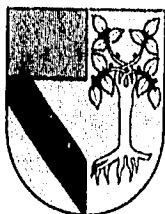


308917  
49  
24



# UNIVERSIDAD PANAMERICANA

ESCUELA DE INGENIERIA

CON ESTUDIOS INCORPORADOS A LA UNIVERSIDAD  
NACIONAL AUTONOMA DE MEXICO

SIMULACION POR COMPUTADORA DE LOS  
SISTEMAS DE CONTROL DIGITALES

## T E S I S

QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA  
AREA INGENIERIA MECANICA  
P R E S E N T A :  
EDGAR QUINTO CAMPOS

DIRECTOR: DR. STANISLAW RACZYNSKI GAWIN

TESIS CON  
FALLA DE ORIGEN

MEXICO, D. F.

1996



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# INDICE

<b>INTRODUCCIÓN</b> .....	3
<b>1. SIMULACIÓN</b> .....	7
1.1 QUÉ ES LA SIMULACIÓN	
1.2 MODELADO DE SISTEMAS	
¿Por qué modelar?	
Elección del tipo de modelo	
Principios usados en el modelado	
1.3 BREVE RESEÑA HISTÓRICA	
Evolución de la tecnología de simulación	
1.4 UTILIZACIÓN DE LA TECNOLOGÍA EN LA MANUFACTURA	
1.5 INGENIERÍA CONCURRENTE	
1.6 LENGUAJES DE SIMULACIÓN	
Lenguaje de simulación PASION	
<b>2. ACCIONES BÁSICAS DE CONTROL</b> .....	21
2.1 INTRODUCCIÓN	
2.2 ACCIONES BÁSICAS DE CONTROL	
<b>3. SISTEMAS LINEALES Y NO LINEALES</b> .....	31
3.1 MODELO MATEMÁTICO	
3.1.1 Representación abstracta de los sistemas físicos	
3.1.2 Simplicidad vs. Exactitud.	
3.2 SISTEMAS LINEALES	
3.2.1 Ventajas de los sistemas lineales.	
3.2.2 Clasificación de las ecuaciones diferenciales.	
3.3 SISTEMAS NO LINEALES	
<b>4. SISTEMAS DE TIEMPO DISCRETO</b> .....	35
4.1 INTRODUCCIÓN	
4.2 TRANSFORMADA Z	
4.3 UN EJEMPLO DE ANÁLISIS DE ESTABILIDAD	
<b>5. SIMULACIÓN UTILIZANDO UN MODELO DEL VEHÍCULO PARA UNA SOLA LLANTA</b> .....	47
5.1 DESCRIPCIÓN DE LA INTERACCIÓN ENTRE EL SUELO Y LAS LLANTAS DE UN VEHÍCULO	
Funciones que desempeñan los neumáticos	
5.1.1 Llantas con cuerdas radiales y cuerdas diagonales	

- 5.1.2 Momentos y fuerzas que actúan sobre una llanta
- 5.1.3 Resistencia a la rodadura de las llantas.
- 5.1.4 Factores que afectan la resistencia a la rodadura
- 5.2 COEFICIENTE DE FRICCIÓN EN FUNCIÓN DEL DESLIZAMIENTO
- 5.2.1 Para suelo seco
- 5.2.2 Para suelo mojado
- 5.2.3 Para suelo con hielo

### 5.3 MODELO SIMPLIFICADO UTILIZADO EN LA SIMULACIÓN

- 5.3.1 Objetivos de la simulación
- 5.3.2 Dinámica de la rueda
- 5.3.3 Dinámica del vehículo
- 5.3.4 Relación entre la rueda y el vehículo
- 5.3.5 Reogramas

- 5.3.5.1 Qué son los reogramas

- 5.3.5.2 Reograma del sistema físico

### 5.4 CÁLCULOS NECESARIOS PARA DETERMINAR LOS VALORES NUMÉRICOS EN UN MODELO DE PARÁMETROS CONCENTRADOS

- 5.4.1 Cálculo de las inercias rotacionales equivalentes
- 5.4.2 Fuerzas aerodinámicas

### 5.5 PROCEDIMIENTOS ESPECIALES UTILIZADOS EN EL PROGRAMA

## 6. SIMULACIÓN DEL CONTROLADOR PARA EL SISTEMA DE FRENOS ABS ..... 81

- 6.1 ALGORITMO DE CONTROL
- 6.2 REOGRAMA DEL SISTEMA DE CONTROL
- 6.3 PARÁMETROS QUE SE UTILIZAN
- 6.4 MEJORAMIENTO DEL ÍNDICE DE DESEMPEÑO
- 6.4.1 Descripción de las pruebas
- 6.4.2 Prueba # 1
- 6.4.3 Prueba # 2
- 6.4.4 Prueba # 3
- 6.4.5 Parámetros de mayor influencia
- 6.5 OBTENCIÓN DE LA SUPERFICIE DE RESPUESTA
- 6.6 INFLUENCIA DE LAS PERTURBACIONES ALEATORIAS

<b>7. AUTOMATIZACIÓN DEL PROCESO ITERATIVO DE SIMULACIÓN</b> .....	99
7.1 NECESIDAD DE AUTOMATIZAR PROCESOS	
7.2 ELEMENTOS DEL PROCESO ITERATIVO DE SIMULACIÓN	
7.3 OBTENCIÓN AUTOMATIZADA DE LA SUPERFICIE DE RESPUESTA	
7.4 EL USO DE UN ALGORITMO DE OPTIMIZACIÓN	
7.5 EL CASO DEL CONTROL DE FRENOS ABS	
<b>8. CONCLUSIONES</b> .....	111
<b>BIBLIOGRAFÍA</b> .....	117
<b>APÉNDICES</b> .....	123
APÉNDICE 1	
APÉNDICE 2	
APÉNDICE 3	
APÉNDICE 4	
APÉNDICE 5	
APÉNDICE 6	

**TESIS**

**COMPLETA**

# INTRODUCCIÓN

## INTRODUCCIÓN

El avance de la ciencia ha utilizado a las matemáticas como una poderosa herramienta de análisis y solución de problemas con los que se ha enfrentado el hombre a lo largo de su historia en su afán por un mejor aprovechamiento de los recursos humanos y materiales.

Sin embargo, en muchas ocasiones ha sido necesaria la recopilación de información empírica, que no puede obtenerse de alguna otra manera, dado la complejidad y/o inexactitud de los modelos matemáticos que describen ciertos procesos. Esta adquisición de información de manera empírica, si bien acerca al hombre al conocimiento de leyes que describen (aunque no explican) el comportamiento de un sistema físico, requieren de mucho tiempo y de una gran inversión económica para llevarse a cabo.

Es en esos casos en los que el análisis es complejo o inoperante en los que resulta valioso auxiliarse de medios relativamente modernos como la computadora digital, que son capaces de realizar operaciones que al hombre le tomarían mucho más tiempo, con los que se pueden afrontar problemas reales, de alto nivel de complejidad, que de otra forma serían muy difíciles de resolver de manera satisfactoria, empleando una cantidad de recursos razonable. Una de las aplicaciones de la computadora electrónica en la ingeniería actual, es la simulación de procesos físicos que encontramos en la industria, con el objeto de, en este caso, encontrar controladores digitales junto con sus parámetros óptimos, que permitan un control adecuado de aquéllos.

El empleo de este tipo de simulación en especial facilita la instalación y modificación en los sistemas de control digitales que operan en plantas que, por su naturaleza, no admitirían paros prolongados (o ninguno, incluso) o realización de pruebas costosas. O bien, la simulación como parte del ciclo de diseño puede disminuir el tiempo de desarrollo de un producto de manera significativa. Así pues, se pretende que mediante el uso de la simulación por computadora de dichos controladores digitales, se facilite la labor del ingeniero de control en la industria que utiliza este tipo de sistemas de control automático.

En el primer capítulo de la tesis se habla de la simulación y se presenta un marco histórico que trata de describir la necesidad de dicha técnica como una herramienta valiosa en varios campos de las ciencias y disciplinas sociales. Además se describe brevemente el módulo CMG de PASION, que es el software que ha sido utilizado a lo largo de esta tesis.

En el segundo capítulo se describen las acciones básicas de control y las ventajas de cada una de ellas.



En el tercer capítulo se hace una distinción entre los sistemas lineales y los no lineales y las diferencias en las herramientas necesarias en el análisis de cada caso. Se plantea la necesidad del uso de la simulación sobre todo en el análisis de sistemas no lineales.

En el capítulo cuatro se habla de la transformada Z y se utiliza para describir las regiones de estabilidad de un sistema de control digital para una planta de primer orden.

Como caso práctico se escogió un controlador digital de un sistema de frenos ABS de un automóvil de 2 ejes.

En el capítulo cinco se describe el modelo físico simplificado del automóvil que se usó en este trabajo y los principios físicos en que se basó dicho modelo. Además se describen las funciones que se emplearon en el modelo del sistema físico.

En el capítulo sexto se añade al modelo el control digital y se expone el algoritmo utilizado. En este mismo capítulo se realizan pruebas para mejorar los parámetros de dicho control, primero, de manera gruesa y después, de manera fina, encontrando una superficie de respuesta de los parámetros más importantes, para así poder caracterizar el comportamiento de este sistema de manera gráfica. Además se describe el efecto que tienen en el sistema completo los factores de carácter aleatorio.

En el capítulo séptimo se proponen algunos métodos para la automatización del proceso iterativo de simulación-mejora. La eficacia y factibilidad del empleo de estos métodos dependen de la naturaleza de cada sistema.

Finalmente se presentan de manera sintética las conclusiones del trabajo. En los apéndices se incluyen los programas que fue necesario desarrollar para el capítulo 6 y 7.

# **CAPÍTULO 1**

# 1. SIMULACIÓN

## 1.1 Qué es la simulación.

Simular significa "reproducir artificialmente un fenómeno o las relaciones entrada-salida de un sistema. Cuando esta reproducción está basada en la ejecución de un programa en una computadora digital se denomina simulación por computadora".

Al intentar simular un fenómeno, es conveniente utilizar el enfoque de sistema, entendiendo por sistema como el conjunto o arreglo de objetos unidos mediante una interacción regular o interdependencia. Esta definición es muy amplia, pues abarca desde sistemas físicos, hasta sistemas de producción de una fábrica.

Quando se trata de sistemas físicos es más conveniente considerar un sistema como el conjunto de componentes interconectados en el que existe un conjunto específico de variables dinámicas llamadas entradas y un conjunto dependiente, llamado salidas (o respuestas).

## 1.2 Modelado de sistemas.

### 1.2.1 ¿Por qué modelar?

Muchas veces se necesita adquirir información acerca de un determinado sistema:

- 1) Para poder predecir su comportamiento antes de construirlo, o bajo condiciones que no han sido experimentadas ó
- 2) Para explicar su comportamiento.

Por ejemplo: el sistema económico, el comportamiento de un grupo

---

<sup>1</sup>Cfr. RACZYNSKI, Stanislaw.  
Simulación Por Computadora.  
México. Grupo Noriega Editores. 1993 p 9.

<sup>2</sup>Cfr. GORDON, Geoffrey.  
Systems Simulation.  
New Jersey. Prentice Hall. 1978. p. 1.

<sup>3</sup>Cfr. COCHIN, Ira.  
Analysis and Design of Dynamic Systems.  
New Jersey. Harper & Row Publishers. 1980. p. 1.

social, el comportamiento de un avión o de una planta industrial.

Para lograr esto, existen varias maneras. Una de ellas es experimentar con el sistema mismo. Esto muchas veces resulta impráctico o bastante caro. Otra manera es elaborar un modelo del sistema del que se requiere la información, y realizar los experimentos necesarios en este modelo.

#### 1.2.2 Elección del tipo de modelo.

Ahora bien, este modelo puede ser físico o matemático. En el presente trabajo se emplean modelos matemáticos para representar situaciones físicas reales.

Modelo es una forma de representar el sistema.

El tipo de modelo que se emplee depende, más que de la naturaleza misma del sistema, del tipo de descripción que va a hacerse sobre el sistema.

No siempre necesitamos saber todo lo que se pueda saber acerca de un sistema y para los propósitos que persigue un estudio determinado, no se necesita saber todos los detalles.

Aunque no existen reglas para elaborar los modelos, si hay principios generales que nos pueden orientar en la elección del nivel de exactitud en la representación: el propósito del modelo y el principio de que el modelo no debe ser más complicado de lo necesario. El modelo no solamente es un sustituto del sistema (en los experimentos) sino también una simplificación del mismo.

Se puede definir modelo como el cuerpo de información acerca del sistema reunida con el propósito de estudiar el sistema. Esta información, en los casos de sistemas físicos, debe incluir el comportamiento dinámico del sistema en el tiempo, o bien, la manera como cambia de un estado a otro, conociendo claramente las reglas que rigen estos cambios.

Como el propósito del estudio determina la clase de información que debemos reunir, puede haber varios modelos diferentes para un mismo sistema, dependiendo del enfoque del estudio a realizar.

#### 1.2.3 Principios usados en el modelado:

Lo siguiente no son pasos a seguir en el modelado, sino recomendaciones para poder evaluar el modelo desde distintos puntos de vista.

##### 1.2.3.1 Construcción con bloques.

La descripción del sistema debe estar organizada en una serie de bloques. Esto ayuda a describir las interacciones dentro del sistema, ya que hay partes del sistema que, teniendo sus

variables de entrada, producen una o unas cuantas variables de salida. Así, el sistema puede ser descrito mediante bloques interconectados.

#### 1.2.3.2 Relevancia.

Dependiendo del objetivo del estudio, ciertos aspectos del sistema son relevantes y otros no. Los últimos, no deben ser incluidos en el modelo, ya que lo hacen innecesariamente más complicado.

#### 1.2.3.3 Exactitud.

Dependiendo del aspecto del sistema que se quiera estudiar, hay información que debe conocerse con mayor o menor exactitud. A veces será necesario considerar una característica del sistema, con la mayor exactitud posible; otras, no.

#### 1.2.3.4 Agrupamiento.

Algunas veces no es necesario saber el comportamiento de un subsistema, sino el de un conjunto de ellos. Esto también ayuda a simplificar el modelo.

### 1.3 BREVE RESEÑA HISTÓRICA.

#### EVOLUCIÓN DE LA TECNOLOGÍA DE SIMULACIÓN.

El desarrollo tecnológico dentro del campo de la simulación está relacionado necesariamente con el del hardware y software disponible. El rol que ha jugado la simulación dentro de la tecnología de diseño y manufactura, por ende, ha ido aumentando hasta convertirse en indispensable para la competitividad en industrias en las que el tiempo de desarrollo de un producto o un proceso es crítico.

En cada generación el proceso y el tiempo de ciclo de diseño, así como la complejidad de los sistemas que se diseñan es diferente, ya que se hace uso de herramientas tecnológicas cada vez más avanzadas.

#### 1ª Generación.

Durante los años 40, con la aparición de la computadora electrónica se pudieron tener aplicaciones sobre trayectorias de proyectiles militares. Sin embargo, las personas dedicadas a la simulación todavía estaban muy limitadas por la gran dificultad de desarrollar software de gran escala con un lenguaje de bajo

nivel.

## **2ª Generación.**

Aparecen los lenguajes de mayor nivel, como el FORTRAN, y se realizan simulaciones dinámicas -se refiere al proceso de conducir experimentos para ganar entendimiento acerca del comportamiento de un sistema -. La base de usuarios crece de manera importante: científicos e ingenieros de varias disciplinas, desde físicos hasta biólogos.

## **3ª Generación.**

A fines de los 60's se desarrollan los lenguajes especializados para simular sistemas continuos, como CSSL y ACSL. Estos lenguajes generaban el código en FORTRAN, de manera que el usuario ya no tenía que programar. Como consecuencia, las herramientas de simulación fueron accesibles a un número cada vez mayor de personas.

## **4ª Generación.**

A mediados de los ochentas las compañías re-escribieron su *software* para estaciones de trabajo con el objeto de aprovechar los distintos recursos, como ventanas múltiples y las capacidades gráficas inherentes a dichas plataformas. Surgen entonces lenguajes como Systembuild y Easy 5 (1986) para estaciones de trabajo.

Los procesos a ser simulados se definen en forma gráfica, mediante diagramas de bloques -en los cuales cada línea que une a los bloques representa un flujo de señal y cada bloque un proceso- y cada proceso se define matemáticamente utilizando transformadas de Laplace, ecuaciones diferenciales, matrices de espacio-estado, funciones algebraicas generales o trascendentes. Una característica importante es la posibilidad de agrupar varios elementos del modelo representados por bloques en un "superbloque", que a su vez puede ser agregado a otro modelo. Esto permitía la creación de modelos de subsistemas o componentes.

Nacen las estructuras jerárquicas de datos, que permiten crear compartir componentes entre varios usuarios. De esta manera se podían hacer análisis de sensibilidad y de intercambiabilidad al sustituir varios componentes. Se hace necesaria entonces una herramienta que permita el análisis de los cuantiosos datos producidos mediante simulaciones sofisticadas. Esto hace que surjan programas como Xmath, MATRIX y MATHLAB.

Esto es un fenómeno reciente. Trajo los mismos beneficios que obtuvo la comunidad de programadores con el uso de lenguajes orientados a objetos como el C++: las funciones eran más

generales y fáciles de modificar, más compactas y se podían manejar muchos tipos de datos al ocultarse y encapsularse .

Algunas de las ventajas son que las herramientas permiten al usuario:

- Interpretar los resultados de muchas corridas del programa de simulación mediante funciones graficadoras sofisticadas e interactivas.
- Visualizar y entender las respuestas de los sistemas cambiando las entradas y algunos parámetros internos.
- Permiten manipular los datos obtenidos de la simulación, ganando una visión adicional del comportamiento del sistema, por ejemplo, determinar la frecuencia de un ciclo límite.

#### **5ª Generación. Tecnologías múltiples.**

Las generaciones pasadas se habían caracterizado por basarse en una innovación tecnológica importante: la aparición de las computadoras digitales, el desarrollo de lenguajes de mayor nivel, la creación de lenguajes especializados para simulación y a orientación a objetos de la 4ª generación. En cambio, la quinta generación se caracteriza por hacer uso de múltiples tecnologías. Se redefine el rol de la simulación en el proceso de diseño.

Anteriormente ya se usaba software especializado para el diseño de sistemas electromecánicos complejos, como:

- robots
- vehículos
- aviones
- naves espaciales

En esta generación existe una mayor interacción entre el software generador de modelos para simular y el software relacionado con análisis de ingeniería, como programas de FEA (Finite Element Analysis), Cinemática y Dinámica.

Una vez que se ha llegado a algún resultado, se puede hacer una extensión del uso de la información obtenida para ser usada en la siguiente etapa: Prototipo y prueba.

Anteriormente no se hacía simulación en tiempo real y no se permitía la integración de bancos de pruebas. Ya en esta generación se permite la simulación en tiempo real, gracias a avances en mejorar los procesadores, a correr programas con multiprocesos, obtención de algoritmos rápidos para integración, etc.

Por lo tanto, debido a la conjunción de elementos se ha podido re-definir el papel de la simulación en el proceso de diseño, haciéndola parte esencial del mismo.

Estos elementos son:

- Se introdujeron lenguajes para ecuaciones diferenciales e interfaces gráficas que aceptan diagramas de bloques, por ejemplo, que facilitan la creación de modelos y hacen que éstos sean más accesibles a un mayor número de personas.
- Se incluyen elementos aleatorios como parte de los modelos, de tal forma que éstos puedan ser más realistas. Por ejemplo, el programa ADAPT<sub>x</sub> de Adaptics Inc., realiza análisis de variables aleatorias para caracterizar el sistema que se va a modelar, y genera un modelo estadísticamente óptimo que considera estos factores aleatorios.
- La simulación no se restringe a modelos lineales, de tal forma que los resultados obtenidos serán más exactos y confiables.
- Se aprovecha la información utilizada durante la simulación para su uso en el ámbito de las pruebas y de la elaboración del prototipo.

Es en esta generación que se alcanzan niveles suficientemente buenos como para integrar a las etapas de diseño y desarrollo al proceso de manufactura, trabajando de manera paralela en muchas tareas, en un ambiente de ingeniería concurrente.

#### 1.4 INGENIERÍA CONCURRENTE.

El objetivo de la ingeniería concurrente es la adopción de nuevas tecnologías para resolver problemas en cascada, haciendo énfasis en la capacitación y el trabajo en equipo. La esencia de la ingeniería concurrente es la colaboración trans-funcional, que también incluye actividades no ingenieriles.

Para poder ser competitivos, es necesario que las personas involucradas en el diseño y la manufactura de los productos desarrollen 2 capacidades básicas: a) Poder entender los requerimientos del cliente suficientemente como para traducirlos en requerimientos exactos de diseño. b) Poder entender los medios de mejora continua para lograr diseños robustos que alcancen variaciones conjugadas dentro de los requerimientos de variación.

Para poder adaptarse a nuevas tecnologías es necesario que exista un entrenamiento fuerte en operaciones multifuncionales y relaciones interpersonales, de tal forma que existan habilidades de comunicación inteligente entre las personas de un equipo de trabajo y entre los diferentes equipos de trabajo, de tal manera que, conociendo de manera exacta las características, especificaciones y procesos de producción de los productos, se pueda avanzar de manera continua y siempre integrada en el uso de las herramientas como: CAD, CAM, CIE, CIM y no solamente en el desarrollo de herramientas per se.



El desarrollo de la simulación, como parte esencial del ciclo de diseño, hace posible el trabajo ingenieril de manera concurrente.

## 1.5 LENGUAJES DE SIMULACIÓN.

### 1.5.1 Lenguaje PASION.

El lenguaje PASION es un conjunto de programas relacionados entre sí para poder generar los objetos necesarios para llevar a cabo simulaciones de eventos discretos y de procesos continuos. Este lenguaje fue desarrollado por el Dr. Stanislaw Raczynski en la Universidad Panamericana, con fines educativos, pero también cuenta con versiones profesionales.

#### 1.5.1.1 Módulos que lo componen.

Este lenguaje consta de varios módulos que son:

QMG: "Queuing Model Generator" (Generador de Modelos de Colas).

BGS: "Bond Graph Simulator" (Simulador de Gráficos Bond).

PASION ANIMATOR PAM6. Es el generador de secuencias animadas para procesos discretos.

CMG: "Continuous Model Generator" (Generador de Modelos Continuos).

#### 1.5.1.2 Generación del modelo utilizando el módulo CMG6 .

El módulo CMG6 (Continuous Model Generator) es un programa que es capaz de generar programas objeto (código), a partir de una representación de los sistemas a simular por medio de reogramas.<sup>4</sup> Los reogramas están formados por nodos y ramas. Los nodos representan señales y las ramas funciones de transferencia. En cada reograma se deberá especificar la función de transferencia de cada rama. En todos los tipos de ramas se pueden incluir parámetros  $P[i]$  -con  $i \in$  {conjunto de números enteros}-, cuyo valor puede cambiarse cada vez que se ejecuta el programa de simulación y almacenarse en archivos.

En cada función de transferencia se aceptan los siguientes tipos:

- Rama estática lineal.

Esta rama describe una relación simple entre la entrada y la salida de una función  $y=kx$ , donde  $y$  es la señal de salida,  $x$  la de la entrada y  $k$  es la ganancia.  $k$  puede ser una función definida por el usuario y codificada en Pascal, una constante o un parámetro.

---

<sup>4</sup>Ver inciso 5.3.5.

Rama estática no lineal.  
En esta rama se pueden incluir relaciones de entrada y salida de tipo  $y=f(x)$ , donde  $f(x)$  puede ser una función definida por el usuario -quien deberá suministrar el código de dicha función en Pascal - o bien una de las funciones pre-definidas suministradas por PASION, por ejemplo :

- Función de saturación (ver gráfico 1.6.1).
- Función con zona muerta (ver gráfico 1.6.2).
- Función switch.
- Función de saturación con histéresis y zona muerta (ver gráfico 1.6.3).
- Función de saturación con histéresis
- Función de saturación 2 (descrita con arcotangente)
- Función con zona muerta negativa
- Función afin (recta que no necesariamente pasa por el origen)

Rama dinámica.

En este tipo de rama se aceptan relaciones entre las variables de los nodos, empleando constantes, parámetros y el operador de Laplace, "s". Esta función de transferencia es un cociente: el numerador es la transformada de Laplace de la función de salida y el denominador es la transformada de

FUNCIÓN DE SATURACIÓN

NIVEL DE  
SATURACIÓN

Gráfico 1.6.1

FUNCIÓN CON ZONA MUERTA

ZONA MUERTA

Gráfico 1.6.2

FUNCIÓN DE SATURACIÓN CON HISTÉRESIS Y ZONA MUERTA

Gráfico 1.6.3

Laplace de la función de entrada. Es decir, esta rama representa la función de transferencia equivalente a la ecuación diferencial ordinaria lineal de la forma:

$$\begin{aligned} a_n x^{(n)} + a_{n-1} x^{(n-1)} + \dots + a_1 dx/dt + a_0 x(t) &= \\ = b_m y^{(m)} + b_{m-1} y^{(m-1)} + \dots + b_1 dy/dt + b_0 y(t), \end{aligned}$$

que tendrá la siguiente función de transferencia:

$$G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}$$

- Rama de tipo función. Esta función acepta variables que se usen en programa y debe ser definida por el usuario y codificada en Pascal. La función es monovariante.
- Retraso: sirve para representar la relación entrada-salida de la función:  $y=x(t-a)$ , donde  $y$  es la señal de salida,  $x$  la de entrada y  $a$  es el retraso, que debe ser una constante (o un parámetro, pero no una función).
- "Sample and Hold" (muestreo y retención): Esta rama "lee" la señal de entrada en momentos discretos de tiempo y la mantiene constante como señal de salida hasta el siguiente momento de muestreo. Es necesaria para simular sistemas de control de tiempo discreto.
- Superlink: Un modelo completo puede ser almacenado como una sola rama ("superlink"). Sin embargo, deberá respetar los siguientes lineamientos:
  - a) Solo deberá tener un nodo de entrada y uno de salida.
  - b) Deberá especificarse en qué archivo se encuentra.
  - c) Otro modelo puede usar cualquier número de superlinks, pero sólo se permite un nivel de anidamiento, es decir, un modelo normal puede contener un superlink, pero un superlink no puede contener otro superlink.

Respecto a los nodos, pueden ser de 2 tipos:

- Nodos convencionales, cuya salida será simplemente la suma de las señales de entrada.

- Nodo de tipo función. Esta función podrá ser definida por el usuario, quien deberá incluir su código en Pascal. La ventaja de este tipo de nodos sobre las ramas de tipo función es que los primeros pueden tener múltiples entradas y una sola señal de salida, mientras que las segundas son monovariables.

El programa, además de utilizar la información contenida en el reograma y en las funciones de transferencia, solicita información acerca de lo siguiente:

- Si la simulación es de tipo estocástico o determinístico.  
El tipo **estocástico** se utiliza cuando se necesitan entradas o eventos aleatorios. En estos casos, los resultados de las simulaciones pueden ser analizados por medio del programa VARANS, que calcula trayectorias promedio, conjuntos alcanzables, varianzas e intervalos de confianza para cada variable inclusive.  
El tipo **determinístico** incluye los procesos que no aceptan entradas aleatorias, sino que la relación entre todas las variables es fija y predecible.
- Nombre de los archivos que contienen las funciones utilizadas por el usuario u otros procedimientos y funciones utilizadas en el modelo.
- Nombre del procedimiento de inicio del programa.
- El nombre de los nodos que constituyen las diferentes salidas que el usuario quiere ver simuladas.

### 1.5.3 Ejecución de la simulación.

Una vez obtenido el modelo, es decir el programa objeto que generó el módulo CMG de PASION, al ejecutarse nos muestra la simulación del proceso definido por el modelo previamente creado.

Durante la ejecución de este programa, el usuario debe proporcionar la siguiente información:

- tiempo que dura la simulación
- Paso de integración
- Método de integración, que puede ser Runge-Kutta de orden 2 ó Runge-Kutta-Fehlberg de orden 5
- Valor de los parámetros definidos por el usuario, ya sea vía teclado o mediante un archivo de datos.
- Si se requiere archivo con la gráfica de baja resolución.
- Si se requiere gráfica en el plano de fase y de qué variable.
- Si los resultados se van a presentar de manera gráfica o numérica.
- En caso de escoger "gráfica" en el inciso anterior, se pide la escala, que puede ser la misma para todas las salidas o individual.

Posibilidades de interacción con el usuario.

Durante la ejecución del programa se puede interrumpir para cambiar el valor de alguno o algunos de los parámetros definidos por el usuario. Basta con oprimir cualquier tecla para que se detenga el proceso y se pueda cambiar el valor de algún parámetro. Después de hacerlo, se puede continuar con la simulación o volver a empezarla.

Análisis post-proceso.

Una vez que el programa terminó de ejecutarse los resultados son escritos en un archivo "results". En caso de salida gráfica, se muestra una gráfica para cada variable de salida. Es después de la obtención de los resultados que se puede realizar un análisis postproceso, mediante una hoja de cálculo o mediante el programa VARANS, en caso de utilizar simulaciones estocásticas.

## **CAPÍTULO 2**

## 2. ACCIONES BÁSICAS DE CONTROL

### 2.1 Introducción

Controlar significa hacer coincidir dos magnitudes: una real y otra de referencia. En la industria existen muchas magnitudes que se necesita controlar, ya que cambian constantemente: temperatura, presión, posición de un objeto, velocidad de un motor, pH de una substancia, etc.

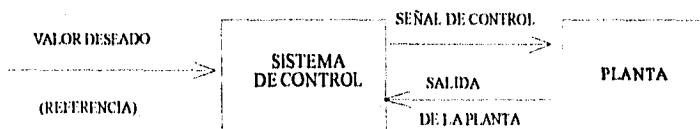


Fig. 2.1. Relación entre un sistema de control y una planta.<sup>1</sup>

En el contexto del control automático, generalmente la magnitud a controlar se considera como la salida del sistema (planta). Para poder controlar la salida de un sistema es necesario compararla con una magnitud de referencia y así poder determinar qué tan lejos está una de otra, o qué tan grande es el error. Pero, ¿Cuál es el origen de los errores o diferencias entre la salida de un sistema y su valor de referencia?

