

ESCUELA POLITECNICA NACIONAL  
FACULTAD DE INGENIERIA ELECTRICA

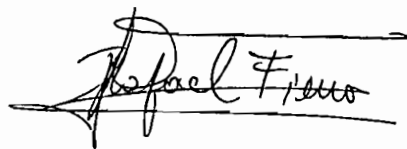
IMPLANTACION DE UN CONTROLADOR  
EXPERTO

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

ANA ELENA ESTEVEZ BARRERA

OCTUBRE DE 1992

Certifico que el presente trabajo  
ha sido realizado en su totalidad  
por la Srta. Elena Estévez B.

A handwritten signature in black ink, reading "Rafael Fierro B.", is written over two horizontal lines. The signature is stylized and cursive.

Ing. Rafael Fierro B.

DIRECTOR DE TESIS

DEDICATORIA :

A mis queridos padres y hermanas

#### AGRADECIMIENTO :

Mi más sincero agradecimiento al Ing. Rafael Fierro por su valiosa ayuda y acertados consejos para el desarrollo de este trabajo. Un reconocimiento especial a todas las personas que de una u otra manera colaboraron desinteresadamente en su realización.

# RESUMEN

El objetivo del presente trabajo fue el desarrollar un sistema experto basado en conocimiento para controlar procesos lineales. Tres algoritmos de control fueron implantados. Ellos son secuencialmente activados por el sistema de control experto, el cual usa la información de su base de conocimiento, del proceso y de las salidas de los controladores, para decidir cual acción de control conectar a la planta.

Dos diferentes técnicas fueron experimentadas y evaluadas. La primera, consiste de un sistema de control experto por conmutación, en el cual el sistema experto conecta un controlador a la planta a la vez. En la segunda técnica, el sistema experto asigna factores de ponderación a los controles. La señal final de control es una combinación de los controladores.

El programa fue desarrollado en C bajo el sistema operativo MS-DOS, el cual consiste de una serie de módulos que son llamados y supervisados por un programa principal.

Para verificar la validez y limitaciones del controlador experto, se realizaron varias simulaciones de procesos físicos comunmente encontrados en la industria.

# INDICE

PAGINA

Introducción .....	i
Objetivos y Alcance.....	iv
Justificación.....	vi

## CAPITULO I : GENERALIDADES

1.1	Inteligencia Artificial.....	1
1.1.1	Definición de Inteligencia Artificial.....	2
1.1.2	Areas de aplicación.....	3
1.1.2.1	Sistemas expertos.....	3
1.1.2.2	Procesamiento de lenguaje natural.	5
1.1.2.3	Reconocimiento del idioma y síntesis de voz.....	6
1.1.2.4	Reconocimiento de imágenes.....	6
1.1.2.5	Robótica.....	6
1.1.2.6	Educación asistida por un computador inteligente.....	7
1.1.2.7	Programación automática.....	8
1.1.2.8	Otras áreas.....	8
1.2	Sistemas expertos.....	8
1.2.1	Definición y características.....	9
1.2.2	Principales ventajas.....	10

1.2.3	Estructura de un sistema experto.....	11
1.2.3.1	Base de conocimiento.....	12
1.2.3.2	Motor de inferencia.....	13
1.2.3.3	Interfaz de usuario.....	14
1.2.4	Aplicaciones de los sistemas expertos.....	14
1.2.5	Limitaciones de los sistemas expertos.....	16
1.3	Ingeniería de Conocimiento.....	17
1.3.1	Terminología utilizada en Ingeniería de conocimiento.....	18
1.3.2	Técnicas de adquisición de conocimiento....	19
1.3.2.1	Deducción.....	19
1.3.2.2	Inducción.....	21

## CAPITULO II : SISTEMAS EXPERTOS EN CONTROL DE PROCESOS

2.1	Introducción.....	23
2.2	Técnicas de sistemas expertos para el control de procesos .....	27
2.2.1	Control basado en conocimiento.....	28
2.2.2	Control basado en reglas y lógica difusa...	30
2.3	Sistemas expertos en tiempo real.....	33
2.3.1	Ejemplos de sistemas expertos en tiempo real.....	35

## Capítulo III : IMPLANTACION DEL CONTROLADOR EXPERTO

3.1	Introducción.....	39
-----	-------------------	----

3.2	Arquitectura del controlador experto.....	40
3.3	Controlador experto.....	41
3.3.1	Control experto 1: Conmutación.....	42
3.3.2	Control experto 2: Factores de ponderación.	43
3.4	Adquisición del Conocimiento.....	44

#### Capítulo IV : ESTRUCTURA DEL PROGRAMA

4.1	Introducción.....	47
4.2	Programa principal.....	49
4.3	Base de conocimiento.....	52
4.4	Simulación del proceso.....	77
4.4.1	Selección del período de muestreo.....	79
4.4.2	Simulación de la planta.....	83
4.4.2.1	Método de integración numérica....	83
4.4.2.2	Tiempo de retardo.....	87
4.4.2.3	Actuador.....	88
4.4.3	Discretización del sistema continuo.....	89
4.5	Algoritmos de control.....	92
4.5.1	Control PID.....	93
4.5.2	Control DEADBEAT.....	100
4.5.3	Control SELF TUNING.....	105
4.5.3.1	Identificación del sistema.....	107
4.5.3.2	Diseño del controlador.....	110
4.5.3.3	Asignación de polos.....	117
4.5.3.4	Pseudocódigo del algoritmo Self Tuning.....	120
4.6	Controlador experto.....	122



5.1	Control de temperatura con retardo de transporte..	124
5.1.1	Controles.....	126
5.1.2	Resultados.....	127
5.2	Control de velocidad y posición de un servomotor..	144
5.2.1	Control de velocidad.....	146
5.2.1.1	Controles.....	148
5.2.1.2	Resultados.....	148
5.2.2	Control de posición.....	165
5.2.2.1	Controles.....	166
5.2.2.2	Resultados.....	167
5.3	Control de un sistema variante en el tiempo.....	184
5.3.1	Controles.....	185
5.3.2	Resultados.....	185

Capítulo VI : CONCLUSIONES.

6.1	Evaluación de la implantación del controlador experto.....	205
6.2	Conclusiones.....	210
6.3	Recomendaciones.....	213

BIBLIOGRAFIA.....	216
-------------------	-----

Apéndice : ABREVIATURAS Y SIMBOLOGIA.....	222
---	-----

# INTRODUCCION

Durante los últimos años se ha incrementado notablemente la atención del mundo científico sobre el área de Inteligencia Artificial, obteniéndose muchos e interesantes progresos.

Inteligencia Artificial como rama de la Informática, realiza o intenta realizar el diseño de sistemas que presentan características asociadas con la inteligencia humana, como son el entendimiento del lenguaje, aprendizaje, razonamiento, resolución de problemas, entre otros.

Existen varias áreas de estudio dentro de Inteligencia Artificial, siendo los Sistemas Expertos una de las subdisciplinas de mayor interés en la actualidad.

Sistemas Expertos son un programa computacional, diseñado para emular el proceso de razonamiento de un experto en un dominio en particular. El rango de aplicación de estos sistemas, se ha incrementado cada día como resultado de la nueva tecnología. Actualmente se han desarrollado aplicaciones en diagnóstico médico, análisis químico, exploración geológica, configuración de sistemas computacionales, educación y control de procesos.

En el área de control de procesos, existe un gran interés en

aplicar estas técnicas para diseñar sistemas de control automatizados.

Existen dos técnicas principales en el campo de control experto de procesos que se indican a continuación:

- Control Basado en Conocimiento.- El cual utiliza técnicas de Sistemas Expertos, para extender el rango de aplicación de los algoritmos de control convencionales. El sistema experto realiza una combinación inteligente de los diferentes algoritmos de control, identificación y monitoreo.
  
- Control Basado en Reglas.- El cual trata de simular las estrategias de control manual del operador del proceso, mediante un conjunto de reglas para seleccionar la acción de control correctiva en diferentes situaciones.

Estos sistemas incorporan una Base de Conocimiento, la cual constituye la parte más importante de la estructura del Controlador Experto, donde la información del proceso y controladores serán procesados.

Los conceptos básicos de Inteligencia Artificial, Sistemas Expertos e Ingeniería de Conocimiento son detallados en el Capítulo I, donde se presentan las diferentes áreas de aplicación de Inteligencia Artificial dando énfasis a los sistemas expertos.

El Capítulo II describe las características de los sistemas expertos como aplicación al control de procesos. Se presentan dos diferentes técnicas de control y a continuación se definen los aspectos de sistemas de control expertos en tiempo real.

En el Capítulo III se describe la arquitectura y técnicas utilizadas para la operación del controlador experto y la metodología empleada para la adquisición del conocimiento.

La estructura del programa con sus diferentes módulos son descritos detalladamente en el Capítulo IV.

Los controladores expertos fueron probados en varias simulaciones que son descritas en el Capítulo V. Este capítulo también contiene los parámetros de diseño para el bloque de controles y resultados de cada caso de estudio.

En el Capítulo VI se presenta una evaluación, conclusiones y recomendaciones de la implantación del controlador experto. Se comentan los resultados obtenidos en el capítulo anterior y se indican algunas futuras ampliaciones y mejoras de este trabajo.

Finalmente, se incluye un apéndice que contiene abreviaturas y la simbología utilizada.

## OBJETIVOS Y ALCANCE

El propósito de esta tesis es realizar un estudio en el cual se utilicen técnicas de Sistemas Expertos en procesos lineales, con la finalidad de obtener un controlador inteligente que mantenga la estabilidad y adecuado funcionamiento de la planta.

Las áreas que cubre este tema son: Sistemas de Control, Modelación y Simulación e Inteligencia Artificial.

En este trabajo se desarrolla un Sistema de Control Basado en Conocimiento que utiliza la información de su base de conocimiento, del sistema a controlar y de las salidas de diferentes algoritmos de control.

Se utiliza dos técnicas de operación para el Controlador Experto:

- Un Control Experto por conmutación, en el cual el sistema experto trabaja como multiplexer, conectando a la planta la mejor acción de control en cada período de muestreo.
  
- Un Control Experto por ponderación, en el cual el sistema experto trabaja como sumador. La señal de

control que actúa sobre la planta, es la combinación lineal de todos los controladores, donde a cada uno se le asigna un valor porcentual con el que contribuye a la señal de control final.

Para estructurar la base de conocimiento fue necesario un período de adquisición de información del proceso y de los controladores a implantarse. Esta información se la obtuvo de varias pruebas con los diferentes controladores, consultas a expertos en control y modelos basados en la teoría de Sistemas de Control.

Para probar y validar la estructura propuesta del Controlador Experto, se realizó un trabajo a nivel de simulación, el cual fue implantado en un computador personal bajo sistema operativo MS-DOS y utilizando el lenguaje de programación C.

Se procedió a simular algunos procesos físicos comunmente encontrados en la industria, con la finalidad de estudiar como responden los procesos lineales, con o sin retardo de transporte y sistemas variantes en el tiempo al trabajar con controladores convencionales; para posteriormente evaluar y comparar los resultados obtenidos al operar los mismos procesos con un Controlador Experto.

# JUSTIFICACION

Los Sistemas de Control han evolucionado a través de la historia de acuerdo al desarrollo de la técnica y necesidades industriales.

Parte de la teoría de Control Clásico con la cual es posible diseñar sistemas que satisfacen un conjunto de requerimientos de funcionamiento. Pero con el desarrollo industrial, los procesos y plantas son más complejos por lo que se requiere la teoría de Control Moderno para afrontar esta creciente complejidad.

Posteriormente con la difusión de computadores, se da mayor interés al Control Digital, convirtiéndose ya en práctica habitual. Pero con la necesidad de diseñar sistemas óptimos, se dirige luego la atención al estudio de los Sistemas de Control Optimo.

Actualmente el interés de estudio está enfocado al Control Inteligente, para la construcción de sistemas automáticos robustos, que lleven a cabo tareas de control, monitoreo y supervisión, obteniéndose así una mayor eficiencia y confiabilidad en procesos industriales.

La utilización de estas técnicas han permitido resolver

# CAPITULO I

## GENERALIDADES

### 1.1 INTELIGENCIA ARTIFICIAL.

Es difícil especificar una fecha exacta del inicio de lo que actualmente se conoce como Inteligencia Artificial (IA). Sin embargo, el nacimiento de IA como un campo de estudio independiente se lo considera aproximadamente a mediados de los años 50, donde las técnicas de IA nacen con el objeto de producir sistemas que sean capaces de presentar un comportamiento inteligente.

Esto significa utilizar computadores como herramientas de simulación, dotados de ciertas capacidades intelectuales lo más cercanas posibles a los seres humanos.

Conforme las técnicas y recursos computacionales se han desarrollado, la investigación en IA ha ido creciendo hasta adquirir un lugar importante entre las ramas de mayor desarrollo en la Informática. Actualmente existen muchos grupos de investigación alrededor del mundo que están trabajando en esta área.



Además, un notable impulso fue dado por el proyecto japonés para el desarrollo de computadores llamados de quinta generación, lo que despertó el interés y entusiasmo general. Este proyecto, junto a la robótica y microelectrónica, representan uno de los primeros pasos hacia una era tecnológica de alcances ilimitados.

### 1.1.1 DEFINICION DE INTELIGENCIA ARTIFICIAL.

Muchas son las definiciones que se han dado a IA, una primera definición se enuncia a continuación:

Inteligencia Artificial es el estudio de cómo hacer que los computadores realicen tareas o actividades para las cuales y en ciertos momentos, los humanos son mejores. [Ref. 2]

Esta definición implica que existen tareas para las cuales los computadores son mejores que los humanos como son: cálculos numéricos, almacenamiento de información y operaciones repetitivas.

Pero los humanos en contraposición a la operación tradicional de los computadores, realizan actividades que requieren inteligencia, como son entender un problema y utilizar el sentido común.

Con estos antecedentes se considera que el objetivo de IA es

hacer a los computadores más "inteligentes". Este concepto es la base para una segunda definición de IA:

Inteligencia Artificial es una subdisciplina de la Informática, la cual realiza el diseño de sistemas computacionales inteligentes, sistemas que muestren características asociadas con la inteligencia humana. *[Ref. 1]*

#### 1.1.2 AREAS DE APLICACION.

El campo de IA se compone de varias áreas de estudio, como se muestra en la Fig. 1.1. A continuación se dará una breve descripción de las aplicaciones más frecuentes e importantes de las técnicas de IA.

##### 1.1.2.1 SISTEMAS EXPERTOS.

Los sistemas expertos son los primeros productos en IA de auténtico impacto comercial. Son programas computacionales que intentan imitar los procesos de razonamiento y conocimiento de un experto al resolver problemas específicos en un dominio en particular. *[Ref. 5].*

Un estudio detallado de esta área se realizará más adelante.

