{\Phi=0} = - 37,384 \text{ grad/s}$$

$$\Phi_{W=0} = - 1,961 \text{ grad}$$

El sistema como se puede ver es asintóticamente estable, pues el ángulo y la velocidad angular convergen al punto de referencia cero.

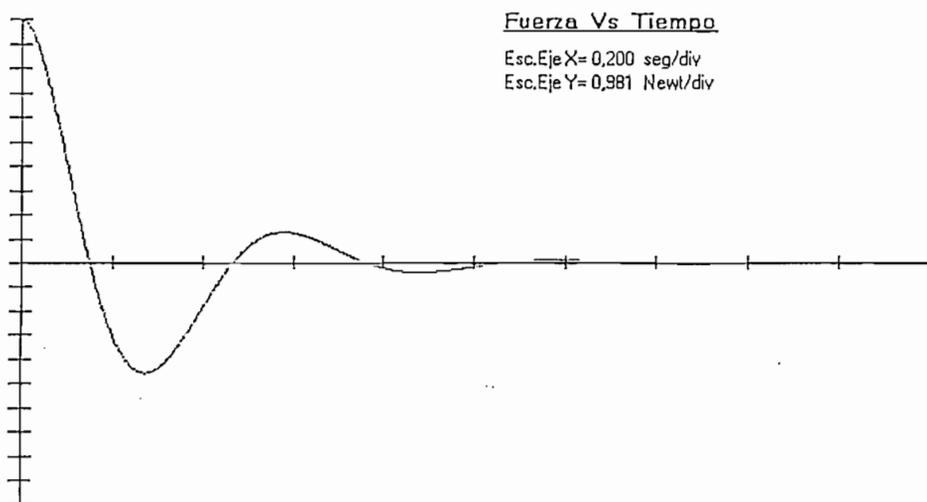


Fig.3.1.5 Gráfica de la Fuerza Vs. Tiempo por el Método de la Bisectriz

Datos de la Fig.3.1.5:

$$f_M = 9,81 \text{ N}$$

$$t_p = 0,27 \text{ seg}$$

$$t_s = 0,99 \text{ seg}$$

$$V_p = -4,506 \text{ N}$$

$$t_{cf} = 0,152 \text{ seg}$$

La forma de esta curva es similar a la del ángulo, lo cual es correcto pues, cuando el ángulo es positivo también lo es la fuerza y viceversa. La gráfica llega a cero lo que indica que se logra hacer cero el ángulo.

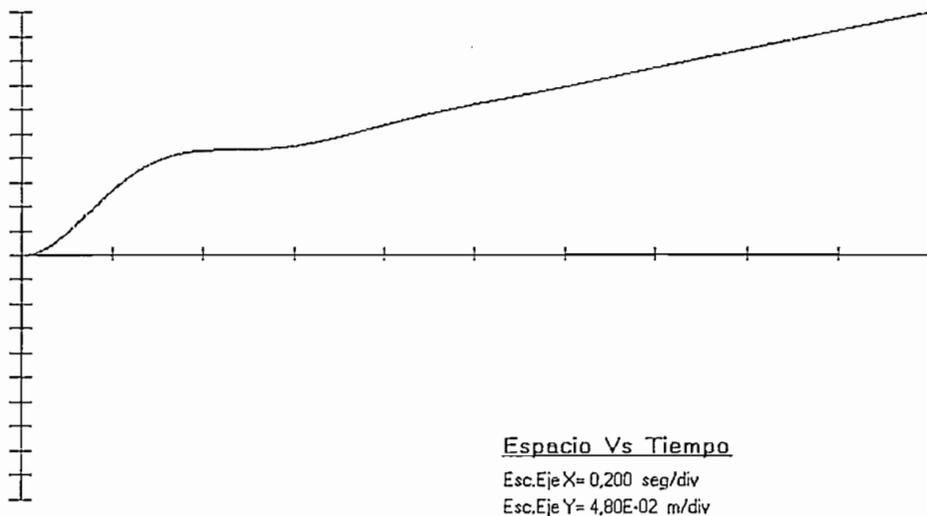


Fig.3.1.6 Gráfica del Espacio Vs. Tiempo por el Método de la Bisectriz

Datos de la Fig.3.1.6:

$$X_{t_s=0,788 \text{ seg}} = 0,252 \text{ m}$$

Como puede verse en la Fig.3.1.6 cuando el ángulo del péndulo a llegado a $t = 0,788\text{seg}$ la posición es 0.252m del centro de la pista, lo que indica que se logra estabilizar al sistema, antes que el carro sobrepase el limite de la pista (0.5m). Además, se ve que a pesar de haberse equilibrado el péndulo, el carro continua alejándose del origen con un movimiento rectilíneo uniforme (ver Fig.3.1.7.), llegando en algún momento a sobrepasar los limites de la pista.

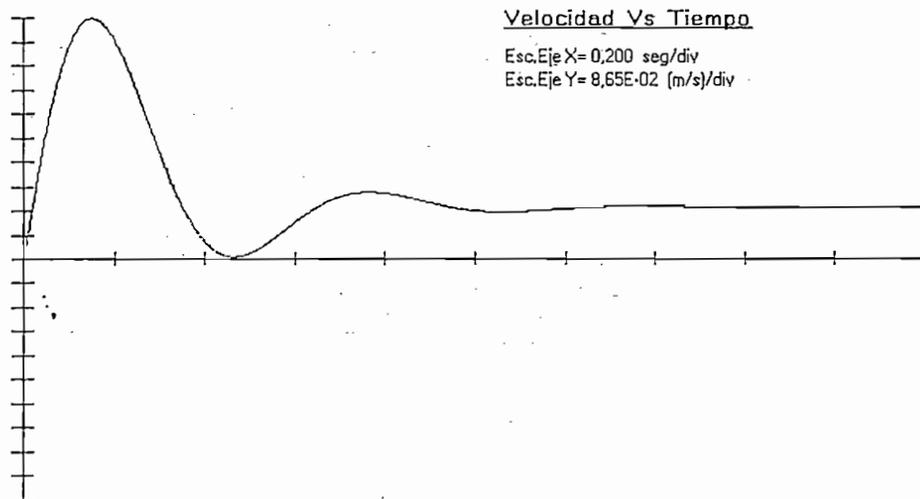


Fig.3.1.7 Gráfica de la Velocidad Vs. Tiempo por el Método de la Bisectriz

Datos de la Fig.3.1.7:

$$t_p = 0,152\text{seg}$$

$$V_p = 0,865\text{m/s}$$

$$V_f = 0,185\text{m/s}$$

La curva de la velocidad del carro (Fig.3.1.7), tiene un valor final de velocidad de 0,185m/s el cual, se mantiene constante aproximadamente desde que el péndulo alcanza su posición de equilibrio angular, ratificándose que el movimiento final del carro es rectilíneo uniforme.

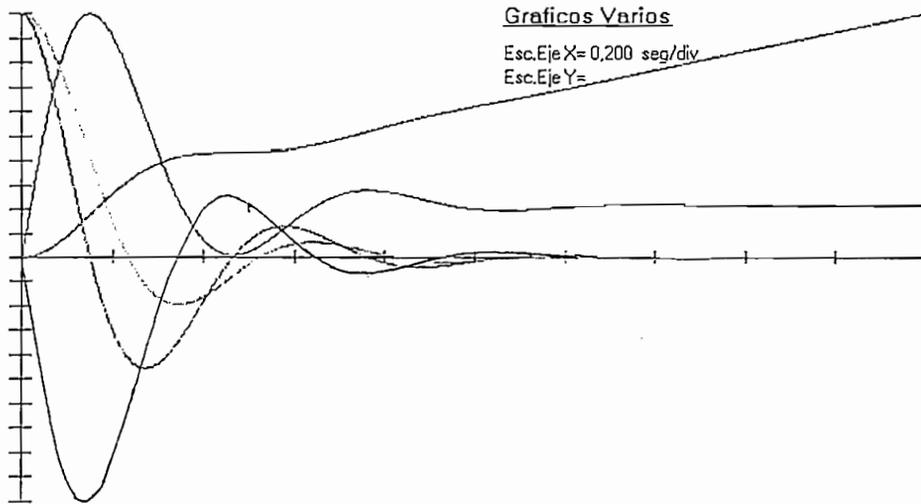


Fig.3.1.8 Gráfica de las cuatro variables y la Fuerza Vs. Tiempo por el Método de la Bisectriz

De la Fig.3.1.8 se puede ver que todas las curvas tienden a cero a excepción del espacio y la velocidad del carro.

El programa presenta información adicional en dos ventanas (ítem) de resultados como se ve en las Fig.3.1.9 y 3.1.10

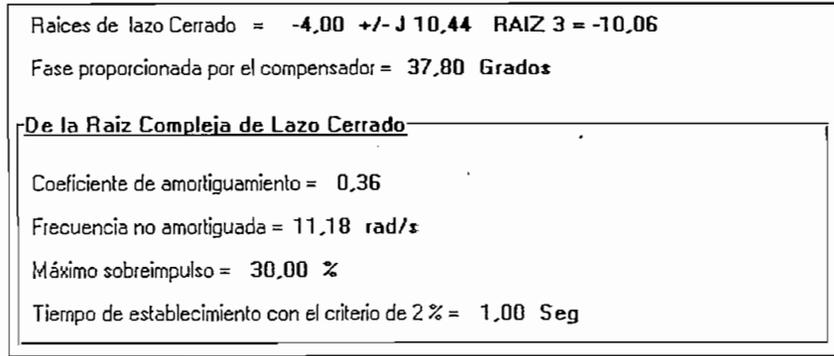


Fig.3.1.9 Resultados del Item 1 por el Método de la Bisectriz

$$G_c = K_c * (1 + C_c) / (1 + P_c)$$

$$K_c = 146,76$$

$$C_c = -6,92$$

$$P_c = -18,06$$

Fig.3.1.10 Resultados del Item 2 por el Método de la Bisectriz

Esta acción de control minimiza la ganancia del compensador K_c , pero no permite que las raíces conjugadas complejas sean dominantes para mejorar y predecir las características del sistema compensado. El método de la bisectriz también permite conocer alrededor de que valores se puede ubicar al cero del compensador, para la compensación con la técnica del cero arbitrario.

Análisis de Resultados con el Método del Cero Arbitrario

Utilizando el programa con los datos generales del péndulo y las condiciones iniciales iguales al caso anterior se realiza la compensación por este método:

Datos ingresados para este control:

$$M_p = 25\%$$

$$t_s = 1 \text{ seg}$$

$$C_c = -11$$

Las gráficas obtenidos del programa con estos datos se presentan a continuación.

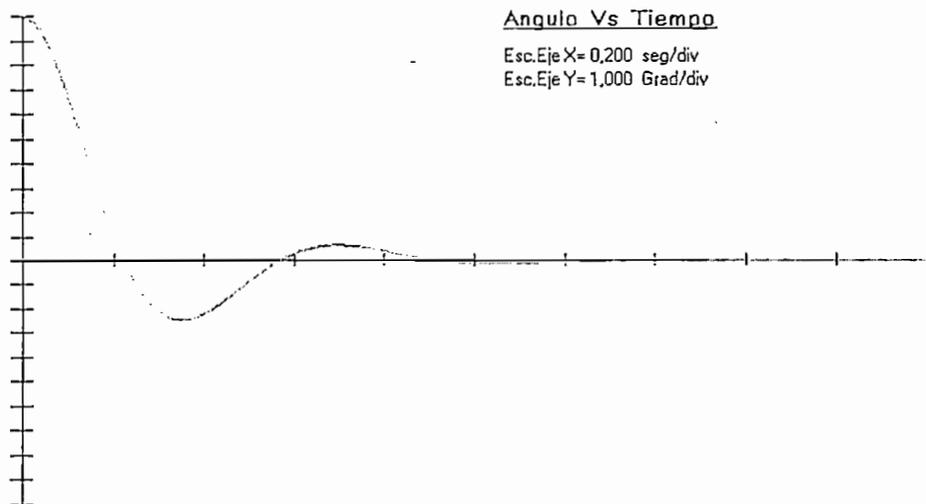


Fig.3.1.11 Gráfica del Angulo Vs. Tiempo por el Método del Cero Arbitrario

Datos de la Fig.3.1.11:

$$t_r = 0,222 \text{ seg}$$

$$t_p = 0,347 \text{ seg}$$

$$t_s = 0,871 \text{ seg}$$

$$V_p = -2,5 \text{ grad}$$

$$M_p = 25\%$$

Con este método se logra obtener el máximo sobreimpulso ingresado como dato, pues al reubicar el cero del compensador a una posición más adecuada se tiene mayor dominancia en la raíces complejas (ver Fig.3.1.18).

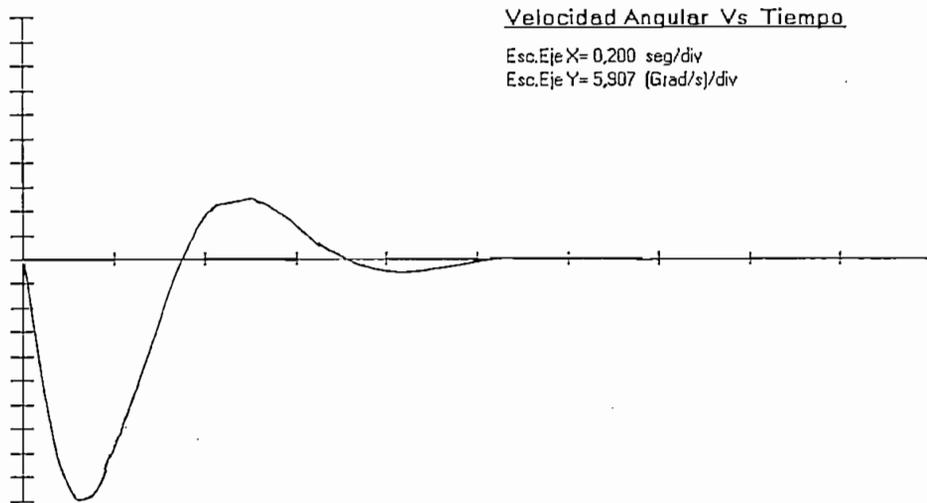


Fig.3.1.12 Gráfica de la Velocidad Angular Vs. Tiempo por el Método del Cero Arbitrario

Datos de la Fig.3.1.12:

$$t_p = 0,128\text{seg}$$

$$t_s = 0,976\text{seg}$$

$$V_p = - 59,07\text{grad/s}$$

$$t_{cva} = 0,347\text{seg}$$

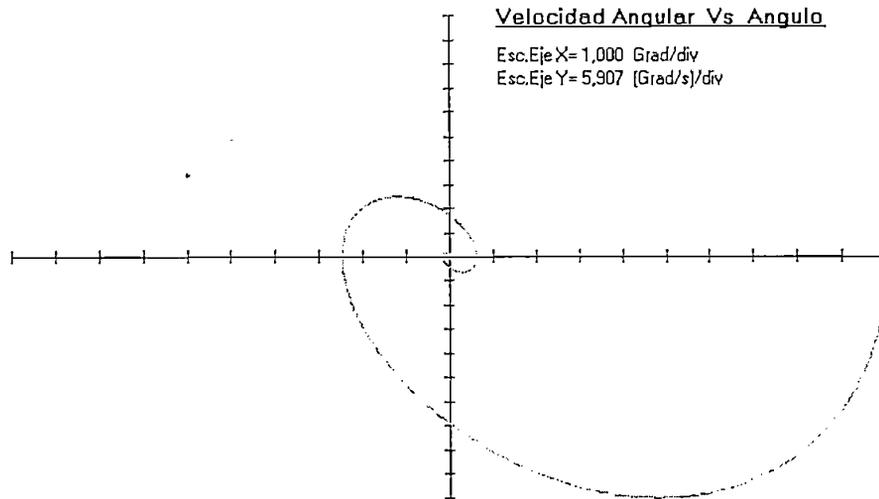


Fig.3.1.13 Gráfica de la Velocidad Angular Vs. Angulo por el Método del Cero Arbitrario

Datos de la Fig.3.1.13:

$$W_{\phi=0} = - 40,895 \text{ grad/s}$$

$$\Phi_{W=0} = - 2,472 \text{ grad}$$

En la Fig.3.1.13 se ve como el ángulo y la velocidad angular convergen a cero por lo que se puede decir que el sistema compensado es asintóticamente estable.

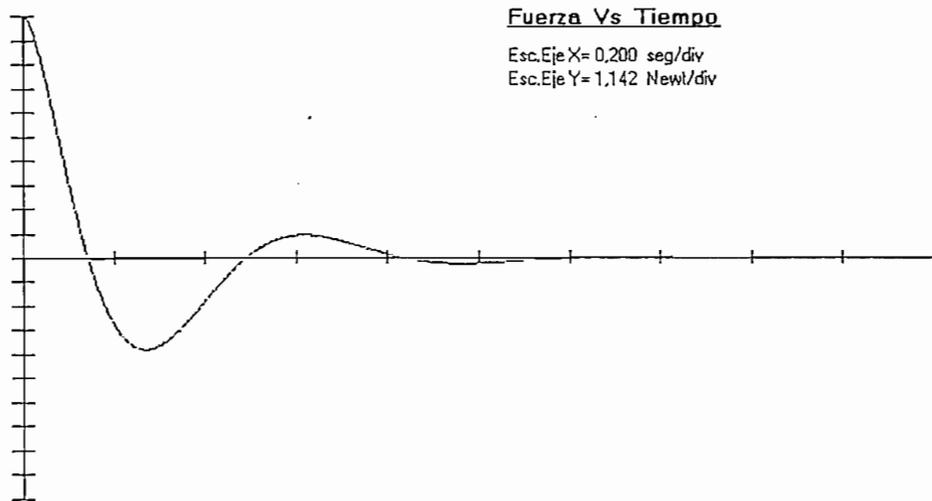


Fig.3.1.14 Gráfica de la Fuerza Vs. Tiempo por el Método del Cero Arbitrario

Datos de la Fig.3.1.14:

$$t_{cf} = 0,142\text{seg}$$

$$t_p = 0,27\text{seg}$$

$$t_s = 1,031\text{seg}$$

$$V_p = - 4,358\text{N}$$

$$f_M = 11,42\text{N}$$

Como se puede ver en la Fig.3.1.14. cuando el sistema alcanza la estabilidad angular la fuerza aplicada al carrito se hace cero.

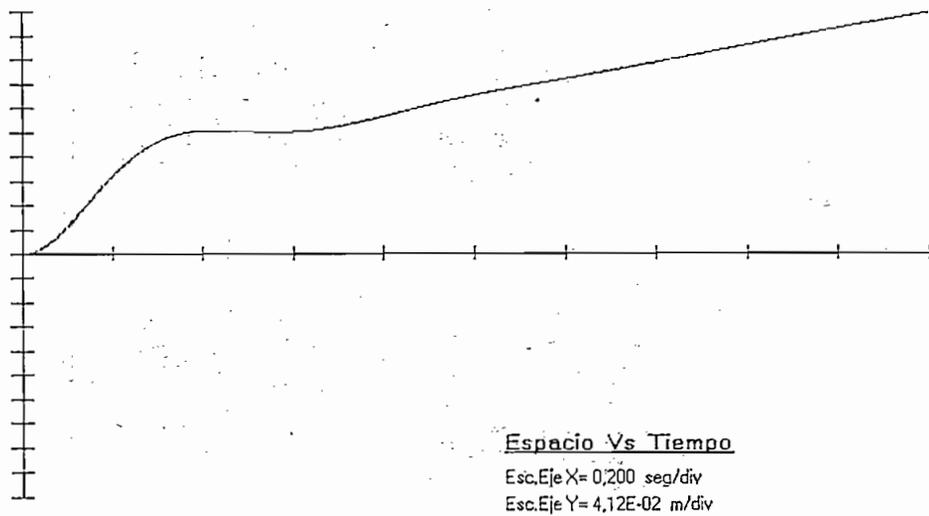


Fig.3.1.15 Gráfica del Espacio Vs. Tiempo por el Método del Cero Arbitrario

Datos de la Fig.3.1.15:

$$X_{ts=0,871\text{seg}} = 0,245\text{m}$$

Al igual que en el método de la bisectriz el carrito continúa moviéndose con velocidad constante y alejándose del origen a pesar que el ángulo haya sido ya corregido.

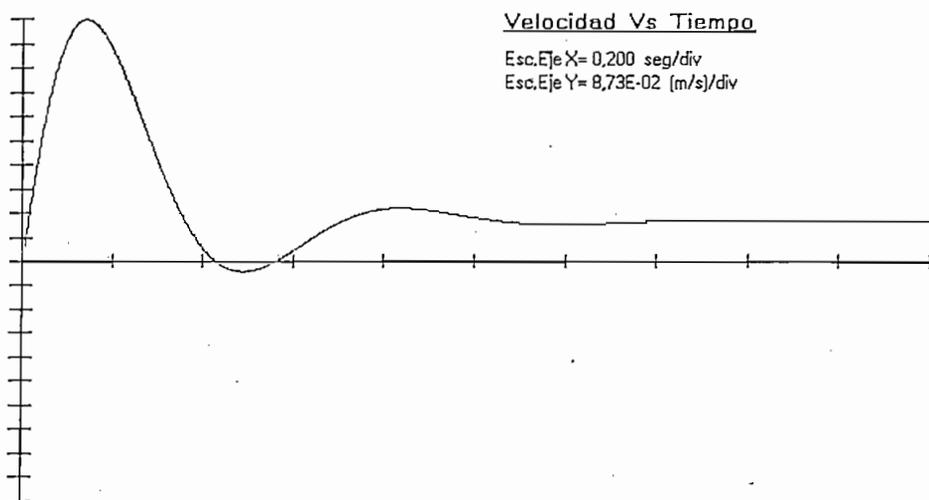


Fig.3.1.16 Gráfica de la Velocidad Vs. Tiempo por el Método del Cero Arbitrario

Datos de la Fig.3.1.16:

$$t_p = 0,145\text{seg}$$

$$V_p = 0,873\text{m/s}$$

$$V_f = 0,147\text{m/s}$$

$$t_{cvc} = 0,430\text{seg}$$

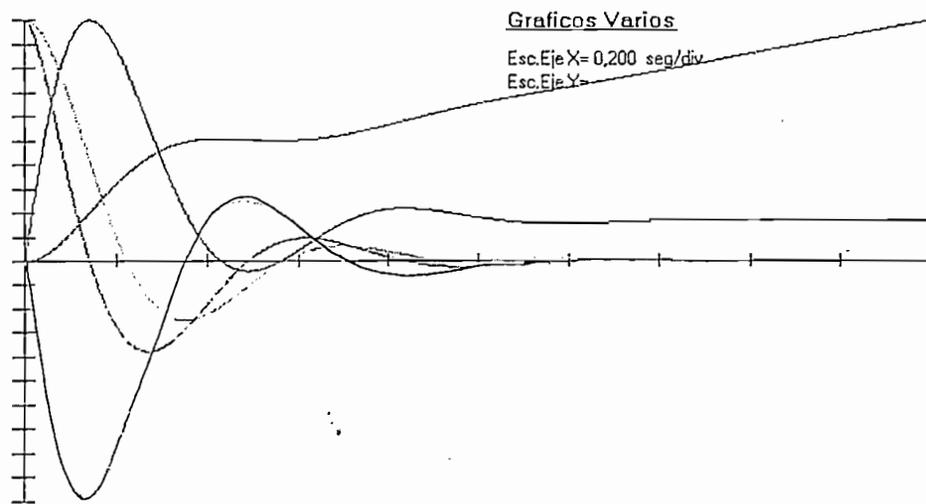


Fig.3.1.17 Gráfica de las cuatro variables y la Fuerza Vs. Tiempo por el Método del Cero Arbitrario

De la figura 3.1.17 se observa que todas las curvas tienden a cero a excepción del espacio y la velocidad de carro.

Al igual que en el caso anterior el programa para este control presenta dos ventanas (item) de resultados.

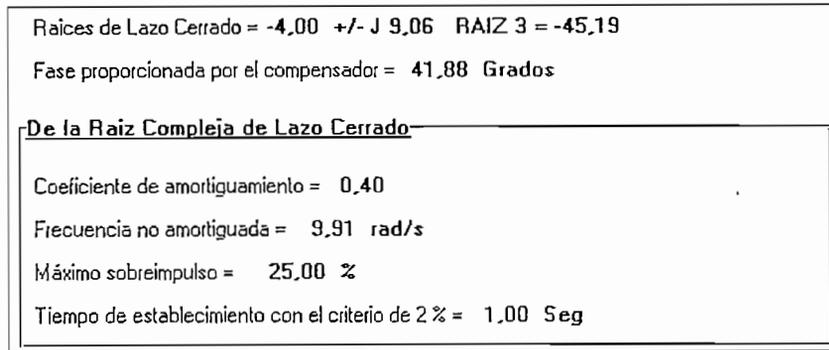


Fig.3.1.18 Resultados del Ítem 1 por el Método del Cero Arbitrario

Como se aprecia en la Fig. 3.1.18 las raíces complejas son las raíces dominantes, logrando con esto, cumplir con las especificaciones de diseño requeridas.

$G_c = K_c * (1 + C_c) / (1 + P_c)$ $K_c = 316,29$ $C_c = -11,00$ $P_c = -53,19$

Fig.3.1.19 Resultados del Ítem 2 por el Método del Cero Arbitrario

La red compensadora obliga a que el L.G.R. pase por la raíces deseadas, pero no determina la dominancia de las mismas.

Es importante que los polos deseados sean los polos dominantes con esto se consigue que el sistema se comporte más como uno de segundo orden que cumple con las especificaciones de diseño.

Las raíces dominantes son las más cercanas al Eje ($j\omega$), las cuales van a definir el comportamiento del sistema.

3.2.- Acciones de Control y Análisis de Resultados.

Las acciones de control que logran estabilizar al sistema son: el Control Proporcional Derivativo y el Control Proporcional Integral Derivativo las cuales se analizará a continuación.

a) Control Proporcional Derivativo

El diagrama de bloques se puede ver en la Fig.2.3.8 del capítulo anterior.

En este control la fuerza que se debe aplicar al carrito está definida por:

$$u(t) = Kp \left[\Phi(t) + Kd \frac{d\Phi(t)}{dt} \right] \quad (\text{Ec.3.2.1})$$

A partir de la Ec.3.1.21, la cual define la dinámica del sistema se tiene:

$$L' M \frac{\partial^2 \Phi(t)}{\partial t^2} - gM\Phi(t) = -u(t) \quad (\text{Ec.3.2.2})$$

reemplazando la Ec.3.2.1 en la Ec.3.2.2 se tiene:

$$L' M \frac{d^2 \Phi(t)}{dt^2} - gM\Phi(t) = -Kp \left[\Phi(t) + Kd \frac{d\Phi(t)}{dt} \right] \quad (\text{Ec.3.2.3})$$

aplicando Laplace a la ecuación anterior y tomando en cuenta C.I.≠0 queda:

$$L' M (s^2 \Phi(s) - s\Phi(0) - \dot{\Phi}(0)) - gM\Phi(s) = -Kp [\Phi(s) + Kd\Phi(s)s - Kd\Phi(0)]$$

donde:

$\Phi(0)$ = es la condición inicial del ángulo

$\dot{\Phi}(0)$ = es la condición inicial de la velocidad angular.

dividiendo para $L'M$ la ecuación anterior se obtiene:

$$s^2\Phi(s) - s\Phi(0) - \dot{\Phi}(0) - \frac{g}{L'}\Phi(s) = -\frac{Kp}{L'M}\Phi(s) - \frac{KpKd\Phi(s)s}{L'M} + \frac{KpKd\dot{\Phi}(0)}{L'M}$$

simplificando e igualando a cero:

$$s^2\Phi(s) - s\Phi(0) - \dot{\Phi}(0) + \frac{KpKd}{L'M}[s\Phi(s) - \Phi(0)] + \Phi(s)\left[-\frac{g}{L'} + \frac{Kp}{L'M}\right] = 0$$

haciendo:

$$a = \frac{KpKd}{L'M} \quad (\text{Ec.3.2.4})$$

$$b = \frac{Kp}{L'M} - \frac{g}{L'} \quad (\text{Ec.3.2.5})$$

reemplazando las Ecs.3.2.4 y 3.2.5 en la ecuación anterior se tiene:

$$s^2\Phi(s) - s\Phi(0) - \dot{\Phi}(0) + as\Phi(s) - a\Phi(0) + b\Phi(s) = 0$$

despejando el ángulo:

$$\Phi(s) = \frac{s\Phi(0) + \dot{\Phi}(0) + a\Phi(0)}{s^2 + as + b} \quad (\text{Ec.3.2.6})$$

comparando el denominador de la Ec.3.2.6 con un sistema de segundo orden subamortiguado se tiene la siguiente igualdad:

$$a = 2\zeta W_n \quad (\text{Ec.3.2.7})$$

$$b = W_n^2 \quad (\text{Ec.3.2.8})$$

igualando las Ecs.3.2.8 y 3.2.5 y despejando Kp se tiene:

$$Kp = \left(W_n^2 + \frac{g}{L'} \right) L' M \quad (\text{Ec.3.2.9})$$

igualando las Ecs.3.2.7 y 3.2.4 y despejando Kd se tiene:

$$Kd = \frac{2\zeta W_n L' M}{Kp} \quad (\text{Ec.3.2.10})$$

desarrollando la Ec.3.2.6 se obtiene:

$$\Phi(s) = \Phi(0) \frac{s}{s^2 + as + b} + \frac{\dot{\Phi}(0) + a\Phi(0)}{b} \frac{b}{s^2 + as + b} \quad (\text{Ec.3.2.11})$$

Se conoce que las transformadas inversas de Laplace son:

$$L^{-1} \left\{ \frac{s-b}{(s-b)^2 + a^2} \right\} = e^{bt} \cos at \quad (\text{Ec.3.2.12})$$

$$L^{-1} \left\{ \frac{1}{(s-b)^2 + a^2} \right\} = \frac{e^{bt} \operatorname{sen} at}{a} \quad (\text{Ec.3.2.13})$$

trabajando con el primer miembro de la Ec.3.2.11 se tiene:

$$\frac{s}{s^2 + as + b} = \frac{s}{s^2 + 2\zeta W_n s + W_n^2} = \frac{s}{(s + \zeta W_n)^2 - \zeta^2 W_n^2 + W_n^2} = \frac{s}{(s + \zeta W_n)^2 - W_n^2(\zeta^2 - 1)}$$

resolviendo la ecuación se obtiene:

$$\frac{s}{s^2 + as + b} = \frac{s + \zeta W_n}{(s + \zeta W_n)^2 + (1 - \zeta^2)W_n^2} - \frac{\zeta W_n}{(s + \zeta W_n)^2 + (1 - \zeta^2)W_n^2}$$

aplicando las ecuaciones Ec.3.2.12 y Ec.3.2.13 en la ecuación anterior se tiene:

$$L^{-1} \left\{ \frac{s + \zeta W_n}{(s + \zeta W_n)^2 + Wd^2} \right\} = e^{-\zeta W_n t} \cos Wd t$$

$$L^{-1} \left\{ \frac{\zeta W_n}{(s + \zeta W_n)^2 + Wd^2} \right\} = \frac{e^{-\zeta W_n t} \operatorname{sen} Wd t}{Wd} \zeta W_n$$

donde:

$$Wd = W_n \sqrt{1 - \zeta^2}$$

entonces queda:

$$L^{-1} \left\{ \frac{s}{s^2 + as + b} \right\} = e^{-\zeta W_n t} \cos[Wd t] - \frac{\zeta W_n e^{-\zeta W_n t} \operatorname{sen}[Wd t]}{Wd}$$

sacando factor común:

$$L^{-1} \left\{ \frac{s}{s^2 + as + b} \right\} = e^{-\zeta W_n t} \left[\cos[Wd t] - \frac{\zeta}{\sqrt{1 - \zeta^2}} \operatorname{sen}[Wd t] \right]$$

$$L^{-1} \left\{ \frac{s}{s^2 + as + b} \right\} = -\frac{e^{-\zeta W_n t}}{\sqrt{1 - \zeta^2}} \left[\zeta \operatorname{sen} Wd t - \sqrt{1 - \zeta^2} \cos Wd t \right] \quad (\text{Ec.3.2.14})$$

se sabe que:

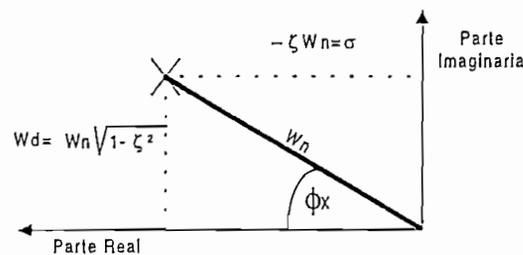


Fig.3.2.1.- Plano Complejo "s"

de donde:

$$\cos \phi_x = \zeta$$

$$\operatorname{sen} \phi_x = \sqrt{1 - \zeta^2}$$

reemplazando las ecuaciones anteriores en la Ec.3.2.14 se tiene:

$$L^{-1} \left\{ \frac{s}{s^2 + as + b} \right\} = -\frac{e^{-\zeta W_n t}}{\sqrt{1 - \zeta^2}} \left[\cos \phi_x \operatorname{sen} Wd t - \operatorname{sen} \phi_x \cos Wd t \right]$$

empleando la relación trigonométrica de suma y resta de ángulos:

$$\operatorname{sen}(A - B) = (\operatorname{sen} A \cos B - \cos A \operatorname{sen} B)$$

utilizando la ecuación anterior se obtiene:

$$L^{-1} \left\{ \frac{s}{s^2 + as + b} \right\} = -\frac{e^{-\zeta W_n t}}{\sqrt{1 - \zeta^2}} \left[\operatorname{sen}(Wd t - \phi_x) \right] \quad (\text{Ec.3.2.15})$$

Trabajando ahora con el segundo término de la Ec.3.2.11 y reemplazando las Ecs.3.2.7 y 3.2.8 se tiene:

$$\frac{b}{s^2 + as + b} = \frac{W_n^2}{s^2 + 2\zeta W_n s + W_n^2} = \frac{W_n^2}{(s + \zeta W_n)^2 - \zeta^2 W_n^2 + W_n^2} = \frac{W_n^2}{(s + \zeta W_n)^2 + W_n^2(1 - \zeta^2)}$$

aplicando la transformada inversa de Laplace:

$$L^{-1} \left\{ \frac{b}{s^2 + as + b} \right\} = \frac{W_n^2 e^{-\zeta W_n t} \operatorname{sen}(Wd t)}{W_n \sqrt{1 - \zeta^2}}$$

simplificando:

$$L^{-1} \left\{ \frac{b}{s^2 + as + b} \right\} = \frac{W_n e^{-\zeta W_n t} \operatorname{sen}(Wd t)}{\sqrt{1 - \zeta^2}} \quad (\text{Ec.3.2.16})$$

entonces reemplazando las Ecs.3.2.15 y 3.2.16 en la Ec.3.2.11 la ecuación del ángulo en el dominio del tiempo queda:

$$\Phi(t) = -\Phi(0) \frac{e^{-\zeta W_n t}}{\sqrt{1 - \zeta^2}} \left[\operatorname{sen}(Wd t - \Phi_x) \right] + \frac{\dot{\Phi}(0) + 2\zeta W_n \Phi(0)}{W_n^2} \cdot \frac{W_n e^{-\zeta W_n t} \operatorname{sen}(Wd t)}{\sqrt{1 - \zeta^2}}$$

simplificando:

$$\Phi(t) = \frac{e^{-\zeta W_n t}}{\sqrt{1 - \zeta^2}} \left[-\Phi(0) \operatorname{sen}(Wd t - \Phi_x) + \frac{[\dot{\Phi}(0) + 2\zeta W_n \Phi(0)]}{W_n} \operatorname{sen}(Wd t) \right] \quad (\text{Ec.3.2.17})$$

Para obtener la ecuación en el tiempo de la velocidad angular se deriva la Ec.3.2.17:

$$\begin{aligned} \dot{\Phi}(t) = & -\frac{\Phi(0)}{\sqrt{1-\zeta^2}} \left[-\zeta W_n e^{-\zeta W_n t} \operatorname{sen}(\sqrt{1-\zeta^2} W_n t - \Phi_x) + e^{-\zeta W_n t} \cos(\sqrt{1-\zeta^2} W_n t - \Phi_x) (\sqrt{1-\zeta^2} W_n) \right] + \dots \\ & \dots + \frac{(\dot{\Phi}(0) + 2\zeta W_n \Phi(0))}{W_n \sqrt{1-\zeta^2}} \left\{ (-\zeta W_n) e^{-\zeta W_n t} \operatorname{sen}(\sqrt{1-\zeta^2} W_n t) + e^{-\zeta W_n t} \cos(\sqrt{1-\zeta^2} W_n t) (\sqrt{1-\zeta^2} W_n) \right\} \end{aligned}$$

resolviendo:

$$\begin{aligned} \dot{\Phi}(t) = & \frac{\Phi(0)}{\sqrt{1-\zeta^2}} W_n e^{-\zeta W_n t} \left[\zeta \operatorname{sen}(\sqrt{1-\zeta^2} W_n t - \Phi_x) - \cos(\sqrt{1-\zeta^2} W_n t - \Phi_x) (\sqrt{1-\zeta^2}) \right] \dots \\ & - \frac{(\dot{\Phi}(0) + 2\zeta W_n \Phi(0))}{W_n \sqrt{1-\zeta^2}} e^{-\zeta W_n t} W_n \left\{ \zeta \operatorname{sen}(\sqrt{1-\zeta^2} W_n t) - \cos(\sqrt{1-\zeta^2} W_n t) (\sqrt{1-\zeta^2}) \right\} \end{aligned}$$

se denomina:

$$\begin{aligned} g &= \frac{\Phi(0) e^{-\zeta W_n t} W_n}{\sqrt{1-\zeta^2}} \\ h &= \frac{(\dot{\Phi}(0) + 2\zeta W_n \Phi(0)) e^{-\zeta W_n t}}{\sqrt{1-\zeta^2}} \end{aligned}$$

reemplazando las ecuaciones que se deducen de la Fig.3.2.1. tenemos:

$$\begin{aligned} \dot{\Phi}(t) = & g \left[\cos \Phi_x \operatorname{sen}(\sqrt{1-\zeta^2} W_n t - \Phi_x) - \cos(\sqrt{1-\zeta^2} W_n t - \Phi_x) \operatorname{sen} \Phi_x \right] \dots \\ & - h \left\{ \cos \Phi_x \operatorname{sen}(\sqrt{1-\zeta^2} W_n t) - \operatorname{sen} \Phi_x \cos(\sqrt{1-\zeta^2} W_n t) \right\} \end{aligned}$$

aplicando la relación trigonométrica de suma y resta de ángulos se tiene:

$$\dot{\Phi}(t) = g \left[\operatorname{sen}(\sqrt{1-\zeta^2} W_n t - 2\Phi_x) \right] - h \left[\operatorname{sen}(\sqrt{1-\zeta^2} W_n t - \Phi_x) \right]$$

reemplazando g y h :

$$\dot{\Phi}(t) = \frac{e^{-\zeta W_n t} W_n}{\sqrt{1-\zeta^2}} \left\{ \Phi(0) \operatorname{sen}(Wd t - 2\Phi_x) - \frac{(\dot{\Phi}(0) + 2\zeta W_n \Phi(0))}{W_n} \operatorname{sen}(Wd t - \Phi_x) \right\} \quad (\text{Ec.3.2.18})$$

Para encontrar la ecuación en el tiempo del espacio y la velocidad del carrito se trabaja con la Ec.3.2.3

$$L' M \frac{d^2 \Phi(t)}{dt^2} - gM\Phi(t) = -Kp \left[\Phi(t) + Kd \frac{d\Phi(t)}{dt} \right]$$

despejando la segunda derivada del ángulo:

$$\ddot{\Phi}(t) = -\Phi(t) \left[\frac{Kp}{L'M} - \frac{g}{L'} \right] - \dot{\Phi}(t) \frac{KpKd}{L'M}$$

reemplazando las Ecs.3.2.4. y 3.2.5 en la ecuación anterior se tiene:

$$\ddot{\Phi}(t) = -b\Phi(t) - a\dot{\Phi}(t)$$

integrando una vez:

$$\int \ddot{\Phi}(t) = -b \int \Phi(t) - a \int \dot{\Phi}(t)$$

$$\dot{\Phi}(t) = -b \int \Phi(t) - a\Phi(t)$$

despejando:

$$\int \Phi(t) = -\frac{a}{b} \Phi(t) - \frac{\dot{\Phi}(t)}{b} \quad (\text{Ec.3.2.19})$$

reemplazando la Ec.2.2.11 en la Ec.3.2.1 se tiene:

$$M\ddot{x}(t) = Kp\Phi(t) + KpKd\dot{\Phi}(t)$$

integrando una vez y despejando la velocidad, queda:

$$\dot{x}(t) = \frac{Kp}{M} \int \Phi(t) + \frac{KpKd}{M} \dot{\Phi}(t) + C_1 \quad (\text{Ec.3.2.20})$$

reemplazando la Ec.3.2.19 en la Ec.3.2.20 se obtiene:

$$\dot{x}(t) = \frac{Kp}{M} \left[-\frac{a}{b} \Phi(t) - \frac{\dot{\Phi}(t)}{b} \right] + \frac{KpKd}{M} \dot{\Phi}(t) + C_1$$

entonces la velocidad del carrito queda:

$$v(t) = \dot{x}(t) = \frac{1}{M} \left\{ Kp \left(Kd - \frac{a}{b} \right) \Phi(t) - \frac{Kp}{b} \dot{\Phi}(t) \right\} + C_1 \quad (\text{Ec.3.2.21})$$

Se integra por segunda vez la ecuación anterior y se reemplaza la Ec.3.2.19 para obtener la ecuación del espacio:

$$x(t) = \frac{Kp}{M} \left(Kd - \frac{a}{b} \right) \left[-\frac{a}{b} \Phi(t) - \frac{\dot{\Phi}(t)}{b} \right] - \frac{Kp}{bM} \Phi(t) + C_1 t + C_2$$

resolviendo:

$$x(t) = \frac{1}{Mb} \left[Kp \left(\frac{a^2}{b} - a Kd - 1 \right) \Phi(t) - Kp \left(Kd - \frac{a}{b} \right) \dot{\Phi}(t) \right] + C_1 t + C_2 \quad (\text{Ec.3.2.22})$$

Las constantes de integración C_1 y C_2 se obtienen, evaluando las Ecs.3.2.21 y 3.2.22 en $t=0$:

$$v(0) = \frac{1}{M} \left\{ Kp \left(Kd - \frac{a}{b} \right) \Phi(0) - \frac{Kp}{b} \dot{\Phi}(0) \right\} + C_1$$

entonces:

$$C_1 = v(0) - \frac{1}{M} \left\{ Kp \left(Kd - \frac{a}{b} \right) \Phi(0) - \frac{Kp}{b} \dot{\Phi}(0) \right\} \quad (\text{Ec.3.2.23})$$

y de la Ec.3.2.22

$$x(0) = \frac{1}{Mb} \left[\left(\frac{a^2}{b} Kp - a Kp Kd - Kp \right) \Phi(0) - Kp \left(Kd - \frac{a}{b} \right) \dot{\Phi}(0) \right] + C_1(0) + C_2$$

entonces:

$$C_2 = x(0) - \frac{1}{Mb} \left[\left(\frac{a^2}{b} Kp - a Kp Kd - Kp \right) \Phi(0) - Kp \left(Kd - \frac{a}{b} \right) \dot{\Phi}(0) \right] \quad (\text{Ec.3.2.24})$$

En conclusión, las Ecs.3.2.17; 3.2.18; 3.2.21 y 3.2.22 representan las ecuaciones en el tiempo de las variables ángulo, velocidad angular, velocidad del carrito y espacio del carrito respectivamente.

$$\Phi(t) = \frac{e^{-\zeta W_n t}}{\sqrt{1-\zeta^2}} \left[-\Phi(0) \operatorname{sen}(W_d t - \Phi_x) + \frac{[\dot{\Phi}(0) + 2\zeta W_n \Phi(0)]}{W_n} \operatorname{sen}(W_d t) \right] \quad (\text{Ec.3.2.17})$$

$$\dot{\Phi}(t) = \frac{e^{-\zeta W_n t} W_n}{\sqrt{1-\zeta^2}} \left\{ \Phi(0) \operatorname{sen}(W_d t - 2\Phi_x) - \frac{(\dot{\Phi}(0) + 2\zeta W_n \Phi(0))}{W_n} \operatorname{sen}(W_d t - \Phi_x) \right\} \quad (\text{Ec.3.2.18})$$

$$\dot{v}(t) = \dot{x}(t) = \frac{1}{M} \left\{ Kp \left(Kd - \frac{a}{b} \right) \Phi(t) - \frac{Kp}{b} \dot{\Phi}(t) \right\} + C_1 \quad (\text{Ec.3.2.21})$$

$$x(t) = \frac{1}{Mb} \left[Kp \left(\frac{a^2}{b} - a Kd - 1 \right) \Phi(t) - Kp \left(Kd - \frac{a}{b} \right) \dot{\Phi}(t) \right] + C_1 t + C_2 \quad (\text{Ec.3.2.22})$$

b) Control Proporcional Integral Derivativo.

El diagrama de bloques está representado en la Fig.2.3.11, donde la acción de control (fuerza) que se aplica al carrito es igual a :

$$u(t) = Kp \left(\Phi(t) + Kd \dot{\Phi}(t) + K_i \int \Phi(t) \right) \quad (\text{Ec.3.2.25})$$

de la Ec.3.1.21 de la dinámica de la planta :

$$L' M \frac{\partial^2 \Phi(t)}{\partial t^2} - g M \Phi(t) = -u(t)$$

igualando la Ec.3.2.25 con la ecuación anterior y pasando al dominio de 's' se tiene:

$$L' Ms^2 [\Phi(s)] - L' Ms\Phi(0) - L' M\dot{\Phi}(0) - gM\Phi(s) = \dots$$

$$\dots - Kp\Phi(s) - KpKd\Phi(s)s + KpKd\Phi(0) - KpK_I \frac{\Phi(s)}{s}$$

simplificando:

$$\Phi(s) \left[L' Ms^2 - gM + Kp + KpKds + \frac{KpK_I}{s} \right] = s\Phi(0)L' M + L' M\dot{\Phi}(0) + KpKd\Phi(0)$$

$$\Phi(s) [L' Ms^3 - gMs + Kps + KpKds^2 + KpK_I] = s^2\Phi(0)L' M + s(L' M\dot{\Phi}(0) + KpKd\Phi(0))$$

despejando el ángulo:

$$\Phi(s) = \frac{s^2\Phi(0) + s[\dot{\Phi}(0) + KpKd\Phi(0) / L' M]}{s^3 + s^2 \frac{KpKd}{L' M} + s \frac{(Kp - gM)}{L' M} + \frac{KpK_I}{L' M}} \quad (\text{Ec.3.2.26})$$

haciendo:

$$Fb_2 = \Phi(0)$$

$$Fb_1 = \dot{\Phi}(0) + \frac{KpKd\Phi(0)}{L' M}$$

Analizando el denominador de la función de transferencia de lazo cerrado Ec.3.2.26 e igualando con un sistema de tercer orden de la siguiente forma:

$$s^3 + s^2 \frac{KpKd}{L' M} + s \frac{Kp - gM}{L' M} + \frac{KpK_I}{L' M} = [(s^2 + 2s\sigma + \sigma^2 + wd^2)(s + y)]$$

donde, 'y' representa una raíz real del sistema del lazo cerrado ubicada en el semiplano izquierdo.

Resolviendo la ecuación anterior:

$$s^3 + s^2 \frac{KpKd}{L' M} + s \frac{Kp - gM}{L' M} + \frac{KpK_I}{L' M} = s^3 + s^2 y + 2s^2 \sigma + 2s\sigma y + (\sigma^2 + wd^2)s + (\sigma^2 + wd^2)y$$

$$s^3 + s^2 \frac{KpKd}{L' M} + s \frac{Kp - gM}{L' M} + \frac{KpK_I}{L' M} = s^3 + s^2 (y + 2\sigma) + s(2\sigma y + \sigma^2 + wd^2) + (\sigma^2 + wd^2)y$$

igualando términos semejantes se tiene:

$$a = \frac{KpKd}{L'M} = y + 2\sigma$$

$$b = \frac{Kp}{L'M} - \frac{g}{L'} = 2\sigma y + \sigma^2 + wd^2$$

$$c = \frac{KpK_l}{L'M} = (\sigma^2 + wd^2) y$$

como:

$$\sigma^2 + wd^2 = \zeta^2 w_n^2 + w_n^2(1 - \zeta^2) = w_n^2$$

$$\sigma = \zeta Wn$$

reemplazando se obtiene:

$$a = y + 2\zeta Wn \quad (\text{Ec.3.2.27})$$

$$b = 2\zeta Wn y + Wn^2 \quad (\text{Ec.3.2.28})$$

$$c = Wn^2 y \quad (\text{Ec.3.2.29})$$

de la Ec.3.2.27 se tiene:

$$Wn = \frac{a - y}{2\zeta} \quad (\text{Ec.3.2.30})$$

y de la Ec.3.2.29 se tiene:

$$Wn^2 = \frac{c}{y} \quad (\text{Ec.3.2.31})$$

reemplazando las Ecs.3.2.30 y 3.2.31 en la Ec.3.2.28 se llega:

$$y^3 - a y^2 + b y - c = 0$$

resolviendo la ecuación de tercer orden se obtiene el valor de 'y', el cual debe ser positivo y real para que el sistema sea estable, y además debe cumplir que ζ tenga un valor mayor que 0 y menor que 1 ($0 < \zeta < 1$).

Los valores a , b y c son conocidos, pues las constantes K_p , K_d , K , son valores que se ingresan en el programa para este control.

Luego de la Ec.3.2.29 y la Ec.3.2.27 se encuentra la Frecuencia Natural no Amortiguada (W_n) y el Coeficiente de Amortiguamiento (ζ):

$$W_n = \sqrt{\frac{c}{y}}$$

$$\zeta = \frac{a - y}{2W_n}$$

resolviendo por fracciones parciales la Ec.3.2.26. resulta:

$$\Phi(s) = \frac{A}{s + y} + \frac{B(s + \sigma)}{(s + \sigma)^2 + Wd^2} + \frac{C Wd}{(s + \sigma)^2 + Wd^2}$$

sacando la transformada inversa de Laplace se tiene:

$$\Phi(t) = Ae^{-y't} + e^{-\sigma t} [B \cos Wd t + C \sen Wd t] \quad (\text{Ec.3.2.32})$$

donde:

$$A = \frac{\Phi(0)y^2 - \left(\dot{\Phi}(0) + \frac{K_p K_d \Phi(0)}{L' M} \right) y}{(-y + \sigma)^2 + Wd^2}$$

$$B = \Phi(0) - A$$

$$C = \frac{\dot{\Phi}(0) + \frac{K_p K_d \Phi(0)}{L' M} - 2\sigma A - B(\sigma + y)}{Wd}$$

derivando la Ec.3.2.32 se obtiene la ecuación de la velocidad angular:

$$\dot{\Phi}(t) = A_1 e^{-y't} + e^{-\sigma t} [B_1 \cos Wd t + C_1 \sen Wd t] \quad (\text{Ec.3.2.33})$$

donde:

$$A_1 = -Ay$$

$$B_1 = -\sigma B + CWd$$

$$C_1 = -Wd B - \sigma C$$

derivando por segunda vez la Ec.3.2.32 se obtiene la aceleración angular que es necesario calcular, para conocer la fuerza, la posición y velocidad del carrito.

$$\ddot{\Phi}(t) = A_2 e^{-\gamma t} + e^{-\sigma t} [B_2 \cos Wdt + C_2 \sin Wdt]$$

donde:

$$A_2 = -A_1 \gamma$$

$$B_2 = -\sigma B_1 + C_1 Wd$$

$$C_2 = -Wd B_1 - \sigma C_1$$

entonces la fuerza se obtiene de la Ec.3.2.2 que es igual a:

$$\therefore u(t) = -L' M \ddot{\Phi}(t) + gM\Phi(t)$$

e igualando la ecuación anterior con la Ec.3.2.25 y despejando la integral del ángulo se tiene:

$$\int \Phi(t) = \frac{L' M \ddot{\Phi}(t) - gM\Phi(t) + Kp\dot{\Phi}(t) + KpKd\dot{\Phi}(t)}{-KpK_1}$$

$$\int \Phi(t) = \ddot{\Phi}(t) \left[\frac{-L' M}{KpK_1} \right] + \dot{\Phi}(t) \left[\frac{-Kd}{K_1} \right] + \Phi(t) \left[\frac{gM}{KpK_1} - \frac{1}{K_1} \right]$$

se denomina:

$$f = -\frac{L' M}{KpK_1}$$

$$d = -\frac{Kd}{K_1}$$

$$h = \left[\frac{gM}{KpK_1} - \frac{1}{K_1} \right]$$

se tiene entonces:

$$\int \ddot{\Phi}(t) = \ddot{\Phi}(t)f + \dot{\Phi}(t)d + \Phi(t)h \quad (\text{Ec.3.2.34})$$

Para el cálculo de la posición y la velocidad del carrito se trabaja con la Ec.3.2.2 la cual al igualarla con la Ec.2.2.11 queda:

$$M\ddot{x}(t) = -L'M\ddot{\Phi}(t) + gM\dot{\Phi}(t) \quad (\text{Ec.3.2.35})$$

integrando una vez la Ec.3.2.35 y reemplazando la Ec.3.2.34 se tiene la velocidad:

$$\begin{aligned} \dot{x}(t) &= \ddot{\Phi}(t)(gf) + \dot{\Phi}(t)[-L'+gd] + \Phi(t)(gh) + C_1 \\ v(t) &= v_{A1}\ddot{\Phi}(t) + v_{A2}\dot{\Phi}(t) + v_{A3}\Phi(t) + C_1 \end{aligned} \quad (\text{Ec.3.2.36})$$

donde:

$$\begin{aligned} v_{A1} &= gf = -g \frac{L'M}{KpK_1} \\ v_{A2} &= -L'+gd = -L'-g \frac{Kd}{K_1} \\ v_{A3} &= gh = g \left[\frac{gM}{KpK_1} - \frac{1}{K_1} \right] \end{aligned}$$

integrando por segunda vez se obtiene la posición:

$$\begin{aligned} \int v(t) &= v_{A1} \int \ddot{\Phi}(t) + v_{A2} \int \dot{\Phi}(t) + v_{A3} \int \Phi(t) + \int C_1 \\ x(t) &= x_{A1}\ddot{\Phi}(t) + x_{A2}\dot{\Phi}(t) + x_{A3}\Phi(t) + C_1t + C_2 \end{aligned} \quad (\text{Ec.3.2.37})$$

donde:

$$x_{A1} = v_{A3}f = -v_{A3} \frac{L'M}{KpK_1}$$

$$x_{A2} = v_{A1} + v_{A3}d = v_{A1} - v_{A3} \frac{Kd}{K_1}$$

$$x_{A3} = v_{A2} + v_{A3}h = v_{A2} + v_{A3} \left[\frac{gM}{KpK_1} - \frac{1}{K_1} \right]$$

Para encontrar el valor de las constantes de integración (C_1 y C_2) se evalúa en $t=0$ las Ecs.3.2.36 y 3.2.37:

$$C_1 = v(0) - v_{A1}\ddot{\Phi}(0) - v_{A2}\dot{\Phi}(0) - v_{A3}\Phi(0)$$

$$C_2 = x(0) - x_{A1}\ddot{\Phi}(0) - x_{A2}\dot{\Phi}(0) - x_{A3}\Phi(0)$$

Como ya se mencionó, la constantes Kp , Kd , y Ki , son ingresadas en el programa como datos, quedando de esta manera definidas todas la ecuaciones en el tiempo.

En conclusión las Ecs.3.2.32; 3.2.33; 3.2.36; y 3.4.37 representan las variables de ángulo, velocidad angular, velocidad del carrito y posición del carrito respectivamente.

$$\Phi(t) = Ae^{-\gamma t} + e^{-\sigma t} [B \cos Wd t + C \text{sen} Wd t] \quad (\text{Ec.3.2.32})$$

$$\dot{\Phi}(t) = A_1 e^{-\gamma t} + e^{-\sigma t} [B_1 \cos Wd t + C_1 \text{sen} Wd t] \quad (\text{Ec.3.2.33})$$

$$v(t) = v_{A1}\ddot{\Phi}(t) + v_{A2}\dot{\Phi}(t) + v_{A3}\Phi(t) + C_1 \quad (\text{Ec.3.2.36})$$

$$x(t) = x_{A1}\ddot{\Phi}(t) + x_{A2}\dot{\Phi}(t) + x_{A3}\Phi(t) + C_1 t + C_2 \quad (\text{Ec.3.2.37})$$

Análisis de Resultados del Control Proporcional Derivativo

En este tipo de control, el programa permite ingresar dos opciones de datos: en la primera se puede ingresar la frecuencia no amortiguada (ω_n) y el Coeficiente de amortiguamiento (ζ), mientras que en la segunda opción se ingresan la constantes K_p y K_d .

Con los mismos datos generales y condiciones iniciales ingresados como ejemplo en el control de compensación por redes se tiene:

Datos del péndulo

$$M = 1\text{Kg}$$

$$m = 0,1\text{Kg}$$

$$L = 0,5\text{m}$$

$$g = 9,8\text{m/s}^2$$

$$\text{Longitud de pista} = 1\text{m}$$

Condiciones iniciales

$$\text{Espacio} = 0\text{m}$$

$$\text{Velocidad} = 0\text{m/s}$$

$$\text{Ángulo} = 10\text{grad}$$

$$\text{Vel. Angular} = 0\text{grad/s}$$

Datos ingresados para el Control P.D.

$$\omega_n = 8\text{rad/s}$$

$$\zeta = 0,4$$

A continuación se presenta las mismas definiciones para el análisis de los gráficos:

- Tiempo de crecimiento: t_r
- Tiempo en alcanzar el primer pico : t_p
- Tiempo de establecimiento criterio del 2 % : t_s
- Tiempo del primer cambio de dirección de la velocidad del carro: t_{cvc}

- Tiempo del primer cambio de dirección de la fuerza: t_{cf}
- Tiempo del primer cambio de dirección de la velocidad angular: t_{cva}
- Valor pico de la primera oscilación: V_p
- Máximo sobreimpulso: M_p
- Fuerza máxima aplicada: f_M
- Espacio recorrido por el carro a t_s del ángulo : X_{1s}
- Velocidad angular en el primer punto donde el ángulo es cero : $W_{\Phi=0}$
- Angulo en el primer punto donde la velocidad angular es cero : $\Phi_{W=0}$
- Velocidad final con la que el carro queda moviéndose después de estabilizar al péndulo: V_f

Las gráficas obtenidas del programa son:

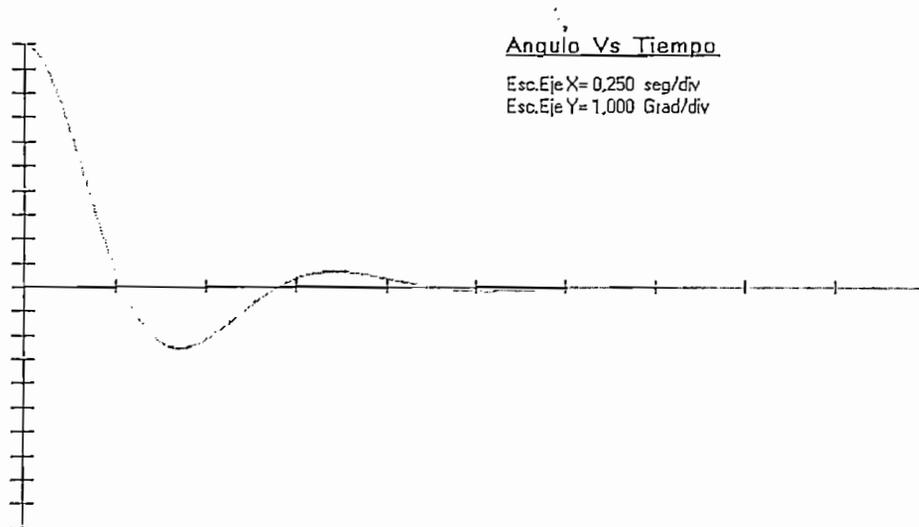


Fig.3.2.2.- Gráfica del Angulo Vs. Tiempo para el Control PD

Datos de la Fig.3.2.2:

$$t_r = 0,265\text{seg}$$

$$t_p = 0,426\text{seg}$$

$$t_s = 1,289\text{seg}$$

$$V_p = -2,538\text{grad}$$

$$M_p = 25,38\%$$

Con este control se logra obtener el máximo sobreimpulso ingresado como dato, porque, sólo existen dos raíces conjugadas de lazo cerrado que determinan el comportamiento del sistema (ver Fig.3.2.9)

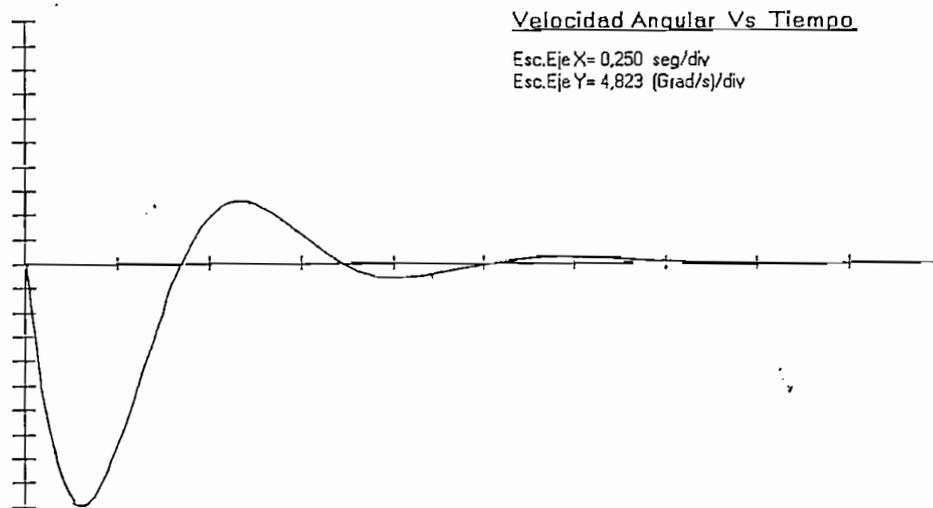


Fig.3.2.3.- Gráfica de la Velocidad Angular Vs. Tiempo para el Control PD

Datos de la Fig.3.2.3:

$$t_p = 0,161\text{seg}$$

$$t_s = 1,198\text{seg}$$

$$V_p = - 48,23\text{grad/s}$$

$$t_{cva} = 0,431\text{seg}$$

Al igual que en los otros métodos la velocidad angular llega a cero, lo cual indica que el péndulo llega a estabilizarse..

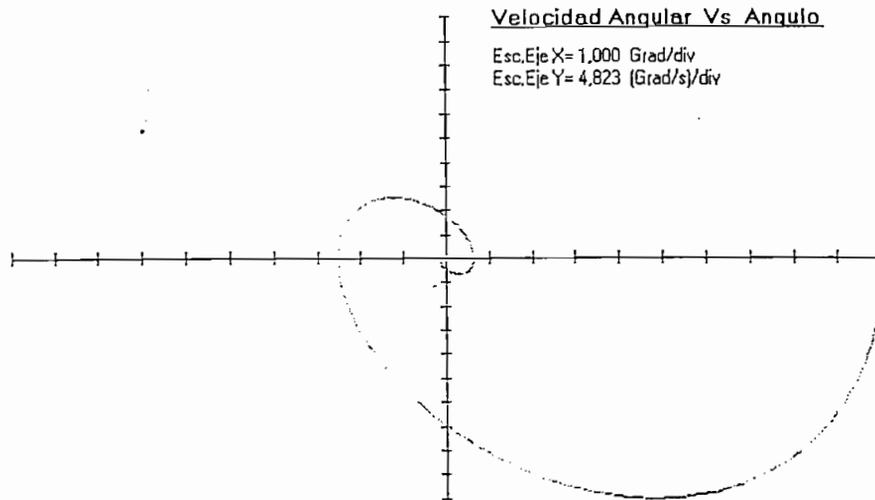


Fig.3.2.4.- Gráfica de la Velocidad Angular Vs. Angulo para el Control PD

Datos de la Fig.3.2.4:

$$W_{\phi=0} = -33,593 \text{ grad/s}$$

$$\Phi_{W=0} = - 2,552 \text{ grad}$$

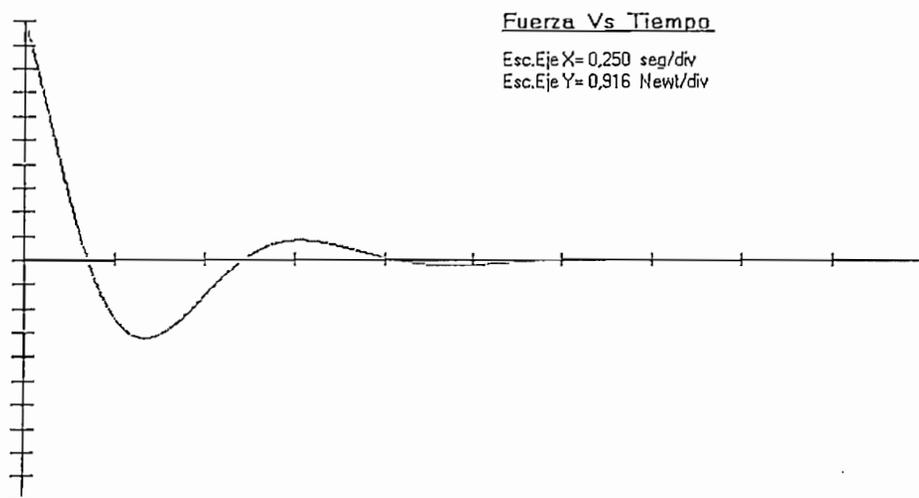


Fig.3.2.5.- Gráfica de la Fuerza Vs. Tiempo para el Control PD

Datos de la Fig.3.2.5:

$$t_p = 0,335\text{seg}$$

$$t_{cf} = 0,178\text{seg}$$

$$t_s = 1\text{seg}$$

$$f_M = 9,16\text{N}$$

$$V_p = - 2,98\text{N}$$

Observando la Fig.3.3.5, se puede decir que la fuerza converge a cero, ratificando así que el sistema se equilibra, pues ya no es necesario aplicar fuerza alguna.

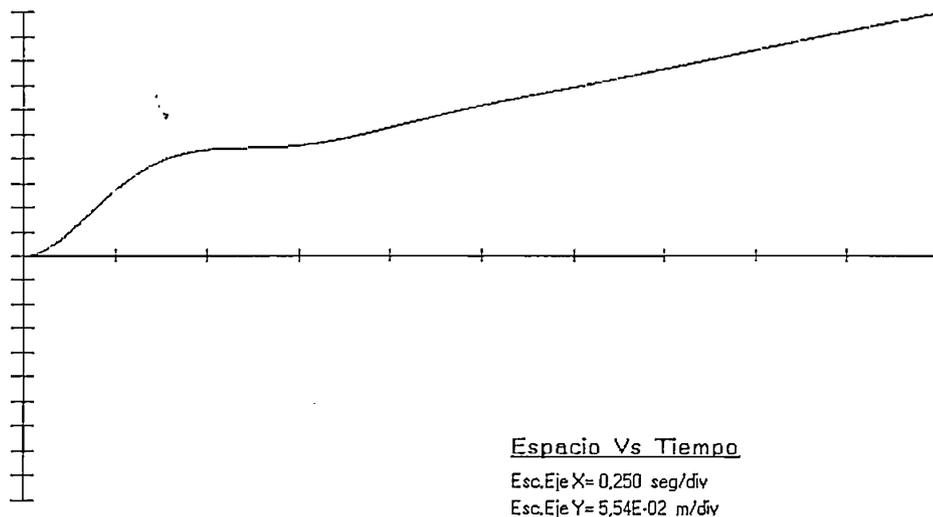


Fig.3.2.6.- Gráfica del Espacio Vs. Tiempo para el Control PD

Datos de la Fig.3.2.6:

$$X_{t_s=1,289\text{seg}} = 0,349\text{m}$$

Como puede verse en la Fig.3.2.7, a pesar de haberse equilibrado el péndulo, el carrito continúa alejándose del origen, con lo cual en algún momento llegará a sobrepasar los límites de la pista.

El péndulo llega a equilibrarse aproximadamente 0,349m, antes de que el carrito choque con los bordes de la pista.

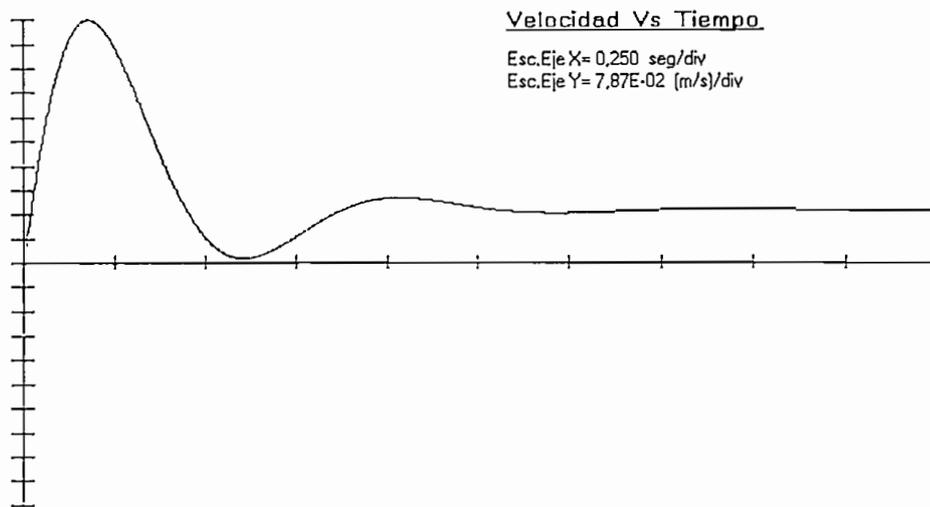


Fig.3.2.7.- Gráfica de la Velocidad Vs. Tiempo para el Control PD

Datos de la Fig.3.2.7:

$$t_p = 0,182\text{seg}$$

$$V_f = 0,174\text{m/s}$$

$$V_p = 0,787\text{m/s}$$

La curva de la velocidad no llega a hacerse cero, estabilizándose en un valor constante de 0,797m/s, indicando que el carrito queda con un movimiento rectilíneo uniforme.

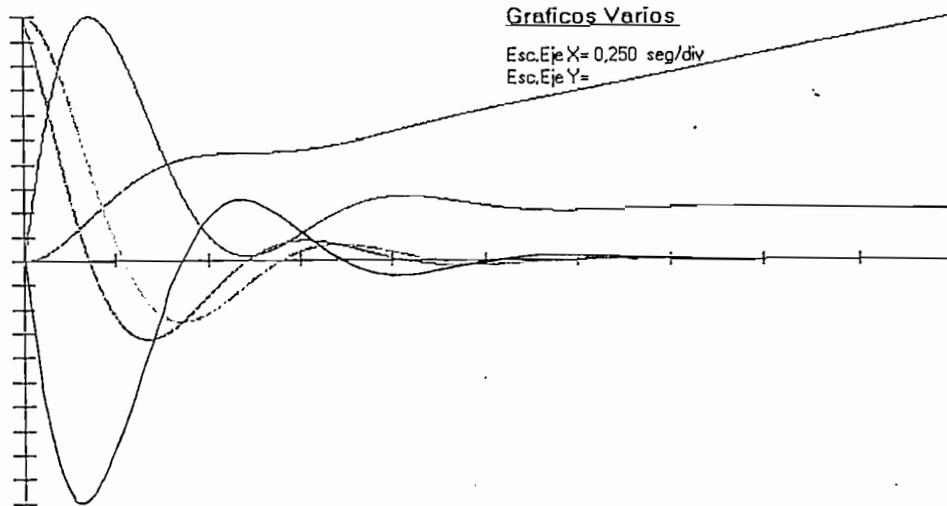


Fig.3.2.8.- Gráfica de la s Cuatro Variables y la Fuerza Vs. Tiempo para el Control PD

En la Fig.3.2.8 se observa claramente que el ángulo, la velocidad angular y la fuerza llegan a cero equilibrando al péndulo, mientras que el carrito continúa alejándose del origen con una velocidad positiva constante.