Los sistemas físicos en los cuales podemos observar estas magnitudes físicas a ser controladas, normalmente están sujetos a condiciones internas o externas, que afectan de manera adversa el valor de salida del sistema. Estos factores se llaman perturbaciones, y pueden ser internos o externos, predecibles o

---

<sup>1</sup> En control automático se considera como planta cualquier sistema que se quiere controlar.

aleatorios. Estas perturbaciones causan una desviación de la salida del sistema con respecto a la referencia de control o valor deseado, es decir, producen un error (Ver Fig. 2.1).

Un sistema de control se encarga de corregir ese error. Para hacerlo, el sistema de control produce señales de control. La señal de control evidentemente depende del error, pero la forma en que aquélla se relaciona con éste, depende de la acción de control, es decir, de la manera como se toma en cuenta el error para producir la señal de control.

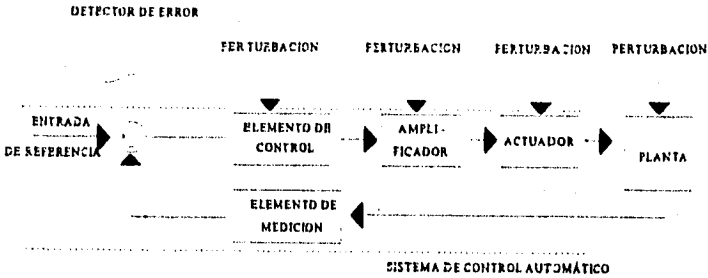


Fig. 2.2 Partes del sistema de control sujeto a diversas perturbaciones.

### 2.1.2 Clasificación de los controles automáticos por su acción de control

De acuerdo a su acción de control los controles automáticos se pueden clasificar en 6:

- Controles de 2 posiciones o de si-no
- Controles proporcionales
- Controles integrales
- Controles proporcionales e integrales
- Controles proporcionales y derivados
- Controles proporcionales, derivados e integrales



### 2.1.2.1 Acción de 2 posiciones.

En el controlador de 2 posiciones, sólo existen dos valores posibles, generalmente encendido y apagado. Sea la señal de salida del control  $m(t)$  y el error actuante  $e(t)$ . Entonces:

$$\begin{aligned} m(t) &= M1 \text{ si } e(t) > 0, \\ m(t) &= M2 \text{ si } e(t) < 0. \end{aligned}$$

Brecha diferencial

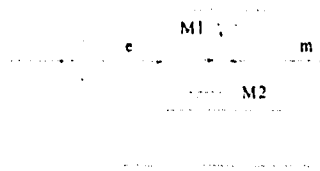


Fig. 2.3

Generalmente  $M1$  es el valor máximo y  $M2=0$  ó  $M2$  es un valor pequeño, o  $M2=-M1$ .

Normalmente, los controladores de 2 posiciones funcionan en realidad de esta otra forma:

$$m(t) = M1 \text{ si } e(t) > 0 + \Delta e,$$

$$m(t) = M2 \text{ si } e(t) < 0,$$

de tal forma que existe una "brecha diferencial" ( ver Fig. 2.3) de ancho  $\Delta e$ , que puede ser inevitable o puede existir de manera intencionada. Esta brecha provoca que se presenten oscilaciones en la acción de control, que serán más frecuentes, cuanto menor sea  $\Delta e$ , y tendrán mayor amplitud mientras mayor sea  $\Delta e$ .

### 2.1.2.2 Acción de control proporcional

En un control de acción proporcional, la relación entre la señal del control y el error, es la siguiente:

$$\frac{M(s)}{E(s)} = K_p$$

donde  $K_p$  se denomina sensibilidad proporcional o ganancia. Básicamente es un amplificador con ganancia ajustable.

### 2.1.2.3 Acción de control integral.

En la acción integral la salida del controlador es directamente proporcional al error acumulado:

$$\frac{M(s)}{E(s)} = \frac{K_p}{s}$$

#### 2.1.2.4 Acción de control proporcional e integral

La acción de control proporcional e integral se puede describir de la siguiente manera:

$$\frac{M(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} \right)$$

donde  $K_p$  representa la sensibilidad proporcional o ganancia y  $T_i$  el tiempo integral. Ambos son regulables.  $T_i$  es el tiempo que tarda en duplicarse la señal de control ante un error en forma de función escalón.

#### 2.1.2.4 Acción de control proporcional y derivada.

La relación entre  $M(s)$  y  $E(s)$  es:

$$\frac{M(s)}{E(s)} = K_p (1 + T_d s)$$

donde  $K_p$  es la sensibilidad proporcional y  $T_d$  el tiempo derivativo. En este tipo de acción la señal del control es proporcional a la rapidez con que cambia el error.

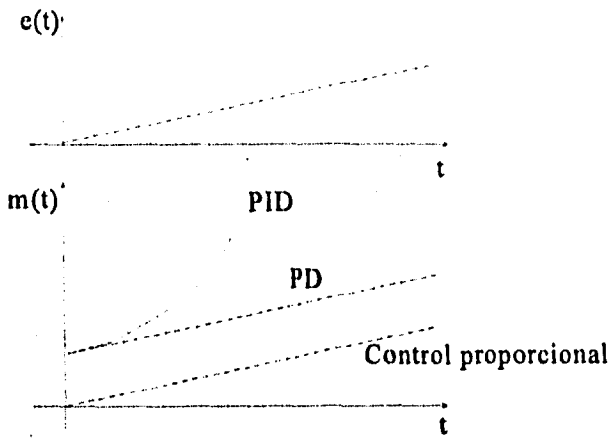
Como en la acción integral y proporcional, la constante  $K_p$  afecta a ambas partes. La constante  $T_d$  es el intervalo de tiempo en el que la acción derivativa se adelanta a la señal de control que se produciría si solamente hubiera el efecto proporcional.

#### 2.1.2.5 Acción de control proporcional, derivada e integral.

En esta modalidad se tienen las ventajas de las acciones anteriores. La función de transferencia es:

$$\frac{M(s)}{E(s)} = K_p \left( 1 + T_d s + \frac{1}{T_i s} \right)$$

A continuación se muestra una grafica en la que se comparan los 3 tipos de acción de control ante una función de error de tipo rampa:



# **CAPÍTULO 3**

## 3. SISTEMAS LINEALES Y NO LINEALES.

### 3.1 Modelo matemático.

#### 3.1.1 Representación abstracta.

Una de las maneras de representar de manera abstracta los sistemas dinámicos, es mediante las ecuaciones diferenciales. Al caracterizar un sistema por una ecuación diferencial, se puede conocer su respuesta al aplicar determinada entrada (o función excitadora), si se puede resolver la ecuación diferencial asociada a dicho sistema.

El modelo matemático que se escoja para analizar un determinado sistema físico depende del tipo de análisis que se vaya a hacer. Por ejemplo, si busca un sistema de control óptimo, es mejor describirlo utilizando un conjunto de ecuaciones diferenciales de primer orden. Si se va a analizar la respuesta a la frecuencia de un sistema con una sola entrada y una sola salida, lo mejor es describirlo en términos de funciones de transferencia.

#### 3.1.2 Simplicidad vs. exactitud.

Al hacer un análisis de un sistema dinámico, se debe determinar antes si se trata de un examen cualitativo o cuantitativo, cuáles son los requerimientos de exactitud del análisis y su objetivo principal. Se debe hacer un compromiso entre exactitud y simplicidad, ya que puede suceder que un modelo se acerque más a la realidad que otro, pero sea muy difícil de resolver.

Los factores que deben observarse:

- **Rango de operación.** Si solamente se necesita conocer el desempeño de un sistema dentro de un rango de operación pequeño, entonces puede ser suficiente obtener un modelo simplificado, en el que las pérdidas de exactitud al eliminar características como parámetros distribuidos y alinealidades no sean de gran importancia.
- **Frecuencia de operación.** Puede haber un modelo que describa bien el sistema a bajas frecuencias pero no a altas. Esto suele suceder con modelos de parámetros concentrados. Cuando se alcanzan frecuencias en las que son insuficientes dichos modelos, se podrá entonces utilizar modelos de parámetros distribuidos, que describen de manera más completa al sistema.

### 3.2 Sistemas lineales.

Una representación de un sistema físico de manera lineal presenta ventajas. A continuación se define qué son los sistemas lineales y se explican sus ventajas. Después se presenta una breve

clasificación de las ecuaciones diferenciales para poder determinar cuáles son las ecuaciones diferenciales lineales, que son las que se utilizan para representar los sistemas lineales.

### 3.2.1 Ventajas de los sistemas lineales.

La principal ventaja de los sistemas lineales es que se puede aplicar, en su análisis, el principio de superposición. En el contexto de los sistemas dinámicos, este principio establece que si un sistema es excitado, de manera separada, por dos o más funciones excitadoras al mismo tiempo, la respuesta del sistema será equivalente a sumar la respuesta que tiene el sistema a cada entrada por separado.

De esta manera, si se puede expresar una función excitadora como una combinación lineal de funciones excitadoras conocidas, entonces la respuesta del sistema se puede encontrar fácilmente si se suman las respuestas a dichas "funciones básicas". Y no sólo eso: también se puede analizar la respuesta del sistema no solamente a una sola entrada, sino a varias, por ejemplo, un sistema de control con disturbios en distintos puntos del sistema.

### 3.2.2 Clasificación de las ecuaciones diferenciales:

Las ecuaciones diferenciales pueden ser ordinarias o parciales. Las primeras se caracterizan por incluir derivadas de funciones que solamente dependen de una variable, por ejemplo, la ecuación del movimiento armónico simple:

$$\frac{d^2y}{dt^2} + \omega^2 y = 0$$

Este tipo de ecuaciones se utilizan para describir sistemas de parámetros concentrados.

Para describir sistemas de parámetros distribuidos, se utilizan las ecuaciones diferenciales parciales, por ejemplo la ecuación de transferencia de calor en una dimensión:

$$k \frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t}$$

Además, de acuerdo al orden de la derivada de mayor orden, se clasifican en ecuaciones diferenciales de primer, segundo o n-ésimo orden. Las dos anteriores son de segundo orden.

También pueden ser lineales o no lineales. Las ecuaciones diferenciales lineales se caracterizan por tener la forma:

$$a_n(x) \frac{d^n y}{dx^n} + a_{n-1}(x) \frac{d^{n-1} y}{dx^{n-1}} + \dots + a_1(x) \frac{dy}{dx} + a_0(x) y = g(x)$$

y reunir dos cualidades básicas:

- La variable dependiente  $y$  y todas sus derivadas tienen exponente 1 (ó 0 cuando no existen).
- Los coeficientes de las derivadas dependen de  $x$  solamente.

Sin embargo, hay que distinguir entre coeficientes constantes o variables, pues se asocian con sistemas invariables o variables en el tiempo, respectivamente.

Se puede decir que los modelos matemáticos de los sistemas dinámicos más sencillos son los que se expresan por medio de ecuaciones diferenciales ordinarias, lineales y de coeficientes constantes.

### 3.3 Sistemas no lineales.

Existen sistemas que incluyen funciones de transferencia que pueden ser funciones algebraicas no lineales.

Si las funciones algebraicas lineales se definen como

$$F(x_1, x_2, x_3, \dots, x_n) = a_1 x_1 + a_2 x_2 + a_3 x_3 + \dots + a_n x_n = b,$$

donde  $a_1, a_2, a_3, a_n$  son constantes,

entonces existen ecuaciones algebraicas no lineales, que son las que no cumplen con las características anteriormente mencionadas. Por ejemplo:

$$y = \text{sen } x$$

$$y = x^3$$

$$z = x^2 + y^2$$

Otros ejemplos de ecuaciones algebraicas no lineales son las funciones de transferencia estáticas no lineales que se utilizan en la generación de modelos del módulo CMG de PASION. Algunas de estas funciones se mencionan en el capítulo I, inciso F.

En el ámbito de las ecuaciones diferenciales las ecuaciones no lineales son las que no cumplen con las condiciones de linealidad establecidas en el inciso B de este capítulo. Ejemplos:

$$y \frac{dy^2}{dx^2} + 2 \frac{dy}{dx} = x$$

y

$$\frac{d^3y}{dx^3} + y^2 = 0$$

Las transformadas de Laplace en forma de polinomios describen ecuaciones diferenciales lineales, cuyo orden es igual al mayor exponente de  $s$ .

Aunque muchos sistemas físicos se describan con ecuaciones diferenciales lineales, en realidad la mayor parte de aquéllos solamente son lineales en un restringido rango de operación. De tal forma que para una descripción del sistema en un rango más amplio será necesario utilizar ecuaciones no lineales, ya sean diferenciales o algebraicas. Por ejemplo, las características de saturación de algunos sistemas, la acción de controladores de tipo si-no, etc.

La principal desventaja matemática de este tipo de sistemas es que no se puede aplicar el principio de superposición y su solución matemática resulta realmente complicada. Es entonces que se puede obtener ventajas de utilizar otros métodos no puramente matemáticos, como la simulación por computadora.



# **CAPÍTULO 4**

## 4. SISTEMAS DE TIEMPO DISCRETO

### 4.1 INTRODUCCIÓN.

Se llama sistemas de tiempo discreto o de datos muestreados a sistemas dinámicos en los cuales una o más variables pueden variar solamente en determinados instantes. Normalmente estos instantes se denotan por  $kT$ , con  $k=1,2,3,\dots$ . Cualquier sistema que incluya una computadora digital como elemento de control es un sistema de tiempo discreto.

En estos instantes se realiza una medición física o se lee la memoria de una computadora digital.  $T$  es el lapso de tiempo entre un instante de muestreo y el siguiente. Si  $T$  es constante la forma de muestreo se denomina periódica. En esta tesis se hablara exclusivamente de muestreos periódicos. La diferencia entre los sistemas de tiempo continuo y de tiempo discreto es que en estos últimos las señales aparecen en forma de datos muestrales. En algunos sistemas de control digital y en esta tesis, la señal que representa el esfuerzo de control es constante de manera fragmentaria, es decir, es igual para  $kT-t < (k+1)T$ , es decir, solamente varia en puntos discretos de tiempo.

#### 4.1.1 Cuantificación.

El hecho de que una señal sea continua significa que puede tomar infinito numero de valores. En cambio, un sistema digital, solo tiene posibilidades limitadas, ya que los posibles estados del sistema, expresados, digamos, mediante una cantidad finita de números binarios, son limitadas. Por lo tanto, al incluir una computadora en el sistema de control forzosamente se hacen necesarios conversores digital-a-analógico y analógico-a-digital. Entonces, las señales que recibe la computadora son aproximaciones de las señales reales. Este proceso se denomina *cuantificación*. En este proceso no solamente se aproxima una señal analógica a una señal digital en un momento dado, sino que también existen lapsos de tiempo completos con información proveniente del sistema continuo, que son representados por un solo valor digital (producto de cuantificar uno o varios de los valores de dicho intervalo). El proceso de cuantificación se ilustra en la figura 4.3.1. En esta figura se ve cómo el tiempo se divide en lapsos disjuntos  $h_i$ . Estos lapsos no necesariamente son de la misma longitud. Sin embargo, se caracterizan por aproximar a todos los valores dentro de ese lapso de tiempo  $x(t)$ , con  $t \in h_i$  por un único valor  $y=Q(x)$ . En este ejemplo,  $x(t)$  representa la señal continua (analógica) y  $y=Q(x)$  la salida digital, con  $Q$  como función cuantificante.

La señal analógica se puede cuantificar en tiempo y en amplitud. El gráfico 4.1.1 ilustra la cuantificación en amplitud. En los gráficos 4.1.2 y 4.1.3 se muestra una señal discreta en el tiempo y una señal discreta en el tiempo cuantificada, respectivamente.

Gráfico 4.1.1.

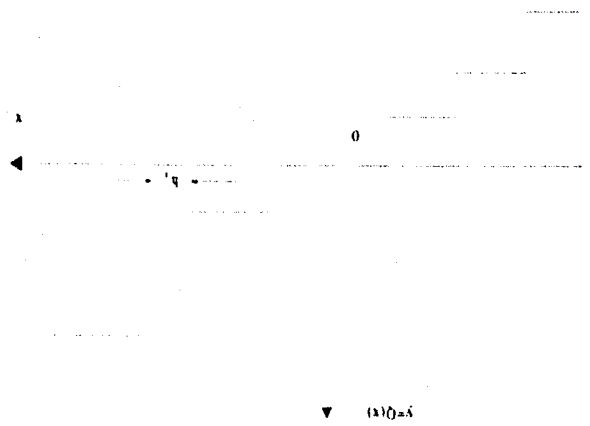


Gráfico 4.1.2



Cabe mencionar que una señal discreta en el tiempo implica que hay lapsos de tiempo en los que no se recibe información. Además, si la señal discreta esta cuantificada solo puede

tomar un número finito de valores (Valores discretos), que resultan ser aproximaciones digitales, como ya se había mencionado anteriormente

#### 4.1.2 Algunas definiciones.

Con el fin de expresar adecuadamente las partes que pueden conformar un sistema de control digital será necesario identificar y describir de manera breve en que consiste cada una.

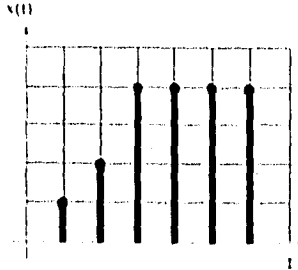


Gráfico 4.1.3.

**Transductor:** Es un dispositivo que convierte una señal de entrada a una señal de salida con otra forma.

**Transductor de muestreo de datos:** En este tipo de transductor las señales de entrada y de salida solamente se producen en instantes discretos de tiempo. Generalmente, estos instantes están espaciados de manera homogénea y se dice que es periódico. Sin embargo, las amplitudes de la señal no están cuantificadas.

**Transductor analógico-a-digital:** En este tipo de transductor la entrada es una señal continua y la salida es una señal cuantificada (datos binarios).

**Transductor digital-a-analógico:** Es el caso inverso: la señal de entrada es una señal cuantificada y la de salida es una señal expresada en términos de voltaje (magnitud física).

Los controles digitales, que pueden ser computadoras, funcionan con base numérica. Si el sistema que va a controlarse es un sistema de tiempo continuo, necesariamente se requiere un transductor digital-a-analógico y uno analógico-a-digital. Además, las señales suelen obtenerse mediante muestreos periódicos.

#### 4.1.3 Ventajas de los controladores digitales.

Las ventajas de los controladores digitales sobre los analógicos son:

4.1.3.1 Los controladores digitales pueden hacer cálculos complejos a alta velocidad, con exactitud constante. El

costo se incrementa poco al aumentar la complejidad de los cálculos, en cambio, en los controladores analógicos se incrementa rápidamente el costo al aumentar la complejidad, si se mantiene constante la exactitud.

4.1.3.2 Los controladores digitales son sumamente versátiles, pues basta cambiar el programa, para cambiar completamente la forma de operación. No se puede decir lo mismo de los controladores analógicos.

#### 4.2 TRANSFORMADA Z.

La transformada Z es una herramienta matemática para el análisis de los sistemas de control de tiempo discreto. La relación que guarda la transformada Z respecto a los sistemas lineales de tiempo discreto, invariantes en el tiempo, es análoga a la que existe entre la transformada de Laplace y los sistemas lineales invariantes en el tiempo de sistemas continuos.

##### 4.2.1 Muestreadores y dispositivos de retención.

Como los sistemas digitales funcionan con señales discretas, es necesario muestrearlas para poderlas procesar. Un muestreador convencional contiene interruptores que se cierran para sólo admitir señales cada T segundos. El tiempo de muestreo deberá ser despreciable comparado con la constante de tiempo más significativa de la planta.

Entonces, el muestreador convierte una señal continua en un tren de pulsos, producidos en los instantes  $0, T, 2T, 3T, \dots$ , donde T es el periodo de muestreo. Puede haber dos señales diferentes que en los instantes de muestreo sean iguales. Desde el "punto de vista" de la computadora, se trata de la misma señal, pues producen la misma señal muestreada.

Como entre dos instantes de muestreo no se recibe información alguna, se utiliza un dispositivo de retención que alisa la señal muestreada, de tal forma que la salida del

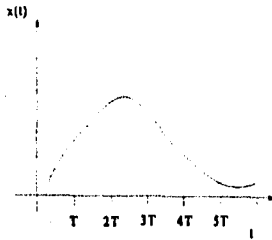


Gráfico 4.2.1.

dispositivo de retención es una aproximación de la señal que se muestrea.

Por ejemplo, si la señal que va a ser muestreada es  $x(t)$ , representada en el gráfico 4.2.1, produciría una señal  $x^*(t)$  a la salida del muestreador (gráfico 4.2.2) y una señal  $x_h(t)$  a la salida del dispositivo de retención (gráfico 4.2.3).

El dispositivo de retención más simple convierte la señal muestreada en una señal constante entre los instantes de muestreo, de tal forma que:

$$x_h(kT+T) = x(kT) \text{ para } 0 \leq t \leq T.$$

Gráfico 4.2.2.

Este dispositivo de retención es un filtro pasa bajo que alisa la señal de entrada y la convierte en una señal constante de manera fragmentada. Es un dispositivo de retención de orden cero.

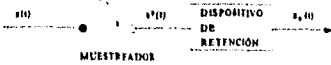


Gráfico 4.2.4.

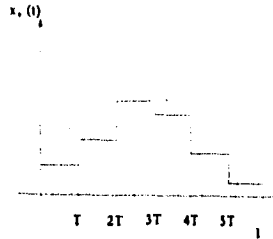


Gráfico 4.2.3.

Su función de transferencia es la siguiente:

$$G_h = \frac{1 - e^{-Ts}}{s} \quad (4-1)$$

Básicamente lo que hace un dispositivo de retención es integrar la señal  $x^*(t)$  entre dos instantes de muestreo consecutivos. Pero como la integral de una función impulso es una constante, la

entrada a un dispositivo de retención de orden cero es un tren de funciones impulso.

La relación entre la entrada y la salida de un muestreador se puede describir de la siguiente manera:

$$x^*(t) = \delta_T(t) x(t) \quad (4-2)$$

donde  $\delta_T$  es un tren de impulsos que se puede definir de la siguiente manera:

$$\delta_T = \sum_{k=0}^{\infty} \delta(t-kT) \quad (4-3)$$

y  $\delta(t-kT)$  es una función impulso unitario producida en  $t=kT$ .

Entonces la función muestreada se podría definir como:

$$x^*(t) = \sum_{k=0}^{\infty} x(t)\delta(t-kT) \quad (4-4)$$

Pero como la función  $\delta(t-kT)$  vale cero en todos los instantes en los que  $t \neq 0, T, 2T, \dots, kT$ , solamente interesan los valores de  $x(t)$  en los que  $t$  coincide con  $0, T, 2T, 3T, \dots, kT$ . Por lo tanto, la función anterior es equivalente a:

$$x^*(t) = \sum_{k=0}^{\infty} x(kT)\delta(t-kT) \quad (4-5)$$

Si se toma la transformada de Laplace de esta señal  $x^*(t)$  modulada por el tren de impulsos  $\delta_T$ , resulta que:

$$\mathcal{L}[x^*(t)] = \mathcal{L}\left[\sum_{k=0}^{\infty} x(kT)\delta(t-kT)\right] = \sum_{k=0}^{\infty} x(kT)\mathcal{L}[\delta(t-kT)] \quad (4-6)$$

$$X^*(s) = \mathcal{L}[x^*(t)] = \sum_{k=0}^{\infty} x(kT)e^{-kTs} \quad (4-7)$$

Si se define a  $z$  como:

$$z = e^{Ts}$$

Entonces la transformada  $z$  se define como la transformada de Laplace del tren de pulsos de la señal muestreada y se expresa en términos de  $z$  de la siguiente manera:

$$X(z) = X^*(s) = X^* \left( \frac{1}{T} \ln z \right) = \sum_{k=0}^{\infty} x(kT) z^{-k} \quad (4-9)$$

La operación de la transformada  $z$  sobre una función de pulsos en el dominio del tiempo  $x^*(t)$  se representa como:

$$Z[x^*(t)] = X(z)$$

$X(z)$  solamente es función de  $x(t)$ , pero con  $t=0, T, 2T, \dots, kT$ . Por tanto, la transformada inversa de  $z$  proporciona información de  $x(t)$  únicamente en los instantes  $t=0, T, 2T, \dots, kT$ .

Como se dijo anteriormente, la transformada  $z$  tiene usos análogos a la transformada de Laplace, por ejemplo, el análisis de estabilidad de los sistemas de tiempo discreto.

#### 4.3 UN EJEMPLO DE ANÁLISIS DE ESTABILIDAD.

A continuación se va a ilustrar el uso de la transformada  $z$  mediante un ejemplo concreto: se analizará el sistema de tiempo discreto siguiente:

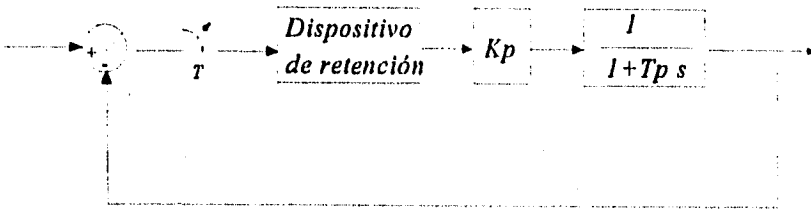


Gráfico 4.3.1.



Se trata de un sistema de control proporcional antecedido por un sistema de muestreo y un dispositivo de retención. La planta a controlar es de primer orden. El sistema tiene retroalimentación negativa. El dispositivo de retención se considera de orden cero.  $K$  es la constante del control proporcional y  $T_p$  es la constante de tiempo de la planta que se pretende controlar.

Para analizar este sistema necesitamos obtener primeramente la función de transferencia del sistema abierto, que es:

$$G(s) = \frac{1 - e^{-sT}}{s} K_p \frac{1}{1 + T_p s} \quad (4-10)$$

Para realizar el análisis de estabilidad de este sistema es necesario obtener la transformada  $z$  del sistema de lazo abierto. Se sabe que  $Z(1 - e^{-sT}) = 1 - z^{-1}$ , ya que  $z = e^{sT}$ . Entonces,

$$Z\left\{\frac{1 - e^{-sT}}{s}\right\} = T(1 - z^{-1}) / (z - 1)$$

Se puede utilizar el método de mapeo de polos, que dice que la transformada  $z$  deberá tener los mismos polos y los mismos ceros que esta función. Sin embargo, un cero en infinito - como en este caso - se transformará en un cero en  $-1$ . Entonces la transformada  $z$  tendrá dos polos: uno en cero y otro en  $-(1/T_p)$ . Además, deberá tener un cero en  $-1$ :

$$Z\left\{\frac{K_p}{1 + T_p s} \frac{1}{s}\right\} (1 - z^{-1}) = K K_p \frac{z + 1}{z - e^{-T/T_p}} \frac{1 - z^{-1}}{z - 1} \quad (4-11)$$

Es fácil verificar que  $z - e^{-T/T_p}$  es cero cuando  $s = -1/T_p$ , que implica que  $z = e^{-T/T_p}$ . Lo mismo será para  $s = 0$  ( $z = 1$ ) - ya que  $z = e^{sT}$ .

Solamente faltaría determinar el valor de  $K$ , que se obtiene haciendo  $s = j\omega = 0$  en:

$$1 = \frac{z + 1}{z - e^{-T/T_p}} K = \frac{e^{sT} + 1}{z - e^{-T/T_p}} K \quad (4-12)$$

Luego

$$K = \frac{1 - e^{-T/T_p}}{2} \quad (4-13)$$

Por tanto, a función  $G(z)$  sería (simplificando):

$$G(z) = \frac{K_p(1-e^{-T/T_r})(z+1)}{(2z)(z-e^{-T/T_r})} \quad (4-14)$$

Ahora, suponiendo valores concretos para  $K_p$  y  $T_r$ , por ejemplo, 1 y 10, respectivamente, tendríamos que analizar los polos del sistema:

$$\frac{G(z)}{1+G(z)} \quad (4-15)$$

y y entonces la ecuación característica del sistema es:

$$1 + G(z) = 0 \quad (4-16)$$

Es decir,

$$2z(z-e^{-T/10}) + (1-e^{-T/10})(z+1) = 0 \quad (4-17)$$

Haciendo  $A=e^{-T/10}$  y desarrollando esta expresión se obtiene:

$$2z^2 + z(1-3A) + (1-A) = 0 \quad (4-18)$$

La condición de estabilidad es que las raíces de la ecuación característica estén dentro del círculo unitario  $|z|=1$ . Si están sobre la circunferencia, las raíces no deberán de ser múltiples.

Se resolverá dicha ecuación para diferentes valores de  $T$  para así poder observar cómo al cambiar el tiempo de muestreo, se afecta la estabilidad de un sistema de tiempo discreto.

A continuación se presentan varias soluciones utilizando diferentes valores de T:

T	$z_1$	$ z_1 $	$z_2$	$ z_2 $
0.5	1.2700	1.2700	-1.3455	1.3455
1.0	1.0355	1.0355	-1.1783	1.1783
1.5	0.7794	0.7794	-0.9883	0.9883
2.0	0.4781	0.4781	-0.7500	0.7500
2.5	- 0.0699	0.0699	-0.2619	0.2619
3.0	- 0.1944+0.5707j	0.6029	-0.1944-0.5707j	0.6029
3.5	- 0.2215+0.7942j	0.8245	-0.2215-0.7942j	0.8245
4.0	-0.2473+0.9532j	0.9848	-0.2473-0.9532j	0.9848
4.5	-0.2718+1.0779j	1.1117	-0.2718-1.0779j	1.1117
5.0	-0.2951+1.1802j	1.2165	-0.2951-1.1802j	1.2165

Con esto se puede ver claramente cómo con valores de T menores a 1 y mayores a 4.5 el sistema se vuelve inestable. De esta manera se concluye que la estabilidad de un sistema de tiempo discreto lineal, invariante con el tiempo puede ser alterada al cambiar el tiempo de muestreo.

# **CAPÍTULO 5**

## 5. SIMULACIÓN UTILIZANDO UN MODELO DEL VEHÍCULO PARA UNA SOLA LLANTA.

### 5.1. DESCRIPCIÓN DE LA INTERACCIÓN ENTRE EL SUELO Y LAS LLANTAS DE UN VEHÍCULO.

Los neumáticos de un vehículo terrestre tienen las siguientes funciones:

- Soportar el peso del vehículo
- Suavizar la marcha del vehículo sobre las irregularidades de la superficie.
- Proporcionar tracción suficiente durante la aceleración y el frenado
- Proveer estabilidad direccional adecuada y maniobrabilidad. En esta tesis se limitará al tratamiento de los vehículos de superficies duras (excluyendo los vehículos para todo terreno, en los que el suelo se considera deformable).