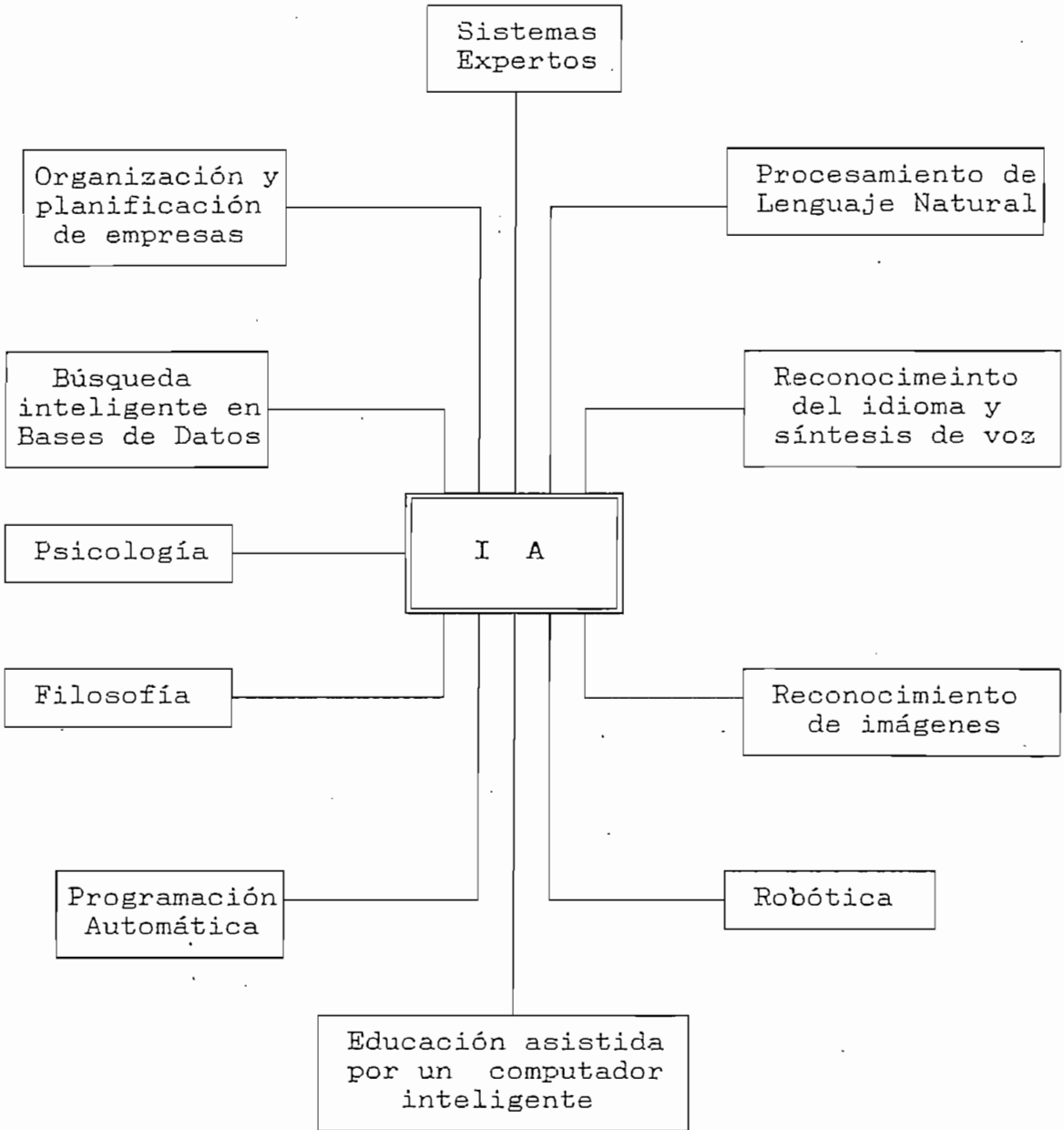


Fig. 1.1: Areas de aplicación de IA.

### 1.1.2.2 PROCESAMIENTO DE LENGUAJE NATURAL.

La utilidad de los computadores está algunas veces limitada por las dificultades de comunicación. Efectivamente es necesario utilizar un lenguaje de programación o un conjunto de comandos para poder comunicarse con el computador.

Por esta razón el objetivo del procesamiento de lenguaje natural (PLN), es hacer posible que las personas y computadores se comuniquen en lenguaje humano, así como el inglés en lugar de lenguajes de programación. [Ref. 1]

El mayor obstáculo para poder alcanzar esta meta es el tamaño y la complejidad de los lenguajes humanos.

Procesamiento de lenguaje natural está dividido en dos subcampos, que son:

- Entendimiento de lenguaje natural, el cual investiga métodos que permitan al computador comprender instrucciones dadas en un inglés ordinario, para que el computador pueda entender a las personas con mayor facilidad.
- Generación de lenguaje natural, el cual hace lo posible para que el computador genere un lenguaje en inglés ordinario, de tal forma que las personas puedan entender al computador fácilmente.

### 1.1.2.3 RECONOCIMIENTO DEL IDIOMA Y SINTESIS DE VOZ.

El objetivo de reconocimiento del idioma es permitir a los computadores entender el idioma humano, de tal forma que puedan oír voces y reconocer las palabras que se están hablando. En esta área se investiga las técnicas de lenguaje natural para simplificar los procesos de comunicación entre humanos y computadores.

### 1.1.2.4 RECONOCIMIENTO DE IMAGENES.

Es el sistema sensor más avanzado de la IA. Consiste en conectar una cámara a un computador, de tal forma que el computador reciba imágenes visuales. El objetivo en esta área es el procesar las imágenes recibidas de tal forma que el computador pueda entender exactamente lo que está enfocando.

### 1.1.2.5 ROBOTICA.

Un robot es un dispositivo electromecánico que puede ser programado para realizar tareas manuales como mover materiales, herramientas, componentes, etc. Pero no todos los robots son considerados parte de IA, como es el caso de un robot que realiza solamente acciones para el cual ha sido preprogramado.

Para controlar algunas acciones físicas del robot móvil es

necesario proporcionarle un cierto grado de inteligencia, de tal forma que pueda ejecutar tareas que son realizadas casi inconscientemente por los humanos.

A este tipo de robots se los denomina robots inteligentes, los cuales incluyen sensores tales como una cámara, que les permiten responder adecuadamente a sucesos inesperados o cambios sucedidos dentro de su ambiente de trabajo, en lugar de simplemente seguir instrucciones.

La investigación dentro de este campo ha servido para desarrollar muchas aplicaciones de IA. Los problemas de control de robots complejos forzaron a desarrollar métodos de planeamiento a altos niveles de abstracción donde se ignoran los detalles, para luego planificar a niveles más y más bajos, en los cuales los detalles adquieren importancia. [Ref. 4]

#### 1.1.2.6 EDUCACION ASISTIDA POR UN COMPUTADOR INTELIGENTE.

Educación asistida por un computador se ha utilizado por años. Actualmente las técnicas de IA han sido aplicadas para el desarrollo de sistemas de enseñanza utilizando computadores inteligentes, con el objeto de crear tutores computarizados que formen sus técnicas de enseñanza ajustándose a modelos de aprendizaje de estudiantes individuales. [Ref. 1]

#### 1.1.2.7 PROGRAMACION AUTOMATICA.

Programación es un proceso para decir al computador exactamente lo que se desea que haga. El desarrollo de un programa computacional requiere de su diseño, escritura, pruebas, depuración y evaluación.

La meta de la programación automática es crear programas especiales que actúen como herramientas inteligentes para asistir a programadores y faciliten el proceso de programación.

#### 1.1.2.8 OTRAS AREAS.

Las áreas de aplicación de IA crecen continuamente a medida que se generaliza su uso y la disponibilidad de herramientas de desarrollo de gran poder y bajo costo. Es así como también se aplica IA en: Filosofía, Psicología, búsqueda inteligente en Base de Datos, organización y planificación de empresas, entre otras.

#### 1.2 SISTEMAS EXPERTOS.

Sistemas Expertos son una importante área de investigación dentro de IA. Constituyéndose actualmente en una de las primeras tecnologías de IA para ayudar a las personas a resolver problemas complejos.

Expertos humanos en cualquier campo son de gran demanda, pero usualmente son difíciles de encontrar o peor aún, existen muchos campos donde los problemas que se presentan son mayores al número de expertos disponibles. La solución a este dilema son los sistemas expertos.

### 1.2.1 DEFINICION Y CARACTERISTICAS

Un sistema experto se lo puede definir como una estructura de programación, diseñada especialmente para representar la experiencia humana en un dominio en particular. [Ref. 1]

Se trata, por tanto, de un instrumento que intenta imitar el comportamiento humano en un determinado nivel. Un sistema experto actúa como un asistente inteligente, como lo haría un experto humano al cual no es posible acceder.

Los sistemas expertos están directamente asociados con la adquisición del conocimiento de expertos humanos, representación del conocimiento y razonamiento.

Las principales características que presentan los sistemas expertos, que los hace diferenciarse de los programas convencionales son:

- Utilizan conocimiento en lugar de datos para realizar una tarea en particular.



- Resuelven problemas por heurística y lógica en lugar de utilizar procedimientos y algoritmos. Donde heurística es una técnica de IA representada por un conjunto de reglas, que permiten encaminar la búsqueda de una solución o conclusión en la dirección correcta.
- Están formados por una base de conocimiento y un programa de control, los cuales son independientes. Esto permite actualizar la base de conocimiento sin tener que recompilar el programa de control.
- Utiliza representación simbólica del conocimiento como son: reglas, redes y estructuras.
- Son capaces de explicar cómo y por qué una conclusión en particular fue alcanzada, o qué regla se utilizó.

### 1.2.2 PRINCIPALES VENTAJAS.

Las principales ventajas de los sistemas expertos son:

- Uno de los principales atractivos de un sistema experto es su disponibilidad y conveniencia. A diferencia de un humano que tiene que descansar, comer, dormir, etc. El sistema experto está disponible las veinte y cuatro horas del día.

Además es posible crear muchos sistemas expertos, mientras que hay un número limitado de expertos humanos. Sus conocimientos pueden ser copiados, almacenados y actualizados, siendo muy difícil la pérdida de los mismos.

- El experto computarizado está siempre a pleno rendimiento, proporcionando las mejores opiniones posibles dentro de las limitaciones de sus conocimientos.
- Un sistema experto carece de personalidad, desapareciendo los problemas de falta de compatibilidad que pueden existir entre personas.
- Para crear un nuevo experto computarizado, simplemente se copia el programa de un computador a otro. Mientras que un humano necesita mucho tiempo para convertirse en un especialista en cierto campo.

### 1.2.3 ESTRUCTURA DE UN SISTEMA EXPERTO.

Todos los sistemas expertos incluyen tres elementos básicos que son:

- Una base de conocimiento.
- Un motor de inferencia y
- Una interfaz de usuario, como se ilustra en la Fig. 1.2.

[Ref. 7]

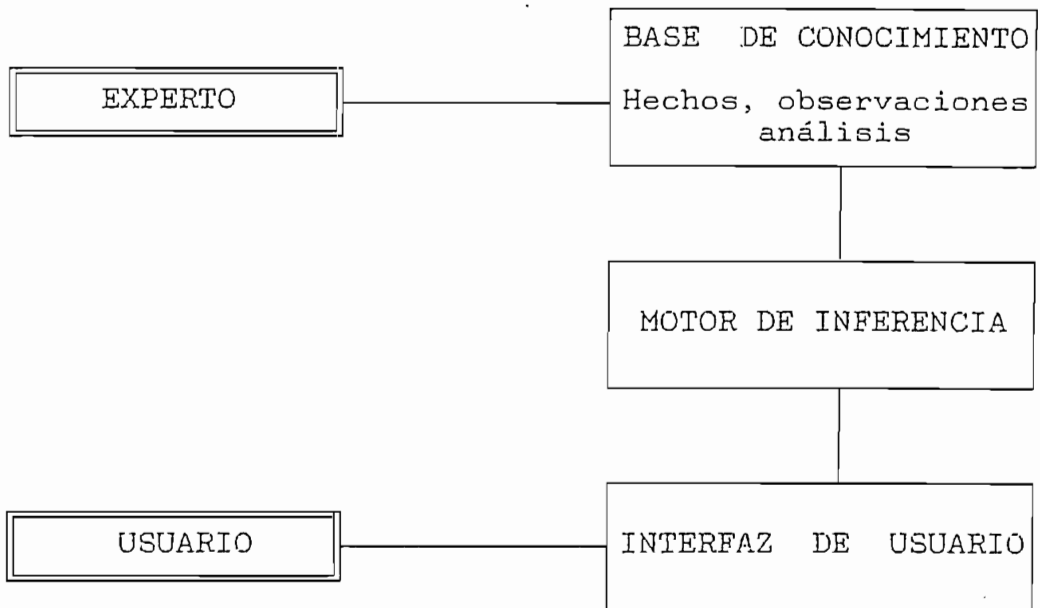


Fig. 1.2: Elementos de un sistema experto.

#### 1.2.3.1 BASE DE CONOCIMIENTO.

El elemento del sistema experto que contiene toda la información del sistema, para entenderlo y resolver problemas, se lo conoce como base de conocimiento. [Ref. 1]

La base de conocimiento incluye dos elementos esenciales:

- Hechos básicos, tales como situaciones de un problema y conceptos.
- Heurística especial o reglas que usan directamente el conocimiento para resolver problemas en un dominio en particular.

Muchas son las técnicas utilizadas para la representación del conocimiento, pero la forma más generalizada es aquella en la que se utilizan reglas de producción. [Ref. 9]

Reglas de producción se pueden definir como aquellas que poseen una estructura tipo:

SI (antecedente) ... ENTONCES (acción)

#### 1.2.3.2 MOTOR DE INFERENCIA.

Es el cerebro del sistema experto, también conocido como estructura de control o intérprete de reglas.

La misión del motor de inferencia es controlar la elección apropiada de las reglas almacenadas en la base de conocimiento. En efecto, el motor de inferencia determina qué regla debe ser invocada, la ejecuta y determina cuando se ha encontrado una solución aceptable.

Para controlar esta elección existen diversas técnicas para construir el motor de inferencia lo que diferencian unos sistemas expertos de otros, tales como:

- Método de encadenamiento hacia adelante, en el cual el motor de inferencia inicia con alguna información y luego intenta encontrar una conclusión que encaje con dicha

información. [Ref. 6]

- Método de encadenamiento hacia atrás, es el contrario al método anterior. En este caso el motor de inferencia comienza con una hipótesis y pide información para confirmarla o negarla. [Ref. 6]

### 1.2.3.3 INTERFAZ DE USUARIO.

Es la encargada de intercambiar información entre el usuario y el sistema experto en un adecuado ambiente de trabajo. Esto puede llevarse a cabo ya sea a través de menús, utilizando una interfaz de lenguaje natural o una interfaz de tipo gráfico, lo cual dependerá de que tan sofisticado sea el sistema experto.

### 1.2.4 APLICACIONES DE LOS SISTEMAS EXPERTOS.

Los sistemas expertos expanden el uso de computadores, para aplicaciones donde no se los utilizaba de forma eficiente con programas convencionales.

Varias aplicaciones de sistemas expertos han aparecido en áreas como: [Ref. 8]

- Medicina: Utilizados para diagnóstico, medicina interna, enfermedades pulmonares, de la sangre, ej. MYCIN, CASNET.

- Electrónica y Computación: Diagnóstico de equipo electrónico y electromecánico, desarrollo de software, detección de fallas en equipos de computación y comunicaciones, manufactura de computadores, diseño VLSI, ej. CRIB, DART.
- Geología: Exploración de minerales, análisis de campos petroleros, ej. PROSPECTOR, IKBM.
- Química: Identificación de estructuras en componentes químicos, cristalografía, ej. DENDRAL.
- Matemáticas: Análisis numérico, probabilidades, ej. MACSYMA.
- Administración de Empresas: Consejero financiero, evaluación, organización y planificación, ej. TAXADVISOR, AUDITOR.
- Educación y entrenamiento: Enseñanza de actividades muy especializadas, ej. SCHOLAR, SOPHIE.
- Aplicaciones militares: Control de tráfico aéreo, optimización del funcionamiento de armas y simuladores, ej. AIRPLAN, KNOBS.
- Control de procesos: Sus aplicaciones y ejemplos se detallarán más adelante.

### 1.2.5 LIMITACIONES DE LOS SISTEMAS EXPERTOS.

Las metodologías disponibles de sistemas expertos son efectivas sólo para aplicaciones simples. Para aplicaciones de relativa complejidad, el código desarrollado es generalmente difícil de entender, depurar y ampliar. Esto se debe a que todavía este campo no ha sido investigado a profundidad. Algunos factores y problemas que limitan la difusión comercial de los sistemas expertos son: *[Ref. 5]*

- El conocimiento no siempre es disponible con facilidad.
- La experiencia es difícil de extraer y representar en un formato conveniente.
- Las técnicas empleadas por cada experto para afrontar un problema en particular pueden ser diferentes, pero aún correctas.
- Los sistemas expertos trabajan bien en dominios limitados.
- El vocabulario que los expertos usan para expresar los hechos y relaciones es frecuentemente limitado o difícilmente comprendido por otros.
- En muchas ocasiones es necesario la ayuda de un ingeniero de conocimiento, lo cual encarece el costo de implantación de un sistema experto.

### 1.3 INGENIERIA DE CONOCIMIENTO.

Ingeniería de conocimiento es el proceso de construir sistemas expertos. Estos sistemas son productos computacionales diseñados para resolver problemas de diferente tipo utilizando técnicas basadas en conocimiento, el cual se lo representa en forma explícita. [Ref. 11]

El proceso de construir un sistema experto consiste de dos actividades:

- Adquisición de conocimiento.
- Representación del conocimiento e implantación del sistema.