A continuación se presentan los resultados obtenidos del control:

Raíces de Lazo Cerrado = $-3,20 \pm j 7,33$
<u>De la Raíz Compleja de Lazo Cerrado</u>
Coefficiente de amortiguamiento = 0,40
Frecuencia no amortiguada = 8,00 rad/s
Máximo sobreimpulso = 25,38 %
Tiempo de establecimiento con el criterio de 2% = 1,25 Seg.
$G_c = K_p * (1 + K_d * S)$
$K_p = 52,47$
$K_d = 0,08$

Fig.3.2.9.- Resultados del programa en el Control P.D.

De la Fig.3.2.9, se observa que el sistema compensado en lazo cerrado tiene una raíz compleja conjugada, la cual determina el comportamiento del sistema.

Análisis de Resultados del Control Proporcional Integral Derivativo

Para el análisis se trabaja con los mismos datos del péndulo y condiciones iniciales que el caso anterior.

Datos ingresados para el Control PID:

$$K_p = 100$$

$$K_d = 0,15$$

$$K_i = 2$$

Las gráficas correspondientes a este control son:

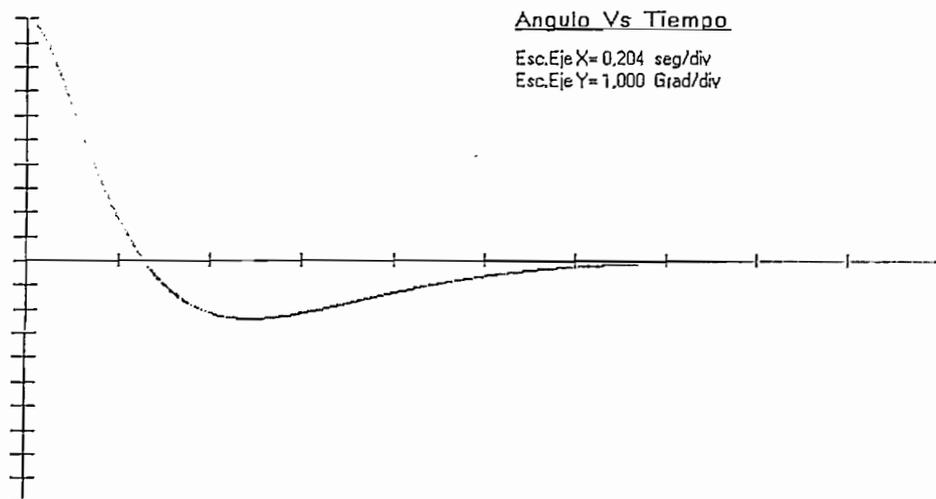


Fig.3.2.10.- Gráfica del Angulo Vs. Tiempo para el Control P.I.D.

Datos de la Fig.3.2.10:

$$t_r = 0,263\text{seg}$$

$$t_p = 0,488\text{seg}$$

$$t_s = 1,282\text{seg}$$

$$V_p = -2,408\text{grad}$$

$$M_p = 24,08\%$$

Este control logra conseguir un máximo sobreimpulso menor al 25%, esto se debe a que las raíces complejas de lazo cerrado no son completamente dominantes como puede verse en la Fig.3.2.17.

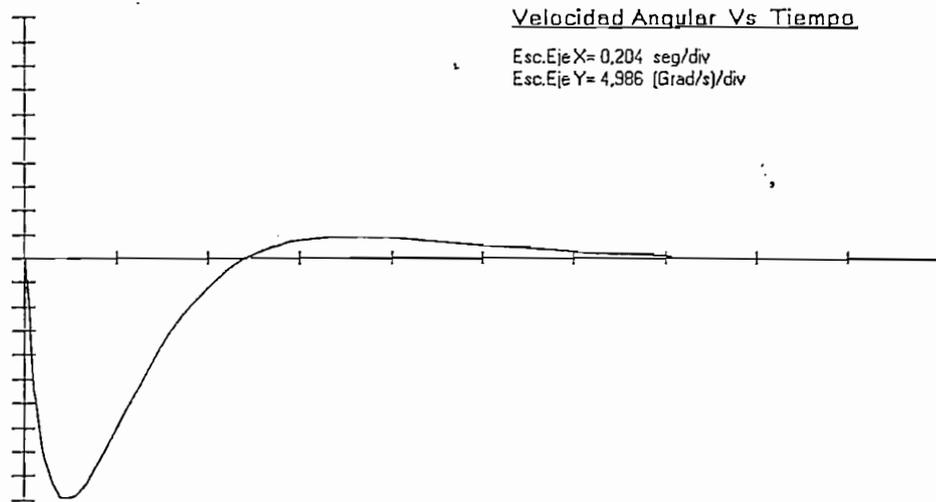


Fig.3.2.11.- Gráfica de la Velocidad Angular Vs. Tiempo para el Control P.I.D.

Datos de la Fig.3.2.11:

$$t_p = 0,096\text{seg}$$

$$t_s = 1,339\text{seg}$$

$$V_p = -49,86\text{grad/s}$$

$$t_{cva} = 0,502\text{seg}$$

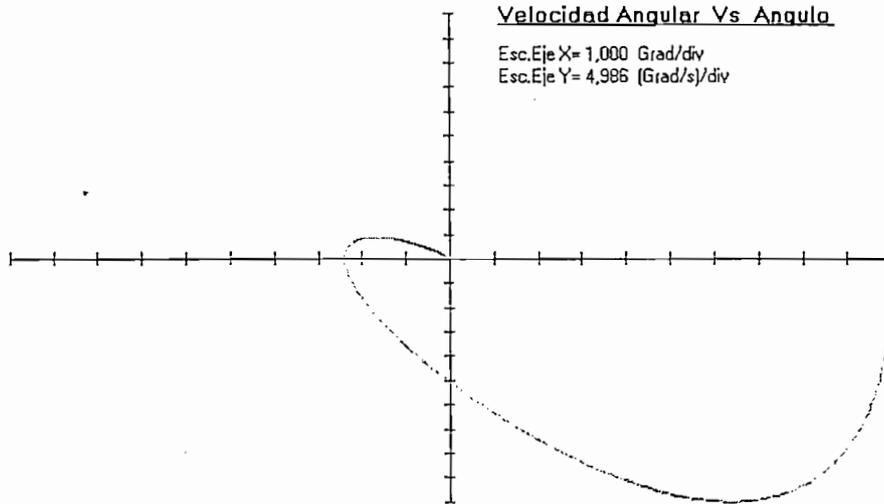


Fig.3.2.12.- Gráfica de la Velocidad Angular Vs. Angulo para el Control P.I.D.

Datos de la Fig.3.2.12:

$$W_{\phi=0} = - 25,316 \text{ grad/s}$$

$$\Phi_{W=0} = - 2,408 \text{ grad}$$

En la Fig.3.2.12 se observa, un comportamiento subamortiguado del sistema, donde el ángulo, y la velocidad angular se hacen cero.

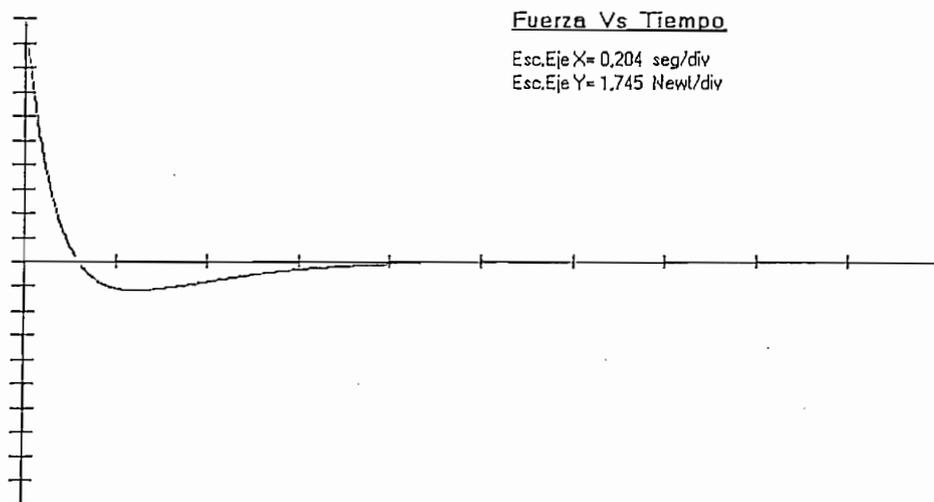


Fig.3.2.13.- Gráfica de la Fuerza Vs. Tiempo para el Control P.I.D.

Datos de la Fig.3.2.13:

$$t_p = 0,251\text{seg}$$

$$t_{cr} = 0,120\text{seg}$$

$$t_s = 0,709\text{seg}$$

$$f_M = 17,45\text{N}$$

$$V_p = -0,203\text{N}$$

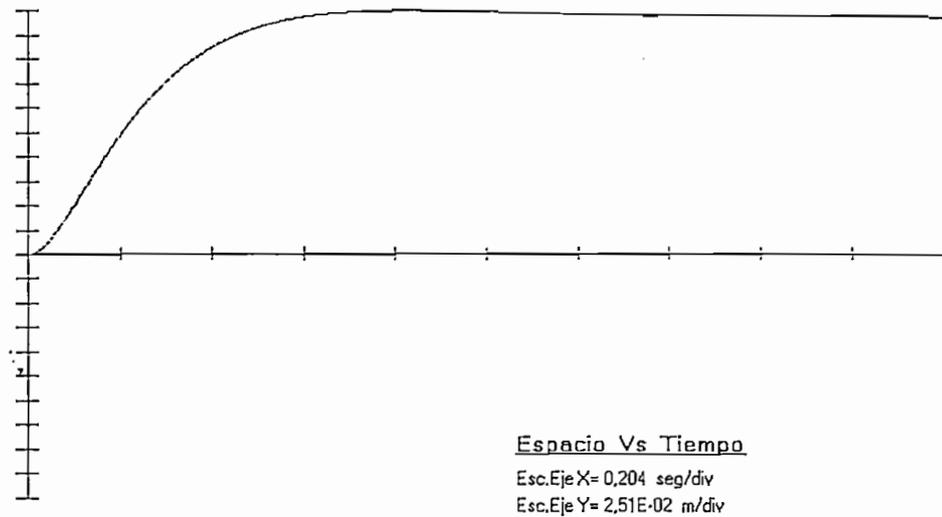


Fig.3.2.14.- Gráfica del Espacio Vs. Tiempo para el Control P.I.D.

Datos de la Fig.3.2.14:

$$V_p = 0,252\text{m}$$

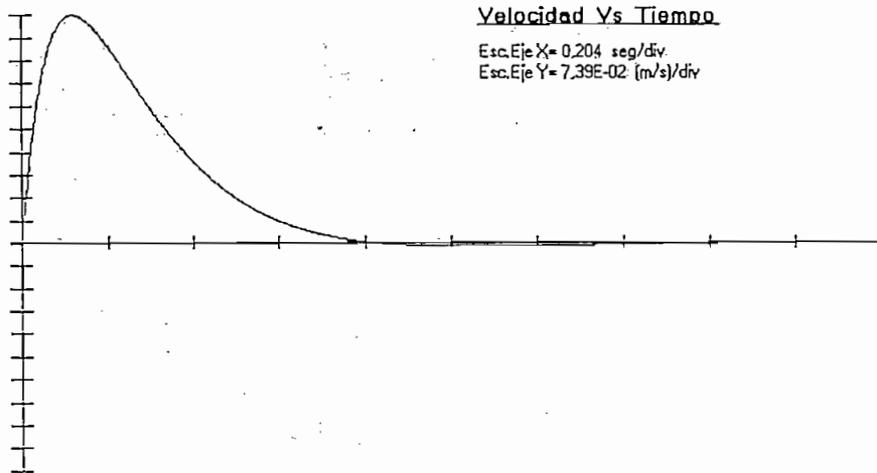


Fig.3.2.15.- Gráfica de la Velocidad Vs. Tiempo para el Control P.I.D.

Datos de la Fig.3.2.15:

$$t_{cvc} = 0,847 \text{seg}$$

$$t_p = 0,121 \text{seg}$$

$$V_p = 0,739 \text{m/s}$$

$$t_s = 0,776 \text{seg}$$

Observando las Fig.3.2.14 y Fig.3.2.15 se observa que el carro llega a detenerse a 0,252m del centro de la pista aproximadamente.

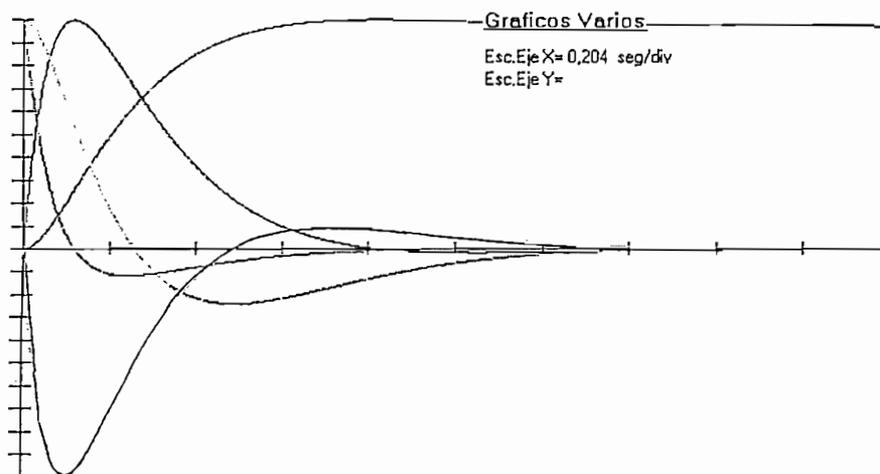


Fig.3.2.16.- Gráfica de las Cuatro Variables y la Fuerza Vs. Tiempo para el Control P.I.D.

Todas las curvas tienden a hacerse cero a excepción de la posición la cual se estabiliza en un valor positivo aproximadamente igual a 0,252m.

Lo resultados obtenidos en el programa para este control son:

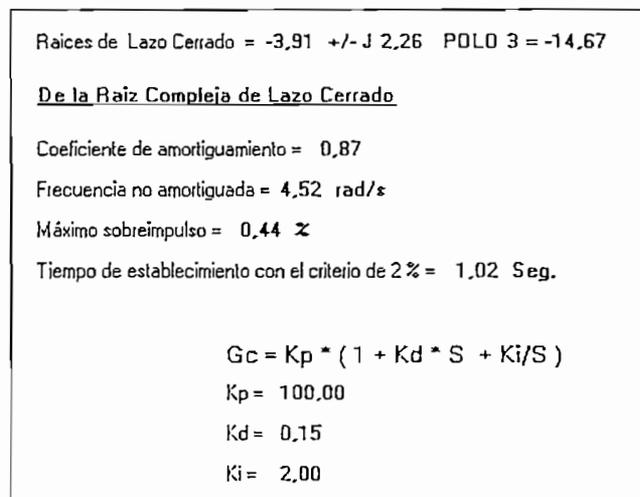


Fig.3.2.17.- Resultados del Programa en el Control P.I.D.

De la Fig.3.2.17, se tiene que el sistema compensado tiene tres raíces de lazo cerrado, dos de las cuales son complejas y una real, la raíz real es un poco menos dominante que la raíz compleja, lo que explica el comportamiento subamortiguado del sistema.

3.3.- Realimentación de Estado y Análisis de Resultados

Para realizar el control de Realimentación de Estado se debe describir el sistema a variables de estado como se detalla a continuación:

$$\begin{aligned}\dot{\underline{X}}(t) &= \underline{A} \underline{X}(t) + \underline{B}u(t) \\ y(t) &= \underline{C} \underline{X}(t)\end{aligned}$$

se llama:

$X_1(t) = x(t)$	Posición del carro
$X_2(t) = \frac{\partial x(t)}{\partial t}$	Velocidad del carro
$X_3(t) = \Phi(t)$	Angulo entre la vertical y el péndulo
$X_4(t) = \frac{\partial \Phi(t)}{\partial t}$	Velocidad angular del péndulo

derivando las ecuaciones anteriores se tiene:

$$\dot{X}_1(t) = X_2(t)$$

$$\dot{X}_2(t) = \frac{1}{M}u(t) \quad (\text{de la Ec.2.2.11})$$

$$\dot{X}_3(t) = X_4(t)$$

$$\dot{X}_4(t) = \frac{g}{L'}X_3(t) - \frac{1}{L'M}u(t) \quad (\text{de las Ecs.2.2.10 y 2.2.11})$$

representando en forma matricial queda:

$$\begin{bmatrix} \dot{X}_1(t) \\ \dot{X}_2(t) \\ \dot{X}_3(t) \\ \dot{X}_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g}{L'} & 0 \end{bmatrix} \begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \\ X_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ M \\ 0 \\ -1 \\ L'M \end{bmatrix} u(t)$$

Como el ángulo $\Phi(t)$ es la variable de salida, la expresión matricial de salida se representa de la siguiente forma:

$$Y(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \\ X_4(t) \end{bmatrix}$$

El diagrama de bloques se presentó en la Fig.2.3.16 del capítulo II.

El objetivo de este control, es medir las cuatro variables de estado (\underline{X}) y realizar una realimentación que sea una combinación lineal de estas componentes, para lo que se multiplica por el vector de realimentación (\underline{K}) y así, es obtenida la entrada de control que estabilizará al sistema.

Como primer paso, se debe chequear que el sistema sea controlable, para comprobar que las suposiciones que se han hecho para la obtención del modelo matemático no hacen que el sistema sea incontrolable, la matriz controlabilidad para un sistema de 4 variables de estado es igual a:

$$Mc = \left[\underline{B} \mid \underline{AB} \mid \underline{A^2B} \mid \underline{A^3B} \right]$$

resolviendo se tiene que:

$$Mc = \begin{bmatrix} 0 & \frac{1}{M} & 0 & 0 \\ \frac{1}{M} & 0 & 0 & 0 \\ 0 & \frac{-1}{L^1 M} & 0 & \frac{-g}{L^2 M} \\ \frac{-1}{L^1 M} & 0 & \frac{-g}{L^2 M} & 0 \end{bmatrix}$$

$$\det(Mc) = \frac{g^2}{L^4 M^4} \neq 0$$

El determinante de la Matriz Controlabilidad (M_c) es diferente de cero, por lo que el sistema es controlable. Lo que quiere decir que, al vector de estado $\underline{X}(t)$ se lo puede llevar desde un estado inicial $\underline{X}(0)$, hasta cualquier estado final arbitrario $\underline{X}(1)$, en un intervalo de tiempo finito, con la aplicación de una señal de control $u(t)$ físicamente factible.

El diagrama de bloques de la Fig.2.3.16 se lo puede desglosar de la manera presentada en la Fig.3.3.1, lo cual nos permite obtener las ecuaciones del Control por Realimentación de Estado:

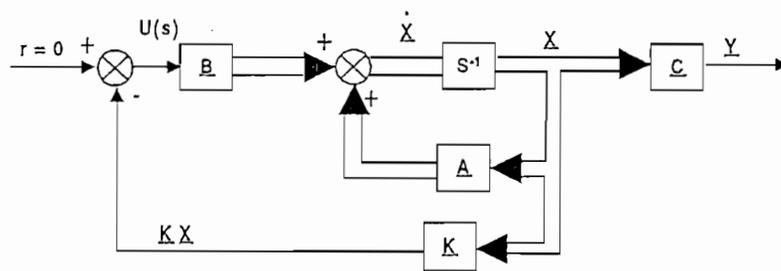


Fig.3.3.1.- Diagrama de Bloques Desglosado del Control por Realimentación de Estado

Anteriormente, se vio que la descripción a variables de estado en forma matricial se puede representar como sigue:

$$\dot{X}(t) = \underline{A} X(t) + \underline{B} u(t)$$

Como se mencionó en la compensación por Realimentación de Estado el parámetro de control $u(t)$ (fuerza) es una función de las variables de estado, la cual esta definida como:

$$u(t) = -\underline{K} \underline{X}(t)$$

$$u(t) = -\begin{bmatrix} K_1 & K_2 & K_3 & K_4 \end{bmatrix} \begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \\ X_4(t) \end{bmatrix} = -[K_1 X_1(t) + K_2 X_2(t) + K_3 X_3(t) + K_4 X_4(t)]$$

donde:

\underline{K} = es el vector de realimentación

$\underline{X}(t)$ = es el vector de estado

$u(t)$ = es la fuerza externa que se aplica al carro

reemplazando esta última ecuación en la ecuación de estado queda:

$$\dot{\underline{X}}(t) = \underline{A}\underline{X}(t) + \underline{B}[-\underline{K}\underline{X}(t)]$$

sacando factor común se tiene:

$$\dot{\underline{X}}(t) = [\underline{A} - \underline{BK}]\underline{X}(t) \quad (\text{Ec.3.3.1})$$

resolviendo:

$$[\underline{A} - \underline{BK}] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g}{L'} & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ M \\ -1 \\ L'M \end{bmatrix} \begin{bmatrix} K_1 & K_2 & K_3 & K_4 \end{bmatrix}$$

$$[\underline{A} - \underline{BK}] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g}{L'} & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{K_1}{M} & \frac{K_2}{M} & \frac{K_3}{M} & \frac{K_4}{M} \\ 0 & 0 & 0 & 0 \\ \frac{-K_1}{L'M} & \frac{-K_2}{L'M} & \frac{-K_3}{L'M} & \frac{-K_4}{L'M} \end{bmatrix}$$

$$[\underline{A} - \underline{BK}] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-K_1}{M} & \frac{-K_2}{M} & \frac{-K_3}{M} & \frac{-K_4}{M} \\ 0 & 0 & 0 & 1 \\ \frac{K_1}{L'M} & \frac{K_2}{L'M} & \frac{gM + K_3}{L'M} & \frac{K_4}{L'M} \end{bmatrix}$$

aplicando la transformada de Laplace a la Ec.3.3.1 se obtiene:

$$s\underline{X}(s) - \underline{X}(0) = [\underline{A} - \underline{BK}] \underline{X}(s)$$

despejando $\underline{X}(0)$:

$$\underline{X}(0) = [s\underline{I} - \underline{A} + \underline{BK}] \underline{X}(s)$$

por lo que $\underline{X}(s)$ es igual a:

$$\underline{X}(s) = [s\underline{I} - \underline{A} + \underline{BK}]^{-1} \underline{X}(0) \quad (\text{Ec.3.3.2})$$

La Ec.3.3.2 es la solución general de la ecuación de estado en el dominio de Laplace.

se encuentra la matriz $[s\underline{I} - \underline{A} + \underline{BK}]$ de lazo cerrado :

$$[s\underline{I} - \underline{A} + \underline{BK}] = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 & 0 \\ -K_1 & -K_2 & -K_3 & -K_4 \\ M & M & M & M \\ 0 & 0 & 0 & 1 \\ K_1 & K_2 & gM + K_3 & K_4 \\ L'M & L'M & L'M & L'M \end{bmatrix}$$

$$[s\underline{I} - \underline{A} + \underline{BK}] = \begin{bmatrix} s & -1 & 0 & 0 \\ \frac{K_1}{L'M} & \frac{sM + K_2}{L'M} & \frac{K_3}{L'M} & \frac{K_4}{L'M} \\ \frac{M}{L'M} & \frac{M}{L'M} & \frac{M}{L'M} & \frac{M}{L'M} \\ 0 & 0 & s & -1 \\ -\frac{K_1}{L'M} & -\frac{K_2}{L'M} & \frac{-gM - K_3}{L'M} & \frac{sL'M - K_4}{L'M} \end{bmatrix}$$

La inversa se obtiene por el método de la matriz adjunta que multiplicada por el vector de condiciones iniciales, permite obtener el vector de estado en el dominio de Laplace de la Ec.3.3.2, entonces:

$$X(s) = \frac{1}{\det} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \\ x_4(0) \end{bmatrix}$$

donde:

- El determinante es igual a:

$$\det = \det[sI - \underline{A} + \underline{BK}]$$

$$\det = s^4 + s^3 \frac{(L'K_2 - K_4)}{L'M} + s^2 \frac{(-gM - K_3 + K_1L')}{L'M} + s \frac{(-gK_2)}{L'M} - \frac{gK_1}{L'M} \quad (\text{Ec.3.3.3})$$

- El primer término de la matriz adjunta es:

$$a_{11} = \begin{vmatrix} \frac{sM + K_2}{M} & \frac{K_2}{M} & \frac{K_4}{M} \\ 0 & s & -1 \\ \frac{-K_2}{L'M} & \frac{-(gM + K_3)}{L'M} & \frac{sL'M - K_4}{L'M} \end{vmatrix}$$

$$a_{11} = s^3 + s^2 \left(\frac{K_2}{M} - \frac{K_4}{L'M} \right) - s \left(\frac{g}{L'} + \frac{K_3}{L'M} \right) - \frac{gK_2}{L'M}$$

- El segundo término de la matriz adjunta es:

$$a_{12} = - \begin{vmatrix} -1 & 0 & 0 \\ 0 & s & -1 \\ \frac{-K_2}{L'M} & \frac{-(gM + K_3)}{L'M} & \frac{sL'M + K_4}{L'M} \end{vmatrix}$$

$$a_{12} = s^2 - s \frac{K_4}{L'M} - \left(\frac{g}{L'} + \frac{K_3}{L'M} \right)$$

- Análogamente se obtienen los otros términos de la matriz adjunta:

$$a_{13} = -s \frac{K_3}{M} - \frac{K_4 g}{L' M}$$

$$a_{14} = -s \frac{K_4}{M} - \frac{K_3}{M}$$

$$a_{21} = -s^2 \frac{K_1}{M} + \frac{K_1 g}{L' M}$$

$$a_{22} = s^3 - s^2 \frac{K_4}{L' M} - s \left(\frac{g}{L'} + \frac{K_3}{L' M} \right)$$

$$a_{23} = -s^2 \frac{K_3}{M} - s \frac{K_4 g}{L' M}$$

$$a_{24} = -s^2 \frac{K_4}{M} - s \frac{K_3}{M}$$

$$a_{31} = s \frac{K_1}{L' M}$$

$$a_{32} = s \frac{K_2}{L' M} + \frac{K_1}{L' M}$$

$$a_{33} = s^3 + s^2 \left(\frac{K_2}{M} - \frac{K_4}{L' M} \right) + s \frac{K_1}{M}$$

$$a_{34} = s^2 + s \frac{K_2}{M} + \frac{K_1}{M}$$

$$a_{41} = s^2 \frac{K_1}{L' M}$$

$$a_{42} = s^2 \frac{K_2}{L' M} + s \frac{K_1}{L' M}$$

$$a_{43} = s^2 \left(\frac{g}{L'} + \frac{K_3}{L' M} \right) + s \frac{K_2 g}{L' M} - \frac{K_1 g}{L' M}$$

$$a_{44} = s^3 + s^2 \frac{K_2}{M} + s \frac{K_1}{M}$$

multiplicando la matriz por el vector de condiciones iniciales queda:

$$\underline{X}(s) = \frac{1}{\det} \begin{bmatrix} a_{11}X_1(0) + a_{12}X_2(0) + a_{13}X_3(0) + a_{14}X_4(0) \\ a_{21}X_1(0) + a_{22}X_2(0) + a_{23}X_3(0) + a_{24}X_4(0) \\ a_{31}X_1(0) + a_{32}X_2(0) + a_{33}X_3(0) + a_{34}X_4(0) \\ a_{41}X_1(0) + a_{42}X_2(0) + a_{43}X_3(0) + a_{44}X_4(0) \end{bmatrix}$$

resolviendo se tiene que los estados del sistema son:

$$X_1(s) = \frac{s^3b_{13} + s^2b_{12} + sb_{11} + b_{10}}{\det}$$

$$X_2(s) = \frac{s^3b_{23} + s^2b_{22} + sb_{21} + b_{20}}{\det}$$

$$X_3(s) = \frac{s^3b_{33} + s^2b_{32} + sb_{31} + b_{30}}{\det}$$

$$X_4(s) = \frac{s^3b_{43} + s^2b_{42} + sb_{41} + b_{40}}{\det}$$

donde:

$$b_{13} = X_1(0) \quad (\text{Ec.3.3.4})$$

$$b_{12} = \left(\frac{K_2}{M} - \frac{K_4}{L'M} \right) X_1(0) + X_2(0) \quad (\text{Ec.3.3.5})$$

$$b_{11} = - \left[\left(\frac{g}{L'} + \frac{K_3}{L'M} \right) X_1(0) + \frac{K_4}{L'M} X_2(0) + \frac{K_3}{M} X_3(0) + \frac{K_4}{M} X_4(0) \right] \quad (\text{Ec.3.3.6})$$

$$b_{10} = - \left[\frac{gK_2}{L'M} X_1(0) + \left(\frac{g}{L'} + \frac{K_3}{L'M} \right) X_2(0) + \frac{K_4g}{L'M} X_3(0) + \frac{K_3X_4(0)}{M} \right] \quad (\text{Ec.3.3.7})$$

$$b_{23} = X_2(0) \quad (\text{Ec.3.3.8})$$

$$b_{22} = - \left[\frac{K_1}{M} X_1(0) + \frac{K_4}{L'M} X_2(0) + \frac{K_4}{M} X_4(0) + \frac{K_3}{M} X_3(0) \right] \quad (\text{Ec.3.3.9})$$

$$b_{21} = - \left[\left(\frac{g}{L'} + \frac{K_3}{L'M} \right) X_2(0) + \frac{K_4 g}{L'M} X_3(0) + \frac{K_3}{M} X_4(0) \right] \quad (\text{Ec.3.3.10})$$

$$b_{20} = \frac{K_1 g}{L'M} X_1(0) \quad (\text{Ec.3.3.11})$$

$$b_{33} = X_3(0) \quad (\text{Ec.3.3.12})$$

$$b_{32} = \left(\frac{K_2}{M} - \frac{K_4}{L'M} \right) X_3(0) + X_4(0) \quad (\text{Ec.3.3.13})$$

$$b_{31} = \frac{K_1}{L'M} X_1(0) + \frac{K_2}{L'M} X_2(0) + \frac{K_1}{M} X_3(0) + \frac{K_2}{M} X_4(0) \quad (\text{Ec.3.3.14})$$

$$b_{30} = \frac{K_1}{L'M} X_2(0) + \frac{K_1}{M} X_4(0) \quad (\text{Ec.3.3.15})$$

$$b_{43} = X_4(0) \quad (\text{Ec.3.3.16})$$

$$b_{42} = \frac{K_1}{L'M} X_1(0) + \frac{K_2}{L'M} X_2(0) + \left(\frac{g}{L'} + \frac{K_3}{L'M} \right) X_3(0) + \frac{K_2}{M} X_4(0) \quad (\text{Ec.3.3.17})$$

$$b_{41} = \frac{K_1}{L'M} X_2(0) + \frac{K_2 g}{L'M} X_3(0) + \frac{K_1}{M} X_4(0) \quad (\text{Ec.3.3.18})$$

$$b_{40} = \frac{K_1 g}{L'M} X_3(0) \quad (\text{Ec.3.3.19})$$

Para el análisis de estabilidad, se tiene que la ecuación característica del sistema en lazo cerrado de cuarto orden es como se indica a continuación.

$$q(s) = s^4 + a_{c3} s^3 + a_{c2} s^2 + a_{c1} s + a_{c0} = 0 \quad (\text{Ec.3.3.20})$$

Para que el sistema sea estable, se debe cumplir que:

- * Todas las raíces deben estar ubicadas en el semiplano izquierdo del plano complejo 's'.
- * Las raíces deben ser cuatro y diferentes de cero.
- * Sí son raíces complejas deben ser conjugadas.
- * Los coeficientes del polinomio deben ser distintos de cero y positivos.

factorizando $q(s)$ queda:

$$q(s) = (s - p_1)(s - p_2)(s - p_3)(s - p_4) = 0$$

donde:

p_1, p_2, p_3, p_4 : son los polos de lazo cerrado deseados y deben ser las raíces de la ecuación característica, llamados Valores Propios de la matriz de lazo cerrado $[sI - \underline{A} + \underline{B} \underline{K}]$, $q(s)$ es el polinomio característico de esta matriz.

multiplicando se obtiene :

$$a_{c3} = -(p_1 + p_2 + p_3 + p_4) \quad (\text{Ec.3.3.21})$$

$$a_{c2} = p_1 p_2 + p_1 p_3 + p_1 p_4 + p_2 p_3 + p_2 p_4 + p_3 p_4 \quad (\text{Ec.3.3.22})$$

$$a_{c1} = -(p_1 p_2 p_3 + p_1 p_3 p_4 + p_1 p_2 p_4 + p_2 p_3 p_4) \quad (\text{Ec.3.3.23})$$

$$a_{c0} = p_1 p_2 p_3 p_4 \quad (\text{Ec.3.3.24})$$

igualando las Ecs.3.3.3 y 3.3.20:

$$\det = q(s)$$

donde:

$$\det = s^4 + s^3 \left(\frac{L'K_2 - K_4}{L'M} \right) + s^2 \left(\frac{-gM - K_3 + K_1L'}{L'M} \right) + s \left(\frac{-gK_2}{L'M} \right) - \frac{gK_1}{L'M}$$

y

$$q(s) = s^4 + a_{c3}s^3 + a_{c2}s^2 + a_{c1}s + a_{c0}$$

se obtiene:

$$a_{c3} = \frac{L'K_2 - K_4}{L'M}$$

$$a_{c2} = \frac{-gM - K_3 + K_1L'}{L'M}$$

$$a_{c1} = \frac{-gK_2}{L'M}$$

$$a_{c0} = \frac{-gK_1}{L'M}$$

despejando se obtiene el vector de realimentación K:

$$K_1 = -\frac{a_{c0}L'M}{g} \quad (\text{Ec.3.3.25})$$

$$K_2 = -\frac{a_{c1}L'M}{g} \quad (\text{Ec.3.3.26})$$

$$K_3 = -L'M \left(\frac{L'}{g} a_{c0} + \frac{g}{L'} + a_{c2} \right) \quad (\text{Ec.3.3.27})$$

$$K_4 = -L'M \left(\frac{L'}{g} a_{c1} + a_{c3} \right) \quad (\text{Ec.3.3.28})$$

Obtenidos los valores de K_1, K_2, K_3, K_4 se puede encontrar $X_1(s), X_2(s), X_3(s), X_4(s)$.

Así, todos los estados presentan la siguiente forma:

$$X_n(s) = \frac{b_{n3}s^3 + b_{n2}s^2 + b_{n1}s + b_{n0}}{s^4 + a_{c3}s^3 + a_{c2}s^2 + a_{c1}s + a_{c0}} = \frac{C_n(s)}{q(s)}$$

Donde, b_{n3} , b_{n2} , b_{n1} , b_{n0} , pueden ser determinados, pues están en función de las condiciones iniciales, de los elementos del vector de realimentación de estado, longitud efectiva del péndulo (L'), masa del carro (M) y la gravedad (g).

Para hallar la respuesta en el tiempo de los estados, se aplica la transformada inversa de Laplace y para obtener la inversa se debe dividir en fracciones parciales a cada una de las ecuaciones de estado, de donde se obtienen nueve casos diferentes, según la ubicación de los polos de lazo cerrado que se ingresan:

1) Todos los polos son reales y distintos:

$$X_n(s) = \frac{A_{n1}}{s-p_1} + \frac{A_{n2}}{s-p_2} + \frac{A_{n3}}{s-p_3} + \frac{A_{n4}}{s-p_4}$$

Aplicando la inversa de Laplace se obtiene la respuesta en el tiempo:

$$x_n(t) = A_{n1}e^{p_1 t} + A_{n2}e^{p_2 t} + A_{n3}e^{p_3 t} + A_{n4}e^{p_4 t}$$

2) Dos polos reales repetidos y dos polos diferentes

$$X_n(s) = \frac{A_{n1}}{s-p_1} + \frac{A_{n2}}{(s-p_1)^2} + \frac{A_{n3}}{s-p_2} + \frac{A_{n4}}{s-p_3}$$

$$x_n(t) = (A_{n1} + A_{n2}t)e^{p_1 t} + A_{n3}e^{p_2 t} + A_{n4}e^{p_3 t}$$

3) Tres polos reales repetidos y un polo diferente

$$X_n(s) = \frac{A_{n1}}{s-p_1} + \frac{A_{n2}}{(s-p_1)^2} + \frac{A_{n3}}{(s-p_1)^3} + \frac{A_{n4}}{s-p_2}$$

$$x_n(t) = \left(A_{n1} + A_{n2}t + \frac{A_{n3}}{2!}t^2 \right) e^{p_1 t} + A_{n4} e^{p_2 t}$$

4) Todos los polos reales repetidos:

$$X_n(s) = \frac{A_{n1}}{s-p_1} + \frac{A_{n2}}{(s-p_1)^2} + \frac{A_{n3}}{(s-p_1)^3} + \frac{A_{n4}}{(s-p_1)^4}$$

$$x_n(t) = \left(A_{n1} + A_{n2}t + \frac{A_{n3}}{2!}t^2 + \frac{A_{n4}}{3!}t^3 \right) e^{p_1 t}$$

5) Cuatro polos repetidos de dos en dos:

$$X_n(s) = \frac{A_{n1}}{s-p_1} + \frac{A_{n2}}{(s-p_1)^2} + \frac{A_{n3}}{s-p_2} + \frac{A_{n4}}{(s-p_2)^2}$$

$$x_n(t) = (A_{n1} + A_{n2}t)e^{p_1 t} + (A_{n3} + A_{n4}t)e^{p_2 t}$$

6) Polos complejos conjugados distintos

$$X_n(s) = \frac{A_{n1}(s+\sigma_1)}{(s+\sigma_1)^2 + Wd_1^2} + \frac{A_{n2}Wd_1}{(s+\sigma_1)^2 + Wd_1^2} + \frac{A_{n3}(s+\sigma_2)}{(s+\sigma_2)^2 + Wd_2^2} + \frac{A_{n4}Wd_2}{(s+\sigma_2)^2 + Wd_2^2}$$

$$x_n(t) = e^{-\sigma_1 t} (A_{n1} \cos Wd_1 t + A_{n2} \operatorname{sen} Wd_1 t) + e^{-\sigma_2 t} (A_{n3} \cos Wd_2 t + A_{n4} \operatorname{sen} Wd_2 t)$$

7) Un polo complejo conjugado de multiplicidad dos:

$$X_n(s) = \frac{A_{n1}(s+\sigma)}{(s+\sigma)^2 + Wd^2} + \frac{A_{n2}Wd}{(s+\sigma)^2 + Wd^2} + \frac{A_{n3}(s+\sigma)}{[(s+\sigma)^2 + Wd^2]^2} + \frac{A_{n4}Wd}{[(s+\sigma)^2 + Wd^2]^2}$$

$$x_n(t) = e^{-\sigma t} (A_{n1} \cos Wdt + A_{n2} \operatorname{sen} Wdt) + \frac{A_{n3}t}{2Wd} e^{-\sigma t} \operatorname{sen} Wdt + \frac{A_{n4}}{2Wd} e^{-\sigma t} [\operatorname{sen} Wdt - Wdt \cos Wdt]$$

8) Un polo complejo conjugado y dos reales distintos

$$X_n(s) = \frac{A_{n1}}{(s-p_1)} + \frac{A_{n2}}{(s-p_2)} + \frac{A_{n3}(s+\sigma)}{(s+\sigma)^2 + Wd^2} + \frac{A_{n4}Wd}{(s+\sigma)^2 + Wd^2}$$

$$x_n(t) = A_{n1} e^{p_1 t} + A_{n2} e^{p_2 t} + A_{n3} e^{-\sigma t} \cos Wdt + A_{n4} e^{-\sigma t} \operatorname{sen} Wdt$$

9) Un polo real complejo conjugado y un polo real de multiplicidad 2

$$X_n(s) = \frac{A_{n1}}{(s-p_1)} + \frac{A_{n2}}{(s-p_1)^2} + \frac{A_{n3}(s+\sigma)}{(s+\sigma)^2 + Wd^2} + \frac{A_{n4}Wd}{(s+\sigma)^2 + Wd^2}$$

$$x_n(t) = (A_{n1} + A_{n2})e^{p_1 t} + A_{n3}e^{-\sigma t} \cos Wdt + A_{n4}e^{-\sigma t} \sin Wdt$$

Análisis de Resultados del Control por Realimentación de Estados

En este control se ingresan como datos los polos deseados de lazo cerrado (p_1, p_2, p_3, p_4), los cuales deben estar ubicados en el semiplano izquierdo del plano S , para que el sistema sea estable, pero es necesario también tomar en cuenta la rapidez del sistema, pues, si el sistema se hace muy lento, el péndulo no va a lograr estabilizarse en el tramo de pista que ha sido fijado.

Además el ángulo no debe crecer demasiado, pues si esto sucede, las suposiciones que se hicieron para linealizar el modelo matemático no serán válidas, y por ende, los resultados obtenidos tampoco serían válidos.

El programa desarrollado permite ensayar varias alternativas de ubicación de polos, se usa un ejemplo para analizar el comportamiento de este control, el cual cumple con las condiciones de estabilidad, rapidez y validez de las ecuaciones.

Datos del péndulo:

$$M = 1 \text{Kg}$$

$$m = 0,1 \text{Kg}$$

$$L = 0,5 \text{m}$$

Condiciones Iniciales:

$$\text{Espacio} = 0 \text{m}$$

$$\text{Velocidad} = 0 \text{m/s}$$

$$\text{Ángulo} = 10 \text{grad}$$

$$g = 9,8\text{m/s}$$

$$\text{Vel. Angular} = 0\text{grad/s}$$

$$\text{Longitud de pista} = 1\text{m}$$

Ubicación de polos de lazo cerrado propuesta:

$$\text{Polo1} = - 5,5 + j 2,5$$

$$\text{Polo2} = - 5,5 - j 2,5$$

$$\text{Polo3} = - 0,5$$

$$\text{Polo4} = -1$$

La ubicación de estos polos se realizó ensayando varias alternativas en el programa, cuyo efecto produzca un salida que cumpla con un máximo sobreimpulso inferior al 25% y que estabilice al péndulo antes que el carrito sobrepase los límites de la pista.

A continuación se presentan las mismas definiciones para el análisis de los gráficos:

- Tiempo de crecimiento: t_r
- Tiempo en alcanzar el primer pico : t_p
- Tiempo de establecimiento criterio del 2 % : t_s
- Tiempo del primer cambio de dirección de la velocidad del carro: t_{cvc}
- Tiempo del primer cambio de dirección de la fuerza: t_{cf}
- Tiempo del primer cambio de dirección de la velocidad angular: t_{cva}
- Valor pico de la primera oscilación: V_p
- Máximo sobreimpulso: M_p
- Fuerza máxima aplicada: f_M
- Espacio recorrido por el carro a t_s del ángulo : X_{ts}

- Velocidad angular en el primer punto donde el ángulo es cero : $\dot{\Phi}_{\Phi=0}$
- Angulo en el primer punto donde la velocidad angular es cero : $\Phi_{\dot{\Phi}=0}$
- Velocidad final con la que el carro queda moviéndose después de estabilizar al péndulo: V_f

Aplicando estos datos al programa se obtiene las gráficas de cada uno de los estados en el tiempo, de la fuerza Vs. el tiempo y el diagrama de fase, en las cuales se observan las siguientes características:

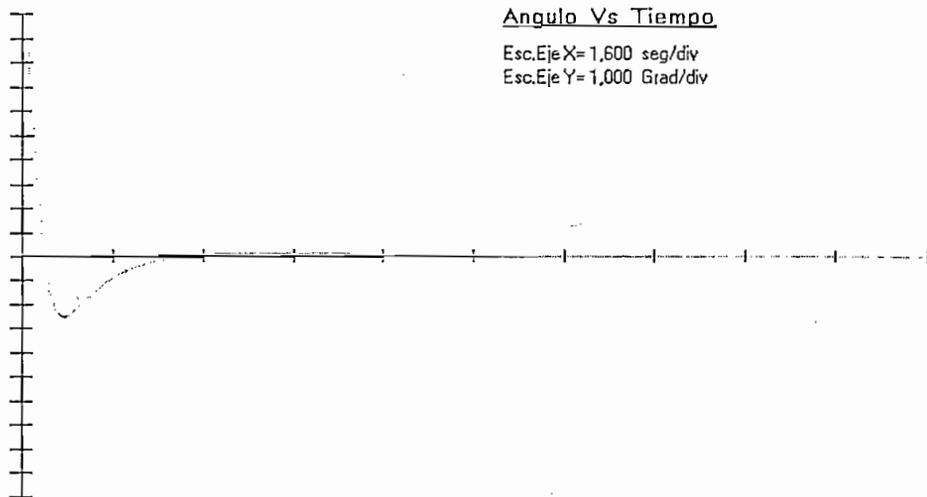


Fig.3.3.2.- Gráfica de el Angulo Vs. Tiempo para el Control por Realimentación de Estado

Datos de la Fig.3.3.2:

$$t_r = 0,383\text{seg}$$

$$t_p = 0,717\text{seg}$$

$$t_s = 2,385\text{seg}$$

$$V_p = -2,458\text{grad}$$

$$M_p = 24,58\%$$

Como se puede apreciar, el sistema se hace estable pues converge a la posición de referencia cero además, la forma en que converge es similar a un sistema de segundo orden en el caso subamortiguado.

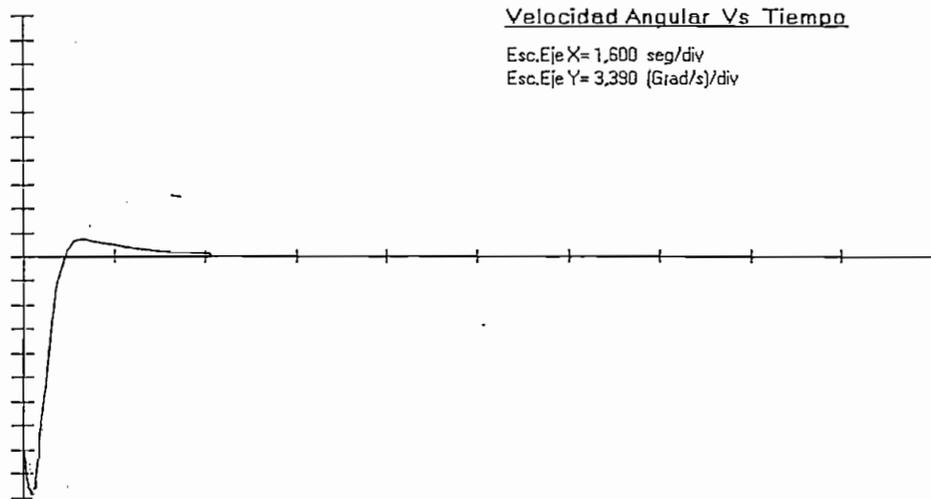


Fig.3.3.3.- Gráfica de la Velocidad Angular Vs. Tiempo para el Control por Realimentación de Estado

Datos de la Fig.3.3.3:

$$V_p = -33,9 \text{ grad/s}$$

$$t_p = 0,133 \text{ seg}$$

$$t_{cva} = 0,717 \text{ seg}$$

$$t_s = 2,3 \text{ seg}$$

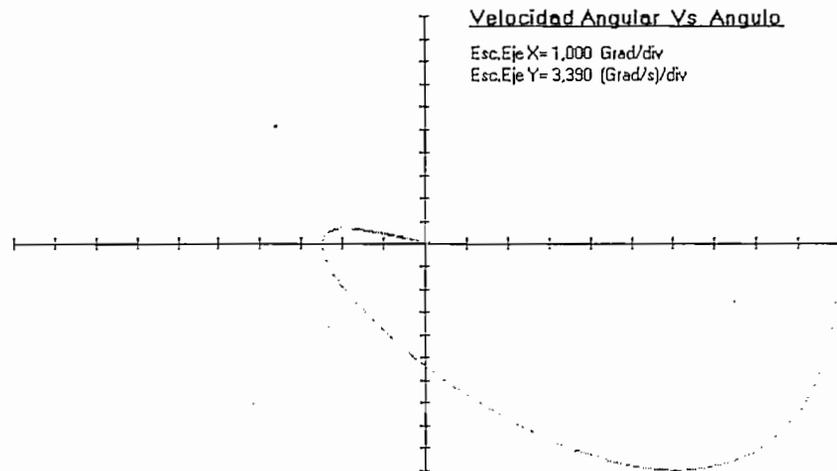


Fig.3.3.4.- Gráfica de la Velocidad Angular Vs. Angulo para el Control por Realimentación de Estado

Datos de la Fig.3.3.4:

$$W_{\phi=0} = -18,2 \text{ grad/s}$$

$$\Phi_{W=0} = -2,5 \text{ grad}$$

En la Fig.3.3.4 se puede observar claramente que el sistema es estable pues, tanto el ángulo como la velocidad angular convergen al punto de referencia cero.

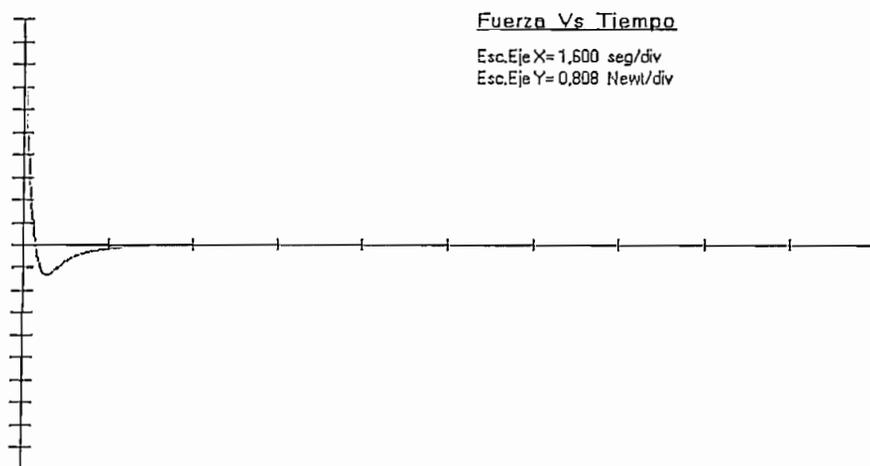


Fig.3.3.5.- Gráfica de la Fuerza Vs. Tiempo para el Control por Realimentación de Estado

Datos de la Fig.3.3.5:

$$V_p = -1,04N$$

$$t_p = 0,439\text{seg}$$

$$t_{cf} = 0,217\text{seg}$$

$$f_M = 8,08N$$

$$t_s = 1,5\text{seg}$$

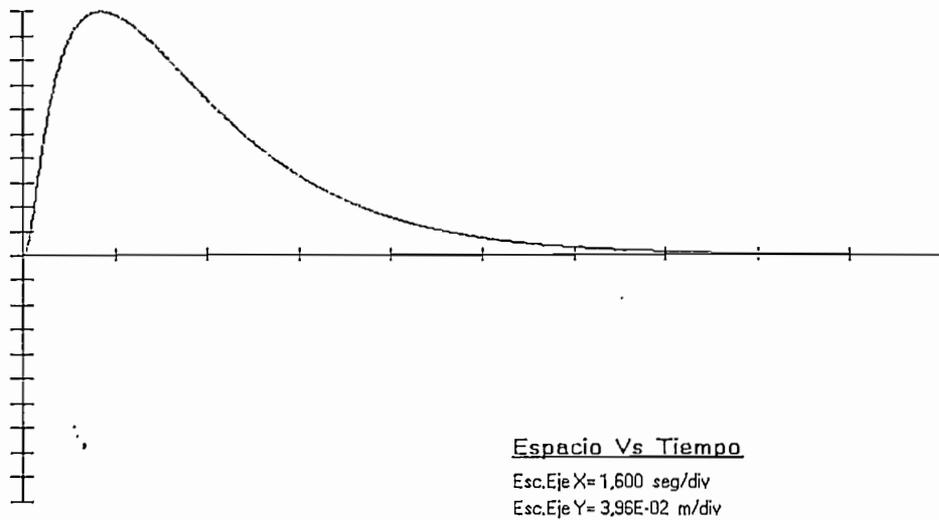


Fig.3.3.6.- Gráfica del Espacio Vs. Tiempo para el Control por Realimentación de Estado

Datos de la Fig.3.3.6:

$$V_p = 0,397\text{m}$$

$$t_p = 1,356\text{seg}$$

$$t_s = 10,5\text{seg}$$

En la Fig.3.3.6 se observa claramente que el carrito no sobrepasa los límites de la pista, comprobando así que los polos propuestos inicialmente satisfacen la condición de rapidez del sistema, y además que la posición final

del carro tiende al origen $X=0$ (por lo que la posición final del carrito será en la mitad de la pista).

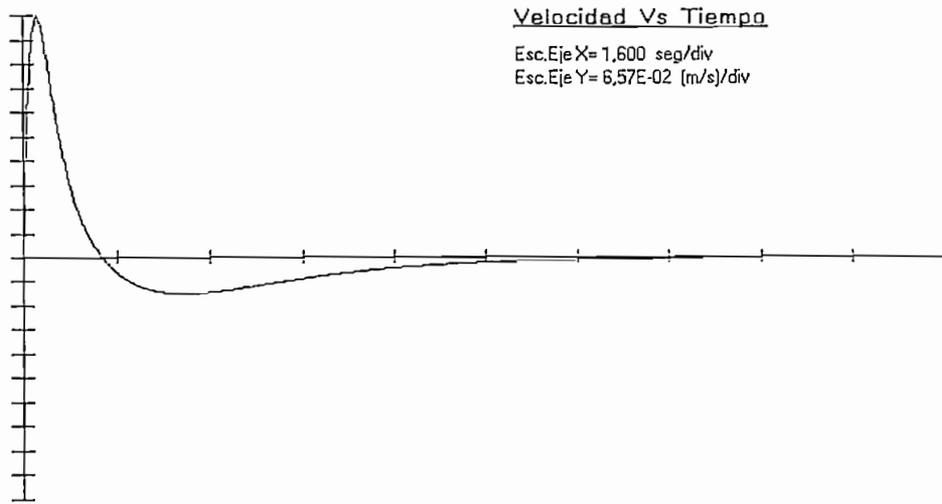


Fig.3.3.7.- Gráfica de la Velocidad del carro Vs. Tiempo para el Control por Realimentación de Estado

Datos de la Fig.3.3.7:

$$V_p = 0,657\text{m/s}$$

$$t_p = 0,217\text{seg}$$

$$t_{cva} = 1,356\text{seg}$$

$$t_s = 8,166\text{seg}$$

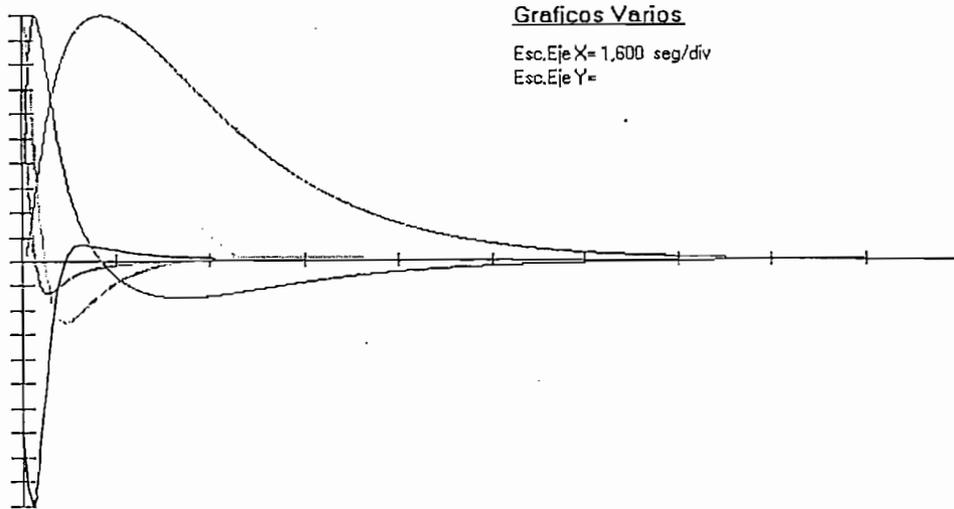


Fig.3.3.8.- Gráfica de los Cuatro Estados y la Fuerza Vs. Tiempo para el Control por Realimentación de Estado

Con estos resultados se puede decir que, con el control por realimentación de estado se logra estabilizar el péndulo, manteniendo a la varilla en posición vertical.

Debido a que este tipo de control tiene una realimentación de las cuatro variables de estado se consigue que el valor final de cada uno de los estados sea cero.

El comportamiento del sistema, según los ensayos realizados muestra que en la gráfica del Angulo Vs. el Tiempo, el máximo sobreimpulso aumenta mientras más lejanos del origen se ubique la parte real de las raíces de lazo cerrado deseadas; pero a la vez el máximo espacio recorrido por el carrito tiende a disminuir, esto se debe a que el tiempo de establecimiento se está disminuyendo, haciendo que el sistema se haga más rápido.

Con la realimentación de estado se consigue un sistema fuertemente estable, con el lugar geométrico de las raíces ubicado parcialmente en el semiplano izquierdo del plano complejo S.

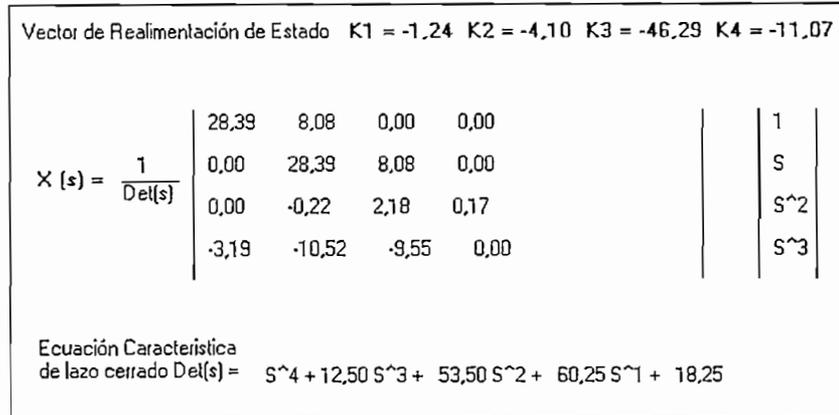


Fig.3.3.9.- Resultados del Item 1 del Control por Realimentación de Estado

El programa presenta tres pantallas de resultados, en el primer ítem de resultados Fig.3.3.9 se muestra el vector de realimentación de estado, la solución general de la ecuación de estado en el dominio S y el polinomio característico de la matriz de lazo cerrado que están en función de los polos deseados que se ingresaron como datos para el control por realimentación de estado.

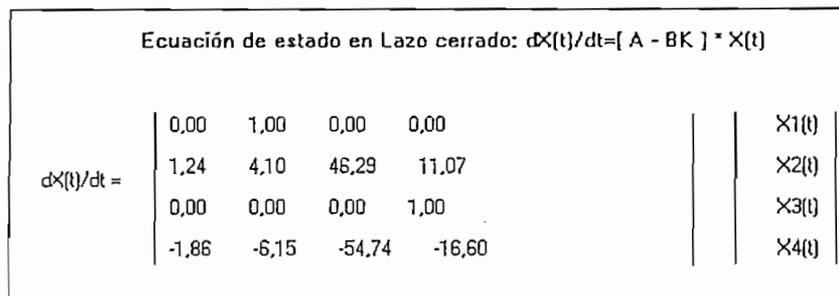


Fig.3.3.10.- Resultados del Item 2 del Control por Realimentación de Estado

El segundo ítem de resultados Fig.3.3.10 muestra la matriz de lazo cerrado [A - B K] y la ecuación de estado en lazo cerrado.

$$X_n(t) = A_{n1} \text{Exp}[-0,50 t] + A_{n2} \text{Exp}[-1,00 t] + \text{Exp}[-5,50 t] * \{ A_{n3} \cos[2,50 t] + A_{n4} \text{sen}[2,50 t] \}$$

CONSTANTES DE LA ECUACION EN EL TIEMPO PARA LOS CUATRO ESTADOS

ANGULO : A11 = 1,56 A12 = -1,53 A13 = -0,03 A14 = -0,36

VELOCIDAD ANGULAR : A21 = -0,78 A22 = 1,53 A23 = -0,75 A24 = 2,03

POSICION : A31 = 0,04 A32 = -0,17 A33 = 0,30 A34 = 0,61

VELOCIDAD : A41 = -0,02 A42 = 0,17 A43 = -0,15 A44 = -4,08

Fig.3.3.11.- Resultados del Ítem 3 del Control por Realimentación de Estado

Como tercer y último ítem de resultados la Fig.3.3.11 presenta la ecuación de la respuesta en el tiempo de los cuatro estados de interés.

En general, se puede concluir que la ubicación de los polos altera la forma de la respuesta del sistema, manteniéndolo estable, pues, las raíces escogidas obviamente siempre se ubicarán en el semiplano izquierdo del plano S.

3.4.- Control Fuzzy y Análisis de Resultados.

Como se mencionó anteriormente, para realizar el control por este método no es necesario conocer el modelo matemático de la planta, sino apenas se necesitan conocer que valores de salida se obtendrían de la planta real al aplicar un cierto valor de fuerza.

Para desarrollar el Control Fuzzy primero se discretizo al sistema según la siguiente descripción:

Uno de los métodos estándares de integración numérica es la integración rectangular por retraso¹, la cual se representa en la Fig.3.4.1.

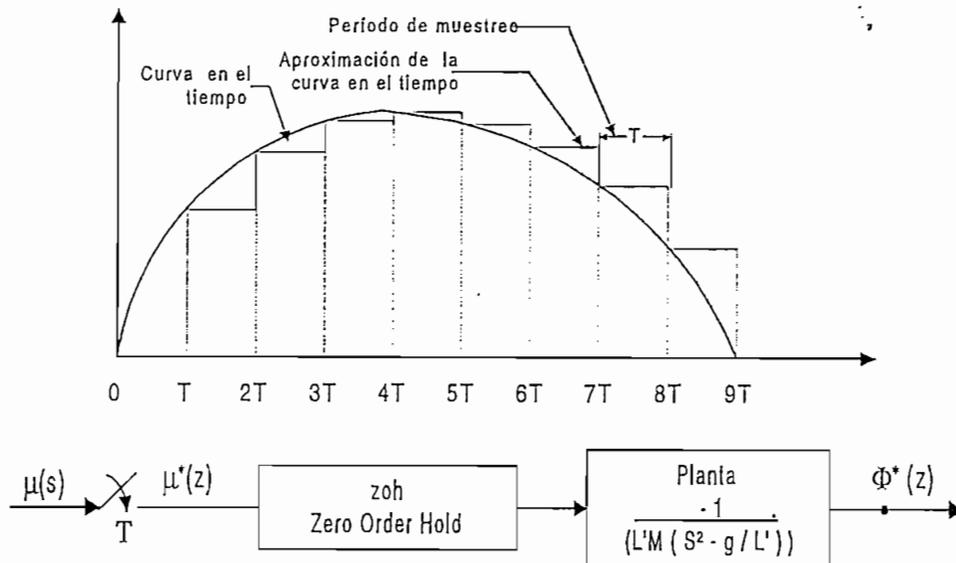


Fig.3.4.1 Sistema de Control Discreto del Péndulo Invertido por Integración Rectangular en Retraso.

¹ KUO Benjamin, *Digital Control Systems*, 1.992., pag. 425

Según la Fig.3.4.1 la función de transferencia discreta del péndulo invertido, aplicando la transformada Z es:

$$Goh\dot{G}_p(z) = \frac{z-1}{z} \cdot Z \left\{ \frac{1}{s} * \frac{-1}{L' M \left(s^2 - \frac{g}{L'} \right)} \right\} = \frac{z-1}{z} Z \left\{ \frac{-1}{L' M s^3 - g M s} \right\}$$

resolviendo:

$$= \frac{z-1}{z} \cdot Z \left\{ \frac{-s^{-3}}{L' M - g M s^{-2}} \right\} \quad (\text{Ec.3.4.1})$$

para transformar de S a Z se conoce que¹ :

$$s^{-1} = T/2 \frac{1+z^{-1}}{1-z^{-1}} \quad (\text{Ec.3.4.2})$$

$$s^{-2} = T^2/12 \frac{1+10z^{-1}+z^{-2}}{(1-z^{-1})^2} \quad (\text{Ec.3.4.3})$$

$$s^{-3} = T^2/2 \frac{z^{-1}+z^{-2}}{(1-z^{-1})^3} \quad (\text{Ec.3.4.4})$$

reemplazando las ecuaciones anteriores en la Ec.3.4.1 se tiene:

$$GohG_p(z) = \frac{z-1}{z} \frac{\frac{-T^2}{2} \left(\frac{z^{-1}+z^{-2}}{(1-z^{-1})^3} \right)}{L' M - g M \frac{T^2}{12} \frac{1+10z^{-1}+z^{-2}}{(1-z^{-1})^2}}$$

simplificando:

$$GohG_p(z) = \frac{-6T^2(z^{-1}+z^{-2})}{(12L' M - gMT^2) + (-24L' M - 10gMT^2)z^{-1} + (12L' M - gMT^2)z^{-2}}$$

¹ MONTOYA Omar, "Desarrollo del modelo flexo base-C y aplicación al control de temperatura de una termoselladora mediante lógica Fuzzy", tesis de grado, 1996, pag. 27 - 29

entonces la función de Transferencia queda:

$$\frac{\Phi(z)}{U(z)} = \frac{-6T^2 z^{-1} + (-6T^2)z^{-2}}{(12L' M - gMT^2) + (-24L' M - 10gMT^2)z^{-1} + (12L' M - gMT^2)z^{-2}}$$

haciendo:

$$\begin{aligned} b_1 &= -6T^2 \\ c_0 &= 12L' M - gMT^2 \\ c_1 &= -24L' M - 10gMT^2 \\ c_2 &= 12L' M - gMT^2 \end{aligned}$$

queda:

$$\frac{\Phi(z)}{U(z)} = \frac{b_1 z^{-1} + b_2 z^{-2}}{c_0 + c_1 z^{-1} + c_2 z^{-2}}$$

desarrollando la ecuación anterior:

$$\Phi(z)c_0 + \Phi(z)c_1 z^{-1} + \Phi(z)c_2 z^{-2} = b_1 z^{-1}U(z) + b_2 z^{-2}U(z)$$

transformando a ecuación de diferencias:

$$\Phi(K)c_0 + c_1\Phi(K-1) + c_2\Phi(K-2) = b_1U(K-1) + b_2U(K-2)$$

despejando se tiene la ecuación en diferencias del ángulo :

$$\Phi(K) = \frac{b_1U(K-1) + b_2U(K-2) - c_1\Phi(K-1) - c_2\Phi(K-2)}{c_0} \quad (\text{Ec.3.4.5})$$

La ecuación aproximada de la velocidad angular se obtiene a partir de la siguiente ecuación:

$$\omega(K) = \frac{\Phi(K) - \Phi(K-1)}{T} \quad (\text{Ec.3.4.6})$$

De igual manera se discretiza la función de transferencia de la posición (Ec.2.2.13):

$$\frac{X(s)}{U(s)} = \frac{1}{Ms^2}$$

$$\frac{X(z)}{U(z)} = \frac{z-1}{z} Z \left\{ \frac{1}{s} \frac{1}{Ms^2} \right\}$$

resolviendo:

$$\frac{X(z)}{U(z)} = \frac{z-1}{z} Z \left\{ \frac{s^{-3}}{M} \right\}$$

reemplazando la Ec.3.4.4 en la ecuación anterior se tiene:

$$\frac{X(z)}{U(z)} = (1-z^{-1}) \frac{T^2 / 2 \ z^{-1} + z^{-2}}{M \ (1-z^{-1})^3}$$

resolviendo:

$$\frac{X(z)}{U(z)} = \frac{T^2 \ z^{-1} + z^{-2}}{2M \ (1-z^{-1})^2}$$

$$\frac{X(z)}{U(z)} = \frac{T^2 z^{-1} + T^2 z^{-2}}{2M - 4Mz^{-1} + 2Mz^{-2}}$$

haciendo:

$$f_1 = T^2$$

$$g_0 = 2M$$

$$g_1 = -4M$$

entonces queda:

$$\frac{X(z)}{U(z)} = \frac{f_1 z^{-1} + f_1 z^{-2}}{g_0 + g_1 z^{-1} + g_0 z^{-2}}$$

transformando a ecuación en diferencias y despejando la posición queda :

$$X(K) = \frac{f_1 U(K-1) + f_1 U(K-2) - g_1 X(K-1) - g_0 X(K-2)}{g_0} \quad (\text{Ec.3.4.7})$$

La ecuación aproximada de la velocidad del carro es:

$$V(K) = \frac{X(K) - X(K-1)}{T} \quad (\text{Ec.3.4.8})$$

donde el período de muestreo T debe ser escogido lo suficientemente pequeño para tener una buena aproximación al modelo real.

Una vez discretizado el sistema los pasos a seguir para aplicar un Control Fuzzy son:

1) Conocimiento a priori del Comportamiento de la Planta

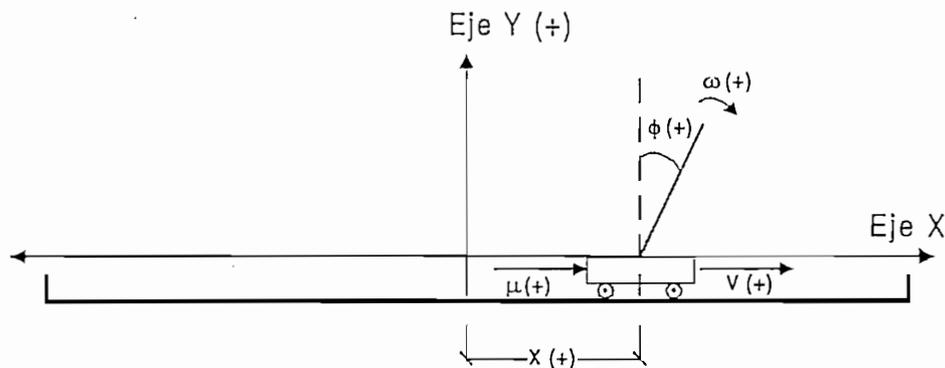


Fig.3.4.2 Definición de Signos Aplicados a las Variables de Entrada y Salida

1.1) Como ya se definió anteriormente (Capítulo II), los signos de las variables son:

- Angulo ($\Phi(t)$) positivo en sentido horario (de la vertical hacia la derecha)
- Velocidad angular ($\omega(t)$) positiva en sentido horario
- Fuerza ($\mu(t)$) positiva hacia la derecha
- Desplazamiento ($x(t)$) positivo desde el centro de la pista hacia la derecha
- Velocidad ($v(t)$) del carro positiva hacia la derecha

1.2) Ahora, las consideraciones generales para el control (comportamiento cualitativo del sistema) son:

- Si existe un ángulo positivo para ser corregido, se debe aplicar una fuerza positiva y mientras más grande sea el ángulo la fuerza a aplicarse debe también aumentar y viceversa.
- Si el ángulo es cero y existe una velocidad angular positiva la cual desestabiliza al péndulo, es necesario aplicar una fuerza positiva para estabilizarlo y mientras mayor sea la velocidad angular la fuerza también debe ser mayor y viceversa.

Con estas ideas se expresa un conocimiento que podía haber sido proporcionado por un proyectista y/o operadores que conocen como trabaja el sistema. Estas ideas se utilizarán para plantear la base de conocimiento de la cual se hablará más adelante.