#### 5.1.1. Llantas con cuerdas radiales y con cuerdas diagonales.

Una llanta es una estructura en forma de toroide y rellena de aire a presión. La forma de construcción es diversa. Básicamente se compone de varios cinturones superpuestos y una cubierta exterior, que es la que ostenta el dibujo, con un patrón determinado.

Los cinturones están hechos de un material compuesto: cuerdas, de un alto módulo de elasticidad y hule (ya sea caucho o hule sintético) de un bajo módulo de elasticidad. Las cuerdas pueden estar orientadas de diferente forma. De ahí que haya llantas con cuerdas orientadas en forma radial y otras con cuerdas orientadas en forma diagonal.

La orientación de la cuerda se describe mediante el ángulo que forma la cuerda con el eje perimetral de la llanta, llamado ángulo de corona.

Las llantas radiales tienen un núcleo de cuerdas transversales al eje perimetral de la llanta, pero también tienen encima cinturones con un bajo ángulo de corona, que hacen que la llanta tenga suficiente estabilidad.

Cada uno de estos tipos de llantas tiene diferente comportamiento. Las llantas con cuerdas radiales tienen mejor desempeño en línea recta que las llantas con cuerdas en diagonal; sin embargo, éstas tienen mejores características de dirección.

En general, las llantas radiales tienen mayor eficiencia, esto es, disipan menos energía, ya que el movimiento permisible entre las cuerdas de una llanta radial, es menor.

### 5.1.2. Momentos y fuerzas que actúan sobre una llanta.

La Sociedad de Ingenieros Automotrices ( SAE ) recomienda la siguiente manera de escoger el marco de referencia para describir las fuerzas y momentos que se ejercen sobre una llanta:

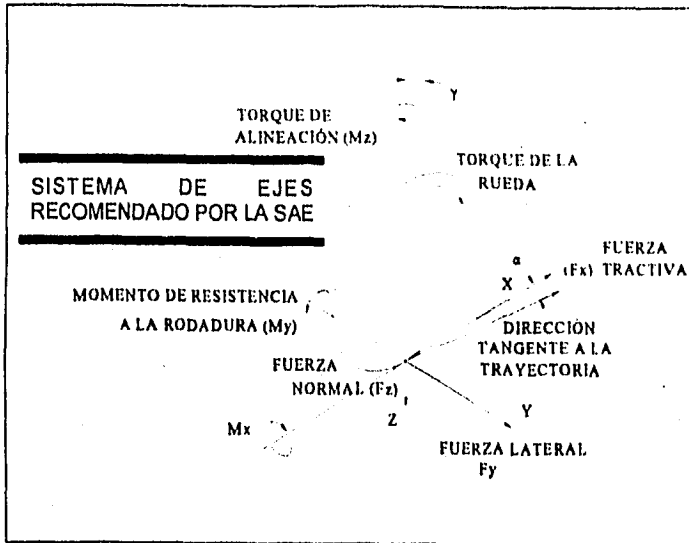


Diagrama 5.1.1. SISTEMA DE EJES EN LA LLANTA

En este sistema de coordenadas se ha escogido el origen en el punto de contacto entre la rueda y el suelo. El eje  $z$  es perpendicular a la superficie del suelo en contacto con la llanta, con la dirección positiva hacia abajo. La dirección  $x$  es la intersección entre el suelo y el plano que contiene a la llanta, con el sentido positivo hacia la dirección de avance de la llanta. El eje  $y$  ha sido escogido para formar un sistema derecho de coordenadas. Alrededor de cada eje se produce un momento:

Alrededor del eje  $z$  se produce el momento de autoalineación, alrededor del eje  $y$  se produce el momento de resistencia a la rodadura y alrededor del eje  $x$  se produce el momento de volcadura. Asimismo, en la dirección de cada eje se produce una fuerza: la fuerza tractiva, la fuerza lateral y la fuerza normal sobre los ejes  $x$ ,  $y$  y  $z$ , respectivamente.

Utilizando este sistema de coordenadas se pueden definir

algunos parámetros de desempeño que resultan útiles, por ejemplo:

- El corrimiento longitudinal del centro de presión es la relación entre el momento de resistencia a la rodadura y la carga normal.
- El corrimiento lateral del centro de presión es la relación entre el momento de volcadura y la carga normal.
- La resultante de los esfuerzos cortantes longitudinales en el área de contacto resulta ser la fuerza tractiva.
- El momento de tracción alrededor del eje de rotación de la rueda es el que provoca una aceleración del vehículo hacia adelante y el momento de frenado provoca su desaceleración.

Ángulos importantes: existen dos ángulos importantes que se pueden referir a este sistema de coordenadas:

- El ángulo de deslizamiento  $\alpha$ , que se forma entre la dirección de la trayectoria de la rueda y la intersección del plano del piso con el plano de la llanta.
- El ángulo *cámbler*  $\gamma$ , que se forma entre el plano *xz* y el plano de la rueda. La fuerza lateral, depende de estos dos ángulos.

### 5.1.3. Resistencia a la rodadura de las llantas.

La resistencia a la rodadura se produce principalmente por la histéresis en los materiales de las llantas, que es el resultado de la deflexión del esqueleto de la llanta al rodar.

Existen otros factores que contribuyen a la resistencia a la rodadura, pero de manera secundaria:

La fricción entre la llanta y el camino, causada por el deslizamiento, la resistencia debida al aire que circula en el interior de la llanta y el efecto de ventilador de la llanta al girar en el aire exterior. Se tienen disponibles datos experimentales que nos dan una idea de la importancia de cada uno de estos factores de pérdida, en un rango de 120 a 152 km / h :

Entre un 90 a 95% se debe a la histéresis interna, de 2 a 10% se debe a la fricción entre la llanta y el suelo, y entre un 1.5 a 3.5%, a la resistencia del aire.

Al rodar la llanta, el esqueleto de la llanta se deflexiona en el área de contacto. Sin embargo, la presión a lo largo del área de contacto no es uniforme. Esta presión es mayor en la zona delantera del área de contacto. La zona delantera se considera la zona más cercana a la dirección del eje *x* en el sistema de coordenadas expuesto anteriormente. Además, el centro de presión se desplaza precisamente hacia adelante, lo cual da lugar a un momento alrededor del eje *y*, que es el momento de resistencia a la rodadura.

Si analizamos el caso de una llanta sin tracción, observamos que ningún par actúa sobre ella, y sin embargo, está en equilibrio

dinámico. Para conseguir esto, necesariamente se necesita una fuerza que contrarreste al momento de resistencia a la rodadura. Esta fuerza es una fuerza horizontal, llamada simplemente resistencia a la rodadura. El coeficiente de resistencia a la rodadura es la relación entre esta fuerza de resistencia a la rodadura y la carga normal de la llanta. También se puede considerar al coeficiente de resistencia a la rodadura como coeficiente de fricción entre la llanta y el suelo. Existen varios factores que afectan la resistencia a la rodadura:

**La construcción de la llanta.** El aumento de pérdidas por histéresis aumenta la resistencia a la rodadura. Por ejemplo, al disminuir el ancho de la pisada o al aumentar el número de capas del esqueleto, aumenta dicha resistencia. Las llantas de construcción radial tienen una menor resistencia a la rodadura que las de cuerdas en diagonal.

**La velocidad de la llanta.** Al aumentar la velocidad de la llanta aumenta el coeficiente de resistencia a la rodadura (Ver Diagrama 5.1.2).

**Material de las llantas.** Las llantas hechas de compuestos de hule sintético tienen una resistencia a la rodadura ligeramente mayor que las de hule natural.

**Condiciones del suelo.** Influyen de manera significativa. No solamente existen diferencias entre distintos tipos de suelo - de mayor a menor resistencia a la rodadura, están los suelos arenosos, los suelos de dureza media y el concreto -, sino también la manera como se relaciona la presión de inflado de las llantas con la resistencia a la rodadura depende también del tipo de suelo: la presión de inflado de las llantas influye más en la resistencia a la rodadura en suelos de dureza media o baja (Ver Diagrama 5.1.3).

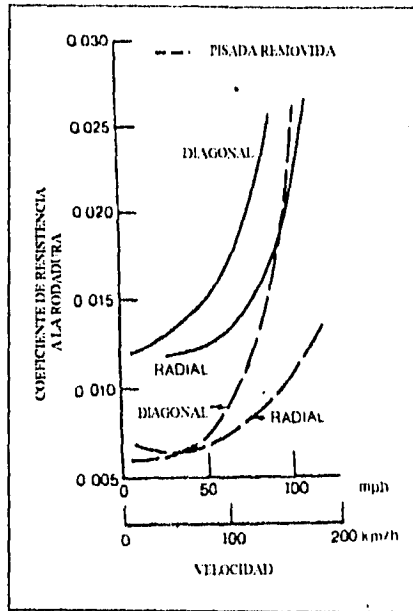


Diagrama 5.1.2.



**Temperatura interior de la llanta.** La resistencia a la rodadura disminuye conforme aumenta la temperatura interna, para una misma velocidad. Para una misma temperatura, aumenta la resistencia a la rodadura con la velocidad.

**Diámetro de la llanta.** Para suelos suaves la resistencia a la rodadura disminuye conforme aumenta el diámetro de la llanta, pero para superficies duras, la influencia del diámetro de la llanta es menor (ver diagrama 5.1.4)

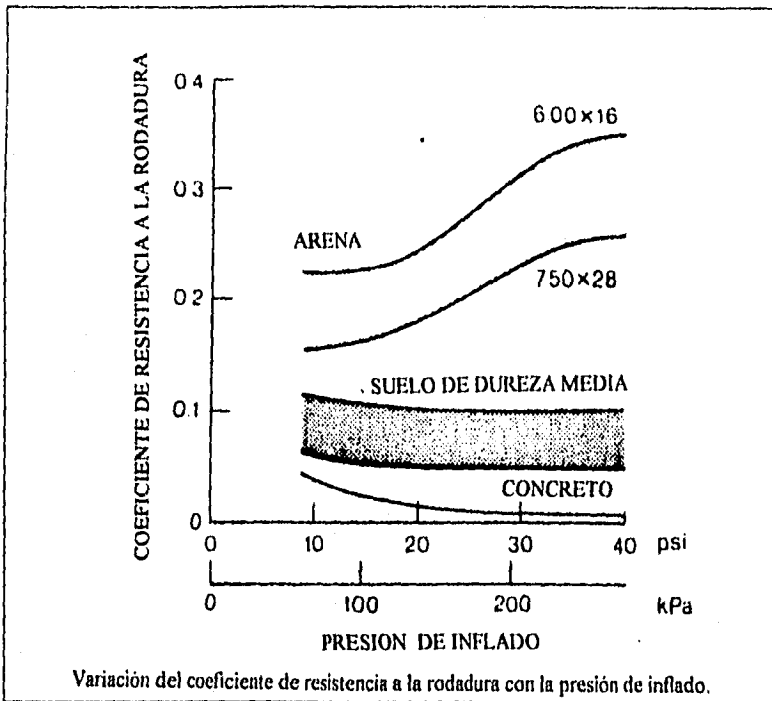
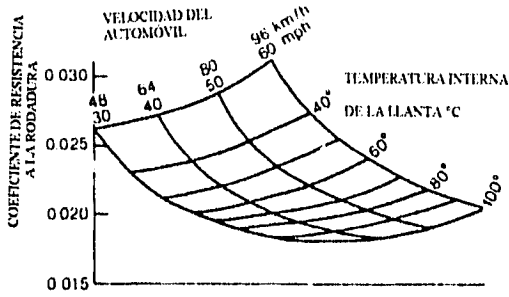
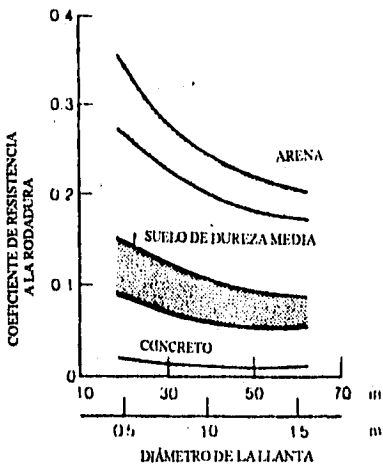


Diagrama 5.1.3.



Efecto de la temperatura interna en el coeficiente de la resistencia a la rodadura de la llanta.



Efecto del diámetro en el coeficiente de resistencia a la rodadura en varias superficies.

Diagrama 5.1.4.

Fuerza de tracción y frenado. La resistencia a la rodadura aumenta conforme aumenta la fuerza de tracción y de frenado (Diagrama 5.1.5).

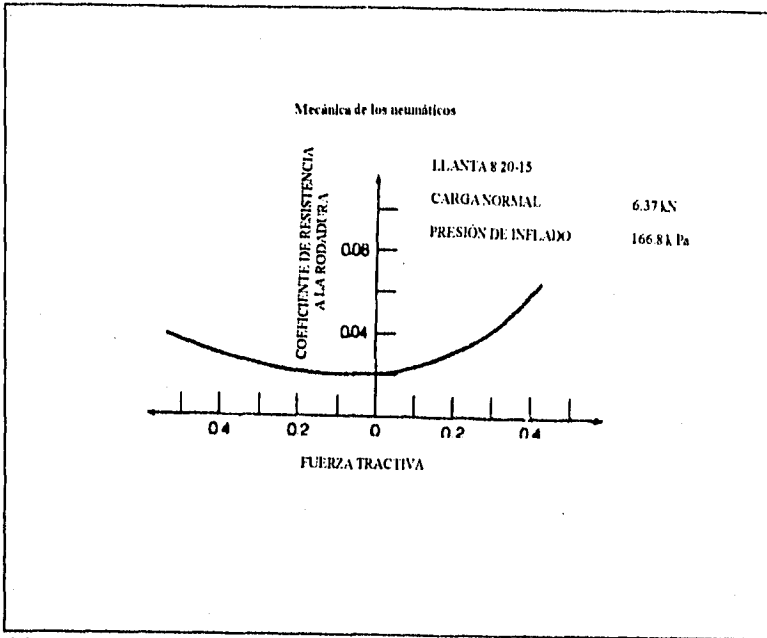


Diagrama 5.1.5.

Las relaciones que existen entre los distintos factores que influyen en la resistencia a la rodadura -tanto parámetros de diseño como parámetros de operación- la resistencia a la rodadura misma, son tan complicadas que solamente existen relaciones empíricas que permitan relacionar algunas de ellas, bajo ciertas condiciones<sup>1</sup>; sin embargo no existe un método analítico para predecir la resistencia a la rodadura tomando en cuenta dichos factores.

<sup>1</sup>Por ejemplo, se puede mencionar la fórmula  $f_r = 0.01 (1 + V / 100)$ , que es una aproximación lineal de la resistencia a la rodadura ( $f_r$ ) en función de la velocidad ( $V$ ), para suelos duros y velocidades de hasta 128 km/h. O bien  $f_r = f_0 + f_1(V/100)^{2.5}$  (Ver diagrama 5.1.6 para encontrar los factores  $f_0$  y  $f_1$ ).

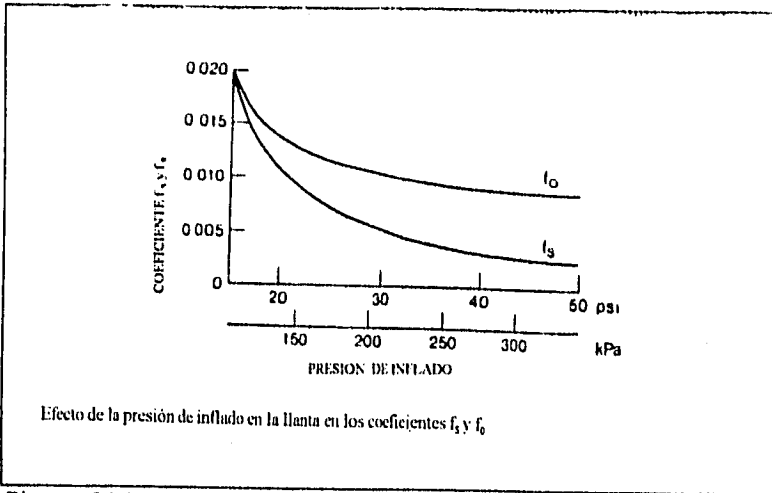


Diagrama 5.1.6.

#### 5.1.4. Relación entre fuerza de tracción (o de frenado) y deslizamiento longitudinal de las llantas.

Al aplicar un momento a la rueda que va a transmitir la potencia, se desarrolla entre la llanta y la superficie de contacto una fuerza de tracción. Al mismo tiempo, la pisada de la llanta frente y dentro del área de contacto se somete a compresión. Asimismo, en las caras laterales de la llanta, se desarrollan los correspondientes esfuerzos cortantes.

Ya que los elementos de la pisada de la llanta (suela) se comprimen antes de entrar al área de contacto, la distancia que la llanta viaja cuando está transmitiendo la tracción, es menor que la de otra llanta sin tracción. Este fenómeno se conoce como deslizamiento por deformación. El deslizamiento de una llanta con tracción de un vehículo, cuando se aplica torque a dicha llanta se define de la siguiente manera:

$$i = \left(1 - \frac{V}{r\omega}\right) \times 100\% = \left(1 - \frac{r_s}{r}\right) \times 100\%$$

donde:

- V: Es la velocidad de translación del centro de la llanta.  
 $\omega$ : Es la velocidad angular de la llanta.  
 r: es el radio de giro de la llanta girando libremente (sin torque).  
 $r_e$ : Es el radio efectivo de giro de la llanta con torque, es decir, es la relación entre la velocidad de translación del centro de la llanta y la velocidad angular de dicha llanta.

Si se aplica un torque de tracción a una llanta, ésta giraría con menor translación que otra que no se hubiera deformado, es decir que  $r\omega > V$ .

Como la fuerza de tracción que se desarrolla sobre la llanta es proporcional al torque aplicado sobre la rueda, bajo condiciones de estado-estable, el deslizamiento es función de la fuerza de tracción.

En términos generales, al principio, esta relación es lineal, ya que se debe básicamente a deformaciones elásticas.

Si aumenta más el torque y la fuerza de tracción se ocasiona que parte de la pisada de la llanta en contacto con el suelo se deslice. Entonces, bajo estas circunstancias, la relación entre la fuerza tractiva y el deslizamiento es no lineal.

Datos experimentales muestran que la condición de máxima fuerza de tracción se produce en algún punto entre el 15 y 20% de deslizamiento, en suelos duros. Cualquier incremento del deslizamiento por encima del correspondiente al máximo posible, desemboca en una condición inestable, que se caracteriza por una caída del coeficiente de fuerza de tracción (la relación entre la fuerza de tracción y la carga normal) desde  $\mu_p$  -valor pico- hasta  $\mu_s$  -valor de deslizamiento puro.

## 5.2. COEFICIENTE DE FRICCIÓN EN FUNCIÓN DEL DESLIZAMIENTO.

### FUNCIONES SELECCIONADAS PARA DIFERENTES CONDICIONES.

Para conocer el desempeño de un vehículo es indispensable conocer también la interacción que se produce entre el suelo y la llanta del vehículo.

Para esto, se han tomado en cuenta las funciones que describen el coeficiente de fricción entre el suelo y la llanta en función del deslizamiento.

El deslizamiento no es más que la diferencia entre la velocidad angular de la llanta y la velocidad angular equivalente del vehículo, comparadas con la velocidad mayor, según se expresa en la siguiente ecuación:

$$\lambda = \frac{\omega_H - \omega_V}{\omega_H} \quad \text{si } \omega_H > \omega_V$$

$$\lambda = \frac{\omega_H - \omega_V}{\omega_V} \quad \text{si } \omega_H < \omega_V \quad (5.1)$$

donde,

$$\omega_V = \frac{V}{R_H} \quad (5.2)$$

$V$  = velocidad lineal del vehículo,

$R_H$  = Radio de la llanta.

Como se dijo antes, la fricción entre la llanta y el suelo, en diferentes condiciones, será función del deslizamiento, como lo muestra la gráfica 5.1.

Se utilizaron las siguientes funciones para describir esta interacción:

$$\begin{aligned} \text{para } \lambda > x & \quad \mu = m \lambda + A \operatorname{sen} ( p x ) - mx \\ \text{para } -x \leq \lambda \leq x & \quad \mu = A \operatorname{sen} ( p \lambda ) \\ \text{para } \lambda < -x & \quad \mu = m \lambda - A \operatorname{sen} ( p x ) + mx \end{aligned}$$

con los parámetros  $x$ ,  $px$ ,  $p$  y  $m$ , para cada condición del suelo:

#### 5.2.1 Para suelo con seco:

$$\begin{aligned} x &= 0.41121 \\ A &= 0.9 \\ px &= 1.61481 \\ p &= 5\pi/4 \\ m &= \tan(-10\pi/180) \end{aligned}$$

---

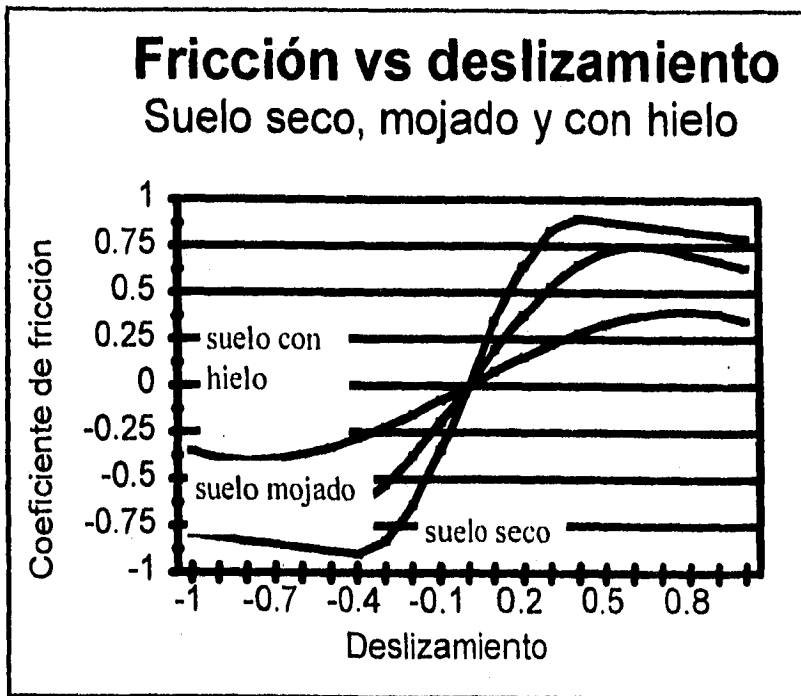
<sup>2</sup>Tan H. y Tomizuka M. "Discrete-Time Controller Design for Robust Vehicle Traction", Control Systems Magazine, Volumen 10, No. 3, pp. 107 - 113, abril, 1990.

5.2.2) Para suelo mojado:

$$\begin{aligned} x &= 0.69914 \\ A &= 0.75 \\ px &= 1.69943 \\ p &= 10 \cdot n / 12 \\ m &= \tan(-20 \cdot n / 180) \end{aligned}$$

5.2.3) Para suelo con hielo:

$$\begin{aligned} x &= 0.88754 \\ A &= 0.4 \\ px &= 1.74268 \\ p &= 5n / 8 \\ m &= \tan(-20 \cdot n / 180) \end{aligned}$$



Gráfica de fricción vs deslizamiento

Gráfica 5.1

### 5.3 MODELO SIMPLIFICADO UTILIZADO EN LA SIMULACION.

#### 5.3.1. Objetivos de la simulación del sistema ABS.

En general, la simulación es una herramienta que permite resolver problemas que, si se simulan, se puede obtener alguna ventaja en cuanto a tiempo de cálculo, realismo de la representación o costo.

En este caso, se pretende mejorar el desempeño del frenado de los vehículos de 2 ejes, mediante el uso de un sistema de frenos ABS.

##### 5.3.1.1 Peligros de bloquear las ruedas.

El sistema ABS pretende precisamente evitar el bloqueo de las ruedas. Lo que podría pasar si estas resultan inmovilizadas es lo siguiente:

- Al bloquearse las ruedas traseras se pierde estabilidad direccional, y esto provoca que el vehículo pueda llegar a girar sobre su eje vertical. Esto se debe a la pérdida de tracción lateral, cuando el deslizamiento es 100%. En este caso, es muy poco lo que el conductor puede hacer para evitar este movimiento.
- Al bloquearse las llantas delanteras se pierde control direccional, pero no necesariamente estabilidad, pues hay un momento de autoalineación que tiende a mantener el automóvil en dirección recta, pero impide virar.

Aunque el primer caso es más grave, ninguno de los dos es deseable.

##### 5.3.1.2 Descripción de los frenos ABS:

Para evitar el bloqueo de las llantas (deslizamiento de 100%) existe el sistema de frenos ABS, que libera la presión del freno antes que comience el bloqueo y reanuda la presión del freno cuando el bloqueo ya no es inminente.

El sistema de frenos ABS consta de los siguientes elementos: (ver diagrama 5.3.0)

1. Sensor
2. Unidad de Control
3. Modulador de presión
4. Cilindro de la rueda

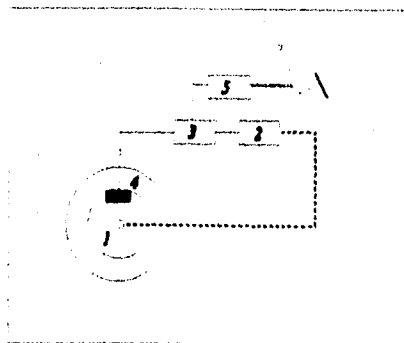


Diagrama 5.3.0.



## 5. Cilindro maestro

### 5.3.1.3 Supuestos acerca del diseño del sistema de frenos.

El sistema de presión en los frenos, que no es objeto de estudio de esta tesis, debe considerar la siguiente condición de diseño: la presión en los frenos delanteros y traseros no debe ser igual, a menos que exista la misma distancia entre las llantas delanteras y traseras y el centro de gravedad del automóvil. En caso contrario, que es el más común, el área de presión en el émbolo del cilindro, debe obedecer al siguiente lineamiento:

donde:

$$\frac{K_{fd}}{K_{ft}} = \frac{W_d}{W_f}$$

$K_{fd}$  = Porcentaje de la fuerza de frenado aplicada en el freno delantero.  
 $K_{ft}$  = Porcentaje de la fuerza de frenado aplicado en el freno trasero.  
 $W_d$  = Peso soportado por las llantas delanteras.  
 $W_f$  = Peso soportado por las llantas traseras.

De otra manera, las llantas traseras o las llantas delanteras, no estarán frenando con la máxima eficiencia, o bien, resultarán bloqueadas mientras las llantas delanteras o las traseras, respectivamente, evitan el bloqueo.

### 5.3.1.4 Objetivo de la simulación.

El objetivo de esta simulación es evaluar la manera como un algoritmo puede controlar un sistema físico (en este caso el automóvil con un freno ABS), de tal forma que se hará más énfasis en los parámetros del control que afectan dicho desempeño, tanto internos como externos, y no tanto en la descripción completa del automóvil, que requeriría de un modelo con 6 grados de libertad. Es por ello que el modelo estará limitado a una sola llanta, que nos permite describir el desempeño del vehículo, en términos de las variables -de mayor importancia- que son físicamente medibles y que controla el sistema ABS. Éstas son:

- Velocidad de translación del automóvil
- Velocidad angular de la llanta
- Deslizamiento

Además, este modelo tendrá parámetros concentrados, que serán calculados posteriormente, en la sección 5.4. Para efecto de describir las relaciones que existen entre estas variables, se plantearán primero las ecuaciones diferenciales de la dinámica de

la rueda y del vehículo, después, se relacionarán las 3 variables, utilizando la relación expuesta anteriormente en la sección 5.2.

### 5.3.2 Dinámica de la rueda.

Como este modelo es de parámetros concentrados, la rueda se considera como la masa que gira a la velocidad angular de la llanta, pero que incluye la inercia de todas las partes mecánicas que conectan al neumático con el motor, incluyendo rodamientos, elementos de transmisión y elementos de montaje para el neumático, de tal forma que se pueda considerar que sobre esta rueda actúan los siguientes momentos, con sus respectivos signos, de acuerdo a la convención que considera como positivo el giro de la rueda que hace que el vehículo avance hacia adelante:

$T_e$  = Par del motor ( + )

$T_f$  = Par de frenado ( - )

$F_v(\omega)$  = Par de amortiguamiento viscoso ( - )

$M_r$  = Momento de resistencia a la rodadura de una rueda ( - )

La ecuación diferencial utilizada para describir un sistema mecánico como este que consta de una inercia rotacional y diversos pares aplicados a ella es:

$$T_e(t) - T_f(t) - F_v(\omega) - M_r(t) = J_{eq} \frac{d\omega}{dt} \quad ( 5 - 1 )$$

donde:

$J_{eq}$  = Inercia rotacional equivalente de la rueda.

$\omega$  = Velocidad angular de la rueda.

Esta es una ecuación diferencial lineal, ya que:

$$F_v = k_v \omega$$

Esto, traducido al lenguaje de diagramas de bloques, se expresa de la siguiente manera:

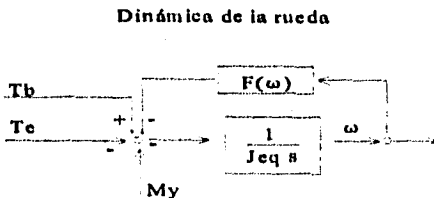


Diagrama 5.3.1.

### 5.3.3 Dinámica del vehículo.

Por otra parte, para describir el movimiento del vehículo completo, únicamente se consideran 2 fuerzas:

- $f$  = La fuerza de tracción de las llantas sobre el vehículo.
- $F(V)$  = La resistencia del viento, que depende de la velocidad lineal del vehículo  $V$ .

Y la ecuación diferencial correspondiente es:

$$f + F_1(V) = M \frac{dV}{dt} \quad (5-2)$$

donde  $M$  es la masa total del vehículo, que se traslada linealmente (en una trayectoria recta sin cambio de dirección).

A diferencia de la ecuación (5-1), la ecuación (5-2) es una ecuación diferencial no lineal, que no podría ser resuelta por el método de la transformada de Laplace, puesto que:

$$F_1(V) = k_2 V^2$$

Lo que, utilizando diagramas de bloques sería:

Dinámica de la masa principal del vehículo

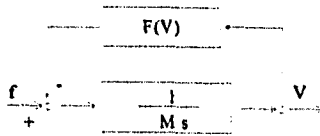


Diagrama 5.3.2.