Adquisición del conocimiento es una actividad que involucra entrevistas y observaciones a expertos humanos, complementándola con la utilización de libros, documentos, sensores o archivos computacionales.

El conocimiento puede ser específico sobre un problema en particular, procedimientos para resolverlo o conocimientos generales. Esta información es luego transferida a una base de conocimiento en un formato que pueda ser entendido por el computador, es decir se implanta el sistema.

La tarea de adquisición no es fácil, considerando los aspectos



subjetivos del ser humano como: preferencias a usar ciertos métodos, exceso de confianza, simplificación, etc. Por lo tanto el prototipo del sistema experto debe ser continuamente probado hasta obtener un funcionamiento adecuado, donde la base de conocimiento haya sido filtrada de toda información subjetiva.

### 1.3.1 TERMINOLOGIA UTILIZADA EN INGENIERIA DE CONOCIMIENTO.

Existen algunos términos usados para describir el proceso de diseño de los sistemas expertos, los cuales no están bien definidos y suelen superponerse.

Estos términos son: adquisición de conocimiento, implantación del sistema, mecanismo de inferencia, etc. La relación entre estos elementos se puede observar en la Fig. 1.3.

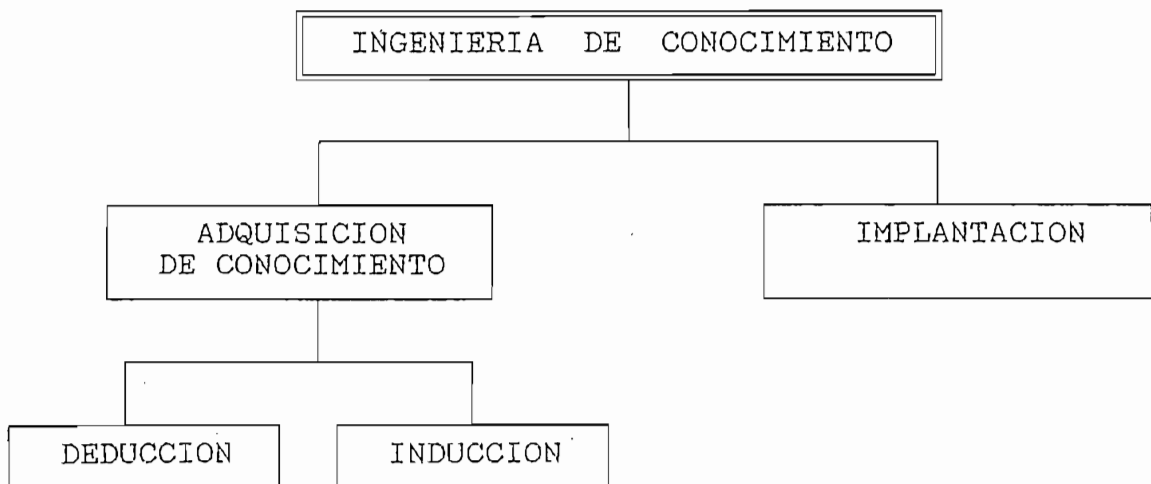


Fig. 1.3: Relación entre los elementos de Ingeniería de Conocimiento.

Como ya se definió anteriormente, adquisición de conocimiento es la transferencia de la experiencia para resolver problemas desde alguna fuente de conocimiento a un programa. [Ref. 11]

Esta definición cubre todo un proceso para el desarrollo de sistemas expertos, que incluye identificación del problema, su conceptualización, estructuración, implantación, pruebas y validación del prototipo.

Una representación esquemática de este proceso se muestra en la Fig. 1.4. [Ref. 12]

### 1.3.2 TECNICAS DE ADQUISICION DE CONOCIMIENTO.

Las técnicas utilizadas para adquisición de conocimiento se las puede clasificar en dos categorías: Deducción e Inducción.

#### 1.3.2.1 DEDUCCION.

Actualmente existe algunas técnicas para deducir el conocimiento. Estas usualmente involucran la colección de información de un experto en cierto dominio, a través de reportes escritos, entrevistas, cuestionarios y técnicas de observación. [Ref. 12]

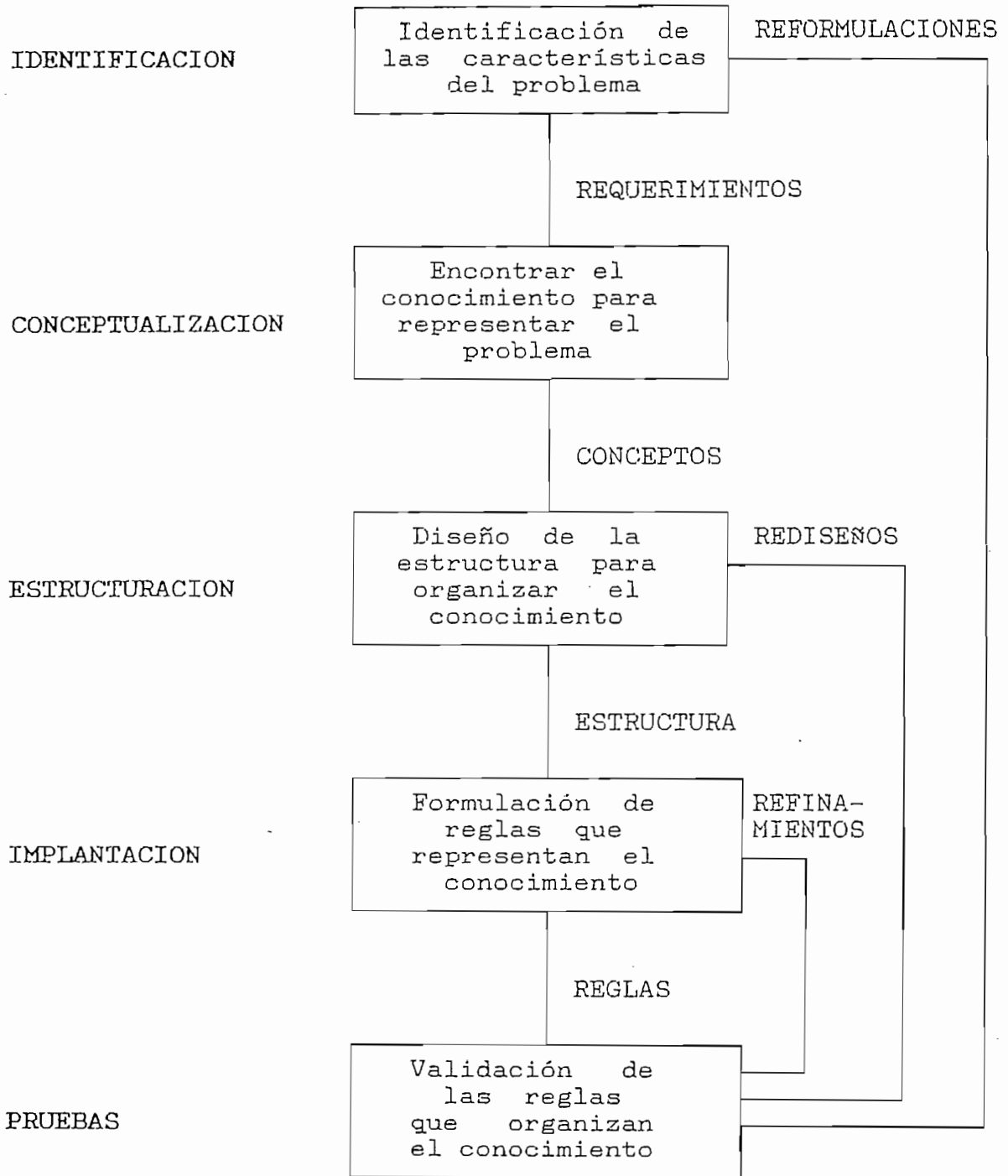


Fig. 1.4: Etapas en el desarrollo de sistemas expertos.

Las entrevistas son las técnicas comúnmente utilizadas y sirven para tener una idea general sobre el campo en estudio. El resultado de éstas, generan una gran cantidad de datos que necesitan ser analizados y conceptualizados por el ingeniero de conocimiento.

Como un sustituto o complemento a las entrevistas se puede utilizar cuestionarios previamente preparados, se recomienda utilizar técnicas definidas en Psicología y Sociología para obtener buenos resultados.

Observaciones al experto, son otra técnica de adquirir el conocimiento. Todas las acciones y actividades del experto son observadas por el ingeniero de conocimiento quien en algunos casos puede hacer grabaciones al trabajo del experto.

La combinación de todas estas técnicas permitirán optimizar el proceso de adquisición de conocimiento.

#### 1.3.2.2 INDUCCION.

Esta técnica involucra la generación de árboles de decisión y de conjuntos de reglas que representan el conocimiento, la que ha sido empleada exitosamente en muchos sistemas expertos actuales. [Ref. 12]

Es común que los expertos tengan dificultad en explicar los

procedimientos empleados por ellos para llegar a tomar decisiones, ya que generalmente hacen simplificaciones y utilizan su experiencia, en ciertos casos, aún sin entender los mecanismos de su propio razonamiento.

Las técnicas inductivas permite a los expertos que usualmente sugieran ejemplos de su experiencia, los cuales son luego analizados por un algoritmo inductivo (uno de los más populares es el algoritmos ID3 de Quinlan, 1979), y un conjunto de reglas son generadas automaticamente de estos ejemplos. [Ref. 10]

Los reglas obtenidas dependerán de los casos escogidos y del algoritmo utilizado, por lo que no hay garantía que las reglas inducidas correspondan a un conocimiento válido. Por esta razón es necesario hacer refinamientos bajo la supervisión del experto, hasta obtener una base de conocimiento aceptable.

La técnica inductiva ha sido utilizada en procesos donde existen gran cantidad de datos pero las reglas de operación no son conocidas, así por ejemplo en predicción de clima, comportamiento de nuevos compuestos químicos, integración simbólica y diseño de separadores petróleo-gas.

## CAPITULO II

# SISTEMAS EXPERTOS EN CONTROL DE PROCESOS

### 2.1 INTRODUCCION.

Aunque la teoría de Sistemas de Control proporciona soluciones para la mayoría de problemas prácticos en el área de control de procesos, sin embargo, existen situaciones que no pueden ser resueltas aplicando unicamente esta teoría. Estos problemas son encontrados especialmente en plantas muy grandes y complejas, que están sujetas a fallas de sus componentes.

Actualmente las técnicas de sistemas expertos son utilizadas en el dominio de control de procesos en dos formas diferentes: *[Ref. 12]*

1. Asistiendo a un ingeniero de control en el diseño de un controlador convencional.
2. Supervisando y controlando el proceso que previamente

era controlado por un operador.

En la primera alternativa, el ingeniero usa al sistema experto como una guía de diseño y ajuste de los parámetros del controlador. Una vez que éste es diseñado y probado, el sistema experto ha completado su tarea; manteniendo una relación indirecta con el proceso o planta.

Este tipo de sistemas expertos son exitosamente empleados en diseño por las siguientes razones:

- Ayudan al diseñador a manejar gran cantidad de restricciones y factores de incertidumbre involucrados en el proceso de diseño.
- Dan un adecuado soporte para usuarios no expertos en el dominio de sistemas de control.
- Pueden dar explicaciones, basadas en conceptos teóricos, de por qué realizó cierto procedimiento.

El sistema experto integra una base de conocimiento, base de conceptos y una herramienta de funciones o algoritmos numéricos para manipular el modelo de la planta en estudio durante la fase de diseño. Todo el proceso es controlado por el usuario quien sigue los procedimientos y observa las explicaciones para completar el diseño.

La estructura de un sistema experto para consulta en control de procesos se indica en la Fig. 2.1.

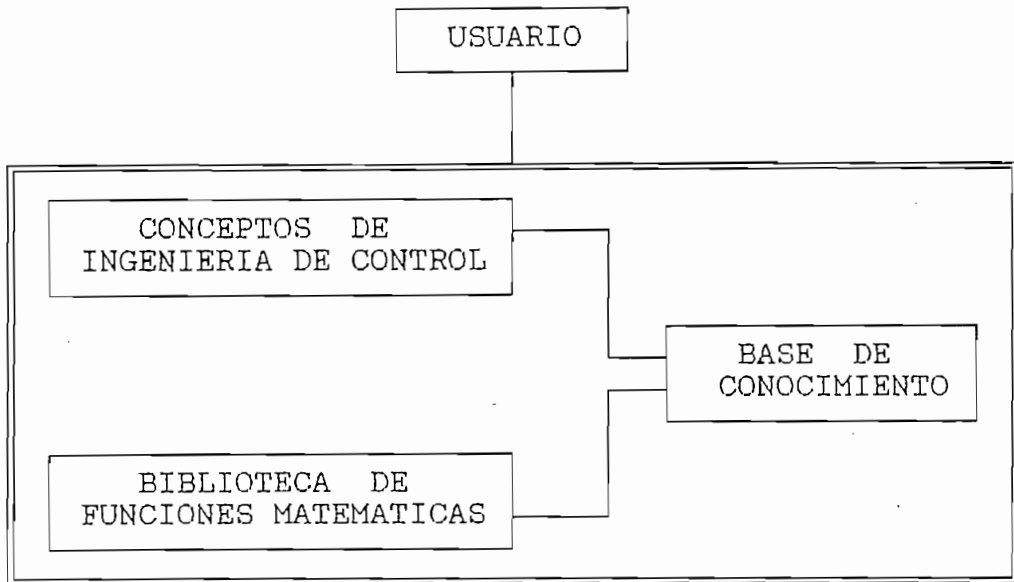


Fig. 2.1: Sistema Experto de consulta en control de procesos.

En la segunda alternativa el sistema experto trabaja directamente con el proceso, lo que involucra una mayor interacción entre el sistema y la planta.

Es necesario considerar que para incrementar la confiabilidad de un proceso industrial y asegurar su adecuado funcionamiento, la presencia de un operador humano juega un papel muy importante, particularmente cuando el proceso está sujeto a contingencias, tales como fallas, variaciones en las condiciones de operación, etc. Es por esta razón la



necesidad de diseñar un controlador que sea capaz de trabajar tan bien como un operador humano experimentado.

Un controlador podrá ser competitivo con un operador humano si el diseño del controlador incorpora algunas de las habilidades del experto humano en resolver problemas. Este tipo de controlador es llamado un controlador experto (CE). Así el objetivo del CE es mantener la estabilidad del sistema y su correcta operación no solamente en régimen normal, sino también en presencia de fallas y perturbaciones.

En este caso existe algunas consideraciones que el CE debe tener presente:

- Los procesos operan continuamente.
- Pueden haber muchos expertos trabajando en diferentes niveles.
- Variaciones de parámetros del proceso se propagan y amplifican a través del sistema.
- El proceso puede cambiar a lo largo del tiempo.
- La estructura de la planta puede ser muy compleja.
- Hay muchos sensores que obtienen datos de la planta.

- Los sensores y componentes del sistema son susceptibles a errores y fallas.
- Diferentes niveles de descripción de la planta pueden ser apropiados para diferentes tareas de control y supervisión.

La estructura del CE se presenta en la Fig. 2.2. [Ref. 18]

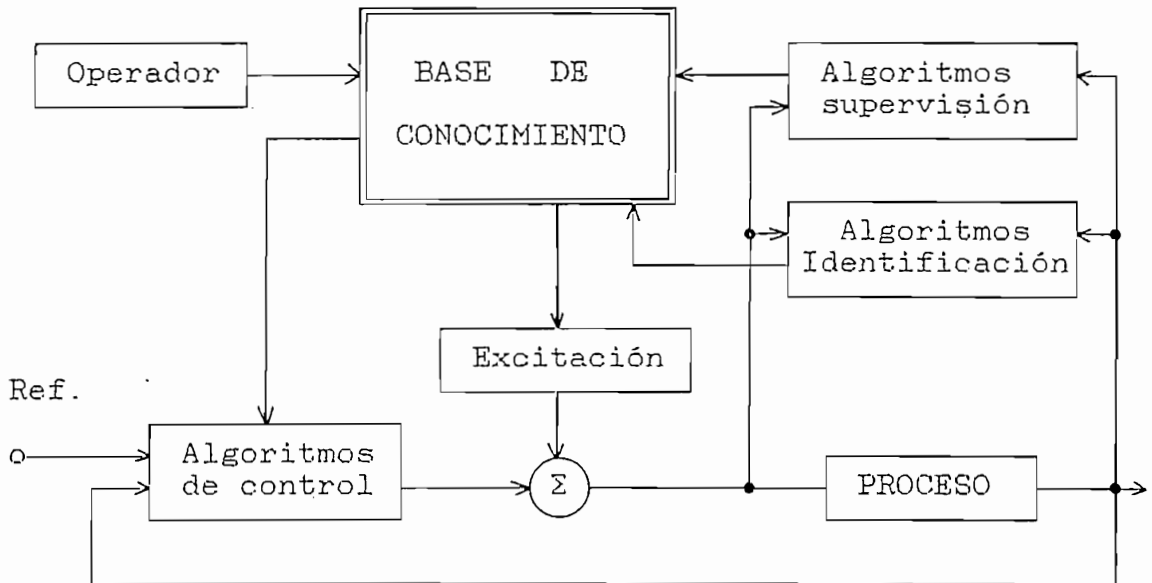


Fig. 2.2: Diagrama de bloques de un controlador experto.