2) Definición de las Variables Físicas a ser Tomadas en Cuenta para el Control

Este punto a su vez se lo puede dividir en cuatro pasos:

2.1) *Definición de las Variables Lingüísticas*

Pueden ser variables controladas y de control y para el presente sistema del péndulo invertido se las clasificó como:

Variables controladas:

- Angulo, representado por la variable lingüística **Ang**

- Velocidad angular (variación del ángulo en instantes sucesivos de tiempo), representada por la variable lingüística **Wang**

Variable de control:

- Fuerza, representada por la variable lingüística **Fuerza**

2.2) Definición de la Cantidad de Etiquetas Fuzzy para cada Variable Lingüística

Estas etiquetas son llamadas también números fuzzy, variables asociadas o conjuntos fuzzy asociados y se han escogido 5 etiquetas asociadas a cada una de las variables lingüísticas.

El escoger un número impar de etiquetas fuzzy da buenos resultados pues, permite dividir el rango que cubre cada variable lingüística de manera simétrica respecto al cero, permitiendo definir de mejor forma el “set point”. Si se aumenta la cantidad, manteniendo el mismo rango, se disminuyen los intervalos de pertenencia de cada variable asociada, logrando así un control más suave a expensas de aumentar en forma considerable el número de reglas necesarias que conforman la base de conocimiento, como se verá más adelante.

A continuación se presenta la gráfica que correspondería a cada una de las variables lingüísticas, con sus 5 etiquetas fuzzy:

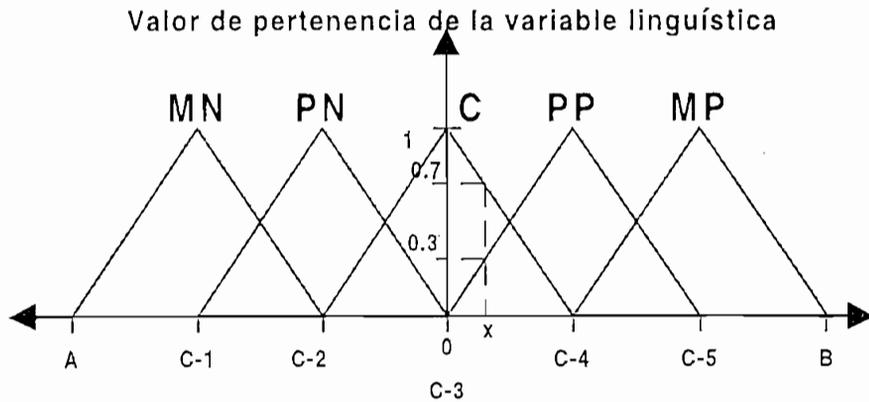


Fig.3.4.3 Representación Gráfica de las Variables Lingüísticas con sus cinco Etiquetas Fuzzy con un Solapamiento del 50%. Segmento [A,C-2] es el Intervalo de Pertenencia de la Etiqueta Fuzzy MN

Donde a cada variable lingüística se le asocia las siguientes etiquetas fuzzy:

MN= Muy Negativo (Representado por 1 en el programa)

PN = Poco Negativo (Representado por 2 en el programa)

C: = Cero (Representado por 3 en el programa)

PP = Poco Positivo (Representado por 4 en el programa)

MP = Muy Positivo (Representado por 5 en el programa)

Se ha escogido la forma triangular para cada etiqueta fuzzy, pues como así se facilita el cálculo del centroide lo cual se verá más adelante, además es la forma más usada en la práctica. A cada etiqueta fuzzy se le ha asignado un número entero, el cual ayuda en la implementación del programa desarrollado para identificar la etiqueta que se está tratando.

Otro punto importante a tomar en cuenta es el grado de solapamiento que existe entre las etiquetas fuzzy (área común de las etiquetas fuzzy), como

se puede ver en la Fig.3.4.3 se tiene un grado de solapamiento del 50% de área, con lo cual se consigue la “suavidad” característica del Control Fuzzy, pues los límites de acción de cada etiqueta fuzzy no están claramente definidos, y así la decisión del controlador (fuerza) se tomará por consenso de las etiquetas fuzzy que intervengan en la toma de decisión; además con este solapamiento se cumple la condición de que para cualquier valor de las variables de entrada/salida dentro del rango tenga valores de pertenencia que sumados sean igual a 1 (normalización). Para el ejemplo de la Fig.3.4.3 se tiene:

- Valor de pertenencia del número x en la etiqueta C es igual a 0.7 =>
 $R_C(x) = 0.7$
- Valor de pertenencia del número x en la etiqueta PP es igual a 0.3
=> $R_{pp}(x) = 0.3$

donde, la suma de los valores de pertenencia es igual a uno:

$$R_C(x) + R_{pp}(x) = 1$$

2.3) Definición del Rango Asociados a cada Variable Lingüística

Se entiende por rango a la distancia que existe entre los puntos máximo y mínimo del eje horizontal de la variable lingüística; como se puede ver en la Fig.3.4.3, la distancia AB es el rango.

El programa desarrollado permite ingresar como dato el rango para cada una de las variables lingüísticas, con lo cual se puede observar el comportamiento del sistema para los diferentes rangos que se ingresen.

Además en el programa desarrollado, el rango siempre será considerado simétrico con respecto al cero, como se muestra en la Fig.3.4.3.

2.4) Definición de los Límites de Acción de cada Etiqueta Fuzzy dentro del Rango

Como se puede observar en el gráfico de la Fig.3.4.3, las etiquetas fuzzy de la variable lingüística han sido distribuidas uniformemente dentro del rango considerado, lo que facilita bastante los cálculos, pues se puede encontrar una expresión que defina el ancho de la base de cada triángulo en función del rango, de la cantidad de etiquetas fuzzy y del grado de solapamiento de estos.

$$Rango = BaseTriangulo - \frac{SOLAP}{100} * BaseTriangulo + BaseTriangulo - \frac{SOLAP}{100} * BaseTriangulo + \dots + BaseTriangulo$$

$$Rango = BaseTriangulo * CNF - (CNF - 1) \frac{SOLAP}{100} * BaseTriangulo$$

$$BaseTriangulo = \frac{Rango}{CNF - (CNF - 1) \frac{SOLAP}{100}} \quad (Ec.3.4.9)$$

donde:

BaseTriangulo = es el ancho de la base de cada etiqueta fuzzy

CNF = es la cantidad de etiquetas fuzzy (en el programa desarrollado es igual a 5)

SOLAP = es el grado de solapamiento que existe entre las etiquetas fuzzy (en el programa es igual a 50 %)

Rango = es el rango considerado para cada una de las variables lingüísticas

3) Elaboración de la Base de Conocimiento.

La base de conocimiento está formada por un conjunto de reglas que no son numéricas, las cuales describen las acciones de control para el sistema, en base a su relación con las variables lingüísticas de entrada (ángulo y velocidad angular) y sus etiquetas Fuzzy.

Estas reglas son frecuentemente llamadas “Declaraciones de Condición Fuzzy” o “Reglas de Control Fuzzy”.

La sintaxis de las Reglas de Control Fuzzy es:

Si Condición_1 y Condición_2 entonces Acción

donde **Condición_1 y Condición_2** es llamado antecedente y la **Acción** es llamada consecuente.

A continuación se da un ejemplo:

Si Ang es MN y Wang es PN entonces Fuerza es MN

Aplicando la sintaxis mencionada y con el conocimiento a priori de la planta, se llega a determinar la base de conocimiento como se indica en la Fig.3.4.4.

		ANGULO				
		NEGATIVO			POSITIVO	
		MN	PN	C	PP	MP
VELOCIDAD ANGULAR	NEGATIVO					
	MN	MN	MN	MN	C	C
	PN	MN	PN	PN	C	C
POSITIVO	C	MN	PN	C	PP	MP
	PP	C	C	PP	PP	MP
	MP	C	C	MP	MP	MP

Fuerza

Fig.3.4.4. Base de Conocimiento para el Control Fuzzy

Si se considera que por el solapamiento existente entre las etiquetas fuzzy de cada variable lingüística de entrada (ángulo y velocidad angular) se pueden activar dos reglas por variable, entonces puede haber un máximo de cuatro reglas activas a la vez, para este caso la acción de control es calculada por el método del centroide de las áreas cuyas reglas de salida hayan sido activas, este método se explicará más adelante.

Además de las reglas anteriores es necesario definir lo que se conoce como **Metareglas**, las cuales definen las estrategias de control cuando las entradas (ángulo y/o velocidad angular) están fuera del rango establecido de control.

El número de etiquetas fuzzy determina la cantidad de reglas fuzzy necesarias, como puede verse en la Fig.3.4.4, para este control se tienen definidas 25 reglas.

4) Por Ultimo, Probar el Modelo para Verificar su Fidelidad con el Proceso Real.

En un proceso real, en el cual el Control Fuzzy modela las acciones de control a partir del la base de conocimiento se lo puede representar como se ve en la Fig.3.4.5.

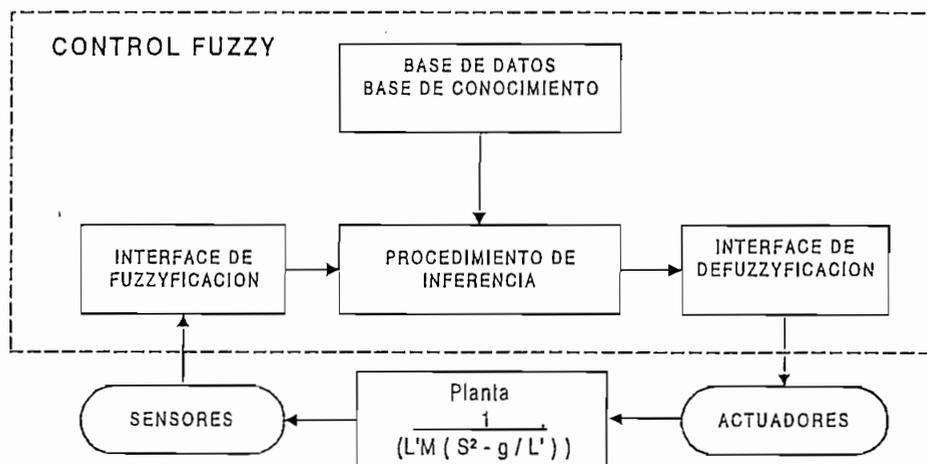


Fig.3.4.5.- Diagrama de Bloques del Control Fuzzy Aplicado al Péndulo Invertido

A continuación se detalla el proceso del Control Fuzzy:

4.1) Interface de Fuzzificación

En esta etapa se toman los valores de las variables de entrada (ángulo y velocidad angular) y se hace un escalamiento para acondicionar los valores a

universos de discurso normalizados, (fuzzificación de valores); esto implica transformar los simples números en etiquetas fuzzy, de modo que pasan a tomar instancias de variables lingüísticas.

Para esto, se presenta un ejemplo con el cual se entenderá de mejor manera como trabaja el Control Fuzzy:

Rangos de las variables lingüísticas:

Rango del ángulo = 30grad

Rango de la velocidad = 129grad/s

Rango de la fuerza = 60N

La base del triángulo de las etiquetas fuzzy para cada uno de las variables lingüísticas se calcula usando la Ec.3.4.9

Base del triángulo del ángulo $\hat{=} 10\text{grad/s}$

Base del triángulo de la velocidad angular = 43grad

Base del triángulo de la fuerza = 20N

Datos a fuzzificar:

Angulo = 3,5grad

Velocidad angular = -4,3grad/s

Con los valores ingresados de ángulo y velocidad angular, se realiza la fuzzificación con lo cual se obtiene el grado de pertenencia para cada etiqueta fuzzy, así se tiene:

Para el ángulo (ver Fig.3.4.3):

$$\mathbf{Rang}_C(3,5) = 0,3$$

$$\mathbf{Rang}_{PP}(3,5) = 0,7$$

De forma análoga se encuentra para la velocidad angular:

$$\mathbf{Rwang}_C(-4,3) = 0,8$$

$$\mathbf{Rwang}_{PN}(-4,3) = 0,2$$

Estos resultados se los presenta en forma matricial, la cual ayuda a manejar de mejor manera los valores de pertenencia asociados a cada etiqueta fuzzy:

Se define las siguientes matrices:

\mathbf{Rang} = matriz de los valores de pertenencia de la variable ángulo

\mathbf{Rwang} = matriz de los valores de pertenencia de la variable velocidad angular.

$\mathbf{Rfuerza}$ = matriz de los valores de pertenencia de la variable fuerza.

	MN	PN	C	PP	MP
$\mathbf{Rang} = [$	0	0	0,3	0,7	0]
$\mathbf{Rwang}=[$	0	0,2	0,8	0	0]

Se entiende que el primer elemento de la matriz \mathbf{Rang} (0), representa el valor de pertenencia en la etiqueta MN del valor del ángulo a fuzzificar, el segundo término (0) representa el valor de pertenencia en la etiqueta PN y así sucesivamente.

4.2) Mecanismo de Inferencia.

Procesa los datos que se obtienen de la fuzzificación junto con las reglas definidas en la base de conocimiento, de modo de inferir las acciones de Control Fuzzy, aplicando el operador de implicación fuzzy y las reglas generales de inferencia de lógica fuzzy, en este caso el resultado de la inferencia para una regla es obtenida por el mínimo valor de los grado de pertenencia de las condiciones, pues el antecedente esta formado por dos condiciones relacionadas con el operador AND (Y).

En el ejemplo de la Fig.3.4.4 se puede observar las cuatro reglas activas (sombreadadas) que se han validado:

- Regla₈: Si Ang es C y Wang es PN entonces Fuerza es PN
- Regla₉: Si Ang es PP y Wang es PN entonces Fuerza es C
- Regla₁₃: Si Ang es C y Wang es C entonces Fuerza es C
- Regla₁₄: Si Ang es PP y Wang es C entonces Fuerza es PP

A continuación se presenta la gráfica del proceso de inferencia:

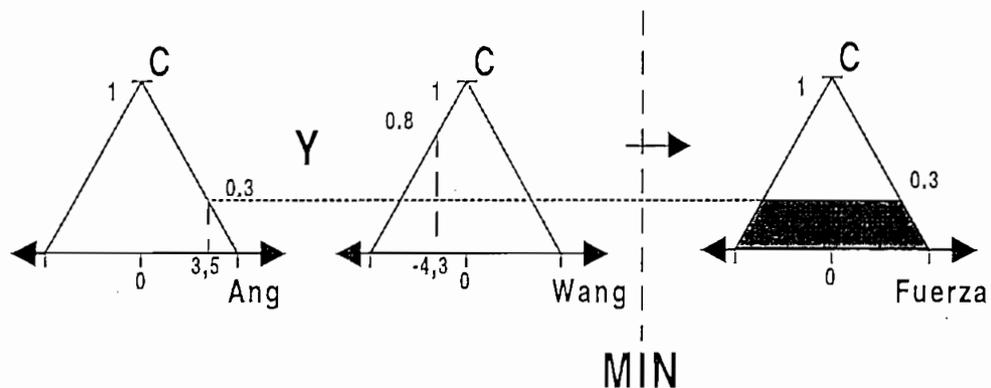


Fig.3.4.6 Mecanismo de Inferencia Fuzzy

Los resultados obtenidos de la inferencia para el ejemplo son:

$$\text{Regla}_8 : \min(\text{Rang}(3) \text{ Y } \text{Rwang}(2)) = \min [(0,3) \wedge (0,2)] \Rightarrow R_{PN}(\mu) = 0,2$$

$$\text{Regla}_9 : \min(\text{Rang}(4) \text{ Y } \text{Rwang}(2)) = \min [(0,7) \wedge (0,2)] \Rightarrow R_C(\mu) = 0,2$$

$$\text{Regla}_{13} : \min(\text{Rang}(3) \text{ Y } \text{Rwang}(3)) = \min [(0,3) \wedge (0,8)] \Rightarrow R_C(\mu) = 0,3$$

$$\text{Regla}_{14} : \min(\text{Rang}(4) \text{ Y } \text{Rwang}(3)) = \min [(0,7) \wedge (0,8)] \Rightarrow R_{PP}(\mu) = 0,7$$

En los resultados anteriores dos reglas activadas tienen el mismo consecuente (en las reglas 13 y 9 la acción de control es C), en este caso el valor mayor es el que se debe considerar.

Entonces la matriz de la variable de salida (fuerza) queda expresada de la siguiente manera:

	MN	PN	C	PP	MP
Rfuerza=[0	0,2	0,3	0,7	0]

4.3) Interface de Defuzzificación

Transforma las acciones de Control Fuzzy en acciones de Control No-Fuzzy, esto significa efectuar un escalamiento, de modo de compatibilizar los valores normalizados que vienen del proceso de inferencia en una señal de control a ser enviada al proceso.

La acción final de control es calculada por la unión de las contribuciones proporcionadas por cada regla activa provenientes de la inferencia como se muestra en la Fig.3.4.7.

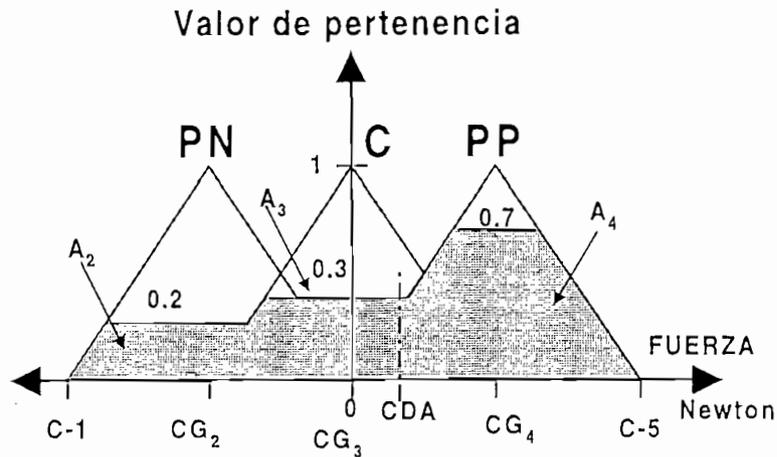


Fig.3.4.7 Area Solución Obtenida en el Proceso de Inferencia, para ser Defuzzificada

A pesar de no haber un proceso sistemático para escoger la estrategia de defuzzificación el más común es el criterio del centro del área, del cual en base a la experiencia ha dado buenos resultados:

4.3.1) Criterio del Centroide o Centro de Area (CDA):

El método calcula el centro de gravedad del espacio solución formado por los conjuntos de salida resultantes de la inferencia (ver Fig.3.4.7).

Este método da buenos resultados, pues, toma en cuenta todas las reglas activas, sus aportes individuales, la forma geométrica del conjunto de salida,

además obtiene una señal de control suave y continua sin saltos bruscos.

Un elemento importante a tomar en cuenta en cada etiqueta fuzzy es el centroide o centro de gravedad respecto al eje Y que en este caso se encuentra ubicado en el centro de cada triángulo, esto es, en la mitad de la base, puesto que este servirá para calcular el centroide de la superficie solución (ver Fig.3.4.7) que resultó del proceso de inferencia.

El centroide con respecto al eje 'Y' de una superficie se define como:

$$CDA = \frac{M_y}{A} = \frac{\int x \cdot dA}{A}$$

donde:

M_y = es el primer momento con respecto al eje "Y"

A = es el área

x = distancia desde el origen hasta el diferencial del área

El centroide de una superficie es independiente de los ejes de referencia empleados, pues es, solamente una propiedad de la superficie en sí misma. Por lo tanto, si una superficie puede subdividirse en superficies componentes simples, cuyos centroides se obtienen por simple inspección, fácilmente se puede determinar el momento de toda la superficie con respecto a un eje, en el plano; sumando los primeros momentos de las componentes. Donde cada momento es el producto del área multiplicado por la distancia centroidal correspondiente¹.

¹ IRVING H. Shames, Ingeniería Mecánica, Estática, tomo 1, reimpresión 1979. pag. 219-221

Para el ejemplo de la Fig.3.4.7 se tiene una superficie formada por superficies más simples solapadas, donde el primer momento del área sombreada con respecto al eje 'Y' sería:

$$M_y = CG_2 * A_2 + CG_3 * A_3 + CG_4 * A_4$$

Para facilitar los cálculos se aplica la correlación producto en el cálculo de los centroides en vez de la correlación mínima².

La correlación producto se define como:

Área de la superficie original:

$$y = f(x) \implies A_o = \int f(x) dx$$

Área de la superficie modificada:

$$y_m = r * f(x) \implies A_m = r * \int f(x) dx$$

Así, se presenta una relación entre las áreas del triángulo original y el triángulo modificado al que se le ha aplicado la técnica de correlación producto:

$$A_m = r * A_o$$

De esta manera se puede generalizar el cálculo del centroide para una superficie de salida con múltiples reglas activas solapadas:

$$CDA = \frac{\sum CG_i * r_i * A_i}{\sum r_i * A_i}$$

Todas las áreas son iguales por lo que la ecuación se simplifica a:

$$CDA = \frac{\sum CG_i * r_i}{\sum r_i}$$

² MONTROYA Omar, "Desarrollo del modelo flexo base-C y aplicación al control de temperatura de una termoselladora mediante lógica Fuzzy" tesis de grado, 1996.

Como se han definido cinco etiquetas fuzzy la expresión general que calcula el centroide de la superficie solución queda expresada como:

$$CDA = \frac{CG_1 * r_1 + CG_2 * r_2 + CG_3 * r_3 + CG_4 * r_4 + CG_5 * r_5}{r_1 + r_2 + r_3 + r_4 + r_5}$$

donde:

r_i = son los elementos de la matriz R fuerza.

CG_i = son los centroides de cada uno de las etiquetas fuzzy de la variable fuerza.

CDA= es el centroide o centro de área de la superficie resultante de la inferencia.

para el ejemplo el centro de área es:

$$CDA = \frac{-20 * 0 - 10 * 0,2 + 0 * 0,3 + 10 * 0,7 + 20 * 0}{0 + 0,2 + 0,3 + 0,7 + 0} = 4,166$$

$$CDA = 4,166N$$

Así se ha encontrado el valor de la señal de control a ser enviada a la planta, obteniéndose con esto una nueva lectura de ángulo y velocidad angular a ser fuzzificados (utilizando las Ec.3.4.5 y 3.4.6) repitiéndose nuevamente el proceso, hasta conseguir equilibrar al péndulo.

Análisis de Resultados del Control de Fuzzy

Una vez que se ha presentado el método para el desarrollo del Control Fuzzy, se procede a realizar el análisis de resultados usando las mismas definiciones empleadas en los controles anteriores, así se tiene:

- Tiempo de crecimiento: t_r
- Tiempo en alcanzar el primer pico : t_p
- Tiempo de establecimiento criterio del 2 % : t_s
- Tiempo del primer cambio de dirección de la velocidad del carro: t_{cvc}
- Tiempo del primer cambio de dirección de la fuerza: t_{cf}
- Tiempo del primer cambio de dirección de la velocidad angular: t_{cva}
- Valor pico de la primera oscilación: V_p
- Máximo sobreimpulso: M_p
- Fuerza máxima aplicada: f_M
- Espacio recorrido por el carro a t_s del ángulo : X_{t_s}
- Velocidad angular en el primer punto donde el ángulo es cero : $W_{\phi=0}$
- Angulo en el primer punto donde la velocidad angular es cero : $\Phi_{W=0}$
- Velocidad final con la que el carro queda moviéndose después de estabilizar al péndulo: V_f

Siguiendo el mismo modelo empleado en los controles anteriores se tiene:

Datos generales del péndulo

$$M = 1\text{Kg}$$

$$m = 0,1\text{Kg}$$

$$L = 0,5\text{m}$$

$$g = 9,8\text{m/s}^2$$

$$\text{Longitud de la pista} = 1\text{m}$$

Condiciones Iniciales

$$\text{Espacio} = 0\text{m}$$

$$\text{Velocidad} = 0\text{m/s}$$

$$\text{Ángulo} = 10\text{grad}$$

$$\text{Vel. Angular} = 0\text{grad/s}$$

El ingreso de datos para este control se lo realiza en dos ventanas:

En la primera ventana, se ingresan los valores de rango para las variables ángulo, velocidad angular y fuerza, así como también el tiempo de simulación. El periodo de muestreo, es calculado dividiendo el tiempo de simulación para mil, por lo tanto para tener un adecuada precisión es recomendable que el tiempo de simulación, no sea muy grande.

En la segunda ventana se ingresa la base de conocimiento.

Los datos ingresados para este control:

Rango de la fuerza=60N

Rango de ángulo = 30grad

Rango de la velocidad angular=150grad/s

Tiempo de simulación = 2seg

Tiempo de muestreo = 0,002seg

Base de conocimiento

		ANGULO					
		MN	PN	C	PP	MP	
VELOCIDAD ANGULAR	MN	MN	MN	MN	C	C	MN = Muy Negativo
	PN	MN	PN	PN	PP	PP	PN = Poco Negativo
	C	MN	PN	C	PP	MP	C = Cero
	PP	PN	PN	PP	PP	MP	PP = Poco Positivo
	MP	C	C	MP	MP	MP	MP = Muy Positivo

Fig.3.4.8. Base de Conocimientos Ingresada como Dato para el Control Fuzzy

Con estos datos y con la ayuda del programa desarrollado se obtienen las siguientes curvas:

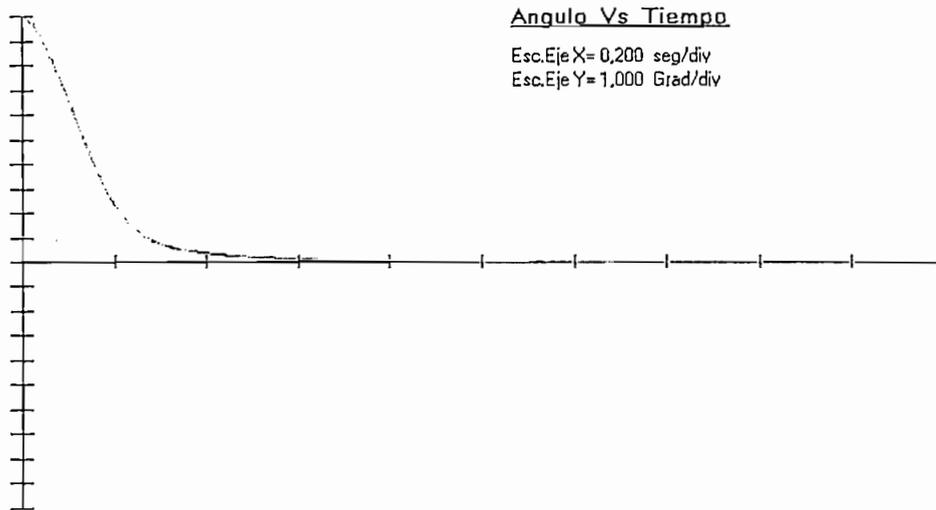


Fig.3.4.9 Gráfica del Ángulo Vs. Tiempo para el Control Fuzzy

Datos de la Fig.3.4.9:

$$t_r = 0,67\text{seg}$$

Como puede verse en la Fig.3.4.9 la curva no presenta un máximo sobrepulso, teniendo una curva característica de un sistema de sobreamortiguado. Para este ejemplo el ángulo converge a cero sin tomar valores negativos.

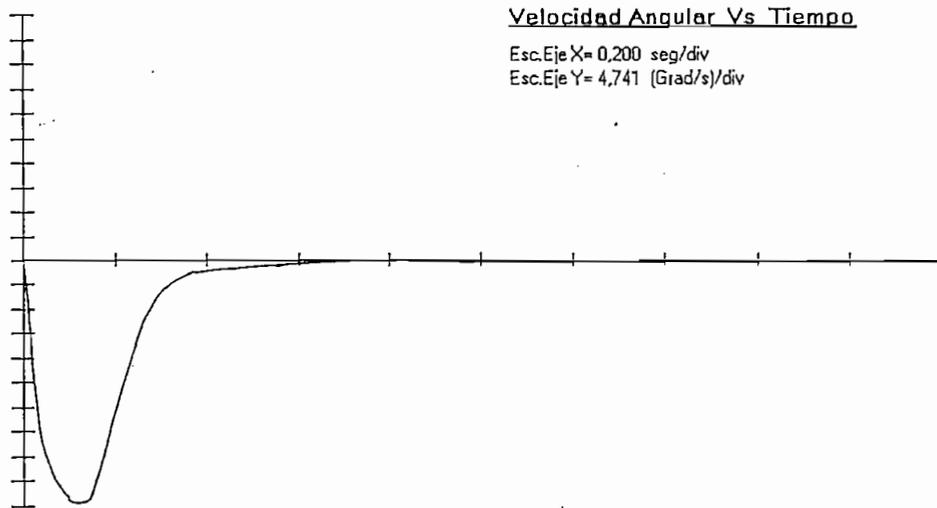


Fig.3.4.10 Gráfica de la Velocidad angular Vs. Tiempo para el Control Fuzzy

Datos de la Fig.3.4.10:

$$t_p = 0,128\text{seg}$$

$$t_s = 0,562\text{seg}$$

$$V_p = -47,41\text{grad/s}$$

De la Fig.3.4.10 se observa que la velocidad no se hace positiva en ningún momento, sino que converge suavemente a cero.

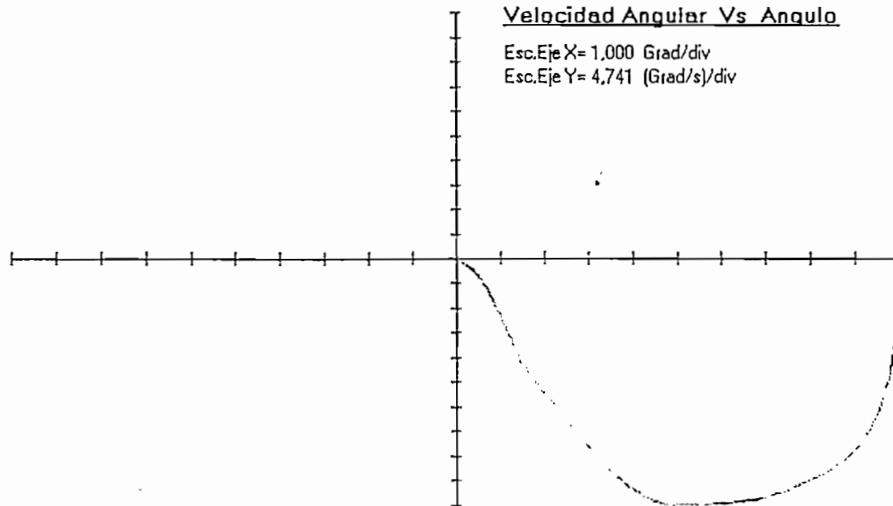


Fig.3.4.11 Gráfica de la Velocidad angular Vs. Angulo para el Control Fuzzy

La Fig.3.4.11 muestra que el ángulo y la velocidad angular convergen a cero y que en ningún momento cambian de sentido.

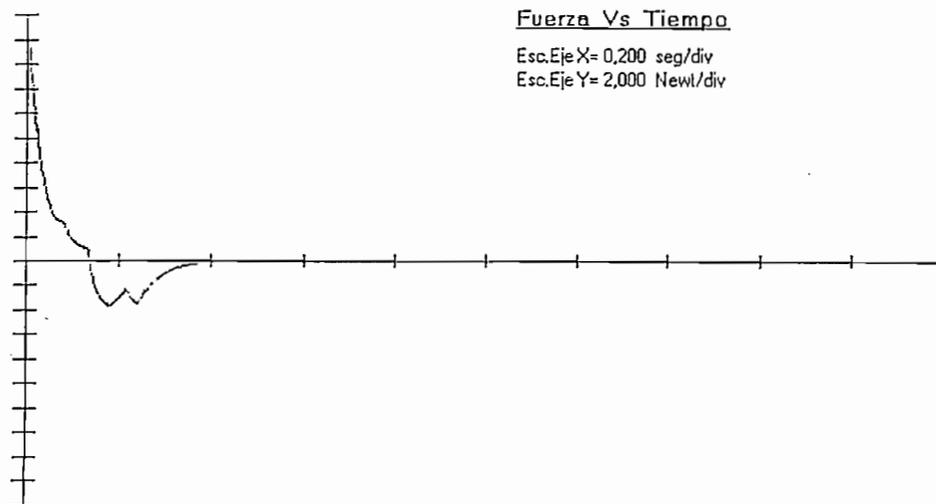


Fig.3.4.12 Gráfica de Fuerza Vs. Tiempo para el Control Fuzzy

Datos de la Fig.3.4.12:

$$t_{cf} = 0,135\text{seg}$$

$$t_p = 0,18\text{seg}$$

$$t_s = 0,343\text{seg}$$

$$V_p = -3,624\text{N}$$

$$f_M = 20\text{N}$$

De la gráfica de la fuerza Fig.3.4.12 se observa que la curva de la fuerza varía suavemente sin cambios bruscos, y su valor de convergencia es cero.

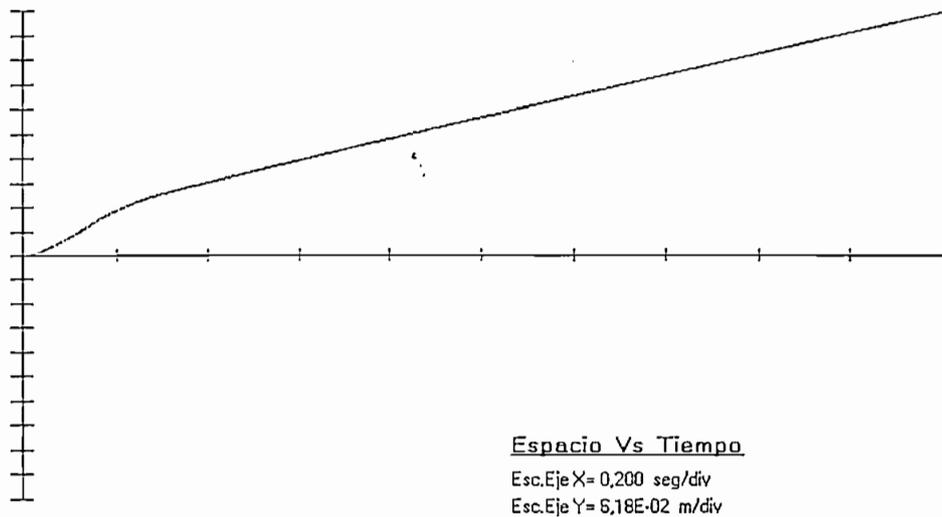


Fig.3.4.13 Gráfica de Espacio Vs. Tiempo para el Control Fuzzy

Datos de la Fig.3.1.13:

$$X_{t_s=0,67\text{seg}} = 0,26\text{m}$$

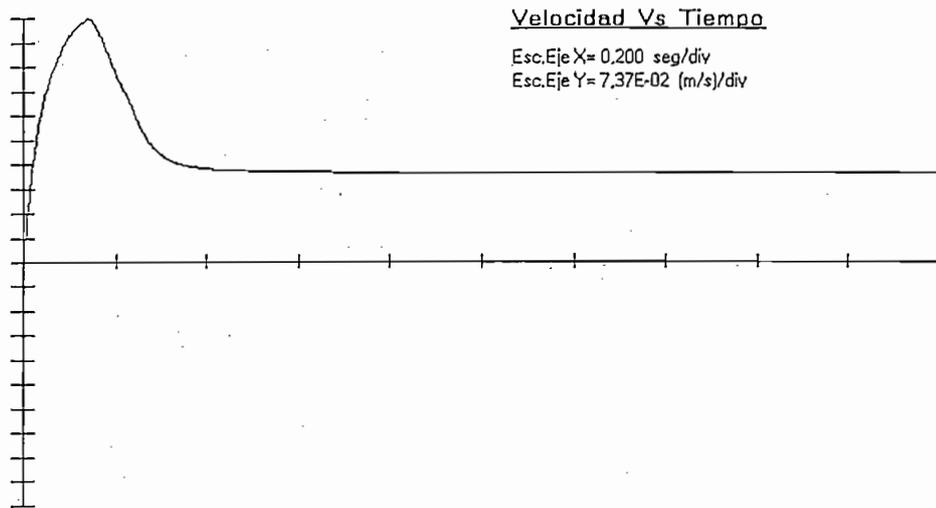


Fig.3.4.14 Gráfica de Velocidad Vs. Tiempo para el Control Fuzzy

Datos de la Fig.3.1.14:

$$t_p = 0,142\text{seg}$$

$$V_p = 0,737\text{m/s}$$

$$V_r = 0,267\text{m/s}$$

De las gráficas Figs.3.4.13 y 3.4.14 se observa que, cuando el control llega a estabilizar el péndulo, la posición del carro es aproximadamente 0,260m del centro de la pista, luego el carrito continúa alejándose del origen con una velocidad constante de 0,267m/s.

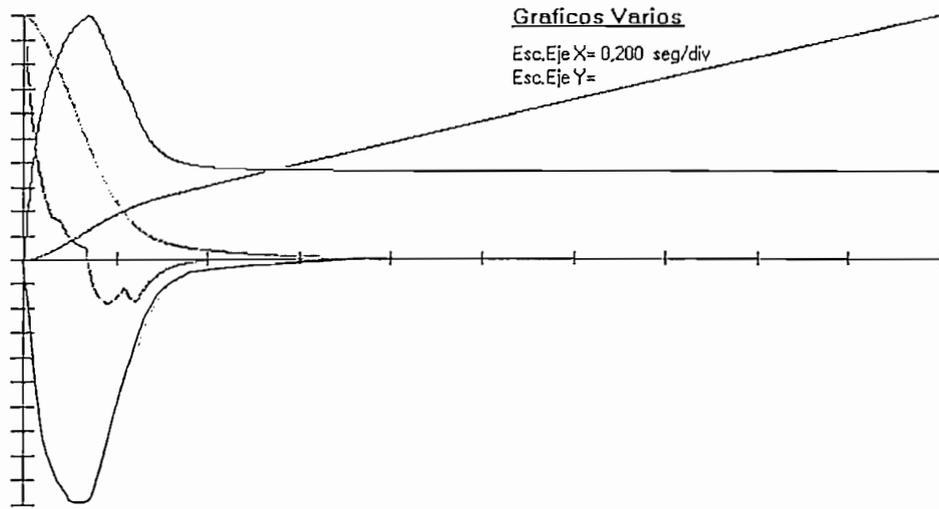


Fig.3.4.15 Gráfica de las Cuatro Variables y la Fuerza Vs. Tiempo para el Control Fuzzy

En la Fig.3.4.15 se observa claramente como las curvas del ángulo, velocidad angular y fuerza convergen a cero, llevando al péndulo a su posición vertical, además como el carro se mueve con movimiento rectilíneo uniforme desde que el ángulo se hace cero.

El programa para este control presenta cuatro ventanas de resultados que se muestran en las Figs.3.4.16, 3.4.17, 3.4.18, 3.4.19.

		ANGULO				
		Tabla				
		MN	PN	C	PP	MP
VELOCIDAD ANGULAR	MN			42	97	55
	PN			934	1001	67
	C			892	904	12
	PP					
	MP					

Fig.3.4.16 Resultados del Control Fuzzy. Ventana que Muestra el Número de veces que han sido Ejecutadas las Reglas.

La ventana de resultados de la Fig.3.4.16 es muy importante para setear los valores de entrada del Control Fuzzy, pues aquí se muestra el número de ocasiones que el control requiere de una regla en particular (conocimiento particular). Además se ve también que reglas no han sido utilizadas debido a que el sistema no cumplió con las condiciones que imponen estas reglas.

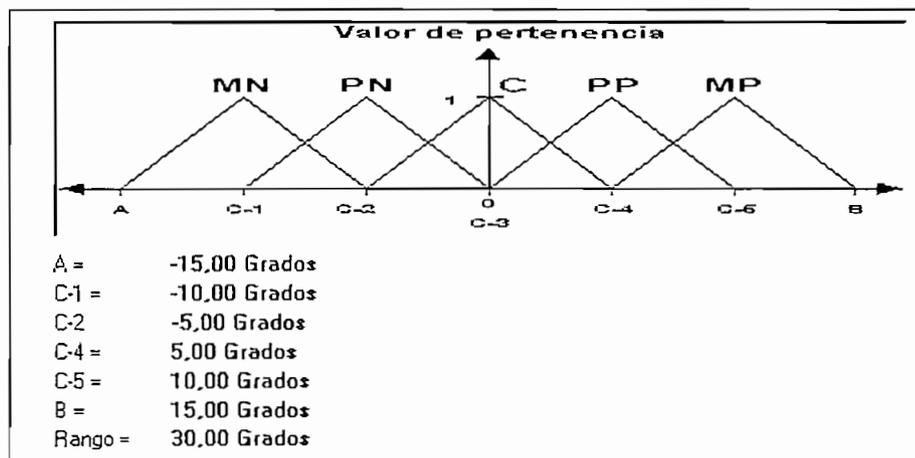


Fig.3.4.17 Resultados del Control Fuzzy. Ventana que Muestra la Distribución de las Etiquetas Fuzzy de la Variable Lingüística Angulo dentro del Rango

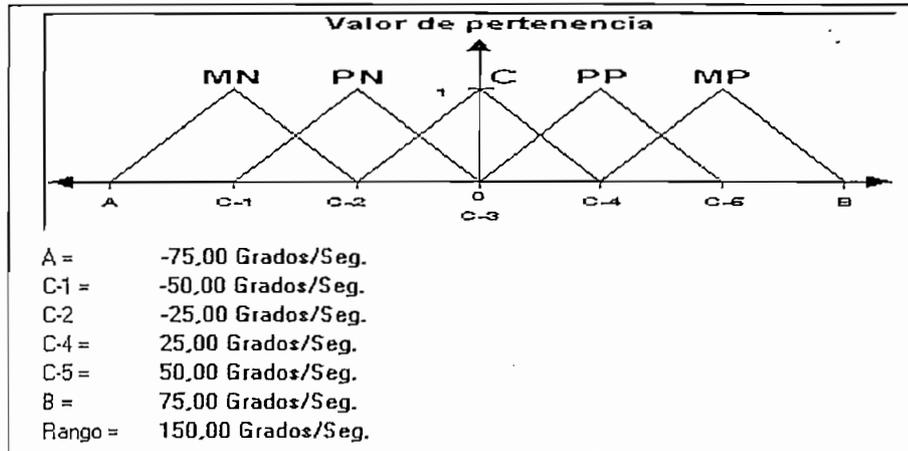


Fig.3.4.18 Resultados del Control Fuzzy. Ventana que Muestra la Distribución de las Etiquetas Fuzzy de la Variable Lingüística Velocidad Angular dentro del Rango

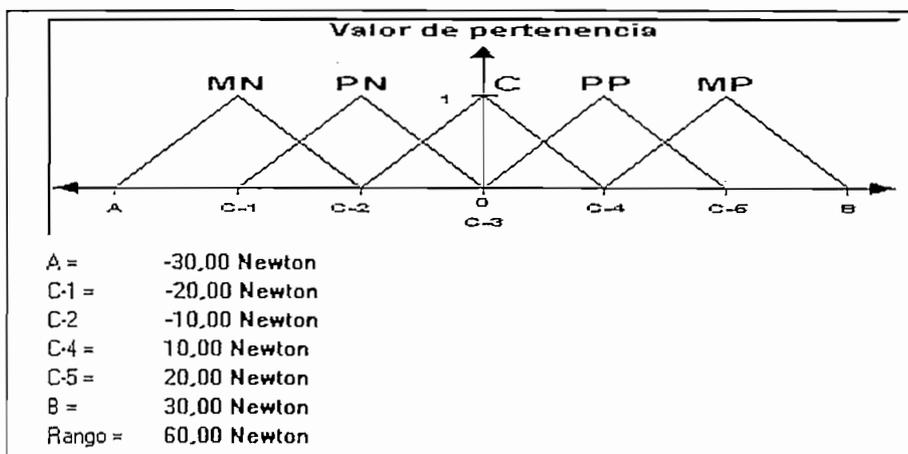


Fig.3.4.19 Resultados del Control Fuzzy. Ventana que Muestra la Distribución de las Etiquetas Fuzzy de la Variable Lingüística Velocidad Angular dentro del Rango

Los cambios que se hagan a la base de conocimiento afectan directamente en la respuesta del sistema; la base de conocimiento expuesta como ejemplo no es la única que llega a equilibrar al péndulo, antes que el carrito sobrepase los límites de la pista.

3.5.- Comparación Inicial de las Técnicas de Control Aplicadas

Después de realizar varios ensayos con las técnicas de compensación implementadas en el programa desarrollado y analizando sus resultados y gráficas, usando los mismos datos generales del péndulo y condiciones iniciales para todos los controles, se puede apreciar como influye el cambio de un parámetro del compensador en el comportamiento del sistema.

Así, la compensación por Red de Adelanto de Fase, utilizando el método del cero arbitrario, permite ubicar de manera más adecuada el cero de la red compensadora, logrando que las raíces complejas de lazo cerrado deseadas sean las dominantes, y por ende que la respuesta transitoria cumpla con las especificaciones de diseño.

Con el programa desarrollado se verifica que mientras más lejos del origen se ubique el cero del compensador de la Red de Adelanto de Fase, aumenta la dominancia de las raíces complejas de lazo cerrado, aumentando también la ganancia del compensador (K_c).

Con las compensaciones, Red de Adelanto de Fase, Acciones de Control y Control Fuzzy, la fuerza que equilibra el péndulo no está en función del espacio, ni de la velocidad del carro, motivo por el cual se observa que estas curvas no necesariamente se hacen cero, quedando el carro con una velocidad constante y alejándose del origen (ver Figs.3.4.14 y 3.4.13) o con velocidad cero y con un cierto valor de distancia (ver Figs.3.2.15 y 3.2.14).

En el control Proporcional Derivativo las raíces complejas conjugadas de lazo cerrado (las cuales son las raíces deseadas), hacen que el comportamiento del sistema satisfaga los requerimientos de diseño, esto se debe a que el Control Proporcional Derivativo únicamente añade un cero a la función de transferencia de la planta y no como en los controles de Red de Adelanto y P.I.D , que aumentan un polo a la función de transferencia el cual dificulta que se cumpla con las condiciones de diseño, pues en este caso las raíces complejas no son necesariamente las dominantes.

En el control por realimentación de estado se observa que todas las gráficas de las variables de estado (ángulo, velocidad angular, posición y velocidad) se hagan cero, esto se debe a que al realimentar las cuatro variables de estado, toma una mayor información del sistema, teniendo así una acción de control (fuerza) que está en función de todo el vector de estado.

En el Control Fuzzy para los datos que se ingresaron el sistema no tiene un comportamiento subamortiguado pues, en el plano de fase (ver Fig.3.4.11) durante la trayectoria al punto de equilibrio, el ángulo y la velocidad angular no cambian de sentido; adicionalmente se observa que se logra equilibrar el péndulo con bastante rapidez (ver Fig.3.4.9).

Para todos los controles se debe advertir, que si, en la gráfica del ángulo, se sobrepasan los 11 grados, las relaciones trigonométricas (Ec.2.2.8. y Ec.2.2.9) de linealidad definidas en el capítulo II, pierden validez, y por ende los resultados obtenidos se alejarán de los datos que se obtuviesen de una planta real.

A excepción del Control Fuzzy, todos los controles hacen que el sistema, sea condicionalmente estable.

CAPITULO IV: DESARROLLO DEL PROGRAMA

4.1.- Simulación Dinámica.

La simulación dinámica se inicia cuando se selecciona la opción “Correr Animación” la cual está dentro del menú “Simulación”; esta opción es posible ejecutar sólo cuando ya se han realizado los cálculos del control seleccionado, pues se usan los valores almacenados de las variables ángulo, velocidad angular, posición, velocidad del carro y fuerza que fueron guardados en cinco arreglos diferentes, en el momento que se seleccionó la opción “Hacer Cálculos”

La simulación dinámica del sistema está formada principalmente por nueve elementos que se muestran en la Fig.4.1.1.

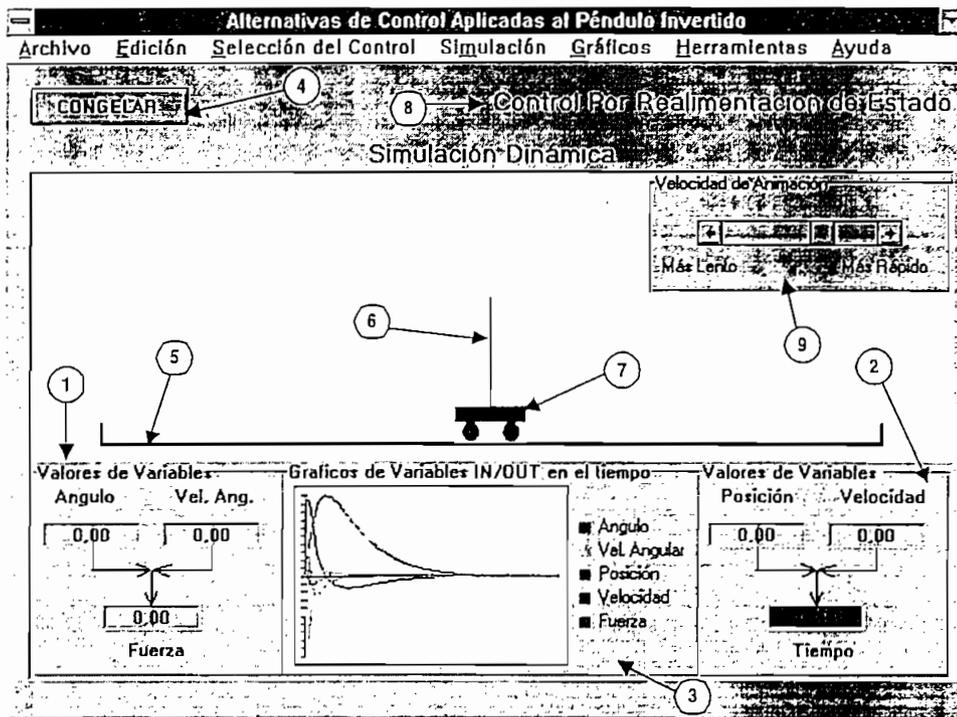


Fig.4.1.1. Pantalla de la Simulación Dinámica

- 1.-En esta ventana se muestra los valores instantáneos que toman las variables ángulo, velocidad angular y fuerza.
- 2.-Muestra los valores instantáneos que toman las variables velocidad y posición del carrito, adicionalmente se tiene un indicador del tiempo transcurrido.
- 3.-Esta ventana permite observar las gráficas de las curvas ángulo, velocidad angular, posición, velocidad y fuerza, las mismas que se van graficando según el tiempo transcurrido. Cada curva es graficada con un color diferente: verde para el ángulo, celeste para la velocidad angular, rojo para la posición, azul para la velocidad y morado para la fuerza.
- 4.-Este botón permite que en cualquier instante se pueda detener o continuar (congelar o descongelar) la animación. Al detener la

simulación se observa como están las diferentes variables y se puede ir viendo como va trabajando el control.

- 5.-La pista se escala dependiendo del valor que se ingrese.
- 6.-La varilla, cuya posición varía de acuerdo a los valores almacenados de ángulo y posición.
- 7.-El carrito el cual se desplaza a lo largo de la pista de acuerdo a los valores almacenados de posición.
- 8.-Etiqueta que indica el tipo de control con el que se está trabajando.
- 9.-En esta ventana se encuentra la barra que permite aumentar o disminuir la velocidad con que se realiza la animación

La animación se realiza en base a un Timer (reloj), el cual hace que se llame a su subrutina a intervalos iguales de tiempo, así el movimiento del carro y el péndulo están determinados por el intervalo de activación del reloj. El reloj además posibilita que se realicen otros eventos tales como el botón de Congelar / Descongelar y la variación de la velocidad de la simulación como se explicará mas adelante.

4.2.- Algoritmos para el Cálculo del Péndulo Invertido.

El diagrama de flujo de la Fig.4.2.1 representa el algoritmo general del programa desarrollado.

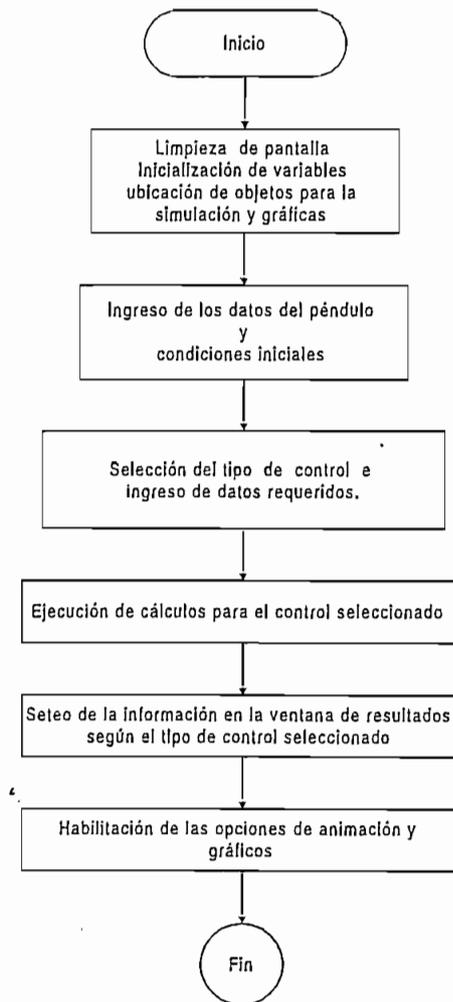


Fig.4.2.1.- Diagrama General del Programa Desarrollado.

Al ejecutar el programa, lo primero que se hace es limpiar la pantalla, luego inicializar los valores preestablecidos del péndulo, condiciones iniciales, y los datos de entrada de cada control, luego ubica los objetos que aparecerán en las opciones de simulación y gráficos.

El programa espera que se ingresen los datos generales y condiciones iniciales del péndulo ya sea con la opción “**Nuevo Péndulo**” o “**Péndulo por Defecto**” del menú “**Archivo**”. La primera opción permite cambiar los datos

del péndulo y condiciones iniciales (ver Fig.4.2.2.) y mantiene los cambios realizados que pudieran haberse hecho en los datos ingresados para cada control, mientras que la segunda opción sólo permite ver los datos generales del péndulo e inicializa los valores preestablecidos para cada control.

Datos Del Pendulo	
Datos Generales	
M =	<input type="text" value="1"/> Kg
m =	<input type="text" value="0,10"/> Kg
L =	<input type="text" value="0,50"/> m
g =	<input type="text" value="9,80"/> m/s ²
Long. Pista =	<input type="text" value="1"/> m
<input type="button" value="MODIFICAR"/>	
Condiciones Iniciales	
Espacio =	<input type="text" value="0"/> m
Velocidad =	<input type="text" value="0"/> m/s
Angulo =	<input type="text" value="10"/> grados
Vel. Ang. =	<input type="text" value="0"/> grados/s
<input type="button" value="OK"/>	
<input type="button" value="CANCEL"/>	

Fig.4.2.2 Datos Generales del Péndulo y Condiciones Iniciales

Luego se escoge el tipo de control con el menú "Selección de Control" donde se encuentran los siguientes controles:

Realimentación de Estado en el cual los datos a ingresar son:
cuatro polos de lazo cerrado

Red de Adelanto de Fase el cual se realiza a través de dos métodos: el de la bisectriz cuyos datos de ingreso son el máximo sobreimpulso porcentual y el tiempo de establecimiento en segundos, y el método del cero arbitrario donde adicionalmente del máximo sobreimpulso y el tiempo de establecimiento se ingresa el cero del compensador.

Acciones de Control P.D. y P.I.D. donde para la acción proporcional derivativa se tiene como datos la frecuencia natural no amortiguada (W_n) y el coeficiente de amortiguamiento (ζ) o las constantes, proporcional (K_p) y derivativa (K_d), mientras que para el control proporcional integral derivativo se ingresan como datos las constantes, proporcional (K_p), derivativa (K_d) e integral (K_i)

Control Fuzzy cuyos datos son los rangos de las variables ángulo, velocidad angular y fuerza, el tiempo de simulación y la base de conocimiento.

Después de haberse seleccionado el tipo de control se realizan los cálculos con la opción "Hacer cálculos" del menú "Simulación" colocando la información más importante en la ventana de resultados y habilitándose las opciones de "Correr animación" y "Gráficos".

En los puntos siguientes se trata con más detalle las rutinas de los cálculos realizados para cada control y el proceso de la simulación dinámica.

4.2.1.- Diagramas de Flujo de la Simulación Dinámica.

La animación del sistema está formada principalmente por dos subrutinas: “mnuAnimacion_Click” y “TimerSim_Timer”, la primera subrutina se la ejecuta con la opción “correr Animación” y ésta a su vez habilita un reloj cuya subrutina es TimerSim_Timer, la misma que se ejecuta a intervalos iguales de tiempo hasta que sea deshabilitado el reloj terminando así la simulación. A continuación se presenta el diagrama de flujo de la subrutina “mnuAnimacion_Click” (ver Fig.4.2.1.1)

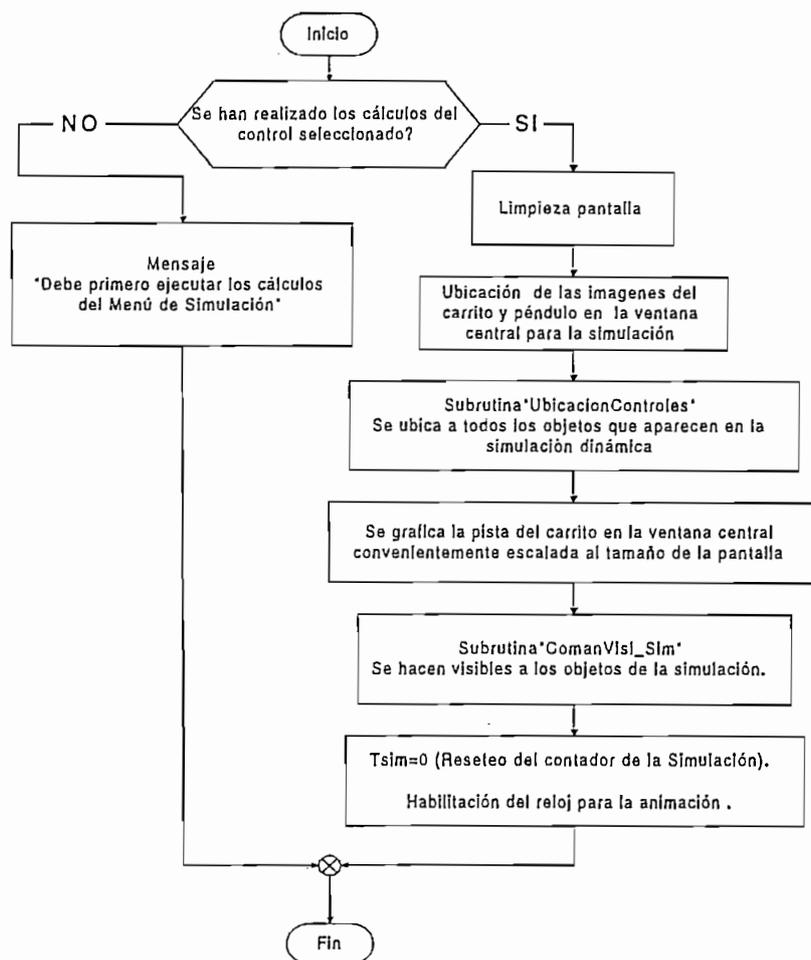


Fig.4.2.1.1.- Diagrama de Flujo de la Subrutina “mnuAnimación_Click” para la Simulación Dinámica

El movimiento del carrito y por ende del péndulo se lo realiza en base de la subrutina "TimerSim_Timer" cuyo diagrama de flujo esta representado en la Fig.4.2.1.2.

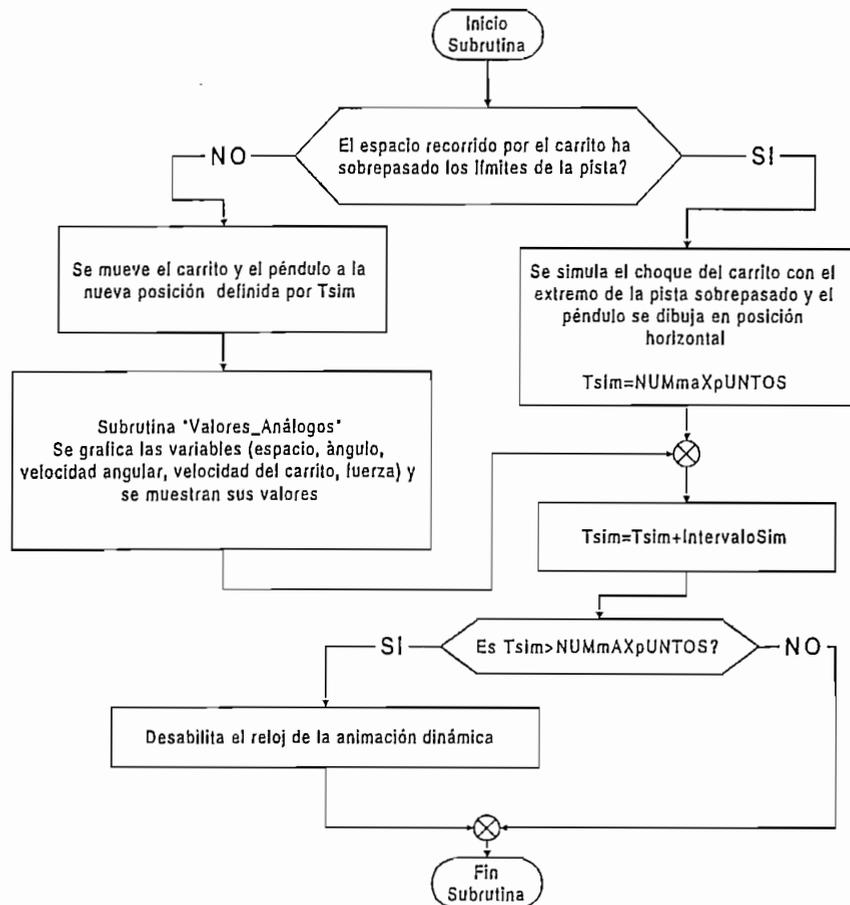


Fig.4.2.1.2.- Diagrama de Flujo de la Subrutina "TimerSim_Timer" para la Animación.

En cada instante que ocurre la interrupción del reloj (Timer) se ejecuta la Subrutina "TimerSim_Timer", la cual cambia la posición de la imagen del carrito al valor que se encuentra almacenado en el arreglo de la variable espacio (este valor fue guardado cuando se realizaron los cálculos del control seleccionado), dando el efecto de movimiento del carro a lo largo de la pista. El extremo superior del péndulo es ubica de acuerdo al valor almacenado del

ángulo, mientras que el extremo inferior del mismo se ubica de acuerdo al valor de espacio almacenado, teniendo como resultado final el movimiento conjunto del carro y el péndulo como se aprecia en la Fig.4.1.1.

Cuando el espacio recorrido por el carro ha sobrepasado los límites de la pista, se simula el choque del carro con el extremo de la pista, colocando el péndulo en posición horizontal.

La simulación termina cuando el contador de la simulación Tsim es mayor a la constante NUMMAXpUNTOS, haciendo que se deshabilite el reloj.

La variable Tsim es la que determina cuales valores de los arreglos espacio y ángulo se deben mostrar en la simulación. Mientras la variable IntervaloSim junto con el seteo del tiempo de activación del reloj permiten variar la velocidad de la animación.

Además del movimiento del carro y péndulo se presentan los valores instantáneos de las variables (espacio, ángulo, velocidad angular, velocidad del carrito, fuerza), así como también sus gráficas en el tiempo, éstas son realizadas en la subrutina "Valores_Análogos" cuyo diagrama de flujo se presenta en la Fig.4.2.1.3.

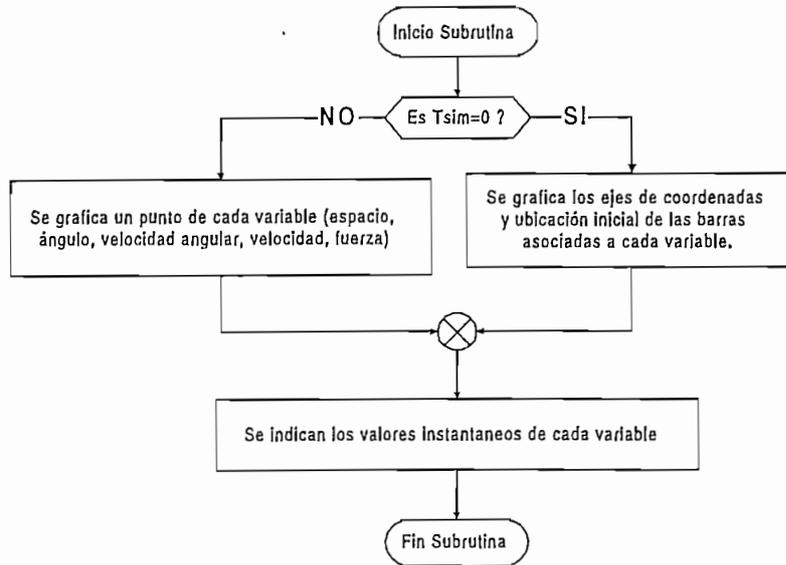


Fig.4.2.1.3.- Diagrama de Flujo de la Subrutina "Valores Análogos" para la Simulación Dinámica

En esta subrutina (Fig.4.2.1.3) se chequea primero si Tsim es igual a cero, si lo es, se grafican los ejes de coordenadas, en la ventana en la cual se presentan las curvas en el tiempo y si no lo es pasa directamente a graficar un punto de cada una de las variables (espacio, ángulo, velocidad angular, velocidad del carrito, fuerza). Luego se presentan los valores instantáneos de cada una de las variables y el tiempo transcurrido de simulación, en las ventanas correspondientes 1 y 2 (ver Fig.4.1.1).

Además a cada variable se asocia una barra cuya longitud está en función del valor que tomen las variables, tomando la barra el color rojo para valores positivos y verde para los negativos.

4.2.2.- Diagramas de Flujo de las Alternativas de Control.

Como ya se mencionó anteriormente, el programa se desarrolló para cinco tipos de control, cuyos diagramas de flujo se presentan a continuación siendo estos expresados en forma simplificada para dar una idea clara del funcionamiento del programa:

1) Compensación por Red de Adelanto de Fase

El método de la bisectriz tiene el siguiente diagrama de bloques:

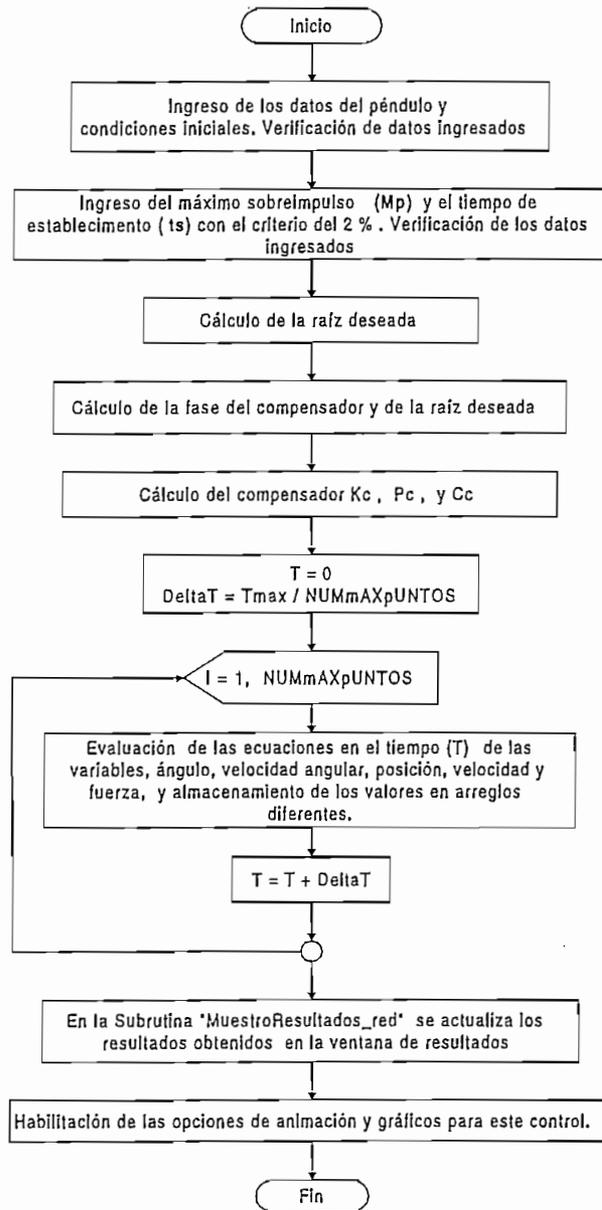


Fig.4.2.2.1. Diagrama de Flujo de la Compensación por Red de Adelanto de Fase Utilizando el Método de la Bisectriz

La variable T_{max} expresada en el diagrama de flujo anterior representa el tiempo de simulación, y ΔT es el intervalo de tiempo de evaluación de las ecuaciones.

En el diagrama de flujo de la Fig.4.2.2.1 el cálculo de la raíces deseadas se lo hace a partir de la Ecs.3.1.5 y 3.1.6 que son:

$$\sigma = \zeta Wn \quad (\text{Ec.3.1.5})$$

$$Wd = W_n \sqrt{1 - \zeta^2} \quad (\text{Ec.3.1.6})$$

El cálculo de las fases del compensador y de la raíz deseada se las calcula con las Ecs.3.1.12 y 3.1.9 respectivamente, las cuales se describen a continuación:

$$\theta_c = \text{tg}^{-1} \frac{-2\sigma Wd}{\left(\sigma^2 - Wd^2 - \frac{g}{L'} \right)} \quad (\text{Ec.3.1.12})$$

$$\theta_{PD} = 180^\circ - \arctan\left(\frac{Wd}{\sigma}\right) \quad (\text{Ec.3.1.9})$$

Las constantes del compensador se calculan a través de las Ecs.3.1.13, 3.1.14 y 3.1.17.

$$P_c = \zeta Wn + \frac{Wd}{\tan\left(\frac{\theta_{PD} - \theta_c}{2}\right)} \quad (\text{Ec.3.1.13})$$

$$C_c = \zeta Wn + \frac{Wd}{\tan\left(\frac{\theta_{PD} + \theta_c}{2}\right)} \quad (\text{Ec.3.1.14})$$

$$K_c = \frac{\sqrt{(-\sigma + P_c)^2 + Wd^2}}{\sqrt{(-\sigma + C_c)^2 + Wd^2}} \cdot ML' \left[\sqrt{(2\sigma Wd)^2 + \left(\sigma^2 - Wd^2 - \frac{g}{L'} \right)^2} \right] \quad (\text{Ec.3.1.17})$$

Para la evaluación de la ecuaciones en el tiempo de las variables ángulo, velocidad angular, espacio, velocidad y fuerza se utilizan las Ecs.3.1.28, 3.1.29, 3.1.36, 3.1.37 y 3.1.22 respectivamente las cuales son:

$$\Phi(t) = Ae^{-\gamma t} + Be^{-\sigma t} \cos Wdt + Ce^{-\sigma t} \operatorname{sen} Wdt \quad (\text{Ec.3.1.28})$$

$$\omega(t) = A_1 e^{-\gamma t} + B_1 e^{-\sigma t} \cos Wdt + C_1 e^{-\sigma t} \operatorname{sen} Wdt \quad (\text{Ec.3.1.29})$$

$$v(t) = v_{A_1} u(t) + v_{A_2} \ddot{\Phi}(t) + v_{A_3} \Phi(t) + C_1 \quad (\text{Ec.3.1.36})$$

$$x(t) = X_{A_1} \dot{x}(t) + X_{A_2} \Phi(t) + X_{A_3} u(t) + C_1 t + C_2 \quad (\text{Ec.3.1.37})$$

$$u(t) = -L' M \frac{\partial^2 \Phi(t)}{\partial t^2} + gM\Phi(t) \quad (\text{Ec.3.1.22})$$

Todos los cálculos anteriores se lo realizan dentro de la subrutina "CalculosAdelBisectriz".

El diagrama de flujo por el método del cero arbitrario está representado por la Fig.4.2.2.2.