### 5.3.4 Relación entre la rueda y el vehículo.

Para establecer este vínculo se utiliza la función  $\mu(\omega, V)$ , descrita anteriormente. Además, se calcula  $M_y$  y  $f$  de la siguiente forma:

$$f = M N g \mu(\omega, V) / n$$

$$M_y = M g \mu(\omega, V) R / n$$

Donde: N=Numero de llantas con tracción.  
 g=aceleración de la gravedad.  
 n=Numero total de llantas (con tracción e no).  
 R=Radio de la llanta.

De esta manera, la rueda y el vehiculo se relacionan de la siguiente manera:

DINÁMICA DEL SISTEMA FÍSICO

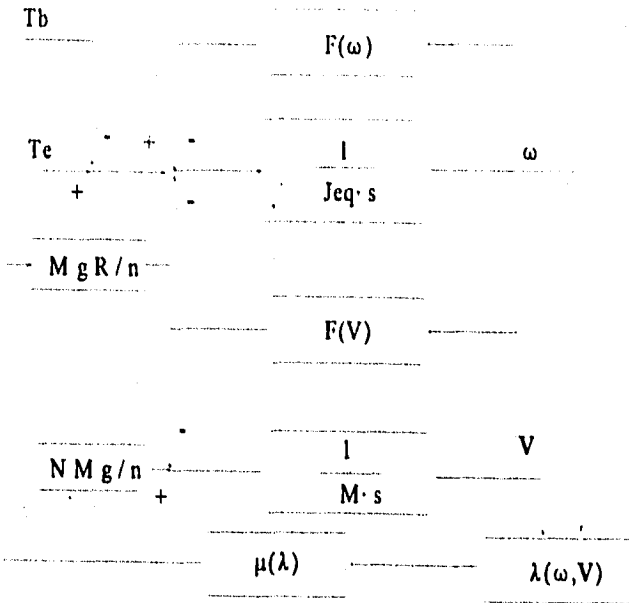


Diagrama 5.3.3.

5.3.5 Reogramas.

5.3.5.1 Qué son los reogramas.

Como ya se dijo anteriormente una de las formas de representar gráficamente los sistemas es mediante el uso de diagramas de

bloques, en donde cada bloque representa un elemento que forma parte del sistema y en cada flecha se marca la señal que se transfiere entre un bloque y otro.

Por ejemplo, en diagrama de bloques que representa a un sistema compuesto por una masa que se acelera linealmente debido a una fuerza, con la velocidad como variable de salida, es el siguiente:



Diagrama 5.3.4.

En los reogramas únicamente se utilizan nodos y ramas con dirección y sentido. Cada nodo representa una variable del sistema. Las flechas representan la relación entre dichas variables y sobre cada flecha se escribe la ganancia y la función de transferencia que hay entre dos variables. Las señales que fluyen entre los nodos, sólo pueden hacerlo en una sola dirección. El mismo sistema que está representado en el diagrama 5.3.4, se representa de la siguiente forma: ( Diagrama 5.3.5 )

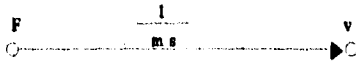


Diagrama 5.3.5.

La ventaja del reograma sobre el diagrama de bloques es que existe una fórmula, debida a S. J. Mason para calcular la relación entre la entrada y la salida. Esto resulta particularmente útil en sistemas de control complejos, pues evita tener que reducir el reograma para conocer esta relación.

#### Definiciones necesarias:

**Nodo:** es un punto que representa una variable o una señal. En el ejemplo anterior, se tienen 2 nodos: F y v.

**Transmitancia:** es una ganancia entre dos nodos.

**Rama:** Es un segmento de línea con dirección y sentido que une dos nodos. La ganancia de una rama es una transmitancia o una función algebraica.

**Nodo de entrada o fuente:** Es un nodo del que solamente salen ramas.

**Nodo de salida o sumidero:** Es un nodo al que solamente entran ramas.

**Nodo mixto:** Es aquel que tiene tanto ramas que llegan, como ramas que salen.

**Camino o trayecto:** Es un recorrido de ramas conectadas en el sentido de las flechas de las ramas. Si no se cruza ningún nodo más de una sola vez, la trayectoria es abierta. Si el trayecto finaliza en el mismo nodo del cuál partió, y no cruza ningún otro nodo más de una vez, es un trayecto cerrado. Si un camino o trayecto cruza algún nodo más de una vez, pero finaliza en un nodo diferente a aquél del cuál partió, no es ni abierto ni cerrado.

**Lazo:** Es un trayecto cerrado.

**Ganancia de lazo:** Es el producto de las transmitancias de las ramas de dicho lazo.

**Lazos disjuntos:** son lazos que no poseen ningún nodo en común.

**Trayecto directo:** es el trayecto de un nodo de entrada a un nodo de salida que no cruza ningún nodo más de una vez.

**Ganancia de trayecto directo:** la ganancia de trayecto directo es el producto de las transmitancias de las ramas de un trayecto directo.

La fórmula es la siguiente:

$$P = \frac{1}{\Delta} \sum_k P_k \Delta_k$$

donde,

$P_k$  = ganancia del trayecto o transmitancia del  $k$ -ésimo trayecto directo.

$\Delta$  = **Determinante del gráfico.**

=  $1 -$  (suma de todas las ganancias de lazo distintas) + (suma de los productos de las ganancias de todas las combinaciones posibles de dos lazos disjuntos) - (suma de los productos de ganancias de todas las combinaciones posibles de lazos disjuntos) + ... =  $1 - \sum L_a + \sum L_b L_c - \sum L_d L_e L_f$

donde

$\sum_a L_a$  = suma de todas las ganancias de lazo distintas.

$\sum_{b,c} L_b L_c$  = suma de los productos de ganancias de todas las combinaciones posibles de dos lazos no adjuntos.

$\sum_{d,s,i} I_d I_s I_i$  = suma de los productos de las ganancias de todas las combinaciones posibles de 3 lazos no adjuntos.

$\Delta_k$  = coeficiente del determinante del k-ésimo trayecto directo del gráfico, con los lazos adjuntos al k-ésimo trayecto directo quitados. Se obtiene quitando del determinante del gráfico los lazos adjuntos al k-ésimo término.

### 5.3.5.2 Reograma del sistema físico (sin control).

A continuación se presenta el reograma del sistema automóvil-freno, utilizado en este trabajo:

#### REOGRAMA DEL SISTEMA FÍSICO A CONTROLAR

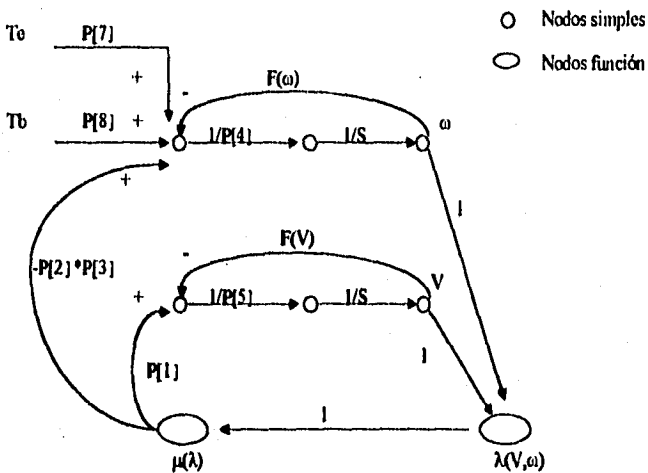


Diagrama 5.3.6.

Este modelo solamente considera la dinámica de una llanta, ya que nuestro interés se centra en el freno y no en el movimiento tridimensional del vehículo.

En este sistema físico la variable que se va a controlar es el

deslizamiento  $\lambda$ , que es lo que en último término ocasiona un mejor o más pobre desempeño de los frenos.

El módulo CMG6 de Pasion permite utilizar parámetros definidos por el usuario en las diferentes funciones de transferencia entre nodos, ya sea nodos de tipo simple o de tipo función.

Para este reograma se utilizaron los siguientes parámetros, como partes de las diferentes funciones de transferencia:

- p[1] = Fuerza normal que ejerce el vehículo sobre una llanta · número de ruedas con tracción.
- p[2] = Fuerza normal que ejerce el vehículo sobre cada llanta.
- p[3] = Radio de la llanta.
- p[4] = Inercia rotacional de la rueda.
- p[5] = Masa del automóvil que es acelerada.
- p[6] = Retardo ( en segundos ) para aplicar el freno.
- p[7] = Ganancia del par para el motor.
- p[8] = Ganancia del par del freno.
- p[9] = Ganancia de la función de fuerza aerodinámica.
- p[10] = Ganancia de la función de fuerza de fricción viscosa.

#### 5.4. Cálculos necesarios para generar el modelo de parámetros concentrados.

##### 5.4.1 Cálculo de inercias rotacionales equivalentes.

Para un modelo de una sola llanta, se toma en cuenta una sola inercia rotacional girando a la misma velocidad que la rueda ( $\omega_{11}$ ).

En esta inercia se debe incluir la inercia de las partes principales entre el motor y la llanta, inclusive: El volante de inercia, la transmisión -compuesta de 4 engranes-, la flecha que conecta la transmisión con la llanta y la llanta misma.

A continuación se presenta un resumen de los cálculos efectuados . El objetivo de estos cálculos es presentar la manera de hacerlos más que llegar a un resultado exacto.



## MASAS ROTATIVAS QUE SE CONSIDERAN:

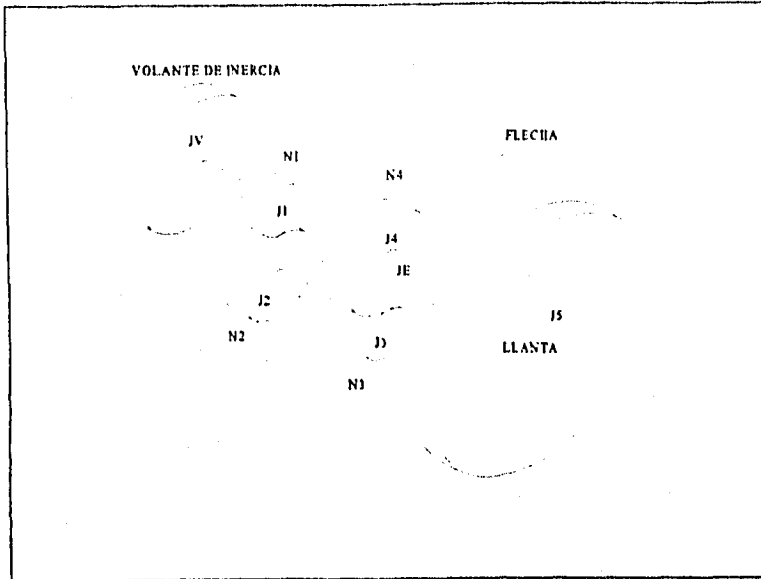


Diagrama 5.4.1. Masas consideradas en el cálculo de las inercias.

**Cálculo de la inercia rotacional.**

Elementos de la transmisión motor-rueda.

Suposiciones generales:

densidad del acero: (semejante a la del hierro)  
7870 kg / m<sup>3</sup>momento rotacional de inercia de un cilindro:  
 $J = m D^2 / 8$ Velocidad del motor:  
1000 rpm o bien  
104.71976 rad / s.

Cálculo de cada elemento:

1. Volante de inercia				
Diámetro:	0.34 m	JV=0.5162503	kg*m <sup>2</sup>	
Espesor:	0.05 m			
Masa:	35.73 kg			
2. Engrane 1				
Diámetro:	0.08 m	J1=0.0006329	kg*m <sup>2</sup>	
Espesor:	0.02 m			
Masa:	0.79 kg			
3. Engrane 2				
Diámetro:	0.12 m	J2=0.0032043	kg*m <sup>2</sup>	
Espesor:	0.02 m	N2/N1=1.5		
Masa:	1.78 kg			
4. Engrane 3				
Diámetro:	0.10 m	J3=0.0015453	kg*m <sup>2</sup>	
Espesor:	0.02 m			
Masa:	1.24 kg			
5. Engrane 4				
Diámetro:	0.10 m	J4=0.0015453	kg*m <sup>2</sup>	
Espesor:	0.02 m	N4/N3=1		
Masa:	1.24 kg			
6. Eje				
Diámetro:	0.05 m	JE=0.0031388	kg*m <sup>2</sup>	
Espesor:	0.65 m			
Masa:	10.04 kg			
7. Llanta				
Diámetro:	0.58 m	JLL=0.3364	kg*m <sup>2</sup>	
Espesor:	0.15 m			
Masa:	8.00 kg			

$$JEQ3 = JV + J1$$

$$JEQ3 = 0.5168832 \text{ kg*m}^2$$

incluye

JV

J1

$$JEQ2 = J2 + J3$$

$$JEQ2 = 0.0047495 \text{ kg*m}^2$$

$$JEQ4 = JEQ1 + JEQ2 (N4/N3)^2$$

$$JEQ4 = 0.3458336 \text{ kg*m}^2$$

incluye	incluye:
J2	JEQ2 $\omega_{eq} = \omega_{1i}$
J3	JEQ1
JEQ1 = JLL + JE + J4	JEQ5 = JEQ4 + JEQ3 (N2/N1)
JEQ1 = 0.3410841 kg*m2	JEQ5 = 1.5088209 kg*m2
incluye:	incluye:
J4 $\omega_{eq} = \omega_{1i}$	JEQ4
JE	JEQ3
JLL	

Por lo tanto, en un modelo de parámetros concentrados, se pueden sustituir las masas que se consideraron en el diagrama por una sola inercia rotacional (JEQ5), con la misma resistencia a girar, colocada en el lugar donde se encuentra la llanta, girando a la misma velocidad a la que giraría la llanta.

#### Cálculo de velocidades:

Velocidad de la rueda:	69.81317 rad/s
Velocidad del automóvil:	20.245819 m/s, o bien
	72.88495 km/h

#### 5.4.2 Fuerzas aerodinámicas.

Las fuerzas de origen aerodinámico tienen una influencia significativa en el desempeño de los vehículos, cuando viajan a velocidades moderadas o altas.

La resistencia aerodinámica de un vehículo se deriva de tres fuentes principales:

- 5.4.2.1 Resistencia debida a la forma, que es causada por la turbulencia dejada por el vehículo a su paso. Depende de la forma del cuerpo, particularmente de la parte de atrás. Normalmente, ésta es la parte más significativa de la resistencia aerodinámica.
- 5.4.2.2 Fricción superficial, que es causada por las fuerzas de corte que son ejercidas sobre las superficies exteriores del vehículo, por las corrientes de aire. Para un acabado superficial normal en los automóviles de pasajeros, este componente es aproximadamente el 10% de la resistencia aerodinámica total.
- 5.4.2.3 Resistencia debida al flujo de aire a través del sistema de radiador o del interior del vehículo con propósitos de enfriamiento y ventilación. Depende de la manera como se diseñe el flujo de aire. Esta forma de resistencia, representa un

porcentaje muy pequeño de la fuerza total de resistencia aerodinámica.

La resistencia aerodinámica ( $R_a$ ) se puede expresar de la siguiente manera:

$$R_a = \frac{\rho}{2} C_d A_f V^2$$

donde:

- $\rho$  es la densidad del aire,
- $C_d$  es el coeficiente de resistencia aerodinámica que representa el efecto de los 3 componentes de la resistencia aerodinámica descritos anteriormente,
- $A_f$  es el área característica, normalmente considerada como el área frontal que se proyecta en la dirección del movimiento,
- $V$  es la velocidad del vehículo relativa al viento.

De acuerdo al tipo de vehículo y a su forma, se han obtenido algunos datos representativos:

Tipo de vehículo	$C_d$ (adimensional)
Automóvil de pasajeros	0.3-0.6
Convertible	0.4 -0.65
Automóvil de pasajeros	0.25-0.3
Autobús	0.6-0.7
Camión de carga	0.8-1.0
Trailer	1.3
Motocicleta y conductor	1.8

Para el modelo que se utiliza en la tesis, podemos escoger un  $C_d$  de 0.458. Lo más común es usar condiciones atmosféricas ( que influyen en la densidad del aire ) estándar. Éstas son:

temperatura de 288.5 °K  
 presión barométrica de 101.32 kPa.

Entonces, @ (T=288.5 K, P=101.32 kPa),  $\rho=1.2315 \text{ kg / m}^3$

El área frontal, depende del vehículo que se trate. Para un vehículo de 1.5 m de ancho y 1.38 m de altura  $A_f=2.07 \text{ m}^2$ .

Utilizando la ecuación anterior para distintas velocidades y

manteniendo las condiciones estándar anteriormente descritas, se tiene:

V (m/S)	Ra= ( $\rho/2$ ) C <sub>D</sub> A <sub>f</sub> V <sup>2</sup> (N)
10	58.38
20	233.51
26.82 (60 mph)	419.99
27.78 (100 km/h)	450.44
40 (144 km/h)	934.03

## 5.5 Funciones especiales utilizados en el programa.

### 5.5.1 Función del par del motor.

Una de las entradas del sistema es el par aplicado por el motor a las inercias rotacionales que van a ser aceleradas. En esta función se considera que el par del motor es constante y se representa mediante una función escalon, que vuelve a ser cero a partir de cierto momento, indicado por uno de los parámetros del modelo, el parametro P[11].

A continuación se muestra el código en Pascal de la función motor (time), utilizada en la simulación:

```
FUNCTION MOTOR (TIME: REAL):REAL;
begin
  if time<=p[11] then motor:=step(time)
  else motor:=0;
end;
```

### 5.5.2 Función para aplicar el par de frenado, cuando no existe control ABS:

Cuando no existe control ABS, el par de frenado que se aplica no depende del deslizamiento. Sin embargo, hay que tener en cuenta que este par tiene un valor máximo que depende de la presión en el freno, la velocidad relativa entre el tambor del freno y la balata y el material del freno. Este par máximo se aplica cuando la velocidad angular es mayor a cero, y cuando ésta está dentro de una vecindad  $\Delta e=0.001$  del cero, se deberá oponer al movimiento, calculando primero el par que se está aplicando a la rueda. Si este par es menor al par máximo aplicable, simplemente se aplica un par igual y de sentido contrario. Si es mayor, se aplica el par máximo, pero con signo contrario al par aplicado sobre la rueda, ya que así se cubren los casos en que la velocidad angular es negativa. Para

que funcione este freno cercano a  $\omega=0$ , se requiere que en el cuerpo del programa generado por CMG, se añada freno (time,w) al nodo de la rueda, en último lugar, para calcular todos los pares que actúan en ella y que podrían causar movimiento. En este caso, esos pares se acumulan en el nodo INN [13] antes de aplicar el freno.

El par de frenado se empieza a aplicar después de cierto momento, indicado por el parámetro del modelo P[6]. Este momento correspondería al momento en el que el conductor acciona el pedal del freno.

A continuación se muestra el código en Pascal de esta función:

```
FUNCTION FRENO (TIME:REAL; W: REAL):REAL;
(INN[13] ES LA SUMA DE MOMENTOS
 QUE ACTÚAN SOBRE LA RUEDA HASTA
 UN MOMENTO ANTES DE FRENAR)
```

```
begin
  if time > p[6] then
    begin
      if W>0 then freno:=step(time);
      if W<0.001 then
        begin
          If abs(inn[13])<abs(p[8]) then freno:=-inn[13]/p[8]
          else freno:=-1*signo(inn[13]);
        end;
      end
    else
      freno:=0;
  end;
end;
```

### 5.5.3 Función para simular el comportamiento del freno controlado (FRENOACT)

El controlador consta de un microprocesador que recibe las señales de los sensores y realiza los cálculos necesarios, de acuerdo a un algoritmo de control. Después manda una señal de control a dispositivos actuadores, que en este caso puede ser un modulador de presión, para que haga que el freno tenga un mayor o menor par de frenado.

La función FRENOACT trata de describir la manera como se comportaría el freno controlado, en función de la señal recibida de la unidad de control y de la velocidad angular que en ese momento

la unidad de control y de la velocidad angular que en ese momento experimente la rueda.

#### Características importantes:

- El par máximo aplicable depende de la configuración física del freno. Entonces, puede haber señales de control que demanden pares de frenado mayores de los que el freno puede producir. Entonces, estas señales deben ser traducidas a la aplicación del par máximo posible.
- Si el vehículo está próximo al reposo, el par de frenado no deberá producir un movimiento hacia atrás (en sentido contrario del movimiento del vehículo). Esto, físicamente se realiza sin necesidad de calcular nada, pues en circunstancias de proximidad al reposo, el freno se comporta como una fuerza de reacción, proporcional al par experimentado por el eje de la rueda. Este fenómeno se toma en cuenta en esta función, para evitar comportamientos alejados de la realidad.

En este modelo de freno controlado, no se entra en detalles acerca de la dinámica del modulador de presión que precede a la aplicación de una fuerza de frenado y se supone que la función de transferencia entre la señal de control y el par de frenado es una constante de proporcionalidad estática.

#### Significado de los parámetros:

- P[22]: Es el par máximo que puede aplicar el freno con una configuración física fija.
- P[21]: Es la ganancia entre el par de frenado y la señal de control que recibe el modulador de presión.
- P[6]: Es el instante en el que inicia el proceso de frenado. Equivale al instante en el que el conductor acciona el freno, pisándolo hasta el fondo, para tener acceso al proceso de frenado en la menor distancia posible.
- xst[1] Representa la variable del vector estado que corresponde a la velocidad angular de la rueda.
- inn[3] En el programa principal generado por CMG6 corresponde al nodo en el que se aplican sobre la rueda los siguientes pares: la resistencia a la rodadura, el par del motor, el par de frenado, y la resistencia viscosa, que es proporcional a la velocidad angular.

La función FRENOACT tiene un parámetro: *signal*. Este valor corresponde a la señal de control, que es la señal de salida del algoritmo de control.

A continuación se presenta el código en Pascal de esta función:

```

FUNCTION FRENOACT (signal: REAL):REAL;
VAR
  temp,temp2:real;
  sgn:integer;
  retraso: integer;
  ya:boolean;
begin;
  temp2:=signal*p[21];
  temp:=signal*p[21];
  retraso:=0;
  ya:=false;
  if temp>=0 then sgn:=1 else sgn:=-1;
  if abs(temp)>p[22] then temp2 :=p[22]*sgn;
  if temp>0 then temp2 :=0;
  ya:=time>=p[6];
  if ya then retraso:=1;
  if xst[1]>=0.001 then
    temp2 :=temp2 *retraso
  else
    begin
      if abs( inn[3])<abs(p[21]*p[22]) then temp2:=-inn[3]
      else temp2 :=-signo(inn[3])*p[22];
      temp2:=temp2*retraso;
    end;
  frenoact:=temp2;
end;

```

#### 5.5.4 Función para calcular el deslizamiento (LAMBDA).

Como ya se había mencionado antes (ec. 5.1), el deslizamiento es una medida de comparación entre la velocidad angular de la rueda y la velocidad angular equivalente del vehículo:

$$\lambda = \frac{\omega_r - \omega_v}{X} \quad (5-3)$$

donde X es el mayor de  $\omega_v$  y  $\omega_r$ .

$\lambda$  resulta ser un indicador normalizado cuya función es comparar  $\omega_r$  y  $\omega_v$ . Por lo tanto,  $-1 < \lambda < 1$ , pues el deslizamiento puede ser



positivo o negativo, dependiendo de si se trata de una situación de aceleración o desaceleración, respectivamente.

Sin embargo esta descripción no es suficiente. Supóngase el caso en que  $\omega_w < 0, \omega_v < 0$  y además  $|\omega_v| > |\omega_w|$ . De acuerdo a la ecuación (5-3),  $X = \omega_v$ , ya que  $\omega < \omega_v$ . Esto ocasionaría que  $\lambda$  no fuera la variable normalizada que pretende ser.

Como  $\omega_v$  y  $\omega_w$  pueden ser números reales, existen 7 casos posibles que son importantes. A cada caso le corresponde una respuesta deseada:

Positivo o negativo	Otras condiciones	$\lambda$	situación	caso
$\omega_w > 0$ y $\omega_v > 0$	$\omega_w > \omega_v$	$\lambda > 0$	Aceleración	1
	$\omega_w < \omega_v$	$\lambda < 0$	Desaceleración	2
$\omega_w < 0$ y $\omega_v < 0$	$\omega_w > \omega_v$	$\lambda > 0$	Aceleración (frenado en reversa)	3
	$\omega_w < \omega_v$	$\lambda < 0$	Desaceleración (aceleración en reversa)	4
$\omega_w = 0$	$\omega_v < 0$	$\lambda = 1$	Aceleración	5
	$\omega_v > 0$	$\lambda = -1$	Desaceleración	6
$\omega_v = 0$	$\omega_w < 0$	$\lambda = -1$	Desaceleración	7
	$\omega_w > 0$	$\lambda = 1$	Aceleración	8
$\omega_w < 0$ y $\omega_v > 0$	$ \omega_v  >  \omega_w $	$\lambda < 0$	Desaceleración	9
	$ \omega_v  <  \omega_w $	$\lambda < 0$	Desaceleración	10
$\omega_w > 0$ y $\omega_v < 0$	$ \omega_v  >  \omega_w $	$\lambda > 0$	Aceleración	11
	$ \omega_v  <  \omega_w $	$\lambda > 0$	Aceleración	12
$\omega_w = \omega_v = 0$		$\lambda = 0$	El coeficiente de fricción es cero.	13

#### Análisis de los diferentes casos:

El objeto del siguiente análisis es determinar en qué casos la ecuación (5-3) es adecuada a la situación dada.

**Casos 1 y 2.** La ecuación (5-3) se cumple sin necesidad de hacer modificaciones.

**Casos 3 y 4.** La ecuación (5-3) no se cumple, ya que  $\lambda$  resulta de signo contrario al esperado. Esto se resuelve cambiando de signo la fórmula para este caso en especial.

**Casos 6 y 8.** La ecuación (5-3) sí se cumple.

**Casos 5 y 7.** La ecuación (5-3) no se cumple ( $\lambda$  es de signo contrario). El procedimiento va a asignar a  $\lambda$  el valor cero. En el caso 5, esto no es importante, ya que en una situación de frenado, el procedimiento FRENOACT se encargará de mantener a la rueda bloqueada mientras  $\omega=0$ . El caso 7 no llega a darse en una situación de frenado, utilizando pasos de integración pequeños.

**Casos 9, 10, 11 y 12.** Aunque  $\lambda$  tiene el signo correcto, en todos estos casos  $|\lambda|>1$ , con lo que ya no cumple con la condición de ser un parámetro normalizado. Además, el valor de  $\lambda$  sería un valor incierto, ya que  $\omega$  sólo está definida para valores de  $\lambda$  entre -1 y 1. Por eso, en la función se considera que en estos casos,  $\lambda$  debe adoptar arbitrariamente los valores de 1 o -1.

**Caso 13.** La ecuación (5-3) produciría una división entre cero. Para evitar esto, en la función se asigna a  $\lambda$  un valor de cero si  $-0.005 < X < 0.005$ . Esto también evita errores computacionales al dividir dos números muy pequeños entre sí. A continuación se muestra el código en Pascal para esta función:

```
function lambda(v,w:real):real;
var
  xx:real;
  zero:boolean;
begin
  zero:=false;
  if (w>~v/p[3]) then xx:=w
    else xx:=v/p[3];
  if abs(xx)~0.005 then
    begin {2}
      lambda:=(w-v/p[3])/xx;
      if ((w<0) and (v/p[3]<0)) then lambda:=(v/p[3]-w)/xx;
      if (signo(w)*signo(v/p[3])<0) then lambda:=signo(w);
    end; {2}
  else
    begin {3}
      lambda:=0;
      if abs(v/p[3])>0.005 or (abs(w)>0.005) then
        lambda:=signo(w-v/p[3]);
    end {3}
  end;
end;
```

### 5.5.5 Función para calcular $\mu$ ( $\mu$ ).

La función  $\mu(\text{dist}, \lambda)$ , permite calcular el coeficiente de fricción entre la llanta y el suelo en función del deslizamiento  $\lambda$ . Además el parámetro *dist* permite introducir perturbaciones aleatorias, que son producto de la variación también aleatoria de las condiciones del suelo. Los procedimientos seco, mojado y hielo solamente consisten en asignar los valores a las variables *a*, *m*, *pp*, *b* y *x*, de la manera descrita en el inciso 5.2. El parámetro *p[23]* sirve para controlar la magnitud del mayor disturbio aleatorio posible.

A continuación se muestra el código en Pascal de esta función:

```
function mu(DIST:REAL,l:real):real;
var
  a,m,pp,b,x,mu: real;
begin {mu}
  case piso of
    1: seco;
    2: mojado;
    3: hielo;
    4: azar;
  end; {case}
  if l>x then mu:=m*l+b;
  if (l>=-x) and (l<=x) then mu:=a*sin(pp*l);
  if l<-x then mu:=m*l-b;
  if abs(mu)>0.1 then mu:=-mu-(dist*p[23]) else mu:=mu;
end; {function mu}
```

### 5.5.6. Función para calcular $f(v)$

Esta función calcula la resistencia aerodinámica de acuerdo a la ecuación presentada en la sección 5.4.2. El código es el siguiente:

```
function fv(v:real):real;
begin
  fv:=p[9]*SQR(V);
end;
```

5.5.7 Función para calcular  $f(w)$ 

Calcula la resistencia viscosa al movimiento:

```
function fw(w:real):real;
begin
  FW:=P[10]*W;
end;
```

## 5.5.8 Función LT:

Calcula la diferencia entre el deslizamiento ( $\lambda$ ) deseado y el deslizamiento real:

```
function lt(desliz:real):real;
begin
  lt:=desliz-P[20];
end;
```

## 5.5.9 Función Fb

Calcula la señal de control que se manda al modulador de presión del freno, de acuerdo al algoritmo de control:

```
function fb(wk:real):real;
var
  x1, b3: real;
begin
  x1:=xst[2]/p[3];
  b3:=1/p[4];
  fb:=-wk*xst[2]/(p[3]*p[4]);
end;
```

# CAPÍTULO 6

ESTO TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

## 6. SIMULACIÓN DEL SISTEMA FÍSICO CON EL CONTROL DE TIEMPO DISCRETO.

### 6.1 ALGORITMO DE CONTROL

El algoritmo de control de un sistema de control digital es la esencia del control. El algoritmo influye de manera importante en la robustez del control y en su facilidad de aplicación.