## 2.2 TECNICAS DE SISTEMAS EXPERTOS PARA EL CONTROL DE PROCESOS.

Existen dos técnicas que se pueden utilizar para el control de procesos en lazo cerrado:

Control Basado en Conocimiento (CBC) que intenta obtener un ajuste óptimo de parámetros en un controlador adaptivo convencional.

Control Basado en Reglas y Lógica Difusa (CBR) que simula y automatiza la habilidad de un operador experimentado en controlar una planta manualmente.

### 2.2.1 CONTROL BASADO EN CONOCIMIENTO.

Utiliza las técnicas de sistemas expertos para extender el rango de aplicación de los algoritmos de control convencional. El sistema experto constituye una combinación inteligente de los diferentes algoritmos de control, identificación y monitoreo. [Ref. 19]

En la Fig. 2.3 se presenta la estructura y relación de los elementos que forman la arquitectura del Control Basado en Conocimiento.

El Control Basado en Conocimiento consiste de:

- Base de conocimiento.
- Motor de inferencia
- Información del preprocesador
- Varios algoritmos de control.

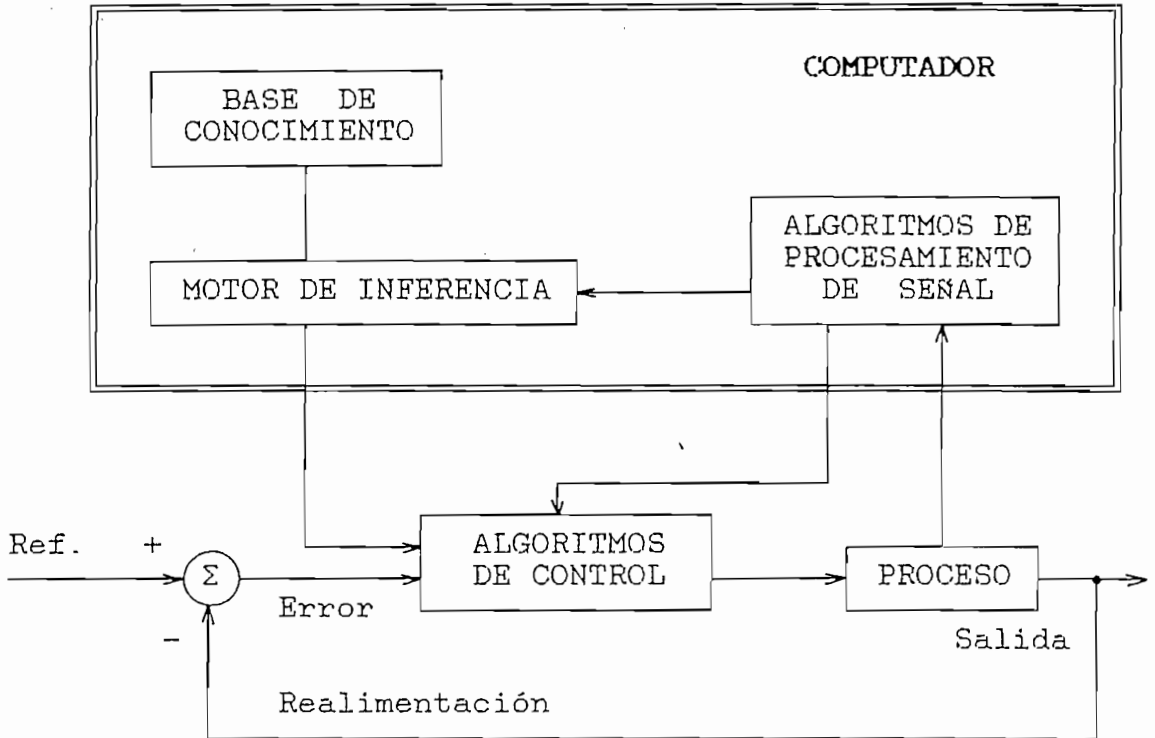


Fig. 2.3: Arquitectura del Control Basado en Conocimiento.

Los componentes del CBC pueden estar organizados bajo una estructura jerárquica mostrada en la Fig. 2.4. [Ref. 17]

El nivel más alto de la estructura está compuesta por la base de conocimiento y el motor de inferencia. Todos los razonamientos para detección de fallas, diagnóstico y selección de los esquemas de control son realizados en este nivel.

El motor de inferencia toma decisiones basadas en el conocimiento almacenado y el estado actual del sistema que es suministrado por el preprocesador, que se encuentra en el segundo nivel.

El preprocesador convierte las medidas tomadas en el proceso a un grupo de señales que describen el funcionamiento del mismo. El último nivel constituye los algoritmos de control.

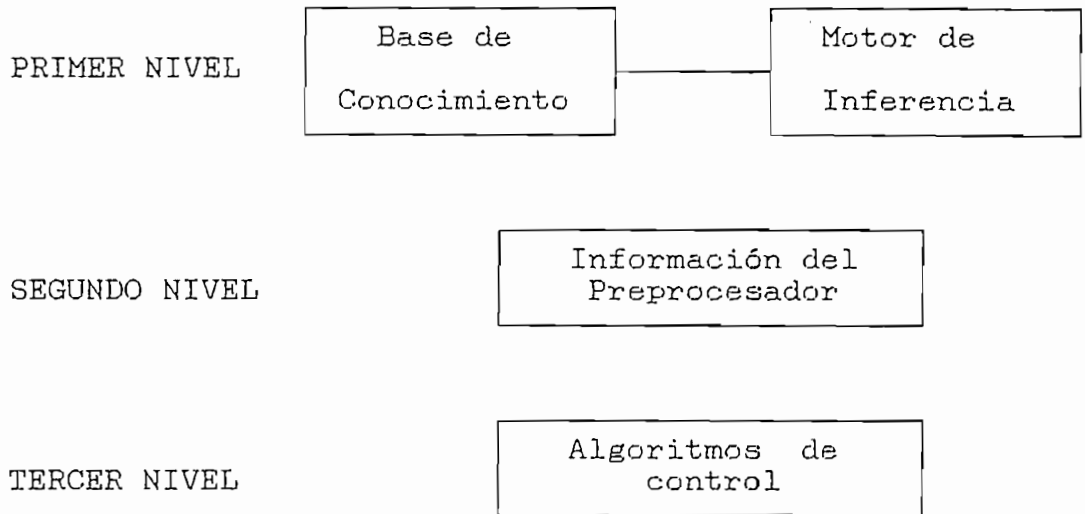


Fig. 2.4: Organización Jerárquica del controlador basado en conocimiento.

CBC tiene tres principales problemas por resolver. El primer problema consiste en cómo se podría adquirir el conocimiento. El segundo problema es la representación e implantación de este conocimiento y finalmente los requerimientos de software y hardware que este tipo de aplicación necesita. [Ref. 8]

### 2.2.2 CONTROL BASADO EN REGLAS Y LOGICA DIFUSA.

Control Basado en Reglas, intenta modelar las estrategias de control manual de los operadores humanos, las que son

Una vez adquirido el conocimiento, el ingeniero de sistemas, se encuentra muchas veces en problemas para poder representar la información adquirida del experto, ya que este conocimiento es inexacto, lleno de términos como: frecuente, bajo, alto, medio, etc.

Para poder representar este tipo de definiciones vagas se utiliza las técnicas de Lógica Difusa. Técnicas de CBR y lógica difusa fueron originalmente introducidas por Zadeh (1973), como un medio para capturar experiencia de un experto y tratar con incertidumbre. [Ref. 20]

En lógica difusa, términos como alto, medio, bajo son representadas en escalas, ej. la temperatura que se muestra en la Fig. 2.5.

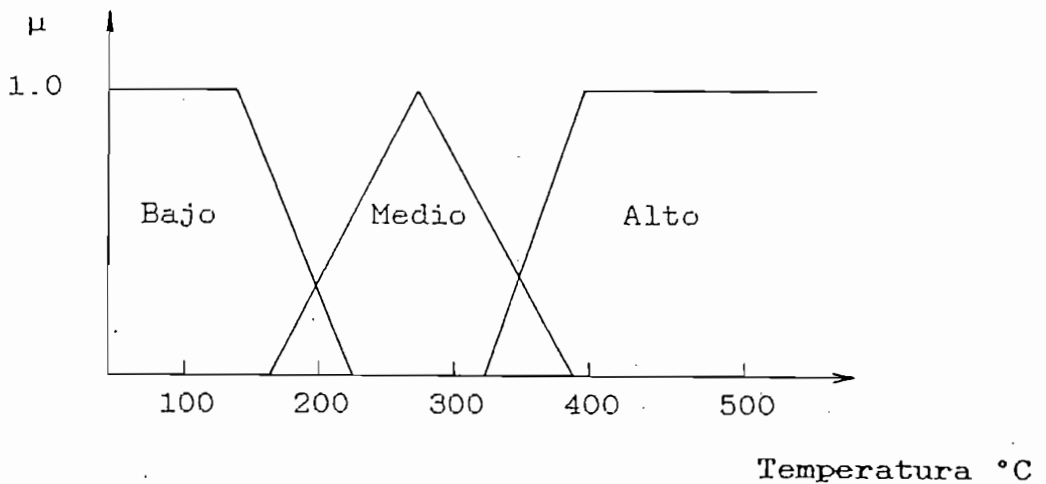


Fig. 2.5: Representación en Lógica Difusa.

Para cada término, existe un rango de temperatura sobre el cual puede ser aplicado. Así mismo, existe otro rango de temperatura sobre el cual no se lo puede aplicar.

En lógica convencional se utilizan relaciones booleanas, donde se tienen dos valores verdadero (1) y falso (0), mientras que en la teoría de Lógica Difusa se define una función de pertenencia  $\mu$ , la cual asocia a cada término con un número real entre 0 y 1.

Así en lógica convencional, "La temperatura es alta", es verdadero o falso dependiendo si la medición pertenece o no al conjunto de temperatura alta, sin embargo, Lógica Difusa permite grados de verdad intermedios entre verdadero y falso, lo que hace posible decir que cierta temperatura es alta con un grado de verdad, por ejemplo de 0.5.

De esta forma, la teoría de Lógica Difusa abre la posibilidad de representar el conocimiento inexacto, cuando a través de un conjunto de reglas no es posible. Un estudio más detallado de esta técnica se puede encontrar en las [Ref. 12] y [Ref. 20].

### 2.3 SISTEMAS EXPERTOS EN TIEMPO REAL.

Se los puede considerar como una extensión de los sistemas expertos para aplicaciones a tiempo real en control de un

Por esta razón es importante estimar el tiempo necesario para procesamientos en el computador.

- El sistema debe responder a eventos asincrónicos, lo que incluye interrumpir tareas menos importantes y enfocar la atención a problemas primordiales.
  
- Se puede utilizar procesamiento paralelo, donde el sistema experto podría llevar a cabo actividades en forma concurrente. Para implantar esta técnica es necesario utilizar el software y hardware adecuados.
  
- El sistema debe ser capaz de operar con información inexacta, por ejemplo pérdida de información de los sensores.
  
- La velocidad de ejecución debe ser lo suficientemente alta.

### 2.3.1 EJEMPLOS DE SISTEMAS EXPERTOS EN TIEMPO REAL.

Actualmente, no existe sistemas expertos en tiempo real que cumplan las características anteriormente mencionadas. Sin embargo, muchos trabajos se han realizado, los cuales en cierto grado satisfacen estos requerimientos. Algunos sistemas expertos en tiempo real disponibles en el mercado son: [Ref. 8]



**PICON:** Sistema de control experto en tiempo real desarrollado por LISP Machine INC., utiliza los datos provenientes de sensores de la planta, para diagnosticar fallas en el proceso.

**EXACT:** Controlador experto que fue implantado con técnicas convencionales por FOXBORO. Está basado en modelos de identificación para ajuste de un controlador PID adaptivo. Los parámetros del controlador son ajustados utilizando heurística.

**ESAC:** Es un regulador auto-ajustable que emplea técnicas de sistemas expertos en tres diferentes módulos: el sistema identificador, sistema de control y el supervisor.

**RESCU:** Consiste de cinco procesos, cuatro de ellos son convencionales, mientras que el quinto corresponde a la base de conocimiento. Utiliza esencialmente un motor de inferencia con encadenamiento hacia adelante para medir datos que estiman la calidad del producto.

**HEXCON:** Es un sistema híbrido, que combina control computarizado convencional y técnicas de sistemas basados en conocimiento. El sistema trabaja utilizando inferencia con encadenamiento hacia adelante y atrás. HEXCON es implementado en PASCAL y utiliza el microprocesador 8086.

**MUSE:** Es un paquete de desarrollo de sistemas basados en

conocimiento, diseñado por Cambridge Consultants Ltd. Emplea el motor de inferencia con encadenamiento hacia adelante y atrás, además emplea representación de estructuras.

ART: Es una herramienta de desarrollo de Inteligencia Artificial, muy útil para sistemas basados en conocimiento en tiempo real. Su mecanismo de inferencia utiliza la técnica de encadenamiento hacia adelante.

G2: Es uno de los más populares y versátiles sistemas expertos en tiempo real aplicados a procesos industriales. Integra funciones para control de procesos, técnicas CIM (Manufactura Integrada por Computador), monitoreo y pruebas automáticas. Puede ser ejecutado en diferentes tipos de computadores así como también en estaciones de trabajo para IA. [Ref. 12]

G2 hace uso de esquemas de representación de conocimiento tales como: modelos de sistemas, dinámica de sistemas, modelos analíticos o intuitivos que pueden ser comparados con el proceso real trabajando en línea con la planta.

La interfaz del usuario de G2 ha sido cuidadosamente diseñada, para que el conocimiento pueda ser ingresado fácilmente y que el sistema pueda ser configurado a las necesidades del ingeniero de control.

El procesamiento de información puede ser distribuido sobre

una red de computadores, con lo cual se tendría un ambiente de trabajo jerárquico, donde el nivel más bajo podría ser la lectura y pruebas de sensores.

**EXTASE:** Es un sistema de manejo de alarmas inteligente, tiene como objetivo encontrar la causa de una señal de alarma, es una herramienta más de diagnóstico que de control y fue aplicado a un sistema térmico, considerando 24 lazos de control, alrededor de 150 sensores y 50 alarmas.

También existen muchas aplicaciones fuera de línea (off-line), es decir que no se conectan directamente al proceso, que se las utiliza como herramientas de consulta y diseño de sistemas de control experto.

## CAPITULO III

### IMPLANTACION DEL CONTROLADOR EXPERTO

#### 3.1 INTRODUCCION.

El presente trabajo constituye una continuación de "An Expert System For Supervisory Process Control" [Ref. 8], donde se desarrolla un sistema de control basado en conocimiento para control de procesos lineales. Técnicas convencionales de control son utilizadas para producir un esquema de control adaptivo, las cuales son supervisadas por el sistema experto.

El objetivo de este proyecto es aplicar un sistema de control experto basado en conocimiento a un proceso lineal que permita satisfacer las siguientes características:

- Que pueda controlar satisfactoriamente procesos lineales los cuales se encuentren variando en el tiempo o sujetos a perturbaciones.
- Que produzca una respuesta rápida con sobreimpulso aceptable.

- Que la base de conocimiento pueda ser fácilmente modificada y actualizada.
- Que se necesite una mínima cantidad de conocimiento del funcionamiento del proceso.

Para lograr este propósito, el control experto utiliza la información de su base de conocimiento, del proceso a controlar y salidas de los algoritmos de control, para decidir la acción de control que debe llevar a cabo.