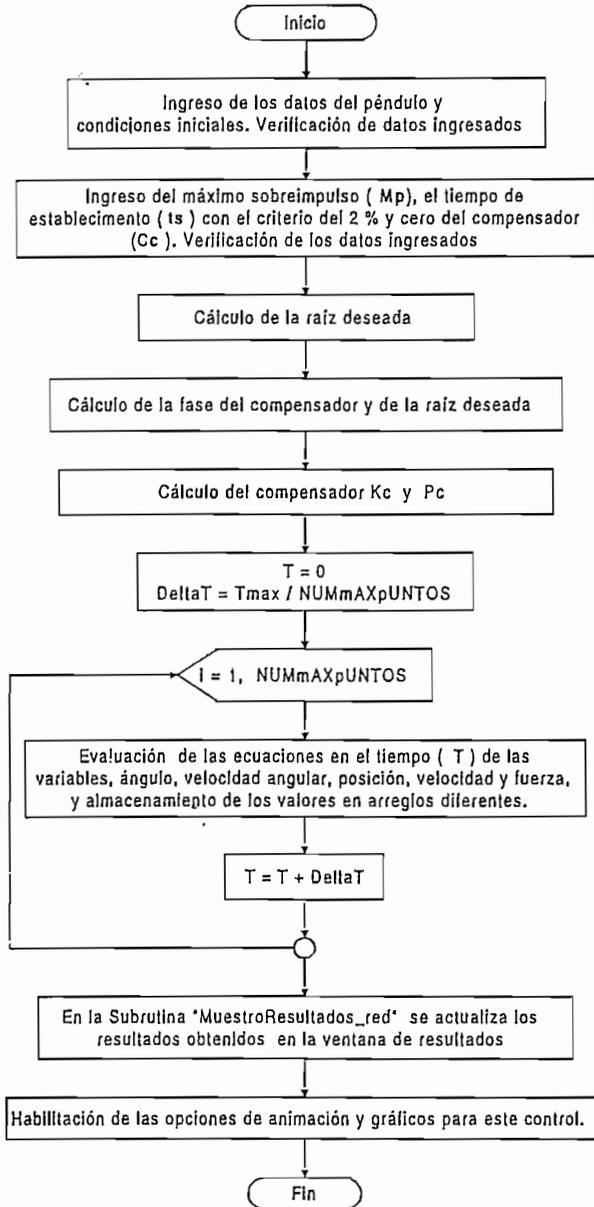


Fig.4.2.2.2. Diagrama de flujo de la compensación por red de adelanto de fase utilizando el método del cero arbitrario

A diferencia del método de la bisectriz, el cero es un dato que se ingresa y el polo dependiendo de la ubicación del cero se lo cálculo a través de la Ecs.3.1.18, 3.1.19 y 3.1.20 que son:

a).- Si $-\sigma + C_c > 0$:

$$P_c = \sigma + \frac{Wd}{\tan \left[\tan^{-1} \left(\frac{Wd}{-\sigma + C_c} \right) - \theta_c \right]} \quad (\text{Ec.3.1.18})$$

b).- Si $-\sigma + C_c = 0$:

$$P_c = \sigma + \frac{Wd}{\tan [90 - \theta_c]} \quad (\text{Ec.3.1.19})$$

c).- Si $-\sigma + C_c < 0$:

$$P_c = \sigma + \frac{Wd}{\tan \left[180 + \tan^{-1} \left(\frac{Wd}{-\sigma + C_c} \right) - \theta_c \right]} \quad (\text{Ec.3.1.20})$$

2) Acción de Control Proporcional Derivativo

Su diagrama de flujo se lo representa a continuación:

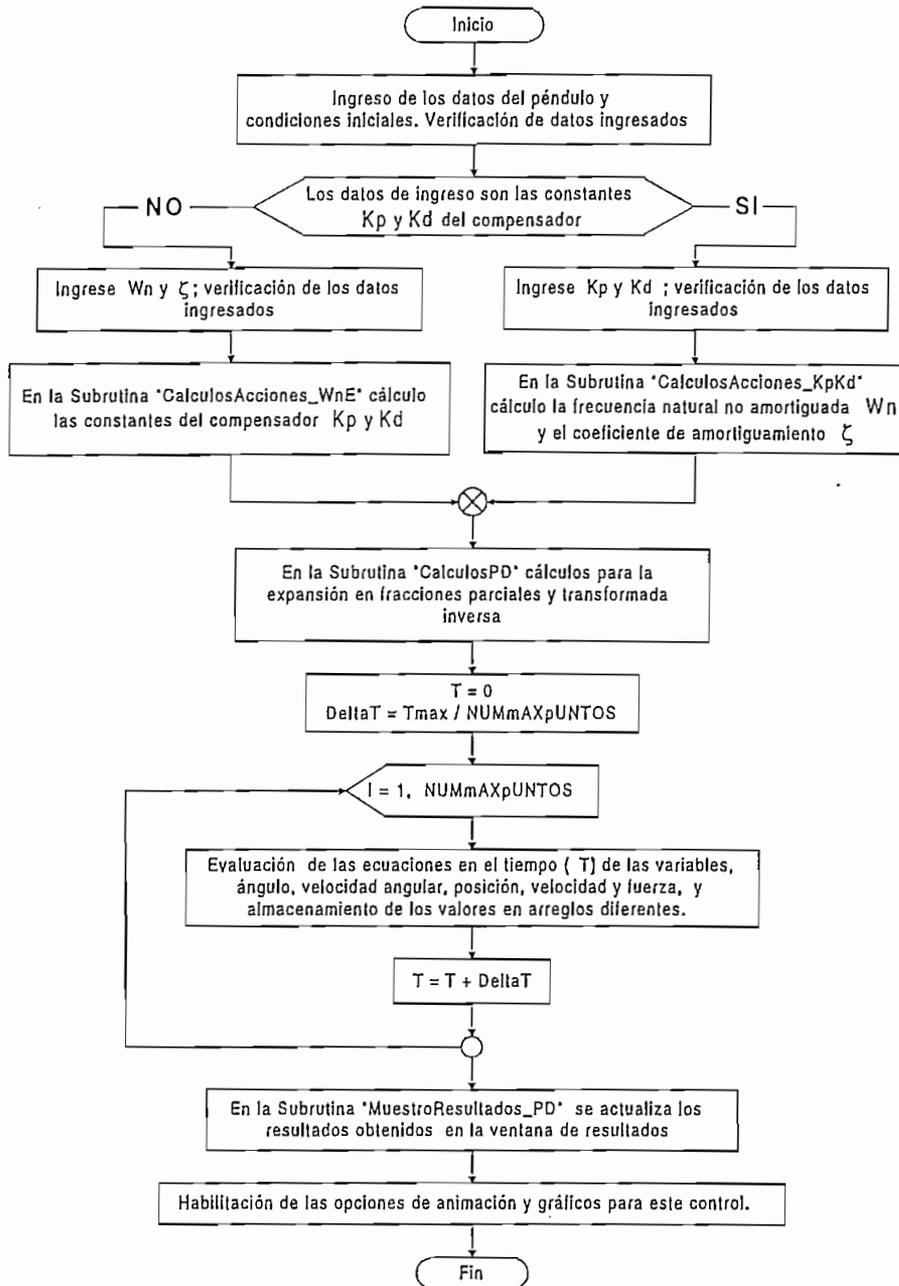


Fig.4.2.2.3. Diagrama de flujo de la Acción de Control Proporcional Derivativo

El cálculo de las constantes del compensador (K_p y K_d) conociendo la frecuencia natural no amortiguada y el coeficiente de amortiguamiento, se lo hace a través de las Ecs.3.2.9 y 3.2.10.

$$Kp = \left(W_n^2 + \frac{g}{L'} \right) L' M \quad (\text{Ec.3.2.9})$$

$$Kd = \frac{2\zeta W_n L' M}{Kp} \quad (\text{Ec.3.2.10})$$

Y el cálculo de la frecuencia natural no amortiguada y del coeficiente de amortiguamiento conociendo las constantes de compensador (Kp y Kd) se lo hace a partir de las Ecs.3.2.7 y 3.2.8:

$$\zeta = \frac{Kp Kd}{2 W_n L' M} \quad (\text{de la Ec.3.2.7})$$

$$W_n = \sqrt{\frac{Kp}{L' M} - \frac{g}{L'}} \quad (\text{de la Ec.3.2.8})$$

Para la evaluación de la ecuaciones en el tiempo de las variables ángulo, velocidad angular, espacio, velocidad y fuerza se utilizan las Ecs.3.2.17, 3.2.18, 3.2.22, 3.2.21 y 3.2.1 respectivamente las cuales son:

$$\Phi(t) = \frac{e^{-\zeta W_n t}}{\sqrt{1-\zeta^2}} \left[-\Phi(0) \text{sen}(Wd t - \Phi_x) + \frac{[\dot{\Phi}(0) + 2\zeta W_n \Phi(0)]}{W_n} \text{sen}(Wd t) \right] \quad (\text{Ec.3.2.17})$$

$$\dot{\Phi}(t) = \frac{e^{-\zeta W_n t} W_n}{\sqrt{1-\zeta^2}} \left\{ \Phi(0) \text{sen}(Wd t - 2\Phi_x) - \frac{(\dot{\Phi}(0) + 2\zeta W_n \Phi(0))}{W_n} \text{sen}(Wd t - \Phi_x) \right\} \quad (\text{Ec.3.2.18})$$

$$x(t) = \frac{1}{Mb} \left[Kp \left(\frac{a^2}{b} - a Kd - 1 \right) \Phi(t) - Kp \left(Kd - \frac{a}{b} \right) \dot{\Phi}(t) \right] + C_1 t + C_2 \quad (\text{Ec.3.2.22})$$

$$v(t) = \dot{x}(t) = \frac{1}{M} \left\{ Kp \left(Kd - \frac{a}{b} \right) \Phi(t) - \frac{Kp}{b} \dot{\Phi}(t) \right\} + C_1 \quad (\text{Ec.3.2.21})$$

$$u(t) = Kp \left[\Phi(t) + Kd \frac{d\Phi(t)}{dt} \right] \quad (\text{Ec.3.2.1})$$

3) Acción de Control Proporcional Integral Derivativo

El diagrama de flujo para este control se lo detalla a continuación:

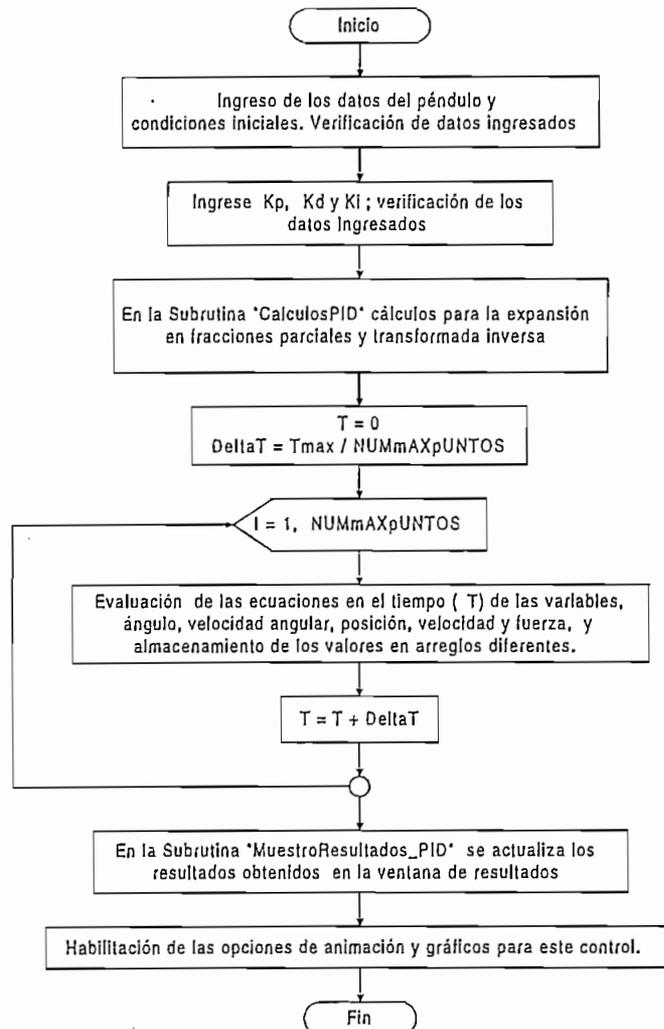


Fig.4.2.2.4. Diagrama de Flujo de la Acción de Control Proporcional Integral Derivativo

La evaluación de la ecuaciones en el tiempo de las variables ángulo, velocidad angular, espacio, velocidad y fuerza se lo hace de acuerdo a la siguientes Ecs.3.2.32, 3.2.33, 3.2.37, 3.2.36 y 3.2.2 respectivamente las cuales son:

$$\Phi(t) = Ae^{-\gamma t} + e^{-\sigma t} [B \cos Wd t + C \operatorname{sen} Wd t] \quad (\text{Ec.3.2.32})$$

$$\dot{\Phi}(t) = A_1 e^{-\gamma t} + e^{-\sigma t} [B_1 \cos Wd t + C_1 \operatorname{sen} Wd t] \quad (\text{Ec.3.2.33})$$

$$x(t) = x_{A1} \ddot{\Phi}(t) + x_{A2} \dot{\Phi}(t) + x_{A3} \Phi(t) + C_1 t + C_2 \quad (\text{Ec.3.2.37})$$

$$v(t) = v_{A1} \ddot{\Phi}(t) + v_{A2} \dot{\Phi}(t) + v_{A3} \Phi(t) + C_1 \quad (\text{Ec.3.2.36})$$

$$u(t) = -L M \ddot{\Phi}(t) + g M \Phi(t) \quad (\text{Ec.3.2.2})$$

4) Realimentación de Estado

Su diagrama de flujo es el siguiente:

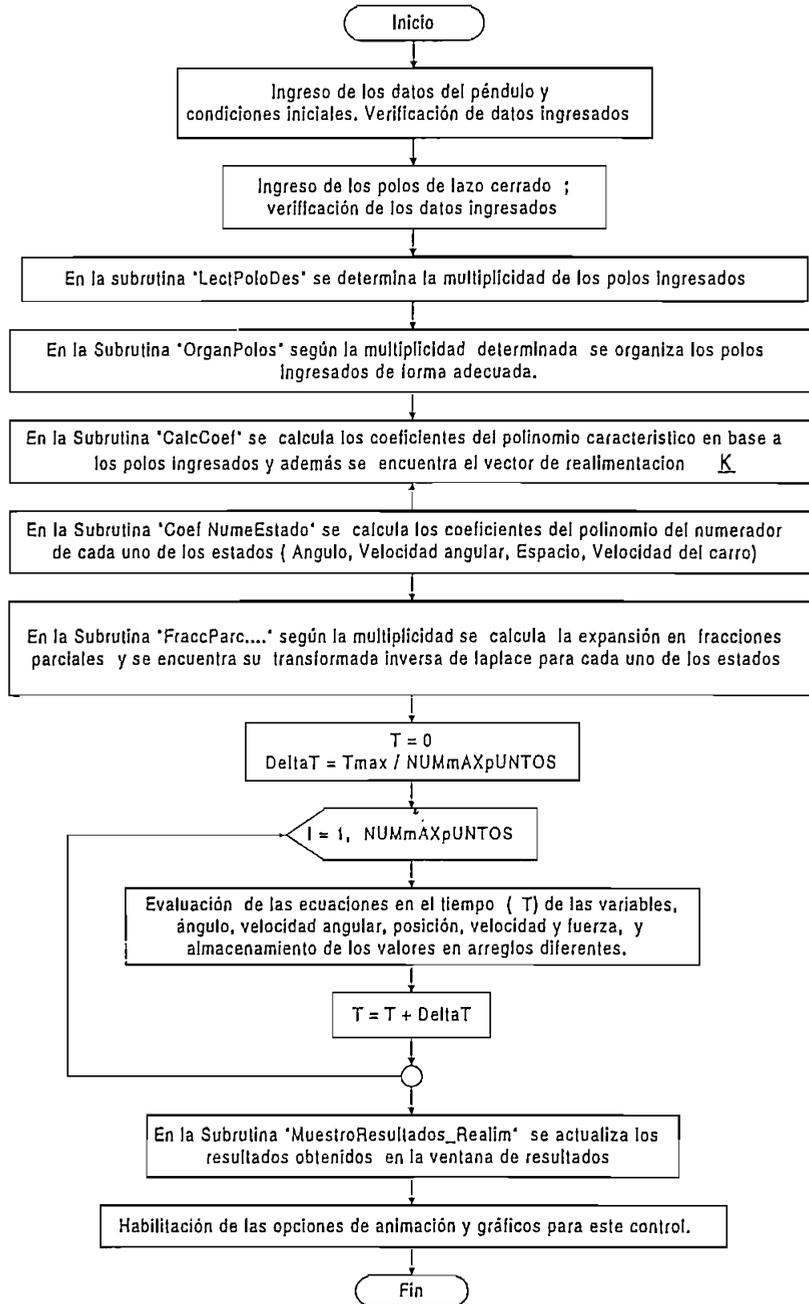


Fig.4.2.2.5. Diagrama de flujo del Control por Realimentación de Estado

Donde el vector de realimentación de estado se calcula en base a las Ecs.3.3.25 , 3.3.26, 3.3.27 y 3.3.28 las cuales son:

$$K_1 = -\frac{a_{c0}L'M}{g} \quad (\text{Ec.3.3.25})$$

$$K_2 = -\frac{a_{c1} L' M}{g} \quad (\text{Ec.3.3.26})$$

$$K_3 = -L' M \left(\frac{L'}{g} a_{c0} + \frac{g}{L'} + a_{c2} \right) \quad (\text{Ec.3.3.27})$$

$$K_4 = -L' M \left(\frac{L'}{g} a_{c1} + a_{c3} \right) \quad (\text{Ec.3.3.28})$$

El cálculo de los coeficientes del numerador para cada uno de los estados se lo realiza en base a las Ecs.3.3.4 hasta la Ec.3.3.19.

Dependiendo de los polos ingresados la evaluación de las ecuaciones en el tiempo para cada estado se calcula con las fórmulas desarrolladas en el capítulo III que son:

Todos los polos son reales y distintos:

$$x_n(t) = A_{n1} e^{p_1 t} + A_{n2} e^{p_2 t} + A_{n3} e^{p_3 t} + A_{n4} e^{p_4 t}$$

Dos polos reales repetidos y dos polos diferentes

$$x_n(t) = (A_{n1} + A_{n2} t) e^{p_1 t} + A_{n3} e^{p_2 t} + A_{n4} e^{p_3 t}$$

Tres polos reales repetidos y un polo diferente

$$x_n(t) = \left(A_{n1} + A_{n2} t + \frac{A_{n3}}{2!} t^2 \right) e^{p_1 t} + A_{n4} e^{p_2 t}$$

Todos los polos reales repetidos:

$$x_n(t) = \left(A_{n1} + A_{n2} t + \frac{A_{n3}}{2!} t^2 + \frac{A_{n4}}{3!} t^3 \right) e^{p_1 t}$$

Cuatro polos repetidos de dos en dos:

$$x_n(t) = (A_{n1} + A_{n2} t) e^{p_1 t} + (A_{n3} + A_{n4} t) e^{p_2 t}$$

Polos complejos conjugados distintos

$$x_n(t) = e^{-\sigma_1 t} (A_{n1} \cos Wd_1 t + A_{n2} \text{sen } Wd_1 t) + e^{-\sigma_2 t} (A_{n3} \cos Wd_2 t + A_{n4} \text{sen } Wd_2 t)$$

Un polo complejo conjugado de multiplicidad dos:

$$x_n(t) = e^{-\sigma t} (A_{n1} \cos Wdt + A_{n2} \text{sen } Wdt) + \frac{A_{n3} t}{2Wd} e^{-\sigma t} \text{sen } Wdt + \frac{A_{n4}}{2Wd^2} e^{-\sigma t} [\text{sen } Wdt - Wdt \cos Wdt]$$

Un polo complejo conjugado y dos reales distintos

$$x_n(t) = A_{n1} e^{p_1 t} + A_{n2} e^{p_2 t} + A_{n3} e^{-\sigma t} \cos Wdt + A_{n4} e^{-\sigma t} \text{sen } Wdt$$

Un polo real complejo conjugado y un polo real de multiplicidad 2

$$x_n(t) = (A_{n1} + A_{n2}) e^{p_1 t} + A_{n3} e^{-\sigma t} \cos Wdt + A_{n4} e^{-\sigma t} \text{sen } Wdt$$

Y la fuerza es evaluada con la siguiente ecuación:

$$u(t) = K_1 X_1(t) + K_2 X_2(t) + K_3 X_3(t) + K_4 X_4(t)$$

5) Control Fuzzy

El diagrama de flujo del Control Fuzzy simplificado se lo presenta en las Figs.4.2.2.6 y 4.2.2.7:

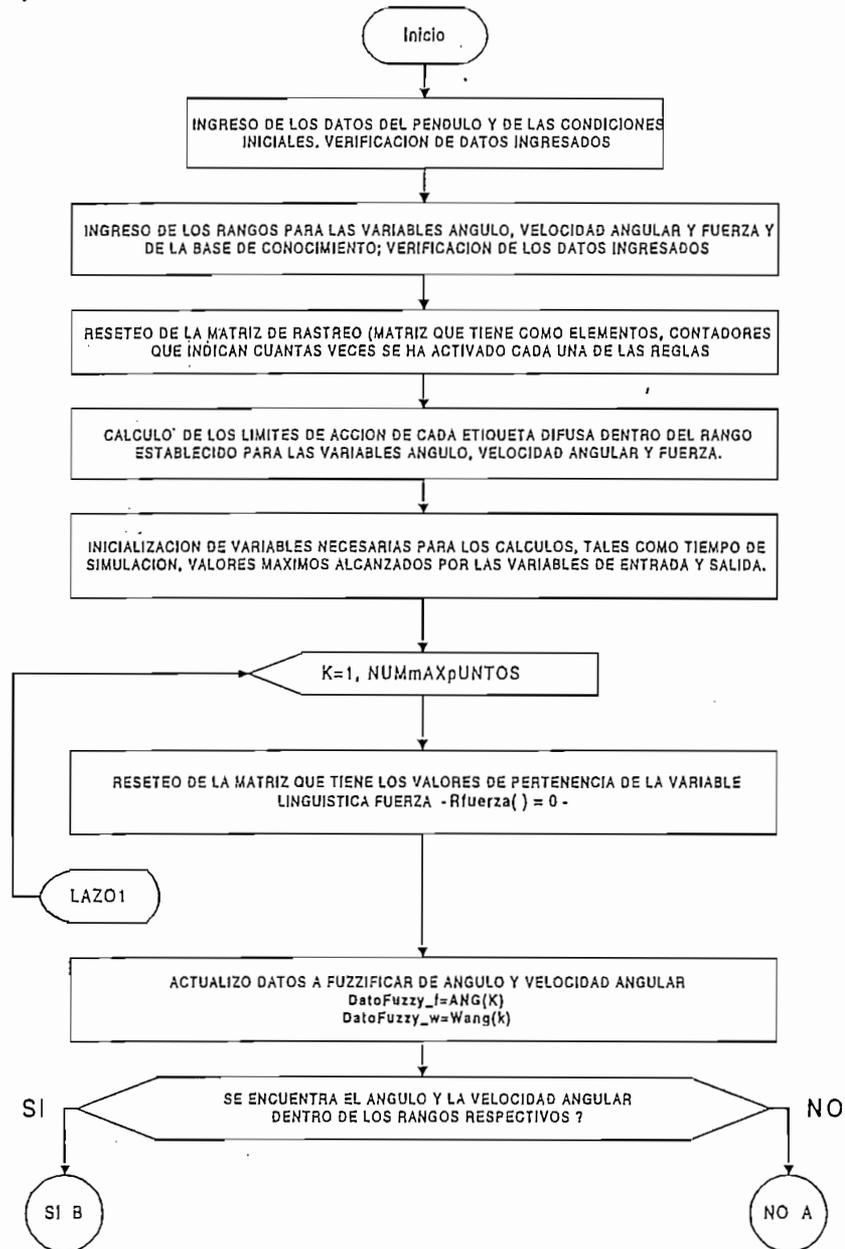


Fig.4.2.2.6. Diagrama de Flujo del Control Fuzzy (Primera parte)

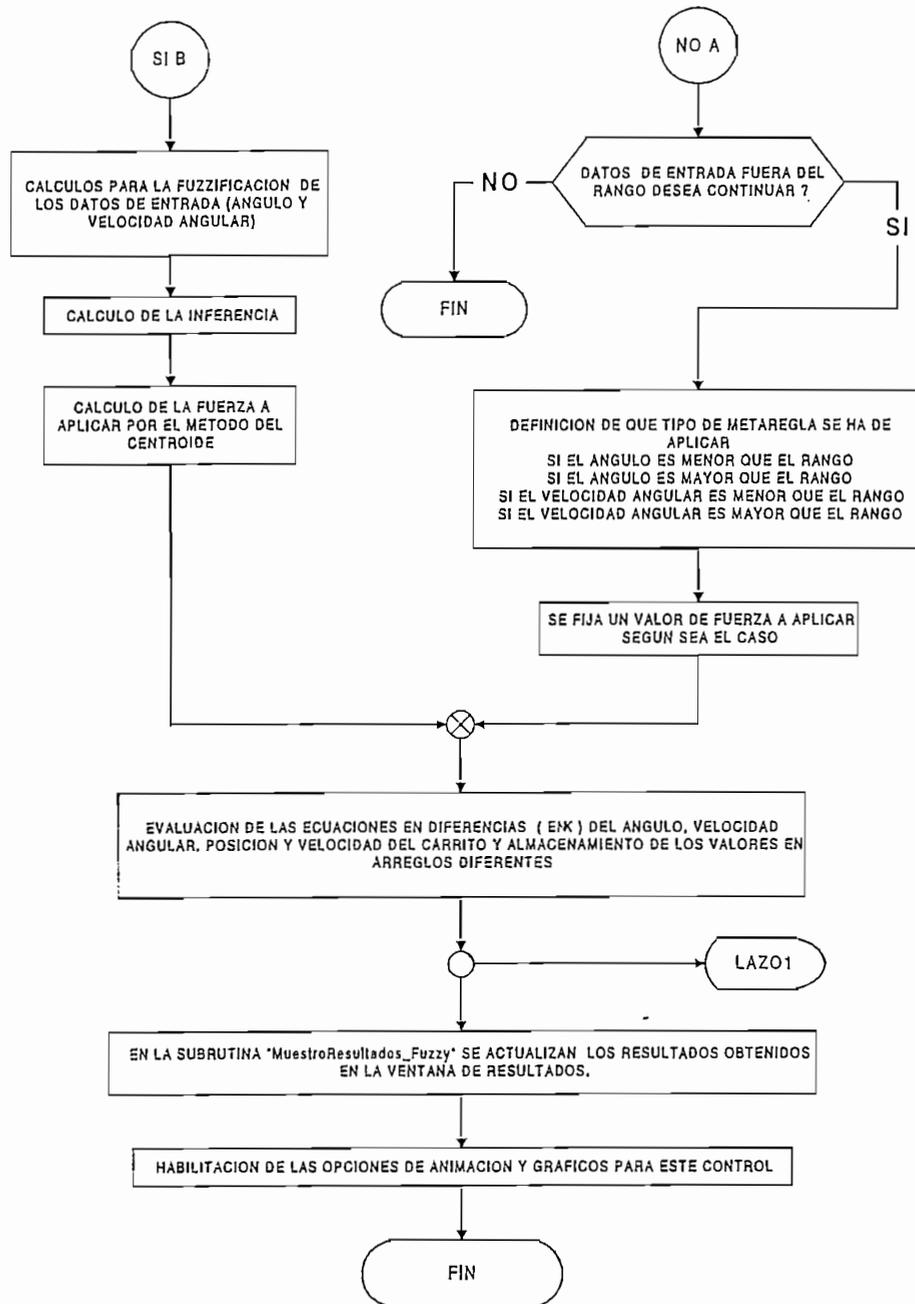


Fig.4.2.2.7. Diagrama de Flujo del Control Fuzzy (Segunda parte)

Para la definición de los límites de acción para cada etiqueta Fuzzy y para el cálculo de sus centroides el diagrama de flujo se lo presenta en la Fig.4.2.2.8, donde se definen las variables:

Solap = solapamiento de las etiquetas fuzzy.

RangoMin = valor mínimo del rango.

Base = base de cada una de las etiquetas fuzzy .

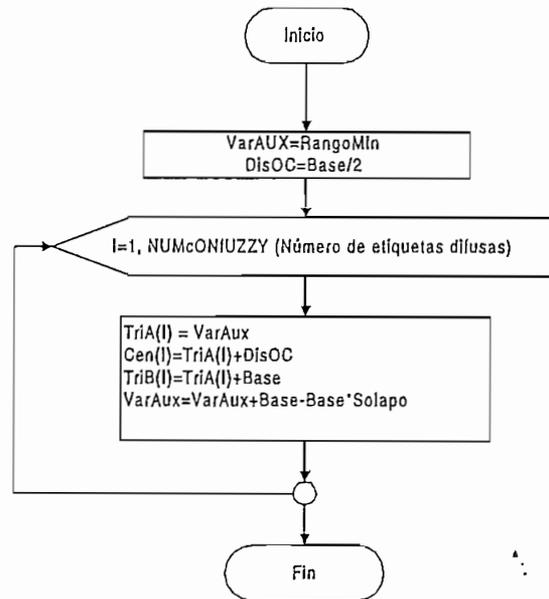


Fig.4.2.2.8. Diagrama de Flujo del Cálculo de los Límites de Acción de las Etiquetas Fuzzy

El mismo diagrama de flujo de la Fig.4.2.2.8 se lo utiliza para el cálculo de los parámetros de las etiquetas fuzzy de las variables ángulo, velocidad angular y fuerza

El diagrama de flujo de la Fig.4.2.2.9 representa la fuzzificación de los valores del ángulo. El mismo procedimiento se lo utiliza para la variable velocidad angular.

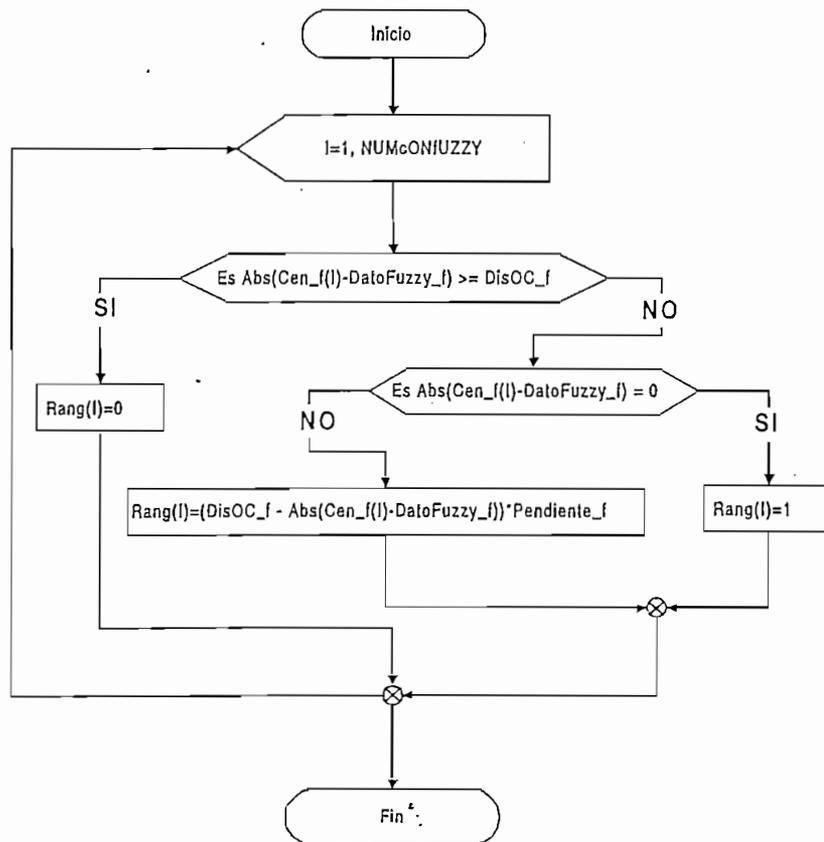


Fig.4.2.2.9. Diagrama de Flujo de la Fuzzificación del Ángulo

donde:

$Cen_f()$ = arreglo de los centroides de las etiquetas fuzzy de ángulo

$DatoFuzzy_f$ = dato del ángulo a ser fuzzificado

$Rang()$ = arreglo de los valores de pertenencia del ángulo

$Pendiente_f$ = pendiente de las etiquetas del ángulo

El diagrama de flujo de la Fig.4.2.2.10 representa el mecanismo de inferencia.

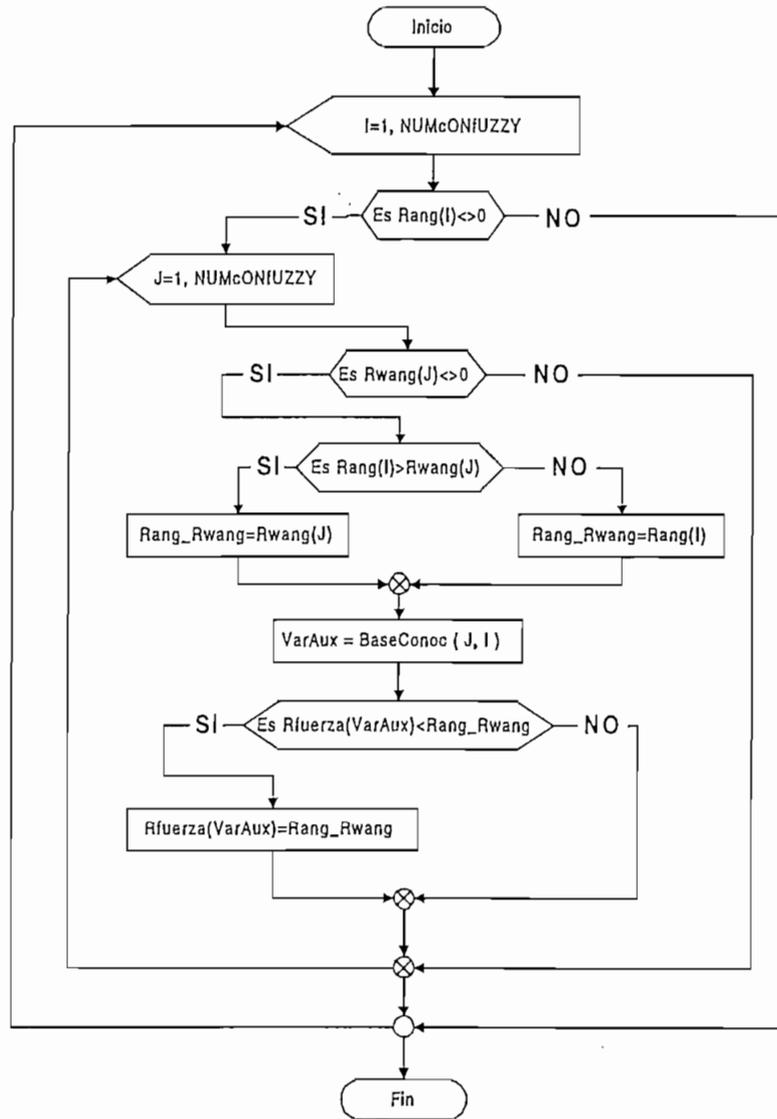


Fig.4.2.2.10. Diagrama de Flujo del Mecanismo de Inferencia

donde:

$Rwang()$ = arreglo de los valores de pertenencia de la velocidad angular

$Rfuerza()$ = arreglo de los valores de pertenencia de la fuerza

$BaseConoc()$ = arreglo que contiene la base de conocimientos.

El diagrama de la Fig.4.2.2.11 representa el cálculo de la fuerza por el método del centroide (defuzzificación):

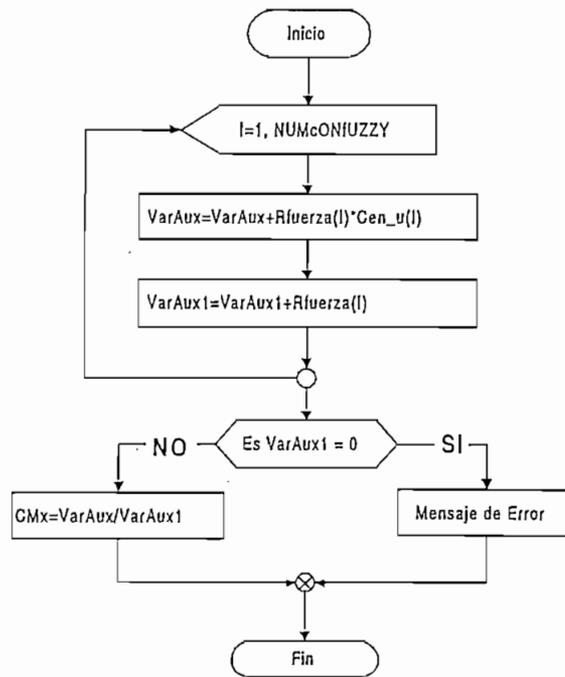


Fig.4.2.2.11. Diagrama de Flujo del Cálculo del Centroide

donde CMx representa el centroide de la superficie solución (valor de la fuerza a aplicar).

Las ecuaciones en diferencias de las variables ángulo, velocidad angular, posición y velocidad ha ser evaluadas son las Ecs.3.4.5, 3.4.6, 3.4.7 y 3.4.8 respectivamente.

$$\Phi(K) = \frac{b_1 U(K-1) + b_1 U(K-2) - c_1 \Phi(K-1) - c_2 \Phi(K-2)}{c_0} \quad (\text{Ec.3.4.5})$$

$$\omega(K) = \frac{\Phi(K) - \Phi(K-1)}{T} \quad (\text{Ec.3.4.6})$$

$$X(K) = \frac{f_1 U(K-1) + f_1 U(K-2) - g_1 X(K-1) - g_0 X(K-2)}{g_0} \quad (\text{Ec.3.4.7})$$

$$V(K) = \frac{X(K) - X(K-1)}{T} \quad (\text{Ec.3.4.8})$$

El valor de la fuerza a almacenar queda definido por el centroide

$$u(K) = CMx$$

Por último, cabe mencionar que en todo diagrama de flujo representado anteriormente se han obviado algunos pasos de chequeo y procedimientos intermedios de cálculo, para no complicar demasiado los diagramas, dando así una idea clara del funcionamiento general del programa.

CAPITULO V: RESULTADOS Y CONCLUSIONES

5.1.- Resultados de la Simulación Dinámica.

Como resultado a varios ensayos con el programa se ha llegado a las siguientes conclusiones acerca de la simulación dinámica.

* La simulación dinámica se la puede activar para cada uno de los controles, permitiendo a su vez observar cómo varían las gráficas de todas las variables de entrada (ángulo, velocidad angular, espacio y velocidad) y salida (fuerza).

* En el programa la animación se realiza en base a un reloj el cual a intervalos regulares de tiempo actualiza la información presentada en la pantalla, facilitando con esto que la rapidez con que se simule sea relativamente independiente del computador en el cual se esté corriendo el programa (Cuando el tiempo que demora la ejecución de la subrutina "TimerSim_Timer" es menor al intervalo de activación del reloj).

* El programa permite detener la simulación en cualquier momento así como también variar la velocidad de animación para mejorar la recolección de información procedente del control.

* En la simulación dinámica se pueden obtener los valores instantáneos de cada una de las variables de entrada y salida con la ayuda del botón Congelar/Descongelar.

* Al realizar la simulación, todos los objetos presentados en ésta son escalados adecuadamente, ocupando la totalidad de la pantalla de tal manera de permitir una mejor apreciación.

* En los datos de entrada se permite cambiar la longitud de la pista para que en la simulación dinámica se observe el mínimo valor de la pista requerido para que el péndulo llegue a estabilizarse.

* Cuando la compensación es demasiado lenta, el péndulo no logra estabilizarse dentro de la pista establecida, en cuyo caso en la simulación se representa el choque del carrito contra los extremos de la pista.

* En el caso en que el péndulo logra estabilizarse y el carrito queda moviéndose con velocidad constante, de igual manera que en el caso anterior se representa en la simulación un choque del carrito con los extremos de la pista, siempre y cuando en el tiempo de simulación se sobrepase uno de dichos extremos.

* Se recomienda que el programa sea corrido con un monitor a color para poder distinguir de mejor manera a las variables pues las gráficas de estas son presentadas en diferentes colores.

* En conclusión se puede decir que el uso de la simulación dinámica como una herramienta para el aprendizaje de las teorías de control, ayudan para un fortalecimiento de los conocimientos teóricos de las diferentes compensaciones implementadas en el presente trabajo de tesis.

5.2.- Comparación de los Resultados Analíticos

En conclusión a los resultados obtenidos al aplicar las diferentes alternativas de control se llegó a establecer lo siguiente:

* El sistema sin compensar es inestable, pues tiene un polo en el semiplano derecho (ver Fig.2.2.8), definiéndose así como un sistema de fase no mínima.

* En los controles de Acciones, Redes y Realimentación de Estado se puede observar que la respuesta transitoria del sistema está directamente relacionada con la localización de las raíces de la ecuación característica en el plano S .

* El inconveniente del método del Cero Arbitrario es que la ganancia debe ser bastante grande para conseguir que las raíces complejas sean las raíces dominantes y que el sistema cumpla las especificaciones de diseño, por lo que la red es activa pues, posee un alto valor de ganancia requiriéndose de un amplificador para que el sistema sea adecuadamente compensado.

* Se debe tomar en cuenta que el valor de la relación de amortiguamiento esté entre 0.4 y 0.8 para que el sistema no tenga un excesivo sobreimpulso ($\zeta < 0.4$) o que responda muy lentamente¹ ($\zeta > 0.8$). El tiempo de establecimiento (t_s) quedará entonces determinado por el valor de la frecuencia no amortiguada (ω_n), donde a valores grandes de ω_n el t_s disminuye y viceversa.

* En las Acciones de Control implementadas se comprueba que al disminuir el valor de la constante de proporcionalidad el tiempo de establecimiento aumenta y viceversa.

* Con el Control por Realimentación de Estado se puede localizar a los polos de la F.T. de lazo cerrado según las especificaciones que se requieren para la respuesta del sistema, y por los resultados obtenidos, este control es la mejor opción cuando se conoce el modelo matemático de la planta; su limitante es el gran desarrollo matemático requerido.

* El Control Fuzzy permite una aproximación intuitiva a la solución del problema mediante la formulación de las reglas que definen la base del conocimiento llevando al sistema a condiciones deseables de estabilidad.

* En el Control Fuzzy es fácil modificar sus premisas y operaciones para realizar la compensación, en caso de tener que hacer un cambio en éste; no sucede lo mismo con los otros controles implementados pues, cualquier modificación por mínima que sea implica un desarrollo matemático completo para obtener nuevas ecuaciones.

¹ OGATA Katsuhiko, "Ingeniería De Control Moderna" , 1980, pag. 252.

* En el Control Fuzzy el uso de la matriz de rastreo (ver Fig.3.4.16) ayuda para que se pueda determinar una adecuada base de conocimiento y un adecuado valor para los rangos de las variables que hagan cumplir al sistema los requerimientos de rapidez y estabilización requeridos.

* Cabe anotar también la importancia del Control Fuzzy como una herramienta para la estabilidad del sistema cuyas características de versatilidad, robustez y eficiencia son demostrados en el trabajo desarrollado.

En cuanto al programa desarrollado se ha llegado a las siguientes conclusiones.

* El programa permite obtener todas las curvas de interés relativas al problema y mediante el uso del cursor se facilita la obtención de sus valores.

* A pesar de que las curvas indicaran que el sistema retorna a la condición de reposo, es necesario chequear que el carrito no haya excedido en ningún momento los límites físicos de la pista pues, en este caso sería necesario modificar los datos de entrada del compensador .

* Manteniendo los datos generales del péndulo y al realizar el cálculo para cada control los resultados se van almacenando en sus ventanas respectivas, lo cual permite una comparación entre ellos.

En esta sección no se ha incluido un análisis de los resultados gráficos de cada compensación, pues estos fueron realizados en el numeral 3.5 del capítulo III y al final de cada compensación.

5.3.- Conclusiones.

En lo que respecta al trabajo de tesis y lenguaje de programación utilizado se concluye:

* El sistema de programación Visual Basic para Windows ofrece una herramienta de diseño visual que proporciona un uso simple y fácil con una programación “orientada a eventos”, por lo que el programa desarrollado resultó ser un programa amigable, de fácil manejo y muy funcional, que puede correrse sin problemas tanto en Windows 95 como en Windows 3.1, requiriendo una velocidad de proceso mínima (computador 386 recomendable 8Mby de RAM) para tener una velocidad de animación adecuada.

* El programa desarrollado en este lenguaje es una colección de microprogramas que están relacionados unos a otros y cada uno de ellos es ejecutado a raíz de eventos realizados por el usuario, esto hace que el programa pueda ser desarrollado en forma modular facilitándose así la elaboración y depuración de éste, consiguiendo desarrollar un programa complejo con facilidad y rapidez.

* Al ser un programa desarrollado completamente en Windows permite compartir la información con otras aplicaciones como por ejemplo se puede almacenar información procedente del programa en el portapapeles e incrustarla en otra aplicación de Windows como es Word, facilitando con esto al estudiante crear reportes en base a los resultados obtenidos.

* El programa desarrollado permite realizar las compensaciones por Red de Adelanto de Fase, acción de Control Proporcional Derivativo, acción de Control Proporcional Integral Derivativo, Realimentación de Estado y Control Fuzzy aplicadas a un modelo del Péndulo Invertido desarrollado en el capítulo II, visualizándose los resultados obtenidos en ventanas diferentes de tal manera que el estudiante pueda hacer un análisis del comportamiento físico del sistema ante las diferentes compensaciones propuestas.

* El programa implementado presenta información gráfica y analítica sobre el sistema compensado, en cuyos resultados se puede ver la estabilidad del sistema.

* Por último cabe mencionar que se ha creado un programa que constituye una útil herramienta para el estudiante, para una mayor comprensión de sistemas de control y en especial de las alternativas de control desarrolladas, así con los resultados obtenidos, los análisis y los comentarios realizados se cree haber cumplido los objetivos planteados en este trabajo de tesis; recomendado su utilización especialmente en el Laboratorio como una ayuda didáctica en el diseño de compensaciones.

BIBLIOGRAFIA

DISTEFANO J., STUBBERUD A, WILLIAMS Y., "Sistemas de Control por Retroalimentación", Ed. McGraw -Hill, México, Colección Shaum, 1972.

OGATA K., "Dinámica de Sistemas", Ed., Prentice Hall Hispanoamérica, Mexico, 1987.

OGATA K., "Ingeniería de Control Moderna", Ed., Prentice Hall, Mexico, 1980.

KUO B., "Sistemas Automáticos de Control", Ed., Continental, Mexico, 1986.

KUO B., "Digital Control System", Ed. International, Segunda Edición., Estados Unidos de América, 1992.

IRVING H., "Ingeniería Mecánica. Estática", tomo I, España, 1979.

AZANZA J., "Simulación Dinámica del Problema de la Bola Suspendida Mediante un Programa para Windows", Tesis de Grado, E.P.N., Quito, 1994.

MONTALVAN G., "Estudio del Péndulo Invertido y Simulación Dinámica Mediante un Programa para Windows", Tesis de Grado, E.P.N., Quito, 1994.

MONTOYA O., "Desarrollo del módulo FlexoBase-C y Aplicación al Control de Temperatura de una Termoselladora Mediante Lógica Difusa", Tesis de Grado, E.P.N., Quito, 1996.

NARVAEZ E., "Modelación. Control y Simulación Dinámica de dos Sistemas Típicos Inestables", Tesis de Grado, E.P.N., Quito, 1991.

IEEE, "Transactions of Fuzzy System", Vol. I, No. 1, Ed. IEEE, 1992.

GOMEDE F., GUDWIN R., ANDRADE M., "Controle de Processos for Lógica Fuzzy", Revista COMA1 92, De. Mindem, San Pablo, Brazil, 1992.

GOREN C., "Visual Basic 4 Enterprise", Ed. Que Corporation, 1996.

MICROSOFT EDUCATION AND CERTIFICATION., "Programming with Microsoft Visual Basic 4.0", Ed. Microsoft Corporation 1995.

MICROSOFT EDUCATION AND CERTIFICATION., "Fundamentals of Microsoft Visual Basic 4.0", Ed. Microsoft Corporation 1995.

NELSON R., "Guía Completa de Visual Basic para Windows", Ed. McGraw-Hill.

MICROSOFT., "Programmer's Guide", Programming System for Windows, Versión 3.0, Ed. Microsoft Corporation, Estados Unidos de América, 1993.

MICROSOFT., "Language Reference", Programming System for Windows, Versión 3.0, Ed. Microsoft Corporation, Estados Unidos de América, 1993.

MICROSOFT., "Turbo C++ for Windows Ver.3.0.", Programmer's Guide, Ed. Borland Interatinal, Estados Unidos de América, 1991.

ANEXOS

A.- Nomenclatura

Para el presente trabajo se utilizó la siguiente nomenclatura:

$u(t)$:	fuerza que actúa sobre el carro para mantener la varilla vertical.
$x(t)$:	desplazamiento tomado desde el centro de la pista hasta el centro del carro.
$\Phi(t)$:	ángulo formado entre el péndulo y la vertical.
$\omega(t)$:	velocidad angular del péndulo.
α :	aceleración angular del péndulo.
a_c :	aceleración del carro.
$v(t)$:	velocidad del carro.
g :	gravedad.
$H(t)$:	reacción horizontal debida al punto de contacto varilla-carro.
$V(t)$:	reacción vertical debida al punto de contacto varilla-carro.
L' :	longitud efectiva del péndulo.
L :	longitud desde el pivote al centro de gravedad de la varilla.
J :	momento de inercia del péndulo.
m :	masa del péndulo.
M :	masa del carro.
N :	normal del carro debido al contacto entre carro-pista.

a_x :	componente de la aceleración del péndulo en el eje "x".
a_y :	componente de la aceleración del péndulo en el eje "y".
K_p :	constante de proporcional (acciones de control).
K_d :	constante derivativa (acciones de control).
K_I :	constante de integración (acciones de control).
K_c :	constante de ganancia del compensador (red de adelanto de fase).
C_c :	cero del compensador (red de adelanto de fase).
P_c :	polo del compensador (red de adelanto de fase).
$P_{D1,2}$:	raíces deseadas del sistema compensado.
t_s :	tiempo de establecimiento criterio del 2%.
M_p :	máximo sobreimpulso.
L_n :	logaritmo natural.
W_n :	frecuencia natural no amortiguada.
ζ :	coeficiente de amortiguamiento.
θ_{planta} :	fase del sistema sin compensar.
θ_C :	fase que proporciona el compensador.
θ_{PD} :	fase de las raíces deseadas.
W_d :	Parte imaginaria de la raíz compleja conjugada de lazo cerrado.
σ :	Parte real de la raíz compleja conjugada de lazo cerrado.
t_r :	tiempo de crecimiento.
t_p :	tiempo en alcanzar el primer pico.
t_{cvc} :	tiempo del primer cambio de dirección de la velocidad del carro.
t_{cf} :	tiempo del primer cambio de dirección de la fuerza.
t_{cva} :	tiempo del primer cambio de dirección de la velocidad angular.
V_p :	valor pico de la primera oscilación.

- f_M : fuerza máxima aplicada.
- X_{ts} : espacio recorrido por el carro a t_s del ángulo.
- $W_{\phi=0}$: velocidad angular en el primer punto donde el ángulo es cero.
- $\Phi_{W=0}$: ángulo en el primer punto donde la velocidad angular es cero.
- V_f : velocidad final con la que el carro queda moviéndose después de estabilizar al péndulo.
- G_c : función de transferencia del compensador.
- M_c : matriz controlabilidad.
- \underline{K} : vector de realimentación.
- $X(t)$: vector de estado (control de realimentación de estado).
- p_1, p_2, p_3, p_4 : polos de lazo cerrado (realimentación de estado).
- Ang : variable ángulo (control Fuzzy).
- $Wang$: variable velocidad angular (control Fuzzy).
- MN : muy negativo (etiqueta fuzzy).
- PN : poco negativo (etiqueta fuzzy).
- C : cero (etiqueta fuzzy).
- PP : poco positivo (etiqueta fuzzy).
- MP : muy positivo (etiqueta fuzzy).
- $R_c(x)$: valor de pertenencia del número x en la etiqueta lingüística C (cero).
- CNF : cantidad de etiquetas Fuzzy.
- $SOLAP$: grado de solapamiento entre las etiquetas Fuzzy.
- $Rang$: matriz de los valores de pertenencia de la variable ángulo.
- $Rwang$: matriz de los valores de pertenencia de la variable velocidad angular.
- $Rfuerza$: matriz de los valores de pertenencia de la variable fuerza

- CDA:** centro de área (centroide) de la superficie resultante (control Fuzzy).
- ri:** elementos de la matriz Rfuerza (control Fuzzy).
- CG₁:** centroides de cada uno de las etiquetas Fuzzy de la variable Fuerza.

B.- Manual de Usuario

A continuación se presenta una guía de instalación y manejo del programa desarrollado, para una mejor utilización de este.

Instalación del Programa en el Computador

El programa "Pe_Fuzzy" tiene un archivo llamado **Setup.exe** (como la mayoría de las aplicaciones para Windows), con el cual se instala el programa en el disco duro del computador, de la siguiente manera:

- 1) Antes de instalar el software en el computador primero se debe correr windows, luego se inserta el disquete del programa Pe_Fuzzy en la unidad de disquete del computador.

- 2) Se debe correr el programa instalador desde el administrador de archivos de windows (a:\setup.exe).

- 3) Se siguen todas las instrucciones que se indiquen en el programa instalador contestando a sus preguntas sobre las opciones de instalación. El programa instalador copiará todo el software requerido al disco duro y por último añadirá una ventana de grupo y un icono llamado Pe_Fuzzy en la ventana del administrador de programas.

4) Ejecución: La forma más fácil de arrancar el programa es hacer doble “click” sobre el icono **Pe_Fuzzy**. Otra posibilidad es hacer un doble “click” sobre el archivo ejecutable **Pe_Fuzzy.exe** (programa para la simulación dinámica del péndulo invertido), en el administrador de archivos de windows.

Guía de Utilización del Programa Pe_Fuzzy.

Después que ha sido ejecutado el programa, éste muestra la ventana de presentación (Acerca de Pe_Fuzzy...) en la pantalla, la cual está presente por pocos instantes para luego ingresar al menú principal del programa quedando así listo para ser utilizado.

Dentro del menú principal se tiene:

- Archivo
- Edición
- Selección del Control
- Simulación
- Gráficos
- Herramientas
- Ayuda

Cada uno de estos menús pueden ser activados ya sea con el ratón (haciendo “click” izquierdo) o con el teclado (ALT+letra subrayada).

Dentro del menú Archivo se encuentran las siguientes opciones:

- **Nuevo Péndulo:** en esta opción se presenta una ventana donde pueden ser ingresados los datos generales y condiciones iniciales del péndulo (ver Fig.4.2.2).
- **Péndulo por Defecto:** se presenta la misma pantalla de la opción anterior con valores preestablecidos sin tener la posibilidad de cambiarlos, adicionalmente con esta opción se inicializa al sistema cargando todos los datos preestablecidos.
- **Imprimir:** con esta opción se pueden imprimir tanto los gráficos de las curvas como la simulación dinámica.
- **Salir:** permite abandonar el programa.

En el menú **Edición** sólo se presenta la opción **Copiar** la cual carga en el portapapeles de windows el contenido de las gráficas.

El menú **Selección del Control** al ser activado, presenta una ventana en la cual se escoge el tipo de control y se ingresan los datos dependiendo del control seleccionado.

- Para el control por **Red de Adelanto de Fase** se presentan dos opciones: el **Método de la Bisectriz** donde se ingresan el **Máximo sobreimpulso** y tiempo de establecimiento y en el **Método del Cero Arbitrario** donde se ingresa **Máximo sobreimpulso**, tiempo de establecimiento y el **cero del compensador**.
- Para el caso **Acciones de Control** se presentan tres opciones: **Acción PD** donde se ingresan la frecuencia natural no amortiguada (ω_n) y el coeficiente de amortiguamiento (ζ), **Acción PD conociendo K_p , K_d**

donde se ingresan K_p y K_d y **Acción PID** donde se ingresan las constantes del compensador K_p , K_d , K_i .

- Para el control de **Realimentación de Estado** se ingresan los polos de lazo cerrado $Polo_1$, $Polo_2$, $Polo_3$ y $Polo_4$.
- Por último en el **Control Fuzzy** se tienen dos ventanas de ingresos de datos, en la primera ingresan los rangos de las variables de entrada (ángulo, velocidad angular), la variable de salida (fuerza), el tiempo de simulación y en la segunda ventana se ingresa la base de conocimiento.

En el menú **Simulación** se tiene tres opciones:

- **Hacer Cálculos**: esta opción realiza los cálculos del control seleccionado, almacena en cinco arreglos diferentes los valores de las variables ángulo, velocidad angular, espacio, velocidad, fuerza y carga los resultados más importantes en la ventana de resultados.
- **Correr Animación**: realiza la simulación dinámica del sistema.
- **Resultados**: se despliega una ventana la cual contiene información de los datos generales del péndulo y de todos los resultados provenientes de los cálculos de cada control.

En el menú **Gráficos** se presentan 7 opciones las cuales son:

- **Espacio Vs. Tiempo**: gráfica el espacio recorrido por el carrito Vs. el tiempo

- Velocidad Vs. Tiempo: gráfica la velocidad del carrito Vs. el tiempo
- Angulo Vs. Tiempo :gráfica el ángulo del péndulo Vs. el tiempo
- Velocidad Ang. Vs. Tiempo gráfica la velocidad angular del péndulo Vs. el tiempo
- Fuerza Vs. Tiempo gráfica la fuerza aplicada al carrito Vs. el tiempo
- Varios: permite seleccionar uno o más gráficos para ser graficados al mismo.
- Velocidad Ang. Vs. Angulo gráfica la velocidad angular Vs. el ángulo del péndulo.

En el menú Herramienta se tienen dos opciones: el Reloj y Calculadora la cuales ejecutan los aplicaciones de Windows correspondientes.

En el menú Ayuda se tienen dos opciones:

- Contenido la cual abre el archivo de ayuda del programa, muestra la ventana de contenido.
- Buscar Ayuda Sobre... la cual abre el archivo de ayuda del programa, muestra la caja de dialogo de búsqueda.
- Uso de Ayuda abre el archivo de utilización de la ayuda de Windows.
- Acerca de Pe_Fuzzy... que muestra la ventana de presentación del programa.

C.- Listado de Programa

'Modulo Inicial

'Definición de un número complejo

Type complejo

 r As Single

 I As Single

End Type

Global msg As String 'Mensajes

Global Estilo As Integer 'Estilo del Mensaje

Global Titulo As String 'Estilo del Título

Public Const PI = 3.141592654

Public Const RADaGRADOS = 180 / PI 'constante para conversión de radianes a grados

Public Const GRADOSaRAD = PI / 180 'constante para conversión de grados a radianes

Global PuntoDecimal As String 'Para Definición del punto Decimal

'Polos complejos ingresados para el control por

'realimentación de estado consta de dos reales de

'simple precisión (R y I)

Public PoloD(1 To 4) As complejo

'Para la selección del tipo de control

Public TipoControl As Integer

Public Const SINcONTROL = 0

Public Const CONTROLrESTADOS = 1

Public Const CONTROLrEDES = 2

Public Const CONTROLaCCIONES = 3

Public Const CONTROLfUZZY = 4

'Para la definición del tipo de red de adelanto de fase

'a aplicar

Public TipoRedAdelanto As Integer

Public Const REDaDELbISECTRIZ = 1

Public Const REDaDELcERO = 2

'Valores iniciales para los datos generales del péndulo y CI

Public Const LONGpEND = 0.5 'Mitad de la longitud del péndulo

Public Const GRAVeDAD = 9.8 'Gravedad

Public Const MASAcARRO = 1 'Masa del carro

Public Const MASApENDULO = 0.1 'Masa del péndulo

Public Const iNERCiaJ = 0.00833 'inercia del péndulo

Public Const LONGITUDpISTA = 1 'Longitud de la pista 1 metro

Public Const ESPiNIC = 0 'Valor inicial del espacio

Public Const VELiNIC = 0 'Valor inicial de la velocidad

Public Const ANGiNIC = 10 'Valor inicial de el ángulo

Public Const VELaNGiNIC = 0 'Valor inicial de la velocidad angular

'Valores iniciales para los polos de lazo cerrado en el control

'por Realimentación de estado

Public Const POLOrEAL1 = -5.5

```

Public Const POLOiMAGINARIO1 = 2.5
Public Const POLOrEAL2 = -5.5
Public Const POLOiMAGINARIO2 = 2.5
Public Const POLOrEAL3 = -0.5
Public Const POLOiMAGINARIO3 = 0
Public Const POLOrEAL4 = -1
Public Const POLOiMAGINARIO4 = 0
'Definición constantes para el control por red de adelanto de fase

```

```

Public Const RED_Mp = 25
Public Const RED_Ts = 1
Public Const Red_Cc = -11

```

'Definición de constantes para el control PID

```

Public Const PID_Kp_CONST = 100
Public Const PID_Kd_CONST = 0.15
Public Const PID_Ki_CONST = 2

```

'Definición de constantes para el control PD

```

Public Const PD_Kp_CONST = 52
Public Const PD_Kd_CONST = 0.08

```

```

Public Const PD_Wn_CONST = 8
Public Const PD_Si_CONST = 0.4

```

'Definición de constantes para el control fuzzy

```

Public Const rANGOaNGULO = 30
Public Const rANGOvELaNGULAR = 150
Public Const rANGOfUERZA = 60
Public Const tiEMPOfUZZY = 0.002 'Constante de Tiempo de muestreo para
    'inicialización de variables

```

```

Public Const tiEMPOsIMULACION = 2 'Tiempo de simulación

```

'Para la definición del tipo de acción de control

'a aplicar

```

Public TipoAcciones As Integer
Public Const ACCIONPD_WnE = 1
Public Const ACCIONPD = 2
Public Const ACCIONPID = 3

```

```

Public Const NUMmAXpUNTOS = 1000 'Numero de Puntos máximos por Gráfico

```

'Subrutina que sirve para mostrar mensajes

```

Public Sub Mostrar()

```

```

    Estilo = vbOKOnly + vbInformation ' Define buttons.

```

```

    Titulo = "MENSAJES" ' Define Titulo.

```

```

    MsgBox msg, Estilo, Titulo ' Display message.