En este caso, la ventaja de este algoritmo de control es la manera de manejar la incertidumbre de las condiciones que pueden fungir como disturbios: las condiciones del suelo y de la llanta, que determina el coeficiente de fricción entre ambos, y la carga del vehículo, principalmente. A su vez, en las condiciones de la llanta pueden influir otros factores, como la presión del aire de la llanta, la temperatura dentro de la llanta, la velocidad del automóvil, etc.

El algoritmo de control utilizado es un algoritmo de tiempo discreto, y es el siguiente:

$$w_{k+1} = w_k - K_f (\lambda_{k+1} - \lambda_{k+1}) - (\lambda_{k+1} - \lambda_k) / \Delta t$$

para cada instante de tiempo  $t$ .

A su vez,  $w_{k+1}$  es la entrada para la función  $Fb\{w\}$ , que es la señal que manda el control al regulador de presión del freno:

$$Fb(w_{k+1}) = w_{k+1} \frac{-J \omega}{R_H} \quad (6-1)$$

6.2. REOGRAMA

El diagrama que se utiliza para mostrar el flujo de las señales a lo largo del sistema, o reograma, es el siguiente:

REOGRAMA DEL SISTEMA DE CONTROL

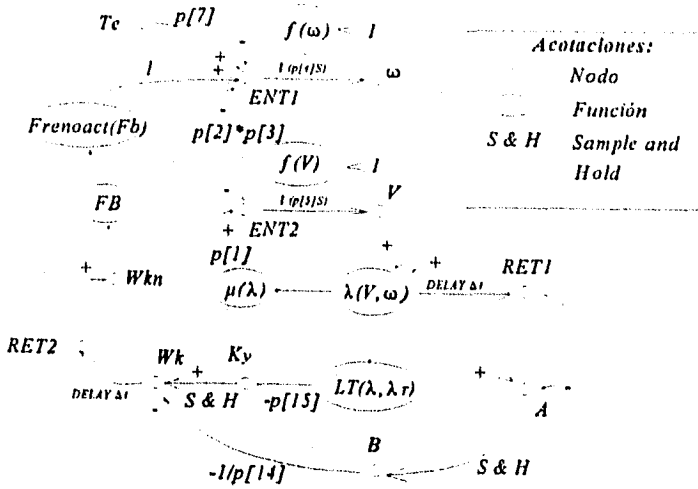


Diagrama 6.2.1.

En el diagrama anterior, cuando no se indica una función de transferencia se supone igual a 1.

6.3. PARÁMETROS QUE SE UTILIZAN:

Los parámetros que se usan en este modelo, son los mismos que en el anterior, pero además se utilizan los siguientes:

- P[12] Se utiliza en la función de muestreo y almacenamiento (Sample & hold) de la rama entre Ky y Wk.
- P[13] Se utiliza en la función de muestreo (S & H) en la rama entre A y B.
- P[14] La rama entre B y Wk tiene ganancia  $-1/p[14]$ . Su valor es igual al del incremento de tiempo  $\Delta t$  utilizado en las funciones de muestreo (S & H).
- P[15] La ganancia entre los nodos LT y KY es  $-p[15]$ .

- P[16] Este parámetro se utiliza en la función LT. P[16] es el valor del deslizamiento que el control tratará de lograr.
- P[17] La función de transferencia desde WK hasta RET2 es de tipo retraso (*delay*). En el parámetro p[17] se almacenan los valores de manera momentánea.
- P[18] Se utiliza en la función S&H, entre KY y WK.
- P[19] Se utiliza en la función S&H, entre A y B.
- P[20] Este parámetro se utiliza en la función LT. p[16] es el valor del deslizamiento que el control tratará de alcanzar.
- P[21] Se utiliza en la función FRENOACT. Corresponde a la ganancia entre la señal de control y el par de frenado. Debe ser negativa.
- P[22] Este parámetro también se utiliza en la función FRENOACT. Es el par máximo aplicable por el freno, que depende de las características físicas del freno.

#### 6.4 MEJORAMIENTO DEL ÍNDICE DE COMPORTAMIENTO

El controlador tiene como finalidad aproximar el valor del deslizamiento  $\lambda$  a un valor óptimo  $\lambda_r$ , a pesar de los cambios inciertos en las condiciones de operación. Se necesita algún parámetro de comparación que nos permita evaluar su desempeño. Se ha escogido la distancia de frenado como índice de comportamiento del controlador. Esta función se define de la siguiente manera:

$$D = \int_{t_0}^{t_f} V(t) dt \quad (6-2)$$

donde,

- t es el tiempo,  
 t<sub>0</sub> y t<sub>f</sub> son los límites del intervalo de integración, es decir, el intervalo de tiempo en el que se evalúa el desempeño,  
 V(t) es la velocidad lineal.

El uso de este medida como índice de desempeño representa una manera indirecta de evaluar el desempeño del controlador enfocando la atención a la función objetivo que se desea mejorar. Se hicieron pruebas utilizando el modelo del controlador expuesto anteriormente, y calculando la distancia de frenado en cada prueba, para apreciar los cambios en el índice de comportamiento del controlador, al variar los valores asignados a los parámetros de dicho controlador:



- Ganancia del actuado del freno ( $p[21]$ ).
- Ganancia  $k_c$  dentro del algoritmo de control ( $p[15]$ ).
- Tiempo de muestreo  $\Delta t$ , del control digital, que se identifica con los siguientes parámetros:  
 $P[12]$ ,  $P[13]$ ,  $P[14]$  y  $P[17]$ .
- Par máximo aplicable por el freno.

#### 6.4.1. Descripción de las pruebas:

Para evaluar dicha habilidad de controlar, se simula el comportamiento del sistema físico y del controlador, utilizando las siguientes condiciones iniciales:

$$V(t=0) = 27.776 \text{ m/s} = 100 \text{ km/h.}$$

$$\omega(t=0) = 107.9469 \text{ rad/s}$$

$$D(t=0) = 0, \text{ que son las variables del vector estado.}$$

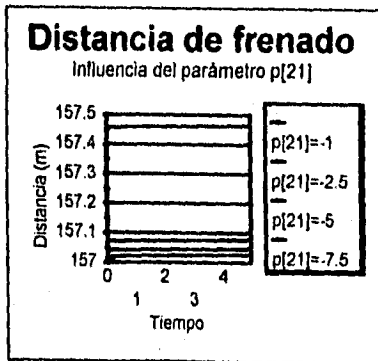
La simulación se lleva a cabo durante 15 segundos, utilizando la función  $\mu(\lambda)$  correspondiente a piso seco, mojado y con hielo, sucesivamente.

El vehículo mantiene su velocidad de manera constante hasta que en el segundo 2, se interrumpe el torque del motor y en el segundo 3 se aplica el freno, que tratará de llevar a la variable  $\lambda$  a  $\lambda_s = -0.175$ , que es el valor que produce el máximo coeficiente de fricción sobre un piso promedio.

Una vez que se llevó a cabo la prueba, se escoge el valor del parámetro más conveniente y se utiliza este mismo valor para continuar con las pruebas posteriores.

#### 6.4.2 Resultados de las pruebas.

##### 6.4.2.1 PRUEBAS EN PISO SECO:



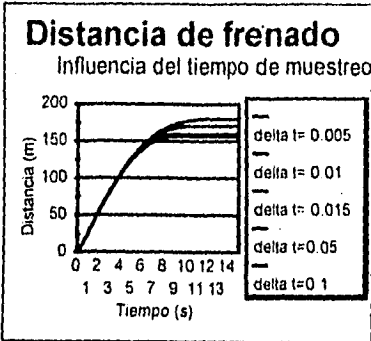
##### Prueba # 1

Se hizo variar el parámetro que corresponde a la ganancia del actuador del freno, usando los siguientes valores:  $p[21] = -7.5, -5, -2.5$  y  $-1$ , sucesivamente.

Los resultados de estas pruebas se muestran en la gráfica 6.4.1: el mejor parámetro fue  $P[21] = -1$ .

Gráfica 6.4.1.

Prueba # 2.

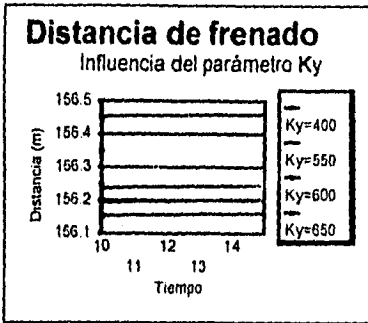


Gráfica 6.2.

Se corrieron varias pruebas con distintos valores para el tiempo de muestreo, que, en el modelo del controlador, corresponde a los siguientes parámetros: P[12], P[13], P[14], P[17]. Se le asignaron consecutivamente los siguientes valores:  $\Delta t=0.005$ , 0.01, 0.015, 0.05, 0.1 s. (Gráfica 6.2).

Es más conveniente utilizar un tiempo de muestreo de 0.005 s.

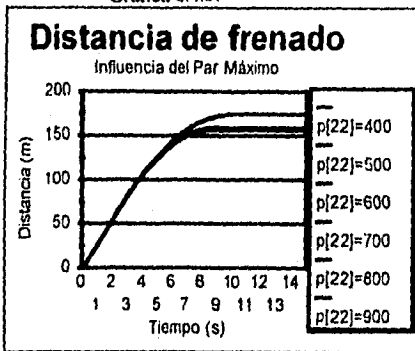
Prueba # 3.



Gráfica 6.4.3.

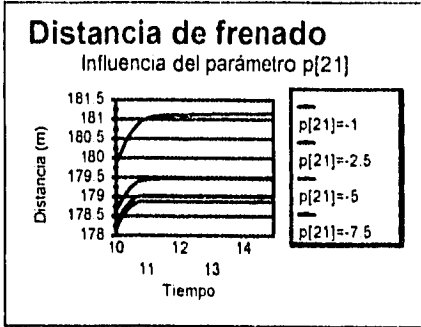
A continuación se probó el algoritmo utilizando los siguientes valores para el parámetro  $k_y$ , que es parte del algoritmo de control. Los resultados se ilustran en a Gráfica 6.4.3. Es claro que, sin tomar en cuenta otros factores, el mejor valor para el parámetro  $K_y$  650.

Prueba # 4.



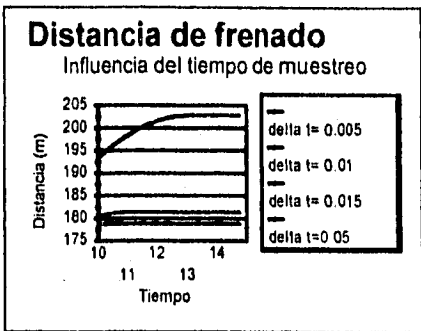
En esta prueba se hizo variar el parámetro correspondiente al par máximo aplicable por el controlador del freno desde 400 N·m hasta 900 Nm. El par máximo aplicable que produce los mejores resultados es el de 600 N·m.

6.4.2 PRUEBAS EN ASFALTO MOJADO



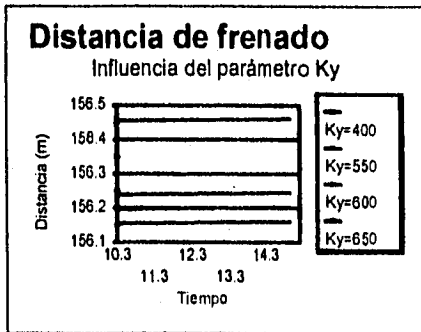
Prueba # 5.

Se hicieron los mismos cambios sobre el parámetro p[21] que en la prueba # 1, y, a diferencia de ésta, los resultados favorecen más a la opción P[21]=-7.5.



Prueba # 6.

Es análoga a la prueba # 2. La mejor distancia de frenado está relacionada con  $\Delta t=0.005$ , como es de esperarse.

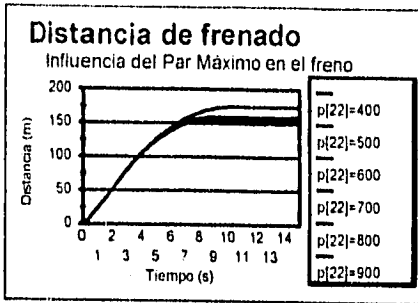


Prueba # 7.

En este caso, el mejor parámetro Ky es 650, mayor que el obtenido en la prueba # 4.

## Prueba # 8.

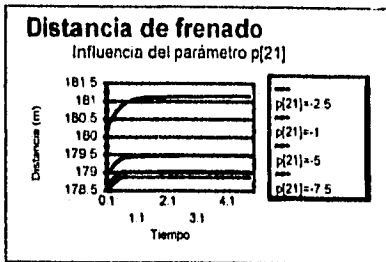
Aquí el mejor parámetro par máximo para frenar es 700 N·m.



## 6.4.3 PRUEBAS EN ASFALTO CON HIELO.

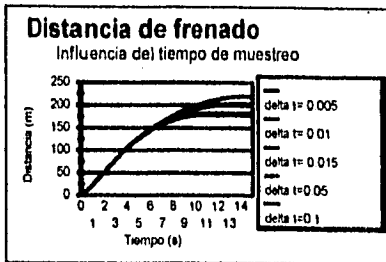
## Prueba # 9.

Se hicieron los mismos cambios sobre el parámetro p[21] que en la prueba # 1, y, a diferencia de ésta, los resultados favorecen más a la opción P[21]=-7.5.



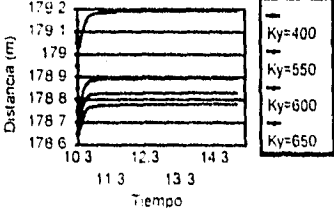
## Prueba # 10.

Es análoga a la prueba # 2. La mejor distancia de frenado está relacionada con  $\Delta t=0.005$ , como es de esperarse.



### Distancia de frenado

Influencia del parámetro  $K_y$

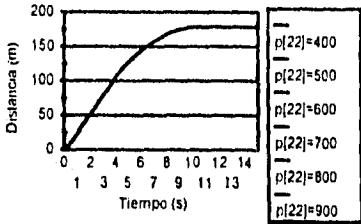


Prueba # 11.

En este caso, el mejor parámetro  $K_y$  es 650, mayor que el obtenido en la prueba # 4.

### Distancia de frenado

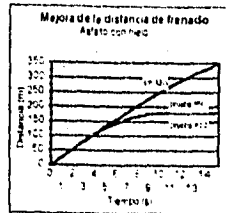
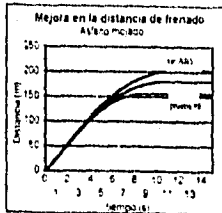
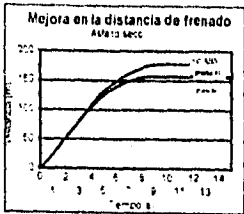
Influencia del Par Máximo en el freno



Prueba # 12.

Aquí el mejor parámetro par máximo para frenar es 700 N·m.

Finalmente, es interesante comparar el desempeño de un freno ABS con un freno convencional. Esta comparación se presenta a continuación, en donde se muestra también la mejora sucesiva de las pruebas 1 a 12, en 3 tipos de superficies diferentes.



#### 6.4.3 Determinación de los parámetros de mayor influencia en el desempeño del controlador.

Las pruebas realizadas anteriormente no solamente permiten un mejoramiento de indicadores de desempeño, sino también muestran la mayor o menor influencia de cada uno de los parámetros anteriores en el desempeño del frenado.

A continuación se muestra una tabla en la que se toman los parámetros que variaron en las pruebas anteriores y se compara la razón

$$\frac{\Delta i_j}{\Delta p_j} \quad (6-3)$$

donde

$\Delta i_j$ , es el incremento en el índice de desempeño,  
 $\Delta p_j$ , es el incremento en el j-ésimo parámetro.

## INFLUENCIA DE LOS DISTINTOS PARÁMETROS EN EL DESEMPEÑO:

En  $t=14.99$  (para asfalto mojado)

PARÁ-METRO	No.	VALOR	D (m)	RAZÓN DE CAMBIO	MEJOR VALOR:
Ganancia del actuado del freno	21	-1	181.15061	1.11648	-7.5
		-2.5	179.47589	0.17793	
		-5	179.03107	-2.84617	
		-6.25	182.58878	2.96662	
		-7.5	178.88050		
Tiempo de muestreo	12, 13, 14, 17	0.005	178.83203	533.09800	0.005
		0.010	181.34752	0.00000	
		0.015	181.34752	612.73029	
		0.050	202.79308	340.18400	
		0.100	203.80228		
Ky (Algo-ritmo de control)	15	400	156.46015	-0.00143	650
		550	156.24496	0.00085	
		600	156.20262	-0.00083	
		650	156.16095		
Par máximo permisible en el freno.	22	400	174.26548	-0.15501	700
		500	158.76486	-0.02660	
		600	156.10509	-0.00050	
		700	156.05511	0.00096	
		800	156.15100	-0.00082	
		900	156.06923		

La distancia de frenado mínima es de 156.0551 m.

De acuerdo a lo anterior, se compararán las razones de cambio (6-3) de los distintos parámetros, en el punto más cercano al valor del parámetro que produce el error menor:

Parámetro:	Razón de cambio
Ganancia del actuado del freno	2.96662
Tiempo de muestreo	503.09800
Ky	-0.00083
Par máximo permisible en el freno	0.00096

Podemos ver, después de las pruebas anteriores, que el mejor desempeño se consigue al utilizar los siguientes valores para los parámetros mencionados:

$P[21] = -7.5$   
 $Ky = 650$   
 $P[22] = 700$   
 Tiempo de muestreo = 0.005 s

Además se puede deducir que los parámetros de mayor influencia, de acuerdo a la distancia de frenado son el tiempo de muestreo y la ganancia del freno, ya que, cerca del mejor punto de operación encontrado hasta ahora, la razón  $\Delta I_d / \Delta p_i$  es mayor.

#### 6.5 OBTENCIÓN DE LA SUPERFICIE DE RESPUESTA, EN FUNCIÓN DE LOS DOS PARÁMETROS DE MAYOR INFLUENCIA.

Dado que los parámetros de mayor influencia en el índice de desempeño son el tiempo de muestreo y la ganancia del freno, ahora se harán variar dichos parámetros en la vecindad del menor punto de operación encontrado hasta ahora, para generar una superficie, definida como:

$$I_d = f(t, p[21])$$

donde  $I_d$  es el índice de desempeño más adecuado para comparar el desempeño del vehículo en pruebas de

frenado con una duración cuya variación es significativa; la distancia de frenado.

$\Delta t$  representa el tiempo de muestreo, pero utilizando un paso de integración de 1 ms.

$P[21]$  es la ganancia del freno, es decir, la relación entre el par de frenado y la señal que manda el controlador al modulador de presión del freno.

Así, haciendo variar  $\Delta t$  desde 4 ms hasta 13 ms, y haciendo variar  $Ky$  desde -1 hasta -10, obtenemos un conjunto de 100 parejas ordenadas. Para cada pareja ordenada, se hizo una corrida del programa de simulación con las siguientes características:



- Paso de integración: 1 ms.
- Método de integración: Runge-Kutta-Fehlberg de orden 5.
- Velocidad angular inicial: 107.9469 rad/s.
- Velocidad inicial: 27.776 m/s.
- Tiempo de simulación: 10 segundos.
- Tiempo en que se deja de aplicar el par del motor:  $t = 2$  s.
- Tiempo en que empieza el frenado:  $t = 3$  s.
- Índice de comportamiento graficado en el eje z: Distancia de frenado.

En el gráfico 6.5.1 se puede observar cómo a medida que disminuye el tiempo de muestreo, el tiempo de frenado es menor y cómo a medida que el valor absoluto de la ganancia del freno es menor, el tiempo de frenado es mayor y la superficie se hace más irregular.

También, con los datos obtenidos (gráfico 6.5.1), podemos nuevamente escoger el punto con mejor desempeño. Éste es:

p [21]=-10  
 Ky=550  
 p [22]=600  
 Tiempo de muestreo=0.006 s

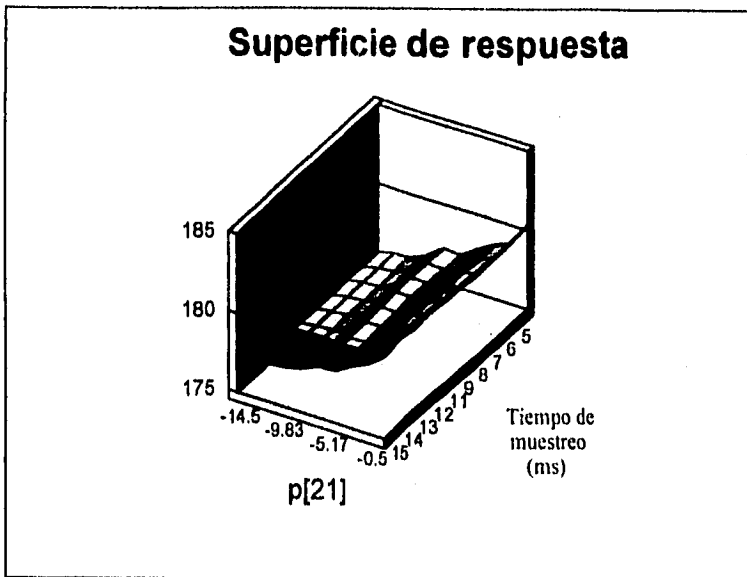


Gráfico 6.5.1.

## 6.6 Influencia de perturbaciones aleatorias.

Este algoritmo fue diseñado para ser poco sensible a las variaciones ocasionadas por fluctuaciones de las condiciones de operación del vehículo o del medio ambiente: carga del vehículo, presión de inflado de las llantas, características de fricción de la superficie, dibujo de las llantas, temperatura de las llantas, etc. En este apartado se pretende examinar la influencia de variaciones aleatorias del coeficiente de fricción  $\mu$  y sus efectos en el índice de desempeño.

### 6.6.1 Descripción del experimento.

En este primer experimento se cambia la función  $\mu(\lambda)$  por la función  $\mu(\lambda, \text{pert})$ , donde  $\text{pert}$  es una variable aleatoria, con una distribución de probabilidad en forma de escalón unitario. Esta variable requerirá de introducir un nuevo nodo al diagrama de flujo de señales (ver diagrama 6.6.1).

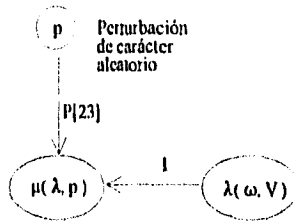


Diagrama 6.6.1.

Aunque  $\mu$  seguirá dependiendo principalmente del deslizamiento  $\lambda$ , se supondrá que habrá pequeñas perturbaciones aleatorias, de magnitud pequeña, que se suman a la función  $\mu(\lambda)$ :

$$\mu(\lambda, p) = \mu(\lambda) + \text{pert} \cdot p[23]$$

donde  $\mu(\lambda)$  es la misma función que se definió anteriormente,  
 $\text{pert}$  es una variable aleatoria entre 0 y 1, con igual probabilidad para cualquier punto dentro de este intervalo,  
 $p[23]$  es una constante que sirve para controlar la magnitud máxima posible de la mayor perturbación aleatoria.

Estas variaciones aleatorias tratan de imitar la influencia de la naturaleza irregular de una superficie real. El objeto del experimento es averiguar cuál es el efecto de estas perturbaciones aleatorias en variaciones en la distancia de frenado.

Indicador de desempeño utilizado.

Como indicador de desempeño a utilizar se escogió el más utilizado en evaluar el desempeño de los frenos: la distancia de frenado.

Manera de comparar un desempeño con otro:

Se consideraran datos provenientes de 30 corridas, llevadas a cabo con las mismas condiciones ( con los mismos parámetros ). En cada corrida se registrara la trayectoria de la variable distancia a lo largo del tiempo. Una trayectoria es el conjunto de parejas  $(t, d)$ , con  $i=0, \Delta T, 2\Delta T, 3\Delta T, \dots, t$ , donde  $T$  es el paso de integración y  $t_i$  es el instante de tiempo en que se termina la simulación.

Cada trayectoria se almacena en un archivo con extensión \*.sim para que posteriormente sea analizado mediante el programa VARAN5, que es parte del ambiente PASION.

El programa VARAN5 calcula la trayectoria promedio, que se puede definir como el con junto de puntos  $(t, d_{i, PROM})$ , con  $i=0, \Delta T, 2\Delta T, 3\Delta T, \dots, t_i$  donde

$$d_{i, PROM} = \frac{1}{n} \sum_{j=0}^n d_{ij}$$

y  $j$  es el número de trayectorias diferentes.

De manera similar, el programa VARAN5 calcula la varianza para cada punto, los conjuntos alcanzables y los intervalos de confianza a un nivel de confianza seleccionable.

En este experimento se harán 30 corridas (30 trayectorias diferentes) para cada parámetro  $P[23]$  diferente. Este parámetro controla la magnitud máxima posible de la perturbación aleatoria. En cada conjunto de 30 trayectorias se comparará el  $i$ -ésimo punto (con su media y varianza) con el  $i$ -ésimo punto de los otros conjuntos de 30 trayectorias.

Condiciones de simulación:

- Paso de integración: 1 ms.
- Método de integración: Runge-Kutta-Fehlberg de orden 5.
- Velocidad angular inicial: 70 rad/s.
- Velocidad inicial: 10.71 m/s.
- Tiempo de simulación: 10 segundos.
- Tiempo en que se deja de aplicar el par del motor:  $t=2$  s.
- Tiempo en que empieza el frenado:  $t = 3$  s.

## Pruebas.

Se probaron perturbaciones aleatorias con una magnitud máxima posible de 0.01, 0.05 y 0.075, que corresponden a una variación sobre el coeficiente máximo de fricción posible del 1.1%, 5.6%, y 8.3% respectivamente. Se tomaron 30 muestras para cada caso, y se observó un cambio en la distancia de frenado media, lo mismo que en las varianzas asociadas a cada caso. Las variaciones mínimas y máximas expuestas, son el límite superior e inferior de un intervalo de confianza del 90%.

Variación	Distanciado (m )	Límites del intervalo
1.1%	34.4337	34.4337
		34.4337
5.6%	34.5951	34.6379
		34.5524
8.3%	34.6627	34.7216
		34.6039

De esta manera, los resultados de una simulación pueden ser más realistas y nos muestran la habilidad del sistema vehículo-freno ABS para manejar las perturbaciones aleatorias. En el peor caso, una perturbación en el coeficiente de fricción del 8.3%, sólo causa una variación del 0.34%, con una confiabilidad del 90%. Es decir, el 90% de las veces, la variación en la distancia de frenado ocasionada por variaciones del 8.3% será de menos del 0.34%.

# **CAPÍTULO 7**

## 7. AUTOMATIZACIÓN DEL PROCESO ITERATIVO DE SIMULACIÓN.

La simulación como herramienta requiere de la realización de numerosos experimentos. Esto puede conducir a la necesidad de procesos automatizados para obtener información proveniente de modelos computacionales de sistemas reales.

### 7.1 Necesidad de automatizar procesos.

Con ayuda del módulo CMG de PASION se pueden obtener datos, pero, ¿qué significado tienen y cuál es la orientación que se le va a dar a la investigación, cuya herramienta es la simulación?

El programa que al ejecutarse reproduce artificialmente las relaciones de entrada y salida del sistema de interés será identificado con el nombre de modelo. Por si mismo este programa no resuelve ningún problema, pero sí da elementos necesarios para hacerlo. La resolución del problema entonces, requerirá además de metodologías adecuadas para definir el problema de manera adecuada y encontrar mejores alternativas con el menor uso de recursos.

Ambas metodologías involucran el uso repetitivo del proceso de simulación. Esto implica un tiempo de uso de computadora y cierta organización necesaria en la información que se obtenga. Estas dos tareas pueden ser llevadas a cabo de manera más eficiente mediante el uso de programas que administren y dirijan el proceso de simulación de acuerdo a estrategias bien definidas .

A continuación se explican las razones por las cuales se recomienda la automatización de procesos repetitivos que involucren la simulación por computadora:

#### 7.1.1 Tiempo de proceso.

Los procesos de simulación, en el caso del sistema de frenos ABS, se caracterizan por tener una duración prolongada (entre 15 y 30 min.). Además, en condiciones normales, se necesita de alimentar al programa con ciertos datos cada vez que se ejecuta el programa de simulación y esto consume tiempo al usuario, cuya presencia no siempre es necesaria. Esto se puede evitar utilizando programas que alimenten los datos necesarios al modelo y recopilen los resultados en archivos destinados para ello.

#### 7.1.2 Planeación de los experimentos.

La planeación anticipada de los experimentos permite lograr de manera más eficiente y clara sus objetivos . Es por esto que se recomienda plantear desde el principio:

- Objetivo del experimento.
- Condiciones de cada experimento en particular.
- Información de entrada para cada experimento.
- Manera de manejar la información generada por el programa de simulación, que puede llegar a ser bastante cuantiosa.

Se pueden programar rutinas que generen archivos de parámetros, ya sea de acuerdo a rangos que se deseen explorar, o bien, de acuerdo a algún algoritmo que continuamente evalúe y seleccione la mejor dirección de avance para llegar a conjuntos de parámetros que mejoren de manera significativa el desempeño del sistema que se estudia.

La automatización de los procesos de simulación facilita pues esta tarea de planeación y de recopilación de información de manera importante.

## 6.2 Elementos del proceso iterativo de simulación.

Este proceso requiere de la definición de varios elementos, que al reunirse, pueden ser usados de manera eficaz en la simulación sistemática de sistemas que se quiere diseñar o mejorar. Estos elementos pueden ser el experimento en sí, información relativa a los componentes del sistema que se está diseñando (en este caso las características del controlador digital y del sistema controlado, es decir el vehículo y el sistema de frenos), algoritmos de control, modelo dinámico del sistema, el tipo de información que se quiere obtener. Estos elementos son básicamente 3: el modelo, que es el programa que lleva a cabo la simulación, el archivo de parámetros, que contiene información que se puede cambiar cada vez que se ejecuta el programa-modelo y el programa que va a organizar el proceso iterativo de simulación, que puede ser para obtener una superficie de respuesta o para encontrar el conjunto de parámetros óptimo, a través de un algoritmo de optimización. A su vez, cada elemento está formado de otros: (ver diagrama 7.1.1)

Elementos que conforman el **modelo**:

- Tipo de prueba que se va a realizar. Esta prueba puede ser determinística o estocástica. Además el tipo de prueba determina las características de la misma.
- Modelo dinámico. Se refiere al modelo matemático que se usa para describir el comportamiento del sistema.
- Índice de desempeño. Funciona como parámetro para saber si un conjunto de parámetros produce un mejor desempeño del sistema de estudio que otro.
- Variables de interés. Los resultados de la simulación se harán en términos de las variables que sean consideradas como variables

de salida. Estas son las variables de interés.