### 3.2 ARQUITECTURA DEL CONTROLADOR EXPERTO.

La arquitectura del controlador experto se muestra en la Fig. 3.1, la cual mantiene la estructura de los sistemas de control basado en conocimiento que se detalló en la sección 2.2.1.

Considerando esta estructura y organización jerárquica del control experto basado en conocimiento, se procedió a desarrollar un programa computacional, donde el componente más importante lo constituye la base de conocimiento.

Otro elemento del controlador experto es el bloque de controles, en el cual se proponen tres tipos de algoritmos de control convencionales: PID, DEADBEAT y SELF TUNING.

Como elementos externos al proceso tenemos al actuador, tipo de excitación y la interfaz de adquisición de datos.

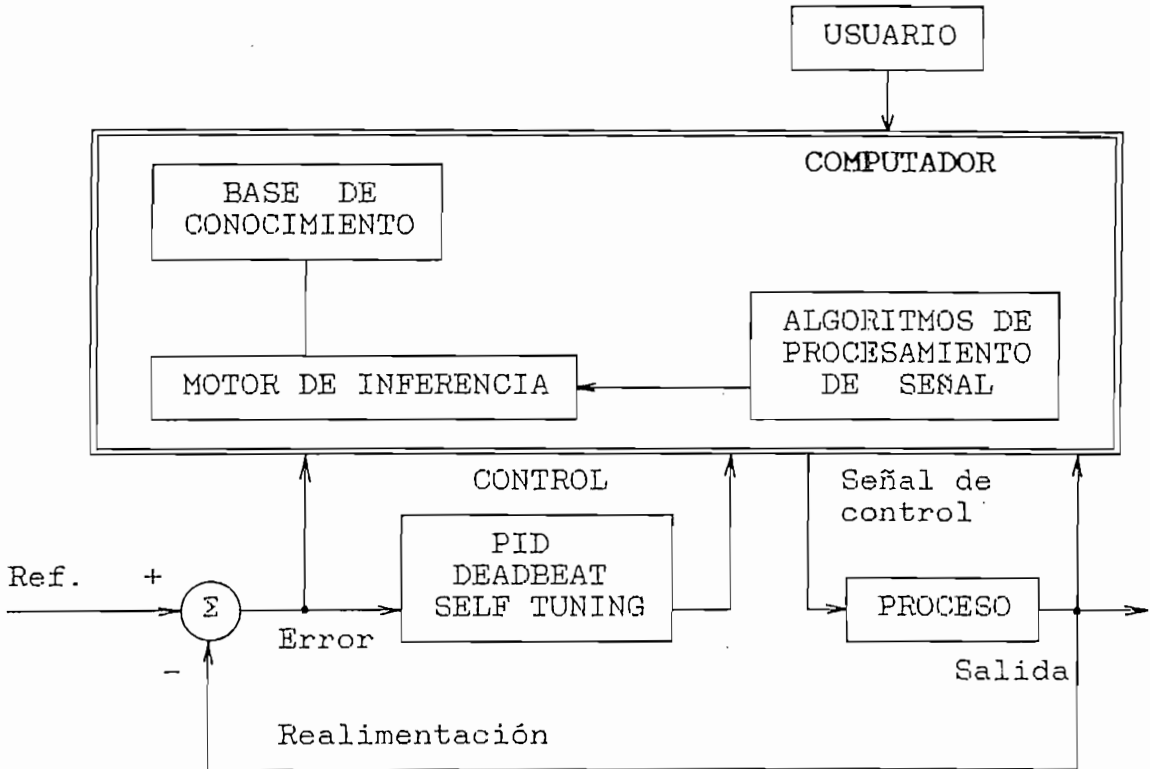


Fig. 3.1: Arquitectura del Controlador Experto.

### 3.3 CONTROLADOR EXPERTO.

Se propone dos técnicas de operación para el sistema experto:

- CONTROL EXPERTO 1.- (CE1) Selecciona la mejor acción de control a ser conectada a la planta en cada intervalo de muestreo.
- CONTROL EXPERTO 2.- (CE2) Asigna factores de ponderación

a las acciones de control, así los controles contribuyen en algún grado a la señal de control final.

### 3.3.1 CONTROL EXPERTO 1: CONMUTACION.

El sistema basado en conocimiento decide cuál es la mejor acción de control a ser conectada al proceso en cada intervalo de muestreo. De esta forma el control experto funciona como un switch de múltiples entradas y una sola salida.

La Fig. 3.2 muestra el diagrama de bloques del Control Experto 1. Las entradas corresponden a las señales de cada uno de los controles y la salida es la señal de control seleccionada, que será aplicada al proceso.

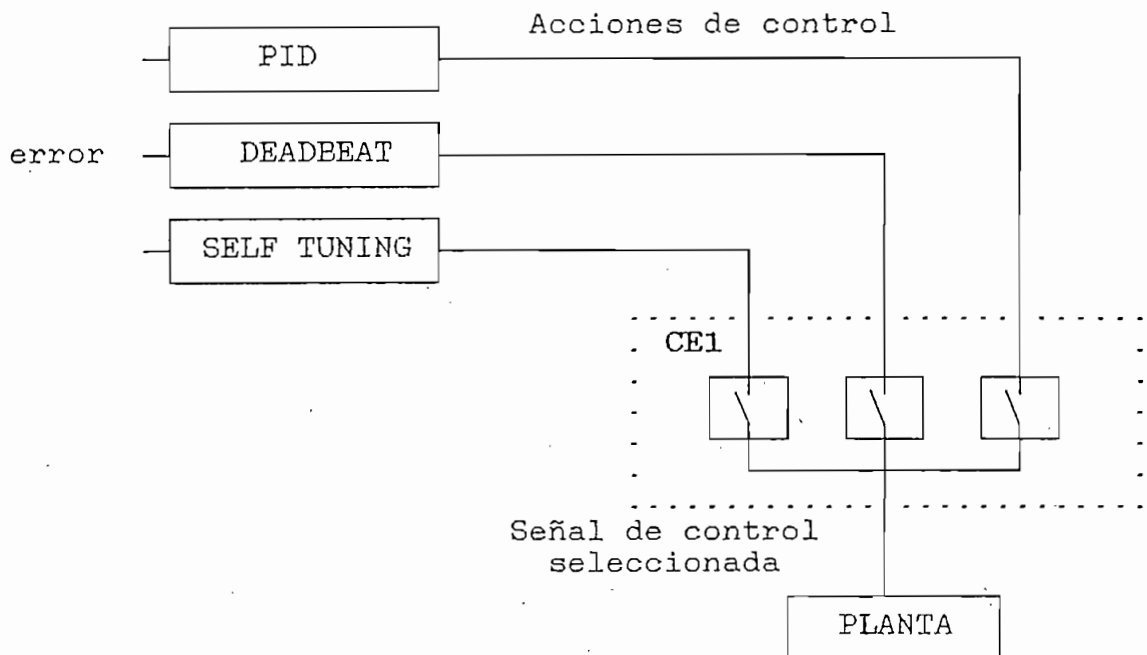


Fig. 3.2: Control Experto 1.

### 3.3.2 CONTROL EXPERTO 2: FACTORES DE PONDERACION.

En este caso el sistema experto opera como un sumador, utilizando la misma base de conocimiento que el Control Experto 1. La señal de control que actuará sobre la planta será una combinación lineal de todas las acciones de control, donde a cada una se ha asignado un valor porcentual con el que contribuirá a la señal de control final.

La ponderación dada a cada uno de los controles dependerá del comportamiento del proceso en cada instante de muestreo. La Fig. 3.3 presenta el diagrama de bloques para el Control Experto 2.

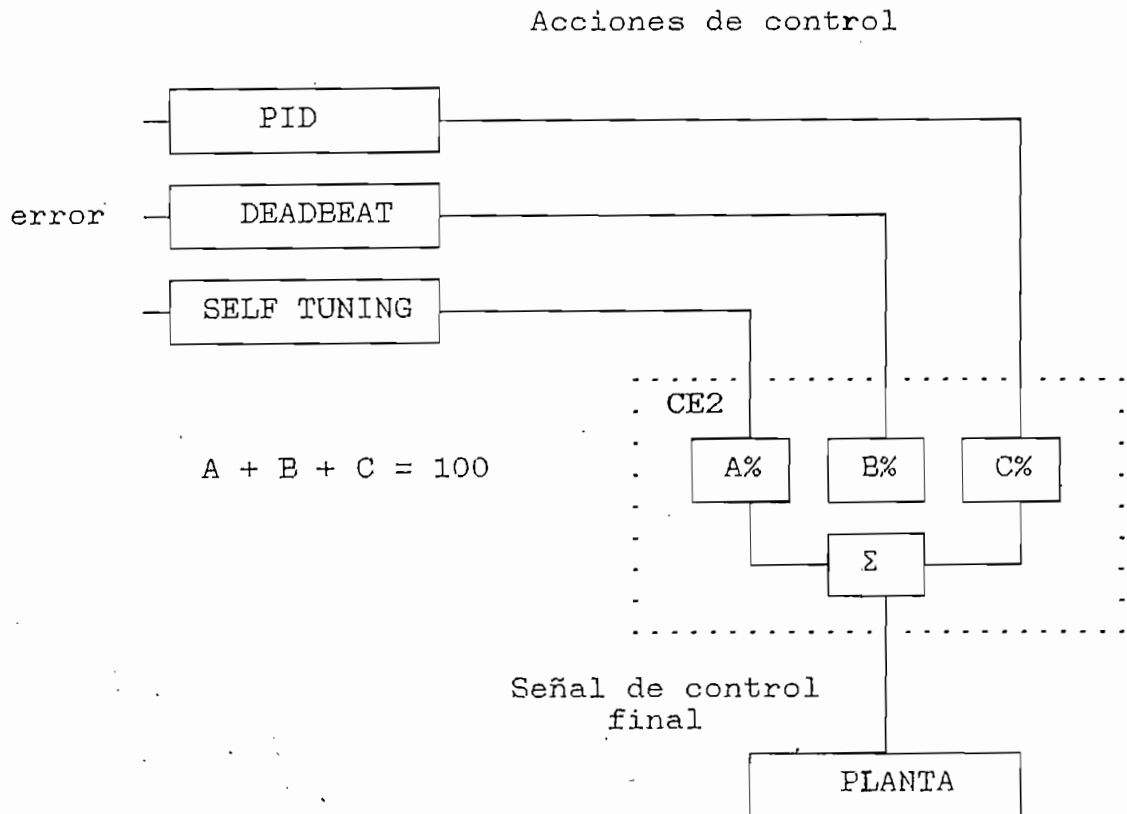


Fig. 3.3: Control Experto 2.



### 3.4 ADQUISICION DEL CONOCIMIENTO.

El objetivo básico de las técnicas del controlador experto desarrollado en este trabajo, fue el derivar un grupo de señales de control, las cuales son manipuladas por el controlador experto, el que decide cuándo utilizar un algoritmo en particular y así mantener un adecuado funcionamiento del proceso.

Para conseguir esta meta, fue necesario un período de adquisición de conocimiento, lo que involucra una serie de actividades para extraer la información necesaria del experto, en este caso un ingeniero de control, y trasladarla posteriormente a una Base de Conocimiento.

Para desarrollar la base de conocimiento se tomó como referencia el esquema que se presenta en la sección 1.3, donde se detalla el proceso a seguir para adquirir el conocimiento sobre un dominio en particular y poder elaborar un sistema experto.

En este caso se delimitó el campo de estudio del controlador experto, a sistemas lineales con o sin tiempo de retardo y variantes en el tiempo.

Definido el alcance del trabajo y siguiendo el esquema propuesto para el desarrollo de la base de conocimiento, fue necesario obtener un conocimiento básico del funcionamiento de

libre de ruido y acondicionada.

Una vez implantado el bloque de controles, se procedió a realizar pruebas para evaluar la operación de los procesos con cada uno de ellos. De esta forma es posible adquirir la experiencia necesaria, para extraer las mejores características de cada acción de control.

Con la finalidad de complementar el conocimiento adquirido, se realizó paralelamente consultas a ingenieros de control e investigaciones de los conceptos básicos en este dominio en libros y revistas especializadas.

El siguiente paso constituye la representación de la experiencia en una base de conocimiento estructurada por un conjunto de reglas de la forma:

*SI (condición) ... ENTONCES (acción)*

Una descripción detallada de la base de conocimiento se presenta en el siguiente capítulo.

Finalmente el motor de inferencia que opera bajo la técnica de encadenamiento hacia adelante, selecciona la regla que debe ser ejecutada en el momento adecuado. [Ref. 20]. Este se encuentra integrado dentro del ambiente de trabajo del lenguaje C.

## CAPITULO IV

### ESTRUCTURA DEL PROGRAMA

#### 4.1 INTRODUCCION.

El controlador experto basado en conocimiento ha sido desarrollado bajo sistema operativo MS-DOS y utilizando el compilador QUICK C Versión 2.01 para obtener el módulo ejecutable.

Se consideró al lenguaje C como herramienta de programación, ya que produce programas rápidos, eficientes y facilita la programación estructurada.

Es verdad que existen otros lenguajes utilizados para la implantación de sistemas expertos, pero son difíciles de adaptar a una programación general. Por otro lado la existencia de numerosas librerías y programas, que soportan interfaces de entrada/salida y opciones de gráficos, proporcionaron una variedad de alternativas para trabajar en la forma más productiva.

Para la elaboración del programa se utilizó una estructura

modular, la que satisface la arquitectura del controlador experto detallada anteriormente. Este mecanismo de programación permitió que cada componente del sistema sea desarrollado por módulos, los que fueron grabados separadamente y luego incluidos al programa ejecutable en el proceso de compilación.

En la Fig. 4.1 se presenta la estructura modular del programa y la relación entre módulos.

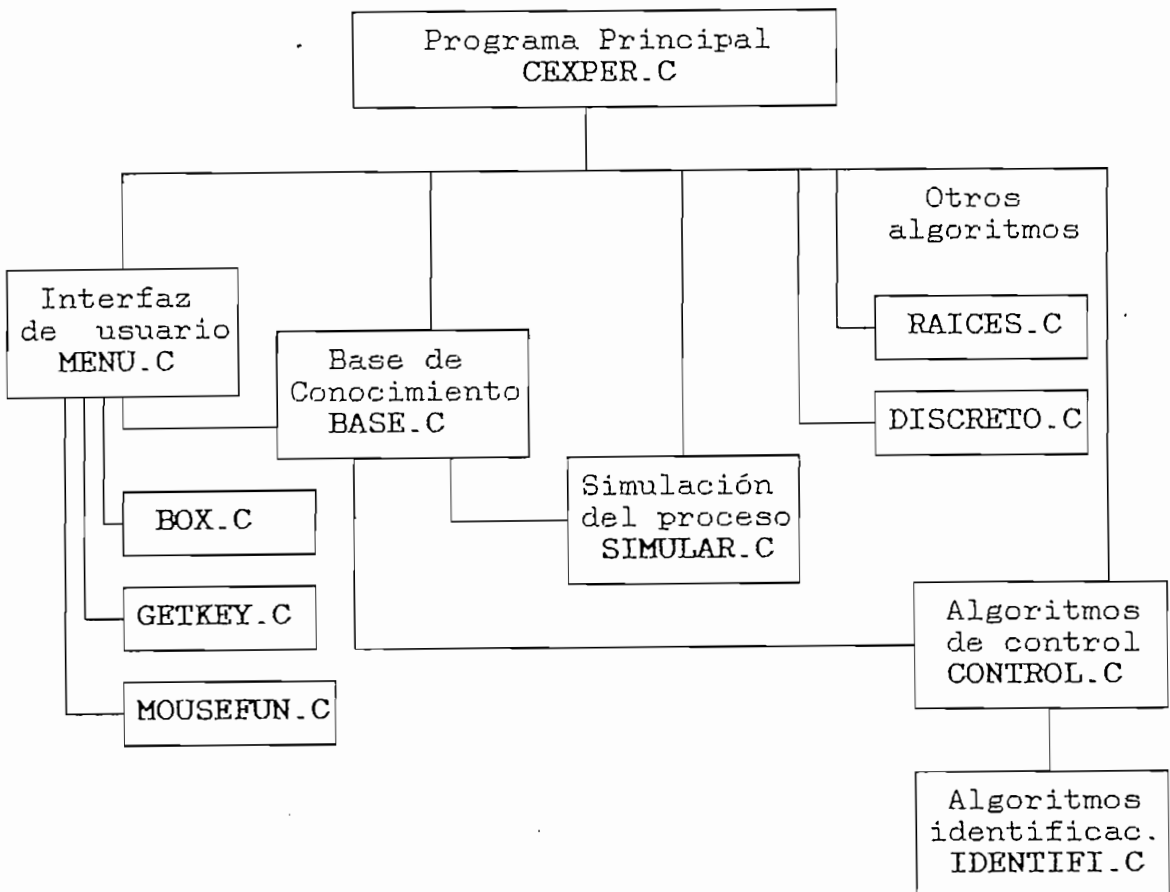


Fig. 4.1: Estructura modular del programa.