```

```

End Sub

```

```

Public Sub ChequeoDatos(txtDatoCheck As Control)

```

```

'-----
On Error GoTo ErrorDato ' habilitación de manejo de error

```

```

Err.Clear
'-----
Dim ValorCsng As Single
Dim Datocheck As String
Dim NumLetras As Integer
Dim I As Integer ' Contador

Datocheck = txtDatoCheck.Text
NumLetras = Len(Datocheck)

Valor = CSng(Datocheck)

If Valor = 0 Then
    txtDatoCheck.Text = 0
End If

For I = 1 To NumLetras
    If (Mid(Datocheck, I, 1) = ".") Or (Mid(Datocheck, I, 1) = ",") Then
        If PuntoDecimal <> Mid(Datocheck, I, 1) Then
            txtDatoCheck.Text = Val(txtDatoCheck.Text)
            I = NumLetras
        Else
            I = NumLetras
        End If
    End If
End If
Next I
'-----
ErrorDato:      'Manejo de Errores

If Err.Number <> 0 Then 'Chequeo de si existe error
    If Val(Datocheck) <> 0 Then
        txtDatoCheck.Text = Val(Datocheck)
    Else
        txtDatoCheck.Text = 0
    End If

End If
'-----
End Sub

```

'Alternativa

'Declaración de Variables y Constantes Generales
'valores por defecto del péndulo y sus variables

```

Dim Lp As Single 'Longitud efectiva del péndulo L'
Dim LongP As Single 'longitud del péndulo
Dim Gr As Single 'Gravedad g
Dim mP As Single 'masa del péndulo m
Dim Mc As Single 'masa del carro M
Dim Jp As Single 'momento de inercia del péndulo J

```

'Valores iniciales y sus variables

```
Dim Xest(1 To 4) As Single 'Xest(1) Posición inicial del carro
                          'Xest(2) Velocidad inicial del carro
                          'Xest(3) Angulo inicial del péndulo
                          'Xest(4) Velocidad angular inicial del péndulo
```

'Para la selección del caso de expresión en fracciones
'parciales para el control por realimentación de estado

```
Dim CasoRealim As Integer
Const CUATROprDIF = 1
Const DOSprIGDOSprDIF = 2
Const TRESprIGUNprDIF = 3
Const CUATROprIG = 4
Const DOSprIGDOSprIG = 5
Const DOSpcDIF = 6
Const DOSpcIG = 7
Const UNpcDOSprDIF = 8
Const UNpcDOSprIG = 9
```

'Coeficientes del polinomio característico obtenidos
'de los polos de lazo cerrado ingresados
' $q(s) = s^3 * AcDesed(3) + s^2 * AcDesed(2) + s * AcDesed(1) + AcDesed(0)$
Dim AcDesed(0 To 3) As Single

'Vector de realimentación de estado
Dim Kest(1 To 4) As Single
'Valores de los coeficientes del numerador de los vectores de
'estados
Dim b(1 To 4, 0 To 3) As Single
'Constantes de la división en fracciones parciales de los
'vectores de estado
Dim a(1 To 4, 1 To 4) As Single

'Constantes y Variables requeridas para realizar los gráficos
Const NumBarra = 10 'Numero utilizado para la barra de proceso
'de tal manera que cada 10 veces (NumBarra)
'cambie el indicador de proceso
Const DIV = 10 'Numero de divisiones para el gráfico
Const ESPESOReJES = 1.4 'Espesor de la línea de ejes
Const ESPESORgRAFICOS = 1.2 'Espesor de la línea de las curvas de
'los diferentes estados
Dim DeltaT As Single 'Paso para el gráfico
Dim Tmax As Single 'Tiempo máximo que sera evaluado para el
'gráfico
Dim Ies As Integer 'Contador para los arreglos
Dim FdeX1max As Single 'Espacio máximo alcanzado utilizado
'para el gráfico
Dim FdeX2max As Single 'Velocidad máximo alcanzado utilizado
'para el gráfico

Dim FdeX3max As Single 'Angulo máximo alcanzado utilizado
 'para el gráfico
 Dim FdeX4max As Single 'Velocidad angular máximo alcanzado utilizado
 'para el gráfico
 Dim FdeU5max As Single 'Velocidad angular máximo alcanzado utilizado
 'para el gráfico
 'Matriz donde se almacenaran los valores encontrados de la
 'Respuesta en el tiempo de cada estado y la entrada
 Dim Esp(-1 To 1001) As Single 'Matriz para el espacio
 Dim Vel(-1 To 1001) As Single 'Matriz para la velocidad
 Dim Ang(-1 To 1001) As Single 'Matriz para el ángulo
 Dim Wang(-1 To 1001) As Single 'Matriz para la velocidad angular
 Dim Fuerza(-1 To 1001) As Single 'Matriz para la Fuerza

 'Matriz de los valores de pertenencia asociados a la variable fuerza para
 'el control fuzzy
 'Dim R_u(1 To 1000) As Single 'Matriz los grados de pertenencia
 'para la Fuerza

 'Definición de colores para los gráficos
 Const ROJO = &HFF&
 Const VERDE = &HFF00&
 Const AZUL = &HFF0000
 Const LILA = &HFF00FF
 Const CELESTE = &HFFFFFF00
 Const NEGRO = &H0&

 'Definición de variables y constantes para el control proporcional
 'derivativo

 Dim Wn As Single 'Definición de la frecuencia natural no amortiguada
 Dim Si As Single 'Coeficiente de amortiguamiento

 'Definición de variables y constantes para el control por red de
 'adelanto de fase por el método de ubicación del cero arbitrario
 'y el método de la bisectriz

 Dim RedcMp As Single 'Almacena el valor el sobretiro para control
 'Del cero arbitrario
 Dim RedcTs As Single 'Almacena el valor del tiempo de establecimiento
 'para Control del cero arbitrario
 Dim RedcCc As Single 'Almacena la ubicación del cero de la
 'red de adelanto de fase

 Dim RedbMp As Single 'Almacena el valor el sobretiro método bisectriz
 Dim RedbTs As Single 'Almacena el valor del tiempo de establecimiento
 'método de la bisectriz

 'Definición de variables y constantes para el control PID

 Dim A_Kp As Single
 Dim A_Kd As Single
 Dim A_Ki As Single

'Definición de variables y constantes para el control PD

```
Dim Ad_Kp As Single
Dim Ad_Kd As Single
```

'Definición de constantes y variables requeridas para la animación

```
Const PASOsIMULACION = 10 'Paso para la simulación
Dim IntervaloSim As Single 'Almacena el intervalo de simulación
                          'Por omisión tiene un valor de 10
Dim CambioVelocidad As Integer 'Bandera que chequea si se ha producido un
                              'Cambio en la velocidad de animación
                              '1 indica cambio y 0 sin cambio
```

```
Const ESPESORpISTA = 3 'Espesor de la Pista para el carrito
```

```
Dim Tsim As Integer 'Paso de tiempo para la simulación
Dim LongPista As Single
```

```
Dim CuadroX As Single 'Para definir la escala "X" del area donde
                      'va a estar el carrito para la animación
Dim CuadroY As Single 'Para definir la escala "y" del area donde
                      'va a estar el carrito para la animación
                      'picGraficos.Scale(-x,y)-(x,-y)
```

'Definición de variables y constantes para el control FUZZY

```
Dim Tfuzzy As Single 'Variable de tiempo de muestreo para
                    'las ecuaciones
                    'de simulación para inicialización
```

```
Dim Tsimulacion As Single 'Variable que contiene el tiempo de simulación
```

```
Const NUMCONFUZZY = 5 'Numero de conjuntos fuzzy
```

```
Dim BaseConoc(1 To NUMCONFUZZY, 1 To NUMCONFUZZY) As Integer
                    'Valores para la definición de la base del
                    'conocimiento
```

```
Dim Rango_u As Single 'Valor del rango de la fuerza
Dim Rango_f As Single 'Valor del rango de el ángulo
Dim Rango_w As Single 'Valor del rango de la velocidad angular
```

'Realimentación de Estado

'Función que permite el calculo de la suma de dos
'numeros complejos

```
Private Function SumaComp(OPer1 As complejo, OPer2 As complejo) As complejo
SumaComp.r = OPer1.r + OPer2.r
SumaComp.I = OPer1.I + OPer2.I
End Function
```

'Realimentación de Estado

'Función que permite el cálculo de la multiplicación de dos

```
'numeros complejos
Private Function Multcomp(OPer1 As complejo, OPer2 As _
    complejo) As complejo
Multcomp.r = OPer1.r * OPer2.r - OPer1.I * OPer2.I
Multcomp.I = OPer1.r * OPer2.I + OPer1.I * OPer2.r
End Function
```

```
Private Sub cmdCongelar_Click()
If (cmdCongelar.Caption = "CONGELAR") And (TimerSim.Enabled = True) Then
    cmdCongelar.Caption = "DESCONGELAR"
    TimerSim.Enabled = False
ElseIf (cmdCongelar.Caption = "DESCONGELAR") And (TimerSim.Enabled = False) Then
    cmdCongelar.Caption = "CONGELAR"
    TimerSim.Enabled = True
End If
End Sub
```

```
Private Sub Form_Load()
ComandosInvisibles
InicializacionVariables
Load frmResultados
Load frmDatosPendulo
Load frmTiposDeControl
mnuNuevoPendulo.Tag = 0
mnuPenduloPorDefecto.Tag = 0
mnuSelecionControl.Tag = 0
mnuCalculos.Tag = 0
frmDatosPendulo.cmdModificar.Tag = 0
End Sub
```

```
Private Sub Form_Resize()
UbicacionControles
frmPresenacion.Hide
End Sub
```

```
Private Sub HScrollVelocidad_Change()
CambioVelocidad = 1
End Sub
```

```
Private Sub mnuAcerca_Click()
'-----
On Error Resume Next 'habilitación de manejo de error
Err.Clear
'-----
frmPresenacion.cmdPresentOK.Visible = True
```

```
frmPresentacion.Show 1
End Sub
```

```
Private Sub mnuAnguloVsTiempo_Click()
'-----
On Error Resume Next 'habilitación de manejo de error
Err.Clear
'-----
If mnuCalculos.Tag = 1 Then
    Dim Tiempo As Single
    'Dim Ies As Integer
    lblTitulo.Caption = "Angulo Vs Tiempo"
    UbicacionControles
    lblEscalaEjeX.Caption = Format(Tmax / 10, "0.000") & " seg/div"
    lblEscalaEjeY.Caption = Format(FdeX3max / 10 * RADaGRADOS, _
        "0.000") & " Grad/div"
    picGraficos.Scale (-Tmax / (3 * DIV), FdeX3max + FdeX3max / DIV) - _
        (Tmax + Tmax / (3 * DIV), -FdeX3max - FdeX3max / DIV)
    picGraficos.Cls
    picGraficos.Picture = LoadPicture()
    picGraficos.DrawWidth = ESPESOReJES

    'Dibujando ejes
    DibujoEjes Tmax, FdeX3max, DIV
    picGraficos.CurrentX = 0
    picGraficos.CurrentY = Ang(0)
    Tiempo = 0
    picGraficos.DrawWidth = ESPESORgRAFIcos

    For Ies = 1 To NUMmAXpUNTOS Step 1
        Tiempo = Tiempo + DeltaT
        picGraficos.Line -(Tiempo, Ang(Ies)), VERDE
    Next Ies
Else

    msg = "Debe ejecutar los 'Calculos' del menu de 'Simulación' primero"
    Mostrar
End If
End Sub
```

```
Private Sub mnuAnimacion_Click()
'-----
On Error Resume Next 'habilitación de manejo de error
Err.Clear
'-----
If mnuCalculos.Tag = 1 Then

    Dim YPista As Single

    'Ubicación del carrito y el péndulo en el centro del cuadro
    picGraficos.Picture = LoadPicture() 'Limpieza de pantalla
```

```

picGraficos.ScaleMode = 1      'Definiendo la escala en Twips

ImgCarro.Top = picGraficos.Height / 2 - ImgCarro.Height / 2
ImgCarro.Left = picGraficos.Width / 2 - ImgCarro.Width / 2
LinePendulo.X1 = ImgCarro.Left + ImgCarro.Width / 2
LinePendulo.X2 = LinePendulo.X1
LinePendulo.Y1 = ImgCarro.Top
LinePendulo.Y2 = picGraficos.Height / 4
lblTitulo.Caption = "Simulación Dinámica"
UbicacionControles
picGraficos.Scale (-CuadroX, CuadroY)-(CuadroX, -CuadroY)
picGraficos.Cls
picGraficos.DrawWidth = ESPESOrpISTA

'Se define la altura del borde de la pista
YPista = -ImgCarro.Height + ImgCarro.Top
'
'
'      YPista |   L/2   |   L/2   | |
'      |-----|-----|
'          C/2           C/2
'      Long/20
'
'Se dibuja la pista para el carrito
'
'Dibujo de la pista aumentando el tamaño del carrito

picGraficos.Line (-LongPista / 2 - ImgCarro.Width / 2, _
                 0)-(-LongPista _
                 / 2 - ImgCarro.Width / 2, YPista), NEGRO

picGraficos.Line -(LongPista / 2 + ImgCarro.Width / 2, YPista), NEGRO
picGraficos.Line -(LongPista / 2 + ImgCarro.Width / 2, 0), NEGRO

ComanVisi_Sim
Tsim = 0
cmdCongelar.Caption = "CONGELAR"
TimerSim.Enabled = True
Else
    msg = "Debe primero ejecutar los 'Calculos' del menu de 'Simulación'"
    Mostrar
End If

End Sub

Private Sub mbuscar_Click()
'-----
On Error Resume Next ' Manejo de Error
Err.Clear
'-----

With cdgDialogo

```

```

.HelpCommand = cdIHelpPartialKey
.HelpFile = "C:\NIGD\CONTROL\VB\PROGRAMA\pe_fuzzy.HLP"
.HelpFile = "pe_fuzzy.HLP"
.ShowHelp
End With

'-----
If Err.Number <> 0 Then 'Chequeo, de si existe error

    msg = "Error # " & Str(Err.Number) & " Se ha Generado un Error Tipo " _
        & Chr(13) & Err.Description & Chr(13) & "Hay Problemas con el Archivo de Ayuda"

    Estilo = vbOKOnly + vbCritical ' Define botones
    Titulo = "Error en la Ayuda" ' Define título
    MsgBox msg, Estilo, Titulo
End If
'-----
End Sub

Private Sub mnuCalculadora_Click()
'-----
On Error Resume Next ' habilitación de manejo de error
Err.Clear
'-----
    Dim MyAppID As String

    MyAppID = Shell("calc.exe", 1) 'Path del programa calculadora

End Sub

Private Sub mnuCalculos_Click()
'-----
On Error GoTo ManejoError ' habilitación de manejo de error
Err.Clear
'-----

If mnuSelecionControl.Tag = 1 Then

    Ies = 0 'Se inicializa contador de arreglos

ComandosInvisibles
    Select Case TipoControl
    ' Realimentación de estado
    Case CONTROLrESTADOS
        CalculosRealimentacionEstado

    ' Control por Redes
    Case CONTROLrEDES

        If TipoRedAdelanto = REDaDELbISECTRIZ Then

```

```

        CalculosAdelBisectriz
    Else
        CalculosAdelantoCero
    End If
' Acciones de control Proporcional Derivativa
Case CONTROLACCIONES

    If TipoAcciones = ACCIONPD_WnE Then
        CalculosAcciones_WnE

    ElseIf TipoAcciones = ACCIONPD Then
        CalculosAccionesPD_KpKd

    ElseIf TipoAcciones = ACCIONPID Then
        CalculosAccionesPID

    End If

' Control Fuzzy
Case CONTROLFUZZY
    CalculosControlFuzzy

End Select

If Mes >= NUMmAXpUNTOS Then
    Beep
    Beep
    mnuCalculos.Tag = 1
End If

Else

    msg = "Debe Seleccionar el Tipo de Control primero"
    Mostrar
End If
'-----
ManejoError:      'Manejo de Errores

If Err.Number <> 0 Then 'Chequeo de, si existe error

    msg = "Error # " & Str(Err.Number) & " Se ha Generado un Error Tipo " _
        & Chr(13) & Err.Description & Chr(13) & "Ingrese Nuevos Datos"

    Estilo = vbOKOnly + vbCritical ' Define botones
    Titulo = "Error" ' Define título

    MsgBox msg, Estilo, Titulo
    lblBarraBase.Visible = False
    lblBarraProceso.Visible = False
End If
'-----
End Sub

```

```
Private Sub mnuContenido_Click()
```

```
'-----
On Error Resume Next ' Manejo de Error
Err.Clear
'-----
```

```
With cdgDialogo
```

```
.HelpCommand = cdIHelpContents
```

```
'HelpFile = "C:\NIGD\CONTROL\VB\PROGRAMA\pe_fuzzy.HLP"
```

```
.HelpFile = "pe_fuzzy.HLP"
```

```
.ShowHelp
```

```
End With
```

```
'-----
If Err.Number <> 0 Then 'Chequeo de, si existe error
```

```
msg = "Error # " & Str(Err.Number) & " Se ha Generado un Error Tipo " _
& Chr(13) & Err.Description & Chr(13) & "Hay Problemas con el Archivo de Ayuda"
```

```
Estilo = vbOKOnly + vbCritical ' Define botones
```

```
Titulo = "Error en la Ayuda" ' Define titulo
```

```
MsgBox msg, Estilo, Titulo
```

```
End If
```

```
'-----
End Sub
```

```
Private Sub mnuCopiar_Click()
```

```
'-----
On Error Resume Next ' habilitación de manejo de error
Err.Clear
'-----
```

```
If picGraficos.Visible = True Then
```

```
'Definición de variables para la ubicación de titulos temporales
```

```
'Se copia los títulos y escalas del eje x y eje y en etiquetas que solo serán
```

```
'visibles un instante en el momento de la copia
```

```
Dim AltoVentana As Single 'Guarda el alto de la ventana de gráficos para dos
```

```
Dim AnchoVentana As Single 'Guarda el ancho de la ventana de gráficos para dos
```

```
Const CF_BITMAP = 2 ' DEFINO FORMATO DE BITMAP.
```

```
If lblTitulo.Caption = "Velocidad Angular Vs Angulo" Then
```

```
AltoVentana = Abs(picGraficos.ScaleHeight / 2)
```

```
AnchoVentana = Abs(picGraficos.ScaleWidth / 20)
```

```
Else
```

```
AltoVentana = Abs(picGraficos.ScaleHeight / 2)
```

```
AnchoVentana = Abs(picGraficos.ScaleWidth / 2)
```

```
End If
```

```

If (lblTitulo.Caption <> "Espacio Vs Tiempo") Or _
(FdeX1max < 0) Then

    picGraficos.CurrentX = AnchoVentana
    picGraficos.CurrentY = AltoVentana * 19 / 20
    picGraficos.Font.Bold = True
    picGraficos.Font.Size = 11
    picGraficos.Font.Underline = True
    picGraficos.Print lblTitulo.Caption

    picGraficos.CurrentX = AnchoVentana
    picGraficos.CurrentY = AltoVentana * 8 / 10
    picGraficos.Font.Bold = False
    picGraficos.Font.Size = 9
    picGraficos.Font.Underline = False
    picGraficos.Print "Esc.Eje X=" & lblEscalaEjeX.Caption

    picGraficos.CurrentX = AnchoVentana
    picGraficos.CurrentY = AltoVentana * 5 / 7
    picGraficos.Print "Esc.Eje Y=" & lblEscalaEjeY.Caption

Else

    picGraficos.CurrentX = AnchoVentana
    picGraficos.CurrentY = -AltoVentana * 10 / 15
    picGraficos.Font.Bold = True
    picGraficos.Font.Size = 11
    picGraficos.Font.Underline = True
    picGraficos.Print lblTitulo.Caption

    picGraficos.CurrentX = AnchoVentana
    picGraficos.CurrentY = -AltoVentana * 8 / 10
    picGraficos.Font.Bold = False
    picGraficos.Font.Size = 9
    picGraficos.Font.Underline = False
    picGraficos.Print "Esc.Eje X=" & lblEscalaEjeX.Caption

    picGraficos.CurrentX = AnchoVentana
    picGraficos.CurrentY = -AltoVentana * 18 / 20
    picGraficos.Print "Esc.Eje Y=" & lblEscalaEjeY.Caption

End If

Clipboard.Clear ' BORRO Clipboard.

picGraficos.Picture = picGraficos.Image
Clipboard.SetData picGraficos.Picture, CF_BITMAP

'SavePicture picGraficos.Picture, "GRA_TES.BMP"

Else
    msg = "No hay Gráfico que Copiar al Portapapeles"
    Mostrar
End If

```

'Manejo de errores

If Err.Number <> 0 Then 'Chequeo de, si existe error

msg = "Error # " & Str(Err.Number) & " Se ha Producido " _
& Chr(13) & "Intente Nuevamente"

Estilo = vbOKOnly + vbCritical ' Define botones
Titulo = "Error al Copiar al Portapapeles" ' Define titulo
MsgBox msg, Estilo, Titulo

End If

End Sub

Private Sub mnuEspacioVsTiempo_Click()

On Error Resume Next ' habilitación de manejo de error
Err.Clear

If mnuCalculos.Tag = 1 Then

Dim Tiempo As Single

'Dim Ies As Integer

lblTitulo.Caption = "Espacio Vs Tiempo"

UbicacionControles

lblEscalaEjeX.Caption = Format(Tmax / DIV, "0.000") & " seg/div"

lblEscalaEjeY.Caption = Format(FdeX1max / DIV, "0.00E+00") & " m/div"

picGraficos.Scale (-Tmax / (3 * DIV), FdeX1max + FdeX1max / DIV) -
(Tmax + Tmax / (3 * DIV), -FdeX1max - FdeX1max / DIV)

picGraficos.Cls

picGraficos.Picture = LoadPicture()

picGraficos.DrawWidth = ESPESOReJES

'Subrutina para dibujar ejes

DibujoEjes Tmax, FdeX1max, DIV

picGraficos.CurrentX = 0

picGraficos.CurrentY = Esp(0)

Tiempo = 0

picGraficos.DrawWidth = ESPESORgGRAFICOS

For Ies = 1 To NUMmAXpUNTOS Step 1

Tiempo = Tiempo + DeltaT

picGraficos.Line -(Tiempo, Esp(Ies)), ROJO

Next Ies

Else

msg = "Debe ejecutar los 'Calculos' del menu de 'Simulación' primero"

Mostrar

End If

End Sub

```

Private Sub mnuFuerzaVsTiempo_Click()
'-----
On Error Resume Next 'habilitación de manejo de error
Err.Clear
'-----
If mnuCalculos.Tag = 1 Then
Dim Tiempo As Single
'Dim Ies As Integer
lblTitulo.Caption = "Fuerza Vs Tiempo"
UbicacionControles
lblEscalaEjeX.Caption = Format(Tmax / 10, "0.000") & " seg/div"
lblEscalaEjeY.Caption = Format(FdeU5max / 10, "0.000") _
& " Newt/div"
picGraficos.Scale (-Tmax / (3 * DIV), FdeU5max + FdeU5max / DIV)- _
(Tmax + Tmax / (3 * DIV), -FdeU5max - FdeU5max / DIV)
picGraficos.Cls
picGraficos.Picture = LoadPicture()
picGraficos.DrawWidth = ESPESOReJES
DibujoEjes Tmax, FdeU5max, DIV

picGraficos.CurrentX = 0
picGraficos.CurrentY = Fuerza(0)
Tiempo = 0
picGraficos.DrawWidth = ESPESORgGRAFICOS
For Ies = 1 To NUMmAXpUNTOS Step 1
Tiempo = Tiempo + DeltaT
picGraficos.Line -(Tiempo, Fuerza(Ies)), LILA
Next Ies
Else

msg = "Debe ejecutar los 'Calculos' del menu de 'Simulación' primero"
Mostrar
End If
End Sub

Private Sub mnuImprimir_Click()
'-----
On Error GoTo ErrorHandler 'habilitación de manejo de error
Err.Clear
'-----
If picGraficos.Visible = True Then

If (cmdCongelar.Caption = "CONGELAR") And (TimerSim.Enabled = True) Then
cmdCongelar.Caption = "DESCONGELAR"
TimerSim.Enabled = False

End If
cdgDialogo.Flags = cdlPDNoSelection Or cdlPDNoPageNums Or _
cdlPDHidePrintToFile Or cdlPDUseDevModeCopies
cdgDialogo.ShowPrinter
picGraficos.Picture = picGraficos.Image
picGraficos_m.Picture = picGraficos_m.Image

```

```

PrintForm

Else
    msg = "No hay Gráfico que Imprimir"
    Mostrar
End If
'-----
ErrorHandler:      'Manejo de Errores

If Err.Number <> 0 Then 'Chequeo de si existe error

    If Err.Number <> 32755 Then

        msg = "Error #" & Str(Err.Number) & " Se ha producido un " _
            & "error de impresión " _
            & Chr(13) & "Intente nuevamente"

        Estilo = vbOKOnly + vbCritical ' Define botones
        Titulo = "Error de Impresión" ' Define titulo
        MsgBox msg, Estilo, Titulo
    End If
End If
'-----

End Sub

Private Sub mnuNuevoPendulo_Click()
'-----
On Error Resume Next 'habilitación de manejo de error
Err.Clear
'-----
    mnuNuevoPendulo.Tag = 1
    ComandosInvisibles
    PenduloNuevo
End Sub

Private Sub mnuPenduloPorDefecto_Click()
'-----
On Error Resume Next 'habilitación de manejo de error
Err.Clear
'-----
    mnuPenduloPorDefecto.Tag = 1
    ComandosInvisibles

With frmDatosPendulo
    .txtMc.Locked = True
    .txtmP.Locked = True
    .txtLongP.Locked = True
    .txtGr.Locked = True
    .txtLongPista.Locked = True

```

```

.txtJpendulo.Locked = True
.txtEspIni.Locked = True
.txtVelIni.Locked = True
.txtAngIni.Locked = True
.txtVelAngIni.Locked = True

End With

PenduloDefecto

With frmDatosPendulo
.txtMc.Locked = False
.txtmP.Locked = False
.txtLongP.Locked = False
.txtGr.Locked = False
.txtLongPista.Locked = False
.txtJpendulo.Locked = False
.txtEspIni.Locked = False
.txtVelIni.Locked = False
.txtAngIni.Locked = False
.txtVelAngIni.Locked = False
End With

End Sub

Private Sub mnuRelej_Click()
'-----
On Error Resume Next 'habilitación de manejo de error
Err.Clear
'-----
Dim MyAppID As String

MyAppID = Shell("clock.exe", 1) 'Se ejecuta el programa reloj

End Sub

Private Sub mnuResultados_Click()
'-----
On Error Resume Next 'habilitación de manejo de error
Err.Clear
'-----
frmResultados.Show 1
End Sub

Private Sub mnuSalir_Click()

msg = "Esta seguro que desea Salir ?" ' Se define el mensaje.
Estilo = vbYesNo + vbQuestion + vbDefaultButton1 ' Define botones

```

```

Titulo = "Salir del Programa"           ' Define Título
Help = "pe_fuzzy.HLP"                 ' Define archivo de ayuda
Ctxt = 130                             ' Define tópico

Response = MsgBox(msg, Estilo, Titulo, Help, Ctxt) ' Display message.

If Response = vbYes Then ' Respuesta si.
    Unload frmResultados
    Unload frmDatosPendulo
    Unload frmTiposDeControl
    Unload frmPresenacion
    Unload frmAlternativa
    Unload frmGraficosVarios
    End ' Acción de salir del programa
Else ' Respuesta no
    ' No sale del programa
End If

End Sub

'Realimentación de Estado
'Ingreso de datos de los polos de lazo cerrado
Private Sub LectPoloDes()

'Discriminación del Caso de realimentación de estado y da un valor a la
'variable CasoRealim

If (PoloD(1).I <> 0 Or PoloD(3).I <> 0) Then
    If (PoloD(1).I <> 0 And PoloD(3).I <> 0) Then
        If ((PoloD(1).I = PoloD(3).I) And _
            (PoloD(1).r = PoloD(3).r)) Then
            CasoRealim = DOSpcIG
        Else
            CasoRealim = DOSpcDIF
        End If
    Else
        If (PoloD(1).I = 0) Then
            If (PoloD(1).r = PoloD(2).r) Then
                CasoRealim = UNpcDOSprIG
            Else
                CasoRealim = UNpcDOSprDIF
            End If
        Else
            If (PoloD(3).r = PoloD(4).r) Then
                CasoRealim = UNpcDOSprIG
            Else
                CasoRealim = UNpcDOSprDIF
            End If
        End If
    End If
End If
'Casos de polos reales
Else

```

```

If (PoloD(1).r = PoloD(2).r And PoloD(1).r = _
    PoloD(3).r And PoloD(1).r = PoloD(4).r) Then
    CasoRealim = CUATROprIG
ElseIf ((PoloD(1).r = PoloD(2).r And PoloD(1).r = _
    PoloD(3).r) Or (PoloD(1).r = PoloD(2).r And PoloD(1).r = _
    PoloD(4).r) Or (PoloD(2).r = PoloD(3).r And PoloD(2).r = _
    PoloD(4).r) Or (PoloD(1).r = PoloD(3).r And PoloD(1).r = _
    PoloD(4).r)) Then
    CasoRealim = TRESprIGUNprDIF
ElseIf ((PoloD(1).r <> PoloD(2).r) And (PoloD(1).r <> PoloD(3).r) And _
    (PoloD(1).r <> PoloD(4).r) And (PoloD(2).r <> PoloD(3).r) And _
    (PoloD(2).r <> PoloD(4).r) And (PoloD(3).r <> PoloD(4).r)) Then
    CasoRealim = CUATROprDIF
ElseIf ((PoloD(1).r = PoloD(2).r And PoloD(3).r = PoloD(4).r) Or _
    (PoloD(1).r = PoloD(3).r And PoloD(2).r = PoloD(4).r) Or _
    (PoloD(1).r = PoloD(4).r And PoloD(2).r = PoloD(3).r)) Then
    CasoRealim = DOSprIGDOSprIG
Else
    CasoRealim = DOSprIGDOSprDIF
End If
End If
End Sub

```

'Realimentación de Estado

'Cálculo de los coeficientes de el polinomio característico $q(s)$
'del sistema en base a los polos de lazo cerrado ingresados.
'Cálculo de los elementos del vector de realimentación K_1, K_2, K_3, K_4
' $q(s) = s^4 + AcDesed(3) * s^3 + AcDesed(2) * s^2 + AcDesed(1) * s + AcDesed(0)$

Private Sub CalcCoef()

AcDesed(0) = Multcomp(Multcomp(PoloD(1), PoloD(2)), Multcomp(PoloD(3), _
PoloD(4))).r

AcDesed(1) = -(SumaComp(SumaComp(Multcomp(PoloD(1), Multcomp(PoloD(2), _
PoloD(3))), Multcomp(PoloD(1), Multcomp(PoloD(3), PoloD(4))), _
SumaComp(Multcomp(PoloD(1), Multcomp(PoloD(2), PoloD(4))), _
Multcomp(PoloD(2), Multcomp(PoloD(3), PoloD(4)))).r

AcDesed(2) = SumaComp(SumaComp(SumaComp(SumaComp(SumaComp(Multcomp(PoloD(1), _
PoloD(2)), Multcomp(PoloD(1), PoloD(3))), Multcomp(PoloD(1), PoloD(4))), _
Multcomp(PoloD(2), PoloD(3))), Multcomp(PoloD(2), PoloD(4))), _
Multcomp(PoloD(3), PoloD(4))).r

AcDesed(3) = -(SumaComp(SumaComp(SumaComp(PoloD(1), PoloD(2)), PoloD(3)), _
PoloD(4))).r

'CALCULO DEL VECTOR DE REALIMENTACION DE ESTADO

Kest(1) = -AcDesed(0) * Lp * Mc / Gr

Kest(2) = -AcDesed(1) * Lp * Mc / Gr

Kest(3) = -Lp * Mc * (Lp * AcDesed(0) / Gr + Gr / Lp + AcDesed(2))

Kest(4) = -Lp * Mc * (Lp * AcDesed(1) / Gr + AcDesed(3))

End Sub

'Realimentación de Estado

'Subrutina que organiza los polos ingresados a un modo estandar

Private Sub OrganPolos()

'Selección para ordenar los polos en el orden adecuado

Select Case CasoRealim

Case DOSprIGDOSprDIF

 If PoloD(1).r = PoloD(3).r Then

 SwaP PoloD(2).r, PoloD(3).r

 ElseIf PoloD(1).r = PoloD(4).r Then

 SwaP PoloD(2).r, PoloD(4).r

 ElseIf PoloD(2).r = PoloD(3).r Then

 SwaP PoloD(1).r, PoloD(3).r

 ElseIf PoloD(2).r = PoloD(4).r Then

 SwaP PoloD(1).r, PoloD(4).r

 ElseIf PoloD(3).r = PoloD(4).r Then

 SwaP PoloD(1).r, PoloD(3).r

 SwaP PoloD(2).r, PoloD(4).r

 End If

Case TRESprIGUNprDIF ' Tres polos reales repetidos

 If (PoloD(1).r = PoloD(2).r And PoloD(1).r = PoloD(4).r) Then

 SwaP PoloD(3).r, PoloD(4).r

 ElseIf (PoloD(2).r = PoloD(3).r And PoloD(2).r = PoloD(4).r) Then

 SwaP PoloD(1).r, PoloD(4).r

 ElseIf (PoloD(1).r = PoloD(3).r And PoloD(1).r = PoloD(4).r) Then

 SwaP PoloD(2).r, PoloD(4).r

 End If

Case CUATROprIG ' Cuatro polos reales iguales

Case DOSprIGDOSprIG 'Dos polos reales iguales dos polos reales iguales

 If (PoloD(1).r = PoloD(3).r) Then

 SwaP PoloD(2).r, PoloD(3).r

 ElseIf (PoloD(1).r = PoloD(4).r) Then

 SwaP PoloD(2).r, PoloD(4).r

 End If

Case UNpcDOSprDIF, UNpcDOSprIG

 If (PoloD(1).I = 0) Then

 SwaP PoloD(1).I, PoloD(3).I

 SwaP PoloD(1).r, PoloD(3).r

 SwaP PoloD(2).I, PoloD(4).I

 SwaP PoloD(2).r, PoloD(4).r

 End If

Case Else ' Cuatro polos reales diferentes

' Dos polos complejos conjugados distintos

' Dos polos complejos conjugados iguales

End Select

With frmTiposDeControl

 .txtParteReal(1).Text = CStr(PoloD(1).r)

 .txtParteReal(2).Text = CStr(PoloD(2).r)

```

.txtParteReal(3).Text = CStr(PoloD(3).r)
.txtParteReal(4).Text = CStr(PoloD(4).r)
If PoloD(1).I = 0 Then
.txtParteImag(1).Text = ""
.lblparteImag(2).Caption = ""
Else
.txtParteImag(1).Text = CStr(PoloD(1).I)
.lblparteImag(2).Caption = CStr(-PoloD(2).I)
End If

If PoloD(3).I = 0 Then
.txtParteImag(3).Text = ""
.lblparteImag(4).Caption = ""
Else
.txtParteImag(3).Text = CStr(PoloD(3).I)
.lblparteImag(4).Caption = CStr(-PoloD(4).I)
End If
End With
End Sub

'Realimentación de Estado
'Subrutina que realiza un SWAP entre dos valores
Private Sub SwaP(Nume1 As Single, Nume2 As Single)
Dim Temp As Single
Temp = Nume1
Nume1 = Nume2
Nume2 = Temp
End Sub

'Valores del péndulo que por defecto son seteados que
'permiten realizar los cálculos de las compensaciones
Private Sub PenduloDefecto()
mnuSeleccionControl.Tag = 0
mnuCalculos.Tag = 0
With frmDatosPendulo
.txtMc.Text = Format(MASAcARRO, "0.00")
.txtmP.Text = Format(MASApENDULO, "0.00")
.txtLongP.Text = Format(LONGpEND, "0.00")
.txtGr.Text = Format(GRAVeDAD, "0.00")
.txtLongPista.Text = Format(LONGITUDpISTA, "0.00")
.txtJpendulo.Text = Format(iNERCIaJ, "0.00000")
.txtEspIni.Text = Format(ESPiNIC, "0.00")
.txtVelIni.Text = Format(VELiNIC, "0.00")
.txtAngIni.Text = Format(ANGiNIC, "0.00")
.txtVelAngIni.Text = Format(VELaNGiNIC, "0.00")
End With
frmDatosPendulo.Show 1
With frmDatosPendulo
If frmDatosPendulo.txtChequeoBoton.Text = "OK" Then
'Se presionó el boton "OK"

```

```

Mc = CSng(.txtMc.Text)
mP = CSng(.txtmP.Text)
LongP = CSng(.txtLongP.Text)
Gr = CSng(.txtGr.Text)
LongPista = CSng(.txtLongPista.Text)
Jp = CSng(.txtJpendulo)
Xest(1) = CSng(.txtEspIni.Text)
Xest(2) = CSng(.txtVelIni.Text)
Xest(3) = CSng(.txtAngIni.Text) * GRADOsaRAD
Xest(4) = CSng(.txtVelAngIni.Text) * GRADOsaRAD
InicializacionVariables
ResetVentanaResultados
MuestroResGenerales
ElseIf frmDatosPendulo.txtChequeoBoton.Text = "CANCEL" Then
'Se presionó el boton "CANCEL"
.txtMc.Text = Format(Mc, "0.00")
.txtmP.Text = Format(mP, "0.00")
.txtLongP.Text = Format(LongP, "0.00")
.txtGr.Text = Format(Gr, "0.00")
.txtLongPista.Text = Format(LongPista, "0.00")
.txtJpendulo.Text = Format(Jp, "0.00000")
.txtEspIni.Text = Format(Xest(1), "0.00")
.txtVelIni.Text = Format(Xest(2), "0.00")
.txtAngIni.Text = Format((Xest(3) * RADaGRADOs), "0.00")
.txtVelAngIni.Text = Format((Xest(3) * RADaGRADOs), "0.00")
Else
'No se presionó "nada"
End If
End With

If frmDatosPendulo.cmdModificar.Tag = 0 Then 'Péndulo con centro de gravedad en

    Jp = mP * LongP * LongP / 3 'el centro
Else
    Jp = Jp 'el dato de Jp es mantenido como dato
    'de ingreso
End If

Lp = (Jp + mP * LongP ^ 2) / (mP * LongP)
CuadroX = LongPista * 13 / 20 'Para definir
    'la escala "X" del área donde
    'va a estar el carrito para la animación
CuadroY = LongPista / 4 'Para definir la escala "y" del área donde
    'va a estar el carrito para la animación
    'picGraficos.Scale(-x,y)-(x,-y)

End Sub

'Realimentación de Estado
'Cálculo de los coeficientes del polinomio del numerador de cada estado

```

```

'X1(s)=(b(1,3)*s^3+b(1,2)*s^2+b(1,1)*s+b(1,0))/q(s)
'X2(s)=(b(2,3)*s^3+b(2,2)*s^2+b(2,1)*s+b(2,0))/q(s)
'X3(s)=(b(3,3)*s^3+b(3,2)*s^2+b(3,1)*s+b(3,0))/q(s)
'X4(s)=(b(4,3)*s^3+b(4,2)*s^2+b(4,1)*s+b(4,0))/q(s)
Private Sub CoefNumeEstado()
b(1, 3) = Xest(1)
b(1, 2) = (Kest(2) - Kest(4) / Lp) * Xest(1) / Mc + Xest(2)
b(1, 1) = -(Gr + Kest(3) / Mc) * Xest(1) / Lp - Kest(4) * _
    Xest(2) / (Lp * Mc) - Kest(3) * Xest(3) / Mc - _
    Kest(4) * Xest(4) / Mc
b(1, 0) = -Gr * Kest(2) * Xest(1) / (Lp * Mc) - (Gr + Kest(3) / _
    Mc) * Xest(2) / Lp - Kest(4) * Gr * Xest(3) / (Lp * Mc) _
    - Kest(3) * Xest(4) / Mc
b(2, 3) = Xest(2)
b(2, 2) = -Kest(1) * Xest(1) / Mc - Kest(4) * Xest(2) / (Lp * Mc) _
    - Kest(3) * Xest(3) / Mc - Kest(4) * Xest(4) / Mc
b(2, 1) = -(Gr + Kest(3) / Mc) * Xest(2) / Lp - Kest(4) * Gr * _
    Xest(3) / (Lp * Mc) - Kest(3) * Xest(4) / Mc
b(2, 0) = Kest(1) * Gr * Xest(1) / (Lp * Mc)

b(3, 3) = Xest(3)
b(3, 2) = (Kest(2) - Kest(4) / Lp) * Xest(3) / Mc + Xest(4)
b(3, 1) = (Kest(1) * Xest(1) + Kest(2) * Xest(2)) / (Lp * _
    Mc) + (Kest(1) * Xest(3) + Kest(2) * Xest(4)) / Mc
b(3, 0) = (Kest(1) * Xest(2) / Lp + Kest(1) * Xest(4)) / Mc
b(4, 3) = Xest(4)
b(4, 2) = (Kest(1) * Xest(1) + Kest(2) * Xest(2)) / (Lp * Mc) _
    + (Gr + Kest(3) / Mc) * Xest(3) / Lp + Kest(2) * _
    Xest(4) / Mc
b(4, 1) = (Kest(1) * Xest(2) + Kest(2) * Gr * Xest(3)) / _
    (Lp * Mc) + Kest(1) * Xest(4) / Mc
b(4, 0) = Kest(1) * Gr * Xest(3) / (Lp * Mc)
End Sub

```

'Subrutina que encuentra las incognitas de un sistema de
'ecuaciones de un sistema de N ecuaciones

```
Private Sub SistemaEcuaciones(N As Integer, Asis() As Double, _
    Xsis() As Double)
```

```
Dim P As Integer, Q As Integer, K As Integer
```

```
Dim II As Integer, JJ As Integer, I As Integer
```

```
Dim J As Integer, T As Integer, M As Integer
```

```
Dim Lsis As Integer, JIsis As Integer, Gsis As Integer
```

```
Dim Bsis As Double, Wsis As Double, Ssis As Double
```

```
P = N + 1
```

```
Q = N - 1
```

```
For K = 1 To Q
```

```
    II = K + 1
```

```
    JJ = K + 1
```

```
    For I = II To N
```

```
        For J = JJ To P
```

```
            If (Asis(K, K) = 0) Then
```

```
                T = K + 1
```

```

    For M = K To P
        Bsis = Asis(K, M)
        Asis(K, M) = Asis(T, M)
        Asis(T, M) = Bsis
    Next M
End If
Wsis = Asis(I, K) / Asis(K, K)
Asis(I, J) = Asis(I, J) - Asis(K, J) * Wsis
Next J
Next I
Next K
Gsis = N
If Asis(Gsis, Gsis) = 0 Then
    'mostrar mensajes

    msg = "NO HAY SOLUCION"
    Mostrar
    'mostrar mensajes
    Exit Sub

End If
Xsis(Gsis) = Asis(Gsis, P) / Asis(Gsis, Gsis)
For I = 1 To Q
    Lsis = N - I
    JIsis = Lsis + 1
    Ssis = 0
    For J = JIsis To N
        Ssis = Ssis + Asis(Lsis, J) * Xsis(J)
    Next J
    Xsis(Lsis) = (Asis(Lsis, P) - Ssis) / Asis(Lsis, Lsis)
Next I
End Sub

```

```

'Realimentación de Estado
'Cálculo de la constante "A" de la division en Fracciones
'Parciales
Private Function An1CuatroPRdif(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An1CuatroPRdif = (Num3 * PoloD(1).r ^ 3 + Num2 * _
    PoloD(1).r ^ 2 + Num1 * PoloD(1).r + Num0) _
    / ((PoloD(1).r - PoloD(2).r) * (PoloD(1).r - _
    PoloD(3).r) * (PoloD(1).r - PoloD(4).r))
End Function

```

```

'Realimentación de Estado
'Cálculo de la constante "B" de la division en Fracciones
'Parciales
Private Function An2CuatroPRdif(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An2CuatroPRdif = (Num3 * PoloD(2).r ^ 3 + Num2 * _
    PoloD(2).r ^ 2 + Num1 * PoloD(2).r + Num0) _
    / ((PoloD(2).r - PoloD(1).r) * (PoloD(2).r - _
    PoloD(3).r) * (PoloD(2).r - PoloD(4).r))

```

```

End Function
'Realimentación de Estado
'Cálculo de la constante "C" de la division en Fracciones
'Parciales
Private Function An3CuatroPRdif(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An3CuatroPRdif = (Num3 * PoloD(3).r ^ 3 + Num2 * _
    PoloD(3).r ^ 2 + Num1 * PoloD(3).r + Num0) _
    / ((PoloD(3).r - PoloD(1).r) * (PoloD(3).r - _
    PoloD(2).r) * (PoloD(3).r - PoloD(4).r))

```

```

End Function

'Realimentación de Estado
'Cálculo de la constante "D" de la division en Fracciones
'Parciales
Private Function An4CuatroPRdif(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An4CuatroPRdif = (Num3 * PoloD(4).r ^ 3 + Num2 * _
    PoloD(4).r ^ 2 + Num1 * PoloD(4).r + Num0) _
    / ((PoloD(4).r - PoloD(1).r) * (PoloD(4).r - _
    PoloD(2).r) * (PoloD(4).r - PoloD(3).r))
End Function

```

```

'Realimentación de Estado
'Cálculo de la constante "B" de la division en Fracciones
'Parciales
Private Function An2DosPRigDosPRdif(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An2DosPRigDosPRdif = (Num3 * PoloD(1).r ^ 3 + Num2 * _
    PoloD(1).r ^ 2 + Num1 * PoloD(1).r + Num0) _
    / ((PoloD(1).r - PoloD(3).r) * (PoloD(1).r - _
    PoloD(4).r))
End Function

```

```

'Realimentación de Estado
'Cálculo de la constante "C" de la division en Fracciones
'Parciales
Private Function An3DosPRigDosPRdif(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An3DosPRigDosPRdif = (Num3 * PoloD(3).r ^ 3 + Num2 * _
    PoloD(3).r ^ 2 + Num1 * PoloD(3).r + Num0) _
    / ((PoloD(3).r - PoloD(1).r) ^ 2 * (PoloD(3).r - _
    PoloD(4).r))
End Function

```

```

'Realimentación de Estado
'Cálculo de la constante "D" de la division en Fracciones
'Parciales
Private Function An4DosPRigDosPRdif(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An4DosPRigDosPRdif = (Num3 * PoloD(4).r ^ 3 + Num2 * _
    PoloD(4).r ^ 2 + Num1 * PoloD(4).r + Num0) _
    / ((PoloD(4).r - PoloD(1).r) ^ 2 * (PoloD(4).r - _

```

```

    PoloD(3).r))
End Función

```

```

'Realimentación de Estado
'Cálculo de la constante "C" de la division en Fracciones
'Parciales
Private Function An3TresPRigunPRdif(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An3TresPRigunPRdif = (Num3 * PoloD(1).r ^ 3 + Num2 * _
    PoloD(1).r ^ 2 + Num1 * PoloD(1).r + Num0) _
    / (PoloD(1).r - PoloD(4).r)

```

```

End Function
'Realimentación de Estado
'Cálculo de la constante "D" de la division en Fracciones
'Parciales
Private Function An4TresPRigunPRdif(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An4TresPRigunPRdif = (Num3 * PoloD(4).r ^ 3 + Num2 * _
    PoloD(4).r ^ 2 + Num1 * PoloD(4).r + Num0) _
    / (PoloD(4).r - PoloD(1).r) ^ 3
End Function

```

```

'Realimentación de Estado
'Cálculo de la constante "D" de la division en Fracciones
'Parciales :
Private Function An4CuatroPRig(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An4CuatroPRig = Num3 * PoloD(1).r ^ 3 + Num2 * _
    PoloD(1).r ^ 2 + Num1 * PoloD(1).r + Num0
End Function

```

```

'Realimentación de Estado
'Cálculo de la constante "B" de la division en Fracciones
'Parciales
Private Function An2DosPRigDosPRig(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An2DosPRigDosPRig = (Num3 * PoloD(1).r ^ 3 + Num2 * _
    PoloD(1).r ^ 2 + Num1 * PoloD(1).r + Num0) _
    / (PoloD(1).r - PoloD(3).r) ^ 2
End Function

```

```

'Realimentación de Estado
'Cálculo de la constante "C" de la division en Fracciones
'Parciales
Private Function An3DosPRigDosPRig(Num3 As Single, Num2 As _
    Single, NumB As Single, NumD As Single) As Single

An3DosPRigDosPRig = (Num2 + Num3 * (2 * PoloD(3).r + _
    PoloD(1).r) - NumB - NumD) / (PoloD(3).r - _
    PoloD(1).r)
End Function

```

```
'Realimentación de Estado
'Cálculo de la constante "D" de la division en Fracciones
'Parciales
Private Function An4DosPRigDosPRig(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An4DosPRigDosPRig = (Num3 * PoloD(3).r ^ 3 + Num2 * _
    PoloD(3).r ^ 2 + Num1 * PoloD(3).r + Num0) _
    / (PoloD(3).r - PoloD(1).r) ^ 2

End Function
```

```
'Realimentación de Estado
'Cálculo de la constante "An1","An2","An3","An4" de la division en
'
'Fracciones Parciales de los cuatro estados
Private Sub FraccParcCUATROprDIF()
'Constantes de la division en fracciones parciales de X1(s)
a(1, 1) = An1CuatroPRdif(b(1, 3), b(1, 2), b(1, 1), b(1, 0))
a(1, 2) = An2CuatroPRdif(b(1, 3), b(1, 2), b(1, 1), b(1, 0))
a(1, 3) = An3CuatroPRdif(b(1, 3), b(1, 2), b(1, 1), b(1, 0))
a(1, 4) = An4CuatroPRdif(b(1, 3), b(1, 2), b(1, 1), b(1, 0))

'Constantes de la division en fracciones parciales de X2(s)
a(2, 1) = An1CuatroPRdif(b(2, 3), b(2, 2), b(2, 1), b(2, 0))
a(2, 2) = An2CuatroPRdif(b(2, 3), b(2, 2), b(2, 1), b(2, 0))
a(2, 3) = An3CuatroPRdif(b(2, 3), b(2, 2), b(2, 1), b(2, 0))
a(2, 4) = An4CuatroPRdif(b(2, 3), b(2, 2), b(2, 1), b(2, 0))

'Constantes de la division en fracciones parciales de X3(s)
a(3, 1) = An1CuatroPRdif(b(3, 3), b(3, 2), b(3, 1), b(3, 0))
a(3, 2) = An2CuatroPRdif(b(3, 3), b(3, 2), b(3, 1), b(3, 0))
a(3, 3) = An3CuatroPRdif(b(3, 3), b(3, 2), b(3, 1), b(3, 0))
a(3, 4) = An4CuatroPRdif(b(3, 3), b(3, 2), b(3, 1), b(3, 0))

'Constantes de la division en fracciones parciales de X4(s)
a(4, 1) = An1CuatroPRdif(b(4, 3), b(4, 2), b(4, 1), b(4, 0))
a(4, 2) = An2CuatroPRdif(b(4, 3), b(4, 2), b(4, 1), b(4, 0))
a(4, 3) = An3CuatroPRdif(b(4, 3), b(4, 2), b(4, 1), b(4, 0))
a(4, 4) = An4CuatroPRdif(b(4, 3), b(4, 2), b(4, 1), b(4, 0))

End Sub
```

```
'Realimentación de Estado
'Cálculo de la constante "A","B","C","D" de la division en
'Fracciones Parciales de los cuatro estados
Private Sub FraccParcDOSprIGDOSprDIF()
'Constantes de la division en fracciones parciales de X1(s)
a(1, 2) = An2DosPRigDosPRdif(b(1, 3), b(1, 2), b(1, 1), b(1, 0))
a(1, 3) = An3DosPRigDosPRdif(b(1, 3), b(1, 2), b(1, 1), b(1, 0))
a(1, 4) = An4DosPRigDosPRdif(b(1, 3), b(1, 2), b(1, 1), b(1, 0))
```

$$a(1, 1) = b(1, 3) - a(1, 3) - a(1, 4)$$

'Constantes de la division en fracciones parciales de X2(s)

$$a(2, 2) = \text{An2DosPRigDosPRdif}(b(2, 3), b(2, 2), b(2, 1), b(2, 0))$$

$$a(2, 3) = \text{An3DosPRigDosPRdif}(b(2, 3), b(2, 2), b(2, 1), b(2, 0))$$

$$a(2, 4) = \text{An4DosPRigDosPRdif}(b(2, 3), b(2, 2), b(2, 1), b(2, 0))$$

$$a(2, 1) = b(2, 3) - a(2, 3) - a(2, 4)$$

'Constantes de la division en fracciones parciales de X3(s)

$$a(3, 2) = \text{An2DosPRigDosPRdif}(b(3, 3), b(3, 2), b(3, 1), b(3, 0))$$

$$a(3, 3) = \text{An3DosPRigDosPRdif}(b(3, 3), b(3, 2), b(3, 1), b(3, 0))$$

$$a(3, 4) = \text{An4DosPRigDosPRdif}(b(3, 3), b(3, 2), b(3, 1), b(3, 0))$$

$$a(3, 1) = b(3, 3) - a(3, 3) - a(3, 4)$$

'Constantes de la division en fracciones parciales de X4(s)

$$a(4, 2) = \text{An2DosPRigDosPRdif}(b(4, 3), b(4, 2), b(4, 1), b(4, 0))$$

$$a(4, 3) = \text{An3DosPRigDosPRdif}(b(4, 3), b(4, 2), b(4, 1), b(4, 0))$$

$$a(4, 4) = \text{An4DosPRigDosPRdif}(b(4, 3), b(4, 2), b(4, 1), b(4, 0))$$

$$a(4, 1) = b(4, 3) - a(4, 3) - a(4, 4)$$

End Sub

'Realimentación de Estado

'Cálculo de la constante "A", "B", "C", "D" de la division en

'Fracciones Parciales de los cuatro estados

Private Sub FraccParcDOSPrIGDOSPrIG()

'Constantes de la division en fracciones parciales de X1(s)

$$a(1, 2) = \text{An2DosPRigDosPRig}(b(1, 3), b(1, 2), b(1, 1), b(1, 0))$$

$$a(1, 4) = \text{An4DosPRigDosPRig}(b(1, 3), b(1, 2), b(1, 1), b(1, 0))$$

$$a(1, 3) = \text{An3DosPRigDosPRig}(b(1, 3), b(1, 2) _$$

$$\quad , a(1, 2), a(1, 4))$$

$$a(1, 1) = b(1, 3) - a(1, 3)$$

'Constantes de la division en fracciones parciales de X2(s)

$$a(2, 2) = \text{An2DosPRigDosPRig}(b(2, 3), b(2, 2), b(2, 1), b(2, 0))$$

$$a(2, 4) = \text{An4DosPRigDosPRig}(b(2, 3), b(2, 2), b(2, 1), b(2, 0))$$

$$a(2, 3) = \text{An3DosPRigDosPRig}(b(2, 3), b(2, 2) _$$

$$\quad , a(2, 2), a(2, 4))$$

$$a(2, 1) = b(2, 3) - a(2, 3)$$

'Constantes de la division en fracciones parciales de X3(s)

$$a(3, 2) = \text{An2DosPRigDosPRig}(b(3, 3), b(3, 2), b(3, 1), b(3, 0))$$

$$a(3, 4) = \text{An4DosPRigDosPRig}(b(3, 3), b(3, 2), b(3, 1), b(3, 0))$$

$$a(3, 3) = \text{An3DosPRigDosPRig}(b(3, 3), b(3, 2) _$$

$$\quad , a(3, 2), a(3, 4))$$

$$a(3, 1) = b(3, 3) - a(3, 3)$$

'Constantes de la division en fracciones parciales de X4(s)

$$a(4, 2) = \text{An2DosPRigDosPRig}(b(4, 3), b(4, 2), b(4, 1), b(4, 0))$$

$$a(4, 4) = \text{An4DosPRigDosPRig}(b(4, 3), b(4, 2), b(4, 1), b(4, 0))$$

$$a(4, 3) = \text{An3DosPRigDosPRig}(b(4, 3), b(4, 2) _$$

$$\quad , a(4, 2), a(4, 4))$$

$$a(4, 1) = b(4, 3) - a(4, 3)$$

End Sub

'Realimentación de Estado
 'Cálculo de la constante "A","B","C","D" de la division en
 'Fracciones Parciales de los cuatro estados
 Private Sub FraccParcDOSpcDIF()
 ReDim At(1 To 4, 1 To 5) As Double
 ReDim Xt(1 To 4) As Double
 Dim Orden As Integer
 Dim AtPaso As Variant
 Orden = 4
 'Constantes de la division en fracciones parciales de X1(s)

MatrizINIFraParDOSpcDIF At()
 $A_t(1, 5) = b(1, 3)$
 $A_t(2, 5) = b(1, 2)$
 $A_t(3, 5) = b(1, 1)$
 $A_t(4, 5) = b(1, 0)$

SistemaEcuaciones Orden, At(), Xt()

$a(1, 1) = X_t(1)$
 $a(1, 2) = X_t(2)$
 $a(1, 3) = X_t(3)$
 $a(1, 4) = X_t(4)$

'Constantes de la division en fracciones parciales de X2(s)
 MatrizINIFraParDOSpcDIF At()
 $A_t(1, 5) = b(2, 3)$
 $A_t(2, 5) = b(2, 2)$
 $A_t(3, 5) = b(2, 1)$
 $A_t(4, 5) = b(2, 0)$

SistemaEcuaciones Orden, At(), Xt()

$a(2, 1) = X_t(1)$
 $a(2, 2) = X_t(2)$
 $a(2, 3) = X_t(3)$
 $a(2, 4) = X_t(4)$

'Constantes de la division en fracciones parciales de X3(s)
 MatrizINIFraParDOSpcDIF At()
 $A_t(1, 5) = b(3, 3)$
 $A_t(2, 5) = b(3, 2)$
 $A_t(3, 5) = b(3, 1)$
 $A_t(4, 5) = b(3, 0)$
 SistemaEcuaciones Orden, At(), Xt()

$a(3, 1) = X_t(1)$
 $a(3, 2) = X_t(2)$
 $a(3, 3) = X_t(3)$
 $a(3, 4) = X_t(4)$

```
'Constantes de la division en fracciones parciales de X4(s)
MatrizINIFraParDOSpcDIF At()
At(1, 5) = b(4, 3)
At(2, 5) = b(4, 2)
At(3, 5) = b(4, 1)
At(4, 5) = b(4, 0)
```

```
SistemaEcuaciones Orden, At(), Xt()
```

```
a(4, 1) = Xt(1)
a(4, 2) = Xt(2)
a(4, 3) = Xt(3)
a(4, 4) = Xt(4)
```

```
End Sub
```

```
'Realimentación de Estado
'Cálculo de la constante "A", "B", "C", "D" de la division en
'Fracciones Parciales de los cuatro estados
Private Sub FraccParcDOSpcIG()
```

```
'Constantes de la division en fracciones parciales de X1(s)
```

```
a(1, 1) = b(1, 3)
a(1, 2) = (b(1, 2) + 3 * PoloD(1).r * a(1, 1)) _
  / PoloD(1).I
a(1, 3) = b(1, 1) - a(1, 1) * (3 * PoloD(1).r ^ 2 + _
  PoloD(1).I ^ 2) + 2 * PoloD(1).r * PoloD(1).I _
  * a(1, 2)
```

```
a(1, 4) = b(1, 0) + a(1, 1) * (PoloD(1).r ^ 3 + _
  PoloD(1).I ^ 2 * PoloD(1).r) - a(1, 2) * _
  (PoloD(1).I * PoloD(1).r ^ 2 + PoloD(1).I ^ 3) _
  + a(1, 3) * PoloD(1).r
```

```
'Constantes de la division en fracciones parciales de X2(s)
```

```
a(2, 1) = b(2, 3)
a(2, 2) = (b(2, 2) + 3 * PoloD(1).r * a(2, 1)) _
  / PoloD(1).I
a(2, 3) = b(2, 1) - a(2, 1) * (3 * PoloD(1).r ^ 2 + _
  PoloD(1).I ^ 2) + 2 * PoloD(1).r * PoloD(1).I _
  * a(2, 2)
```

```
a(2, 4) = b(2, 0) + a(2, 1) * (PoloD(1).r ^ 3 + _
  PoloD(1).I ^ 2 * PoloD(1).r) - a(2, 2) * _
  (PoloD(1).I * PoloD(1).r ^ 2 + PoloD(1).I ^ 3) _
  + a(2, 3) * PoloD(1).r
```

```
'Constantes de la division en fracciones parciales de X3(s)
```

```
a(3, 1) = b(3, 3)
a(3, 2) = (b(3, 2) + 3 * PoloD(1).r * a(3, 1)) _
  / PoloD(1).I
a(3, 3) = b(3, 1) - a(3, 1) * (3 * PoloD(1).r ^ 2 + _
```

$$\text{PoloD}(1).I^2 + 2 * \text{PoloD}(1).r * \text{PoloD}(1).I _ \\ * a(3, 2)$$

$$a(3, 4) = b(3, 0) + a(3, 1) * (\text{PoloD}(1).r^3 + _ \\ \text{PoloD}(1).I^2 * \text{PoloD}(1).r) - a(3, 2) * _ \\ (\text{PoloD}(1).I * \text{PoloD}(1).r^2 + \text{PoloD}(1).I^3) _ \\ + a(3, 3) * \text{PoloD}(1).r$$

'Constantes de la division en fracciones parciales de X4(s)

$$a(4, 1) = b(4, 3)$$

$$a(4, 2) = (b(4, 2) + 3 * \text{PoloD}(1).r * a(4, 1)) _ \\ / \text{PoloD}(1).I$$

$$a(4, 3) = b(4, 1) - a(4, 1) * (3 * \text{PoloD}(1).r^2 + _ \\ \text{PoloD}(1).I^2) + 2 * \text{PoloD}(1).r * \text{PoloD}(1).I _ \\ * a(4, 2)$$

$$a(4, 4) = b(4, 0) + a(4, 1) * (\text{PoloD}(1).r^3 + _ \\ \text{PoloD}(1).I^2 * \text{PoloD}(1).r) - a(4, 2) * _ \\ (\text{PoloD}(1).I * \text{PoloD}(1).r^2 + \text{PoloD}(1).I^3) _ \\ + a(4, 3) * \text{PoloD}(1).r$$

End Sub

'Realimentación de Estado

'Cálculo de la constante "A","B","C","D" de la division en

'Fracciones Parciales de los cuatro estados

Private Sub FraccParcTRESprIGUNprDIF()

'Constantes de la division en fracciones parciales de X1(s)

$$a(1, 3) = \text{An3TresPRigunPRdif}(b(1, 3), b(1, 2), b(1, 1), b(1, 0))$$

$$a(1, 4) = \text{An4TresPRigunPRdif}(b(1, 3), b(1, 2), b(1, 1), b(1, 0))$$

$$a(1, 1) = b(1, 3) - a(1, 4)$$

$$a(1, 2) = b(1, 2) + a(1, 1) * (\text{PoloD}(4).r + 2 * _ \\ \text{PoloD}(1).r) + 3 * a(1, 4) * \text{PoloD}(1).r$$

'Constantes de la division en fracciones parciales de X2(s)

$$a(2, 3) = \text{An3TresPRigunPRdif}(b(2, 3), b(2, 2), b(2, 1), b(2, 0))$$

$$a(2, 4) = \text{An4TresPRigunPRdif}(b(2, 3), b(2, 2), b(2, 1), b(2, 0))$$

$$a(2, 1) = b(2, 3) - a(2, 4)$$

$$a(2, 2) = b(2, 2) + a(2, 1) * (\text{PoloD}(4).r + 2 * _ \\ \text{PoloD}(1).r) + 3 * a(2, 4) * \text{PoloD}(1).r$$

'Constantes de la division en fracciones parciales de X3(s)

$$a(3, 3) = \text{An3TresPRigunPRdif}(b(3, 3), b(3, 2), b(3, 1), b(3, 0))$$

$$a(3, 4) = \text{An4TresPRigunPRdif}(b(3, 3), b(3, 2), b(3, 1), b(3, 0))$$

$$a(3, 1) = b(3, 3) - a(3, 4)$$

$$a(3, 2) = b(3, 2) + a(3, 1) * (\text{PoloD}(4).r + 2 * _ \\ \text{PoloD}(1).r) + 3 * a(3, 4) * \text{PoloD}(1).r$$

'Constantes de la division en fracciones parciales de X4(s)

$$a(4, 3) = \text{An3TresPRigunPRdif}(b(4, 3), b(4, 2), b(4, 1), b(4, 0))$$

$$a(4, 4) = \text{An4TresPRigunPRdif}(b(4, 3), b(4, 2), b(4, 1), b(4, 0))$$

$$a(4, 1) = b(4, 3) - a(4, 4)$$

$$a(4, 2) = b(4, 2) + a(4, 1) * (\text{PoloD}(4).r + 2 * _ \\ \text{PoloD}(1).r)$$

```

      PoloD(1).r) + 3 * a(4, 4) * PoloD(1).r
End Sub

```

```

'Realimentación de Estado
'Cálculo de la constante "A","B","C","D" de la division en
'Fracciones Parciales de los cuatro estados
Private Sub FraccParcCUATROprIG()
'Constantes de la division en fracciones parciales de X1(s)
a(1, 4) = An4CuatroPRig(b(1, 3), b(1, 2), b(1, 1), b(1, 0))
a(1, 1) = b(1, 3)
a(1, 2) = b(1, 2) + 3 * PoloD(1).r * b(1, 3)
a(1, 3) = b(1, 1) - 3 * PoloD(1).r ^ 2 * a(1, 1) _
      + 2 * a(1, 2) * PoloD(1).r
'Constantes de la division en fracciones parciales de X2(s)
a(2, 4) = An4CuatroPRig(b(2, 3), b(2, 2), b(2, 1), b(2, 0))
a(2, 1) = b(2, 3)
a(2, 2) = b(2, 2) + 3 * PoloD(1).r * b(2, 3)
a(2, 3) = b(2, 1) - 3 * PoloD(1).r ^ 2 * a(2, 1) _
      + 2 * a(2, 2) * PoloD(1).r
'Constantes de la division en fracciones parciales de X3(s)
a(3, 4) = An4CuatroPRig(b(3, 3), b(3, 2), b(3, 1), b(3, 0))
a(3, 1) = b(3, 3)
a(3, 2) = b(3, 2) + 3 * PoloD(1).r * b(3, 3)
a(3, 3) = b(3, 1) - 3 * PoloD(1).r ^ 2 * a(3, 1) _
      + 2 * a(3, 2) * PoloD(1).r
'Constantes de la division en fracciones parciales de X4(s)
a(4, 4) = An4CuatroPRig(b(4, 3), b(4, 2), b(4, 1), b(4, 0))
a(4, 1) = b(4, 3)
a(4, 2) = b(4, 2) + 3 * PoloD(1).r * b(4, 3)
a(4, 3) = b(4, 1) - 3 * PoloD(1).r ^ 2 * a(4, 1) _
      + 2 * a(4, 2) * PoloD(1).r
End Sub

```

```

'Realimentación de Estado
'Cálculo de la constante "A","B","C","D" de la division en
'Fracciones Parciales de los cuatro estados
Private Sub FraccParcUNpcDOSprDIF()
'Constantes de la division en fracciones parciales de X1(s)
a(1, 1) = An1UnPCdosPRdif(b(1, 3), b(1, 2), b(1, 1), b(1, 0))
a(1, 2) = An2UnPCdosPRdif(b(1, 3), b(1, 2), b(1, 1), b(1, 0))
a(1, 3) = b(1, 3) - a(1, 1) - a(1, 2)
a(1, 4) = An4UnPCdosPRdif(b(1, 2), a(1, 1), a(1, 2), a(1, 3))

'Constantes de la division en fracciones parciales de X2(s)
a(2, 1) = An1UnPCdosPRdif(b(2, 3), b(2, 2), b(2, 1), b(2, 0))
a(2, 2) = An2UnPCdosPRdif(b(2, 3), b(2, 2), b(2, 1), b(2, 0))
a(2, 3) = b(2, 3) - a(2, 1) - a(2, 2)
a(2, 4) = An4UnPCdosPRdif(b(2, 2), a(2, 1), a(2, 2), a(2, 3))

'Constantes de la division en fracciones parciales de X3(s)
a(3, 1) = An1UnPCdosPRdif(b(3, 3), b(3, 2), b(3, 1), b(3, 0))

```

```

a(3, 2) = An2UnPCdosPRdif(b(3, 3), b(3, 2), b(3, 1), b(3, 0))
a(3, 3) = b(3, 3) - a(3, 1) - a(3, 2)
a(3, 4) = An4UnPCdosPRdif(b(3, 2), a(3, 1), a(3, 2), a(3, 3))

```

```

'Constantes de la division en fracciones parciales de X4(s)
a(4, 1) = An1UnPCdosPRdif(b(4, 3), b(4, 2), b(4, 1), b(4, 0))
a(4, 2) = An2UnPCdosPRdif(b(4, 3), b(4, 2), b(4, 1), b(4, 0))
a(4, 3) = b(4, 3) - a(4, 1) - a(4, 2)
a(4, 4) = An4UnPCdosPRdif(b(4, 2), a(4, 1), a(4, 2), a(4, 3))

```

```
End Sub
```

```
'Realimentación de Estado
```

```
'Cálculo de la constante "A" de la division en Fracciones
```

```
'Parciales
```

```
Private Function An1UnPCdosPRdif(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An1UnPCdosPRdif = (Num3 * PoloD(3).r ^ 3 + Num2 * _
    PoloD(3).r ^ 2 + Num1 * PoloD(3).r + Num0) _
    / ((PoloD(3).r - PoloD(4).r) * ((PoloD(3).r - _
    PoloD(1).r) ^ 2 + PoloD(1).I ^ 2))

```

```
End Function
```

```
'Realimentación de Estado
```

```
'Cálculo de la constante "B" de la division en Fracciones
```

```
'Parciales
```

```
Private Function An2UnPCdosPRdif(Num3 As Single, Num2 As _
    Single, Num1 As Single, Num0 As Single) As Single
An2UnPCdosPRdif = (Num3 * PoloD(4).r ^ 3 + Num2 * _
    PoloD(4).r ^ 2 + Num1 * PoloD(4).r + Num0) _
    / ((PoloD(4).r - PoloD(3).r) * ((PoloD(4).r - _
    PoloD(1).r) ^ 2 + PoloD(1).I ^ 2))

```

```
End Function
```

```
'Realimentación de Estado
```

```
'Cálculo de la constante "D" de la division en Fracciones
```

```
'Parciales
```

```
Private Function An4UnPCdosPRdif(Num2 As Single, NumA As _
    Single, NumB As Single, NumC As Single) As Single
An4UnPCdosPRdif = (Num2 + NumA * (2 * PoloD(1).r + _
    PoloD(4).r) + NumB * (2 * PoloD(1).r + _
    PoloD(3).r) + NumC * (PoloD(3).r + PoloD(4).r _
    + PoloD(1).r)) / PoloD(1).I

```

```
End Function
```

```
'Realimentación de Estado
```

```
'Cálculo de la constante "A", "B", "C", "D" de la division en
```

```
'Fracciones Parciales de los cuatro estados
```

```
Private Sub FraccParcUNpcDOSprIG()
```

```
Dim At(1 To 4, 1 To 5) As Double
```

```
Dim Xt(1 To 4) As Double
```

Dim Orden As Integer

Orden = 4

'Constantes de la division en fracciones parciales de X1(s)

MatrizINIFraParUNpcDOSprIG At()

At(1, 5) = b(1, 3)

At(2, 5) = b(1, 2)

At(3, 5) = b(1, 1)

At(4, 5) = b(1, 0)

SistemaEcuaciones Orden, At(), Xt()

a(1, 1) = Xt(1)

a(1, 2) = Xt(2)

a(1, 3) = Xt(3)

a(1, 4) = Xt(4)

'Constantes de la division en fracciones parciales de X2(s)

MatrizINIFraParUNpcDOSprIG At()

At(1, 5) = b(2, 3)

At(2, 5) = b(2, 2)

At(3, 5) = b(2, 1)

At(4, 5) = b(2, 0)

AtPaso = At()

SistemaEcuaciones Orden, At(), Xt()

a(2, 1) = Xt(1)

a(2, 2) = Xt(2)

a(2, 3) = Xt(3)

a(2, 4) = Xt(4)

'Constantes de la division en fracciones parciales de X3(s)

MatrizINIFraParUNpcDOSprIG At()

At(1, 5) = b(3, 3)

At(2, 5) = b(3, 2)

At(3, 5) = b(3, 1)

At(4, 5) = b(3, 0)

SistemaEcuaciones Orden, At(), Xt()

a(3, 1) = Xt(1)

a(3, 2) = Xt(2)

a(3, 3) = Xt(3)

a(3, 4) = Xt(4)

'Constantes de la division en fracciones parciales de X4(s)

MatrizINIFraParUNpcDOSprIG At()

At(1, 5) = b(4, 3)

At(2, 5) = b(4, 2)

At(3, 5) = b(4, 1)

At(4, 5) = b(4, 0)

SistemaEcuaciones Orden, At(), Xt()

a(4, 1) = Xt(1)

a(4, 2) = Xt(2)

a(4, 3) = Xt(3)

$a(4, 4) = X_t(4)$

End Sub

'Realimentación de Estado

'Valores que deben ser seteados para el caso de division en
'fracciones parciales de dos polos complejos diferentes

Private Sub MatrizINIFraParDOSpcDIF(At() As Double)

At(1, 1) = 1

At(1, 2) = 0

At(1, 3) = 1

At(1, 4) = 0

At(2, 1) = -2 * PoloD(3).r - PoloD(1).r

At(2, 2) = PoloD(1).I

At(2, 3) = -2 * PoloD(1).r - PoloD(3).r

At(2, 4) = PoloD(3).I

At(3, 1) = PoloD(3).r ^ 2 + PoloD(3).I ^ 2 + 2 _
* PoloD(3).r * PoloD(1).r

At(3, 2) = -2 * PoloD(3).r * PoloD(1).I

At(3, 3) = PoloD(1).r ^ 2 + PoloD(1).I ^ 2 + 2 _
* PoloD(1).r * PoloD(3).r

At(3, 4) = -2 * PoloD(1).r * PoloD(3).I

At(4, 1) = -PoloD(3).r ^ 2 * PoloD(1).r - _
PoloD(3).I ^ 2 * PoloD(1).r

At(4, 2) = PoloD(1).I * PoloD(3).r ^ 2 + _
PoloD(3).I ^ 2 * PoloD(1).I

At(4, 3) = -PoloD(1).r ^ 2 * PoloD(3).r - _
PoloD(1).I ^ 2 * PoloD(3).r

At(4, 4) = PoloD(3).I * PoloD(1).r ^ 2 + _
PoloD(1).I ^ 2 * PoloD(3).I

End Sub

'Realimentación de Estado

'Valores que deben ser seteados para el caso de division en
'fracciones parciales de Un polo complejo y dos
'polos reales iguales

Private Sub MatrizINIFraParUNpcDOSprIG(At() As Double)

At(1, 1) = 1

At(1, 2) = 0

At(1, 3) = 1

At(1, 4) = 0

At(2, 1) = -2 * PoloD(1).r - PoloD(3).r

At(2, 2) = 1

At(2, 3) = -2 * PoloD(3).r - PoloD(1).r

At(2, 4) = PoloD(1).I

At(3, 1) = PoloD(1).r ^ 2 + PoloD(1).I ^ 2 + 2 _
* PoloD(1).r * PoloD(3).r

At(3, 2) = -2 * PoloD(1).r

```
At(3, 3) = PoloD(3).r ^ 2 + 2 * PoloD(3).r * PoloD(1).r
At(3, 4) = -2 * PoloD(3).r * PoloD(1).I
```

```
At(4, 1) = -PoloD(3).r * (PoloD(1).r ^ 2 + PoloD(1).I ^ 2)
At(4, 2) = PoloD(1).r ^ 2 + PoloD(1).I ^ 2
At(4, 3) = -PoloD(1).r * PoloD(3).r ^ 2
At(4, 4) = PoloD(1).I * PoloD(3).r ^ 2
```

```
End Sub
```

'Realimentación de estado

'Función que calcula el valor los estado X1,X2,X3,X4, para el caso
'de cuatro polos reales diferentes

```
Private Function FdeXCuatroPRdif(NumT As Single, NumA As Single, _
    NumB As Single, NumC As Single, NumD As Single) As Single
FdeXCuatroPRdif = NumA * Exp(PoloD(1).r * NumT) + _
    NumB * Exp(PoloD(2).r * NumT) + NumC * Exp(PoloD(3).r _
    * NumT) + NumD * Exp(PoloD(4).r * NumT)
```

```
End Function
```

'Realimentación de estado

'Función que calcula el valor los estado X1,X2,X3,X4, para el caso
'de dos polos reales iguales y dos polos reales diferentes

```
Private Function FdeXDosPRigdosPRdif(NumT As Single, NumA As Single, _
    NumB As Single, NumC As Single, NumD As Single) As Single
FdeXDosPRigdosPRdif = Exp(PoloD(1).r * NumT) * (NumA + _
    NumB * NumT) + NumC * Exp(PoloD(3).r * NumT) + _
    NumD * Exp(PoloD(4).r * NumT)
```

```
End Function
```

'Realimentación de estado

'Función que calcula el valor los estado X1,X2,X3,X4, para el caso
'de tres polos reales iguales y un polo real diferente

```
Private Function FdeXTresPRigunPRdif(NumT As Single, NumA As Single, _
    NumB As Single, NumC As Single, NumD As Single) As Single
FdeXTresPRigunPRdif = Exp(PoloD(1).r * NumT) * (NumA + _
    NumB * NumT + NumC / 2 * NumT ^ 2) + NumD * _
    Exp(PoloD(4).r * NumT)
```

```
End Function
```

'Realimentación de estado

'Función que calcula el valor los estado X1,X2,X3,X4, para el caso
'de cuatro polos reales iguales

```
Private Function FdeXCuatroPRig(NumT As Single, NumA As Single, _
    NumB As Single, NumC As Single, NumD As Single) As Single
FdeXCuatroPRig = Exp(PoloD(1).r * NumT) * (NumA + _
    NumB * NumT + NumC / 2 * NumT ^ 2 + NumD / 6 * NumT ^ 3)
```

```
End Function
```

```
'Realimentación de estado
'Función que calcula el valor los estado X1,X2,X3,X4, para el caso
'de cuatro polos repetidos de dos en dos
Private Function FdeXDosPRigdosPRig(NumT As Single, NumA As Single, _
    NumB As Single, NumC As Single, NumD As Single) As Single
FdeXDosPRigdosPRig = Exp(PoloD(1).r * NumT) * (NumA + _
    NumB * NumT) + Exp(PoloD(3).r * NumT) * (NumC + _
    NumD * NumT)
End Function
```

```
'Realimentación de estado
'Función que calcula el valor los estado X1,X2,X3,X4, para el caso
'de Dos polos complejos diferentes
Private Function FdeXDosPCdif(NumT As Single, NumA As Single, _
    NumB As Single, NumC As Single, NumD As Single) As Single
FdeXDosPCdif = Exp(PoloD(1).r * NumT) * (NumA * _
    Cos(PoloD(1).I * NumT) + NumB * Sin(PoloD(1).I * _
    NumT)) + Exp(PoloD(3).r * NumT) * (NumC * _
    Cos(PoloD(3).I * NumT) + NumD * Sin(PoloD(3).I * _
    NumT))
End Function
```

```
'Realimentación de estado
'Función que calcula el valor los estado X1,X2,X3,X4, para el caso
'de Dos polos complejos iguales'
Private Function FdeXDosPCig(NumT As Single, NumA As Single, _
    NumB As Single, NumC As Single, NumD As Single) As Single
FdeXDosPCig = Exp(PoloD(1).r * NumT) * (NumA * _
    Cos(PoloD(1).I * NumT) + NumB * Sin(PoloD(1).I * _
    NumT) + NumC * NumT / (2 * PoloD(1).I) * _
    Sin(PoloD(1).I * NumT) + NumD / (2 * PoloD(1).I ^ 3) _
    * (Sin(PoloD(1).I * NumT) - PoloD(1).I * NumT * _
    Cos(PoloD(1).I * NumT)))
End Function
```

```
'Realimentación de estado
'Función que calcula el valor los estado X1,X2,X3,X4, para el caso
'de un polo complejo conjugado y dos polos reales diferentes
Private Function FdeXUnPCigdosPRdif(NumT As Single, NumA As Single, _
    NumB As Single, NumC As Single, NumD As Single) As Single
FdeXUnPCigdosPRdif = NumA * Exp(PoloD(3).r * NumT) + _
    NumB * Exp(PoloD(4).r * NumT) + Exp(PoloD(1).r * _
    NumT) * (NumC * Cos(PoloD(1).I * NumT) + NumD * _
    Sin(PoloD(1).I * NumT))
End Function
```

```
'Realimentación de estado
'Función que calcula el valor los estado X1,X2,X3,X4, para el caso
'de un polo complejo conjugado y dos polos reales iguales
Private Function FdeXUnPCigdosPRig(NumT As Single, NumA As Single, _
    NumB As Single, NumC As Single, NumD As Single) As Single
FdeXUnPCigdosPRig = Exp(PoloD(3).r * NumT) * (NumA + NumB * _
    NumT) + Exp(PoloD(1).r * NumT) * (NumC * Cos(PoloD(1).I * _
    NumT) + NumD * Sin(PoloD(1).I * NumT))
End Function
```

```

Private Sub mnuSelecionControl_Click()
'-----
On Error Resume Next 'habilitación de manejo de error
Err.Clear
'-----
If (mnuNuevoPendulo.Tag = 1 Or mnuPenduloPorDefecto.Tag = 1) Then
    mnuSelecionControl.Tag = 1
    ComandosInvisibles
'Carga de valores el la forma tipos de control
With frmTiposDeControl
    'Cargo polos para el control por realimentación de estado
    .txtParteReal(1).Text = Format(PoloD(1).r, "0.00")
    .txtParteImag(1).Text = Format(PoloD(1).I, "0.00")
    .txtParteReal(2).Text = Format(PoloD(2).r, "0.00")
    .lblparteImag(2).Caption = Format(PoloD(2).I, "0.00")
    .txtParteReal(3).Text = Format(PoloD(3).r, "0.00")
    .txtParteImag(3).Text = Format(PoloD(3).I, "0.00")
    .txtParteReal(4).Text = Format(PoloD(4).r, "0.00")
    .lblparteImag(4).Caption = Format(PoloD(4).I, "0.00")

    'Cargo los valores de Mp, Ts y Cc para el control por Red adelante
    .txtBisectrizMp.Text = Format(RedbMp * 100, "0.00")
    .txtBisectrizTs.Text = Format(RedbTs, "0.00")
    .txtCeroMp.Text = Format(RedcMp * 100, "0.00")
    .txtCeroTs.Text = Format(RedcTs, "0.00")
    .txtCeroCc.Text = Format(RedcCc, "0.00")

    'Cargo los valores para las acciones de control
    .txtWn.Text = Format(Wn, "0.00")
    .txtSi.Text = Format(Si, "0.000")
    .txtpdkp.Text = Format(Ad_Kp, "0.000")
    .txtpdkd.Text = Format(Ad_Kd, "0.000")
    .txtPIDKp.Text = Format(A_Kp, "0.000")
    .txtPIDKd.Text = Format(A_Kd, "0.000")
    .txtPIDKi.Text = Format(A_Ki, "0.000")

    'Cargo valores del control fuzzy

    .txtRangoFuerza.Text = Format(Rango_u, "0.00") 'Rango Fuerza
    .txtRangoAngulo.Text = Format(Rango_f * RADaGRADOs, "0.00") 'Rango Angulo
    .txtRangoWang.Text = Format(Rango_w * RADaGRADOs, "0.00") 'Rango Velocidad Angular
    .txtTiempoSimulacion.Text = Format(Tsimulacion, "0.000") 'Tiempo de simulación
    .lblTiempoMuestreo.Caption = Tfuzzy 'Tiempo de muestreo

End With
'Muestro la ventana de tipos de control en modo modal
    frmTiposDeControl.Show 1

With frmTiposDeControl
    If frmTiposDeControl.txtChequeoBoton.Text = "OK" Then
        'Se presiono el boton "OK"

        'Valores ingresados por las acciones de control
    
```