- Procedimientos: Los procedimientos dependen de las condiciones de la superficie que se consideren, de si la salida del programa será en forma de trayectorias o de puntos y del modelo dinámico.

ELEMENTOS DEL PROCESO DE SIMULACIÓN ITERATIVA

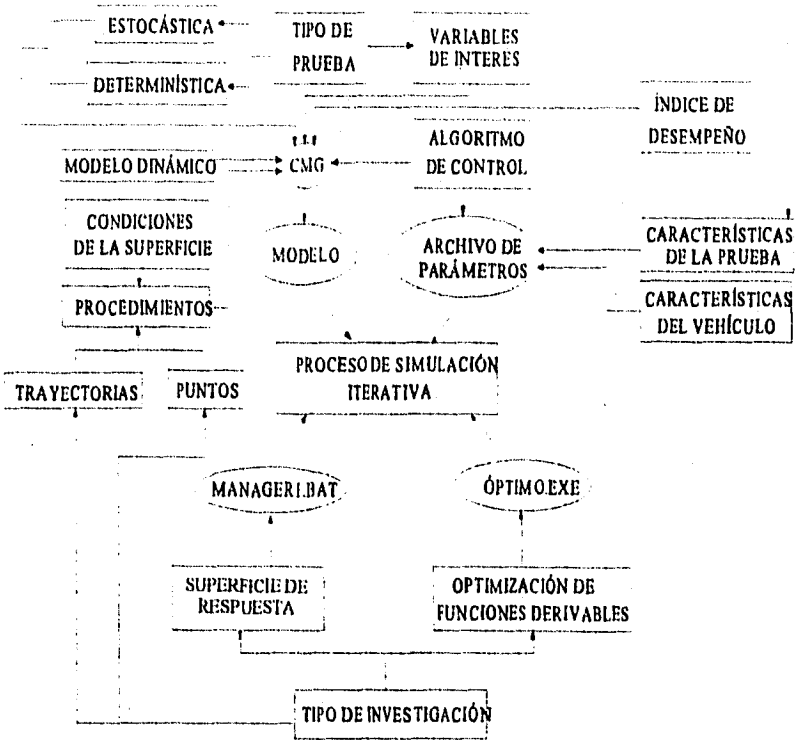


Diagrama 7.1.1.



Por otra parte el **archivo de parámetros** contiene información acerca de:

- Características de la prueba, por ejemplo, condiciones iniciales, momentos en que se aplican y se dejan de aplicar las funciones excitadoras de entrada del sistema.
- Características del vehículo (peso, radio de la llanta, ruedas con tracción y con freno).
- Parámetros del algoritmo de control.

### 7.3. OBTENCIÓN AUTOMATIZADA DE LA SUPERFICIE DE RESPUESTA.

#### 7.3.1 Trabajo previo.

De manera preliminar será necesario determinar cuáles son los parámetros de mayor influencia en el comportamiento del sistema (ver capítulo anterior). Una vez conocido esto se pueden seleccionar los dos parámetros de mayor influencia. Como se pretende lograr una descripción gráfica descriptiva del sistema en cuestión, entonces se presentará la información en forma de superficie (ver capítulo 6, inciso 6.5). Cada punto de la superficie corresponde a un punto  $z=f(x,y)$ , donde  $z$  es el índice de desempeño para evaluar el sistema y  $x$  y  $y$  son la pareja de parámetros de mayor influencia a que se hizo referencia anteriormente. Entonces serán necesarias parejas de puntos  $(x,y)$  tan numerosos como exactitud se requiera en la representación gráfica, por ejemplo, 100 parejas. Esto implica 100 conjuntos de parámetros diferentes. Se hace deseable entonces la ayuda de un programa que produzca estos conjuntos y los almacene en archivos que sean fáciles de procesar.

#### 7.3.2 Trabajo hecho por el programa.

##### 7.3.2.1 Generación de conjuntos de parámetros.

La tarea no es complicada pero es muy laboriosa. Se escoge un intervalo en el cual se va a probar un número finito de valores de cada parámetro, por ejemplo 10, ubicados de manera equidistante, para simplificar. Además, sería necesario un punto "base", a partir del cual se pueda definir el rango de variación. Entonces:

$$x_i = (x_0 - \frac{R_x}{2}) + \frac{R_x}{n} i$$

$$y_j = (y_0 - \frac{R_y}{2}) + \frac{R_y}{n} j$$

con  $i=0, 1, \dots, n-1,$   
 $j=0, 1, \dots, n-1;$

donde:

$R_x$  es la magnitud del intervalo de variación de  $x$ ,  
 $R_y$  es la magnitud del intervalo de variación de  $y$ ,  
 $n$  es el número de valores diferentes para cada parámetro,  
 $(x_0, y_0)$  es el punto de origen del intervalo.

Así pues, la misión del programa consiste en gestionar la evaluación de la función  $S(x,y)$ , para todas las parejas  $(x, y)$  donde  $S(x,y)$  es el índice de desempeño asociado al punto  $(x,y)$  y los índices  $i, j=0, 1, \dots, n-1$ . La evaluación de la función  $S(x, y)$  implica simular el sistema mediante el modelo.  $x$  y  $y$  son solamente 2 parámetros, elementos del conjunto de parámetros que forman el archivo de parámetros al que se hace referencia en el diagrama 7.1.1.

#### 7.3.2.2 Descripción del funcionamiento del programa.

Este programa se incluye en el apéndice 5. Dicho programa pide los siguientes datos:

- Número del 1<sup>er</sup> parámetro a ser variado ( $x$ ). En el archivo de parámetros cada uno de ellos tiene asociado un número. Rango de variación para el 1<sup>er</sup> parámetro.
- Número del 2<sup>o</sup> parámetro a ser variado.
- Rango de variación para el 2<sup>o</sup> parámetro ( $y$ ).
- Nombre del archivo del conjunto de parámetros original. Se refiere al punto "base" al que se hizo referencia anteriormente.

El programa genera 100 archivos. Cada uno de ellos corresponde a un conjunto de parámetros diferente. Los archivos generados llevan el nombre SUPXXYY.PAR.

XX varía de 0 al 9.  
 YY varía de 0 al 9.

De esta manera estos archivos podrán ser procesados fácilmente por un programa de procesamiento en lotes.

### 7.3.2.3 Programa de procesamiento en lotes.

Este programa aprovecha las capacidades de procesamiento en lotes que tiene el sistema operativo DOS. Lo que hace es procesar uno por uno todos los archivos de tipo SUP?????.PAR que existan en el directorio actual y copiar cada uno al archivo PARAMETROS.PAR. Después se llamará al programa C8D.EXE, por ejemplo, para efectuar una simulación con este conjunto de parámetros. El modelo escribirá siempre el resultado único en el archivo DATOS.DAT, que es un archivo de texto. El archivo PARAMETROS.PAR contendrá un conjunto de parámetros diferente cada vez, en cambio, el archivo PUNTOS.DAT irá acumulando la información de cada resultado obtenido, hasta terminar el proceso por lotes.

Junto con cada resultado asociado a cada conjunto de parámetros, se escriben los parámetros ( x, y ) que hicieron posible este resultado.

Finalmente se puede analizar y graficar esta información en una hoja de calculo comercial, ya que el formato es ASCII, separado con comas.

## 7.4 EL USO DE UN ALGORITMO DE OPTIMIZACIÓN.

Existen funciones susceptibles a ser optimizadas mediante algoritmos computacionales que encuentran de manera sistemática el conjunto de valores que hace que una función sea mínima.

### 7.4.1 Método del gradiente.

Con este algoritmo se pueden minimizar funciones  $f_0(z): \mathbb{R}^n \rightarrow \mathbb{R}^1$ , que sean continuas y diferenciables.

El algoritmo se puede implantar siguiendo los siguientes pasos:

1. Seleccionar un punto  $z_0 \in \mathbb{R}^n$ , seleccionar una  $\beta \in (0.5, 0.8)$  y hacer  $i=0$ .
2. Calcular  $h(z_i) = -\nabla f_0(z_i)$ .
3. Si  $h(z_i)=0$ , fin. Si no ir al paso 4.
4. Hacer  $\lambda=1$ .
5. Calcular

$$\Delta = f^0(z + \lambda h(z_i)) - f^0(z_i) + \frac{\lambda}{2} \|\nabla f^0(z_i)\|^2$$

6. Si  $\Delta \leq 0$  hacer  $\lambda_i = \lambda$  e ir al paso 7. Si no, hacer  $\lambda = \beta \lambda$  e ir al paso 5.
7. Hacer  $z_{i+1} = z_i + \lambda_i h(z_i)$ ; hacer  $i = i+1$  e ir al paso 2.

La desventaja de este algoritmo es que las funciones deberán ser convexas.

#### 7.4.2 Método de los gradientes conjugados.

Este algoritmo también sirve para optimizar funciones derivables dos veces y continuas y, además, tiene la ventaja de que las funciones a optimizar no son necesariamente convexas.

Consta de los siguientes pasos: (implantación del algoritmo de Polak-Ribière)

1. Seleccionar una  $z_0 \in \mathbb{R}^n$  y parámetros  $\epsilon' > 0$ ,  $\epsilon'' > 0$ ,  $\beta \in (0, 1)$ ,  $\beta' \in (0, 1)$ ,  $\beta'' \in (0, 1)$ .
2. Hacer  $g_0 = h_0 = -\nabla f^0(z_0)$ ; hacer  $e' = e'$ ,  $e'' = e''$ ,  $i = 0$ ;
3. Si  $\nabla f^0(z_0) = 0$  fin. Si no, ir al paso 2.
4. Hacer que  $z = z_i$ ,  $h = (1/\|h_i\|)h_i$ .
5. Definir  $\theta: \mathbb{R}^1 \rightarrow \mathbb{R}^1$  así:

$$\theta(x) = f^0(z+xh) - f^0(z)$$

6. Hacer  $x = 0$ .

7. Calcular:

$$\theta'(x) = \nabla f^0(z+xh) \cdot h$$

8. Si  $\theta'(x) = 0$  ir al paso 15, si no, ir al paso 9.
9. Sea  $\lambda = 1$ .
10. Calcular

$$\Delta = \theta(x - \lambda \theta'(x)) - \theta(x) + \frac{1}{2} \lambda \theta'(x)^2$$

11. Si  $\Delta \leq 0$  entonces  $x = x - \lambda \theta'(x)$  e ir al paso 12, si no,  $\lambda = \beta \lambda$  e ir al paso 10.
12. Calcular  $\nabla f^0(z+xh)$ .
13. Si  $\nabla f^0(z+xh) = 0$  hacer  $z_{i+1} = z+xh$  y fin; si no, ir al paso 14.
14. Calcular  $\theta'(x)$  de acuerdo a la ecuación del paso 7. Hacer  $\delta = \theta'(x) / \|\nabla f^0(z+xh)\|$ .
15. Si  $|\delta| \leq \epsilon'$  ir al paso 16; si no, ir al paso 9.
16. Hacer:
 
$$z_{i+1} = z+xh$$

$$y_i = \frac{(g_{i+1} - g_i)g_{i+1}}{\|g_i\|^2}$$

$$g_{i+1} = -\nabla f^0(z+xh)$$

$$h_{i+1} = g_{i+1} + y_i h_i$$

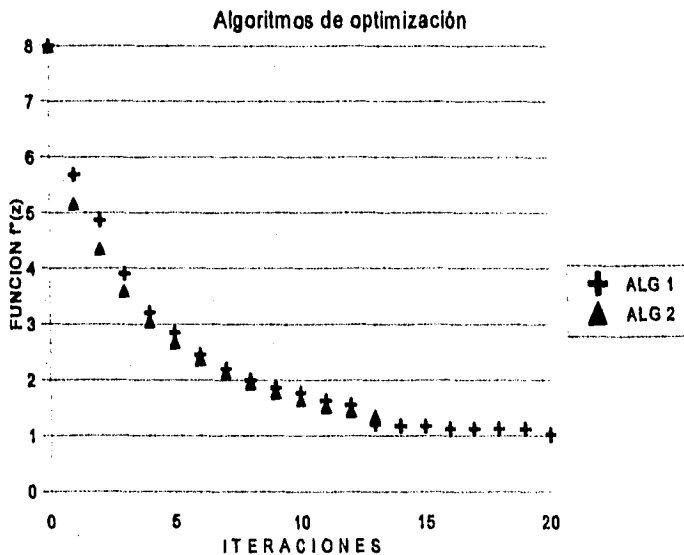
17. Hacer  $i=i+1$ .

18. Si  $g_i \cdot h_i > \|g_i\| \|h_i\|$ , ir al paso 3; si no, hacer  $e' = \beta' e'$ ,  $e'' = \beta'' e''$ ; ir al paso 3.

#### 7.4.3 Programas elaborados.

Ambos algoritmos se codificaron en el programa que se encuentra en el apéndice 4. Se hicieron pruebas para verificar el funcionamiento del algoritmo. La prueba consistió en minimizar la función  $f^0(z) = \exp[(z^1)^2 + 5(z^2)^2] + (z^2)^2 + 80(z^2)^2$ . Se supone que  $z$  es un vector con componentes  $z^1$  y  $z^2$ . Se partió del siguiente punto:  $z_0^1 = 1.32$ ,  $z_0^2 = -0.07$ . Los resultados de estas pruebas se pueden apreciar en la gráfica 7.4.1.

## Rapidez de convergencia



### 7.5 EL CASO DEL CONTROL DEL SISTEMA DE FRENOS ABS.

En este caso, se obtuvo fácilmente la superficie de respuesta (Ver capítulo 6). Sin embargo, se encontró que el índice de desempeño, que en este caso es la distancia de frenado, no es una función derivable, lo cual resulta evidente al examinar la gráfica 7.5.1. Al no ser una función derivable no es susceptible de ser optimizada mediante los algoritmos expuestos anteriormente.

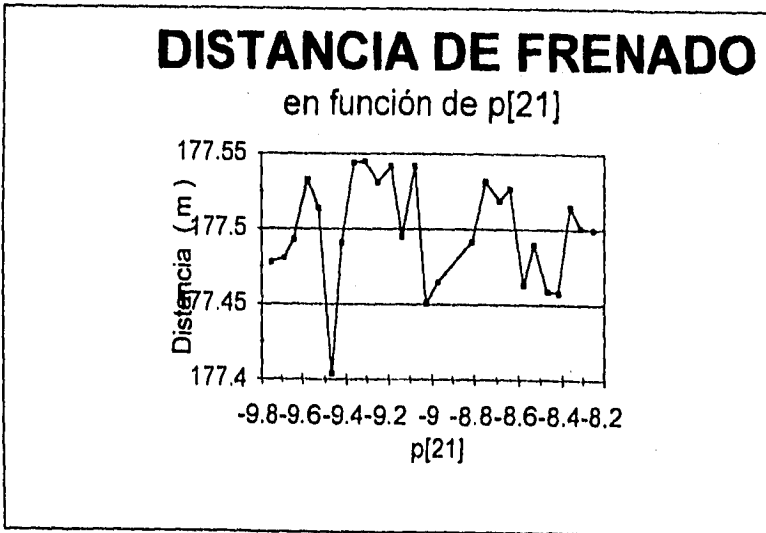


Gráfico 7.5.1.

# CONCLUSIONES

## 8. CONCLUSIONES

El módulo CMG de PASION ha resultado ser una herramienta eficaz en el análisis y diseño de sistemas electromecánicos, en este caso, de sistemas de control digitales, pues es un sistema flexible y que permite la solución de hasta 50 ecuaciones diferenciales, en modelos pequeños, o más, en modelos grandes. Además, los datos utilizados como entrada, son fácilmente cambiables, de acuerdo a las necesidades de los experimentos a realizar.

Aunque el algoritmo de control presentado en esta tesis no es un ejemplo exacto de acciones de control proporcionales, derivativas e integrales (pues éstas están definidas básicamente para sistemas continuos) si emplea de manera eficaz el concepto de control derivativo y control proporcional adaptado a sistemas discretos en el tiempo.

Otra característica de este sistema digital es que, como se trata de controlar procesos de duración finita, en realidad no se puede hablar de error estacionario. Sin embargo este error, que podría ser un indicador del buen estado del sistema de control (idealmente igual a cero), podría sustituirse (como indicador), por un valor determinado del índice de desempeño escogido (la distancia), con unas condiciones de prueba fijas y perfectamente definidas.

Acercas del sistema escogido, hablando de modelos lineales y no lineales, se pueden hacer las siguientes observaciones:

1. Este modelo está simplificando adecuadamente, de manera que solamente las variables relevantes estén incluidas en el vector de estado del sistema: la velocidad lineal y la velocidad angular de la llanta del automóvil. Aunque las demás variables (no relevantes) se necesitan para describir la dinámica del automóvil, en este estudio es suficiente con las 2 que se han escogido, pues son las influencias principales en la interacción entre vehículo y llanta ( $\lambda$ ), que es la variable a controlar.
2. La exactitud de los resultados es suficientemente buena<sup>1</sup> como para poder distinguir entre un sistema de control y otro mejor.
3. Partes esenciales del modelo utilizado, son elementos altamente no lineales, por lo que es imposible representar este sistema con ecuaciones diferenciales lineales, al menos

---

<sup>1</sup>Ver referencia (4).



en el rango de operación de interés. Esto implica renunciar a sistemas de resolución de ecuaciones diferenciales como la transformada de Laplace y a teorías de control que sólo se aplican a sistemas lineales, como el estudio de polos y ceros de las funciones de transferencia. Es entonces cuando resalta la utilidad de la simulación como alternativa adecuada.

- 4 Aunque el método de la transformada Z no es aplicable a sistemas no lineales (como el ejemplo presentado en esta tesis), sí aporta elementos importantes en el análisis cualitativo, al mostrar de manera sencilla la importancia del tiempo de muestreo como factor de influencia significativa en la estabilidad de un sistema de control digital. Una elección inadecuada del tiempo de muestreo puede incluso impedir que se consiga la estabilidad.

En el capítulo quinto se desarrolla el modelo del sistema físico de la manera más simplificada posible, manteniendo el compromiso entre simplicidad y exactitud.

En el capítulo sexto se prueba el modelo propuesto y su algoritmo de control. A través de una estrategia algo primitiva, pero efectiva, dado lo abrupto del terreno de estudio, se encuentran trayectorias de mejora interesantes:

Al usar un sistema de frenos ABS las distancias de frenado se pueden mejorar en un 12.7%, 23.3% y 38.3% para asfalto seco, mojado y con hielo, respectivamente.

Se llegó a este resultado después de múltiples ensayos, y se obtuvo finalmente un conjunto de parámetros que permiten el funcionamiento del sistema a niveles aceptables y superiores al desempeño inicial del sistema. Además se prueba la habilidad del sistema para mantener un desempeño aceptable al incluir la simulación del proceso de frenado ante perturbaciones de carácter aleatorio, como podría ser la rugosidad de la superficie o cualquier otra variación en las condiciones que influyen de manera más importante: el peso del vehículo (y su distribución) y la fricción.

En el último capítulo se habla de la necesidad de sistematizar los procesos de simulación, es decir, desarrollar una estrategia de ataque para encontrar las mejoras que se buscan con la menor inversión de recursos.

Se desarrollaron algunos programas para un manejo más sencillo de grupos de parámetros y administración de las corridas de

simulación, de manera que se consuman menos horas-hombre y se promueva una mejor planeación de los experimentos.

Se propone además utilizar algunos algoritmos de optimización y establecer un enlace vía *software*, entre la simulación por computadora digital y los programas desarrollados que utilizan este tipo de algoritmos de optimización. Sin embargo, se deberá asegurar antes la continuidad de las funciones que se pretende optimizar. Mucho de la dificultad en optimizar un sistema de control depende del algoritmo que se utilice y el sistema de que se trate. En esta ocasión me he encontrado con un sistema que por su naturaleza discontinua es imposible de optimizar, utilizando algoritmos de optimización que requieran evaluar derivadas.

Quisiera hacer énfasis en la importancia de las técnicas de simulación dentro del desarrollo tecnológico actual. No es casualidad que el surgimiento de la simulación haya revolucionado el ciclo de diseño, dentro del área de desarrollo de nuevos productos en industrias tan competitivas como la automotriz.

Finalmente quisiera reconocer la valía de la simulación por computadora como herramienta de diseño y análisis para el ingeniero en el umbral del siglo XXI. No podría admitir que siempre es fácil crear e implementar modelos que funcionen realmente, para predecir el comportamiento de un sistema físico y de manera realista. Sin embargo, la simulación por computadora y la amplia variedad de *software* disponible facilitan enormemente esta tarea.

# **BIBLIOGRAFÍA**

## BIBLIOGRAFÍA

1. WONG J. Y.  
Theory of Ground Vehicle.  
John Wiley and Sons.  
Canadá, 1978.
2. POLAK E.  
Computational Methods in Optimization  
Academic Press.  
E.U.A. 1977.
3. OGATA K.  
Ingeniería de Control Moderna.  
Prentice Hall.  
México, 1980.
4. TAN H. y TOMIZUKA M.  
"Discrete-Time Controller Design for Robust Vehicle Traction",  
Control Systems Magazine,  
Volumen 10, No. 3, pp. 107 - 113,  
E.U.A. abril, 1990.
5. ADDIEGO J.  
"The New Technology of Simulation",  
Automotive Engineering, Publicación de SAE Internacional,  
(The Society of Automotive Engineers).  
Volumen 100, No. 11, pp. 49 - 51.  
E.U.A. Noviembre de 1992.
6. RACZYNSKI, Stanislaw.  
Manual de PASION  
Universidad Panamericana.  
México, 1995.
7. RACZYNSKI Stanislaw.  
Simulación por Computadora.  
Grupo Noriega Editores.  
México, 1993.

8. COCHIN, Ira.  
Analysis and Design of Dynamic Systems.  
New Jersey. Harper & Row Publishers.  
E.U.A. 1980.
9. GORDON, Goffrey .  
System Simulation.  
New Jersey. Prentice Hall.  
E.U.A. 1978.
10. GOLDSTEIN, Larry Joel.  
Turbo Pascal.-Introducción a la programación orientada a objetos.  
Prentice Hall.  
México, 1992.
11. Microsoft Corporation.  
Microsoft MS-DOS. Guía del usuario.  
Publicado por Phoenix Technologies, Ltd.  
E.U.A. 1988.

# APÉNDICES

## APENDICE 1

Programa para simulación del sistema físico.  
DINSYS5 .PAS

```
{ $V- }
{ $M 64000,0,400000 }
PROGRAM MODEL;

USES CRT,GRAPH,PROCU;

CONST N=2; NDL=0; NLN=27;
TYPE VECTOR=ARRAY[1..N] OF REAL; s15=STRING[15]; PARAM=ARRAY[1..50] OF
REAL; TABLAS=ARRAY[1..10,1..200] OF REAL;
TABLA=ARRAY[1..1] OF REAL; VAR
K,KXX,KXXM,K00FIL,NP,RD,POZ,PDL,NSTEP:INTEGER; XX:TABLAS;
INN:ARRAY[1..27] OF REAL; XYPL:ARRAY[1..250] OF REAL; NX:STRING[N];
DLL:ARRAY[1..1,1..1] OF REAL; NUR:ARRAY[1..1] OF INTEGER; XST:VECTOR;
P:PARAM; OU:ARRAY[1..10] OF REAL; NMOU:ARRAY[1..10] OF s15;
SCM,HMIN,SMALL:REAL; XR:ARRAY[1..1,1..1] OF REAL; PISO:INTEGER;

PROCEDURE DERIV(I:INTEGER; VAR D,XST:VECTOR; TIME:REAL; VAR P:
PARAM); FORWARD;

{Put here user-defined functions, if any.}

(*$1 STT.PAS*)
(*$1 PROCE4b.PAS*)
(*$1 funcprog.pas*)

PROCEDURE INDEF;
BEGIN

1NN[2]:=MOTOR(TIME);
1NN[20]:=XST[2];
1NN[21]:=XST[1];
1NN[23]:=XST[1];
1NN[25]:=XST[2];
1NN[13]:=(P[7])*1NN[2];
1NN[22]:=LAMBDA(1NN[20],1NN[21]);
1NN[24]:=FW(1NN[23]);
1NN[26]:=FV(1NN[25]);
1NN[11]:=MU(1NN[22]);
```

122

```
INN[12]: =MU(INN[22]);
INN[13]: =INN[13]+(-1)*INN[24];
INN[17]: =(-1)*INN[26];
INN[17]: =INN[17]+(P[1])*INN[1];
INN[13]: =1NN[13]+(-p[2]*p[3])*1NN[12];
INN[1]: =FRENO(TIME ,xst[1]);
INN[13]: =1NN[13]+P[8]*1NN[1];
INN[14]: =(1-p[4])*1NN[13];
INN[18]: =(1-fp[5])*INN[17]; END;
```

```
PROCEDURE DERIV;
BEGIN
INDEF(XST,TIME);
D[1]: =1.0/(1.0)*(-0.0)*XST[1]+(1)*1NN[14];
D[2]: =1.0/(1.0)*(-0.0)*XST[2]+(1)*1NN[18];
END;
```

```
PROCEDURE OUTP;
BEGIN
OU[1]: =MOTOR(TIME);
OU[2]: =FRENO(TIME ,XST[1]);
OU[3]: =XST[2];
OU[4]: =XST[1];
OU[5]: =LAMBDA(INN[20],1NN[21]);
OU[6]: =MU(INN[22]);
OU[7]: =(P[7])*1NN[2];
OU[7]: =OU[7]+(-p[2]*p[3])*1NN[12];
OU[7]: =OU[7]+(-1)*1NN[24]; OU[7]: =OU[7]+(P[8])*1NN[1];
OU[8]: =(P[1])*INN[1]; OU[8]: =OU[8]+(-1)*1NN[26]; OU[9]: =FV(INN[25]);
OU[10]: =FW(INN[23]);
END;
```

```
BEGIN
parprog;
NMOU[1]: = 'TE'; NMOU[2]: = 'TB';
NMOU[3]: = 'V'; NMOU[4]: = 'W';
NMOU[5]: = '*LAMBDA';
NMOU[6]: = '*MU(DESL)'; NMOU[7]: = 'ENTI';
NMOU[8]: = 'ENT2';
NMOU[9]: = '*FV';
NMOU[10]: = '*FW';
RD: =0;
MAIN(10,27,1) END.
```



Archivo complementario:  
 FUNCPRG .PAS

FUNCTION MOTOR (TIME: REAL):REAL;

begin

if timec=p[1] then motor:=step(time) else motor:=0;  
 {if timec(p[1]+1) then motor:=0 else motor:=0.6\*step(time); if time=p[1] then

p[4]:=1.6832;}

end;

function signo (x:real):real;

begin

if x<0 then signo:=-1;

if x=0 then signo:=0;

if x>0 then signo:=1;

end;

FUNCTION FRENO (TIME: REAL;W:REAL):REAL;

begin

IF TIME > P[6] THEN (INN[3] ES EN ESTE MOMENTO LOS PARES  
 QUE ACTUAN SOBRE EL EJE ANTES DEL FRENO)

BEGIN

if W>0 then freno:=step(time);

if W<0.001 then

begin

IF ABS(INN[3]) < ABS(P[6]) THEN freno:=-inn[3]/p[6]

else freno:=-1 \*signo(inn[3]);

end;

END ELSE FRENO: =0;

end;

FUNCTION FRENOACT (signal: REAL):REAL;

var

temp:real;

sgn: integer;

begin

frenoact: =signal\*p[2 1];

124

```
temp: =signal*p[2 1];
if temp:=0 then sgn:=1 else sgn:=-1;
if abs(temp)>p[22] then frenoact:=300*sgn,
if temp>0 then frenoact:=0;
end;
```

```
function fv(v: real): real;
begin
  IF XST[2]:=0 THEN f f: =p[9]*v ELSE fv: =0;
end;
```

```
function fw(w: real): real;
begin
  fw:=p[1 0]*W
end;
```

```
function lambda(v,w: real): real;
var
```

```
  xx :real;
begin
  if (w/(vfp[3])) then xx:=w
  else xx:=vfp[3];
  if abs(xx)>0.005 then
    begin {2}
      lambda: =(w-vfp[3])/xx;
      if ((w<0) and (v/p[3]<0) ) then lambda:=(v/p[3]-w)/xx;
      if (signo (w)* signo (v/p[3])) < 0 then
        lambda:=signo(w);
    end {2}
  else lambda:=0;
end;
```

```
function fb(wk: real): real;
var
  xl ,b3: real;
begin
  xl: =xst[2]fp[3];
  b3: = 1 fp[4];
  {fb: =-(wk)*xl fb3;}
  fb: =-wk*xst[2]/(p[3]*p[4]);
```

```
end;
```

```
function mu(l:real):real;  
var  
a,m,p,px,x: real;
```

```
function tan(x: real): real;  
begin  
    tan:=sin(x)/cos(x);  
end;
```

```
procedure seco;  
begin  
    x:=0.41 121;  
    a:=0.9;  
    px:=1.61481;  
    p:=5*pi/4;  
    m:=tan(-10*pi/180);
```

```
end; {seco}  
procedure mojado;  
begin
```

```
    x:=0.69914;  
    a:=0.75;  
    px:=1.69943;  
    p:=10*pi/2;  
    m:=tan(-20*pi/180);  
end; {mojado}
```

```
procedure hielo;  
begin
```

```
    x:=0.28754;  
    a:=0.4;  
    px:=1.74268;  
    p:=5*pi/8;  
    m:=tan(-20*pi/180);  
end; {hielo}
```

```
begin
```

```
    case piso of  
        1: seco;  
        2: mojado;  
        3: hielo;  
    end; {case}
```

126

```
    if l>x then
        mu:=m*l+a*sin(px)-m*x;
    if ( (l>=-x) and (l<=x) ) then
        mu:=a*sin(p*l);
    if l<x then
        mu:=m*l-a*sin(px)+m*x;
end; {function mu}

function lt(desliz: real): real;
begin
    lt:=desliz-p[20];
end;

procedure parprog;
procedure pregunta;
begin
    gotoxy ( 1 0, 1 0);
    write ('==> ');
    readln(piso);
end; {pregunta}

begin
    clrscr;
    writeln('Suelo: ');
    writeln(' 1. Seco');
    writeln(' 2. Mojado');
    writeln(' 3. Hielo');
    repeat pregunta until ( piso>0) and {piso<4} );
end;
```