El programa desarrollado está formado por los siguientes archivos:

- Lista de módulos para el compilador: CEXPER.MAK
- Archivo ejecutable: CEXPER.EXE
- Archivos de funciones tipo y definiciones: FUNCION.H, MENU.H, BOX.H, T\_COLORS.H, GETKEY.H, MOUSEFUN.H
- Módulos del programa: CEXPER.C, MENU.C, BOX.C, MOUSEFUN.C, GETKEY.C, RAICES.C, SIMULAR.C, CONTROL.C, IDENTIFI.C, DISCRETO.C, BASE.C.

#### 4.2 PROGRAMA PRINCIPAL.

El módulo CEXPER.C tiene la función de supervisar la comunicación ente los diferentes módulos y llamarlos cuando son requeridos. Es el encargado de verificar la operación correcta de los dispositivos de entrada/salida y mostrar mensajes de error en caso de ser necesario. Incluye además la definición de variables y arreglos que son utilizados para generación de resultados.

Para ejecutar el programa principal se digita CEXPER, con lo que se invoca al módulo: inferfaz de usuario MENU.C, el cual permite la comunicación usuario - computador a través de menús.

La elaboración de estos menús, el control del mouse y teclado, utiliza algunos programas de la [Ref. 25] : MENU.C, BOX.C, MOUSEFUN.C y GETKEY.C. El menú que se presenta en pantalla es el que se indica en la Fig. 4.2.

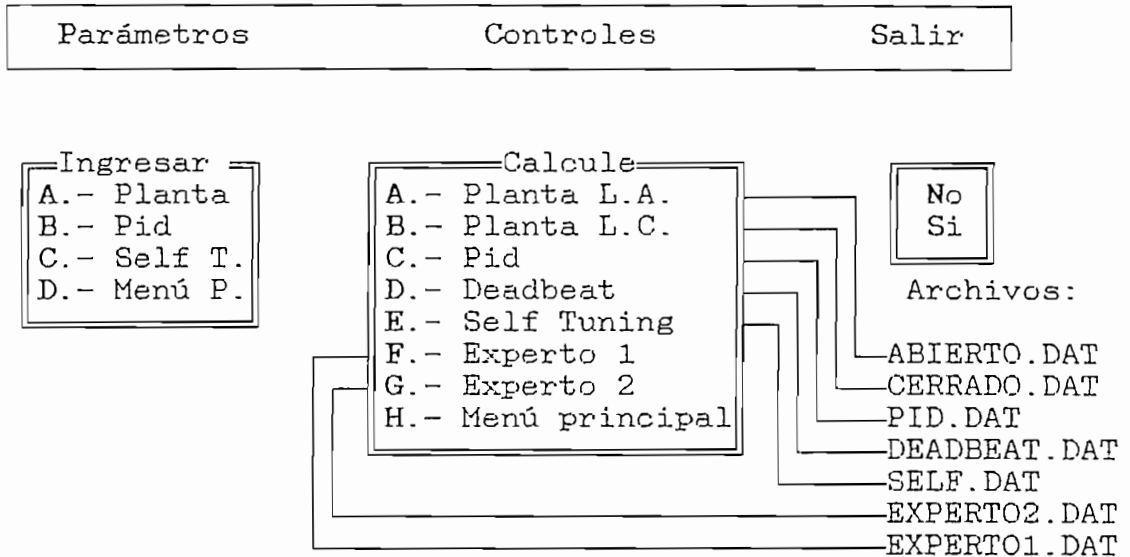


Fig. 4.2: Interfaz de usuario.

Para iniciar, se debe seleccionar la primera opción del menú Parámetros, el que despliega el submenú Ingresar, que se indica en la Fig. 4.2. Este permite el ingreso de la función de transferencia de la planta y otros parámetros necesarios para el análisis como: el período de muestreo, tiempo de retardo, límite del actuador y factor de variación de parámetros. Además se dispone de opciones para el ingreso de información sobre el bloque controlador.

Una vez ingresados los datos de la planta y controles, se puede

Para salir del programa, seleccionar la opción Salir del menú principal y finalmente Si del submenú.

#### 4.3 BASE DE CONOCIMIENTO.

Basados en la experiencia adquirida con los sistemas lineales, las consultas e investigaciones realizadas, se desarrolla el módulo que contiene la base de conocimiento, al que se denomina BASE.C.

Este archivo está formado por dos funciones EXPERTO1() y EXPERTO2(), cada uno de los cuales posee un conjunto de reglas que satisfacen las condiciones de operación impuestas por las técnicas del controlador experto que se definieron en la sección 3.3. Donde EXPERTO1() contiene las reglas de CE1 y EXPERTO2() las reglas de CE2.

Para la formulación de estas reglas se considera la señal de error del sistema, la cual es dividida en diferentes regiones de operación, como se indica en la Fig. 4.3.

Las regiones contenidas en la señal de error son:

1. Cambio de la referencia o set point.
2. Zona de tiempo muerto o retardo de transporte.
3. Región donde el error es mayor al 10% de la referencia y su derivada negativa.

4. Región donde el valor absoluto del error menor o igual al 10% de la referencia.
5. Región de error negativo y menor al -10% del set point.
6. Región de error positivo, mayor al 10% del set point y derivada positiva.

Además se consideran algunos criterios en el desarrollo de las reglas de producción:

- 1.- Mantener como objetivo una respuesta ideal del sistema, la que presente un mínimo sobreimpulso, sea rápida, posea un corto tiempo de establecimiento y un error en estado estable igual a cero.
- 2.- Considerar el orden del sistema  $n$ , al momento de seleccionar el control por conectar a la planta o el grado de ponderación asignado al mismo, ya que los procesos se comportan de diferente manera.
- 3.- Para el caso de variación de parámetros, utilizar una señal de referencia persistente, lo que significa que esta señal puede variar entre 0 al set point y mantenerse en ese valor por cierto intervalo de tiempo. •

En base al orden del sistema, las regiones de la señal error indicadas anteriormente y el nivel de la referencia, se presenta las siguientes reglas que forman parte de la base de conocimiento:



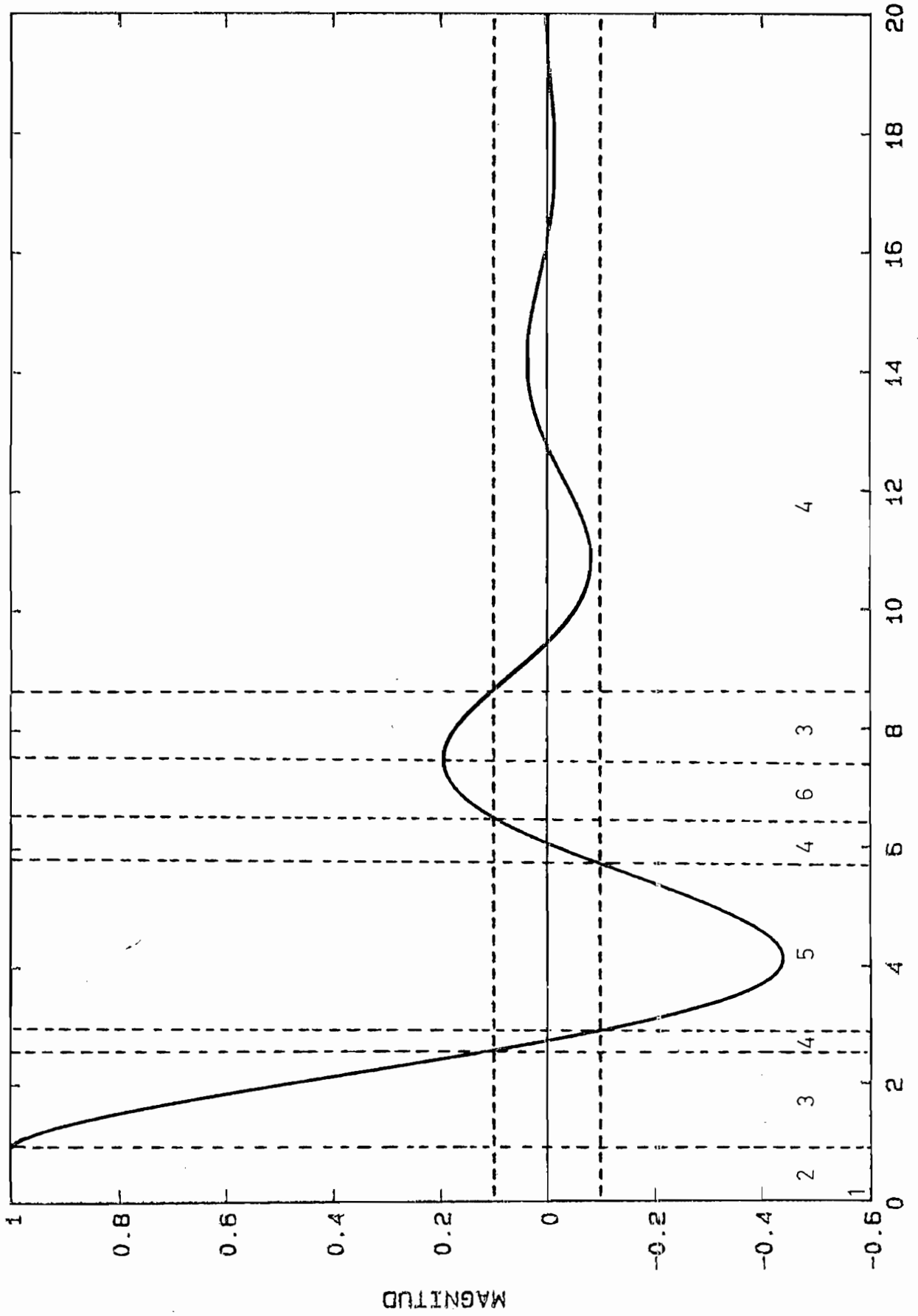


Fig. 4.3 Regiones de la señal de error con referencia positiva

1.- Si el sistema es de primer orden:

1.1 El nivel de la referencia es igual al set point.

1.1.1 En el momento que exista un cambio de la referencia.

- a) EXPERTO1(): La señal de control Deadbeat es seleccionada para acelerar la respuesta de la planta, constituyendo la primera decisión de control siempre que suceda una variación del set point. La regla correspondiente se indica a continuación:

SI *Existe un cambio de la referencia*

ENTONCES *Control = Deadbeat*

- b) EXPERTO2(): El mayor grado de ponderación es asignado al control escogido para la función EXPERTO1().

SI *Existe un cambio de la referencia*

ENTONCES *Señal de Control = 70% Deadbeat + 20% Self Tuning  
+ 10% Pid*

1.1.2 Una vez que el sistema se ha inicializado y no existe mayor variación del error, es decir el valor del mismo se mantiene muy cercano al de la referencia y su incremento en este caso representado por su derivada es menor a 0.1, se determina entonces que el sistema posee retardo de transporte.

- a) **EXPERTO1():** Selecciona el control Self Tuning para asegurar que la respuesta de la planta se mantenga dentro de las características especificadas anteriormente.

Esto es posible, ya que el controlador toma datos de la respuesta del sistema para calcular la señal de control adecuada, de esta forma se evita conectar al proceso acciones de control no válidas en el tiempo.

SI  $error > 99\% \text{ referencia}$

Y  $|\delta error| < 0.1$

ENTONCES  $Control = Self Tuning$

- b) **EXPERTO2():**

SI  $error > 99\% \text{ referencia}$

Y  $|\delta error| < 0.1$

ENTONCES  $Señal \ de \ control = 80\% \ Self \ Tuning + 20\% \ Pid$

1.1.3 Si el error del sistema es mayor al 10% de la referencia y su derivada es negativa.

- a) **EXPERTO1():** El control Self Tuning es utilizado para mantener un acercamiento gradual de la respuesta del sistema a la referencia. La regla que satisface esta condición se expresa de la siguiente forma:

SI  $error > 10\% referencia$   
 Y  $|\delta error| < 0$   
 ENTONCES  $Control = Self\ Tuning$

b) EXPERTO2(=

SI  $error > 10\% referencia$   
 Y  $|\delta error| < 0$   
 ENTONCES  $Señal\ de\ control = 90\% Self\ Tuning + 10\% Pid$

1.1.4 Si el valor absoluto del error es menor o igual al 10% de la referencia.

a) EXPERTO1(= Conecta el control Self Tuning para mantener la respuesta de la planta con un error en estado estable igual a cero, evitando de esta forma que la señal de salida sobrepase la banda del +/- 10% de la referencia. La regla correspondiente es:

SI  $|\error| \leq 10\% referencia$   
 ENTONCES  $Control = Self\ Tuning$

b) EXPERTO2(=

SI  $|\error| \leq 10\% referencia$   
 ENTONCES  $Señal\ de\ control = 80\% Self\ Tuning + 20\% Pid$

1.1.5 Si el error es negativo y menor a -10% de la

SI *error > 10% referencia*  
 Y *δerror > 0*  
 ENTONCES *Control = max(Pid, Self Tuning)*

b) EXPERTO2():

SI *error > 10% referencia*  
 Y *δerror > 0*  
 ENTONCES *Señal de control = 90% max(Pid, Self Tuning) +*  
*10% min(Pid, Self Tuning)*

2.- Si el sistema es de segundo u orden superior:

2.1 El nivel de la referencia es igual al set point.

2.1.1 En el momento que exista un cambio de la referencia.

a) EXPERTO1(): Cuando existe un cambio en la referencia, la señal de error es máxima y para llevar este valor lo más rápido posible a cero de igual forma que en el literal 1.1.1, se selecciona el control Deadbeat como la primera acción de control. De esta forma se acelera la respuesta de la planta hacia el valor de referencia. En este caso la regla se define de la siguiente forma:

SI *Existe un cambio de la referencia*

ENTONCES *Control = Deadbeat*

- b) EXPERTO2(): El mayor grado de ponderación es asignado al control escogido para la función EXPERTO1().

SI *Existe un cambio de la referencia*

ENTONCES *Señal de Control = 70% Deadbeat + 20% Self Tuning  
+ 10% Pid*

2.1.2 Bajo la misma consideración realizada en el literal 1.1.2 al identificar retardo de transporte tenemos:

- a) EXPERTO1(): Escoge el control Self Tuning para conectar al proceso.

SI *error > 99% referencia*

Y *|\deltaerror| < 0.1*

ENTONCES *Control = Self Tuning*

- b) EXPERTO2():

SI *error > 99% referencia*

Y *|\deltaerror| < 0.1*

ENTONCES *Señal de control = 80% Self Tuning + 20% Pid*

2.1.3 Si el error es mayor al 10% de la referencia y su derivada es negativa.

- a) EXPERTO1(): Selecciona la señal del control Self Tuning. El criterio que justifica esta elección es el mismo que se indica en el literal 1.1.3.

```

SI          error > 10% referencia
Y          |δerror| < 0
ENTONCES   Control = Self Tuning

```

- b) EXPERTO2():

```

SI          error > 10% referencia
Y          |δerror| < 0
ENTONCES   Señal de control = 80% Self Tuning + 20% Pid

```

2.1.4 Si el valor absoluto del error es menor o igual al 10% de la referencia.

- a) EXPERTO1(): Conecta el control Pid, para disminuir el sobreimpulso que presenta la respuesta de la planta, manteniendo a la vez un error en estado estable igual a cero.