'Valores de ingresados por la acción PD conociendo

'Wn y Si (E)

Wn = CSng(.txtWn.Text)

Si = CSng(.txtSi.Text)

'Valores ingresados por la acción PD conociendo

'Kp y Kd

Ad_Kp = CSng(.txtpdkp.Text)

Ad_Kd = CSng(.txtpdkd.Text)

'Valores ingresados por la acción PID

A_Kp = CSng(.txtPIDKp.Text)

A_Kd = CSng(.txtPIDKd.Text)

A_Ki = CSng(.txtPIDKi.Text)

'Valores ingresados por el control de realimentación de estado

PoloD(1).r = CSng(.txtParteReal(1).Text)

PoloD(1).I = Abs(CSng(.txtParteImag(1).Text))

PoloD(2).r = CSng(.txtParteReal(2).Text)

PoloD(2).I = -Abs(CSng(.txtParteImag(1).Text))

PoloD(3).r = CSng(.txtParteReal(3).Text)

PoloD(3).I = Abs(CSng(.txtParteImag(3).Text))

PoloD(4).r = CSng(.txtParteReal(4).Text)

PoloD(4).I = -Abs(CSng(.txtParteImag(3).Text))

'Valores ingresados por el control de red de adelanto de fase

RedbMp = CSng(.txtBisectrizMp.Text) / 100

RedbTs = CSng(.txtBisectrizTs.Text)

RedcMp = CSng(.txtCeroMp.Text) / 100

RedcTs = CSng(.txtCeroTs.Text)

RedcCc = CSng(.txtCeroCc.Text)

'Valores de ingresados por el control fuzzy

Rango_u = CSng(.txtRangoFuerza.Text) 'Rango de 40 Newton

Rango_f = CSng(.txtRangoAngulo.Text) * GRADOSA RAD 'Rango del ángulo

Rango_w = CSng(.txtRangoWang.Text) * GRADOSA RAD 'Rango de la

'velocidad angular

TsimulaciOn = CSng(.txtTiempoSimulacion.Text)

Tfuzzy = TsimulaciOn / NUMmAXpUNTOS

'Lectura de la base de conocimiento

BaseConoc(1, 1) = LectBaseConoc(.lblConocimiento(0).Caption)

BaseConoc(1, 2) = LectBaseConoc(.lblConocimiento(1).Caption)

BaseConoc(1, 3) = LectBaseConoc(.lblConocimiento(2).Caption)

BaseConoc(1, 4) = LectBaseConoc(.lblConocimiento(3).Caption)

BaseConoc(1, 5) = LectBaseConoc(.lblConocimiento(4).Caption)

BaseConoc(2, 1) = LectBaseConoc(.lblConocimiento(5).Caption)

BaseConoc(2, 2) = LectBaseConoc(.lblConocimiento(6).Caption)

BaseConoc(2, 3) = LectBaseConoc(.lblConocimiento(7).Caption)

BaseConoc(2, 4) = LectBaseConoc(.lblConocimiento(8).Caption)

BaseConoc(2, 5) = LectBaseConoc(.lblConocimiento(9).Caption)

```

BaseConoc(3, 1) = LectBaseConoc(.lblConocimiento(10).Caption)
BaseConoc(3, 2) = LectBaseConoc(.lblConocimiento(11).Caption)
BaseConoc(3, 3) = LectBaseConoc(.lblConocimiento(12).Caption)
BaseConoc(3, 4) = LectBaseConoc(.lblConocimiento(13).Caption)
BaseConoc(3, 5) = LectBaseConoc(.lblConocimiento(14).Caption)

```

```

BaseConoc(4, 1) = LectBaseConoc(.lblConocimiento(15).Caption)
BaseConoc(4, 2) = LectBaseConoc(.lblConocimiento(16).Caption)
BaseConoc(4, 3) = LectBaseConoc(.lblConocimiento(17).Caption)
BaseConoc(4, 4) = LectBaseConoc(.lblConocimiento(18).Caption)
BaseConoc(4, 5) = LectBaseConoc(.lblConocimiento(19).Caption)

```

```

BaseConoc(5, 1) = LectBaseConoc(.lblConocimiento(20).Caption)
BaseConoc(5, 2) = LectBaseConoc(.lblConocimiento(21).Caption)
BaseConoc(5, 3) = LectBaseConoc(.lblConocimiento(22).Caption)
BaseConoc(5, 4) = LectBaseConoc(.lblConocimiento(23).Caption)
BaseConoc(5, 5) = LectBaseConoc(.lblConocimiento(24).Caption)

```

```

'Se presiono el boton "CANCEL"
ElseIf frmTiposDeControl.txtChequeoBoton.Text = "CANCEL" Then

```

```

'Valores ingresados por el control proporcional conociendo

```

```

'Wn y Si
.txtWn.Text = CStr(Wn)
.txtSi.Text = CStr(Si)

```

```

'Valores ingresados por la acción PD conociendo

```

```

'Kp y Kd
.txtpdkp.Text = CStr(Ad_Kp)
.txtpdkd.Text = CStr(Ad_Kd)

```

```

'Valores ingresados por la acción PID

```

```

.txtPIDKp.Text = CStr(A_Kp)
.txtPIDKd.Text = CStr(A_Kd)
.txtPIDKi.Text = CStr(A_Ki)

```

```

'Valores ingresados por el control de realimentación de estado

```

```

.txtParteReal(1).Text = CStr(PoloD(1).r)
.txtParteImag(1).Text = CStr(PoloD(1).I)
.txtParteReal(2).Text = CStr(PoloD(2).r)
.lblparteImag(2).Caption = CStr(PoloD(2).I)
.txtParteReal(3).Text = CStr(PoloD(3).r)
.txtParteImag(3).Text = CStr(PoloD(3).I)
.txtParteReal(4).Text = CStr(PoloD(4).r)
.lblparteImag(4).Caption = CStr(PoloD(4).I)

```

```

'Valores ingresados por el control de red de adelanto de fase

```

```

If TipoRedAdelanto = REDaDELbISECTRIZ Then

```

```

.txtBisectrizMp.Text = CStr(RedbMp * 100)
.txtBisectrizTs.Text = CStr(RedbTs)

```

```

Else

```

```

.txtCeroMp.Text = CStr(RedcMp * 100)
.txtCeroTs.Text = CStr(RedcTs)
.txtCeroCc.Text = CStr(RedcCc)

```

```

End If

```

```

'Valores del control fuzzy
.txtRangoFuerza.Text = Format(Rango_u, "0.00")          'Rango Fuerza
.txtRangoAngulo.Text = Format(Rango_f * RADaGRADOs, "0.00") 'Rango Angulo
.txtRangoWang.Text = Format(Rango_w * RADaGRADOs, "0.00") 'Rango Velocidad Angular
.txtTiempoSimulacion.Text = Format(TsimulaciOn, "0.00") 'Tiempo de simulación
.lblTiempoMuestreo.Caption = Tfuzzy                    'Tiempo de muestreo

'Restauro la base de conocimiento
.lblConocimiento(0).Caption = RestauroBaseConoc(BaseConoc(1, 1))
.lblConocimiento(1).Caption = RestauroBaseConoc(BaseConoc(1, 2))
.lblConocimiento(2).Caption = RestauroBaseConoc(BaseConoc(1, 3))
.lblConocimiento(3).Caption = RestauroBaseConoc(BaseConoc(1, 4))
.lblConocimiento(4).Caption = RestauroBaseConoc(BaseConoc(1, 5))

.lblConocimiento(5).Caption = RestauroBaseConoc(BaseConoc(2, 1))
.lblConocimiento(6).Caption = RestauroBaseConoc(BaseConoc(2, 2))
.lblConocimiento(7).Caption = RestauroBaseConoc(BaseConoc(2, 3))
.lblConocimiento(8).Caption = RestauroBaseConoc(BaseConoc(2, 4))
.lblConocimiento(9).Caption = RestauroBaseConoc(BaseConoc(2, 5))

.lblConocimiento(10).Caption = RestauroBaseConoc(BaseConoc(3, 1))
.lblConocimiento(11).Caption = RestauroBaseConoc(BaseConoc(3, 2))
.lblConocimiento(12).Caption = RestauroBaseConoc(BaseConoc(3, 3))
.lblConocimiento(13).Caption = RestauroBaseConoc(BaseConoc(3, 4))
.lblConocimiento(14).Caption = RestauroBaseConoc(BaseConoc(3, 5))

.lblConocimiento(15).Caption = RestauroBaseConoc(BaseConoc(4, 1))
.lblConocimiento(16).Caption = RestauroBaseConoc(BaseConoc(4, 2))
.lblConocimiento(17).Caption = RestauroBaseConoc(BaseConoc(4, 3))
.lblConocimiento(18).Caption = RestauroBaseConoc(BaseConoc(4, 4))
.lblConocimiento(19).Caption = RestauroBaseConoc(BaseConoc(4, 5))

.lblConocimiento(20).Caption = RestauroBaseConoc(BaseConoc(5, 1))
.lblConocimiento(21).Caption = RestauroBaseConoc(BaseConoc(5, 2))
.lblConocimiento(22).Caption = RestauroBaseConoc(BaseConoc(5, 3))
.lblConocimiento(23).Caption = RestauroBaseConoc(BaseConoc(5, 4))
.lblConocimiento(24).Caption = RestauroBaseConoc(BaseConoc(5, 5))

Else
'No se presionó "nada"
End If
End With
Else

msg = "Primero se debe ingresar los datos del Péndulo en el menu" _
& " de Archivo "
Mostrar
End If
End Sub

```

```

Private Sub mnuUsoAyuda_Click()
'-----
On Error Resume Next ' Manejo de Error
Err.Clear
'-----

With cdgDialogo
.HelpCommand = cdlHelpHelpOnHelp
.ShowHelp
End With

'-----
If Err.Number <> 0 Then 'Chequeo de si existe error

    msg = "Error #" & Str(Err.Number) & " Se ha Generado un Error Tipo " _
        & Chr(13) & Err.Description & Chr(13) & "Hay Problemas con el Archivo de Ayuda"

    Estilo = vbOKOnly + vbCritical ' Define botones
    Titulo = "Error en la Ayuda" ' Define titulo
    MsgBox msg, Estilo, Titulo
End If
'-----
End Sub

```

```

Private Sub mnuVarios_Click()
'-----
On Error Resume Next ' habilitación de manejo de error
Err.Clear
'-----
If mnuCalculos.Tag = 1 Then
    Dim Tiempo As Single
    'Dim Ies As Integer
    TimerSim.Enabled = False
    cmdCongelar.Caption = "CONGELAR"
    frmGraficosVarios.Show 1
    If (frmGraficosVarios.ChkEspTiemp.Value = 1 Or _
        frmGraficosVarios.ChkvelTiemp.Value = 1 Or _
        frmGraficosVarios.ChkangTiemp.Value = 1 Or _
        frmGraficosVarios.ChkwangTiemp.Value = 1 Or _
        frmGraficosVarios.ChkfuerTiemp.Value = 1) Then

        picGraficos.Cls
        picGraficos.Picture = LoadPicture()
        picGraficos.DrawWidth = ESPESOReJES
        'Subrutina para dibujar ejes
        lblTitulo.Caption = "Gráficos Varios"
        UbicacionControles
        lblEscalaEjeX.Caption = Format(Tmax / DIV, "0.000") & " seg/div"
        lblEscalaEjeY.Caption = ""
        picGraficos.Scale (-Tmax / (3 * DIV), FdeX1max + FdeX1max / DIV)-
            (Tmax + Tmax / (3 * DIV), -FdeX1max - FdeX1max / DIV)
        DibujoEjes Tmax, FdeX1max, DIV

```

```

'Ha seleccionado que se grafique la curva de la posición del carrito
If frmGraficosVarios.ChkEspTiemp.Value = Checked Then

    picGraficos.Scale (-Tmax / (3 * DIV), FdeX1max + FdeX1max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeX1max - FdeX1max / DIV)
    picGraficos.CurrentX = 0
    picGraficos.CurrentY = Esp(0)
    Tiempo = 0
    picGraficos.DrawWidth = ESPESORgGRAFICOS
    For Ies = 1 To NUMmAXpUNTOS Step 1
        Tiempo = Tiempo + DeltaT
        picGraficos.Line -(Tiempo, Esp(Ies)), ROJO
    Next Ies
End If

'Ha seleccionado que se grafique la curva de la velocidad del carrito
If frmGraficosVarios.ChkvelTiemp.Value = Checked Then

    picGraficos.Scale (-Tmax / (3 * DIV), FdeX2max + FdeX2max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeX2max - FdeX2max / DIV)
    picGraficos.CurrentX = 0
    picGraficos.CurrentY = Vel(0)
    Tiempo = 0
    picGraficos.DrawWidth = ESPESORgGRAFICOS
    For Ies = 1 To NUMmAXpUNTOS Step 1
        Tiempo = Tiempo + DeltaT
        picGraficos.Line -(Tiempo, Vel(Ies)), AZUL
    Next Ies
End If

'Ha seleccionado que se grafique la curva del ángulo
If frmGraficosVarios.ChkangTiemp.Value = Checked Then

    picGraficos.Scale (-Tmax / (3 * DIV), FdeX3max + FdeX3max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeX3max - FdeX3max / DIV)
    picGraficos.CurrentX = 0
    picGraficos.CurrentY = Ang(0)
    Tiempo = 0
    picGraficos.DrawWidth = ESPESORgGRAFICOS
    For Ies = 1 To NUMmAXpUNTOS Step 1
        Tiempo = Tiempo + DeltaT
        picGraficos.Line -(Tiempo, Ang(Ies)), VERDE
    Next Ies
End If

'Ha seleccionado que se grafique la curva de la velocidad angular
If frmGraficosVarios.ChkwangTiemp.Value = Checked Then

    picGraficos.Scale (-Tmax / (3 * DIV), FdeX4max + FdeX4max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeX4max - FdeX4max / DIV)
    picGraficos.CurrentX = 0
    picGraficos.CurrentY = Wang(0)
    Tiempo = 0
    picGraficos.DrawWidth = ESPESORgGRAFICOS
    For Ies = 1 To NUMmAXpUNTOS Step 1

```

```

    Tiempo = Tiempo + DeltaT
    picGraficos.Line -(Tiempo, Wang(Ies)), CELESTE
Next Ies
End If

'Ha seleccionado que se grafique la curva de la fuerza
If frmGraficosVarios.ChkfuerTiemp.Value = Checked Then

    picGraficos.Scale (-Tmax / (3 * DIV), FdeU5max + FdeU5max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeU5max - FdeU5max / DIV)
    picGraficos.CurrentX = 0
    picGraficos.CurrentY = Fuerza(0)
    Tiempo = 0
    picGraficos.DrawWidth = ESPESORgGRAFICOS
    For Ies = 1 To NUMmAXpUNTOS Step 1
        Tiempo = Tiempo + DeltaT
        picGraficos.Line -(Tiempo, Fuerza(Ies)), LILA
    Next Ies
End If
End If

Else

    msg = "Debe ejecutar los 'Calculos' del menu de 'Simulación' primero"
    Mostrar
End If
End Sub

Private Sub mnuVelocidadAngVsAngulo_Click()
'-----
On Error Resume Next 'habilitación de manejo de error
Err.Clear
'-----
If mnuCalculos.Tag = 1 Then
'Dim Ies As Integer
lblTitulo.Caption = "Velocidad Angular Vs Angulo"
UbicacionControles
lblEscalaEjeX.Caption = Format(FdeX3max / DIV * RADaGRADOS, _
    "0.000") & " Grad/div"
lblEscalaEjeY.Caption = Format(FdeX4max / DIV * RADaGRADOS, _
    "0.000") & " (Grad/s)/div"
picGraficos.Scale (-FdeX3max - FdeX3max / DIV, FdeX4max _
    + FdeX4max / DIV)-(FdeX3max + FdeX3max / DIV, _
    -FdeX4max - FdeX4max / DIV)
picGraficos.Cls
picGraficos.Picture = LoadPicture()
picGraficos.DrawWidth = ESPESOReJES

'Subrutina para dibujar ejes
DibujoEjesWangVsAng FdeX3max, FdeX4max, DIV
picGraficos.CurrentX = Ang(0)
picGraficos.CurrentY = Wang(0)
picGraficos.DrawWidth = ESPESORgGRAFICOS

```

```

For Ies = 1 To NUMmAXpUNTOS Step 1
    picGraficos.Line -(Ang(Ies), Wang(Ies)), VERDE
Next Ies
Else
    msg = "Debe ejecutar los 'Calculos' del menu de 'Simulación' primero"
    Mostrar
End If
End Sub

```

```

Private Sub mnuVelocidadAngVsTiempo_Click()
'-----
On Error Resume Next ' habilitación de manejo de error
Err.Clear
'-----
If mnuCalculos.Tag = 1 Then
Dim Tiempo As Single
'Dim Ies As Integer
lblTitulo.Caption = "Velocidad Angular Vs Tiempo"
UbicacionControles
lblEscalaEjeX.Caption = Format(Tmax / 10, "0.000") & " seg/div"
lblEscalaEjeY.Caption = Format(FdeX4max / 10 * RADaGRADOs, _
    "0.000") & " (Grad/s)/div"
picGraficos.Scale (-Tmax / (3 * DIV), FdeX4max + FdeX4max / DIV) - _
    (Tmax + Tmax / (3 * DIV), -FdeX4max - FdeX4max / DIV)
picGraficos.Cls
picGraficos.Picture = LoadPicture()
picGraficos.DrawWidth = ESPESOReJES
'Dibujando Ejes
DibujoEjes Tmax, FdeX4max, DIV

picGraficos.CurrentX = 0
picGraficos.CurrentY = Wang(0)
Tiempo = 0
picGraficos.DrawWidth = ESPESORgRAFIcos
For Ies = 1 To NUMmAXpUNTOS Step 1
    Tiempo = Tiempo + DeltaT
    picGraficos.Line -(Tiempo, Wang(Ies)), CELESTE
Next Ies
Else
    msg = "Debe ejecutar los 'Calculos' del menu de 'Simulación' primero"
    Mostrar
End If
End Sub

```

```

Private Sub mnuVelocidadVsTiempo_Click()
'-----
On Error Resume Next ' habilitación de manejo de error
Err.Clear
'-----
If mnuCalculos.Tag = 1 Then

```

```

Dim Tiempo As Single
'Dim Ies As Integer
lblTitulo.Caption = "Velocidad Vs Tiempo"
UbicacionControles
lblEscalaEjeX.Caption = Format(Tmax / 10, "0.000") & " seg/div"
lblEscalaEjeY.Caption = Format(FdeX2max / 10, "0.00E+00") & " (m/s)/div"
picGraficos.Scale (-Tmax / (3 * DIV), FdeX2max + FdeX2max / DIV) - _
    (Tmax + Tmax / (3 * DIV), -FdeX2max - FdeX2max / DIV)
picGraficos.Cls
picGraficos.Picture = LoadPicture()
picGraficos.DrawWidth = ESPESOReJES
'Dibujando ejes
DibujoEjes Tmax, FdeX2max, DIV
picGraficos.CurrentX = 0
picGraficos.CurrentY = Vel(0)
Tiempo = 0
picGraficos.DrawWidth = ESPESORgGRAFICOS
For Ies = 1 To NUMmAXpUNTOS Step 1
    Tiempo = Tiempo + DeltaT
    picGraficos.Line -(Tiempo, Vel(Ies)), AZUL
Next Ies
Else

msg = "Debe ejecutar los 'Calculos' del menu de 'Simulación' primero"
Mostrar
End If
End Sub

Private Sub picGraficos_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)

If lblTitulo.Caption = "Gráficos Varios" Then

    If ROJO = picGraficos.Point(X, Y) Then 'Si selecciono la curva de
        'Espacio Vs tiempo
        lblCurva.Visible = True
        ShapeCurva.Visible = True
        ShapeCurva.BackColor = picGraficos.Point(X, Y)
        lblCurva.Caption = "Escala: Espacio Vs tiempo"
        lblCurva.Left = picGraficos.Left + picGraficos.Width - lblCurva.Width
        lblCurva.Top = picGraficos.Top - 2 * lblCurva.Height
        ShapeCurva.Left = lblCurva.Left - 1.5 * ShapeCurva.Width
        ShapeCurva.Top = lblCurva.Top
        lblEscalaEjeY.Caption = Format(FdeX1max / DIV, "0.00E+00") & " m/div"
        picGraficos.Scale (-Tmax / (3 * DIV), FdeX1max + FdeX1max / DIV) - _
            (Tmax + Tmax / (3 * DIV), -FdeX1max - FdeX1max / DIV)

    ElseIf AZUL = picGraficos.Point(X, Y) Then
        lblCurva.Visible = True
        ShapeCurva.Visible = True
        ShapeCurva.BackColor = picGraficos.Point(X, Y)
        lblCurva.Caption = "Escala: Velocidad Vs tiempo"
        lblCurva.Left = picGraficos.Left + picGraficos.Width - lblCurva.Width
        lblCurva.Top = picGraficos.Top - 2 * lblCurva.Height

```

```

ShapeCurva.Left = lblCurva.Left - 1.5 * ShapeCurva.Width
ShapeCurva.Top = lblCurva.Top
lblEscalaEjeY.Caption = Format(FdeX2max / 10, "0.00E+00") & " (m/s/div)"
picGraficos.Scale (-Tmax / (3 * DIV), FdeX2max + FdeX2max / DIV)- _
    (Tmax + Tmax / (3 * DIV), -FdeX2max - FdeX2max / DIV)

```

```

ElseIf VERDE = picGraficos.Point(X, Y) Then
    lblCurva.Visible = True
    ShapeCurva.Visible = True
    ShapeCurva.BackColor = picGraficos.Point(X, Y)
    lblCurva.Caption = "Escala: Angulo Vs tiempo"
    lblCurva.Left = picGraficos.Left + picGraficos.Width - lblCurva.Width
    lblCurva.Top = picGraficos.Top - 2 * lblCurva.Height
    ShapeCurva.Left = lblCurva.Left - 1.5 * ShapeCurva.Width
    ShapeCurva.Top = lblCurva.Top
    lblEscalaEjeY.Caption = Format(FdeX3max / 10 * RADaGRADOS, _
        "0.000") & " Grad/div"
    picGraficos.Scale (-Tmax / (3 * DIV), FdeX3max + FdeX3max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeX3max - FdeX3max / DIV)

```

```

ElseIf CELESTE = picGraficos.Point(X, Y) Then
    lblCurva.Visible = True
    ShapeCurva.Visible = True
    ShapeCurva.BackColor = picGraficos.Point(X, Y)
    lblCurva.Caption = "Escala: Velocidad Angulo Vs tiempo"
    lblCurva.Left = picGraficos.Left + picGraficos.Width - lblCurva.Width
    lblCurva.Top = picGraficos.Top - 2 * lblCurva.Height
    ShapeCurva.Left = lblCurva.Left - 1.5 * ShapeCurva.Width
    ShapeCurva.Top = lblCurva.Top
    lblEscalaEjeY.Caption = Format(FdeX4max / 10 * RADaGRADOS, _
        "0.000") & " (Grad/s)/div"
    picGraficos.Scale (-Tmax / (3 * DIV), FdeX4max + FdeX4max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeX4max - FdeX4max / DIV)

```

```

ElseIf LILA = picGraficos.Point(X, Y) Then
    lblCurva.Visible = True
    ShapeCurva.Visible = True
    ShapeCurva.BackColor = picGraficos.Point(X, Y)
    lblCurva.Caption = "Escala: Fuerza Vs tiempo"
    lblCurva.Left = picGraficos.Left + picGraficos.Width - lblCurva.Width
    lblCurva.Top = picGraficos.Top - 2 * lblCurva.Height
    ShapeCurva.Left = lblCurva.Left - 1.5 * ShapeCurva.Width
    ShapeCurva.Top = lblCurva.Top
    lblEscalaEjeY.Caption = Format(FdeU5max / 10, "0.000") _
        & " Newt/div"
    picGraficos.Scale (-Tmax / (3 * DIV), FdeU5max + FdeU5max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeU5max - FdeU5max / DIV)

```

```

End If
End If
End Sub

```

```

Private Sub picGraficos_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

```

```

If (lblTitulo.Caption = "Angulo Vs Tiempo") Or _
  (lblTitulo.Caption = "Velocidad Angular Vs Tiempo") Or _
  lblCurva.Caption = "Escala: Angulo Vs tiempo" Or _
  lblCurva.Caption = "Escala: Velocidad Angulo Vs tiempo" Then
  lblEjeYtext.Caption = Format(Y * RADaGRADOs, "0.000")
  lblEjeXtext.Caption = Format(CSng(X), "0.000")
ElseIf lblTitulo.Caption = "Velocidad Angular Vs Angulo" Then
  lblEjeYtext.Caption = Format(Y * RADaGRADOs, "0.000")
  lblEjeXtext.Caption = Format(X * RADaGRADOs, "0.000")
ElseIf lblTitulo.Caption = "Gráficos Varios" And _
  lblCurva.Visible = False Then
  lblEjeYtext.Caption = ""
  lblEjeXtext.Caption = Format(X, "0.000")
Else
  lblEjeXtext.Caption = Format(X, "0.000")
  lblEjeYtext.Caption = Format(Y, "0.000")
End If
End Sub

```

```

'Subrutina que dibuja los ejes coordenados para los gráficos
'Espacio Vs tiempo, Velocidad Vs tiempo, Angulo Vs tiempo
'Velocidad Angular Vs tiempo y Fuerza Vs tiempo
Private Sub DibujoEjes(NumXmax As Single, NumYmax _
  As Single, NumDiv As Single)
  Dim Xset As Single, Yset As Single
  Dim LineaX As Single, LineaY As Single
  Dim I As Integer
  'UbicacionControles

```

```

ComandosVisibles
picGraficos.Line (0, 0)-(NumXmax, 0)
picGraficos.Line (0, NumYmax)-(0, -NumYmax)
Xset = NumXmax / NumDiv
Yset = NumYmax / NumDiv
LineaX = NumYmax / (4 * NumDiv)
LineaY = NumXmax / (8 * NumDiv)
For I = 1 To NumDiv
  picGraficos.Line (Xset * I, LineaX)-(Xset * I, -LineaX)
Next I
For I = -NumDiv To NumDiv Step 1
  picGraficos.Line (-LineaY, Yset * I)-(LineaY, Yset * I)
Next I
End Sub

```

```

'Subrutina que dibuja los ejes coordenados para el gráfico
'Velocidad angular Vs Angulo.
Private Sub DibujoEjesWangVsAng(NumXmax As Single, NumYmax _
  As Single, NumDiv As Single)
  Dim Xset As Single, Yset As Single
  Dim LineaX As Single, LineaY As Single
  Dim I As Integer

```

```

'UbicacionControles
ComandosVisibles
picGraficos.Line (-NumXmax, 0)-(NumXmax, 0) 'Eje X
picGraficos.Line (0, NumYmax)-(0, -NumYmax) 'Eje Y
Xset = NumXmax / NumDiv
Yset = NumYmax / NumDiv
LineaX = NumYmax / 40 'Longitud del segmento en Eje X
LineaY = NumXmax / 80 'Longitud del segmento en Eje Y
'Divisiones para el eje X
For I = -NumDiv To NumDiv Step 1
picGraficos.Line (Xset * I, LineaX)-(Xset * I, -LineaX)
Next I
'Divisiones para el eje Y
For I = -NumDiv To NumDiv Step 1
picGraficos.Line (-LineaY, Yset * I)-(LineaY, Yset * I)
Next I
End Sub

```

'Realimentación de estado
'Subrutina que calcula los estados en función del tiempo y
'los valores son almacenados en una matriz para su posterior
'utilización en la realización de las gráficas para el caso
'correspondiente

```

Private Sub XtCUATROprDIF()
Dim Tiempo As Single
'Dim Ies As Integer
FdeX1max = 0
FdeX2max = 0
FdeX3max = 0
FdeX4max = 0
FdeU5max = 0
Tiempo = 0
For Ies = 0 To NUMmAXpUNTOS Step 1
Esp(Ies) = FdeXCuatroPRdif(Tiempo, a(1, 1), _
a(1, 2), a(1, 3), a(1, 4))

If FdeX1max < Abs(Esp(Ies)) Then
FdeX1max = Abs(Esp(Ies))
End If

Vel(Ies) = FdeXCuatroPRdif(Tiempo, a(2, 1), _
a(2, 2), a(2, 3), a(2, 4))
If FdeX2max < Abs(Vel(Ies)) Then
FdeX2max = Abs(Vel(Ies))
End If

Ang(Ies) = FdeXCuatroPRdif(Tiempo, a(3, 1), _
a(3, 2), a(3, 3), a(3, 4))
If FdeX3max < Abs(Ang(Ies)) Then
FdeX3max = Abs(Ang(Ies))
End If

Wang(Ies) = FdeXCuatroPRdif(Tiempo, a(4, 1), _
a(4, 2), a(4, 3), a(4, 4))

```

```

    . If FdeX4max < Abs(Wang(Ies)) Then
        FdeX4max = Abs(Wang(Ies))
    End If

    Fuerza(Ies) = -(Kest(1) * Esp(Ies) + Kest(2) *
        * Vel(Ies) + Kest(3) * Ang(Ies) + Kest(4) *
        * Wang(Ies))
    If FdeU5max < Abs(Fuerza(Ies)) Then
        FdeU5max = Abs(Fuerza(Ies))
    End If

    Tiempo = Tiempo + DeltaT

    If (Ies Mod NumBarra) = 0 Then
        BarraProceso Ies
    End If

Next Ies
End Sub

'Realimentación de estado
'Subrutiña que calcula los estados en función del tiempo y
'los valores son almacenados en una matriz para su posterior
'utilización en la realización de las gráficas para el caso
'correspondiente
Private Sub XtDOSprIGDOSprDIF()
    Dim Tiempo As Single
    'Dim Ies As Integer
    FdeX1max = 0
    FdeX2max = 0
    FdeX3max = 0
    FdeX4max = 0
    FdeU5max = 0
    Tiempo = 0
    For Ies = 0 To NUMmAXpUNTOS Step 1
        Esp(Ies) = FdeXDosPRigdosPRdif(Tiempo, a(1, 1), _
            a(1, 2), a(1, 3), a(1, 4))
        If FdeX1max < Abs(Esp(Ies)) Then
            FdeX1max = Abs(Esp(Ies))
        End If

        Vel(Ies) = FdeXDosPRigdosPRdif(Tiempo, a(2, 1), _
            a(2, 2), a(2, 3), a(2, 4))
        If FdeX2max < Abs(Vel(Ies)) Then
            FdeX2max = Abs(Vel(Ies))
        End If

        Ang(Ies) = FdeXDosPRigdosPRdif(Tiempo, a(3, 1), _
            a(3, 2), a(3, 3), a(3, 4))
        If FdeX3max < Abs(Ang(Ies)) Then
            FdeX3max = Abs(Ang(Ies))
        End If

        Wang(Ies) = FdeXDosPRigdosPRdif(Tiempo, a(4, 1), _

```

```

        a(4, 2), a(4, 3), a(4, 4))
    If FdeX4max < Abs(Wang(Ies)) Then
        FdeX4max = Abs(Wang(Ies))
    End If

    Fuerza(Ies) = -(Kest(1) * Esp(Ies) + Kest(2) *
        * Vel(Ies) + Kest(3) * Ang(Ies) + Kest(4) *
        * Wang(Ies))
    If FdeU5max < Abs(Fuerza(Ies)) Then
        FdeU5max = Abs(Fuerza(Ies))
    End If

    Tiempo = Tiempo + DeltaT

    If (Ies Mod NumBarra) = 0 Then
        BarraProceso Ies
    End If
Next Ies

End Sub

'Realimentación de estado
'Subrutina que calcula los estados en función del tiempo y
'los valores son almacenados en una matriz para su posterior
'utilización en la realización de las gráficas para el caso
'correspondiente
Private Sub XtTRESprIGUNprDIF()
    Dim Tiempo As Single
    'Dim Ies As Integer
    FdeX1max = 0
    FdeX2max = 0
    FdeX3max = 0
    FdeX4max = 0
    FdeU5max = 0
    Tiempo = 0
    For Ies = 0 To NUMmAXpUNTOS Step 1
        Esp(Ies) = FdeXTresPRigunPRdif(Tiempo, a(1, 1), _
            a(1, 2), a(1, 3), a(1, 4))
        If FdeX1max < Abs(Esp(Ies)) Then
            FdeX1max = Abs(Esp(Ies))
        End If

        Vel(Ies) = FdeXTresPRigunPRdif(Tiempo, a(2, 1), _
            a(2, 2), a(2, 3), a(2, 4))
        If FdeX2max < Abs(Vel(Ies)) Then
            FdeX2max = Abs(Vel(Ies))
        End If
        Ang(Ies) = FdeXTresPRigunPRdif(Tiempo, a(3, 1), _
            a(3, 2), a(3, 3), a(3, 4))
        If FdeX3max < Abs(Ang(Ies)) Then
            FdeX3max = Abs(Ang(Ies))
        End If
    
```

```

Wang(Ies) = FdeXTresPRigunPRdif(Tiempo, a(4, 1), _
    a(4, 2), a(4, 3), a(4, 4))
If FdeX4max < Abs(Wang(Ies)) Then
    FdeX4max = Abs(Wang(Ies))
End If

Fuerza(Ies) = -(Kest(1) * Esp(Ies) + Kest(2) _
    * Vel(Ies) + Kest(3) * Ang(Ies) + Kest(4) _
    * Wang(Ies))
If FdeU5max < Abs(Fuerza(Ies)) Then
    FdeU5max = Abs(Fuerza(Ies))
End If

Tiempo = Tiempo + DeltaT

If (Ies Mod NumBarra) = 0 Then
    BarraProceso Ies
End If
Next Ies
End Sub

```

'Realimentación de estado
'Subrutina que calcula los estados en función del tiempo y
'los valores son almacenados en una matriz para su posterior
'utilización en la realización de las gráficas para el caso
'correspondiente

```

Private Sub XtCUATROprIG()
Dim Tiempo As Single
'Dim Ies As Integer
FdeX1max = 0
FdeX2max = 0
FdeX3max = 0
FdeX4max = 0
FdeU5max = 0
Tiempo = 0
For Ies = 0 To NUMmAXpUNTOS Step 1
    Esp(Ies) = FdeXCuatroPRig(Tiempo, a(1, 1), _
        a(1, 2), a(1, 3), a(1, 4))
    If FdeX1max < Abs(Esp(Ies)) Then
        FdeX1max = Abs(Esp(Ies))
    End If

    Vel(Ies) = FdeXCuatroPRig(Tiempo, a(2, 1), _
        a(2, 2), a(2, 3), a(2, 4))
    If FdeX2max < Abs(Vel(Ies)) Then
        FdeX2max = Abs(Vel(Ies))
    End If

    Ang(Ies) = FdeXCuatroPRig(Tiempo, a(3, 1), _
        a(3, 2), a(3, 3), a(3, 4))
    If FdeX3max < Abs(Ang(Ies)) Then
        FdeX3max = Abs(Ang(Ies))
    End If

```

```

Wang(Ies) = FdeXCuatroPRig(Tiempo, a(4, 1), _
    a(4, 2), a(4, 3), a(4, 4))
If FdeX4max < Abs(Wang(Ies)) Then
    FdeX4max = Abs(Wang(Ies))
End If

Fuerza(Ies) = -(Kest(1) * Esp(Ies) + Kest(2) _
    * Vel(Ies) + Kest(3) * Ang(Ies) + Kest(4) _
    * Wang(Ies))
If FdeU5max < Abs(Fuerza(Ies)) Then
    FdeU5max = Abs(Fuerza(Ies))
End If

Tiempo = Tiempo + DeltaT

If (Ies Mod NumBarra) = 0 Then
    BarraProceso Ies
End If
Next Ies
End Sub

```

'Realimentación de estado
'Subrutina que calcula los estados en función del tiempo y
'los valores son almacenados en una matriz para su posterior
'utilización en la realización de las gráficas para el caso
'correspondiente

```

Private Sub XtDOSprIGDOSprIG()
Dim Tiempo As Single
'Dim Ies As Integer
FdeX1max = 0
FdeX2max = 0
FdeX3max = 0
FdeX4max = 0
FdeU5max = 0
Tiempo = 0
For Ies = 0 To NUMmAXpUNTOS Step 1
    Esp(Ies) = FdeXDosPRigdosPRig(Tiempo, a(1, 1), _
        a(1, 2), a(1, 3), a(1, 4))
    If FdeX1max < Abs(Esp(Ies)) Then
        FdeX1max = Abs(Esp(Ies))
    End If

    Vel(Ies) = FdeXDosPRigdosPRig(Tiempo, a(2, 1), _
        a(2, 2), a(2, 3), a(2, 4))
    If FdeX2max < Abs(Vel(Ies)) Then
        FdeX2max = Abs(Vel(Ies))
    End If

    Ang(Ies) = FdeXDosPRigdosPRig(Tiempo, a(3, 1), _
        a(3, 2), a(3, 3), a(3, 4))
    If FdeX3max < Abs(Ang(Ies)) Then
        FdeX3max = Abs(Ang(Ies))
    End If

```

```

Wang(Ies) = FdeXDosPRigdosPRig(Tiempo, a(4, 1), _
    a(4, 2), a(4, 3), a(4, 4))
If FdeX4max < Abs(Wang(Ies)) Then
    FdeX4max = Abs(Wang(Ies))
End If

```

```

Fuerza(Ies) = -(Kest(1) * Esp(Ies) + Kest(2) _
    * Vel(Ies) + Kest(3) * Ang(Ies) + Kest(4) _
    * Wang(Ies))
If FdeU5max < Abs(Fuerza(Ies)) Then
    FdeU5max = Abs(Fuerza(Ies))
End If

```

```

Tiempo = Tiempo + DeltaT

```

```

If (Ies Mod NumBarra) = 0 Then
    BarraProceso Ies
End If

```

```

Next Ies
End Sub

```

'Realimentación de estado
'Subrutina que calcula los estados en función del tiempo y
'los valores son almacenados en una matriz para su posterior
'utilización en la realización de las gráficas para el caso
'correspondiente

```

Private Sub XtDOSpcDIF()

```

```

Dim Tiempo As Single

```

```

'Dim Ies As Integer

```

```

FdeX1max = 0

```

```

FdeX2max = 0

```

```

FdeX3max = 0

```

```

FdeX4max = 0

```

```

FdeU5max = 0

```

```

Tiempo = 0

```

```

For Ies = 0 To NUMmAXpUNTOS Step 1

```

```

    Esp(Ies) = FdeXDosPCdif(Tiempo, a(1, 1), _
        a(1, 2), a(1, 3), a(1, 4))

```

```

    If FdeX1max < Abs(Esp(Ies)) Then

```

```

        FdeX1max = Abs(Esp(Ies))

```

```

    End If

```

```

    Vel(Ies) = FdeXDosPCdif(Tiempo, a(2, 1), _
        a(2, 2), a(2, 3), a(2, 4))

```

```

    If FdeX2max < Abs(Vel(Ies)) Then

```

```

        FdeX2max = Abs(Vel(Ies))

```

```

    End If

```

```

    Ang(Ies) = FdeXDosPCdif(Tiempo, a(3, 1), _
        a(3, 2), a(3, 3), a(3, 4))

```

```

    If FdeX3max < Abs(Ang(Ies)) Then

```

```

        FdeX3max = Abs(Ang(Ies))

```

```

    End If

```

```

Wang(Ies) = FdeXDosPCdif(Tiempo, a(4, 1), _
    a(4, 2), a(4, 3), a(4, 4))
If FdeX4max < Abs(Wang(Ies)) Then
    FdeX4max = Abs(Wang(Ies))
End If

Fuerza(Ies) = -(Kest(1) * Esp(Ies) + Kest(2) _
    * Vel(Ies) + Kest(3) * Ang(Ies) + Kest(4) _
    * Wang(Ies))
If FdeU5max < Abs(Fuerza(Ies)) Then
    FdeU5max = Abs(Fuerza(Ies))
End If

Tiempo = Tiempo + DeltaT

If (Ies Mod NumBarra) = 0 Then
    BarraProceso Ies
End If
Next Ies
End Sub

```

'Realimentación de estado
'Subrutina que calcula los estados en función del tiempo y
'los valores son almacenados en una matriz para su posterior
'utilización en la realización de las gráficas para el caso
'correspondiente

```

Private Sub XtDOSpcIG()
Dim Tiempo As Single
'Dim Ies As Integer
FdeX1max = 0
FdeX2max = 0
FdeX3max = 0
FdeX4max = 0
FdeU5max = 0
Tiempo = 0
For Ies = 0 To NUMmAXpUNTOS Step 1
    Esp(Ies) = FdeXDosPCig(Tiempo, a(1, 1), _
        a(1, 2), a(1, 3), a(1, 4))
    If FdeX1max < Abs(Esp(Ies)) Then
        FdeX1max = Abs(Esp(Ies))
    End If

    Vel(Ies) = FdeXDosPCig(Tiempo, a(2, 1), _
        a(2, 2), a(2, 3), a(2, 4))
    If FdeX2max < Abs(Vel(Ies)) Then
        FdeX2max = Abs(Vel(Ies))
    End If

    Ang(Ies) = FdeXDosPCig(Tiempo, a(3, 1), _
        a(3, 2), a(3, 3), a(3, 4))
    If FdeX3max < Abs(Ang(Ies)) Then
        FdeX3max = Abs(Ang(Ies))
    End If

```

```

Wang(Ies) = FdeXDosPCig(Tiempo, a(4, 1), _
    a(4, 2), a(4, 3), a(4, 4))
If FdeX4max < Abs(Wang(Ies)) Then
    FdeX4max = Abs(Wang(Ies))
End If

```

```

Fuerza(Ies) = -(Kest(1) * Esp(Ies) + Kest(2) _
    * Vel(Ies) + Kest(3) * Ang(Ies) + Kest(4) _
    * Wang(Ies))
If FdeU5max < Abs(Fuerza(Ies)) Then
    FdeU5max = Abs(Fuerza(Ies))
End If

```

```

Tiempo = Tiempo + DeltaT

```

```

If (Ies Mod NumBarra) = 0 Then
    BarraProceso Ies
End If
Next Ies
End Sub

```

'Realimentación de estado
'Subrutina que calcula los estados en función del tiempo y
'los valores son almacenados en una matriz para su posterior
'utilización en la realización de las gráficas para el caso
'correspondiente

```

Private Sub XtUNpcDOSPrDIF()

```

```

Dim Tiempo As Single

```

```

'Dim Ies As Integer

```

```

FdeX1max = 0

```

```

FdeX2max = 0

```

```

FdeX3max = 0

```

```

FdeX4max = 0

```

```

FdeU5max = 0

```

```

Tiempo = 0

```

```

For Ies = 0 To NUMmAXpUNTOS Step 1

```

```

    Esp(Ies) = FdeXUnPCigdosPRdif(Tiempo, a(1, 1), _
        a(1, 2), a(1, 3), a(1, 4))

```

```

    If FdeX1max < Abs(Esp(Ies)) Then

```

```

        FdeX1max = Abs(Esp(Ies))

```

```

    End If

```

```

    Vel(Ies) = FdeXUnPCigdosPRdif(Tiempo, a(2, 1), _
        a(2, 2), a(2, 3), a(2, 4))

```

```

    If FdeX2max < Abs(Vel(Ies)) Then

```

```

        FdeX2max = Abs(Vel(Ies))

```

```

    End If

```

```

    Ang(Ies) = FdeXUnPCigdosPRdif(Tiempo, a(3, 1), _
        a(3, 2), a(3, 3), a(3, 4))

```

```

    If FdeX3max < Abs(Ang(Ies)) Then

```

```

        FdeX3max = Abs(Ang(Ies))

```

```

End If

Wang(Ies) = FdeXUnPCigdosPRdif(Tiempo, a(4, 1), _
    a(4, 2), a(4, 3), a(4, 4))
If FdeX4max < Abs(Wang(Ies)) Then
    FdeX4max = Abs(Wang(Ies))
End If

Fuerza(Ies) = -(Kest(1) * Esp(Ies) + Kest(2) _
    * Vel(Ies) + Kest(3) * Ang(Ies) + Kest(4) _
    * Wang(Ies))
If FdeU5max < Abs(Fuerza(Ies)) Then
    FdeU5max = Abs(Fuerza(Ies))
End If

Tiempo = Tiempo + DeltaT

If (Ies Mod NumBarra) = 0 Then
    BarraProceso Ies
End If
Next Ies
End Sub

```

'Realimentación de estado
'Subrutina que calcula los estados en función del tiempo y
'los valores son almacenados en una matriz para su posterior
'utilización en la realización de las gráficas para el caso
'correspondiente

```

Private Sub XtUNpcDOSprIG()
Dim Tiempo As Single
'Dim Ies As Integer
FdeX1max = 0
FdeX2max = 0
FdeX3max = 0
FdeX4max = 0
FdeU5max = 0
Tiempo = 0
For Ies = 0 To NUMmAXpUNTOS Step 1
    Esp(Ies) = FdeXUnPCigdosPRig(Tiempo, a(1, 1), _
        a(1, 2), a(1, 3), a(1, 4))
    If FdeX1max < Abs(Esp(Ies)) Then
        FdeX1max = Abs(Esp(Ies))
    End If

    Vel(Ies) = FdeXUnPCigdosPRig(Tiempo, a(2, 1), _
        a(2, 2), a(2, 3), a(2, 4))
    If FdeX2max < Abs(Vel(Ies)) Then
        FdeX2max = Abs(Vel(Ies))
    End If

    Ang(Ies) = FdeXUnPCigdosPRig(Tiempo, a(3, 1), _
        a(3, 2), a(3, 3), a(3, 4))
    If FdeX3max < Abs(Ang(Ies)) Then
        FdeX3max = Abs(Ang(Ies))
    End If

```

```

End If

Wang(Ies) = FdeXUnPCigdosPRig(Tiempo, a(4, 1), _
    a(4, 2), a(4, 3), a(4, 4))
If FdeX4max < Abs(Wang(Ies)) Then
    FdeX4max = Abs(Wang(Ies))
End If

```

```

Fuerza(Ies) = -(Kest(1) * Esp(Ies) + Kest(2) _
    * Vel(Ies) + Kest(3) * Ang(Ies) + Kest(4) _
    * Wang(Ies))
If FdeU5max < Abs(Fuerza(Ies)) Then
    FdeU5max = Abs(Fuerza(Ies))
End If

```

```
Tiempo = Tiempo + DeltaT
```

```

If (Ies Mod NumBarra) = 0 Then
    BarraProceso Ies

```

```
End If
```

```
Next Ies
```

```
End Sub
```

'Subrutina que hace visibles los componentes de la forma
'frmAlternativa para los gráficos de variables de entrada salida Vs tiempo

```
Private Sub ComandosVisibles()
```

```

picGraficos.Visible = True
lblEjeXtext.Visible = True
lblEjeYtext.Visible = True
LblEjeX.Visible = True
lblEjeY.Visible = True
lblEscalaEjeX.Visible = True
lblEscalaEjeY.Visible = True
lblTitulo.Visible = True
lblTipoControl.Visible = True

```

'Objetos visibles solo en la simulación

```
LinePendulo.Visible = False
```

```
ImgCarro.Visible = False
```

```
FrameValores.Visible = False 'Cuadro que contiene las variables de ángulo
    'Velocidad angular y fuerza
```

```
FrameValores_L.Visible = False 'Cuadro que contiene las variables de posición
    'Velocidad y tiempo
```

```
FrameGraficos.Visible = False 'Gráfico que contiene las curvas de las
    'variables de entrada y salida para simulación
```

```
FrameVelocidad.Visible = False 'Cuadro del Scroll de velocidad de simulación
```

```
cmdCongelar.Visible = False 'Boton que congela o descongela la imagen
    'de simulación
```

```
lblCurva.Visible = False 'Cuadros indicadores del tipo de curva
```

```
ShapeCurva.Visible = False 'Se hacen invisibles
End Sub
```

```
'Subrutina que hace invisibles los componentes de la forma
```

```
'frmAlternativa
```

```
Private Sub ComandosInvisibles()
```

```
TimerSim.Enabled = False
```

```
cmdCongelar.Caption = "CONGELAR"
```

```
picGraficos.Picture = LoadPicture()
```

```
picGraficos.Visible = False
```

```
lblEjeXtext.Visible = False
```

```
lblEjeYtext.Visible = False
```

```
LblEjeX.Visible = False
```

```
lblEjeY.Visible = False
```

```
lblEscalaEjeX.Visible = False
```

```
lblEscalaEjeY.Visible = False
```

```
lblTitulo.Visible = False
```

```
lblTipoControl.Visible = False
```

```
LinePendulo.Visible = False
```

```
ImgCarro.Visible = False
```

```
FrameValores.Visible = False 'Cuadro que contiene las variables de ángulo
'Velocidad angular y fuerza
```

```
FrameValores_L.Visible = False 'Cuadro que contiene las variables de posición
'Velocidad y tiempo
```

```
FrameGraficos.Visible = False 'Gráfico que contiene las curvas de las
'variables de entrada y salida para simulación
```

```
FrameVelocidad.Visible = False 'Cuadro del scroll de la velocidad de simulación
```

```
cmdCongelar.Visible = False 'Boton que congela o descongela la imagen
'de simulación
```

```
'Para gráficos varios
```

```
lblCurva.Visible = False
```

```
ShapeCurva.Visible = False
```

```
End Sub
```

```
'Subrutina que ubica los componentes de la forma "frmAlternativa"
```

```
'Cuando se cambia el tamaño de la forma
```

```
Private Sub UbicacionControles()
```

```
Dim VarAuxEscala As Integer
```

```
Dim VarAuxEscalafrm As Integer
```

```
VarAuxEscala = picGraficos.ScaleMode
```

```
picGraficos.ScaleMode = 1
```

```
VarAuxEscalafrm = frmAlternativa.ScaleMode
```

```
frmAlternativa.ScaleMode = 1
```

```
'Ubicación de formas
```

```
frmDatosPendulo.Left = frmAlternativa.Width / 2 - frmDatosPendulo.Width / 2
```

```
frmDatosPendulo.Top = frmAlternativa.Height / 2 - frmDatosPendulo.Height / 2
```

```
frmTiposDeControl.Left = frmAlternativa.Width / 2 - frmTiposDeControl.Width / 2
frmTiposDeControl.Top = frmAlternativa.Height / 2 - frmTiposDeControl.Height / 2
```

```
frmResultados.Left = frmAlternativa.Width / 2 - frmResultados.Width / 2
frmResultados.Top = frmAlternativa.Height / 2 - frmResultados.Height / 2
```

```
frmGraficosVarios.Left = frmAlternativa.Width / 2 - frmGraficosVarios.Width / 2
frmGraficosVarios.Top = frmAlternativa.Height / 2 - frmGraficosVarios.Height / 2
```

```
frmPresentacion.Left = frmAlternativa.Width / 2 - frmPresentacion.Width / 2
frmPresentacion.Top = frmAlternativa.Height / 2 - frmPresentacion.Height / 2
```

'Ubicación de la ventana de gráficos

```
picGraficos.Height = frmAlternativa.Height * 0.7
picGraficos.Width = frmAlternativa.Width * 0.95
picGraficos.Left = frmAlternativa.Width / 2 - picGraficos.Width / 2
picGraficos.Top = frmAlternativa.Height / 2 - picGraficos.Height / 2
```

```
lblTitulo.Left = frmAlternativa.Width / 2 - lblTitulo.Width / 2
lblTitulo.Top = picGraficos.Top - lblTitulo.Height * 1.1
lblEjeY.Left = picGraficos.Left
lblEjeY.Top = lblTitulo.Top - lblEjeY.Height * 1.4
lblEjeYtext.Left = lblEjeY.Left + lblEjeY.Width
lblEjeYtext.Top = lblEjeY.Top
lblEscalaEjeY.Left = lblEjeYtext.Left + lblEjeYtext.Width * 1.1
lblEscalaEjeY.Top = lblEjeYtext.Top
```

```
LblEjeX.Left = lblEjeY.Left
LblEjeX.Top = lblEjeY.Top - LblEjeX.Height * 1.3
lblEjeXtext.Left = LblEjeX.Left + LblEjeX.Width
lblEjeXtext.Top = LblEjeX.Top
lblEscalaEjeX.Left = lblEjeXtext.Left + lblEjeXtext.Width * 1.1
lblEscalaEjeX.Top = lblEjeXtext.Top
```

```
lblTipoControl.Left = picGraficos.Left + _
    picGraficos.Width - lblTipoControl.Width
lblTipoControl.Top = lblEjeXtext.Top
```

'Para la animación

'Definición del area donde va a estar el carrito para la animación
'picGraficos.Scale(-x,y)-(-x,-y)

```
cmdCongelar.Left = picGraficos.Left 'Boton que congela o descongela la imagen
cmdCongelar.Top = lblTipoControl.Top 'de simulación
```

```
FrameValores.Left = 0 'Cuadro que contiene las variables de ángulo
    'Velocidad angular y fuerza
```

```
FrameValores.Top = picGraficos.Height - FrameValores.Height
```

```
FrameValores_L.Left = picGraficos.Width - FrameValores_L.Width 'Cuadro que contiene las variables de
posición
```

```

                                'Velocidad y tiempo
FrameValores_L.Top = FrameValores.Top

If (picGraficos.Width - FrameValores.Width - FrameValores_L.Width) > 0 Then
    FrameGraficos.Width = picGraficos.Width - _
        FrameValores.Width - FrameValores_L.Width
Else
End If

FrameGraficos.Left = FrameValores.Width 'Cuadro que contiene las variables de ángulo
                                'Velocidad angular y fuerza
FrameGraficos.Top = FrameValores.Top

FrameVelocidad.Top = 0          'cuadro que contiene la velocidad
FrameVelocidad.Left = picGraficos.Width - FrameVelocidad.Width 'De simulación

'frmAlternativa.ScaleMode = VarAuxEscalafrm

If lblTitulo.Caption = "Simulación Dinámica" Then

    picGraficos.Scale (-CuadroX, CuadroY)-(CuadroX, -CuadroY)

ElseIf lblTitulo.Caption = "Gráficos Varios" Then

    If ROJO = ShapeCurva.BackColor Then 'Si selecciono la curva de
        'Espacio Vs tiempo
        picGraficos.Scale (-Tmax / (3 * DIV), FdeX1max + FdeX1max / DIV)- _
            (Tmax + Tmax / (3 * DIV), -FdeX1max - FdeX1max / DIV)

    ElseIf AZUL = ShapeCurva.BackColor Then
        picGraficos.Scale (-Tmax / (3 * DIV), FdeX2max + FdeX2max / DIV)- _
            (Tmax + Tmax / (3 * DIV), -FdeX2max - FdeX2max / DIV)

    ElseIf VERDE = ShapeCurva.BackColor Then
        picGraficos.Scale (-Tmax / (3 * DIV), FdeX3max + FdeX3max / DIV)- _
            (Tmax + Tmax / (3 * DIV), -FdeX3max - FdeX3max / DIV)

    ElseIf CELESTE = ShapeCurva.BackColor Then
        picGraficos.Scale (-Tmax / (3 * DIV), FdeX4max + FdeX4max / DIV)- _
            (Tmax + Tmax / (3 * DIV), -FdeX4max - FdeX4max / DIV)

    ElseIf LILA = ShapeCurva.BackColor Then
        picGraficos.Scale (-Tmax / (3 * DIV), FdeU5max + FdeU5max / DIV)- _
            (Tmax + Tmax / (3 * DIV), -FdeU5max - FdeU5max / DIV)
    Else
        picGraficos.Scale (-Tmax / (3 * DIV), FdeU5max + FdeU5max / DIV)- _
            (Tmax + Tmax / (3 * DIV), -FdeU5max - FdeU5max / DIV)
    End If

ElseIf lblTitulo.Caption = "Espacio Vs Tiempo" Then
    picGraficos.Scale (-Tmax / (3 * DIV), FdeX1max + FdeX1max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeX1max - FdeX1max / DIV)

ElseIf lblTitulo.Caption = "Velocidad Vs Tiempo" Then

```

```

picGraficos.Scale (-Tmax / (3 * DIV), FdeX2max + FdeX2max / DIV)- _
(Tmax + Tmax / (3 * DIV), -FdeX2max - FdeX2max / DIV)

ElseIf lblTitulo.Caption = "Angulo Vs Tiempo" Then
picGraficos.Scale (-Tmax / (3 * DIV), FdeX3max + FdeX3max / DIV)- _
(Tmax + Tmax / (3 * DIV), -FdeX3max - FdeX3max / DIV)

ElseIf lblTitulo.Caption = "Velocidad Angular Vs Tiempo" Then
picGraficos.Scale (-Tmax / (3 * DIV), FdeX4max + FdeX4max / DIV)- _
(Tmax + Tmax / (3 * DIV), -FdeX4max - FdeX4max / DIV)

ElseIf lblTitulo.Caption = "Fuerza Vs Tiempo" Then
picGraficos.Scale (-Tmax / (3 * DIV), FdeU5max + FdeU5max / DIV)- _
(Tmax + Tmax / (3 * DIV), -FdeU5max - FdeU5max / DIV)

ElseIf lblTitulo.Caption = "Velocidad Angular Vs Angulo" Then
picGraficos.Scale (-FdeX3max - FdeX3max / DIV, FdeX4max _
+ FdeX4max / DIV)-(FdeX3max + FdeX3max / DIV, _
-FdeX4max - FdeX4max / DIV)
End If

End Sub

'Subrutina en la cual se inicializan las variables para todos los
'controles
Private Sub InicializacionVariables()

PuntoDecimal = Mid(CStr(GRAVeDAD), 2, 1) 'Inicialización del punto decimal
Gr = GRAVeDAD
mP = MASApENDULO
Mc = MASAcARRO
LongP = LONGpEND
Jp = iNERCIaJ
'Jp = mP * LongP * LongP / 3
'Lp = (Jp + mP * LongP ^ 2) / (mP * LongP)
Xest(1) = ESPiNIC
Xest(2) = VELiNIC
Xest(3) = ANGiNIC * GRADOSaRAD
Xest(4) = VELaNGiNIC * GRADOSaRAD

'Para la simulación
LongPista = LONGITUDpISTA 'Para Proceso de simulación
IntervaloSim = PASOsIMULACION
frmAlternativa.HScrollVelocidad.Value = IntervaloSim

'Inicialización de variables
CambioVelocidad = 0 'Indica que no ha existido cambio en velocidad
'de animación

'Control de realimentación de estado
PoloD(1).r = POLOrEAL1
PoloD(1).I = POLOiMAGINARIO1
PoloD(2).r = POLOrEAL2
PoloD(2).I = POLOiMAGINARIO2

```

```

PoloD(3).r = POLOrEAL3
PoloD(3).I = POLOiMAGINARIO3
PoloD(4).r = POLOrEAL4
PoloD(4).I = POLOiMAGINARIO4

```

```
'Control por Acciones
```

```
'Control Proporcional derivativo Conociendo Wn y Si
```

```
Wn = PD_Wn_CONST
```

```
Si = PD_Si_CONST
```

```
'Control Proporcional derivativo Conociendo Kp y Kd
```

```
Ad_Kp = PD_Kp_CONST
```

```
Ad_Kd = PD_Kd_CONST
```

```
'Control PID
```

```
A_Kp = PID_Kp_CONST
```

```
A_Kd = PID_Kd_CONST
```

```
A_Ki = PID_Ki_CONST
```

```
'Control Red de Adelanto de Fase por el metodo del cero arbitrario
```

```
RedcMp = RED_Mp / 100
```

```
RedcTs = RED_Ts
```

```
RedcCc = Red_Cc
```

```
'Control Red de Adelanto de Fase por el metodo de la bisectriz
```

```
RedbMp = RED_Mp / 100
```

```
RedbTs = RED_Ts
```

```
'Control fuzzy
```

```
Tfuzzy = tiEMPofUZZY
```

```
TsimulaciOn = tIEMPOsIMULACION
```

```
Rango_u = rANGOfUERZA
```

```
Rango_f = rANGOaNGULO * GRADOsRAD
```

```
Rango_w = rANGOvELaNGULAR * GRADOsRAD
```

```
'Inicialización de la base de conocimiento
```

```
'Definición de base de conocimiento, Baseconoc(I,J), I define la fila
```

```
'y J define la columna
```

```
'Numeros asociados a los conjuntos fuzzy
```

```
'Muy Negativo MN = 1
```

```
'Poco Negativo PN = 2
```

```
'Cero C = 3
```

```
'Poco Positivo PP = 4
```

```
'Muy Positivo MP = 5
```

```
'
```

```
'
```

```
ANGULO
```

```
'
```

```
| MN | PN | C | PP | MP |
```

```
'
```

```
| MN| MN | MN | mn | C | C |
```

```
'
```

```
| PN| MN | PN | pn | pp | C |
```

```
'
```

```
Velocidad |-----|
```

```
'
```

```
Angular |C| mn | pn | c | pp | mp |
```

```

'      |-----|
'      | PP| C | c | pp | pp | MP |
'      |-----|
'      | MP| C | C | MP | MP | MP |
'      |-----|

```

With frmTiposDeControl

```

.lblConocimiento(0).Caption = "MN"
.lblConocimiento(1).Caption = "MN"
.lblConocimiento(2).Caption = "MN"
.lblConocimiento(3).Caption = "C"
.lblConocimiento(4).Caption = "C"

```

```

.lblConocimiento(5).Caption = "MN"
.lblConocimiento(6).Caption = "PN"
.lblConocimiento(7).Caption = "PN"
.lblConocimiento(8).Caption = "PP"
.lblConocimiento(9).Caption = "PP"

```

```

.lblConocimiento(10).Caption = "MN"
.lblConocimiento(11).Caption = "PN"
.lblConocimiento(12).Caption = "C"
.lblConocimiento(13).Caption = "PP"
.lblConocimiento(14).Caption = "MP"

```

```

.lblConocimiento(15).Caption = "PN"
.lblConocimiento(16).Caption = "PN"
.lblConocimiento(17).Caption = "PP"
.lblConocimiento(18).Caption = "PP"
.lblConocimiento(19).Caption = "MP"

```

```

.lblConocimiento(20).Caption = "C"
.lblConocimiento(21).Caption = "C"
.lblConocimiento(22).Caption = "MP"
.lblConocimiento(23).Caption = "MP"
.lblConocimiento(24).Caption = "MP"

```

End With

```

'Se define la escala de la forma frmalternativa en
'píxeles (Twips)
frmAlternativa.ScaleMode = 1

```

```

CuadroX = LONGITUDpISTA * 13 / 20 'Para definir
      'la escala "X" del área donde
      'va a estar el carrito para la animación
CuadroY = LONGITUDpISTA / 4 'Para definir la escala "y" del área donde
      'va a estar el carrito para la animación
      'picGraficos.Scale(-x,y)-(x,-y)

```

End Sub

```

'Subrutina que realiza los cálculos para el control
'Proporcional derivativo conociendo Wn y Si
Private Sub CalculosAcciones_WnE()
'Declaración de variables para las acciones de control
Dim Kp As Single
Dim Kd As Single

Kp = Mc * Lp * (Wn ^ 2 + Gr / Lp)
Kd = (2 * Si * Wn * Lp * Mc) / Kp
CalculosPD Wn, Si, Kp, Kd

'Cargo los resultados obtenidos en la ventana de resultados
MuestraResultados_PD Wn, Si, Kp, Kd

'Pongo el Tipo de control utilizado que se presentará en el
'momento de observar las gráficas en el tiempo de los diferentes
'estados
lblTipoControl.Caption = "Control PD conociendo Wn y E"
frmResultados.SSTabResultados.Tab = 2
End Sub

```

```

'Subrutina principal de los cálculos para el control por
'Realimentación de estado
Private Sub CalculosRealimentacionEstado()
'Se determina el valor a la variable CasoRealim
LectPoloDes

'Organiza los polos ingresados de una manera normalizada
OrganPolos

'Calculo de los coeficientes del polinomio característico
'con los polos ingresados y los valores de el vector de
'realimentación de estado.
CalcCoef

' Cálculos de coeficientes de los polinomios del numerador
'del el vector de estado
CoefNumeEstado
Tmax = 8 / Abs(PoloMaximo)
DeltaT = Tmax / NUMmAXpUNTOS
Select Case CasoRealim
Case CUATROprDIF
FraccParcCUATROprDIF
XiCUATROprDIF

Case DOSprIGDOSprDIF
FraccParcDOSprIGDOSprDIF
XiDOSprIGDOSprDIF

Case TRESprIGUNprDIF
FraccParcTRESprIGUNprDIF
XiTRESprIGUNprDIF

```

```
Case CUATROprIG
  FraccParcCUATROprIG
  XtCUATROprIG
```

```
Case DOSprIGDOSprIG
  FraccParcDOSprIGDOSprIG
  XtDOSprIGDOSprIG
```

```
Case DOSpcDIF
  FraccParcDOSpcDIF
  XtDOSpcDIF
```

```
Case DOSpcIG
  FraccParcDOSpcIG
  XtDOSpcIG
```

```
Case UNpcDOSprDIF
  FraccParcUNpcDOSprDIF
  XtUNpcDOSprDIF
```

```
Case UNpcDOSprIG
  FraccParcUNpcDOSprIG
  XtUNpcDOSprIG
```

```
End Select
```

```
'Carga de resultados en la ventana de resultados
MuestraResultados_Realim
```

```
'Se pongo el Tipo de control utilizado que se presentará en el
'momento de observar las gráficas en el tiempo de los diferentes
'estados
lblTipoControl.Caption = "Control Por Realimentación de Estado"
frmResultados.SSTabResultados.Tab = 1
End Sub
```

```
'Subrutina que realiza los calculos para el control por
'Red de adelanto de fase por el metodo de la Bisectriz
Private Sub CalculosAdelBisectriz()
Dim RedbPc As Single      'Ubicación del polo de la red de Gc
Dim RedbCc As Single      'Ubicación del cero de la red de Gc
Dim RedbKc As Single      'Ganancia de Gc
Dim R_Si As Single
Dim R_Wn As Single
Dim Wd As Single
Dim Ro As Single
Dim Fi_Pd As Single      'Angulo del polo dominante deseado
Dim Fi_c As Single      'Angulo que va a dar el compensador
Dim Fb(0 To 2) As Single 'Coeficientes del numerador del
                        'estado Angulo en S
Dim Da(0 To 3) As Single 'Coeficientes del Denominador de
                        'los estado Angulo y espacio en S
```

```

Dim Ub(0 To 3) As Single 'Coeficientes del numerador del
                          'estado Fuerza en S
Dim Fi_A(1 To 3) As Single 'Coeficientes de la division en
                          'fracciones parciales del ángulo
Dim FiW_A(1 To 3) As Single 'Coeficientes de la division en
                          'fracciones parciales de la velocidad angular
Dim FiA_A(1 To 3) As Single 'Coeficientes de la division en
                          'fracciones parciales de la aceleración angular

Dim U_A(1 To 2) As Single 'Coeficientes de la división en
                          'fracciones parciales de la fuerza
Dim X_A(1 To 3) As Single 'Coeficientes de la division en
                          'fracciones parciales del espacio
Dim V_A(1 To 3) As Single 'Coeficientes de la division en
                          'fracciones parciales de la velocidad

Dim Acel_W As Double      'Para almacenar el valor de la aceleración
                          'angular como no se lo gráfica no es
                          'necesario definir un arreglo
                          'fracciones parciales de la velocidad

Dim Acel_Ini              ' condición inicial de la aceleración angular
Dim Cr1 As Double         'Constantes de la integración en el cálculo
                          'de la velocidad y el espacio
Dim Cr2 As Double         'Constantes de la integración en el calculo
                          'de la velocidad y el espacio
Dim Y As Single           'Valor de la division en fracciones
                          'parciales del denominador de Fi(s)
Dim DenRed As Single      'variable temporal
Dim T As Single
'Dim Ies As Integer

Dim BandError As Integer  'bandera que se pone en uno cuando
                          'los valores de el ángulo del compensador
                          'no es posible compensar

BandError = 0
R_Si = Sqr((Log(RedbMp)) ^ 2 / (PI ^ 2 + (Log(RedbMp)) ^ 2))
R_Wn = 4 / (R_Si * RedbTs)
Wd = R_Wn * Sqr(1 - R_Si ^ 2)
Ro = R_Si * R_Wn
DenRed = Ro ^ 2 - Wd ^ 2 - Gr / Lp

If DenRed < 0 Then
    Fi_c = Atn(-2 * Ro * Wd / DenRed)
ElseIf DenRed = 0 Then
    Fi_c = -270 * GRADOSaRAD
    BandError = 1
Else
    Fi_c = -180 * GRADOSaRAD + Atn(-2 * Ro * Wd / DenRed)
    BandError = 1
End If

Fi_Pd = PI - Atn(Wd / Ro)