**APENDICE 2**

Programa de simulacion del sistema controlado.  
CONTROL6 .PAS

```

{$V-}
{$M 64000,0,400000}
PROGRAM MODEL;

IUSES CRT,GRAPH,PROCU;

CONST N=3 ND1=2; NLN=46;
TYPE VECTOR=ARRAY[1..N] OF REAL; s1 5=STRING[1 5]; PARAM=ARRAY[1 ..50] OF
REAL;
    TABLAS=ARR 4Y[1..10,1..200] OF REAL;
    TABLA=ARRAY[1..1] OF REAL;
VAR
    K,KXX,KXXM,KOOFIL,NP,RD,PO7,PDL,NSTEP:INTEGER; }OF: TABLAS;
    INN:ARRAY[1..46] OF REAL;
    XYPL:ARRAY[1..250] OF REAL;
    NX: STRING[N];
    DLL:ARRAY[1..NDL,1..500] OF REAL;
    NIJR:ARRAY[1..NDL,] OF INTEGER;
    XST:VECTDR;
    P:PARAM;
    OU:ARRAY[1..9] OF REAL;
    NMOU:ARRAY[1..9] OF s15;
    SCM,HMIN,SMALL:REAL;
    XXX001: REAL;
    XR:ARRAY[1..1,1..1] OF REAL;
    PID: BOOLEA'i';
    piso: integer;

PROCEDURE DERIV(I:INTEGER; VAR D,XST:VECTOR; TIME:REAL; VAR P:
PARAM); FORWARD;

{Put here user-defined functions, if any.}

(*$1 STT.PAS*)
(*$1 PROCE4b.PAS*)
(*$1 funcx.pas*)

PROCEDURE INDEF;
BEGIN

```

```

{TE} INN[1]:=MOTOR(TIME);
{RET1->A} 1NN[27]:=RET(p[1 7],TIME,1);
{RET2->WKN} INN[35]:=RET(p[17],TIME,2);
{W->*FW} INN[4]:=.XST[1];
xxx0001:=XST[2];
{V->*FV} INN[7]:=xxx001;
{V->*LAMBDA} INN[9]:=xxx001;
{W->*LAMBDA} INN[10]:=XST[1];
{ENTI->W} INN[3]:=(P[7])*1NN[1];
{*FW->ENTI} INN[5]:=FW(INN[4]);
{*FV->ENT2} INN[8]:=FV(1NN[7]);
xxx001:=LAMBDA(INN[9],1NN[10]);
{*LAMBDA->*MU} INN[12]:=xxx001;
{*LAMBDA->RET1} INN[15]:=xxx001;
{*LAMBDA->A} INN[26]:=xxx001;
{*LAMBDA->*LT} INN[38]:=xxx001;
{ENTI->W} INN[3]:=1NN[3]+(-1)*INN[5];
{ENT2->V} INN[6]:=(-1)*INN[8];
xxx001:=MU(INN[12]);
{*MJ->ENT2} INN[13]:=xxx001;
{*MU->ENTI} INN[14]:=xxx001;
{A->B} INN[28]:=(1)*INN[26];
xxx001:=LT(INN[38]);
{*LT->KY} INN[39]:=xxx001;
{*LT->*LTXTIME} INN[44]:=xxx001;
{ENT2->V} INN[6]:=INN[6]+(P[1])*INN[13];
{ENTI->W} 1NN[3]:=1NN[3]+(-P[2]*P[3])*1NN[14];
{A->B} INN[28]:=1NN[28]+(-1)*INN[27];
{Ky->WK} 1NN[40]:=(-P[15])*1NN[39];
{*LTXTIME->ITAE} 1NN[46]:=LTXTIME(1NN[44]);
{B->WK} INN[29]:=SMP(NSTEP,P[13],P[19],1NN[28]);
xxx001:=SMP(NSTEP,P[12],P[18],INN[40]);
{WK->RET2} INN[32]:=xxx001;
{WK->WKN} INN[34]:=xxx001;
{WK->RET2} INN[32]:=1NN[32]+(-1)*INN[29];
{WK->WKN} INN[34]:=1NN[34]+(-1)*INN[29];
{WKN->*FB} 1NN[37]:=(1)*INN[34];
{WKN->*FB} 1NN[37]:=1NN[37]+(1)*INN[35];
{FB->FRENOACT} INN[41]:=FB(INN[37]);
{*FRENOACT->ENTI} 1NN[42]:=FRENOACT(INN[41]);
{ENTI->W} INN[3]:=INN[3]+(1)*INN[42];
END;

```

```

PROCEDURE DERIV;
BEGIN
INDEF(XST,T IME);
{W} D[1]: = 1.0/(1.0)*(-(0~*XST[1])+(1/P[4])*1NN[3]);
{V} D[2]: = 1.0/(1.0)*(-(0.0)*XST[2]+(1/P[5])*1NN[6]);
{ITAE} D[3]:=1.0/(1.0)*(-(0.0)*XST[3]+1*INN[46]);
END;

```

```

PROCEDURE OUTP;
BEGIN
OU[1]: =XST[1];
OU[2]: =XST[2];
OU[3]: =LAMBDA(1NN[9],1NN[10]);
OU[4]: =MU(INN[12]);
OU[5]: =(P[7])*1NN[1];
OU[5]: =OU[5]+(-1)*INN[5];
OU[5]: =OU[5]+(-P[2]*P[3])*1NN[14];
OU[5]: =OU[5]+(1)*INN[42];
OU[6]: =LT(INN[38]);
OU[7]: =XST[3];
OU[8]: =FB(1NN[37]);
OU[9]: =FRENOP.CT(INN[41]);
END;

```

```

BEGIN
parprog;
NMOU[1]:=W';
NMOU[2]:=V';
NMOU[3]:=*LAMBDA';
NMOU[4]:=*MU';
NMOU[5]:='ENTI';
NMOU[6]:='LT';
NMOU[7]:='ITAE';
NMOU[8]:='FB';
NMOU[9]:='*FRENOACT';
NUR[1]: = 15;
NUR[2]: =32;
RD: =0;
PID: =FALSE;
MAIN(9,46,1)
END.

```

Archivo complementario:  
FUNCX.PAS

FUNCTION MOTOR (TIME: REAL):REAL;

```
begin
  if time<=p[1] then motor:=step(time) else motor:=0;
  (if time<(p[1]+1) then motor:=0 else motor:=0.6*step(time);
  if time=p[1] then p[4]:=1.683~;)
end;
```

function signo (x:real):real;

```
begin
  if x<0 then signo:=-1;
  if x=0 then signo:=0;
  if x>0 then signo:=1;
end;
```

FUNCTION FRENO ~TIME: REAL;W:REAL):REAL;

```
begin
  IF TIME > P[6] THEN
    BEGIN
      if W>=0 then freno:=1 .00
      else freno:=0;
    END
  ELSE FRENO:=0;
end;
```

FIJNCTION FRENOACT (signal: REAL):REAL;

```
var
  temp ,temp2: real;
  sgn: integer; retraso: integer; ya:boolean;
```

```
begin
  temp2:=signal*p[21];
  temp:=signal*p[21];
  retraso:=0;
  ya:=false;
  if temp>=0 then
    sgn:=1
  else sgn:=-1;
  if abs(temp)>p[22] then temp2:=p[22]*sgn;
  if temp>0 then temp2:=0;
  ya:=time>=p[6];
```



```

if ya then retraso:=1;
if xst[1]>=0.001 then
  temp2:=temp2*retraso
else
  begin
    if abs(inn[3])<abs(p[21]*p[22]) then temp2:=-inn[3]
    else temp2:=-signo(inn[3])*p[22];
    temp2:=temp2*retraso;
  end;
frenoact:=temp2;
end;

```

FUNCTION LTXTIME (error: real):real;

```

begin
  if (time>=p[6]) and (abs(inn[12]>0) then
    lxttime:=abs(error)*time else lxttime:=0;
end; {function}

```

function fv(v: real): real;

```

begin
  fv:=p[9]*v;
end;

```

function fw(w: real): real;

```

begin
  fw:=p[10]*w;
end;

```

function lambda(v,w: real): real;

```

var
  xx:real;

```

```

begin

```

```

  if (w>(v/p[3])) then xx:=w
  else xx:=v/p[3];
  if abs(xx)>0.005 then
    begin {2}
      lambda:=(w-v/p[3])/xx;
      if ((w<0) and (v/p[3]<0) ) then lambda:=(v/p[3]-w)/xx;
      if (signo(w)*signo(v/p[3])) < 0 then lambda:=signo(w);
    end {2}

```

132

```
    else lambda:=0;
end;
```

```
function fb(wk:real):real;
```

```
var
    xl ,b3: real;
```

```
begin
```

```
    xl :=xst[2]/p[3];
```

```
    b3: = 1 /p[4];
```

```
    {fb:=- (wk~*xlfb3);}
```

```
    fb: =-wk*xst[2]/(p[3]*p[4]);
```

```
end;
```

```
function mu(l:real):real;
```

```
var
    a,m,p,px x: real;
```

```
function tan(x: real): real;
```

```
begin
```

```
ten: =sin(x)/fcos(x);
```

```
end;
```

```
procedure seco;
```

```
begin
```

```
x:=0.41121;
```

```
a:=0.g;
```

```
px:=1.61481;
```

```
p: =5*pi/4;
```

```
m: =tan(- 1 0*pi/180);
```

```
end; {seco}
```

```
procedure mojado;
```

```
begin
```

```
x: =0.69914;
```

```
a: =0.75;
```

```
px: = 1.69943;
```

```
p:=1 0*pi/12;
```

```
m:=tan(-20*pi/180);
```

```
end; {mojado}
```

```

procedure hielo;
begin
  x:=0.88754;
  a:=0.4;
  px:=1.74268;
  p:=5*pi/8;
  m:=tan(-20*pi/180);
end; {hielo}

```

```

procedure azar;
var
  d:integer;
begin
  d:=random(2);
  d:=d+1;
  case d of
    1: seco;
    2: mojado;
  else seco;
  end; {case}
end; {procedure}

```

```

begin {mu}
case piso of
  1: seco;
  2: mojado;
  3: hielo;
  4: azar;
end; {case}
if l>x then mu:=m*l+a*sin(px)-m*x;
if ((l>=-x) and (l>=x) ) then mu:=a*sin(p*l);
if l<x then mu:=m*l-a*sin(px)+m*x;
end; {function mu}

```

```

function mur (l ,d: real): real;
var
  temp1 temp2: real;
  disturbio:real;
begin
  temp1:=mu(l);
  disturbio:=-0.5 + d;
  disturbio:=disturbio*p[23];

```

134

```
temp2: =temp1 +disturbio;  
mur: =temp2;  
end;
```

```
function lt(desliz: real): real;
```

```
begin  
  lt: =desliz-p[20];  
end;
```

```
procedure parprog;
```

```
procedure pregunta;  
begin  
  gotoxy (10,10);  
  write ('==> ');  
  readln(piso);  
end; (pregunta)
```

```
begin  
  clrscr;  
  writeln('Suelo: l);  
  writeln~' 1. Seco');  
  writeln(' 2. Mojado');  
  writeln(' 3 . Hielo~);  
  writeln(' 4. Variable');  
  repeat pregunta until ( piso>0) and (piso<5) );  
end;
```

**APENDICE 3.**

Programa de simulación del sistema controlado arreglado de tal forma que pueda ser llamado por otro programa externo. Los resultados los escribe en el archivo PUNTOS.DAT.

```
{ $V- }
{ $M 64000 0,400000 }
PROGRAM MODEL,

USES CRT,GRAPH,PROCU;

CONST N=4; NDL=2; NLN=37;

TYPE
  VECTOR=ARRAY[1..N] OF REAL;
  s15=STRING[15];
  PARAM=ARRAY[1..50] OF REAL;
  TABLAS=ARRAY[1..10,1..200] OF REAL;
  TABLA=ARRAY[1..1] OF REAL,
VAR
  K,KXX,KXXM,KOOFIL,NP,RD, POZ,PDL,NSTEP:INTEGER;
  XX:TABLAS;
  INN:ARRAY[1..37] OF REAL;
  XYPL:ARRAY[1..250] OF REAL;
  NX: STRING[N];
  DLL:ARRAY[1..NDL,1..500] OF REAL;
  NUR:ARRAY[1..NDL] OF INTEGER;
  XST:VECTOR; P:PARAM; OU:ARRAY[1..1] OF REAL;
  NMOU:ARRAY[1..1] OF s 15; SCM,HMIN,SMALL: REAL;
  XXX0001 : REAL;
  XR:ARRAY[1..1,1..1] OF REAL;
  PID: BOOLEAN;
  piso: Integer;
```

PROCEDURE DERIV(I:INTEGER; VAR D,XST:VECTOR; TIME:REAL; VAR P:  
PARAM); FORWARD;

{Put here user-defined functions, if any.}

(\* \$1 STT.PAS\*)

(\* \$1 PROCE4K8.PAS\*) {si se necesite un solo resultado}

(\* \$1 funcxsK.pas\*) {escrito en el archivo results}

{cambie proce4k8 por proce4k8d}

PROCEDURE INDEF; (DESPLIEGA SOLO LA DISTANCIA)

BEGIN

```

{TE} INN[2]:=MOTOR(TIME);
{PERT} 1NN[35]:=RANDOM;
{RETI ->A} INN[6]:=RET(p[7],TIME,1);
{RET2->WKN} INN[22]:=RET(p[7],TIME,2);
{W->*FW} INN[4]:=XST[I];
xxx001:=XST[2];
{V->*FV} INN[6]:=xxx001;
{V->*LAMBDA} INN[9]:=xxx001;
{W->*LAMBDA} INN[10]:=XST[I];
{V->D} INN[34]:=XST[2];
{ENTI->W} INN[3]:=(p[7])*1NN[2];
{*FW->ENTI} INN[5]:=FW(INN[4]);
{*FV->ENT2} INN[7]:=FV(1NN[6]);
xxx001:=LAMBDA(INN[9],INN[10]);
{*LAMBDA->*MU} INN[11]:=xxx001;
{*LAMBDA->RETI} INN[14]:=xxx001;
{*LAMBDA->A} 1NN[17]:=xxx001;
{*LAMBDA->*LT} INN[32]:=xxx001;
{ENTI->W} 1NN[3]:=1NN[3]+(-1)*INN[5];
{ENT2->V} INN[8]:=(-1)*INN[7];
xxx001:=MU(INN[35],INN[11]);
{*MU->ENT2} INN[12]:=xxx001;
{*MU->ENTI} INN[13]:=xxx001;
{A->B} INN[19]:=(1)*INN[17];
{*LT->KY} INN[20]:=LT(INN[32]);
{ENT2->V} INN[8]:=INN[8]+(p[1])*INN[12];
{ENTI->W} INN[3]:=INN[3]+(-p[2]-p[3])*INN[13];
{A->B} 1NN[19]:=1NN[19]+(-1)*INN[16];
{KY->WK} INN[26]:=(-p[15])*INN[20];
{B->WK} INN[27]:=SMP(NSTEP,P[13],P[19],INN[19]);
xxx001:=SMP(NSTEP,P[12],P[18],INN[26]);
{WK->RET2} INN[21]:=xxx001;

```

```

{WK->WKN} INN[23]=xxx001;
{WK->RET2} INN[21]=INN[21]+(-1/p[14])*1NN[27];
{WK->WKN} INN[23]=INN[23]+(-1/p[14])*1NN[27];
{WKN->*FB} INN[36]= ( 1 ) *INN[23];
{WKN->*FB} 1NN[36]=1NN[36]+(1)*1NN[22];
{*FB->*FRENOACT} INN[37]=FB(INN[36]);
{FRENOACT->ENTI } INN[31]=FRENOACT(INN[37]);
{ENT 1 ->W3 INN[3]=|INN[3]+(1)*INN[31];

```

END;

PROCEDURE DERIV;

BEGIN

INDEF(XST,TIME);

{W} D[1] = 1.0/(1.0)\*(-0.0)\*XST[1]+(1/p[4])\*1NN[3];

{V} D[2]= 1.0/(1.0)\*(-0.0)\*XST[2]+(1~p[5])~1NN[8];

{D} D[3]= 1.0/(1.0)\*(-0.0)\*XST[3]+( 1 ) \*INN[29];

{D} D[4]= 1.0/(1.0)\*(-0.0)\*XST[4]+( 1 ) \*INN[34];

END;

PROCEDURE OUTP;

BEGIN

OU[1] =XST[4];

END;

BEGIN

parprog;

NMOU[1] = 'D';

NUR[1]=14;

NUR[2]=21;

RD:=0;

PID:=FALSE;

MAIN(1,37, 1)

END.

Archivo complementario:  
PROC4K8 .PAS

Este archivo forma parte del ambiente PASION. Sin embargo, sufrió algunas modificaciones con el objeto de mantener fijas algunas variables y desviar la salida del programa hacia un archivo PUNTOS.DAT. Solamente se presentan las partes que incluyen las partes modificadas. Estas se presentan en cursiva.

```

Procedure rdpa(var tak:boolean);
var t: s80;
xf: char;
u:text;
k: integer;
begin
  tak: =false;
  clrscr; gotoxy( 10 , 1 0);write(
  'Read model parameters from a file ? (Y/N ->);
  {xf:='?';
  while (xf<>'Y') and (xf<>'S') and (xf<~'N') do begin
  xf:=readkey; xf:=upcase(xf) end; write(xf);XF:='Y';
  if xf='N' then exit;
  tak: =true;
  gotoxy(10,12); write('file name -> ');readln(t);
  T: = 'parametros.par' {Archivo para leer parámetros, creado por EQC}

```

```

PROCEDURE APUNTA (var detenido: boolean);
VAR
  ARCH: text;

```

```

BEGIN
  detenido:=true;
  ASSIGN (ARH,'PUNTOS.DAT');
  APPEND (ARCH);
  WRITELN (ARCH, P[21]:4:2,','P[12]:6:3,',';TIME:6:3,','xst[4]:6:3);
  CLOSE (ARCH);

```

```

END;

```

```

BEGIN
  CLOG;
  TEXTCOLOR(15);
  ILKRO: =0;
  TEXTBACKGROUND(0);
  CLRSCR;

```



```

WINDOW(10,5,80,25);
GOTOXY(1,1);
{WRITE('Final time for the simulation -> ');
READLN(TFIN);}TFIN:=15; WRITE('Time step-> ');
READLN(H);H:=0.005;
if 200.0*h>tfirl then begin
h:=tfirl/200.0;
writeln('Step too big. Redefined. h= ',h:8:5) end;
small:=3*h;
Writeln('Method of integration: 1- Runge-Kutta ord.2');
Writeln('      2- Runge-Kutta-Fehlberg ord.5');
WRITE('Option -> ');INTE:=0;
WHILE (INTE<>1) AND (INTE<>2) DO READLN(INTE);INTE:=2;
KXXM:=1; detenido:=false;
WHILE TFIN/H*KXXM<300 DO KXXM:=KXXM+1;
HMIN:=H*0.01;
K:=TRUNC(TFIN/H);
KXX:=KXXM;
KR:=1;
WHILE TRUNC(TFIN/H/KR)>250 DO KR:=KR+1;
PDL:=1;
DEFPAR; WINDOW(1,1,80,25);
CLRSCR;
WINDOW(5,3,80,25);
GOTOXY(1,1);
FOR L:=1 TO NL DO INN[L]:=0.0;
Writeln('Initial conditions');
writeln('Press Esc for zero initial conditions, any other key to read conditions. ');
{xf:=readkey; if xf=chr(27) then
for i:=1 to n do xst[i]:=0.0 else}
FOR L:=1 TO N DO BEGIN WRITE('XST[',L,'] -> ');
{READLN(XST[L]);}
CASE L OF
1:xst[L]:=107.9469;
2:xst[L]:=27.7776;
3:xst[L]:=0;
4:xst[L]:=0;
END;
XST0[L]:=XST[L] END;
window(1,1,80,25); clrscr; WINDOW (7,5,80,25);gotoxy(1,1);
TIME:=0.0;
KOQFIL:=0;
INDEF(XST,0.0); OUTP;

```

```

WRITELN;
WRITE('Numeric or Graphic output ? (N/G) -> ');
C:='N'; {WHILE (C<>'N') AND (c<>'G') DO BEGIN C: =READKEY;
C: =UPCASE(C) END;} WRITELN(C);
WRITE('Generate printer plot file ? (Y/N) -> ');
DRPR:='N'; {WHILE (DRPR<>'N') AND (DRPR<>'Y') DO BEGIN DRPR: =READKEY;
DRPR: =UPCASE(DRPR) END;} WRITELN(DRPR);
IF DRPR='Y' THEN WRITELN(
'Low resolution printer plot will be stored in file PRPLT');
NXY: =0;
IF C='G' THEN BEGIN WRITELN;
WRITELN(
'Type the node for the phase-plane');
WRITE('(must be one of the outputs. RETURN if no XY plot needed)->');
READLN(C2); UPP(C2); DELAY(500) END;
NXY: =0;
IF (C2<>'') THEN
  FOR L:=1 TO M DO IF NMOU[L]=C2 THEN NXY:=L;
FOR L:=1 TO NL DO INN[L]:=0.0;
COIL:=1;
WHILE K DIV COIL>100 DO COIL:=COIL+1;
IF REP=0 THEN BEGIN
  WRITE('No. of repetitions of the simulation run ->');
  READLN(REP) END;
SCM:=1.0;
IF C='G' THEN GRF0;
FOR MM:=1 TO NDL DO FOR L:=1 TO 500 DO DLL[MM,L]:=0.0;
INDEF(XST,0.0);
IF RD>0 THEN BEGIN
  FOR L:=1 TO NL DO FOR J:=1 TO RD+1 DO
    XR[L,J]: =0.0; POZ: = 1 END;
X: =C'; L: =0;
WHILE TRUE DO BEGIN
IF C='N' THEN BEGIN
  CLOG;
  TEXTCOLOR(15) END;
FOR KREP:=1 TO REP DO BEGIN(A)
IF C='N' THEN BEGIN
  WINDOW(1,1,80,25); CLRSCR; GOTOXY(1,1);
  WRITE(' TIME');
  FOR L:=1 TO M DO WRITE(NMOU[L]:8,' ');
  WRITELN;
  WINDOW(1,2,80,25); GOTOXY(1,1) END;

```

```

L:=0; ZNOW(X,L); LIC:=0;
STR(KREP, TY);
IF (C='G') AND (REP> 1 ) THEN
  OUTTEXT(PZX(420),PZY(10),'Trajectory no. '+TY);
IF C='N' THEN- BEGIN
  GOTOXY(60,1); WRITELN('Trajectory no. ',KREP);
  IF KREP=I THEN BEGIN
    ASSIGN(NRS,'RESULTS');
    REWRITE(NRS) END END;
if rep=1 then begin assign(zbir,'robo'); rewrite(zbir) end;
pxu:=round(lfin/h); pxn:=i;
while pxu>500 do begin pxu:=pxu div 2; pxn:=pxn*2 end;
FOR J:=1 TO NLN DO BEGIN STANPOP[J]:=INN[J]; CUAL[J]:= 1 END;
WHILE L<K DO BEGIN(B)
L:=L+1; NSTEP:=L;
INDEF(XST,TIME); OUTP,PUTRET;
IF RD>0 THEN BEGIN (C)
  POZ:=POZ+1;
  IF POZ>RD+1 THEN POZ:=1;
  FOR J:=1 TO NL DO XR[J,POZ]:=INN[J]; END(C)
IF C='N' THEN BEGIN (D)
  IF ((inn[11]=0) AND (NOT DETENIDO)) or ((TIME>14.99) AND (NOT DETENIDO))
  THEN
    APUNTA(detenido);
IF KREP=1 THEN BEGIN (WRITE (NRS,TIME:14,','));
FOR J:=1 TO M DO WRITE (NRS, OU[J]14,','));
WRITELN(NRS) END;
WRITE (TIME:8:3);
FOR J:=1 TO M DO WRITE (OU[J]:10:4,');
WRITELN END (D)
ELSE IF L MOD KR=0 THEN BEGIN
  GRFF(L DIV KR);
  IF (NXY<>0) THEN XYPL[L DIV KR]=OU[NXY] END;
if (rep=1) and ((nstep-1) mod pxn=0) then begin
  for px:= 1 to m do rss.nos[px]; werite (zbir, rss) end;
IF ((L-1) MOD COIL=0) THEN BEGIN (E)
  LIC:=LIC+1;
  IF ILKRO<LIC THEN ILKORO:=LIC;
  IF LIC<=100 THEN BEGIN
    FOR J:= 1 TO M DO IF J<=10 THEN
      XX[J,LIC]:=OU[J] END END (E)
for j:=1 to NLN do begin
  if stanpop[j]<=inn[j] then rusza[j]:=1 else rusza[j]:=-1;

```

```
    stanpop[j]:=inn[j] end;  
IF INTE=1 THEN STATE2(N,H,TIME,XST,P);  
IF INTE=2 THEN STATEF(N,H,TIME,XST,P);  
TIME:=TIME+H;  
X:='C';  
IF REP:=1 THEN IF KEYPRESSED THEN BEGIN {F}  
  IF C='N' THEN CHGPAR(X)  
  ELSE CHGPG(X);  
  IF X='R' THEN begin  
    If rep=1 then rewrite (zbir);  
    ZNOW(X,L) end;  
  END{F}  
END{B}
```

Archivo complementario: PROC48D .PAS

Este archivo forma parte del ambiente Pasion. Sin embargo, sufrió algunas modificaciones con el objeto de mantener fijas algunas variables y desviar la salida del programa hacia un archivo PUNTOS.DAT. Solamente se presentan las partes que incluyen las partes modificadas. Éstas se presentan en cursiva.