En caso de tratarse de sistemas de orden superior a dos, se elige la señal de control Self Tuning, con lo que se da una mejor característica a la respuesta de la planta.

```

SI          |error| ≤ 10% referencia

```

Y  $n = 2$   
 ENTONCES  $Control = Pid$

SI  $|error| \leq 10\% referencia$   
 Y  $n > 2$   
 ENTONCES  $Control = Self Tuning$

b) EXPERTO2():

SI  $|error| \leq 10\% referencia$   
 Y  $n = 2$   
 ENTONCES  $Señal de control = 90\% Pid + 10\% Self Tuning$

SI  $|error| \leq 10\% referencia$   
 Y  $n > 2$   
 ENTONCES  $Señal de control = 90\% Self Tuning + 10\% Pid$

2.1.1.5 Si el error es negativo y menor al -10% de la referencia. Como se indica en el literal 1.1.5 la respuesta del sistema se aleja de la referencia, para evitar que esto suceda se selecciona un controlador lento.

a) EXPERTO1(): Toma la mínima señal de control entre los controles Pid y Self Tuning.

SI  $error < -10\% referencia$   
 ENTONCES  $Control = \min(Pid, Self Tuning)$



b) EXPERTO2():

SI  $error < -10\% \text{ referencia}$

ENTONCES  $Señal \ de \ control = 90\% \min(Pid, \ Self \ Tuning) +$   
 $10\% \max(Pid, \ Self \ Tuning)$

2.1.6 Si el error es mayor al 10% de la referencia y su derivada es positiva.

a) EXPERTO1(): Selecciona la máxima señal de control entre los controles Pid y Self Tuning. La justificación de esta regla se indica en el literal 1.1.6.

SI  $error > 10\% \text{ referencia}$

Y  $\delta error > 0$

ENTONCES  $Control = \max(Pid \ y \ Self \ Tuning)$

b) EXPERTO2():

SI  $error > 10\% \text{ referencia}$

Y  $\delta error > 0$

ENTONCES  $Señal \ de \ control = 90\% \max(Pid, \ Self \ Tuning) +$   
 $10\% \min(Pid, \ Self \ Tuning)$

Cuando el nivel de referencia pasa a tomar valores iguales a 0, la respuesta del sistema trata de alcanzar la posición de

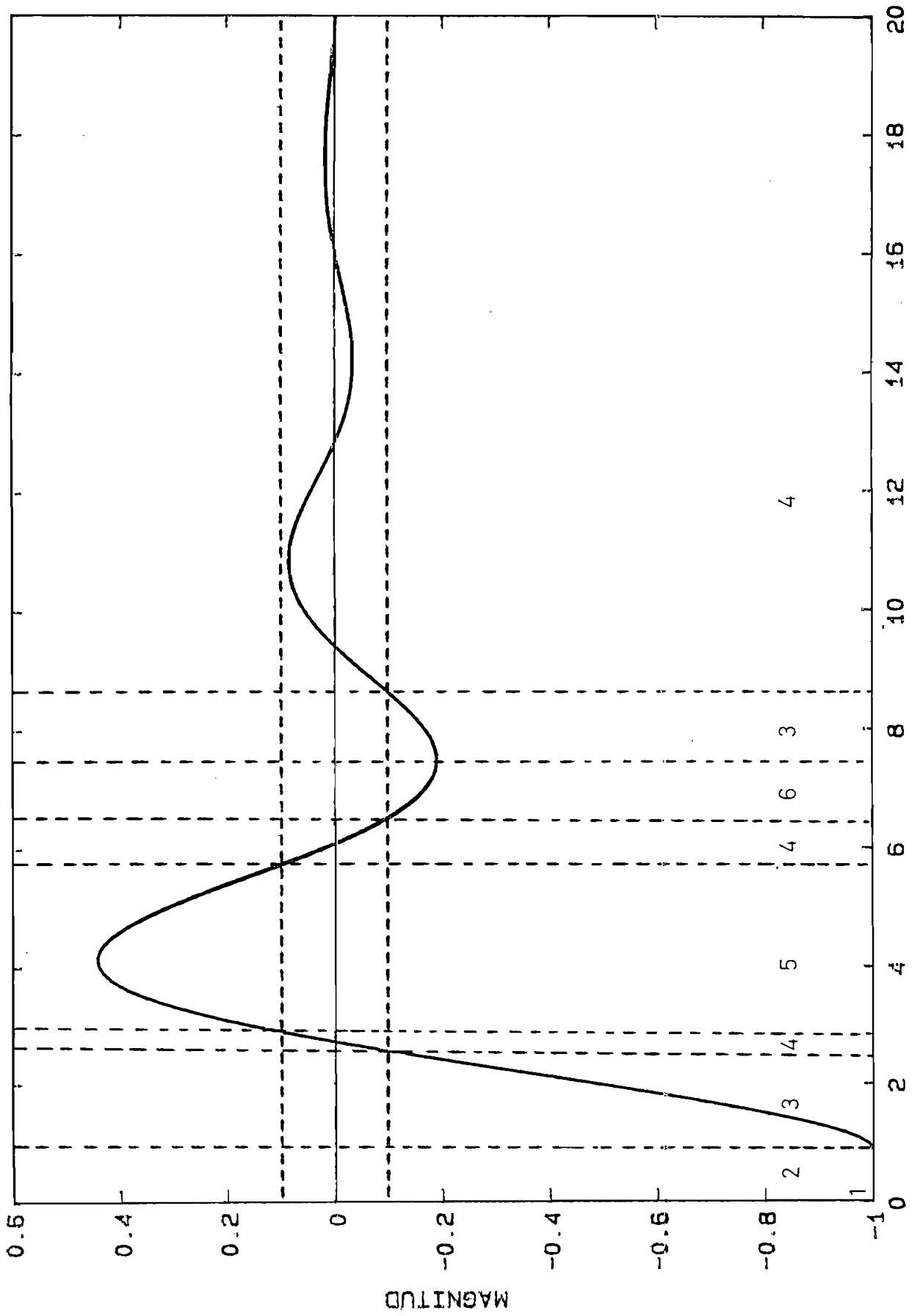


Fig. 4.4 Regiones de la señal de error con referencia = 0

3.- Si el sistema es de primer orden:

3.1 El nivel de la referencia es igual a 0.

3.1.1 En el momento que exista un cambio de la referencia.

a) EXPERTO1():

SI *Existe un cambio de la referencia*  
 ENTONCES *Control = Deadbeat*

b) EXPERTO2():

SI *Existe un cambio de la referencia*  
 ENTONCES *Señal de Control = 70% Deadbeat + 20% Self Tuning*  
*+ 10% Pid*

3.1.2 El sistema posee retardo de transporte.

a) EXPERTO1():

SI *error < -99% referencia*  
 Y *|δerror| < 0.1*  
 ENTONCES *Control = Self Tuning*

b) EXPERTO2():

SI *error < -99% referencia*

Y  $|\delta error| < 0.1$

ENTONCES *Señal de control = 80% Self Tuning + 20% Pid*

3.1.3 Si el error del sistema es menor al -10% de la referencia y su derivada es positiva.

a) EXPERTO1():

SI  $error < -10\% referencia$

Y  $|\delta error| > 0$

ENTONCES *Control = Self Tuning*

b) EXPERTO2():

SI  $error < -10\% referencia$

Y  $|\delta error| > 0$

ENTONCES *Señal de control = 90% Self Tuning + 10% Pid*

3.1.4 Si el valor absoluto del error es menor o igual al 10% de la referencia.

a) EXPERTO1():

SI  $|error| \leq 10\% referencia$

ENTONCES *Control = Self Tuning*

b) EXPERTO2():

SI  $|error| \leq 10\% \text{ referencia}$   
 ENTONCES  $Señal \ de \ control = 80\% \ Self \ Tuning + 20\% \ Pid$

3.1.5 Si el error es positivo y mayor al 10% de la referencia.

a) EXPERTO1():

SI  $error > 10\% \text{ referencia}$   
 ENTONCES  $Control = \max(Pid, \text{Self Tuning})$

b) EXPERTO2():

SI  $error > 10\% \text{ referencia}$   
 ENTONCES  $Señal \ de \ control = 90\% \max(Pid, \text{Self Tuning}) +$   
 $10\% \min(Pid, \text{Self Tuning})$

3.1.6 Si el error es menor al -10% de la referencia y su derivada es negativa.

a) EXPERTO1():

SI  $error < -10\% \text{ referencia}$   
 Y  $\delta error < 0$   
 ENTONCES  $Control = \min(Pid, \text{Self Tuning})$

b) EXPERTO2():

SI  $error < -10\% \text{ referencia}$

Y  $\delta error < 0$   
 ENTONCES  $Señal\ de\ control = 90\% \min(Pid, Self\ Tuning) +$   
 $10\% \max(Pid, Self\ Tuning)$

4.- Si el sistema es de segundo u orden superior:

4.1 El nivel de la referencia es igual a 0.

4.1.1 En el momento que exista un cambio de la referencia.

a) EXPERTO1():

SI  $Existe\ un\ cambio\ de\ la\ referencia$   
 ENTONCES  $Control = Deadbeat$

b) EXPERTO2():

SI  $Existe\ un\ cambio\ de\ la\ referencia$   
 ENTONCES  $Señal\ de\ Control = 70\% Deadbeat + 20\% Self\ Tuning$   
 $+ 10\% Pid$

4.1.2 El sistema posee retardo de transporte.

a) EXPERTO1():

SI  $error < -99\% referencia$   
 Y  $|\delta error| < 0.1$

ENTONCES *Control = Self Tuning*

b) EXPERTO2():

SI *error < -99% referencia*

Y *|\delta error| < 0.1*

ENTONCES *Señal de control = 80% Self Tuning + 20% Pid*

4.1.3 Si el error del sistema es menor al -10% de la referencia y su derivada es positiva.

a) EXPERTO1():

SI *error < -10% referencia*

Y *|\delta error| > 0*

ENTONCES *Control = Self Tuning*

b) EXPERTO2():

SI *error < -10% referencia*

Y *|\delta error| > 0*

ENTONCES *Señal de control = 80% Self Tuning + 20% Pid*

4.1.4 Si el valor absoluto del error es menor o igual al 10% de la referencia.

a) EXPERTO1():

SI *|\delta error| \leq 10% referencia*

Y  $n = 2$   
 ENTONCES  $Control = Pid$

SI  $|error| \leq 10\% referencia$   
 Y  $n > 2$   
 ENTONCES  $Control = Self\ Tuning$

b) EXPERTO2():

SI  $|error| \leq 10\% referencia$   
 Y  $n = 2$   
 ENTONCES  $Señal\ de\ control = 90\% Pid + 10\% Self\ Tuning$

SI  $|error| \leq 10\% referencia$   
 Y  $n > 2$   
 ENTONCES  $Señal\ de\ control = 90\% Self\ Tuning + 10\% Pid$

4.1.5 Si el error es positivo y mayor al 10% de la referencia.

a) EXPERTO1():

SI  $error > 10\% referencia$   
 ENTONCES  $Control = max(Pid, Self\ Tuning)$

b) EXPERTO2():

SI  $error > 10\% referencia$



ENTONCES *Señal de control = 90% max(Pid, Self Tuning) +  
10% min(Pid, Self Tuning)*

4.1.6 Si el error es menor al -10% de la referencia y su derivada es negativa.

a) EXPERTO1():

SI *error < -10% referencia*

Y *δerror < 0*

ENTONCES *Control = min(Pid, Self Tuning)*

b) EXPERTO2():

SI *error < -10% referencia*

Y *δerror < 0*

ENTONCES *Señal de control = 90% min(Pid, Self Tuning) +  
10% max(Pid, Self Tuning)*

Las reglas de los numerales 3 y 4 están formadas por la respectiva regla opuesta definida en los literales 1 y 2, donde se considera referencia mayor a 0.

Como resultado de aplicar las reglas expuestas anteriormente, la base de conocimiento queda estructurada de la siguiente forma:

## 1. Sistemas de primer orden

### 1.1 Referencia = set point

#### 1.1.1 Cambio de referencia

- a) Deadbeat
- b)  $0.7 \text{ Deadbeat} + 0.2 \text{ Self Tuning} + 0.1 \text{ Pid}$

#### 1.1.2 $\text{Error} > 0.99 \text{ referencia y } \text{abs}(\delta\text{error}) < 0.1$

- a) Self Tuning
- b)  $0.8 \text{ Self Tuning} + 0.2 \text{ Pid}$

#### 1.1.3 $\text{Error} > 0.1 \text{ referencia y } \text{abs}(\delta\text{error}) < 0$

- a) Self Tuning
- b)  $0.9 \text{ Self Tuning} + 0.1 \text{ Pid}$

#### 1.1.4 $\text{Abs}(\text{error}) \leq 0.1 \text{ referencia}$

- a) Self Tuning
- b)  $0.8 \text{ Self Tuning} + 0.2 \text{ Pid}$

#### 1.1.5 $\text{Error} < -0.1 \text{ referencia}$

- a)  $\text{min}(\text{Pid}, \text{Self Tuning})$
- b)  $0.9 \text{ min}(\text{Pid}, \text{Self Tuning}) + 0.1 \text{ max}(\text{Pid}, \text{Self Tuning})$

#### 1.1.6 $\text{Error} > 0.1 \text{ referencia y } \delta\text{error} > 0$

- a)  $\text{max}(\text{Pid}, \text{Self Tuning})$
- b)  $0.9 \text{ max}(\text{Pid}, \text{Self Tuning}) + 0.1 \text{ min}(\text{Pid}, \text{Self Tuning})$

## 1.2 Referencia = 0

### 1.2.1 Cambio de referencia

- a) Deadbeat
- b)  $0.7 \text{ Deadbeat} + 0.2 \text{ Self Tuning} + 0.1 \text{ Pid}$

### 1.2.2 $\text{Error} < -0.99 \text{ referencia y } \text{abs}(\delta\text{error}) < 0.1$

- a) Self Tuning
- b)  $0.8 \text{ Self Tuning} + 0.2 \text{ Pid}$

### 1.2.3 $\text{Error} < -0.1 \text{ referencia y } \text{abs}(\delta\text{error}) > 0$

- a) Self Tuning
- b)  $0.9 \text{ Self Tuning} + 0.1 \text{ Pid}$

### 1.2.4 $\text{Abs}(\text{error}) \leq 0.1 \text{ referencia}$

- a) Self Tuning
- b)  $0.8 \text{ Self Tuning} + 0.2 \text{ Pid}$

### 1.2.5 $\text{Error} > 0.1 \text{ referencia}$

- a)  $\text{max}(\text{Pid}, \text{Self Tuning})$
- b)  $0.9 \text{ max}(\text{Pid}, \text{Self Tuning}) + 0.1 \text{ min}(\text{Pid}, \text{Self Tuning})$

### 1.2.6 $\text{Error} < -0.1 \text{ referencia y } \delta\text{error} < 0$

- a)  $\text{min}(\text{Pid}, \text{Self Tuning})$
- b)  $0.9 \text{ min}(\text{Pid}, \text{Self Tuning}) + 0.1 \text{ max}(\text{Pid}, \text{Self Tuning})$

## 2. Sistemas de segundo u orden superior

### 2.1 Referencia = set point

#### 2.1.1 Cambio de referencia

- a) Deadbeat
- b)  $0.7 \text{ Deadbeat} + 0.2 \text{ Self Tuning} + 0.1 \text{ Pid}$

#### 2.1.2 $\text{Error} > 0.99 \text{ referencia y } \text{abs}(\delta\text{error}) < 0.1$

- a) Self Tuning
- b)  $0.8 \text{ Self Tuning} + 0.2 \text{ Pid}$

#### 2.1.3 $\text{Error} > 0.1 \text{ referencia y } \text{abs}(\delta\text{error}) < 0$

- a) Self Tuning
- b)  $0.8 \text{ Self Tuning} + 0.2 \text{ Pid}$

#### 2.1.4 $\text{Abs}(\text{error}) \leq 0.1 \text{ referencia y } n = 2$

- a) Pid
- b)  $0.9 \text{ Pid} + 0.1 \text{ Self Tuning}$

#### 2.1.5 $\text{Abs}(\text{error}) \leq 0.1 \text{ referencia y } n > 2$

- a) Self Tuning
- b)  $0.9 \text{ Self Tuning} + 0.1 \text{ Pid}$

#### 2.1.6 $\text{Error} < -0.1 \text{ referencia}$

- a)  $\text{min}(\text{Pid}, \text{Self Tuning})$
- b)  $0.9 \text{ min}(\text{Pid}, \text{Self Tuning}) + 0.1 \text{ max}(\text{Pid}, \text{Self Tuning})$

2.1.7 Error  $> 0.1$  referencia y  $\delta\text{error} > 0$

a)  $\max(\text{Pid}, \text{Self Tuning})$

b)  $0.9 \max(\text{Pid}, \text{Self Tuning}) + 0.1 \min(\text{Pid}, \text{Self Tuning})$

2.2 Referencia = 0

2.2.1 Cambio de referencia

a) Deadbeat

b)  $0.7 \text{ Deadbeat} + 0.2 \text{ Self Tuning} + 0.1 \text{ Pid}$

2.2.2 Error  $< -0.99$  referencia y  $\text{abs}(\delta\text{error}) < 0.1$

a) Self Tuning

b)  $0.8 \text{ Self Tuning} + 0.2 \text{ Pid}$

2.2.3 Error  $< -0.1$  referencia y  $\text{abs}(\delta\text{error}) > 0$

a) Self Tuning

b)  $0.8 \text{ Self Tuning} + 0.2 \text{ Pid}$

2.2.4  $\text{Abs}(\text{error}) \leq 0.1$  referencia y  $n = 2$

a) Pid

b)  $0.9 \text{ Pid} + 0.1 \text{ Self Tuning}$

2.2.5  $\text{Abs}(\text{error}) \leq 0.1$  referencia y  $n > 2$

a) Self Tuning

b)  $0.9 \text{ Self Tuning} + 0.1 \text{ Pid}$

2.2.6 Error  $> 0.1$  referencia

- a)  $\max(\text{Pid}, \text{Self Tuning})$
- b)  $0.9 \max(\text{Pid}, \text{Self Tuning}) + 0.1 \min(\text{Pid}, \text{Self Tuning})$

2.2.7 Error  $< -0.1$  referencia y  $\delta\text{error} < 0$

- a)  $\min(\text{Pid}, \text{Self Tuning})$
- b)  $0.9 \min(\text{Pid}, \text{Self Tuning}) + 0.1 \max(\text{Pid}, \text{Self Tuning})$

#### 4.4 SIMULACION DEL PROCESO.

La simulación del comportamiento y dinámica del proceso se realiza en el módulo SIMULAR.C, el que integra un conjunto de funciones que permiten obtener la respuesta del sistema.