```

```

If Tan(Fi_Pd / 2 - Fi_c / 2) <> 0 Then

    RedbPc = -Ro - Wd / Tan(Fi_Pd / 2 - Fi_c / 2)
    RedbCc = -Ro - Wd / Tan(Fi_Pd / 2 + Fi_c / 2)
    RedbKc = Mc * Lp * Sqr((-Ro - RedbPc) ^ 2 + Wd ^ 2) * _
        Sqr((Ro ^ 2 - Wd ^ 2 - Gr / Lp) ^ 2 + (2 * Ro * Wd) ^ 2) / _
        Sqr((-Ro - RedbCc) ^ 2 + Wd ^ 2)
    Y = -RedbPc - 2 * Ro
Else
    BandError = 1
End If

'Se chequea si se puede encontrar las ecuaciones de velocidad y posición
If (RedbKc * RedbCc - RedbPc * Mc * Gr <> 0) And BandError = 0 Then

    'Valor inicial de la aceleración angular
    Acel_Ini = ((RedbPc * Gr * Mc - RedbKc * RedbCc) * Xest(3) + _
        (RedbKc - Gr * Mc) * Xest(4)) / (RedbPc * Lp * Mc)

    'Coeficientes de la ecuación de estado del ángulo
    'Fi(s)= Fb2*s^2+Fb1*s+Fb0/(Da3*s^3+Da2*s^2+Da1*s+Da0

    Fb(2) = Xest(3)
    Fb(1) = -Xest(3) * RedbPc + Xest(4)
    Fb(0) = Acel_Ini - Xest(4) * RedbPc + Xest(3) * _
        (RedbKc - Gr * Mc) / (Lp * Mc)

    'Da(3) = 1
    'Da(2) = -RedbPc
    'Da(1) = (RedbKc / Mc - Gr) / Lp
    'Da(0) = (-RedbKc * RedbCc / Mc + RedbPc * Gr) / Lp
    'División de los coeficientes de las ecuaciones en el tiempo de
    'el ángulo , la velocidad angular, y aceleración angular
    'Fi(t)=A(1)*Exp(-Y*t)+Exp(-Ro*t)*(A(2)*cos(Wd*t)+A(3)*sen(Wd*t)
    'Fi_W(t)=A(1)*Exp(-Y*t)+Exp(-Ro*t)*(A(2)*cos(Wd*t)+A(3)*sen(Wd*t)
    'Fi_A(t)=A(1)*Exp(-Y*t)+Exp(-Ro*t)*(A(2)*cos(Wd*t)+A(3)*sen(Wd*t)

    'Angulo
    Fi_A(1) = (Fb(2) * Y ^ 2 - Fb(1) * Y + Fb(0)) / _
        ((-Y + Ro) ^ 2 + Wd ^ 2)
    Fi_A(2) = Fb(2) - Fi_A(1)
    Fi_A(3) = (Fb(1) - 2 * Ro * Fi_A(1) - Fi_A(2) * (Ro + Y)) / Wd

    'Velocidad angular
    FiW_A(1) = -Fi_A(1) * Y
    FiW_A(2) = -Ro * Fi_A(2) + Fi_A(3) * Wd
    FiW_A(3) = -(Wd * Fi_A(2) + Ro * Fi_A(3))

    'Aceleración angular

    FiA_A(2) = -Ro * FiW_A(2) + FiW_A(3) * Wd
    FiA_A(1) = -FiW_A(1) * Y
    FiA_A(3) = -(Wd * FiW_A(2) + Ro * FiW_A(3))

```

'Coeficientes de la ecuación en el tiempo de la fuerza

$$U(t) = A(1) * Fi_A(t) + A(2) * Fi(t)$$

$$U_A(1) = -Lp * Mc$$

$$U_A(2) = Gr * Mc$$

'Coeficientes de la ecuación en el tiempo de la velocidad y
'posición

$$V_A(1) = -Gr / (RedbKc * RedbCc - RedbPc * Mc * Gr)$$

$$V_A(2) = -RedbKc * RedbCc * Lp / (RedbKc * RedbCc - RedbPc * Mc * Gr)$$

$$V_A(3) = Gr * RedbKc / (RedbKc * RedbCc - RedbPc * Mc * Gr)$$

$$X_A(1) = V_A(1) * Mc + V_A(3) * RedbPc * Mc / (RedbKc * RedbCc)$$

$$X_A(2) = V_A(2) + V_A(3) / RedbCc$$

$$X_A(3) = -V_A(3) / (RedbKc * RedbCc)$$

'Tiempo de establecimiento con el criterio del 2% $t_s = 4 / (Si * Wn)$

If $(R_Si * R_Wn) < Y$ Then

$$Tmax = 8 / (R_Si * R_Wn)$$

Else

$$Tmax = 8 / Y$$

End If

$$\Delta T = Tmax / NUMmAXpUNTOS$$

$$FdeX1max = 0$$

$$FdeX2max = 0$$

$$FdeX3max = 0$$

$$FdeX4max = 0$$

$$FdeU5max = 0$$

$$T = 0$$

For Ies = 0 To NUMmAXpUNTOS Step 1

$$Ang(Ies) = Fi_A(1) * Exp(-Y * T) + Exp(-Ro * T) * (Fi_A(2) * Cos(Wd * T) + Fi_A(3) * Sin(Wd * T))$$

If $FdeX3max < Abs(Ang(Ies))$ Then

$$FdeX3max = Abs(Ang(Ies))$$

End If

$$Wang(Ies) = FiW_A(1) * Exp(-Y * T) + Exp(-Ro * T) * (FiW_A(2) * Cos(Wd * T) + FiW_A(3) * Sin(Wd * T))$$

If $FdeX4max < Abs(Wang(Ies))$ Then

$$FdeX4max = Abs(Wang(Ies))$$

End If

```
Acel_W = FiA_A(1) * Exp(-Y * T) + Exp(-Ro * T) * _
          (FiA_A(2) * Cos(Wd * T) + FiA_A(3) * Sin(Wd * T))
```

```
Fuerza(Ies) = Acel_W * U_A(1) + Ang(Ies) * U_A(2)
```

```
If FdeU5max < Abs(Fuerza(Ies)) Then
  FdeU5max = Abs(Fuerza(Ies))
End If
```

```
If Ies = 0 Then
  Cr1 = Xest(2) - (V_A(1) * Fuerza(Ies) _
    + V_A(2) * Wang(Ies) + V_A(3) * Ang(Ies))
```

```
End If
```

```
Vel(Ies) = V_A(1) * Fuerza(Ies) + V_A(2) * Wang(Ies) + _
          V_A(3) * Ang(Ies) + Cr1
```

```
If FdeX2max < Abs(Vel(Ies)) Then
  FdeX2max = Abs(Vel(Ies))
End If
```

```
If Ies = 0 Then
  Cr2 = Xest(1) - X_A(1) * Vel(Ies) - X_A(2) * Ang(Ies) _
    - X_A(3) * Fuerza(Ies)
End If
```

```
Esp(Ies) = X_A(1) * Vel(Ies) + X_A(2) * Ang(Ies) + _
          X_A(3) * Fuerza(Ies) + Cr1 * T + Cr2
```

```
If FdeX1max < Abs(Esp(Ies)) Then
  FdeX1max = Abs(Esp(Ies))
End If
```

```
'incremento el tiempo
```

```
T = T + DeltaT
```

```
If (Ies Mod NumBarra) = 0 Then
  BarraProceso Ies
End If
```

```
Next Ies
```

```
'Se carga los resultados obtenidos en la ventana de resultados
MuestraResultados_red R_Si, R_Wn, Wd, Ro, Fi_c, RedbKc, RedbCc, RedbPc, Y
```

```
'Se pone el Tipo de control utilizado que se presentará en el
'momento de observar las gráficas en el tiempo de los diferentes
'estados
lblTipoControl.Caption = "Red de Adelanto de fase por Bisectriz"
frmResultados.SSTabResultados.Tab = 4
frmResultados.SSTabRedAdelFase.Tab = 0
```

```
Else
```

```
msg = " La ubicación propuesta para el cero del compensador " & _
      "impide la construcción de la red de adelanto de fase"
```

```
Mostrar
```

```
End If
End Sub
```

```
'Subrutina que realiza los calculos para el control por
'Red de adelanto de fase por el método de la ubicación del cero
'Arbitrario
Private Sub CalculosAdelantoCero()
Dim RedcPc As Single      'Ubicación del polo de la red de Gc
Dim RedcKc As Single      'Ganancia de Gc
Dim R_Si As Single
Dim R_Wn As Single
Dim Wd As Single
Dim Ro As Single
Dim Fi_Pd As Single      'Angulo del polo dominante deseado
Dim Fi_c As Single      'Angulo que va a dar el compensador

Dim Fb(0 To 2) As Single  'Coeficientes del numerador del
                        'estado Angulo en S
Dim Da(0 To 3) As Single  'Coeficientes del Denominador de
                        'los estado Angulo y espacio en S
Dim Ub(0 To 3) As Single  'Coeficientes del numerador del
                        'estado Fuerza en S
Dim Fi_A(1 To 3) As Single 'Coeficientes de la division en
                        'fracciones parciales del ángulo
Dim FiW_A(1 To 3) As Single 'Coeficientes de la division en
                        'fracciones parciales de la velocidad angular
Dim FiA_A(1 To 3) As Single 'Coeficientes de la division en
                        'fracciones parciales de la aceleración angular

Dim U_A(1 To 2) As Single 'Coeficientes de la division en
                        'fracciones parciales de la fuerza
Dim X_A(1 To 3) As Single 'Coeficientes de la division en
                        'fracciones parciales del espacio
Dim V_A(1 To 3) As Single 'Coeficientes de la division en

Dim Acel_W As Double      'Para almacenar el valor de la aceleración
                        'angular como no se lo gráfica no es
                        'necesario definir un arreglo
                        'fracciones parciales de la velocidad

Dim Acel_Ini      ' Condición inicial de la aceleración angular
Dim Cr1 As Double  'Constantes de la integración en el calculo
                        'de la velocidad y el espacio
Dim Cr2 As Double  'Constantes de la integración en el calculo
                        'de la velocidad y el espacio
Dim Y As Single    'Valor de la division en fracciones
                        'parciales del denominador de Fi(s)
```

Dim T As Single
'Dim Ies As Integer

$R_Si = \text{Sqr}((\text{Log}(\text{RedcMp}))^2 / (\text{PI}^2 + (\text{Log}(\text{RedcMp}))^2))$
 $R_Wn = 4 / (R_Si * \text{RedcTs})$
 $Wd = R_Wn * \text{Sqr}(1 - R_Si^2)$
 $Ro = R_Si * R_Wn$
 $Fi_c = \text{Atn}(-2 * Ro * Wd / (Ro^2 - Wd^2 - Gr / Lp))$

If Abs(RedcCc) > Ro Then
 $\text{RedcPc} = -Ro - Wd / \text{Tan}(\text{Atn}(Wd / (-Ro - \text{RedcCc})) - Fi_c)$

ElseIf Abs(RedcCc) = Ro Then
 $\text{RedcPc} = -Ro - Wd / \text{Tan}(\text{PI} / 2 - Fi_c)$

Else
 $\text{RedcPc} = -Ro - Wd / \text{Tan}(\text{PI} + \text{Atn}(Wd / (-Ro - \text{RedcCc})) - Fi_c)$
End If

If RedcPc < 0 Then

$\text{RedcKc} = Mc * Lp * \text{Sqr}((-Ro - \text{RedcPc})^2 + Wd^2) * _$
 $\text{Sqr}((Ro^2 - Wd^2 - Gr / Lp)^2 + (2 * Ro * Wd)^2) / _$
 $\text{Sqr}((-Ro - \text{RedcCc})^2 + Wd^2)$
 $Y = -\text{RedcPc} - 2 * Ro$

'Valor inicial de la aceleración angular

$\text{Acel_Ini} = ((\text{RedcPc} * Gr * Mc - \text{RedcKc} * \text{RedcCc}) * \text{Xest}(3) + _$
 $(\text{RedcKc} - Gr * Mc) * \text{Xest}(4)) / (\text{RedcPc} * Lp * Mc)$

'Coeficientes de la ecuación de estado del ángulo

$Fi(s) = Fb2*s^2 + Fb1*s + Fb0 / (Da3*s^3 + Da2*s^2 + Da1*s + Da0)$

$Fb(2) = \text{Xest}(3)$
 $Fb(1) = -\text{Xest}(3) * \text{RedcPc} + \text{Xest}(4)$
 $Fb(0) = \text{Acel_Ini} - \text{Xest}(4) * \text{RedcPc} + \text{Xest}(3) * _$
 $(\text{RedcKc} - Gr * Mc) / (Lp * Mc)$

'Division de los coeficientes de las ecuaciones en el tiempo de

'el ángulo, la velocidad angular, y aceleración angular

$Fi(t) = A(1) * \text{Exp}(-Y*t) + \text{Exp}(-Ro*t) * (A(2) * \cos(Wd*t) + A(3) * \text{sen}(Wd*t))$
 $Fi_W(t) = A(1) * \text{Exp}(-Y*t) + \text{Exp}(-Ro*t) * (A(2) * \cos(Wd*t) + A(3) * \text{sen}(Wd*t))$
 $Fi_A(t) = A(1) * \text{Exp}(-Y*t) + \text{Exp}(-Ro*t) * (A(2) * \cos(Wd*t) + A(3) * \text{sen}(Wd*t))$

'Angulo

$Fi_A(1) = (Fb(2) * Y^2 - Fb(1) * Y + Fb(0)) / _$
 $((-Y + Ro)^2 + Wd^2)$
 $Fi_A(2) = Fb(2) - Fi_A(1)$
 $Fi_A(3) = (Fb(1) - 2 * Ro * Fi_A(1) - Fi_A(2) * (Ro + Y)) / Wd$

'Velocidad angular

$FiW_A(1) = -Fi_A(1) * Y$
 $FiW_A(2) = -Ro * Fi_A(2) + Fi_A(3) * Wd$
 $FiW_A(3) = -(Wd * Fi_A(2) + Ro * Fi_A(3))$

'Aceleración angular

$$\begin{aligned} \text{FiA_A}(1) &= -\text{FiW_A}(1) * Y \\ \text{FiA_A}(2) &= -\text{Ro} * \text{FiW_A}(2) + \text{FiW_A}(3) * \text{Wd} \\ \text{FiA_A}(3) &= -(\text{Wd} * \text{FiW_A}(2) + \text{Ro} * \text{FiW_A}(3)) \end{aligned}$$

'Coeficientes de la ecuación en el tiempo de la fuerza

$$\text{U}(t) = \text{A}(1) * \text{Fi_A}(t) + \text{A}(2) * \text{Fi}(t)$$

$$\text{U_A}(1) = -\text{Lp} * \text{Mc}$$

$$\text{U_A}(2) = \text{Gr} * \text{Mc}$$

'Coeficientes de la ecuación en el tiempo de la velocidad y

'posición

$$\text{V_A}(1) = -\text{Gr} / (\text{RedcKc} * \text{RedcCc} - \text{RedcPc} * \text{Mc} * \text{Gr})$$

$$\text{V_A}(2) = -\text{RedcKc} * \text{RedcCc} * \text{Lp} / (\text{RedcKc} * \text{RedcCc} - \text{RedcPc} * \text{Mc} * \text{Gr})$$

$$\text{V_A}(3) = \text{Gr} * \text{RedcKc} / (\text{RedcKc} * \text{RedcCc} - \text{RedcPc} * \text{Mc} * \text{Gr})$$

$$\text{X_A}(1) = \text{V_A}(1) * \text{Mc} + \text{V_A}(3) * \text{RedcPc} * \text{Mc} / (\text{RedcKc} * \text{RedcCc})$$

$$\text{X_A}(2) = \text{V_A}(2) + \text{V_A}(3) / \text{RedcCc}$$

$$\text{X_A}(3) = -\text{V_A}(3) / (\text{RedcKc} * \text{RedcCc})$$

'Tiempo de establecimiento con el criterio del 2% $t_s = 4 / (\text{Si} * \text{Wn})$

If $(\text{R_Si} * \text{R_Wn}) < Y$ Then

$$\text{Tmax} = 8 / (\text{R_Si} * \text{R_Wn})$$

Else

$$\text{Tmax} = 8 / Y$$

End If

$$\text{DeltaT} = \text{Tmax} / \text{NUMmAXpUNTOS}$$

'Obtención de los valores en el tiempo para los gráficos

$$\text{FdeX1max} = 0$$

$$\text{FdeX2max} = 0$$

$$\text{FdeX3max} = 0$$

$$\text{FdeX4max} = 0$$

$$\text{FdeU5max} = 0$$

$$\text{T} = 0$$

For $\text{Ies} = 0$ To NUMmAXpUNTOS Step 1

$$\begin{aligned} \text{Ang}(\text{Ies}) &= \text{Fi_A}(1) * \text{Exp}(-Y * \text{T}) + \text{Exp}(-\text{Ro} * \text{T}) * (\text{Fi_A}(2) _ \\ &\quad * \text{Cos}(\text{Wd} * \text{T}) + \text{Fi_A}(3) * \text{Sin}(\text{Wd} * \text{T})) \end{aligned}$$

If $\text{FdeX3max} < \text{Abs}(\text{Ang}(\text{Ies}))$ Then

$$\text{FdeX3max} = \text{Abs}(\text{Ang}(\text{Ies}))$$

End If

$$\begin{aligned} \text{Wang}(\text{Ies}) &= \text{FiW_A}(1) * \text{Exp}(-Y * \text{T}) + \text{Exp}(-\text{Ro} * \text{T}) * (\text{FiW_A}(2) _ \\ &\quad * \text{Cos}(\text{Wd} * \text{T}) + \text{FiW_A}(3) * \text{Sin}(\text{Wd} * \text{T})) \end{aligned}$$

If $\text{FdeX4max} < \text{Abs}(\text{Wang}(\text{Ies}))$ Then

$$\text{FdeX4max} = \text{Abs}(\text{Wang}(\text{Ies}))$$

End If

```
Acel_W = FiA_A(1) * Exp(-Y * T) + Exp(-Ro * T) * _
          (FiA_A(2) * Cos(Wd * T) + FiA_A(3) * Sin(Wd * T))
```

```
Fuerza(Ies) = Acel_W * U_A(1) + Ang(Ies) * U_A(2)
```

```
If FdeU5max < Abs(Fuerza(Ies)) Then
```

```
    FdeU5max = Abs(Fuerza(Ies))
```

```
End If
```

```
If Ies = 0 Then
```

```
    Cr1 = Xest(2) - (V_A(1) * Fuerza(Ies) _
                  + V_A(2) * Wang(Ies) + V_A(3) * Ang(Ies))
```

```
End If
```

```
Vel(Ies) = V_A(1) * Fuerza(Ies) + V_A(2) * Wang(Ies) + _
           V_A(3) * Ang(Ies) + Cr1
```

```
If FdeX2max < Abs(Vel(Ies)) Then
```

```
    FdeX2max = Abs(Vel(Ies))
```

```
End If
```

```
If Ies = 0 Then
```

```
    Cr2 = Xest(1) - X_A(1) * Vel(Ies) - X_A(2) * Ang(Ies) _
          - X_A(3) * Fuerza(Ies)
```

```
End If
```

```
Esp(Ies) = X_A(1) * Vel(Ies) + X_A(2) * Ang(Ies) + _
           X_A(3) * Fuerza(Ies) + Cr1 * T + Cr2
```

```
If FdeX1max < Abs(Esp(Ies)) Then
```

```
    FdeX1max = Abs(Esp(Ies))
```

```
End If
```

```
'incremento el tiempo
```

```
    T = T + DeltaT
```

```
If (Ies Mod NumBarra) = 0 Then
```

```
    BarraProceso Ies
```

```
End If
```

```
Next Ies
```

```
'Cargo los resultados obtenidos en la ventana de resultados
```

```
    MuestraResultados_red R_Si, R_Wn, Wd, Ro, Fi_c, RedcKc, RedcCc, RedcPc, Y
```

```
'Se pone el Tipo de control utilizado que se presentará en el
```

```
'momento de observar las gráficas en el tiempo de los diferentes
```

```
'estados
```

```
    lblTipoControl.Caption = "Red de Adelanto de fase por Cero Arbitrario"
```

```
    frmResultados.SSTabResultados.Tab = 4
```

```
    frmResultados.SSTabRedAdelFase.Tab = 1
```

```
Else
```

```
msg = " La ubicación propuesta para el cero del compensador " & _
      "impide la construcción de la red de adelanto de fase"
```

```
Mostrar
```

```
End If
```

```
End Sub
```

```
'Subrutina que realiza los calculos para el control
```

```
'Proporcional derivativo, utilizado para las dos opciones
```

```
'conociendo Kp , Kd y conociendo Wn y Si
```

```
Private Sub CalculosPD(A_Wn As Single, A_Si As Single, Kp As Single, Kd As Single)
```

```
'Declaración de variables para las acciones de control
```

```
Dim Fi_x As Single
```

```
Dim Wd As Single
```

```
Dim Si_r As Single
```

```
Dim Co_fi As Single
```

```
Dim Tiempo As Single
```

```
'Dim Ies As Integer
```

```
Dim a As Single, b As Single, Ces1 As Single, Ces2 As Single
```

```
Dim Ces3 As Single, Ces4 As Single, C1 As Double, C2 As Double
```

```
Si_r = Sqr(1 - A_Si ^ 2)
```

```
Fi_x = Atn(Si_r / A_Si)
```

```
Wd = A_Wn * Si_r
```

```
Co_fi = (Xest(4) + 2 * A_Si * A_Wn * Xest(3)) / A_Wn
```

```
a = 2 * A_Si * A_Wn
```

```
b = A_Wn ^ 2
```

```
Ces1 = Kp * (a ^ 2 / b - a * Kd - 1) / (b * Mc)
```

```
Ces2 = Kp * (Kd - a / b) / (b * Mc)
```

```
Ces3 = Kp * (Kd - a / b) / Mc
```

```
Ces4 = Kp / (b * Mc)
```

```
'Tiempo de establecimiento con el criterio del 2%  $t_s = 4/(Si*Wn)$ 
```

```
Tmax = 8 / (A_Si * A_Wn)
```

```
DeltaT = Tmax / NUMmAXpUNTOS
```

```
FdeX1max = 0
```

```
FdeX2max = 0
```

```
FdeX3max = 0
```

```
FdeX4max = 0
```

```
FdeU5max = 0
```

```
Tiempo = 0
```

```
For Ies = 0 To NUMmAXpUNTOS Step 1
```

```

Ang(Ies) = Exp(-A_Si * A_Wn * Tiempo) / Si_r * (-Xest(3) * _
Sin(Wd * Tiempo - Fi_x) + Co_fi * Sin(Wd * Tiempo))

If FdeX3max < Abs(Ang(Ies)) Then
    FdeX3max = Abs(Ang(Ies))
End If

Wang(Ies) = A_Wn * Exp(-A_Si * A_Wn * Tiempo) / Si_r * _
(Xest(3) * Sin(Wd * Tiempo - 2 * Fi_x) - Co_fi * _
Sin(Wd * Tiempo - Fi_x))

If FdeX4max < Abs(Wang(Ies)) Then
    FdeX4max = Abs(Wang(Ies))
End If

Fuerza(Ies) = Kp * (Ang(Ies) + Kd * Wang(Ies))

If FdeU5max < Abs(Fuerza(Ies)) Then
    FdeU5max = Abs(Fuerza(Ies))
End If

If Ies = 0 Then
    C1 = Xest(2) - Ces3 * Ang(0) + Ces4 * Wang(0)
    C2 = Xest(1) - Ces1 * Ang(0) + Ces2 * Wang(0)
End If

Esp(Ies) = Ces1 * Ang(Ies) - Ces2 * Wang(Ies) + _
C1 * Tiempo + C2

If FdeX1max < Abs(Esp(Ies)) Then
    FdeX1max = Abs(Esp(Ies))
End If

Vel(Ies) = Ces3 * Ang(Ies) - Ces4 * Wang(Ies) + C1

If FdeX2max < Abs(Vel(Ies)) Then
    FdeX2max = Abs(Vel(Ies))
End If

Tiempo = Tiempo + DeltaT

' Barra de proceso

If (Ies Mod NumBarra) = 0 Then
    BarraProceso Ies
End If
Next Ies

End Sub

'Subrutina que realiza los calculos para el control
'Proporcional derivativo conociendo Kp y Kd
Private Sub CalculosAccionesPD_KpKd()

```

```

Dim PD_Wn As Single
Dim PD_Si As Single

If Ad_Kp > Gr * Mc Then

PD_Wn = Sqr(Ad_Kp / (Lp * Mc) - Gr / Lp)
PD_Si = Ad_Kp * Ad_Kd / (Lp * Mc * 2 * PD_Wn)

If PD_Si >= 1 Then
    msg = " Los valores propuestos para Kp y Kd del compensador " & _
        "impide la construcción de la acción de control " & _
        " Proporcional Derivativa. Debe hacer cumplir que " & _
        "'Kp * Kd / (Lp * Mc * 2 * Wn) < 1'"
    Mostrar
Else
    PD_Wn = Sqr(Ad_Kp / (Lp * Mc) - Gr / Lp)
    PD_Si = Ad_Kp * Ad_Kd / (Lp * Mc * 2 * PD_Wn)

    CalculosPD PD_Wn, PD_Si, Ad_Kp, Ad_Kd

'Cargo los resultados obtenidos en la ventana de resultados
MuestraResultados_PD PD_Wn, PD_Si, Ad_Kp, Ad_Kd

'Se pone el Tipo de control utilizado que se presentará en el
'momento de observar las gráficas en el tiempo de los diferentes
'estados
lblTipoControl.Caption = "Control PD conociendo Kp y Kd"
frmResultados.SSTabResultados.Tab = 2

End If

Else
msg = " El valor propuesto para Kp del compensador " & _
    "impide la construcción de la acción de control " & _
    " Proporcional Derivativa. Debe hacer cumplir que 'Kp' " & _
    " sea mayor a 'g*M'"
Mostrar
End If
End Sub

'Subrutina que realiza los calculos para el control
'Proporcional Integral y derivativo conociendo Kp , Kd y Ki
Private Sub CalculosAccionesPID()
Dim a As Single, b As Single, c As Single

Dim PID_Wn As Single, PID_Si As Single
Dim R1 As Single, R2 As Single, R3 As Single
Dim R2j As Single, R3j As Single
Dim Wd As Single
Dim Ro As Single

Dim Fb(0 To 2) As Single 'Coeficientes del numerador del

```

```

'estado Angulo en S
Dim Da(0 To 3) As Single 'Coeficientes del Denominador de
'los estado Angulo

Dim Fi_A(1 To 3) As Single 'Coeficientes de la division en
'fracciones parciales del ángulo
Dim FiW_A(1 To 3) As Single 'Coeficientes de la ecuación en el tiempo
'la velocidad angular
Dim FiA_A(1 To 3) As Single 'Coeficientes de la ecuación en el tiempo
'de la aceleración angular
Dim U_A(1 To 2) As Single 'Coeficientes de la ecuación en el tiempo
'de la fuerza
Dim X_A(1 To 3) As Single 'Coeficientes de la ecuación en el tiempo
'del espacio
Dim V_A(1 To 3) As Single 'Coeficientes de la ecuación en el tiempo
'de la velocidad
Dim C1 As Double 'Constantes de la integración en el cálculo
'de la velocidad y el espacio
Dim C2 As Double 'Constantes de la integración en el cálculo
'de la velocidad y el espacio
Dim Y As Single 'Valor de la division en fracciones
'parciales del denominador de Fi(s)
Dim T As Single 'Tiempo
Dim Acel_W As Double 'Para almacenar el valor de la aceleración
'angular como no se lo gráfica no es
'necesario definir un arreglo
'Dim Ies As Integer

```

```

a = A_Kp * A_Kd / (Lp * Mc)
b = (A_Kp / Mc - Gr) / Lp
c = A_Kp * A_Ki / (Lp * Mc)
EcuacionCubica -a, b, -c, R1, R2, R2j, R3, R3j

```

```

'Una raíz real positiva y un par conjugado
If R2j <> 0 Then

```

```

    If R1 > 0 Then
        Y = R1
        PID_Wn = Sqr(c / Y)
        PID_Si = (a - Y) / (2 * PID_Wn)

        If PID_Si < 0 And PID_Si > 1 Then
            Y = 0
            PID_Wn = 0
            PID_Si = 0
            msg = " Los valores de Kp, Kd y Ki del " & _
                "compensador impide la construcción " & _
                "del compensador PID"

```

```

            Mostrar
        End If
    Else

```

```

        msg = " Los valores de Kp, Kd y Ki del " & _
            "compensador impide la construcción del compensador " & _
            "PID"

```

```

    Mostrar
  End If
Else
' El resultado de las raices nos dice que no hay parte imaginaria

'1) Si existe solo una raíz positiva
  If R1 > 0 And R2 <= 0 And R3 <= 0 Then
    Y = R1
    PID_Wn = Sqr(c / Y)
    PID_Si = (a - Y) / (2 * PID_Wn)

    If PID_Si < 0 Or PID_Si > 1 Then
      Y = 0
      PID_Wn = 0
      PID_Si = 0
      msg = " Los valores de Kp, Kd y Ki del " & _
            "compensador impide la construcción " & _
            "del compensador PID"

      Mostrar
    End If

  ElseIf R2 > 0 And R1 <= 0 And R3 <= 0 Then
    Y = R2
    PID_Wn = Sqr(c / Y)
    PID_Si = (a - Y) / (2 * PID_Wn)

    If PID_Si < 0 Or PID_Si > 1 Then
      Y = 0
      PID_Wn = 0
      PID_Si = 0
      msg = " Los valores de Kp, Kd y Ki del " & _
            "compensador impide la construcción " & _
            "del compensador PID"

      Mostrar
    End If

  ElseIf R3 > 0 And R1 <= 0 And R2 <= 0 Then
    Y = R3
    PID_Wn = Sqr(c / Y)
    PID_Si = (a - Y) / (2 * PID_Wn)

    If PID_Si < 0 Or PID_Si > 1 Then
      Y = 0
      PID_Wn = 0
      PID_Si = 0
      msg = " Los valores de Kp, Kd y Ki del " & _
            "compensador impide la construcción " & _
            "del compensador PID"

      Mostrar
    End If

  '2) Si existe dos raices positivas
  ElseIf R1 > 0 And R2 > 0 And R3 <= 0 Then

```

```

Y = R1
PID_Wn = Sqr(c / Y)
PID_Si = (a - Y) / (2 * PID_Wn)

If PID_Si < 0 Or PID_Si > 1 Then
  Y = R2
  PID_Wn = Sqr(c / Y)
  PID_Si = (a - Y) / (2 * PID_Wn)

  If PID_Si < 0 Or PID_Si > 1 Then
    Y = 0
    PID_Wn = 0
    PID_Si = 0
    msg = " Los valores de Kp, Kd y Ki del " & _
          "compensador impide la construcción " & _
          "del compensador PID"
  
```

```

  Mostrar
End If
End If

```

```

ElseIf R1 > 0 And R3 > 0 And R2 <= 0 Then

```

```

  Y = R1
  PID_Wn = Sqr(c / Y)
  PID_Si = (a - Y) / (2 * PID_Wn)

```

```

  If PID_Si < 0 Or PID_Si > 1 Then
    Y = R3
    PID_Wn = Sqr(c / Y)
    PID_Si = (a - Y) / (2 * PID_Wn)

```

```

    If PID_Si < 0 Or PID_Si > 1 Then
      Y = 0
      PID_Wn = 0
      PID_Si = 0
      msg = " Los valores de Kp, Kd y Ki del " & _
            "compensador impide la construcción " & _
            "del compensador PID"
    
```

```

    Mostrar
  End If
End If

```

```

ElseIf R2 > 0 And R3 > 0 And R1 <= 0 Then

```

```

  Y = R2
  PID_Wn = Sqr(c / Y)
  PID_Si = (a - Y) / (2 * PID_Wn)

```

```

  If PID_Si < 0 Or PID_Si > 1 Then
    Y = R3
    PID_Wn = Sqr(c / Y)
    PID_Si = (a - Y) / (2 * PID_Wn)

```

```

    If PID_Si < 0 Or PID_Si > 1 Then
      Y = 0
    
```

```

PID_Wn = 0
PID_Si = 0
msg = " Los valores de Kp, Kd y Ki del " & _
      "compensador impide la construcción " & _
      "del compensador PID"

Mostrar
End If
End If
'3)Si existe tres raíces positivas
ElseIf R1 > 0 And R2 > 0 And R2 > 0 Then
Y = R1
PID_Wn = Sqr(c / Y)
PID_Si = (a - Y) / (2 * PID_Wn)

If PID_Si < 0 Or PID_Si > 1 Then
Y = R2
PID_Wn = Sqr(c / Y)
PID_Si = (a - Y) / (2 * PID_Wn)

If PID_Si < 0 Or PID_Si > 1 Then
Y = R3
PID_Wn = Sqr(c / Y)
PID_Si = (a - Y) / (2 * PID_Wn)

If PID_Si < 0 Or PID_Si > 1 Then
Y = 0
PID_Wn = 0
PID_Si = 0
msg = " Los valores de Kp, Kd y Ki del " & _
      "compensador impide la construcción " & _
      "del compensador PID"

Mostrar
End If
End If
End If
'4)No existe raíces positivas
Else
msg = " Los valores de Kp, Kd y Ki del " & _
      "compensador impide la construcción " & _
      "del compensador PID"

Mostrar
End If
End If
'Chequeo si alguna de las raíces encontradas permite el control PID

If PID_Si <> 0 Then
Ro = PID_Si * PID_Wn
Wd = PID_Wn * Sqr(1 - PID_Si ^ 2)
If (PID_Si * PID_Wn) < Y Then
Tmax = 8 / (PID_Si * PID_Wn)

Else

```

```

Tmax = 8 / Y

End If

DeltaT = Tmax / NUMmAXpUNTOS
'Coeficientes de la ecuación de estado del ángulo
' Fi(s)= Fb2*s^2+Fb1*s+Fb0/(Da3*s^3+Da2*s^2+Da1*s+Da0

Fb(2) = Xest(3)
Fb(1) = Xest(4) + A_Kp * A_Kd * Xest(3) / (Lp * Mc)
Fb(0) = 0

'Division de los coeficientes de las ecuaciones en el tiempo de
'el ángulo , la velocidad angular, y aceleración angular
'Fi(t)=A(1)*Exp(-Y*t)+Exp(-Ro*t)*(A(2)*cos(Wd*t)+A(3)*sen(Wd*t)
'Fi_W(t)=A(1)*Exp(-Y*t)+Exp(-Ro*t)*(A(2)*cos(Wd*t)+A(3)*sen(Wd*t)
'Fi_A(t)=A(1)*Exp(-Y*t)+Exp(-Ro*t)*(A(2)*cos(Wd*t)+A(3)*sen(Wd*t)
  Fi_A(1) = (Fb(2) * Y ^ 2 - Fb(1) * Y + Fb(0)) / _
            ((-Y + Ro) ^ 2 + Wd ^ 2)
  Fi_A(2) = Fb(2) - Fi_A(1)
  Fi_A(3) = (Fb(1) - 2 * Ro * Fi_A(1) - Fi_A(2) * (Ro + Y)) / Wd

  FiW_A(1) = -Fi_A(1) * Y
  FiW_A(2) = -Ro * Fi_A(2) + Fi_A(3) * Wd
  FiW_A(3) = -(Wd * Fi_A(2) + Ro * Fi_A(3))

  FiA_A(1) = -FiW_A(1) * Y
  FiA_A(2) = -Ro * FiW_A(2) + FiW_A(3) * Wd
  FiA_A(3) = -(Wd * FiW_A(2) + Ro * FiW_A(3))

'Coeficientes de la ecuación en el tiempo de la fuerza
'U(t)=A(1)*Fi_A(t) + A(2)*Fi(t)

U_A(1) = -Lp * Mc
U_A(2) = Gr * Mc

'Coeficientes de la ecuación en el tiempo de la velocidad y
'posición
'V(t)=A(1)*Fi_A(t) + A(2)*Fi_W(t) + A(3)*Fi(t) + C1
'X(t)=A(1)*Fi_A(t) + A(2)*Fi_W(t) + A(3)*Fi(t) + C1*t + C2

V_A(1) = -Gr * Lp * Mc / (A_Kp * A_Ki)
V_A(2) = -Lp - Gr * A_Kd / A_Ki
V_A(3) = Gr * (Gr * Mc / (A_Kp * A_Ki) - 1 / A_Ki)

X_A(1) = -V_A(3) * Lp * Mc / (A_Kp * A_Ki)
X_A(2) = V_A(1) - V_A(3) * A_Kd / A_Ki
X_A(3) = V_A(2) + V_A(3) * (Gr * Mc / (A_Kp * A_Ki) _
  - 1 / A_Ki)

'Obtención de los valores en el tiempo para los gráficos

FdeX1max = 0

```

```

FdeX2max = 0
FdeX3max = 0
FdeX4max = 0
FdeU5max = 0
Tiempo = 0

```

```

For Ies = 0 To NUMmAXpUNTOS Step 1

```

```

    Ang(Ies) = Fi_A(1) * Exp(-Y * T) + Exp(-Ro * T) * _
              (Fi_A(2) * Cos(Wd * T) + Fi_A(3) * Sin(Wd * T))

```

```

    If FdeX3max < Abs(Ang(Ies)) Then
        FdeX3max = Abs(Ang(Ies))
    End If

```

```

    Wang(Ies) = FiW_A(1) * Exp(-Y * T) + Exp(-Ro * T) * _
              (FiW_A(2) * Cos(Wd * T) + FiW_A(3) * Sin(Wd * T))

```

```

    If FdeX4max < Abs(Wang(Ies)) Then
        FdeX4max = Abs(Wang(Ies))
    End If

```

```

    Acel_W = FiA_A(1) * Exp(-Y * T) + Exp(-Ro * T) * _
            (FiA_A(2) * Cos(Wd * T) + FiA_A(3) * Sin(Wd * T))

```

```

    Fuerza(Ies) = Acel_W * U_A(1) + Ang(Ies) * U_A(2)

```

```

    If FdeU5max < Abs(Fuerza(Ies)) Then
        FdeU5max = Abs(Fuerza(Ies))
    End If

```

```

    If Ies = 0 Then

```

```

        C1 = Xest(2) - V_A(1) * Acel_W - V_A(2) * Wang(0) - _
            V_A(3) * Ang(0)

```

```

        C2 = Xest(1) - X_A(1) * Acel_W - X_A(2) * Wang(0) - _
            X_A(3) * Ang(0)

```

```

    End If

```

```

    Esp(Ies) = X_A(1) * Acel_W + X_A(2) * Wang(Ies) + _
              X_A(3) * Ang(Ies) + C1 * T + C2

```

```

    If FdeX1max < Abs(Esp(Ies)) Then
        FdeX1max = Abs(Esp(Ies))
    End If

```

```

    Vel(Ies) = V_A(1) * Acel_W + V_A(2) * Wang(Ies) + _
              V_A(3) * Ang(Ies) + C1

```

```

    If FdeX2max < Abs(Vel(Ies)) Then
        FdeX2max = Abs(Vel(Ies))
    End If

```

```

    T = T + DeltaT

```

```

If (Ies Mod NumBarra) = 0 Then
    BarraProceso Ies
End If

Next Ies

'Cargo los resultados obtenidos en la ventana de resultados
MuestroResultados_PID PID_Wn, PID_Si, Ro, Wd, Y

'Pongo el Tipo de control utilizado que se presentará en el
'momento de observar las gráficas en el tiempo de los diferentes
'estados
lblTipoControl.Caption = "Control PID"
frmResultados.SSTabResultados.Tab = 3
Else
    msg = " Los valores de Kp, Kd y Ki del " & _
        "compensador impide la construcción de un" & _
        "control PID que estabilize al sistema - "

    Mostrar
    lblBarraBase.Visible = False
    lblBarraProceso.Visible = False
End If

End Sub

'Solución de una ecuación cúbica de la forma
'X^3 + A2*X^2 + A1*X + A0 = 0
Private Sub EcuacionCubica(A2 As Single, A1 As Single, _
    A0 As Single, R1 As Single, R2 As Single, R2j As Single, _
    R3 As Single, R3j As Single)

Dim E_q As Single
Dim E_r As Single
Dim E_Raiz As Single
Dim E_y As Single
Dim E_z As Single
Dim E_y1 As Single
Dim E_z1 As Single
Dim E_a As Single
Dim E_b As Single
Dim E_M As Single
Dim E_T As Single
Dim M_e As Single, N_e As Single

E_q = -A2 ^ 2 / 3 + A1
E_r = A0 - A2 * A1 / 3 + 2 * A2 ^ 3 / 27
E_Raiz = E_r ^ 2 / 4 + E_q ^ 3 / 27
If E_Raiz > 0 Then
    E_y1 = -E_r / 2 + Sqr(E_Raiz)

    If (E_y1) < 0 Then
        E_y = -(-E_y1) ^ (1 / 3)
    
```

```

Else
  E_y = (E_y1) ^ (1 / 3)
End If

E_z1 = -E_r / 2 - Sqr(E_Raiz)

If (E_z1) < 0 Then
  E_z = -(-E_z1) ^ (1 / 3)
Else
  E_z = (E_z1) ^ (1 / 3)
End If

R1 = E_y + E_z - A2 / 3
R2 = -(E_y + E_z) / 2 - A2 / 3
R2j = (E_y - E_z) * Sqr(3) / 2
R3 = R2
R3j = -R2j
ElseIf E_Raiz = 0 Then
  If E_r < 0 Then
    E_y = (-E_r / 2) ^ (1 / 3)
  Else
    E_y = -(E_r / 2) ^ (1 / 3)
  End If
  R1 = 2 * E_y - A2 / 3
  R2 = -E_y - A2 / 3
  R2j = 0
  R3 = R2
  R3j = -R2j
Else
  E_a = -E_r / 2
  E_b = Sqr(-E_Raiz)
  E_M = Sqr(E_a ^ 2 + E_b ^ 2)

  If E_a = 0 Then
    E_T = PI / 2
  Else
    E_T = Atn(E_b / E_a)
  End If
  M_e = E_M ^ (1 / 3) * Cos((2 * PI + E_T) / 3)
  N_e = E_M ^ (1 / 3) * Sin((2 * PI + E_T) / 3)
  R1 = 2 * M_e - A2 / 3
  R2 = -M_e - N_e * Sqr(3) - A2 / 3
  R3 = -M_e + N_e * Sqr(3) - A2 / 3
  R2j = 0
  R3j = R2j
End If
End Sub

'Subrutina que carga en la pantalla de resultados los datos
'obtenidos del control proporcional derivativo
Private Sub MuestraResultados_PD(PD_Wn_RESUL As Single, PD_Si_RESUL As Single, _
  PD_Kp As Single, PD_Kd As Single)

```

```
Dim PD_MaxPico As Single 'Almacena el valor del máximo sobreimpulso
                          'para el control PD_WnE
```

```
Dim Ts As Single 'Almacena el tiempo de establecimiento para
                 'el PD_WnE
```

```
Dim PD_Ro As Single
```

```
Dim PD_Wd As Single
```

```
With frmResultados
```

```
'Datos para el control PD conociendo Wn y Si o conociendo Kp y Kd
```

```
PD_MaxPico = 100 * Exp(-PD_Si_RESUL * PI / (Sqr(1 - PD_Si_RESUL ^ 2)))
```

```
PD_Ro = PD_Si_RESUL * PD_Wn_RESUL
```

```
PD_Ts = 4 / (PD_Ro)
```

```
PD_Wd = PD_Wn_RESUL * Sqr(1 - PD_Si_RESUL ^ 2)
```

```
.lblSi.Caption = Format(PD_Si_RESUL, "Standard")
```

```
.lblWn.Caption = Format(PD_Wn_RESUL, "Standard") & " rad/s"
```

```
.lblMp.Caption = Format(PD_MaxPico, "Standard") & " %"
```

```
.lblTs.Caption = Format(PD_Ts, "Standard") & " Seg."
```

```
.lblPDkp.Caption = Format(PD_Kp, "Standard")
```

```
.lblPDkd.Caption = Format(PD_Kd, "Standard")
```

```
.lblPoloDominante.Caption = Format(-PD_Ro, "Standard") & _
                             " +/- J " & Format(PD_Wd, "Standard")
```

```
End With
```

```
End Sub
```

```
'Subrutina que carga en la pantalla de resultados los datos
```

```
'obtenidos del control por la red de adelanto de fase
```

```
Private Sub MuestraResultados_red(Red_Si As Single, Red_Wn As Single, _
    Red_Wd As Single, Red_Ro As Single, Red_Fic As Single, _
    Red_Kc As Single, Red_Cc As Single, Red_Pc As Single, _
    Red_Y As Single)
```

```
With frmResultados
```

```
If TipoRedAdelanto = REDaDELbISECTRIZ Then
```

```
.lblRedbSi.Caption = Format(Red_Si, "Standard")
```

```
.lblRedbWn.Caption = Format(Red_Wn, "Standard") & " rad/s"
```

```
.lblRedbMp.Caption = Format(RedbMp * 100, "Standard") & " %"
```

```
.lblRedbTs.Caption = Format(RedbTs, "Standard") & " Seg"
```

```
.lblRedbPoloD.Caption = Format(-Red_Ro, "Standard") & " +/- J " _
    & Format(Red_Wd, "Standard") _
    & " RAIZ 3 = " & Format(-Red_Y, "Standard")
```

```
.lblRedbFic.Caption = Format(Red_Fic * RADaGRADOS, "Standard") & " Grados"
```

```
.lblRedbKc.Caption = Format(Red_Kc, "Standard")
```

```
.lblRedbCc.Caption = Format(Red_Cc, "Standard")
```

```
.lblRedbPc.Caption = Format(Red_Pc, "Standard")
```

```
Else
```

```

.lblRedcSi.Caption = Format(Red_Si, "Standard")
.lblRedcWn.Caption = Format(Red_Wn, "Standard") & " rad/s"
.lblRedcMp.Caption = Format(RedcMp * 100, "Standard") & " %"
.lblRedcTs.Caption = Format(RedcTs, "Standard") & " Seg"
.lblRedcPoloD.Caption = Format(-Red_Ro, "Standard") _
    & " +/- J " & Format(Red_Wd, "Standard") _
    & " RAIZ 3 = " & Format(-Red_Y, "Standard")
.lblRedcFic.Caption = Format(Red_Fic * RADaGRADOs, "Standard") & " Grados"
.lblRedcKc.Caption = Format(Red_Kc, "Standard")
.lblRedcCc.Caption = Format(Red_Cc, "Standard")
.lblRedcPc.Caption = Format(Red_Pc, "Standard")

End If
End With
End Sub

```

```

'Subrutina que carga en la pantalla de resultados los datos
'generales del péndulo
Private Sub MuestraResGenerales()

```

```

With frmResultados

```

```

'Datos Generales para todos los controles

```

```

.lblMc.Caption = Format(Mc, "0.00") & " Kg"
.lblmP.Caption = Format(mP, "0.00") & " Kg"
.lbl2L.Caption = Format(2 * LongP, "0.00") & " m"
.lblLp.Caption = Format(Lp, "0.00") & " m"
.lblJ.Caption = Format(Jp, "#,##0.00000") & " Kg.m^2"
.lblGr.Caption = Format(Gr, "0.00") & " m/s^2"
.lblLongPista.Caption = Format(LongPista, "0.00") & " m"
.lblEspacioInicial.Caption = Format(Xest(1), "0.00") & " m"
.lblVelocidadInicial.Caption = Format(Xest(2), "0.00") & " m/s^2"
.lblAnguloInicial.Caption = Format(Xest(3) * RADaGRADOs, "0.00") & " Grados"
.lblWanguloInicial.Caption = Format(Xest(4) * RADaGRADOs, "0.00") & " Grados/s"

End With
End Sub

```

```

'Subrutina que carga en la pantalla de resultados los datos
'obtenidos del control por Realimentación de estado

```

```

Private Sub MuestraResultados_Realim()

```

```

With frmResultados

```

```

'Datos para el control por Realimentación de estado

```

```

'Se muestra el vector de estado

```

```

.lblVectorK.Caption = "K1 = " & Format(Kest(1), "Standard") & " " & _
    "K2 = " & Format(Kest(2), "Standard") & " " & _
    "K3 = " & Format(Kest(3), "Standard") & " " & _
    "K4 = " & Format(Kest(4), "Standard") & " "

```

'Se muestran los coeficientes del numerador de los estados X(s)
 .lblXsfila1.Caption = Format(b(1, 0), "Standard") & " " & _
 Format(b(1, 1), "Standard") & " " & _
 Format(b(1, 2), "Standard") & " " & _
 Format(b(1, 3), "Standard") & " "

.lblXsfila2.Caption = Format(b(2, 0), "Standard") & " " & _
 Format(b(2, 1), "Standard") & " " & _
 Format(b(2, 2), "Standard") & " " & _
 Format(b(2, 3), "Standard") & " "

.lblXsfila3.Caption = Format(b(3, 0), "Standard") & " " & _
 Format(b(3, 1), "Standard") & " " & _
 Format(b(3, 2), "Standard") & " " & _
 Format(b(3, 3), "Standard") & " "

.lblXsfila4.Caption = Format(b(4, 0), "Standard") & " " & _
 Format(b(4, 1), "Standard") & " " & _
 Format(b(4, 2), "Standard") & " " & _
 Format(b(4, 3), "Standard") & " "

.lblQs.Caption = "S^4 + " & _
 Format(AcDesed(3), "Standard") & " S^3 + " & " " & _
 Format(AcDesed(2), "Standard") & " S^2 + " & " " & _
 Format(AcDesed(1), "Standard") & " S^1 + " & " " & _
 Format(AcDesed(0), "Standard")

'Se muestran los coeficientes del numerador de los estados X(t)
 .lblXtFila1.Caption = Format(0, "Standard") & " " & _
 Format(1, "Standard") & " " & _
 Format(0, "Standard") & " " & _
 Format(0, "Standard") & " "

.lblXtfila2.Caption = Format(-Kest(1) / Mc, "Standard") & " " & _
 Format(-Kest(2) / Mc, "Standard") & " " & _
 Format(-Kest(3) / Mc, "Standard") & " " & _
 Format(-Kest(4) / Mc, "Standard") & " "

.lblXtfila3.Caption = Format(0, "Standard") & " " & _
 Format(0, "Standard") & " " & _
 Format(0, "Standard") & " " & _
 Format(1, "Standard") & " "

.lblXtfila4.Caption = Format(Kest(1) / (Lp * Mc), "Standard") & " " & _
 Format(Kest(2) / (Lp * Mc), "Standard") & " " & _
 Format((Gr * Mc + Kest(3)) / (Lp * Mc), "Standard") & " " & _
 Format(Kest(4) / (Lp * Mc), "Standard") & " "

'Se muestra constantes de la division en fracciones parciales

.lblcA1n.Caption = "POSICION : " & _
 "A11 = " & Format(a(1, 1), "Standard") & " " & _
 "A12 = " & Format(a(1, 2), "Standard") & " " & _
 "A13 = " & Format(a(1, 3), "Standard") & " " & _
 "A14 = " & Format(a(1, 4), "Standard") & " "

```
.lblcA2n.Caption = "VELOCIDAD : " & _
    "A21 = " & Format(a(2, 1), "Standard") & " " & _
    "A22 = " & Format(a(2, 2), "Standard") & " " & _
    "A23 = " & Format(a(2, 3), "Standard") & " " & _
    "A24 = " & Format(a(2, 4), "Standard") & " "

.lblcA3n.Caption = "ANGULO : " & _
    "A31 = " & Format(a(3, 1), "Standard") & " " & _
    "A32 = " & Format(a(3, 2), "Standard") & " " & _
    "A33 = " & Format(a(3, 3), "Standard") & " " & _
    "A34 = " & Format(a(3, 4), "Standard") & " "

.lblcA4n.Caption = "VELOCIDAD ANGULAR : " & _
    "A41 = " & Format(a(4, 1), "Standard") & " " & _
    "A42 = " & Format(a(4, 2), "Standard") & " " & _
    "A43 = " & Format(a(4, 3), "Standard") & " " & _
    "A44 = " & Format(a(4, 4), "Standard") & " "

Select Case CasoRealim
Case CUATROprDIF
    .lblEcuacion.Caption = "Xn(t)= " & _
    "An1 Exp( " & Format(PoloD(1).r, "Standard") & " t ) + " & _
    "An2 Exp( " & Format(PoloD(2).r, "Standard") & " t ) + " & _
    "An3 Exp( " & Format(PoloD(3).r, "Standard") & " t ) + " & _
    "An4 Exp( " & Format(PoloD(4).r, "Standard") & " t )"

Case DOSprIGDOSprDIF
    .lblEcuacion.Caption = "Xn(t)= " & _
    "( An1 + An2 t ) Exp( " & Format(PoloD(1).r, "Standard") & " t ) + " & _
    "An3 Exp( " & Format(PoloD(3).r, "Standard") & " t ) + " & _
    "An4 Exp( " & Format(PoloD(4).r, "Standard") & " t )"

Case TRESprIGUNprDIF
    .lblEcuacion.Caption = "Xn(t)= " & _
    "( An1 + An2 t + An3 t^2/2 ) Exp( " & _
    & Format(PoloD(1).r, "Standard") & " t ) + " & _
    "An4 Exp( " & Format(PoloD(4).r, "Standard") & " t )"

Case CUATROprIG
    .lblEcuacion.Caption = "Xn(t)= " & _
    "( An1 + An2 t + An3 t^2/2 + An4 t^3/6 ) Exp( " & _
    Format(PoloD(1).r, "Standard") & " t )"

Case DOSprIGDOSprIG
    .lblEcuacion.Caption = "Xn(t)= " & _
    "( An1 + An2 t ) Exp( " & Format(PoloD(1).r, "Standard") & " t ) + " & _
    "( An3 + An4 t ) Exp( " & Format(PoloD(3).r, "Standard") & " t )"

Case DOSpcDIF
    .lblEcuacion.Caption = "Xn(t)= " & _
    "Exp( " & Format(PoloD(1).r, "Standard") & " t ) * " & _
    "{ An1 cos( " & Format(PoloD(1).I, "Standard") & " t ) + " & _
    " An2 sen( " & Format(PoloD(1).I, "Standard") & " t ) } + " & _
    "Exp( " & Format(PoloD(3).r, "Standard") & " t ) * " & _
    "{ An3 cos( " & Format(PoloD(3).I, "Standard") & " t ) + " & _
    " An4 sen( " & Format(PoloD(3).I, "Standard") & " t ) }"
```

Case DOSpcIG

```
.lblEcuacion.Caption = "Xn(t)= " & _
"Exp( " & Format(PoloD(1).r, "Standard") & " t) * " & _
"{ An1 cos( " & Format(PoloD(1).I, "Standard") & " t) + " & _
" An2 sen( " & Format(PoloD(1).I, "Standard") & " t) + " & _
" An3 t/" & Format(2 * PoloD(1).I, "Standard") & _
" sen( " & Format(PoloD(1).I, "Standard") & " t) + " & _
" An4 /" & Format(2 * (PoloD(1).I) ^ 3, "Standard") & _
" * [ sen( " & Format(PoloD(1).I, "Standard") & " t) - " & _
Format(PoloD(1).I, "Standard") & " t " & _
"cos( " & Format(PoloD(1).I, "Standard") & " t) ] }"
```

Case UNpcDOSprDIF

```
.lblEcuacion.Caption = "Xn(t)= " & _
"An1 Exp( " & Format(PoloD(3).r, "Standard") & " t) + " & _
"An2 Exp( " & Format(PoloD(4).r, "Standard") & " t) ÷ " & _
"Exp( " & Format(PoloD(1).r, "Standard") & " t) * " & _
"{ An3 cos( " & Format(PoloD(1).I, "Standard") & " t) + " & _
" An4 sen( " & Format(PoloD(1).I, "Standard") & " t) }"
```

Case UNpcDOSprIG

```
.lblEcuacion.Caption = "Xn(t)= " & _
"( An1 + An2 t) Exp( " & Format(PoloD(3).r, "Standard") & " t) + " & _
"Exp( " & Format(PoloD(1).r, "Standard") & " t) * " & _
"{ An3 cos( " & Format(PoloD(1).I, "Standard") & " t) + " & _
" An4 sen( " & Format(PoloD(1).I, "Standard") & " t) }"
```

End Select

End With

End Sub

'Subrutina que carga en la pantalla de resultados los datos
'obtenidos del control proporcional Integral y derivativo

```
Private Sub MuestraResultados_PID(PIDWn As Single, _
    PIDSi As Single, PID_Ro As Single, _
    PID_Wd As Single, PIDy As Single)
Dim PID_MaxPico As Single 'Almacena el valor del máximo sobreimpulso
'para el control PD_WnE
Dim Ts As Single 'Almacena el tiempo de establecimiento para
'el PD_WnE
```

With frmResultados

'Datos para el control PD conociendo Wn y Si o conociendo Kp y Kd

```
PID_MaxPico = 100 * Exp(-PID_Ro * PI / PID_Wd)
```

```
PD_Ts = 4 / (PID_Ro)
```

```
.lblPID_Si.Caption = Format(PIDSi, "Standard")
```

```
.lblPID_Wn.Caption = Format(PIDWn, "Standard") & " rad/s"
```

```

.lblPID_Mp.Caption = Format(PID_MaxPico, "Standard") & " %"
.lblPID_Ts.Caption = Format(PD_Ts, "Standard") & " Seg."

.lblPIDPoloD.Caption = Format(-PID_Ro, "Standard") & " +/- J " & Format(PID_Wd, "Standard") _
& " POLO 3 = " & Format(-PIDy, "Standard")

.lblPIDkp.Caption = Format(A_Kp, "Standard")
.lblPIDkd.Caption = Format(A_Kd, "Standard")
.lblPIDki.Caption = Format(A_Ki, "Standard")

```

```

End With
End Sub

```

```

'Para Realimentación de estado
'Cálculo del polo con parte real máxima absoluta que
'servirá para el cálculo del tiempo de simulación Tmax
Private Function PoloMaximo() As Single

```

```

If (PoloD(1).r >= PoloD(2).r) And (PoloD(1).r >= PoloD(3).r) _
And (PoloD(1).r >= PoloD(4).r) Then

```

```

    PoloMaximo = PoloD(1).r

```

```

ElseIf (PoloD(2).r >= PoloD(1).r) And (PoloD(2).r >= PoloD(3).r) _
And (PoloD(2).r >= PoloD(4).r) Then

```

```

    PoloMaximo = PoloD(2).r

```

```

ElseIf (PoloD(3).r >= PoloD(1).r) And (PoloD(3).r >= PoloD(2).r) _
And (PoloD(3).r >= PoloD(4).r) Then

```

```

    PoloMaximo = PoloD(3).r

```

```

Else

```

```

    PoloMaximo = PoloD(4).r

```

```

End If

```

```

End Function

```

```

'Animación de la varilla y carrito

```

```

'      |(X2,Y2)
'      |
'      |
'      |(X1,Y1)
'      |-----|
'      |_____|
'      @   @

```

```

Static Sub TimerSim_Timer()

```

```

Dim Y2 As Single

```

```

Dim Y1 As Single

```

```

'Lectura de la velocidad de simulación

```

```

If CambioVelocidad = 1 Then ' Si existe algún cambio
  If HScrollVelocidad.Value = 0 Then
    TimerSim.Interval = 150
    IntervaloSim = 1
  ElseIf HScrollVelocidad.Value > 0 Then
    'Mas Rápida la simulación alcanzando un valor máximo de 30
    TimerSim.Interval = 150
    IntervaloSim = HScrollVelocidad.Value
  Else
    'Mas lenta la simulación alcanzando un valor minimo de -30
    IntervaloSim = 1
    TimerSim.Interval = 150 + 30 * Abs(HScrollVelocidad.Value)
  End If
  CambioVelocidad = 0
Else
  End If
'Fin de la Lectura de la velocidad de simulación

If Esp(Tsim) >= LongPista / 2 Then
  LinePendulo.Y2 = LinePendulo.Y1
  LinePendulo.X1 = LongPista / 2
  LinePendulo.X2 = LinePendulo.X1 + Abs(Y1 - Y2)

  ImgCarro.Left = LinePendulo.X1 - ImgCarro.Width / 2
  ImgCarro.Top = LinePendulo.Y1
  Tsim = NUMmAXpUNTOS + 1

ElseIf Esp(Tsim) <= -LongPista / 2 Then
  LinePendulo.Y2 = LinePendulo.Y1
  LinePendulo.X1 = -LongPista / 2
  LinePendulo.X2 = LinePendulo.X1 - Abs(Y1 - Y2)

  ImgCarro.Left = LinePendulo.X1 + ImgCarro.Width / 2
  ImgCarro.Top = LinePendulo.Y1

  Tsim = NUMmAXpUNTOS + 1
Else
  Y1 = LinePendulo.Y1
  Y2 = LinePendulo.Y2

  LinePendulo.X1 = Esp(Tsim)
  LinePendulo.X2 = Esp(Tsim) + Abs(Y1 - Y2) * Sin(Ang(Tsim))
  ImgCarro.Left = LinePendulo.X1 - ImgCarro.Width / 2
  ImgCarro.Top = LinePendulo.Y1
  Valores_Analogos
End If

Tsim = Tsim + IntervaloSim

If Tsim > NUMmAXpUNTOS Then
  TimerSim.Enabled = False
  cmdCongelar.Caption = "CONGELAR"
  Beep
End If

```

End Sub

'COMANDOS VISIBLES PARA LA SIMULACION DINAMICA

Private Sub ComanVisi_Sim()

picGraficos.Visible = True

lblTitulo.Visible = True

lblTipoControl.Visible = True

LinePendulo.Visible = True

ImgCarro.Visible = True

FrameValores.Visible = True 'Cuadro que contiene las variables de ángulo
'Velocidad angular y fuerza

FrameValores_L.Visible = True 'Cuadro que contiene las variables de posición
'Velocidad y tiempo

FrameGraficos.Visible = True 'Gráfico que contiene las curvas de las
'variables de entrada y salida para simulación

FrameVelocidad.Visible = True 'Cuadro que contiene el Scroll de la velocidad
'de simulación

cmdCongelar.Visible = True 'Botón que congela o descongela la imagen
'de simulación

lblEjeXtext.Visible = False

lblEjeYtext.Visible = False

LblEjeX.Visible = False

lblEjeY.Visible = False

lblEscalaEjeX.Visible = False

lblEscalaEjeY.Visible = False

lblCurva.Visible = False 'Cuadros indicadores del tipo de curva

ShapeCurva.Visible = False 'Se hacen invisibles

End Sub

'Barra de proceso para los cálculos

Private Sub BarraProceso(Num1 As Integer)

Dim Num2 As Single

Dim Num3 As Single

Num3 = lblBarraBase.Width

If Num1 = 0 Then

lblBarraBase.Left = frmAlternativa.Width / 2 - lblBarraBase.Width / 2

lblBarraProceso.Left = lblBarraBase.Left

lblBarraBase.Top = frmAlternativa.Height / 2 - lblBarraBase.Height / 2

lblBarraProceso.Top = lblBarraBase.Top

lblBarraBase.Visible = True

lblBarraProceso.Visible = True

End If

```

Num2 = Num1 / NUMmAXpUNTOS * 100
lblBarraBase.Caption = Num2 & " %"
lblBarraProceso.Width = Num2 / 100 * Num3

```

```

If Num1 = NUMmAXpUNTOS Then
    lblBarraBase.Visible = False
    lblBarraProceso.Visible = False
End If

```

```
End Sub
```

'Subrutina que realiza los cálculos para el control fuzzy,

```

Private Sub CalculosControlFuzzy()
'Variables y constantes para la ubicación de los conjuntos fuzzy
Dim TriA_f(1 To NUMMcONFUZZY) As Single 'Ubicación del inicio del conjunto
    'fuzzy del ángulo
Dim TriB_f(1 To NUMMcONFUZZY) As Single 'Ubicación del fin del conjunto
    'fuzzy del ángulo
Dim Cen_f(1 To NUMMcONFUZZY) As Single 'Centroide del conjunto fuzzy del ángulo

```

```

Dim TriA_w(1 To NUMMcONFUZZY) As Single 'Ubicación del inicio del conjunto
    'fuzzy de la velocidad angular
Dim TriB_w(1 To NUMMcONFUZZY) As Single 'Ubicación del fin del conjunto
    'fuzzy de la velocidad angular
Dim Cen_w(1 To NUMMcONFUZZY) As Single 'Centroide del conjunto fuzzy de
    'la velocidad angular

```

```

Dim TriA_u(1 To NUMMcONFUZZY) As Single 'Ubicación del inicio del conjunto
    'fuzzy de la fuerza
Dim TriB_u(1 To NUMMcONFUZZY) As Single 'Ubicación del fin del conjunto
    'fuzzy de la fuerza
Dim Cen_u(1 To NUMMcONFUZZY) As Single 'Centroide del conjunto fuzzy de
    'la fuerza

```

```

Dim RangoMin_f As Single 'Valor mínimo del rango de el ángulo
Dim Solapo_f As Single 'Solapamiento de los conjuntos fuzzy del ángulo
Dim DisOC_f As Single 'Distancia al centro del conjunto fuzzy del ángulo
Dim Base_f As Single 'Base de cada conjunto fuzzy del ángulo

```

```

Dim RangoMin_w As Single 'Valor mínimo del rango de la velociad angular
Dim Solapo_w As Single 'Solapamiento de los conjuntos fuzzy de
    ' la velociad angular
Dim DisOC_w As Single 'Distancia al centro del conjunto fuzzy
    ' la velociad angular
Dim Base_w As Single 'Base del cada conjunto fuzzy de la velociad angular

```

```

Dim RangoMin_u As Single 'Valor mínimo del rango de la fuerza
Dim Solapo_u As Single 'Solapamiento de los conjuntos fuzzy de la fuerza
Dim DisOC_u As Single 'Distancia al centro del conjunto fuzzy de la fuerza

```

```

Dim Base_u As Single      'Base del cada conjunto fuzzy de la fuerza

Dim I As Integer         'Variable usada para lazos de repetición
Dim J As Integer         'Variable usada para lazos de repetición
Dim K As Integer         'Variable usada para el lazo de repetición
                        'y almacenamiento de valores de los estados
'Dim Ies As Integer      'Variable para la simulación
Dim Varaux As Single     'Variable auxiliar
Dim VarAux1 As Single

'Variables y constantes para la determinación del grado de pertenencia
Dim Pendiente_f As Single 'Valor de la pendiente de el conjunto fuzzy
                        'del ángulo

Dim Pendiente_w As Single 'Valor de la pendiente de el conjunto fuzzy
                        'de la velocidad angular
Dim Pendiente_u As Single 'Valor de la pendiente de el conjunto fuzzy
                        'de la fuerza

Dim DatoFuzzy_f As Single 'Valor ingresado a fuzzyficar del ángulo
Dim DatoFuzzy_w As Single 'Valor ingresado a fuzzyficar de la velocidad
                        ' angular
Dim U_DeFuzzy As Single  'Valor encontrado de la defuzzyficación

Dim Rang(1 To NUMcONFUZZY) As Single 'Matriz de los valores de pertenencia
                        'asociados a la variable ángulo
Dim Rwang(1 To NUMcONFUZZY) As Single 'Matriz de los valores de pertenencia
                        'asociados a la variable velocidad angular
Dim Rang_Rwang As Single  'Valor de pertenencia inferido

Dim Rfuerza(0 To NUMcONFUZZY) As Single 'Matriz de los valores de pertenencia
                        'asociados a la variable de salida
                        'fuzzyficada.
Dim ConteoRastreo(1 To NUMcONFUZZY, 1 To NUMcONFUZZY)

'Constantes para la simulación
Dim TB1 As Single        'Constante del numerador de el ángulo
Dim Tc(0 To 2) As Single 'Constantes del denominador de el ángulo y
                        'velocidad angular
Dim Xb1 As Single        'Constantes del numerador de la posición
Dim Xc(0 To 2) As Single 'Constantes del denominador de la posición

'Método del centroide

Dim CMx As Single        'Variable que define la ubicación del centroide

'Constantes para definir si los valores de ángulo o velocidad angular
'han salido del rango preestablecido
Const NORMaL = 0         'Cuando el valor de las entradas de ángulo
                        'y velocidad angular están dentro del rango
Const ANGULOmENORrANGO = 1 'Cuando el ángulo es menor al valor mínimo
                        'del rango.