VAR

CL: INTEGER;  
HAYPAR: BOOLEAN;  
DRPR: CHAR;

Procedure rdpa(var tak:boolean);

var

t: s80;  
xf: char;  
u:text;  
k: integer;

begin

```

tak:=false;
clrscr;
gotoxy( 10 , 1 0);write(
'Read model parameters from a file ? (Y/N ->');
{xf: =?};
while (xf<>'Y') and (xf<>'S') and (xf<>'N') do begin
xf:=readkey; xf:=upcase(xf) end; write(xf);XF:='Y';
if xf='N' then exit;
tak:=true;
gotoxy(10,12); write('File name-> ');
{readln(t);}
T = parámetros.par;           {Archivo para leer parámetros, creado por EQC}
if not exist(t) then begin
write('File ' t, ' does not exist. ');
xf:=readkey; halt end;
assign(u,t); reset(u);
clrscr; writeln( 'Parameters with value zero are not shown. You can change any of the');
writeln('50 parameters stored. ');
writeln( 'File ' ,t);
for k:=1 to 50 do begin readln(u ,p[k]); if p[k]<>0.0 then writeln('P[',k ,2,']=',p[k]: 12: 5);
if wherey>22 then begin window(40,4,80,25); clrscr end end;
window( 1,1 ,80,25);
close(u); {xf:=readkey;}
clrscr; gotoxy(1 0,1 0);write(

```

```

'Change parameter(s) ? (Y/N)-> ');
xf:='N';
{while (xf<>'Y') and (xf<>'S') and (xf<>'N') do begin
  xf:=readkey; xf:=upcase~xf end;}write(xf);
if xf='N' then exit;
clrscr;
gotoxy( 1 ,3);
while true do begin
write(
'Number of the parameter to change (zero terminates)= ');
readln(k);
if k>0 then begin
  write('P[' ,k, ']=',p[k]: 12:5, ' -> ');
  readln(p[k]) and else begin svpa;exit end end end;

```

```

PROCEDURE MAIN(M,NL,REP:INTEGER);
LABEL 1, CONST E1 =640;
VAR K,KR L LL,NXY,MM,J,A,AA,BB,CC,DD,CH,px,pxn,pxu,
  LIC,ILKRO,COIL,MKR KREP,INTE:INTEGER;
YEE:ARRAY[1..10] OF BOOLEAN; TUTKA:S80; nrs:text;
OUP:ARRAY[1..10] OF INTEGER; XSTO:VECTOR; XXF:CHAR;
C: CHAR; SCAL:ARRAY[ 1 .. 2 ,1 ..10] OF REAL; C2: S 1 5;
dowyk:tbb;xma;xmi:real;
stenpnp:array[1..50] of real;
detenido: boolean;

```

```

PROCEDURE ZNOW(VAR X:CHAR; VAR L:INTEGER);
VAR J ,LL,MM: INTEGER; BEGIN
  NSTEP: =L;
  FOR J: = 1 TO N DO XST[J]: =XST0[J];
  IF RD>0 THEN BEGIN
    FOR LL: = 1 TO NL DO FOR J: = 1 TO RD+ 1 DO
      XR[LL,J]: =0 .0; POZ: = 1 END;
    FOR MM:=1 TO NDL DO FOR LL:=1 TO 500 DO DLL[MM,LL]:=0.0;
    INDEF(XST,0.0); OUTP; PDL: =1; TIME: =0.0;
    IF C='G' THEN INIGR; L:=0;
    X:='C' END;

```

```

PROCEDURE APUNTA(var detenido: boolean);
VAR ARCHR: text;
BEGIN
  writeln ('PASO POR APUNTAi);

```

```

if not detenido then
begin
defenido:=true;
ASSIGN (ARCHR, 'results');
REWRITE (ARCHR)
writeln ('Escribe en el archivo ',xsf[4]:6:3);
writeln (archr, xsf[4]:6:3);
close (archr);
end;
END;

BEGIN
CLOG;
TEXTCOLOR(15); ILKRO:=0;
TEXTBACKGROUND(0); CLRSCR; WINDOW(10,5,80,25);GOTOXY(1,1);
{WRITE ('Final time for the simulation ->');REAQDLN(TFIN);} TFIN:=15;
{WRITE (
'Time step -> ');READLN (H);} H:=0.001;
if 200.0*h>tfin then begin
h:=tfin/200.0;
writeln ('Step too big. Redefined: h='h:8:5) end;
small:=3*h;
WRITELN ('Method of integration: 1 - Runge-Kutta ord.2');
WRITELN ('          2 - Runge-Kutta-Fehlberg ord. 5');
WRITELN ('Option ->');
{INTE:=0;
WHILE (INTE<>1) AND (INTE<>2) DO READLN (INTE)} INTE:=2;
KXXM:=1; defenido:=FALSE;
WHILE TFIN/H*KXXM>300 DO KXXM:=KXXM+1;
HMIN:=H*0.01;
K:=TRUNC(TFIN/H); KXX:=KXXM; KR:=1;
WHILE TRUNC(TFIN/H/KR)>250 DO KR:=KR+1;
PDL:=1;
DEFPAR; WINDOW(1,1,80,25);
CLRSCR; WINDOW(5,3,80,25); GOTOXY(1,1);
FOR L:=1 TO NL DO INN[L]:=0.0;
WRITELN('Initial conditions ');
writeln(
'Press Esc for zero initial conditions, any other key to read conditions. ');
{xf:=readkey; if xf=chr(27) then
for l:=1 to n do xsf[l]:=0;
else}
FOR L:=1 TO N DO BEGIN

```

```

WRITE ('XST[L],L,']->');
READLN(XST[L]);
CASE L OF
1: XST[L]:=107.9469;
2: XST[L]:=27.776;
3: XST[L]:=0;
4: XST[L]:=0;
END;
XST0[L]:=XST[L] END;
window(1,1,80,25); clrscr; WINDOW(7,5,80,25); gotoxy( 1,1);
TIME:=0.0; KOOFIL:=0;
INDEF(XST,0.0); OUTP;
WRITELN;
WRITE('Numeric or Graphic output ? (N/G ->');
C:='N'; {WHILE (C<>'N') AND (C<>'G') DO BEGIN
C:=READKEY;C:=UPCASE(C) END;} WRITELN(C);
WRITE('Generate printer plot file ? (Y/N) ->');
DRPR:='N'; {WHILE (DRPR<>'N') AND (DRPR<>'Y') DO BEGIN
DRPR:=READKEY; DRPR:=UPCASE(DRPR) END;}
WRITELN(DRPR);
IF DRPR='Y' THEN WRITELN(
'Low resolution printer plot will be stored in file PRPLT');
NXY:=0;
IF C='G' THEN BEGIN WRITELN; WRITELN(
'Type the node for the phase plane');
WRITE('(must be one of the outputs. RETURN if no XY plot needed)->');
READLN(C2); UPP(C2); DELAY(500) END;
NXY:=0;
IF (C2<>'') THEN
FOR L:=1 TO M DO IF NMOU[L]=C2 THEN NXY:=L;
FOR L:=1 TO NL DO INN[L]:=0.0;
COIL:= 1;
WHILE K DIV COIL>100 DO COIL:=COIL+1;
IF REP=0 THEN BEGIN
WRITE('No. of repetitions of the simulation run ->');
READLN(REP) END;
SCM:= 1.0;
IF C='G' THEN GRFO;
FOR MM:=1 TO NDL DO FOR L:=1 TO 500 DO DLL[MM,L]:=0.0;
INDEF(XST,0.0);
IF RD>0 THEN BEGIN
FOR L:=1 TO NL DO FOR J:=1 TO RD+1 DO
XR[L,J]:=0.0; POZ:= 1 END;

```



```

X:= 'C'; L:= 0;
WHILE TRUE DO BEGIN
IF C='N' THEN BEGIN
  CLOG; TEXTCOLOR(15) END;
FOR KREP:=1 TO REP DO BEGIN(A)
IF C='N' THEN BEGIN
WINDOW(1,1,80,25); CLRSCR; GOTOXY(1,1);
WRITE(' TIME');
FOR L:=1 TO M DO WRITE(NMOU[L];8, ' '); WRITELN;
WINDOW(1,2,80,25); GOTOXY(1,1) END;
L:=0; ZNOW(X,L); LIC:=0;
STR(KREP, TY);
IF (C='G') AND (REP>1) THEN
  OUTTEXTXY(PZX(420), PZY(10), 'Trajectory no. '+TY);
IF C='N' THEN BEGIN
  GOTOXY(60,1); WRITELN('Trajectory no. ',KREP);
  IF KREP= 1 THEN BEGIN
    {ASSIGN(NRS,'RESULTS'); REWRITE(NRS)} END END;
if rep=1 then begin assign(zbir,'robo'); rewrite(zbir) end;
pxu:=round(tfir/h);
pxn:=i;
while pxu>500 do begin pxu:=pxu div 2; pxn:=pxn*2 end;
FOR J:=1 TO NLN DO BEGIN STANPOP[J]:=INN[J]; CUAL[J]:=1 END;
WHILE L<K DO BEGIN(B)
  L:=L+1; NSTEP:=L;
  INDEF(XST,TIME); OUTP; PUTRET;
  IF RD>0 THEN BEGIN(C)
    POZ:=POZ+1;
    IF POZ>RD+1 THEN POZ:=1;
    FOR J:=1 TO NL DO XR[J,POZ]:=INN[J]; END(C);
  IF C='IN' THEN BEGIN(D)
    IF ((inn[l]=0) AND (NOT DETENIDO)) or ((time>14.99) AND (NOT DETENIDO)) THEN
      APUNTA(detenido);
    IF KREP= 1 THEN BEGIN {WRITE(NRS,time:14, ', ');
      WRITELN(NRS); } END;
    WRITE (TIME:8:3);
    FOR J:=1 TO M DO WRITE(NRS,OU[J]:14, ', ');WRITE('lambda: ',inn[11]:10:4);{borrar=}
    WRITELN; END; {D}
  ELSE IF L MOD KR=0 THEN BEGIN
    GRFF(L DIV KR);
    IF (NXY<>0) THEN XYPL[L DIV KR]:=OU[NXY] END;
  if (rep=1 ) and ((nstep-1 ) mod pxn=0) then begin
    for px:=1 to m do rss.nos[px]:=ou[px]; write(zbir,rss) end;

```

```

IF ((L- 1 ) MOD COIL=0) THEN BEGIN(E)
  LIC:=LIC+1;
  IF ILKRO<LIC THEN ILKRO:=LIC;
  IF LIC<=100 THEN BEGIN
    FOR J: = 1 TO M DO IF J<= 10 THEN
      XX [J,LIC]:=OU[J] END END(E);
for j:=1 to NLN do begin
  if stanpop[j]c=inn[j] then rusza[j]:=1 else rusza[j]:=-1;
  stanpop[j]:=inn[j] end;
IF INTE=1 THEN STATE2(N,H,TIME,XST,P);
IF INTE=2 THEN STATEF(N,H,TIME,XST,P);
TIME: =TIME+H;
X:='C'
IF REP=1 THEN IF KEYPRESSED THEN BEGIN(F)
  IF C='N' THEN CHGPAR(X)
  ELSE CHGPG(X);
  IF X='R' THEN begin
    if rep=1 then rewrite(zbir);
    ZNOW(X,L) end;
  END(F);
END(B);
IF (C='N') AND (KREP=1 ) THEN BEGIN
  {close(nrs);}
  WRITELN(
  'Numeric results stored in file RESULTS') END;
DELAY(3)
IF REP>1 THEN FILARG(XX,LIC ,M ,tfin) END(A);
IF REP>1 THEN GOTO 1;
{X: =READKEY;}
IF C='G' THEN
  OUTTEXTXY(PZX(30),PZY(10),
  'Type Q to quit, R to repeat the simulation -> ')
ELSE BEGIN
  GOTOXY( 1 ,24);
  WRITE('Type Q to quit, R to repeat the simulation -> ') END;
{X:=READKEY; X:=UPCASE(X);X:='Q';IF X='Q' THEN GOTO 1; GLL; ZNOWIX,L)
END;
1:
if c='G' then begin
  FOR L:=1 TO M DO begin
    reset(zbir); px:=0;
    while not eof(zbir) do begin
      read(zbir,rss); px:=px+1;

```

```

if px<=500 then dowyk[px]:=rss.nos[l] end;
cleardevice; if px>500 then px:=500; px:=px-1;
tru(10,10,630,190,px,round(px*0.1),dowyk,xma,xmi,true,
  1,0.0,nstep*h);
outtextxy(15,10,'Trajectory plot for variable '+NMOU[l]);
xf:=readkey;
end end
if (c='G') and (rep=1.) then begin rewrite(zbir);close(zbir) end;
IF (C='G') AND (NXY<=0) THEN XYPLT(K DIV KR,C2);
clog;
TEXTCOLOR(1 5);
(WRITELN(M,' ;ILKRO); READLN;);
IF DRPR='Y' THEN BEGIN
STR(COIL'H: 8.4 ,TUTKA);
PRPL(M,ILKRO,XX,
'PASION Simulation- program CMG5    time step='+TUTKA,
'PRPLT');
WRITELN(
'Low resolution plot stored in PRPLT');
WRITELN(
'COPY or TYPE it to the printer (COPY PRPLT PRN or TYPE PRPLT>PRN)');
END END;

```

**APENDICE 4.**

Programa OPTIMO.PAS para optimizar funciones objetivo derivables. Tambien se incluyen los archivos complementarios.

PROGRAM PRINCIPAL;

USES CRT, DOS;

const d=4;

cero=0.001;

beta=0.6;

ei=0.08715574;

eii=0.99619470;

betai=0.8;

betaii=0.8;

type

  tparvar=record

    cant: integer;

    tabla: array[1..cl, 1..6] of integer;

  end;

tp=array[1..50] of real;

tvector=array[1..d] of real;

VAR

vnabla2,VNABLA,vcero,nzxh: tvector;

parvar: tparvar;

par,pfin,par1,par2,zxh: tp;

fin: boolean;

f,fd,fdl,step: real;

hh h,g:tvector;

nombre1,nomprog: string;

nit: integer;

e1,e2:real;

cerov: boolean;

a,b,tl: real;

(\*\$1 LLAMADA.PAS\*)

(\*\$1 MASTERI.PAS\*)

(\*\$1 GRADIENTE.PAS\*)

```
PROCEDURE ALG1;
```

```
var
```

```
  ct: integer;
```

```
BEGIN
```

```
  nit:=1;
```

```
  REPEAT NABLA(VNABLA,par);
```

```
  revcero(vnabla);
```

```
  if not fin then
```

```
  begin
```

```
    lambda (step);
```

```
    nuevaz;
```

```
    respar;
```

```
    writeln('Fin de la iteracion # ',nit);
```

```
    nit:=nit+1;
```

```
  end;
```

```
  UNTIL FIN;
```

```
END;
```

```
PROCEDURE ALG2;
```

```
label terminacion;
```

```
var
```

```
  c2: integer;
```

```
  nh,ng,x,xi,r1 r2,r3: real;
```

```
  t,ti: real;
```

```
  delta,l,li,delta2 gama: real;
```

```
  quince,once: boolean;
```

```
  par1,par2: tp;
```

```
  parg1 parg2,parh1 parh2,difg: tvector;
```

```
  res: text;
```

```
BEGIN;
```

```
{ALG2}
```

```
  {paso 1}
```

```
  nabra(vnabla,par);
```

```
  for c2:= 1 to d do
```

```
  begin;
```

```
    h[c2]:=-1*vnabla[c2];
```

```
    g[c2]:=h[c2];
```

```
  end;
```

```
  nit:=0;
```

```
{paso 2}
nabla=0? Si sí, fin}
revcero(vnabla);
while not fin do
begin; {while 1 fin}
```

```
{paso 3}
nh:=norma(h); {Calcula la norma del vector h y la guarda en nh}
for c2:=1 to d do
HH[c2]:= (1/nh)*h[c2];      {Normaliza el vector HH}
for c2:=1 to d do writeln('H [',c2,']=',HH[c2]);
```

```
{Paso 5}
x:=0;
```

```
{Paso 6}
xlambdah(par1 ,par2,x,hh);
zxh:=par2; {zxh = z + xh}
nabla(vnabla2 ,par2);
nzxh:=vnabla2;
tl:=tetai(x,nzxh);
{writeln('Tetai=',tl ); readln;}
quince:=false;
```

```
{Paso 7}
if tl =0 then quince:=true;
f:=evaluar(par);
fd:=evaluar(zxh);
{writeln(' Iteracion # ',nil);}
{writeln(' F(z)=' ,f);}
writeln(' F(z+xh)=' ,fd);
readln;}
while not quince do begin, {while 2}
```

```
{paso 8}
l:=1;
a:=f; {a = f(Z)}
ll:=x-l*1;
xlambdah(par1 ,par2 ,ll ,hh);
b:=evaluar(par2); {b = f~z+x-llteta'(x)}}
repeat;
once:=false;
```

```

{paso 9}
{writeln('Lambda = ',1);}
{writeln('x: ',x);
writeln('lambda: ',l);
writeln('teta'(x) ',l);}
xl: =x-l*l;
{r1:=teta(xl);} {Se haga una prueba para evaluar interpolando esta
función}
r1:=(b-a)*l1*(1-l)/(x-l1)+b;
{writeln('x-l*teta'(x) ' x-l*tetai(x,nzxh));
writeln('Tata(x-l*teta~x)',r1);}
r2:=teta(x);
{writeln('Teta(x)',r2);}
r3:=0.5*sqr(tetai(x,nzxh))*l;
{writeln('0.5*1*sqr(tetai(x,nzxh))',r3);}
{delta:=tata(x-l*tetai(x,nzxh))-tata(x) + 0.5*1*sqr(tetai(x,nzxh));3
delta: =r1 +r2+r3;
writeln('Delta= ',delta);

{Paso 10}
if dalla<=cero then
begin {if 1}
x:=x-l*tetai(x,nzxh);
{writeln('x=',x);}
once: =true;
end {if 1}
else l:=l*data;
until once;
xlambdah(par1 ,par2,x hh); {z+xh > par2}
zxh:=par2;
fd: =evaluar(zxh);

{Paso 11}
nabla(nzxh,zxh); {Nabla(z+xh) en nxzh}
for c2:=1 to d do writeln('df-d',c2,'= ',n~xh[c2]);

{Paso 12}
revcero(vnabla2);
if fin then
begin {if 2}
par:=zxh; {z+l= z + xh }
writeln ('SE ALCANZÓ UNA SOLUCIÓN');
goto terminacion;

```

end; {if 2}

{Paso 13}

```
tl := tetai(x,nzxh);
delta2 := ll/norma(nzxh);
writeln('Delta2= ',delta2);
```

{Paso 14}

```
if abs(delta2) <= e1 then quince := true;
end; {while 2}
xlambdah(par1 ,par2,x,hh); {z i+1 en par2 }
nabla(parg2 ,zxh);
for c2:=1 to d do parg2[c2]:=parg2[c2]*(-1); {g i+1 = - nabla (2+XH)}
for c2:=1 to d do difg[c2]:=parg2[c2]-g[c2];
gama:=(ppunto(parg2,difg)/sqr(norma(g)));
for c2:=1 to d do parh2[c2]:=parg2[c2]+gama*hh[c2];
writeln('Fin de la iteracion #' nit);
nit := nit+1;
h := parh2;
g := parg2;
par := par2;
respar; {Escribe los puntos obtenidos hasta ahora en archivo de texto}
```

```
if not((ppunto(g,h))=(e2*norma(g)*norma(h))) then
begin
```

```
  e1:=betai*e1;
  e2:=betaii*e2;
end;
end; {while 1 fin}
```

```
terminacion: writeln('Fin del algoritmo ');
END; {ALG2}
```

```
BEGIN; {PRINCIPAL}
INICIO;
{ALG1;}
alg2;
RESULTADO;
END. {PRINCIPAL}
```



Archivo complementario.  
MASTER 1 .PAS

```
procedure cero(var limpiar:tp);
forward;
```

```
PROCEDURE INICIO;
var
```

```
orig,delta: text;
archinfo: file of tparvar;
archz: file of tp;
ij: integer;
nv repetidos: integer;
```

```
begin
  clrscr;
  e1: =ei;
  e2: =eli;
  fin: =false;
  write('Nombre del archivo de parámetros original: ');
  readln( nombre1);
  write ('Nombre del programa para evaluar la función a optimizar: ');
  readln(nomprog);
  assign (orig, nombrel);      {Nombre del archivo en donde }
  reset (orig);              {se encuentran los parámetros}
  i:=1;
  zero(par);
  while not eof(orig) do
  begin
    readln(orig, par[i]);
    writeln('Parametro ', i, ' = ', par[i]);
    i:=i+1;
  end;
  close (orig);
  for j: =i to 50 do par[j]:=0;
  assign(archz, 'tp .dat');
  rewrite(archz);
  write (archz,par);
  close (archz);
  write('Numero de variables a optimizar: ');
  readln(nv);
  parvar.cant:=nv;
```

```

for i:=1 to d do
begin
  write ('La variable ',i,' ¿, en cuántos parámetros se guarda? ');
  readln(repetidos);
  for j:=1 to 6 do
  begin
    if (j<=repetidos)and(j<=nv) then begin
      write ('Variable # ',i,' Parametro # ',j,' ');
      readln(parvar.tabla[i][j]);
    end else parvar.tabla[i][j]:=0;
  end;
end;
for i:=1 to d do
  for j:=1 to 6 do
    writeln ('Elemento ',i,' ',j,' de parvar' parvar.tabla[i][j]);
  {readln;}
assign (archinfo,'info.dat');
rewrite(archinfo);
write (archinfo, parvar);
close (archinfo);
assign(delta,'delta.dat');
rewrite(delta);
writeln(delta,'0.01');
close (delta);
assign (delta,'optimo.dat');
rewrite (delta);
writeln(delta,'Archivo de resultados del programa optimo.pas');

```

Archivo complementario.  
LLAMADA.PAS

```
PROCEDURE LLAMA_A(ProgramName: string);
{$M $4000,0,0}
{ 1 6K stack, no heap }
var
  CmdLine: string;
  r: text;
begin
  CmdLine: =";
  {writeln('About to Exec. . .');}
  SwapVectors;
  Exec(ProgramName, CmdLine);
  SwapVectors;
  {WriteLn(' . . back from Exec~;)}
  if DosExitCode <> 0 then{ Error? }
  begin;
    {WriteLn~'Error # ', DosExitCode);}
    assign (r,'results');rewrite (r); writeln(r,'1E38');close(r);
  end else writeln('Exec successful. ',
  'Child process exit code = ',
  DosExitCode);} end;
```

Archivo complementario.  
GRADIENTE .PAS

```

function evaluar(param: tp):real;
var
  p,r: text;
  v: real;
  ij: integer;
begin
  assign (p,'parametros.par');
  rewrite (p);
  {writeln('Funcion evaluar');}
  for i:=1 to 50 do writeln(p,param[i]);
  {writeln('Evaluando con par',i,' ',param[i]);}
  close (p);
  llama_a (nomprog);
  {writeln('Lee los resultados');}
  assign(r,'results');
  reset(r);
  readln(r,v);
  close(r);
  evaluar:=v;
end;

procedure zero(var limpiar:tp);
var
  iz:integer;
begin
  for iz: = 1 to 50 do limpiar[iz]: =0;
end;

PROCEDURE NABLA(VAR VNABLA: tvector;par:tp);

VAR
  ij,k:integer;
  inc: array [1 ..d] of tp;
  delta: real;
  texto: text;
  vector,partemp: tp;
  npc: integer;

BEGIN

```

```

assign(texto,'delta.dal');
reset (texto);
readln(texto,delta1 );
delta1:=delta 1/2;
close (texto);

f:=evaluar(par);
for i:=1 to d do
begin
  zero(vector);
  {writeln('Para la variable ',i,' los parametros a cambiar son: ');}
  for j:=1 to 6 do
  begin
    npc:=parvar.tabla[i j];
    {writeln('Parametro a aumentar: ',npc);}
    if npc=0 then vector[npc]:=delta1;
  end;
  zero(partemp);
  for k:=1 to 50 do
  begin;
    writeln('parametros corrientes: ',k,'=',par[k]);
    {writeln('Delta ',k,' ',vector[k]);readln;}
    partemp[k]:=par[k]+vector[k];
  end;
  {writeln('Suma de par[',k,'] con delta de ',vector[k],'=' partemp[k]);}
  writeln;
  {writeln('~Evaluando derivada parcial para la variable: ',i);}
  {writeln('Delta xi = ',delta1 );}
  fd:=evaluar(partemp);
  vnabla[i]:=(fd-f)/delta1;
  {writeln('~f~dx',i,' = ',vnabla[i]);}
  vcero[i]:=cero;
end;
END;

```

```

PROCEDURE XLAMBDAB(VAR P1,P2: TP;l:real; h: tvector);
VAR
  I,J,K:INTEGER;
  s 1 :tvector;
  donde: integer;

BEGIN;
  zero(p1);zero(p2);

```

```

for k:=1 to 50 do p2[k]:=par[k];
for i:=1 to d do
begin
  s1 [i]:=h[i]*t;
  j:=1;
  while parvar.tabla[i]c~o do begin donde:=parvar.tabla[i,j];
  p1 [donde]:=s 1 [i];
  {writeln('lambda*h~',i,'')= ',p 1 [donde]);}
  p2[donde]:=p2[donde]+p 1 [donde];
  j:=j+1;
end;
{writeln('z + lambda*h(z',i,'')= ',p2[i]);}
end;
END;

```

PROCEDURE RESPAR;

VAR

I J:INTEGER;

s 1 :tvector;

donde: integer;

res: text;

begin

{Escribe en el archivo los resultados relevantes}

assign(res,'optimo.dat');

append(res);

for i:=1 to d do

begin

j:=1;

while parvar.tabla[i,j]<>0 do

begin

donde:=parvar.tabla[i,j];

writeln(res,'Iteracion ',nit, ' Parametro ',donde,' = ',par[donde]);

j:=j+1;

end;

end;

close (res);

end;

PROCEDURE LAMBDA(var step:real);

var1: real;

delta:text;

i, j,k: integer;

delta2: extended;

```

partemp1,partemp2: tp;
suficiente:boolean;
fc, fd,fe:extended;
begin
  l:=1 ;
  fd:=0;
  for i:=l to d do
  begin;
    h[i]:=-l*vnabla[i];
    writeln('Vector h[ ',i,'] = ',h[i]);
    fd:=fd+sqr(vnabla[i]);
  end;
  {readln;}
  suficiente:=false;
  while not suficiente do
  begin
    XLAMBAH(PARTEMP 1 ,PARTEMP2 ,L,h);
    fc:=evaluar(partemp2);
    writeln('fc: ',fc);
    fe:=fd*1/2;
    writeln('fe: ',fe);
    writeln('f: ',f);
    delta2:=fc-f+fe;
    writeln ('Delta: ',delta2);
    if delta2<=0.0001 then suficiente:=true else l:=l*beta;
  end;
  step:=l;
  writeln('Lambda = ',1);
  {readln;}
  assign(delta,'delta.dal');
  rewrite (delta);
  writeln(delta,1);
  close (delta);
end;

```

#### PROCEDURE NUEVAZ;

```

var
  i,j: integer;
  partemp1, partemp2: tp;

begin;
  writeln;

```

162

```
writeln('NUEVA Z: ');
XLAMBDAH(PARTEMP 1 ,PARTEMP2 ,STEP ,h);
FOR 1:=1 TO 50 DO BEGIN;
  PAR{1}: =PARTEMP2[i];
  WRITELN('Elemento ',i,' de z es: ',par[i]);
  {readln;}
END;
end;
```

```
procedure revcero(vnabla: tvector);
var
  cl integer;

begin
  cerov: =true;
  for cl:=1 to d do
    if vnabla[cl]<=cero then
      begin
        pfin[cl]: =par[cl];
      end
    else cerov:=false;
  fin: =cerov;
end;
```

```
function NORMA(u:tvector):real;
```

```
var
  inx: integer;
  tot: real;
  mag: real;

begin
  tot:=0;
  for inx:=1 to d do begin;
    tot: =tot+sqrt(u[inx]);
  end;
  mag: =sqrt(tot);
  norma: =mag;
end;
```



```

function teta(x: real): real;
begin
  xlambdah(par1, par2, x hh);
  fd:=evaluar (par2);
  teta:=fd-f;
end;

function ppunto(v1, v2: tvector): real;

var
  ip: integer;
  producto, total: real;

begin
  producto:=0;
  total:=0;
  for ip:=1 to d do
  begin
    producto:=v1[ip]*v2[ip];
    total:=total+producto;
  end;
  ppunto:=total;
end;

function TETA(x: real; nzxh: tvector): real;

var
  par1, p3r2, ph: tp; temp: real;
begin
  temp:=ppunto(nzxh, hh);
  teta:=temp;
end;

PROCEDURE RESULTADO;
VAR
  I: INTEGER;
BEGIN;
  FOR i:=1 to 50 do
    if par[i]<>0 then
      writeln ('Parametro ', i, ' es ', par[i]);
  writeln ('La ultima funcion evaluada fue: ', f);
  readln;
end;

```

**APENDICE 5.****Programa GENARSUP.PAS.**

Este programa sirve para generar archivos de parámetros a partir de una muestra. Básicamente se puede utilizar para generar los archivos que resulten al hacer variar un parámetro dentro de un rango, pero dicha variación será hecha de manera uniforme.

El archivo base deberá ser un archivo de texto, que contenga los valores de los parámetros, ordenados en forma ascendente. Este archivo deberá tener el nombre de ORIGEN.PAR.

El programa pide el número del parámetro a ser variado y el rango en que se va a variar para 2 parámetros. Luego se generan 100 archivos con todas las combinaciones posibles de las variaciones. Estos archivos serán utilizados por el programa manager1 de procesamiento de lotes, que hará que se ejecute una simulación por cada archivo de parámetros y los resultados los guarda en el archivo puntos.dat.

PROGRAM ARCHSUP;                    {Este programa genera los archivos de parámetros para  
generar superficies de respuesta utilizando simulación }

TYPE PARAMETRO = ARRAY [1..50] OF REAL;

VAR

PARORG, PAR: PARAMETRO;  
ARCHF, ARCHD, ARCH: TEXT;  
I, J, k, Z: INTEGER;  
NOM: STRING;  
NP1, NP2: INTEGER;                    {Numero de parametro que varían}  
RANGO 1, RANGO2: REAL;  
P1, P2: REAL;

PROCEDURE PREGUNTA;

BEGIN

  WRITE('Parametro 1 a ser variado: ');  
  READLN(NP1);  
  write ('Longitud del rango de variación: ');  
  readln(rango1);  
  WRITE('Parametro 2 a ser variado: ');  
  READLN(NP2);  
  write ('Longitud del rango de variación: ');  
  readln(rango2);

END;

PROCEDURE LEEORIG;

VAR

CT: INTEGER;  
c50: integer;

BEGIN

ct = 1 ;  
writeln(' Parametros originales: ');  
ASSIGN (ARCH, 'ORIGEN.PAR');  
RESET(ARCH);  
WHILE NOT EOF(ARCH) DO  
BEGIN  
  READLN(ARCH, PARORG[CT]);  
  writeln('Parametro ',ct,',parorg[ct]);  
  CT: =CT+ 1;  
END;  
if ct<50 then for c50:=ct to 50 do parorg[c50]: =0;  
readln;  
CLOSE (ARCH);

END;

PROCEDURE NOMBFRE(A,B: INTEGER; VAR NAME:STRING);

VAR

BASE,ST1,ST2,G: STRING[3];  
NFIN: STRING;

BEGIN

NFIN: =";  
St1: =";  
ST2: =";  
BASE: ='SUP I\_';  
NFIN: =BASE;  
IF A<10 THEN BEGIN;STR(A,G);ST1:=0+G;END ELSE STFR(A,ST1);  
IF B<10 THEN BEGIN; STR(B,G); ST2:=0+G; END ELSE STR(A,ST2);  
NFIN: =NFIN+ST1 +ST2+'.par';  
NAME: =NFIN;

END;

```
PROCEDURE VARIACIONES(K1 ,K2:INTEGER;VAR P1 :REAL;VAR P2:REAL);
```

```
VAR
```

```
  C 1 ,C2: INTEGER;
```

```
BEGIN;
```

```
  P1:=PARORG[NP 1]-RANGO1/2+(RANGO1/9)*K1;
```

```
  P2:=PARORG[NP2]-RANGO2/2+(RANGO2/9)*K2;
```

```
  writeln('Procedimiento variaciones');
```

```
  writeln('Resultados: ' ,p 1 lp2);
```

```
END;
```

```
BEGIN
```

```
  PREGUNTA;
```

```
  LEEORIG;
```

```
  FOR I:=0 TO 9 DO begin if rango2>0 then k:=9 else k:=0;
```

```
  FOR J:=0 TO k DO BEGIN NOMBRE(I ,J ,NOM);
```

```
  VARIACIONES(I ,J ,P1 ,P2);
```

```
  ASSIGN(ARCHD ,NOM);
```

```
  REWRITE(ARCHD);
```

```
  WriteIn('archivo: ',nom);
```

```
  writeln('parametro ',np 1 ,': ' ,p 1 );
```

```
  writeln('parametro ',np2 ,': ' ,p2);
```

```
  writeln('centro1: ' ,parorg[np 1]:4 ,'para el parametro #' ,np1);
```

```
  writeln('centro2: ' ,parorg[np2]:4 ,'para el parametro #' ,np2);
```

```
  FOR Z:=1 TO 50 DO
```

```
    IF Z=NP1 THEN WRITELN (ARCHD,P1)
```

```
    ELSE IF Z=NP2
```

```
      ((Z=NP2)OR(Z=NP2+1~OR(7=NP2+2)OR(7=NP2+5) ACTIVARLA EN CASO DE  
      DELTA T)
```

```
      THEN WRITELN (ARCHD,P2) ELSE
```

```
      WRITELN(ARCHD ,PARORG[Z]);
```

```
  CLOSE(ARCHD);
```

```
  END;
```

```
end;
```

```
END.
```

**APENDICE 6.**

Programa de procesamiento por lotes MANAGER1.BAT

```
insim  
for %%C in ( sup?????.par ) do call manager2 %%C
```

PROGRAMA MANAGER2.BAT

```
copy %1 parametros.par  
c8d
```

Archivo complementario:

Programa INSIM.

Creo el archivo PUNTOS.DAT y lo prepara para recibir los datos de las corridas subsecuentes.

PROGRAM INICIO;

{Este programa crea un archivo para almacenar datos generados por el programa c8d.pas}

VAR

ARCH: TEXT;

BEGIN;

ASSIGN (ARC:H 'PUNTOS.DAT');

REWRITE(ARCH);

WRITELN(arch,'p[2 1 ] ',' ','Delta t ',' ','time' ',' ','D');

CLOSE(ARCH);

END.