```

```

Const ANGULOmAYORrANGO = 4 'Cuando el ángulo es mayor al valor máximo
                             'del rango.
Const WANGmENORrANGO = 5 'Cuando la velocidad angular es menor
                             'al valor mínimo del rango.
Const WANGmAYORrANGO = 7 'Cuando la velocidad angular es mayor
                             'al valor mínimo del rango.
Dim RangosEntradas As Integer 'Variable que asume los valores anteriores
                             'si las entradas estan dentro o fuera del
                             'rango

Const DENTROdErANGO = 0 'Constante que indica que no se ha
                             'presentado ningún caso de fuera de rango
Const FUERAdErANGO = 1 'Constante que indica que se ha presentado por
                             'lo menos un caso de fuera de rango

Dim EnRango As Integer 'Variable de chequeo de fuera de rango,
                             'puede asumir uno de los dos
                             'valores anteriores

Const SicONTINUAR = 0 'Constante que indica que si existe un valor
                             'de fuera de rango se debe continuar
                             'con los calculos
Const NocONTINUAR = 1 'Constante que indica que si existe un valor
                             'de fuera de rango se debe continuar
                             'con los calculos
Dim SeContinua As Integer 'Variable que contiene los valores anteriores
                             '

'Constantes para la simulación del ángulo
TB1 = -6 * Tfuzzy ^ 2
Tc(0) = 12 * Lp * Mc - Gr * Mc * Tfuzzy ^ 2
Tc(1) = -24 * Lp * Mc - 10 * Gr * Mc * Tfuzzy ^ 2
Tc(2) = Tc(0)

'Constantes para la simulación de la posición
Xb1 = Tfuzzy ^ 2
Xc(0) = 2 * Mc
Xc(1) = -4 * Mc
Xc(2) = Xc(0)

'Inicialización de valores para los lazos de repetición
Ang(0) = Xest(3)
Ang(-1) = Xest(3) + Xest(4) * Tfuzzy
Fuerza(0) = 0
Fuerza(-1) = 0
Wang(0) = Xest(4)
Esp(0) = Xest(1)
Esp(-1) = Xest(1) + Xest(2) * Tfuzzy
Vel(0) = Xest(2)

'Reseteo de la matriz de rastreo

For I = 1 To NUMcONFUZZY

```

```

For J = 1 To NUMMcONFUZZY
  ConteoRastreo(I, J) = 0
Next J
Next I
'Inicialización de variables de fuera de rango
RangosEntradas = NORMaL
EnRango = DENTROdErANGO
SeContinua = SicONTINUAR

'UBICACION DE LOS CONJUNTOS FUZZY

'Ubicación de los conjuntos fuzzy ángulo

Solapo_f = 0.5
RangoMin_f = -Rango_f / 2

Base_f = Rango_f / (NUMMcONFUZZY - (NUMMcONFUZZY - 1) * Solapo_f)
DisOC_f = Base_f / 2
Varaux = RangoMin_f

For I = 1 To NUMMcONFUZZY
  TriA_f(I) = Varaux
  Cen_f(I) = TriA_f(I) + DisOC_f
  TriB_f(I) = TriA_f(I) + 2 * DisOC_f
  Varaux = Varaux + Base_f - Base_f * Solapo_f
Next I

'Ubicación de los conjuntos fuzzy de la velocidad angular

Solapo_w = 0.5
RangoMin_w = -Rango_w / 2

Base_w = Rango_w / (NUMMcONFUZZY - (NUMMcONFUZZY - 1) * Solapo_w)
DisOC_w = Base_w / 2
Varaux = RangoMin_w

For I = 1 To NUMMcONFUZZY
  TriA_w(I) = Varaux
  Cen_w(I) = TriA_w(I) + DisOC_w
  TriB_w(I) = TriA_w(I) + 2 * DisOC_w
  Varaux = Varaux + Base_w - Base_w * Solapo_w
Next I

'Ubicación de los conjuntos fuzzy de la fuerza

Solapo_u = 0.5

RangoMin_u = -Rango_u / 2 'Mínimo valor -20 Newton a 20 Newton

Base_u = Rango_u / (NUMMcONFUZZY - (NUMMcONFUZZY - 1) * Solapo_u)
DisOC_u = Base_u / 2
Varaux = RangoMin_u

For I = 1 To NUMMcONFUZZY
  TriA_u(I) = Varaux

```

```

Cen_u(I) = TriA_u(I) + DisOC_u
TriB_u(I) = TriA_u(I) + 2 * DisOC_u
Varaux = Varaux + Base_u - Base_u * Solapo_u
Next I

'Pendientes de los números fuzzy
Pendiente_f = 1 / DisOC_f
Pendiente_w = 1 / DisOC_w
Pendiente_u = 1 / DisOC_u

'Determina el tiempo máximo a dibujar
Tmax = TsimulaciOn
DeltaT = Tfuzzy
FdeX1max = 0
FdeX2max = 0
FdeX3max = 0
FdeX4max = 0
FdeU5max = 0
Ies = 0
'Datos para la fuzzyficación
'DatoFuzzy_f = en radianes
'DatoFuzzy_w = en radianes/segundos
'UDeFuzzy= Valor de la fuerza en Newton a aplicar dato de salida

For K = 0 To NUMmAXpUNTOS Step 1
  For I = 1 To NUMcONfUZZY 'Inicialización del vector fuerza
    Rfuerza(I) = 0
  Next I

  .RangosEntradas = NORMaL
  DatoFuzzy_f = Ang(K)
  DatoFuzzy_w = Wang(K)

'Datos de prueba
'DatoFuzzy_f = 3.5 * GRADOSA RAD
'DatoFuzzy_w = -4.3 * GRADOSA RAD

'Determinación del grado de pertenencia para el ángulo
If DatoFuzzy_f <= RangoMin_f Then

  If EnRango <> FUERAdErANGO Then
    'Si el dato del ángulo es menor que el rango
    msg = "EL RANGO DEL ANGULO DEBE INCREMENTARSE " _
      & "DESEA CONTINUAR" ' Define message.
    Estilo = vbYesNo + vbCritical + vbDefaultButton2 ' Define botones
    Titulo = "ERROR DE RANGO" ' Define Titulo.
    Respuesta = MsgBox(msg, Estilo, Titulo) 'Muestro mensaje

    If Respuesta = vbYes Then ' Presionó el boton YES
      SeContinua = SicONTINUAR
      EnRango = FUERAdErANGO
    Else ' Presionó el boton NO

```

```

        SeContinua = NocONTINUAR
        EnRango = FUERAdErANGO
    End If
Else
End If
RangosEntradas = ANGULOmENORrANGO

ElseIf DatoFuzzy_f >= (Rango_f + RangoMin_f) Then

'Si el dato del ángulo es mayor al rango
If EnRango <> FUERAdErANGO Then
    'Si el dato del ángulo es menor que el rango
    msg = "EL RANGO DEL ANGULO DEBE INCREMENTARSE " _
        & "DESEA CONTINUAR"          ' Define message.
    Estilo = vbYesNo + vbCritical + vbDefaultButton2 ' Define botones
    Titulo = "ERROR DE RANGO" ' Define Titulo.
    Respuesta = MsgBox(msg, Estilo, Titulo) 'Muestro mensaje

    If Respuesta = vbYes Then          ' Presionó el boton YES
        SeContinua = SicONTINUAR
        EnRango = FUERAdErANGO
    Else                               ' Presionó el boton NO
        SeContinua = NocONTINUAR
        EnRango = FUERAdErANGO
    End If
Else
End If

RangosEntradas = ANGULOmAYORrANGO

Else
For I = 1 To NUMcONFUZZY
    If Abs(Cen_f(I) - DatoFuzzy_f) >= DisOC_f Then
        Rang(I) = 0
    ElseIf Abs(Cen_f(I) - DatoFuzzy_f) = 0 Then
        Rang(I) = 1
    Else
        Varaux = DisOC_f - Abs(Cen_f(I) - DatoFuzzy_f)
        Rang(I) = Varaux * Pendiente_f
    End If
Next I
End If

'Chequeo si se termina los cálculos
Select Case SeContinua
Case NocONTINUAR
    lblBarraBase.Visible = False    'Desaparezco la barra de
    lblBarraProceso.Visible = False 'proceso

    Exit For                        'Termina el lazo de control

Case SicONTINUAR
End Select

'Determinación del grado de pertenencia para la velocidad angular

```

```

If DatoFuzzy_w <= RangoMin_w Then
  'Si el dato del ángulo es mayor al rango
  If EnRango <> FUERAdErANGO Then
    'Si el dato del ángulo es menor que el rango
    msg = "EL RANGO DE LA VELOCIDAD ANGULAR DEBE INCREMENTARSE " _
      & "DESEA CONTINUAR" ' Define message.
    Estilo = vbYesNo + vbCritical + vbDefaultButton2 ' Define botones
    Titulo = "ERROR DE RANGO" ' Define Titulo.
    Respuesta = MsgBox(msg, Estilo, Titulo) 'Muestro mensaje

    If Respuesta = vbYes Then ' Presionó el boton YES
      SeContinua = SicCONTINUAR
      EnRango = FUERAdErANGO
    Else ' Presionó el boton NO
      SeContinua = NocCONTINUAR
      EnRango = FUERAdErANGO
    End If
  Else
  End If

  RangosEntradas = WANGmENORrANGO

ElseIf DatoFuzzy_w >= (Rango_w + RangoMin_w) Then
  'Si el dato del ángulo es mayor al rango

  If EnRango <> FUERAdErANGO Then
    'Si el dato del ángulo es menor que el rango
    msg = "EL RANGO DE LA VELOCIDAD ANGULAR DEBE INCREMENTARSE " _
      & "DESEA CONTINUAR" ' Define message.
    Estilo = vbYesNo + vbCritical + vbDefaultButton2 ' Define botones
    Titulo = "ERROR DE RANGO" ' Define Titulo.
    Respuesta = MsgBox(msg, Estilo, Titulo) 'Muestro mensaje

    If Respuesta = vbYes Then ' Presionó el boton YES
      SeContinua = SicCONTINUAR
      EnRango = FUERAdErANGO
    Else ' Presionó el boton NO
      SeContinua = NocCONTINUAR
      EnRango = FUERAdErANGO
    End If
  Else
  End If

  RangosEntradas = WANGmAYORrANGO

Else
  For I = 1 To NUMcONFUZZY

    If Abs(Cen_w(I) - DatoFuzzy_w) >= DisOC_w Then
      Rwang(I) = 0
    ElseIf Abs(Cen_w(I) - DatoFuzzy_w) = 0 Then
      Rwang(I) = 1
    Else
      Varaux = DisOC_w - Abs(Cen_w(I) - DatoFuzzy_w)

```

```

        Rwang(I) = Varaux * Pendiente_w
    End If
Next I
End If

'Chequeo si se termina los cálculos
Select Case SeContinua
    Case NocCONTINUAR
        lblBarraBase.Visible = False    'Desaparezo la barra de
        lblBarraProceso.Visible = False 'proceso

        Exit For    'Termina el lazo de control
    Case SicCONTINUAR
End Select

Select Case RangosEntradas

Case NORMaL
    'Algoritmo de inferencia

    For I = 1 To NUMcONFUZZY
        If Rang(I) <> 0 Then
            For J = 1 To NUMcONFUZZY
                If Rwang(J) <> 0 Then
                    If Rang(I) > Rwang(J) Then
                        Rang_Rwang = Rwang(J)
                    Else
                        Rang_Rwang = Rang(I)
                    End If

                    Varaux = BaseConoc(J, I) 'Define que número
                        'fuzzy se está tratando según
                        'la base de conocimiento

                    'Rfuerza(VarAux) = Rfuerza(VarAux) + Rang_Rwang
                        'Defino que reglas
                        'que se repiten se acumulan
                    If Rfuerza(Varaux) < Rang_Rwang Then
                        Rfuerza(Varaux) = Rang_Rwang
                    End If

                    ConteoRastreo(J, I) = ConteoRastreo(J, I) + 1
                Else
                    End If
            Next J
        Else
            End If
    Next I

'Método del centroide
'Centroide=CMx=[W]1xn X [C]nx1/{[W]1xn X [L]nx1}
'[W] = Rfuerza(I)
'[C]= Cen_u(I)
'[L]= Matriz columna de unos

```

```

Varaux = 0
VarAux1 = 0

For I = 1 To NUMCONFUZZY Step 1
  Varaux = Varaux + Rfuerza(I) * Cen_u(I)
Next I

For I = 1 To NUMCONFUZZY Step 1
  VarAux1 = VarAux1 + Rfuerza(I)
Next I

If VarAux1 = 0 Then
  msg = "El metodo de fuzzy ha obtenido un cero en el " _
    & "denominador CMx= infinito. Cambie los rangos)"

  Estilo = vbOKOnly + vbCritical ' Define botones
  Titulo = "Error" ' Define titulo
  MsgBox msg, Estilo, Titulo

  VarAux1 = 0.01
  SeContinua = NocONTINUAR ' Para el proceso de Cálculo
Else
  CMx = Varaux / VarAux1
End If

Case ANGULOmENORrANGO, WANGmENORrANGO, ANGULOmENORrANGO +
WANGmENORrANGO, _
  ANGULOmENORrANGO + WANGmAYORrANGO

  CMx = RangoMin_u

Case ANGULOmAYORrANGO, WANGmAYORrANGO, ANGULOmAYORrANGO +
WANGmENORrANGO, _
  ANGULOmAYORrANGO + WANGmAYORrANGO

  CMx = Rango_u + RangoMin_u

End Select

'Chequeo si se termina los cálculos
Select Case SeContinua
  Case NocONTINUAR
    lblBarraBase.Visible = False 'Desaparezco la barra de
    lblBarraProceso.Visible = False 'proceso

    Exit For 'Termina el lazo de control
  Case SicONTINUAR
End Select

Fuerza(K) = CMx
Ies = K + 1

Ang(Ies) = (TB1 * Fuerza(Ies - 1) + TB1 * Fuerza(Ies - 2) _

```

$$- Tc(1) * Ang(Ies - 1) - Tc(2) * Ang(Ies - 2)) / Tc(0)$$

If FdeX3max < Abs(Ang(Ies)) Then

FdeX3max = Abs(Ang(Ies))

End If

'Aproximación de la derivada del ángulo

Wang(Ies) = (Ang(Ies) - Ang(Ies - 1)) / Tfuzzy

If FdeX4max < Abs(Wang(Ies)) Then

FdeX4max = Abs(Wang(Ies))

End If

Esp(Ies) = (Xb1 * Fuerza(Ies - 1) + Xb1 * Fuerza(Ies - 2) -
Xc(1) * Esp(Ies - 1) - Xc(2) * Esp(Ies - 2)) / Xc(0)

If FdeX1max < Abs(Esp(Ies)) Then

FdeX1max = Abs(Esp(Ies))

End If

'Aproximación de la derivada del espacio

Vel(Ies) = (Esp(Ies) - Esp(Ies - 1)) / Tfuzzy

If FdeX2max < Abs(Vel(Ies)) Then

FdeX2max = Abs(Vel(Ies))

End If

'Cálculo del valor máximo de la fuerza

If FdeU5max < Abs(Fuerza(K)) Then

FdeU5max = Abs(Fuerza(K))

End If

' Barra de proceso

If (K Mod NumBarra) = 0 Then

BarraProceso K

End If

Next K

'Cargo los resultados obtenidos en la ventana de resultados

MuestraResultados_fuzzy Cen_f(), RangoMin_f, Cen_w(),
RangoMin_w, Cen_u(), RangoMin_u

'Paso los datos del contador a la matriz de resultados

With frmResultados

K = 0

For I = 1 To NUMcONFUZZY

For J = 1 To NUMcONFUZZY

If ConteoRastreo(I, J) = 0 Then

.lblConocimiento(K).Caption = ""

Else

.lblConocimiento(K).Caption = ConteoRastreo(I, J)

End If

```

    K = K + 1
  Next J
Next I
End With
'Se pone el Tipo de control utilizado que se presentará en el
'momento de observar las gráficas en el tiempo de los diferentes
'estados
  lblTipoControl.Caption = "Control Fuzzy"
  frmResultados.SSTabResultados.Tab = 5
End Sub

```

'CONTROL FUZZY

'FUNCION PARA CONVERTIR LA BASE DE CONOCIMIENTO EN NUMEROS

Private Function LectBaseConoc(VariaLinguistica As String) As Integer

```

  If VariaLinguistica = "MN" Then
    LectBaseConoc = 1
  ElseIf VariaLinguistica = "PN" Then
    LectBaseConoc = 2
  ElseIf VariaLinguistica = "C" Then
    LectBaseConoc = 3
  ElseIf VariaLinguistica = "PP" Then
    LectBaseConoc = 4
  ElseIf VariaLinguistica = "MP" Then
    LectBaseConoc = 5
  End If

```

End Function

'Subrutina que carga en la pantalla de resultados los datos
'obtenidos en el contro Fuzzy

```

Private Sub MuestroResultados_fuzzy(F_Cen() As Single, F_min As Single, _
  W_Cen() As Single, W_min As Single, _
  U_Cen() As Single, U_min As Single)

```

With frmResultados

' Datos de los conjuntos fuzzy del ángulo

```

  .lblA_f.Caption = Format(F_min * RADaGRADOS, "Standard") & " Grados"
  .lblC1_f.Caption = Format(F_Cen(1) * RADaGRADOS, "Standard") & " Grados"
  .lblC2_f.Caption = Format(F_Cen(2) * RADaGRADOS, "Standard") & " Grados"
  .lblC4_f.Caption = Format(F_Cen(4) * RADaGRADOS, "Standard") & " Grados"
  .lblC5_f.Caption = Format(F_Cen(5) * RADaGRADOS, "Standard") & " Grados"
  .lblB_f.Caption = Format((Rango_f + F_min) * RADaGRADOS, "Standard") & " Grados"
  .lblRango_f.Caption = Format(Rango_f * RADaGRADOS, "Standard") & " Grados"

```

' Datos de los conjuntos fuzzy de la velocidad angular

```

  .lblA_w.Caption = Format(W_min * RADaGRADOS, "Standard") & " Grados/Seg."
  .lblC1_w.Caption = Format(W_Cen(1) * RADaGRADOS, "Standard") & " Grados/Seg."

```

```
.lblC2_w.Caption = Format(W_Cen(2) * RADaGRADOs, "Standard") & " Grados/Seg."
.lblC4_w.Caption = Format(W_Cen(4) * RADaGRADOs, "Standard") & " Grados/Seg."
.lblC5_w.Caption = Format(W_Cen(5) * RADaGRADOs, "Standard") & " Grados/Seg."
.lblB_w.Caption = Format((Rango_w + W_min) * RADaGRADOs, "Standard") & " Grados/Seg."
.lblRango_w.Caption = Format(Rango_w * RADaGRADOs, "Standard") & " Grados/Seg."
```

' Datos de los conjuntos fuzzy del ángulo

```
.lblA_u.Caption = Format(U_min, "Standard") & " Newton"
.lblC1_u.Caption = Format(U_Cen(1), "Standard") & " Newton"
.lblC2_u.Caption = Format(U_Cen(2), "Standard") & " Newton"
.lblC4_u.Caption = Format(U_Cen(4), "Standard") & " Newton"
.lblC5_u.Caption = Format(U_Cen(5), "Standard") & " Newton"
.lblB_u.Caption = Format((Rango_u + U_min), "Standard") & " Newton"
.lblRango_u.Caption = Format(Rango_u, "Standard") & " Newton"
```

End With
End Sub

'SUBROUTINA PARA MOSTRAR LOS VALORES DE LAS VARIABLES EN LA SIMULACION

Private Sub Valores_Analogos()

Dim TiemAnterior As Integer

If Tsim = 0 Then

'Subrutina para dibujar ejes

Dim Xset As Single, Yset As Single

Dim LineaX As Single, LineaY As Single

picGraficos_m.Cls

picGraficos_m.Picture = LoadPicture()

picGraficos_m.DrawWidth = ESPESOReJES

picGraficos_m.Scale (-Tmax / (3 * DIV), FdeX1max + FdeX1max / DIV) -
(Tmax + Tmax / (3 * DIV), -FdeX1max - FdeX1max / DIV)

picGraficos_m.Line (0, 0)-(Tmax, 0) 'Dibujo Eje X

picGraficos_m.Line (0, FdeX1max)-(0, -FdeX1max)

Xset = Tmax / DIV

Yset = FdeX1max / DIV

LineaX = FdeX1max / (4 * DIV)

LineaY = Tmax / (8 * DIV)

For I = 1 To DIV

picGraficos_m.Line (Xset * I, LineaX)-(Xset * I, -LineaX)

Next I

For I = -DIV To DIV Step 1

picGraficos_m.Line (-LineaY, Yset * I)-(LineaY, Yset * I)

Next I

'Ubicación de la barra positiva y negativa del ángulo

lblPositivo_f.Left = lblBase_f.Left + lblBase_f.Width / 2

lblNegativo_f.Left = lblBase_f.Left + lblBase_f.Width / 2 -
lblNegativo_f.Width

'Ubicación de la barra positiva y negativa de la velocidad angular

lblPositivo_w.Left = lblBase_w.Left + lblBase_w.Width / 2

```

lblNegativo_w.Left = lblBase_w.Left + lblBase_w.Width / 2 - _
    lblNegativo_w.Width

'Ubicación de la barra positiva y negativa de la fuerza
lblPositivo_u.Left = lblBase_u.Left + lblBase_u.Width / 2
lblNegativo_u.Left = lblBase_u.Left + lblBase_u.Width / 2 - _
    lblNegativo_u.Width

'Ubicación de la barra positiva y negativa de la posición
lblPositivo_x.Left = lblBase_x.Left + lblBase_x.Width / 2
lblNegativo_x.Left = lblBase_x.Left + lblBase_x.Width / 2 - _
    lblNegativo_x.Width

'Ubicación de la barra positiva y negativa de la velocidad
lblPositivo_v.Left = lblBase_v.Left + lblBase_v.Width / 2
lblNegativo_v.Left = lblBase_v.Left + lblBase_v.Width / 2 - _
    lblNegativo_v.Width

Else
    'Se grafica los puntos que forman las diferentes curvas Vs el tiempo

    'Dibujo un punto de el espacio
    TiemAnterior = Tsim - IntervaloSim

    'Dibujo un punto de la posición
    picGraficos_m.Scale (-Tmax / (3 * DIV), FdeX1max + FdeX1max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeX1max - FdeX1max / DIV)

    picGraficos_m.DrawWidth = ESPESORgRAFIICOS
    picGraficos_m.Line (TiemAnterior * DeltaT, Esp(TiemAnterior))- _
        (Tsim * DeltaT, Esp(Tsim)), ROJO

    'Dibujo un punto de la velocidad
    picGraficos_m.Scale (-Tmax / (3 * DIV), FdeX2max + FdeX2max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeX2max - FdeX2max / DIV)
    picGraficos_m.Line (TiemAnterior * DeltaT, Vel(TiemAnterior))- _
        (Tsim * DeltaT, Vel(Tsim)), AZUL

    'Dibujo un punto del ángulo
    picGraficos_m.Scale (-Tmax / (3 * DIV), FdeX3max + FdeX3max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeX3max - FdeX3max / DIV)
    picGraficos_m.Line (TiemAnterior * DeltaT, Ang(TiemAnterior))- _
        (Tsim * DeltaT, Ang(Tsim)), VERDE

    'Dibujo un punto de la velocidad angular
    picGraficos_m.Scale (-Tmax / (3 * DIV), FdeX4max + FdeX4max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeX4max - FdeX4max / DIV)
    picGraficos_m.Line (TiemAnterior * DeltaT, Wang(TiemAnterior))- _
        (Tsim * DeltaT, Wang(Tsim)), CELESTE

    'Dibujo un punto de la fuerza
    picGraficos_m.Scale (-Tmax / (3 * DIV), FdeU5max + FdeU5max / DIV)- _
        (Tmax + Tmax / (3 * DIV), -FdeU5max - FdeU5max / DIV)

```

```
picGraficos_m.Line (TiemAnterior * DeltaT, Fuerza(TiemAnterior))- _
(Tsim * DeltaT, Fuerza(Tsim)), LILA
End If
```

```
'Muestro valores de ángulo
lblBase_f.Caption = Format(Ang(Tsim) * RADaGRADOs, "Standard")
```

```
If Ang(Tsim) > 0 Then
  lblPositivo_f.Width = Ang(Tsim) * lblBase_f.Width / (2 * FdeX3max)
  lblNegativo_f.Width = lblBase_f.Width / 100
  lblNegativo_f.Left = lblBase_f.Left + lblBase_f.Width / 2 - _
  lblNegativo_f.Width
ElseIf Ang(Tsim) < 0 Then
  lblNegativo_f.Width = Abs(Ang(Tsim)) * lblBase_f.Width / (2 * FdeX3max)
  lblNegativo_f.Left = lblBase_f.Left + lblBase_f.Width / 2 - _
  lblNegativo_f.Width
  lblPositivo_f.Width = lblBase_f.Width / 100
ElseIf Ang(Tsim) = 0 Then
  lblNegativo_f.Width = lblBase_f.Width / 100
  lblNegativo_f.Left = lblBase_f.Left + lblBase_f.Width / 2 - _
  lblNegativo_f.Width
  lblPositivo_f.Width = lblNegativo_f.Width
End If
```

```
'Muestro valores de Velocidad Angular
```

```
lblBase_w.Caption = Format(Wang(Tsim) * RADaGRADOs, "Standard")
```

```
If Wang(Tsim) > 0 Then
  lblPositivo_w.Width = Wang(Tsim) * lblBase_w.Width / (2 * FdeX4max)
  lblNegativo_w.Width = lblBase_w.Width / 100
  lblNegativo_w.Left = lblBase_w.Left + lblBase_w.Width / 2 - _
  lblNegativo_w.Width
ElseIf Wang(Tsim) < 0 Then
  lblNegativo_w.Width = Abs(Wang(Tsim)) * lblBase_w.Width / (2 * FdeX4max)
  lblNegativo_w.Left = lblBase_w.Left + lblBase_w.Width / 2 - _
  lblNegativo_w.Width
  lblPositivo_w.Width = lblBase_w.Width / 100
ElseIf Wang(Tsim) = 0 Then
  lblNegativo_w.Width = lblBase_w.Width / 100
  lblNegativo_w.Left = lblBase_w.Left + lblBase_w.Width / 2 - _
  lblNegativo_w.Width
  lblPositivo_w.Width = lblNegativo_w.Width
End If
```

```
'Muestro valores de fuerza
```

```
lblBase_u.Caption = Format(Fuerza(Tsim), "Standard")
```

```
If Fuerza(Tsim) > 0 Then
  lblPositivo_u.Width = Fuerza(Tsim) * lblBase_w.Width / (2 * FdeU5max)
  lblNegativo_u.Width = lblBase_u.Width / 100
  lblNegativo_u.Left = lblBase_u.Left + lblBase_u.Width / 2 - _
  lblNegativo_u.Width
```

```

ElseIf Fuerza(Tsim) < 0 Then
    lblNegativo_u.Width = Abs(Fuerza(Tsim)) * lblBase_u.Width / (2 * FdeU5max)
    lblNegativo_u.Left = lblBase_u.Left + lblBase_u.Width / 2 - _
        lblNegativo_u.Width
    lblPositivo_u.Width = lblBase_u.Width / 100
ElseIf Fuerza(Tsim) = 0 Then
    lblNegativo_u.Width = lblBase_u.Width / 100
    lblNegativo_u.Left = lblBase_u.Left + lblBase_u.Width / 2 - _
        lblNegativo_u.Width
    lblPositivo_u.Width = lblNegativo_u.Width
End If

```

'Muestro valores de posición

```
lblBase_x.Caption = Format(Esp(Tsim), "Standard")
```

```

If Esp(Tsim) > 0 Then
    lblPositivo_x.Width = Esp(Tsim) * lblBase_x.Width / (2 * FdeX1max)
    lblNegativo_x.Width = lblBase_x.Width / 100
    lblNegativo_x.Left = lblBase_x.Left + lblBase_x.Width / 2 - _
        lblNegativo_x.Width
ElseIf Esp(Tsim) < 0 Then
    lblNegativo_x.Width = Abs(Esp(Tsim)) * lblBase_x.Width / (2 * FdeX1max)
    lblNegativo_x.Left = lblBase_x.Left + lblBase_x.Width / 2 - _
        lblNegativo_x.Width
    lblPositivo_x.Width = lblBase_x.Width / 100
ElseIf Esp(Tsim) = 0 Then
    lblNegativo_x.Width = lblBase_x.Width / 100
    lblNegativo_x.Left = lblBase_x.Left + lblBase_x.Width / 2 - _
        lblNegativo_x.Width
    lblPositivo_x.Width = lblNegativo_x.Width
End If

```

'Muestro valores de velocidad del carro

```
lblBase_v.Caption = Format(Vel(Tsim), "Standard")
```

```

If Vel(Tsim) > 0 Then
    lblPositivo_v.Width = Vel(Tsim) * lblBase_v.Width / (2 * FdeX2max)
    lblNegativo_v.Width = lblBase_v.Width / 100
    lblNegativo_v.Left = lblBase_v.Left + lblBase_v.Width / 2 - _
        lblNegativo_v.Width
ElseIf Vel(Tsim) < 0 Then
    lblNegativo_v.Width = Abs(Vel(Tsim)) * lblBase_v.Width / (2 * FdeX2max)
    lblNegativo_v.Left = lblBase_v.Left + lblBase_v.Width / 2 - _
        lblNegativo_v.Width
    lblPositivo_v.Width = lblBase_v.Width / 100
ElseIf Vel(Tsim) = 0 Then
    lblNegativo_v.Width = lblBase_v.Width / 100
    lblNegativo_v.Left = lblBase_v.Left + lblBase_v.Width / 2 - _
        lblNegativo_v.Width
    lblPositivo_v.Width = lblNegativo_v.Width
End If

```

'Muestro valores de tiempo transcurrido

```
lblBase_t.Caption = Format(Tsim * DeltaT, "Standard")
lblPositivo_t.Width = (Tsim * DeltaT) * lblBase_t.Width / Tmax
```

```
End Sub
```

```
'RESETEA DATOS DE LA VENTANA DE RESULTADOS
```

```
Private Sub ResetVentanaResultados()
```

```
With frmResultados
```

```
'RESET DE DATOS DE CONTROL REDES
```

```
'Método de la bisectriz
```

```
.lblRedbSi.Caption = "No Definida"
.lblRedbWn.Caption = "No Definida"
.lblRedbMp.Caption = "No Definida"
.lblRedbTs.Caption = "No Definida"
.lblRedbPoloD.Caption = "No Definida"
.lblRedbFic.Caption = "No Definida"
.lblRedbKc.Caption = "No Definida"
.lblRedbCc.Caption = "No Definida"
.lblRedbPc.Caption = "No Definida"
```

```
'Método del cero arbitrario
```

```
.lblRedcSi.Caption = "No Definida"
.lblRedcWn.Caption = "No Definida"
.lblRedcMp.Caption = "No Definida"
.lblRedcTs.Caption = "No Definida"
.lblRedcPoloD.Caption = "No Definida"
.lblRedcFic.Caption = "No Definida"
.lblRedcKc.Caption = "No Definida"
.lblRedcCc.Caption = "No Definida"
.lblRedcPc.Caption = "No Definida"
```

```
'RESET DE DATOS DEL CONTROL POR REALIMENTACION DE ESTADOS
```

```
.lblVectorK.Caption = "No Definida"
```

```
'Se muestran los coeficientes del numerador de los estados X(s)
```

.lblXsfila1.Caption = "b10	b11	b12	b13	"
.lblXsfila2.Caption = "b20	b21	b22	b23	"
.lblXsfila3.Caption = "b30	b31	b32	b33	"
.lblXsfila4.Caption = "b40	b41	b42	b43	"

```
.lblQs.Caption = "S^4 + Ac3 S^3 + Ac2 S^2 + Ac1 S + Ac0"
```

```
'Se muestran los coeficientes del numerador de los estados X(t)
```

.lblXtfila1.Caption = " 0	1	0	0	"
.lblXtfila2.Caption = "-K1/M	-K2/M	-K3/M	-K4/M	"
.lblXtfila3.Caption = " 0	0	0	1	"
.lblXtfila4.Caption = "K1/(L'M)	K2/(L'M)	(gM+K3)/(L'M)	K4/(L'M)	"

```
'Se muestra constantes de la division en fracciones parciales
```

```
.lblcA1n.Caption = "Constantes para la posición= No definida"
```

```
.lblcA2n.Caption = "Constantes para la velocidad del carro= No definida"
```

.lblcA3n.Caption = "Constantes para el ángulo = No definida"
 .lblcA4n.Caption = "Constantes para la Velocidad angular = No definida"
 .lblEcuacion.Caption = "Ecuación en el tiempo = No definida"

'RESET DE DATOS DE CONTROL PROPORCIONAL INTEGRAL DERIVATIVO

.lblPID_Si.Caption = "No Definida"
 .lblPID_Wn.Caption = "No Definida"
 .lblPID_Mp.Caption = "No Definida"
 .lblPID_Ts.Caption = "No Definida"

 .lblPIDPoloD.Caption = "No Definida"
 .lblPIDkp.Caption = "No Definida"
 .lblPIDkd.Caption = "No Definida"
 .lblPIDki.Caption = "No Definida"

'RESET DE DATOS DE CONTROL PROPORCIONAL DERIVATIVO

.lblSi.Caption = "No Definida"
 .lblWn.Caption = "No Definida"
 .lbl_Mp.Caption = "No Definida"
 .lblTs.Caption = "No Definida"
 .lblPDkp.Caption = "No Definida"
 .lblPDkd.Caption = "No Definida"
 .lblPoloDominante.Caption = "No Definida"

'RESET DE DATOS DE CONTROL FUZZY

' Datos de los conjuntos fuzzy del ángulo

.lblA_f.Caption = "No Definida"
 .lblC1_f.Caption = "No Definida"
 .lblC2_f.Caption = "No Definida"
 .lblC4_f.Caption = "No Definida"
 .lblC5_f.Caption = "No Definida"
 .lblB_f.Caption = "No Definida"
 .lblRango_f.Caption = "No Definida"

' Datos de los conjuntos fuzzy de la velocidad angular

.lblA_w.Caption = "No Definida"
 .lblC1_w.Caption = "No Definida"
 .lblC2_w.Caption = "No Definida"
 .lblC4_w.Caption = "No Definida"
 .lblC5_w.Caption = "No Definida"
 .lblB_w.Caption = "No Definida"
 .lblRango_w.Caption = "No Definida"

' Datos de los conjuntos fuzzy del ángulo

.lblA_u.Caption = "No Definida"
 .lblC1_u.Caption = "No Definida"
 .lblC2_u.Caption = "No Definida"
 .lblC4_u.Caption = "No Definida"
 .lblC5_u.Caption = "No Definida"
 .lblB_u.Caption = "No Definida"
 .lblRango_u.Caption = "No Definida"

```
'Reseteo matriz de muestreo
K = 0
For I = 1 To NUMMcONFUZZY
  For J = 1 To NUMMcONFUZZY
    .lblConocimiento(K).Caption = ""
    K = K + 1
  Next J
Next I

End With
End Sub
```

'SUBROUTINA PARA SETEO DE VALORES NUEVOS EN LA VENTANA DE INGRESO DE DATOS
Private Sub PenduloNuevo()

```
mnuSeleccionControl.Tag = 0
mnuCalculos.Tag = 0
With frmDatosPendulo
  .txtMc.Text = Format(Mc, "0.00")
  .txtmP.Text = Format(mP, "0.00")
  .txtLongP.Text = Format(LongP, "0.00")
  .txtGr.Text = Format(Gr, "0.00")
  .txtLongPista.Text = Format(LongPista, "0.00")
  .txtJpendulo.Text = Format(Jp, "0.00000")
  .txtEspIni.Text = Format(Xest(1), "0.00")
  .txtVelIni.Text = Format(Xest(2), "0.00")
  .txtAngIni.Text = Format(Xest(3) * RADaGRADOs, "0.00")
  .txtVelAngIni.Text = Format(Xest(4) * RADaGRADOs, "0.00")

```

End With

```
frmDatosPendulo.Show 1
With frmDatosPendulo
If frmDatosPendulo.txtChequeoBoton.Text = "OK" Then
'Se presionó el boton "OK"
```

```
  Mc = CSng(.txtMc.Text)
  mP = CSng(.txtmP.Text)
  LongP = CSng(.txtLongP.Text)
  Gr = CSng(.txtGr.Text)
  LongPista = CSng(.txtLongPista.Text)
  Jp = CSng(.txtJpendulo)
  Xest(1) = CSng(.txtEspIni.Text)
  Xest(2) = CSng(.txtVelIni.Text)
  Xest(3) = CSng(.txtAngIni.Text) * GRADOsRAD
  Xest(4) = CSng(.txtVelAngIni.Text) * GRADOsRAD
  ResetVentanaResultados
  MuestroResGenerales
```

```
ElseIf frmDatosPendulo.txtChequeoBoton.Text = "CANCEL" Then
'Se presionó el boton "CANCEL"
```

```

.txtMc.Text = Format(Mc, "0.00")
.txtmP.Text = Format(mP, "0.00")
.txtLongP.Text = Format(LongP, "0.00")
.txtGr.Text = Format(Gr, "0.00")
.txtLongPista.Text = Format(LongPista, "0.00")
.txtJpendulo.Text = Format(Jp, "0.00000")
.txtEspIni.Text = Format(Xest(1), "0.00")
.txtVelIni.Text = Format(Xest(2), "0.00")
.txtAngIni.Text = Format((Xest(3) * RADaGRADOS), "0.00")
.txtVelAngIni.Text = Format((Xest(3) * RADaGRADOS), "0.00")
Else
'No se presionó "nada"
End If
End With

If frmDatosPendulo.cmdModificar.Tag = 0 Then      'Péndulo con centro de gravedad en

    Jp = mP * LongP * LongP / 3      'el centro
Else
    Jp = Jp          'el dato de Jp es mantenido como dato
                    'de ingreso
End If

Lp = (Jp + mP * LongP ^ 2) / (mP * LongP)
CuadroX = LongPista * 13 / 20 'Para definir
                    'la escala "X" del área donde
                    'va a estar el carrito para la animación
CuadroY = LongPista / 4      'Para definir la escala "y" del área donde
                    'va a estar el carrito para la animación
                    'picGraficos.Scale(-x,y)-(x,-y)

End Sub

'Control Fuzzy
'Restaura los valores de la base de conocimiento
Private Function RestauroBaseConoc(NumFuzzy As Integer) As String

    If NumFuzzy = 1 Then
        RestauroBaseConoc = "MN"
    ElseIf NumFuzzy = 2 Then
        RestauroBaseConoc = "PN"
    ElseIf NumFuzzy = 3 Then
        RestauroBaseConoc = "C"
    ElseIf NumFuzzy = 4 Then
        RestauroBaseConoc = "PP"
    ElseIf NumFuzzy = 5 Then
        RestauroBaseConoc = "MP"
    End If

End Function

```

'Datos Pendulo

```
Private Sub Form_Load()
```

```
End Sub
```

```
'Chequeo de la condición inicial del ángulo
```

```
Private Sub txtAngIni_LostFocus()
```

```
ChequeoDatos txtAngIni
```

```
If (CSng(txtAngIni.Text) > 45) Or _
```

```
(CSng(txtAngIni.Text) < -45) Then
```

```
'Si el valor ingresado es negativo
```

```
msg = "El valor de la condición inicial del ángulo no es válido, "
```

```
& "ingrese un valor entre -11 y 11 grados para obtener "
```

```
& "mejores resultados (Linealidad del Modelo Matemático)"
```

```
Mostrar
```

```
txtAngIni.Text = Format(ANGiNIC, "0.00")
```

```
txtAngIni.SetFocus
```

```
txtAngIni.SelStart = 0
```

```
txtAngIni.SelLength = Len(txtAngIni.Text)
```

```
End If
```

```
End Sub
```

```
Private Sub txtEspIni_LostFocus()
```

```
ChequeoDatos txtEspIni
```

```
End Sub
```

```
'Chequeo del dato de la gravedad'
```

```
Private Sub txtGr_LostFocus()
```

```
ChequeoDatos txtGr
```

```
If CSng(txtGr.Text) <= 0 Then
```

```
'Si el valor ingresado es negativo
```

```
msg = "El valor de gravedad 'g' no es válido, "
```

```
& "ingrese un valor mayor que cero"
```

```
Mostrar
```

```
txtGr.Text = Format(GRAVeDAD, "0.00")
```

```
txtGr.SetFocus
```

```
txtGr.SelStart = 0
```

```
txtGr.SelLength = Len(txtGr.Text)
```

```
End If
```

```
End Sub
```

```
'Chequeo de el momento de inercia
```

```
Private Sub txtJpendulo_LostFocus()
```

```
ChequeoDatos txtJpendulo
```

```

If CSng(txtJpendulo.Text) <= 0 Then
'Si el valor ingresado es negativo
  msg = "El valor del momento de Inercia 'J' no es válido, " _
    & "ingrese un valor mayor que cero"
  Mostrar
  txtJpendulo.Text = Format(iNERCIAJ, "0.00000")
  txtJpendulo.SetFocus
  txtJpendulo.SelStart = 0
  txtJpendulo.SelLength = Len(txtJpendulo.Text)
End If
End Sub

'Chequeo de la mitad de la longitud del péndulo
Private Sub txtLongP_LostFocus()
ChequeoDatos txtLongP

If CSng(txtLongP.Text) <= 0 Then
'Si el valor ingresado es negativo
  msg = "La distancia entre el extremo inferior del péndulo y " _
    & "el centro de gravedad 'L' no es válido, " _
    & "ingrese un valor mayor que cero"
  Mostrar
  txtLongP.Text = Format(LONGpEND, "0.00")
  txtLongP.SetFocus
  txtLongP.SelStart = 0
  txtLongP.SelLength = Len(txtLongP.Text)
End If

End Sub

'Chequeo de la Longitud de la pista
Private Sub txtLongPista_LostFocus()
ChequeoDatos txtLongPista

If CSng(txtLongPista.Text) <= 0 Then
'Si el valor ingresado es negativo
  msg = "La longitud de la pista 'Long. Pista' no es válido, " _
    & "ingrese un valor mayor que cero"
  Mostrar
  txtLongPista.Text = Format(LONGITUDpISTA, "0.00")
  txtLongPista.SetFocus
  txtLongPista.SelStart = 0
  txtLongPista.SelLength = Len(txtLongPista.Text)
End If
End Sub

'Chequeo del Dato de la masa del carro
Private Sub txtMc_LostFocus()
ChequeoDatos txtMc

If CSng(txtMc.Text) <= 0 Then

```

```
'Si el valor ingresado es negativo
  msg = "El valor de Masa del Carro 'M' no es válido, " _
    & "ingrese un valor mayor que cero"
  Mostrar
  txtMc.Text = Format(MASAcARRO, "0.00")
  txtMc.SetFocus
  txtMc.SelStart = 0
  txtMc.SelLength = Len(txtMc.Text)
```

```
End If
```

```
End Sub
```

```
'Chequeo de datos de la Masa del péndulo
Private Sub txtmP_LostFocus()
```

```
  ChequeoDatos txtmP
```

```
  If CSng(txtmP.Text) <= 0 Then
```

```
    'Si el valor ingresado es negativo
```

```
    msg = "El valor de Masa del Péndulo 'm' no es válido, " _
      & "ingrese un valor mayor que cero"
```

```
    Mostrar
```

```
    txtmP.Text = Format(MASApENDULO, "0.00")
```

```
    txtmP.SetFocus
```

```
    txtmP.SelStart = 0
```

```
    txtmP.SelLength = Len(txtmP.Text)
```

```
  End If
```

```
End Sub
```

```
Private Sub txtVelAngIno_Change()
```

```
End Sub
```

```
'Chequeo de la condición Inicial de la Velocidad angular
```

```
Private Sub txtVelAngIni_LostFocus()
```

```
  ChequeoDatos txtVelAngIni
```

```
End Sub
```

```
Private Sub txtVelIni_LostFocus()
```

```
  ChequeoDatos txtVelIni
```

```
End Sub
```

'Graficos Varios

```
Option Explicit
```

```

Private Sub cmdCancel_Click()
ChkEspTiemp.Value = Unchecked
ChkvelTiemp.Value = Unchecked
ChkangTiemp.Value = Unchecked
ChkwangTiemp.Value = Unchecked
ChkfuerTiemp.Value = Unchecked
frmGraficosVarios.Hide
End Sub

```

```

Private Sub cmdOK_Click()
frmGraficosVarios.Hide
End Sub

```

```

Private Sub Frame1_DragDrop(Source As Control, X As Single, Y As Single)

```

```

End Sub

```

'Presentación

```

Option Explicit

```

```

Private Sub cmdPresentOK_Click()

```

```

cmdPresentOK.Visible = False
frmPresentacion.Hide
End Sub

```

```

Private Sub Form_Load()
frmPresentacion.Left = Screen.Width / 2 - frmPresentacion.Width / 2
frmPresentacion.Top = Screen.Height / 2 - frmPresentacion.Height / 2

```

```

timerPresentacion.Enabled = True

```

```

End Sub

```

```

Private Sub Timer1_Timer()

```

```

End Sub

```

```

Private Sub timerPresentacion_Timer()

```

```

If timerPresentacion.Tag <> 0 Then
timerPresentacion.Enabled = False
Load frmAlternativa
frmAlternativa.Show
Else
timerPresentacion.Tag = 1
End If
End Sub

```

'Resultados

Option Explicit

```
Private Sub cmdOK_Click()
frmResultados.Hide
End Sub
```

```
Private Sub mnuImprimirResultados_Click()
```

```
On Error GoTo ErrorHandler 'Habilitacion del manejo de error
Err.Clear
```

```
frmAlternativa.cdgDialogo.Flags = cdIPDNoSelection Or cdIPDNoPageNums Or _
    cdIPDHidePrintToFile Or cdIPDUseDevModeCopies
frmAlternativa.cdgDialogo.ShowPrinter
frmResultados.PrintForm
```

```
ErrorHandler:      'Manejo de Errores
```

```
If Err.Number <> 0 Then 'Chequeo de si existe error
```

```
Dim Estilo As Integer
```

```
Dim Titulo As String
```

```
    If Err.Number <> 32755 Then
```

```
        msg = "Error # " & Str(Err.Number) & " Se ha Generado un Error Tipo " _
            & Chr(13) & Err.Description & Chr(13) & "Los resultados no se" _
            & " pueden imprimir Intente Nuevamente"
```

```
        Estilo = vbOKOnly + vbCritical ' Define botones
```

```
        Titulo = "Error al Imprimir" ' Define titulo
```

```
        MsgBox msg, Estilo, Titulo
```

```
    End If
```

```
End If
```

```
End Sub
```

```
Private Sub SSTabResultados_DblClick()
```

```
End Sub
```

Tipos De Control

Option Explicit

```
Private Sub SSTab1_DblClick()
```

```
End Sub
```

```
Private Sub cmdCancel_Click()
```

```

txtChequeoBoton.Text = "CANCEL"
frmTiposDeControl.Hide
End Sub

```

```

Private Sub cmdOK_Click()

```

```

'-----
On Error GoTo ErrorDato 'habilitación de manejo de error
Err.Clear

```

```

'-----
If SSTabTipoC.Tab = 0 Then

```

```

'REALIMENTACION DE ESTADOS
TipoControl = CONTROLrESTADOS

```

```

'Chequeo de datos de los polos
'Chequeo parte imaginaria

```

```

'-----
ChequeoDatos txtParteImag(1)
ChequeoDatos txtParteImag(3)

```

```

If txtParteImag(1).Text = "" Then
    txtParteImag(1).Text = 0
End If

```

```

If txtParteImag(3).Text = "" Then
    txtParteImag(3).Text = 0
End If

```

```

lblparteImag(2).Caption = txtParteImag(1).Text
lblparteImag(4).Caption = txtParteImag(3).Text

```

```

If (Val(txtParteImag(1).Text) <> 0) Or (CSng(txtParteImag(1).Text) <> 0) Then
    txtParteReal(2).Text = txtParteReal(1).Text
End If

```

```

If (Val(txtParteImag(3).Text) <> 0) Or (CSng(txtParteImag(3).Text) <> 0) Then
    txtParteReal(4).Text = txtParteReal(3).Text
End If

```

```

'Chequeo de datos Parte real

```

```

'-----
If (CSng(txtParteImag(1).Text) <> 0) Or _
    (Val(txtParteImag(1).Text) <> 0) Then
    txtParteReal(2).Text = txtParteReal(1).Text
End If

```

```

If (CSng(txtParteImag(3).Text) <> 0) Or _
    (Val(txtParteImag(3).Text) <> 0) Then
    txtParteReal(4).Text = txtParteReal(3).Text
End If

```

```

ChequeoDatos txtParteReal(1)
ChequeoDatos txtParteReal(2)
ChequeoDatos txtParteReal(3)

```

ChequeoDatos txtParteReal(4)

'Chequeo que la parte real de los polos ingresados
'estén en el semiplano izquierdo del plano s

```
If CSng(txtParteReal(1).Text) >= 0 Then
    msg = "La parte real del POLO 1 debe ser negativo y diferente de cero"
    Mostrar
    txtParteReal(1).Text = Format(PoloD(1).r, "0.00")
    txtParteReal(1).SetFocus
    txtParteReal(1).SelStart = 0
    txtParteReal(1).SelLength = Len(txtParteReal(1).Text)
    txtChequeoBoton.Text = "Error"
End If
```

```
If CSng(txtParteReal(2).Text) >= 0 Then
    msg = "La parte real del POLO 2 debe ser negativo y diferente de cero"
    Mostrar
    txtParteReal(2).Text = Format(PoloD(2).r, "0.00")
    txtParteReal(2).SetFocus
    txtParteReal(2).SelStart = 0
    txtParteReal(2).SelLength = Len(txtParteReal(2).Text)
    txtChequeoBoton.Text = "Error"
End If
```

```
If CSng(txtParteReal(3).Text) >= 0 Then

    msg = "La parte real del POLO 3 debe ser negativo y diferente de cero"
    Mostrar
    txtParteReal(3).Text = Format(PoloD(3).r, "0.00")
    txtParteReal(3).SetFocus
    txtParteReal(3).SelStart = 0
    txtParteReal(3).SelLength = Len(txtParteReal(3).Text)
    txtChequeoBoton.Text = "Error"
End If
```

```
If CSng(txtParteReal(4).Text) >= 0 Then
    msg = "La parte real del POLO 4 debe ser negativo y diferente de cero"
    Mostrar
    txtParteReal(4).Text = Format(PoloD(4).r, "0.00")
    txtParteReal(4).SetFocus
    txtParteReal(4).SelStart = 0
    txtParteReal(4).SelLength = Len(txtParteReal(4).Text)
    txtChequeoBoton.Text = "Error"
End If
```

'--

```
ElseIf SSTabTipoC.Tab = 1 Then
    TipoControl = CONTROLrEDES
```

```
If SSTabRedAdelanto.Tab = 0 Then
```

```
'METODO DE LA BISECTRIZ
TipoRedAdelanto = REDaDELbISECTRIZ
```

```

'Chequeo de datos
'Chequeo del máximo sobreimpulso
ChequeoDatos txtBisectrizMp
If (CSng(txtBisectrizMp.Text) < 0.1) Or _
  (CSng(txtBisectrizMp.Text) > 100) Then
  msg = "El Máximo Sobreimpulso porcentual debe estar comprendido" _
    & " entre 0.1 % y 100 %"
  Mostrar
  txtBisectrizMp.Text = Format(RED_Mp, "0.00")
  txtBisectrizMp.SetFocus
  txtBisectrizMp.SelStart = 0
  txtBisectrizMp.SelLength = Len(txtBisectrizMp.Text)
  txtChequeoBoton.Text = "Error"
End If

'Chequeo del tiempo de establecimiento
ChequeoDatos txtBisectrizTs
If (CSng(txtBisectrizTs.Text) < 0.1) Or _
  (CSng(txtBisectrizTs.Text) > 30) Then
  msg = "El tiempo de establecimiento debe estar comprendido" _
    & " entre 0.1 y 30 segundos "
  Mostrar
  txtBisectrizTs.Text = Format(RED_Ts, "0.00")
  txtBisectrizTs.SetFocus
  txtBisectrizTs.SelStart = 0
  txtBisectrizTs.SelLength = Len(txtBisectrizTs.Text)
  txtChequeoBoton.Text = "Error"
End If

'--
ElseIf SSTabRedAdelanto.Tab = 1 Then

'METODO DEL CERO ARBITRARIO
TipoRedAdelanto = REDaDELcERO

'Chequeo de Datos
'Chequeo del Máximo sobreimpulso
ChequeoDatos txtCeroMp
If (CSng(txtCeroMp.Text) < 0.1) Or _
  (CSng(txtCeroMp.Text) > 100) Then
  msg = "El Máximo Sobreimpulso porcentual debe estar comprendido" _
    & " entre 0.1 % y 100 %"
  Mostrar
  txtCeroMp.Text = Format(RED_Mp, "0.00")
  txtCeroMp.SetFocus
  txtCeroMp.SelStart = 0
  txtCeroMp.SelLength = Len(txtCeroMp.Text)
  txtChequeoBoton.Text = "Error"
End If

'Chequeo del Tiempo de establecimiento
ChequeoDatos txtCeroTs
If (CSng(txtCeroTs.Text) < 0.1) Or _
  (CSng(txtCeroTs.Text) > 30) Then

```

```

msg = "El tiempo de establecimiento debe estar comprendido" _
    & " entre 0.1 y 30 segundos"
Mostrar
txtCeroTs.Text = Format(RED_Ts, "0.00")
txtCeroTs.SetFocus
txtCeroTs.SelStart = 0
txtCeroTs.SelLength = Len(txtCeroTs.Text)
txtChequeoBoton.Text = "Error"
End If
'Chequeo del cero arbitrario
ChequeoDatos txtCeroCc

If CSng(txtCeroCc.Text) >= 0 Then
    msg = "El valor del cero 'Cc' debe ser negativo y diferente de cero"
    Mostrar
    txtCeroCc.Text = Format(Red_Cc, "0.00")
    txtCeroCc.SetFocus
    txtCeroCc.SelStart = 0
    txtCeroCc.SelLength = Len(txtCeroCc.Text)
    txtChequeoBoton.Text = "Error"
End If

Else
    TipoControl = SINcONTROL
End If

ElseIf SStabTipoC.Tab = 2 Then
    TipoControl = CONTROLaCCIONES

.If SStabAcciones.Tab = 0 Then

'ACCION PD CONOCIENDO WN Y E
TipoAcciones = ACCIONPD_WnE

'Chequeo del dato Wn
ChequeoDatos txtWn
If (CSng(txtWn.Text) <= 0) Then
    msg = "La frecuencia no amortiguada 'Wn' debe" _
        & " ser positivo y diferente de cero "
    Mostrar
    txtWn.Text = Format(PD_Wn_CONST, "0.00")
    txtWn.SetFocus
    txtWn.SelStart = 0
    txtWn.SelLength = Len(txtWn.Text)
    txtChequeoBoton.Text = "Error"
End If

'Chequeo del dato Si
ChequeoDatos txtSi
If (CSng(txtSi.Text) < 0.1) Or _
    (CSng(txtSi.Text) > 0.9) Then
    msg = "El coeficiente de amortiguamiento debe estar comprendido" _
        & " entre 0.1 y 0.9 "
    Mostrar
    txtSi.Text = Format(PD_Si_CONST, "0.000")

```

```

txtSi.SetFocus
txtSi.SelStart = 0
txtSi.SelLength = Len(txtSi.Text)
txtChequeoBoton.Text = "Error"
End If

```

```

ElseIf SSTabAcciones.Tab = 1 Then

```

```

'ACCION PD CONOCIENDO Kp y Kd
TipoAcciones = ACCIONPD

```

```

'Chequeo del dato Kp
'-----

```

```

ChequeoDatos txtpdkp
If CSng(txtpdkp.Text) <= 0 Then
    msg = "El valor de Kp debe ser positivo y diferente de cero"
    Mostrar
    txtpdkp.Text = Format(PD_Kp_CONST, "0.000")
    txtpdkp.SetFocus
    txtpdkp.SelStart = 0
    txtpdkp.SelLength = Len(txtpdkp.Text)
    txtChequeoBoton.Text = "Error"
End If

```

```

'Chequeo del dato Kd
'-----

```

```

ChequeoDatos txtpdkd
If CSng(txtpdkd.Text) <= 0 Then
    msg = "El valor de Kd debe ser positivo y diferente de cero"
    Mostrar
    txtpdkd.Text = Format(PD_Kd_CONST, "0.000")
    txtpdkd.SetFocus
    txtpdkd.SelStart = 0
    txtpdkd.SelLength = Len(txtpdkd.Text)
    txtChequeoBoton.Text = "Error"
End If

```

```

ElseIf SSTabAcciones.Tab = 2 Then
TipoAcciones = ACCIONPID

```

```

'Chequeo del dato Kp
'-----

```

```

ChequeoDatos txtPIDKp
If CSng(txtPIDKp.Text) <= 0 Then
    msg = "El valor de Kp debe ser positivo y diferente de cero"
    Mostrar
    txtPIDKp.Text = Format(PID_Kp_CONST, "0.000")
    txtPIDKp.SetFocus
    txtPIDKp.SelStart = 0
    txtPIDKp.SelLength = Len(txtPIDKp.Text)
    txtChequeoBoton.Text = "Error"
End If

```

'Chequeo del dato Kd

ChequeoDatos txtPIDKd

If CSng(txtPIDKd.Text) <= 0 Then

 msg = "El valor de Kd debe ser positivo y diferente de cero"

 Mostrar

 txtPIDKd.Text = Format(PID_Kd_CONST, "0.000")

 txtPIDKd.SetFocus

 txtPIDKd.SelStart = 0

 txtPIDKd.SelLength = Len(txtPIDKd.Text)

 txtChequeoBoton.Text = "Error"

End If

'Chequeo del dato Ki

ChequeoDatos txtPIDKi

If CSng(txtPIDKi.Text) <= 0 Then

 msg = "El valor de Ki debe ser positivo y diferente de cero"

 Mostrar

 txtPIDKi.Text = Format(PID_Ki_CONST, "0.000")

 txtPIDKi.SetFocus

 txtPIDKi.SelStart = 0

 txtPIDKi.SelLength = Len(txtPIDKi.Text)

 txtChequeoBoton.Text = "Error"

End If

Else

 TipoControl = SINCONTROL

End If

ElseIf SSTabTipoC.Tab = 3 Then

 TipoControl = CONTROLFUZZY

 'Chequeo del rango de la fuerza

 ChequeoDatos txtRangoFuerza

 If CSng(txtRangoFuerza.Text) <= 0 Then

 msg = "El valor del rango de la fuerza debe ser positivo " _
 & "y diferente de cero"

 Mostrar

 txtRangoFuerza.Text = Format(rANGOFUERZA, "0.00")

 txtRangoFuerza.SetFocus

 txtRangoFuerza.SelStart = 0

 txtRangoFuerza.SelLength = Len(txtRangoFuerza.Text)

 txtChequeoBoton.Text = "Error"

 End If

 'Chequeo del rango del ángulo

 ChequeoDatos txtRangoAngulo

 If CSng(txtRangoAngulo.Text) <= 0 Then

 msg = "El valor del rango del ángulo debe ser positivo " _
 & "y diferente de cero"

 Mostrar

```

txtRangoAngulo.Text = Format(rANGOaNGULO, "0.00")
txtRangoAngulo.SetFocus
txtRangoAngulo.SelStart = 0
txtRangoAngulo.SelLength = Lcn(txtRangoAngulo.Text)
txtChequeoBoton.Text = "Error"
End If

'Chequeo del rango de la velocidad angular
'-----
ChequeoDatos txtRangoWang
If CSng(txtRangoWang.Text) <= 0 Then
    msg = "El valor del rango de la velocidad angular debe ser positivo " _
        & "y diferente de cero"
    Mostrar
    txtRangoWang.Text = Format(rANGOVeLaNGULAR, "0.00")
    txtRangoWang.SetFocus
    txtRangoWang.SelStart = 0
    txtRangoWang.SelLength = Len(txtRangoWang.Text)
    txtChequeoBoton.Text = "Error"

End If

'Chequeo del tiempo de simulación
'-----
ChequeoDatos txtTiempoSimulacion
If CSng(txtTiempoSimulacion.Text) <= 0 Then
    msg = "El valor del tiempo de simulación debe ser positivo " _
        & "y diferente de cero"
    Mostrar
    txtTiempoSimulacion.Text = Format(tIEMPOsIMULACION, "0.00")
    txtTiempoSimulacion.SetFocus
    txtTiempoSimulacion.SelStart = 0
    txtTiempoSimulacion.SelLength = Len(txtTiempoSimulacion.Text)
    txtChequeoBoton.Text = "Error"
End If
lblTiempoMuestreo.Caption = CStr(CSng(txtTiempoSimulacion.Text) / NUMmAXpUNTOS)

Else
    TipoControl = SINcONTROL
End If

'-----
ErrorDato:      'Manejo de Errores

If Err.Number <> 0 Then 'Chequeo de si existe error
    txtChequeoBoton.Text = "Error"
End If
'-----

'Cheque si se ha producido algun dato de entrada incorrecto

If txtChequeoBoton.Text = "Error" Then
    txtChequeoBoton.Text = ""
Else
    txtChequeoBoton.Text = "OK"

```

```

frmTiposDeControl.Hide
End If

```

```

End Sub

```

```

Private Sub SSTabTiposControl_DblClick()

```

```

End Sub

```

```

Private Sub Form_Load()

```

```

End Sub

```

```

Private Sub SpinButton1_SpinDown(Index As Integer)

```

```

If lblConocimiento(Index) = "MN" Then
    lblConocimiento(Index) = "PN"
ElseIf lblConocimiento(Index) = "PN" Then
    lblConocimiento(Index) = "C"
ElseIf lblConocimiento(Index) = "C" Then
    lblConocimiento(Index) = "PP"
ElseIf lblConocimiento(Index) = "PP" Then
    lblConocimiento(Index) = "MP"
ElseIf lblConocimiento(Index) = "MP" Then
    lblConocimiento(Index) = "MN"
End If
End Sub

```

```

Private Sub SpinButton1_SpinUp(Index As Integer)

```

```

If lblConocimiento(Index) = "MN" Then
    lblConocimiento(Index) = "MP"
ElseIf lblConocimiento(Index) = "MP" Then
    lblConocimiento(Index) = "PP"
ElseIf lblConocimiento(Index) = "PP" Then
    lblConocimiento(Index) = "C"
ElseIf lblConocimiento(Index) = "C" Then
    lblConocimiento(Index) = "PN"
ElseIf lblConocimiento(Index) = "PN" Then
    lblConocimiento(Index) = "MN"
End If
End Sub

```

```

Private Sub txtBisectrizMp_LostFocus()

```

```

ChequeoDatos txtBisectrizMp

```

```

If (CSng(txtBisectrizMp.Text) < 0.1) Or _
    (CSng(txtBisectrizMp.Text) > 100) Then

```

```

'Si el valor ingresado del máximo sobre impulso debe
'estar comprendido entre 0.1 y 100 %
msg = "El Máximo Sobreimpulso porcentual debe estar comprendido" _
    & " entre 0.1 % y 100 %"
Mostrar
txtBisectrizMp.Text = Format(RED_Mp, "0.00")
txtBisectrizMp.SetFocus
txtBisectrizMp.SelStart = 0
txtBisectrizMp.SelLength = Len(txtBisectrizMp.Text)

End If
End Sub

Private Sub txtBisectrizTs_LostFocus()

ChequeoDatos txtBisectrizTs

If (CSng(txtBisectrizTs.Text) < 0.1) Or _
    (CSng(txtBisectrizTs.Text) > 30) Then
'Si el valor ingresado del tiempo de establecimiento debe
'estar comprendido entre 0.1 seg y 3 seg
msg = "El tiempo de establecimiento debe estar comprendido" _
    & " entre 0.1 y 30 segundos "
Mostrar
txtBisectrizTs.Text = Format(RED_Ts, "0.00")
txtBisectrizTs.SetFocus
txtBisectrizTs.SelStart = 0
txtBisectrizTs.SelLength = Len(txtBisectrizTs.Text)
End If
End Sub

Private Sub txtCeroCc_LostFocus()

ChequeoDatos txtCeroCc
If CSng(txtCeroCc.Text) >= 0 Then
'Si el valor ingresado no puede ser positivo o cero
msg = "El valor del cero 'Cc' debe ser negativo y diferente de cero"
Mostrar
txtCeroCc.Text = Format(Red_Cc, "0.00")
txtCeroCc.SetFocus
txtCeroCc.SelStart = 0
txtCeroCc.SelLength = Len(txtCeroCc.Text)
End If
End Sub

Private Sub txtCeroMp_LostFocus()

ChequeoDatos txtCeroMp
If (CSng(txtCeroMp.Text) < 0.1) Or _
    (CSng(txtCeroMp.Text) > 100) Then
'Si el valor ingresado del máximo sobre impulso debe
'estar comprendido entre 0.1 y 100 %

```

```

msg = "El Máximo Sobreimpulso porcentual debe estar comprendido" _
    & " entre 0.1 % y 100 %"
Mostrar
txtCeroMp.Text = Format(RED_Mp, "0.00")
txtCeroMp.SetFocus
txtCeroMp.SelStart = 0
txtCeroMp.SelLength = Len(txtCeroMp.Text)
End If
End Sub

```

```
Private Sub txtCeroTs_LostFocus()
```

```

ChequeoDatos txtCeroTs
If (CSng(txtCeroTs.Text) < 0.1) Or _
    (CSng(txtCeroTs.Text) > 30) Then
'Si el valor ingresado del tiempo de establecimiento debe
'estar comprendido entre 0.1 seg y 30 seg
msg = "El tiempo de establecimiento debe estar comprendido" _
    & " entre 0.1 y 30 segundos"
Mostrar
txtCeroTs.Text = Format(RED_Ts, "0.00")
txtCeroTs.SetFocus
txtCeroTs.SelStart = 0
txtCeroTs.SelLength = Len(txtCeroTs.Text)
End If
End Sub

```

```
Private Sub txtParteImag_LostFocus(Index As Integer)
```

```

ChequeoDatos txtParteImag(1)
ChequeoDatos txtParteImag(3)

If txtParteImag(1).Text = "" Then
    txtParteImag(1).Text = 0
End If
If txtParteImag(3).Text = "" Then
    txtParteImag(3).Text = 0
End If

lblparteImag(2).Caption = txtParteImag(1).Text
lblparteImag(4).Caption = txtParteImag(3).Text

If (Val(txtParteImag(1).Text) <> 0) Or (CSng(txtParteImag(1).Text) <> 0) Then
    txtParteReal(2).Text = txtParteReal(1).Text
End If

If (Val(txtParteImag(3).Text) <> 0) Or (CSng(txtParteImag(3).Text) <> 0) Then
    txtParteReal(4).Text = txtParteReal(3).Text
End If
End Sub

```

```
Private Sub txtParteReal_LostFocus(Index As Integer)
```

```

If (CSng(txtParteImag(1).Text) <> 0) Or _
  (Val(txtParteImag(1).Text) <> 0) Then
  txtParteReal(2).Text = txtParteReal(1).Text
End If

```

```

If (CSng(txtParteImag(3).Text) <> 0) Or _
  (Val(txtParteImag(3).Text) <> 0) Then
  txtParteReal(4).Text = txtParteReal(3).Text
End If

```

```

ChequeoDatos txtParteReal(1)
ChequeoDatos txtParteReal(2)
ChequeoDatos txtParteReal(3)
ChequeoDatos txtParteReal(4)

```

'Chequeo que la parte real de los polos ingresados
'estén en el semiplano izquierdo del plano s

```

If CSng(txtParteReal(1).Text) >= 0 Then
  'Si el valor ingresado es positivo o cero
  msg = "La parte real del POLO 1 debe ser negativo y diferente de cero"
  Mostrar
  txtParteReal(1).Text = Format(PoloD(1).r, "0.00")
  txtParteReal(1).SetFocus
  txtParteReal(1).SelStart = 0
  txtParteReal(1).SelLength = Len(txtParteReal(1).Text)
End If

```

```

If CSng(txtParteReal(2).Text) >= 0 Then
  'Si el valor ingresado es positivo o cero
  msg = "La parte real del POLO 2 debe ser negativo y diferente de cero"
  Mostrar
  txtParteReal(2).Text = Format(PoloD(2).r, "0.00")
  txtParteReal(2).SetFocus
  txtParteReal(2).SelStart = 0
  txtParteReal(2).SelLength = Len(txtParteReal(2).Text)

```

End If

```

If CSng(txtParteReal(3).Text) >= 0 Then
  'Si el valor ingresado es positivo o cero
  msg = "La parte real del POLO 3 debe ser negativo y diferente de cero"
  Mostrar
  txtParteReal(3).Text = Format(PoloD(3).r, "0.00")
  txtParteReal(3).SetFocus
  txtParteReal(3).SelStart = 0
  txtParteReal(3).SelLength = Len(txtParteReal(3).Text)

```

End If

```

If CSng(txtParteReal(4).Text) >= 0 Then
  'Si el valor ingresado es positivo o cero
  msg = "La parte real del POLO 4 debe ser negativo y diferente de cero"
  Mostrar

```

```

txtParteReal(4).Text = Format(PoloD(4).r, "0.00")
txtParteReal(4).SetFocus
txtParteReal(4).SelStart = 0
txtParteReal(4).SelLength = Len(txtParteReal(4).Text)

End If

End Sub

'Datos del Control PD
Private Sub txtpdkd_LostFocus()

ChequeoDatos txtpdkd

If CSng(txtpdkd.Text) <= 0 Then
'Si el valor ingresado no puede ser negativo o cero
msg = "El valor de Kd debe ser positivo y diferente de cero"
Mostrar
txtpdkd.Text = Format(PD_Kd_CONST, "0.000")
txtpdkd.SetFocus
txtpdkd.SelStart = 0
txtpdkd.SelLength = Len(txtpdkd.Text)

End If

End Sub

'Datos del control PD
Private Sub txtpdkp_LostFocus()

ChequeoDatos txtpdkp
If CSng(txtpdkp.Text) <= 0 Then
'Si el valor ingresado no puede ser negativo o cero
msg = "El valor de Kp debe ser positivo y diferente de cero"
Mostrar
txtpdkp.Text = Format(PD_Kp_CONST, "0.000")
txtpdkp.SetFocus
txtpdkp.SelStart = 0
txtpdkp.SelLength = Len(txtpdkp.Text)

End If

End Sub

Private Sub txtPIDKd_LostFocus()

ChequeoDatos txtPIDKd
If CSng(txtPIDKd.Text) <= 0 Then
'Si el valor ingresado no puede ser negativo o cero
msg = "El valor de Kd debe ser positivo y diferente de cero"
Mostrar
txtPIDKd.Text = Format(PID_Kd_CONST, "0.000")
txtPIDKd.SetFocus

```

```

txtPIDKd.SelStart = 0
txtPIDKd.SelLength = Len(txtPIDKd.Text)

End If
End Sub

Private Sub txtPIDKi_LostFocus()

ChequeoDatos txtPIDKi
If CSng(txtPIDKi.Text) <= 0 Then
'Si el valor ingresado no puede ser negativo o cero
msg = "El valor de Ki debe ser positivo y diferente de cero"
Mostrar
txtPIDKi.Text = Format(PID_Ki_CONST, "0.000")
txtPIDKi.SetFocus
txtPIDKi.SelStart = 0
txtPIDKi.SelLength = Len(txtPIDKi.Text)

End If
End Sub

Private Sub txtPIDKp_LostFocus()

ChequeoDatos txtPIDKp
If CSng(txtPIDKp.Text) <= 0 Then
'Si el valor ingresado no puede ser negativo o cero
msg = "El valor de Kp debe ser positivo y diferente de cero"
Mostrar
txtPIDKp.Text = Format(PID_Kp_CONST, "0.000")
txtPIDKp.SetFocus
txtPIDKp.SelStart = 0
txtPIDKp.SelLength = Len(txtPIDKp.Text)
End If
End Sub

Private Sub txtRangoAngulo_LostFocus()

ChequeoDatos txtRangoAngulo
If CSng(txtRangoAngulo.Text) <= 0 Then
'Si el valor ingresado del rango de la velocidad angular es negativo
msg = "El valor del rango del ángulo debe ser positivo " _
& "y diferente de cero"
Mostrar
txtRangoAngulo.Text = Format(rANGOaNGULO, "0.00")
txtRangoAngulo.SetFocus
txtRangoAngulo.SelStart = 0
txtRangoAngulo.SelLength = Len(txtRangoAngulo.Text)

End If

```

End Sub

Private Sub txtRangoFuerza_LostFocus()

ChequeoDatos txtRangoFuerza

If CSng(txtRangoFuerza.Text) <= 0 Then

'Si el valor ingresado del rango de la fuerza es negativo
msg = "El valor del rango de la fuerza debe ser positivo " _
& "y diferente de cero"

Mostrar

txtRangoFuerza.Text = Format(rANGOFUERZA, "0.00")

txtRangoFuerza.SetFocus

txtRangoFuerza.SelStart = 0

txtRangoFuerza.SelLength = Len(txtRangoFuerza.Text)

End If

End Sub

Private Sub txtRangoWang_LostFocus()

ChequeoDatos txtRangoWang

If CSng(txtRangoWang.Text) <= 0 Then

'Si el valor ingresado del rango de la fuerza es negativo
msg = "El valor del rango de la velocidad angular debe ser positivo " _
& "y diferente de cero"

Mostrar

txtRangoWang.Text = Format(rANGOVELANGULAR, "0.00")

txtRangoWang.SetFocus

txtRangoWang.SelStart = 0

txtRangoWang.SelLength = Len(txtRangoWang.Text)

End If

End Sub

'Control PD Dato de la frecuencia no amortiguada

Private Sub txtSi_LostFocus()

ChequeoDatos txtSi

If (CSng(txtSi.Text) < 0.1) Or _
(CSng(txtSi.Text) > 0.9) Then

'Si el valor ingresado del máximo sobre impulso debe
'estar comprendido entre 0.1 y 0.9

msg = "El coeficiente de amortiguamiento debe estar comprendido" _
& " entre 0.1 y 0.9 "

Mostrar

txtSi.Text = Format(PD_Si_CONST, "0.000")

txtSi.SetFocus

txtSi.SelStart = 0

```
txtSi.SelLength = Len(txtSi.Text)
```

```
End If
End Sub
```

```
Private Sub txtTiempoSimulacion_LostFocus()
```

```
ChequeoDatos txtTiempoSimulacion
If CSng(txtTiempoSimulacion.Text) <= 0 Then
    'Si el valor ingresado del rango de la velocidad angular es negativo
    msg = "El valor del tiempo de simulación debe ser positivo " _
        & "y diferente de cero"
    Mostrar
    txtTiempoSimulacion.Text = Format(tIEMPOsIMULACION, "0.00")
    txtTiempoSimulacion.SetFocus
    txtTiempoSimulacion.SelStart = 0
    txtTiempoSimulacion.SelLength = Len(txtTiempoSimulacion.Text)
```

```
End If
```

```
IblTiempoMuestreo.Caption = CStr(CSng(txtTiempoSimulacion.Text) / NUMmAXpUNTOS)
End Sub
```

```
'Dato del Control PD
```

```
Private Sub txtWn_LostFocus()
```

```
ChequeoDatos txtWn
If (CSng(txtWn.Text) <= 0) Then
    'Si el valor ingresado de la frecuencia no amortiguada debe
    'ser negativo
    msg = "La frecuencia no amortiguada 'Wn' debe" _
        & " ser positivo y diferente de cero "
    Mostrar
    txtWn.Text = Format(PD_Wn_CONST, "0.00")
    txtWn.SetFocus
    txtWn.SelStart = 0
    txtWn.SelLength = Len(txtWn.Text)
```

```
End If
End Sub
```