Para proceder con la simulación se ingresa la función de transferencia del sistema, a través de la interfaz de usuario de la forma:

$$G(S) = \frac{b_n * S^n + b_{n-1} * S^{n-1} + b_{n-2} * S^{n-2} + \dots + b_0}{a_n * S^n + a_{n-1} * S^{n-1} + a_{n-2} * S^{n-2} + \dots + a_0} * e^{-T_r * S} \quad (1)$$

Además se ingresa el límite del actuador LIMITE, la referencia SET\_POINT, factor de variación de parámetros fac y tiempo de muestreo TMUESTREO.

La correspondiente ecuación diferencial del sistema, sin el tiempo de retardo es:

$$a_n Y^{(n)} + a_{n-1} Y^{(n-1)} + \dots + a_0 Y = b_n U^{(n)} + b_{n-1} U^{(n-1)} + \dots + b_0 U \quad (2)$$

La representación del sistema en el espacio de estado, utilizando la forma canónica controlable es:

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= X_3 \\ &\dots \\ &\dots \\ \dot{X}_{n-1} &= X_n \\ \dot{X}_n &= -\frac{a_{n-1}}{a_n} * X_n - \frac{a_{n-2}}{a_n} * X_{n-1} - \dots - \frac{a_0}{a_n} * X_1 - \frac{1}{a_n} * U \end{aligned} \quad (3)$$

Representando en forma matricial:

$$\dot{X} = A * X + B * U \quad (4)$$

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ 0 & 0 & 0 & \dots & 1 \\ -\frac{a_0}{a_n} & -\frac{a_1}{a_n} & -\frac{a_2}{a_n} & \dots & -\frac{a_{n-1}}{a_n} \end{bmatrix} ; \quad B = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ \frac{1}{a_n} \end{bmatrix}$$

Y la ecuación de salida es:

$$Y = b_0 * X_1 + b_1 * X_2 + b_2 * X_3 + \dots + b_{m-1} * X_m + b_m * X_{m-1} \quad (5)$$

En forma matricial:

$$Y = C * X \quad (6)$$

$$C = [b_0 \ b_1 \ b_2 \ b_3 \ \dots \ b_m]$$

Evaluando la ecuación (5) es posible obtener la respuesta de la planta, para lo cual es necesario resolver el sistema de ecuaciones diferenciales presentado por la fórmula (3).

Para esto se emplea el método de integración numérica de Runge Kutta de cuarto orden, el que se desarrolla en la función RUNGE() dentro del módulo SIMULAR.C, que se detalla más adelante.

#### 4.4.1 SELECCION DEL PERIODO DE MUESTREO.

Un parámetro muy importante en el proceso de simulación constituye el período de muestreo. Por esta razón, se deben realizar algunas consideraciones para seleccionar correctamente su valor, ya que éste será utilizado posteriormente en los procesos de integración numérica, discretización de la función de transferencia de la planta y cálculo de la acción de control.



Para estimar el período de muestreo es necesario conocer las raíces del polinomio característico  $P(S)$ .

En la Fig. 4.5 se muestra un diagrama del plano complejo que representa las raíces de  $P(S)$ .

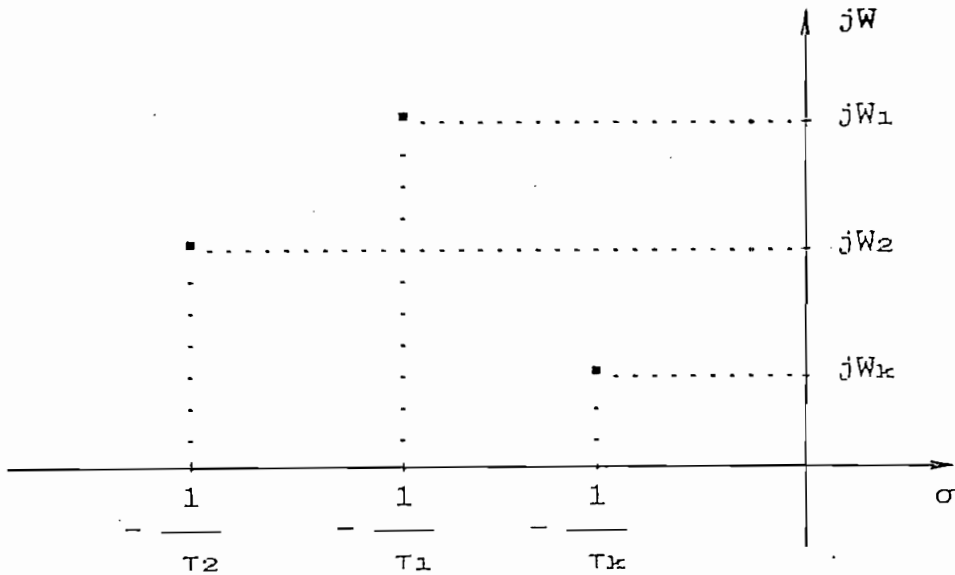


Fig. 4.5: Plano complejo con las raíces de  $P(S)$ .

Donde se tiene:

$$W_k = \frac{2\pi}{T_k} \quad (7)$$

Además, se cumple:

$$\tau_2 < \tau_1 < \dots < \tau_k \quad (8)$$

$$T_1 < T_2 < \dots < T_k$$

Si se denomina  $TR_{\min}$  a la mínima constante de tiempo dado por el eje real y  $TI_{\min}$  a la mínima constante de tiempo dado por el eje imaginario, se obtiene:

$$TR_{\min} = \tau_2 \quad (9)$$

$$TI_{\min} = T_1 \quad (10)$$

Entonces, el valor adecuado del período de muestreo  $T_{\text{MUESTREO}}$ , debe ser mucho menor que la menor constante de tiempo del sistema, es decir: [Ref. 27]

$$T_{\text{MUESTREO}} \ll TR_{\min} \quad (11)$$

y

$$T_{\text{MUESTREO}} \ll TI_{\min} \quad (12)$$

Tomando en cuenta estas consideraciones, se desarrolla el módulo RAICES.C, donde se incluye las funciones necesarias para obtener las raíces del polinomio característico y a partir de ellas estimar el período de muestreo.

El módulo CEXPER.C envía a este módulo el grado y coeficientes del polinomio característico  $P(S)$ , y la función RAIZPOLI2() calcula los ceros del polinomio utilizando el método de NEWTON RAPHSON.

Las raíces evaluadas son retornadas posteriormente al módulo principal, el cual vuelve a enviarlas a la función `tmin()` para evaluar la mínima constante de tiempo del sistema. Las funciones usadas en este módulo se presentan en la Fig. 4.6

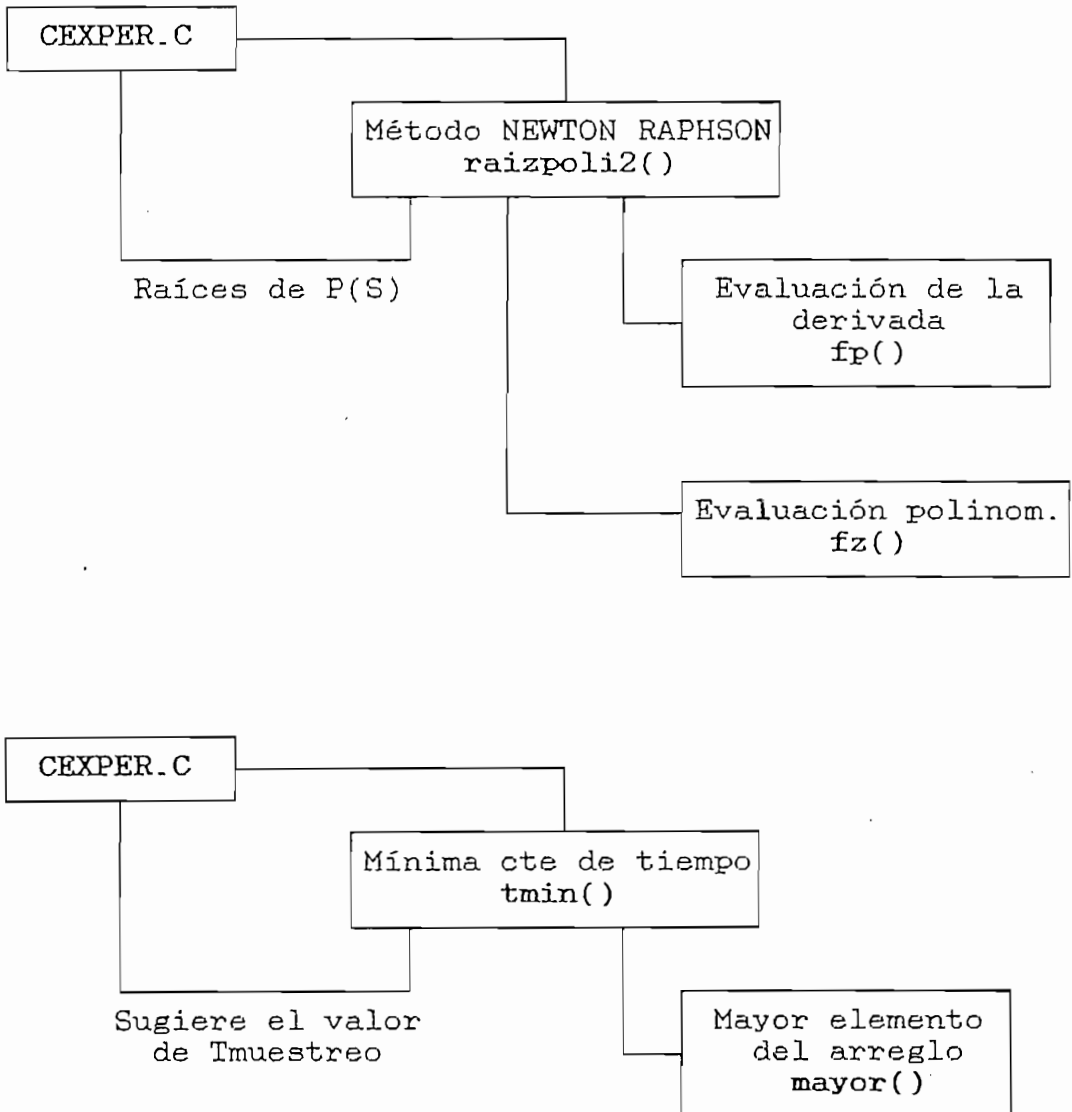


Fig. 4.6: Funciones del módulo RAICES.C .

#### 4.4.2 SIMULACION DE LA PLANTA.

En el módulo SIMULAR.C se incluyen las funciones necesarias para realizar la simulación de procesos lineales, con o sin retardo de transporte y variantes en el tiempo. Las funciones que forman este archivo son:

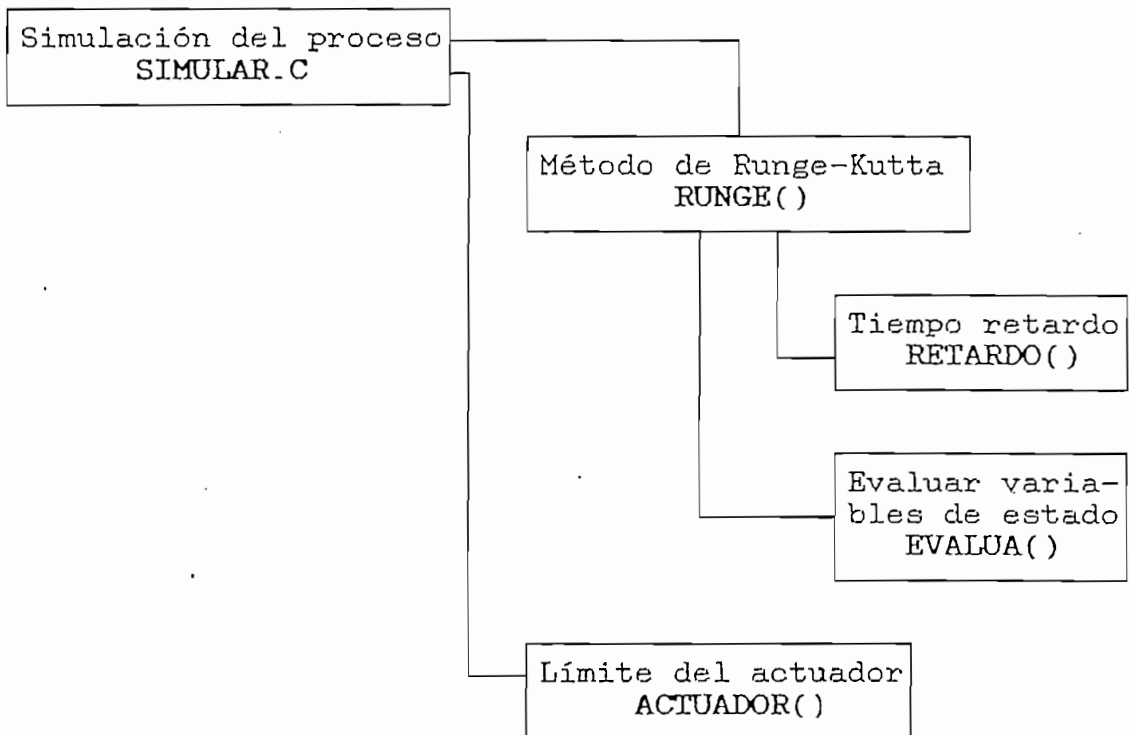


Fig. 4.7: Funciones del módulo SIMULAR.C.

##### 4.4.2.1 METODO DE INTEGRACION NUMERICA.

En la función RUNGE() se desarrolla el método de integración de Runge-Kutta de cuarto orden, cuyo algoritmo dice que para una ecuación diferencial definida por:

$$\frac{dx}{dt} = f(x, t) \quad (13)$$

Se aplica la fórmula recurrente: [Ref. 26]

$$X_{k+1} = X_k + \frac{(K_1 + 2 * K_2 + 2 * K_3 + K_4)}{6} \quad (14)$$

Donde:

$$\begin{aligned} K_1 &= hf(X_k, t_k) \\ K_2 &= hf\left(X_k + \frac{K_1}{2}, t_k + \frac{h}{2}\right) \\ K_3 &= hf\left(X_k + \frac{K_2}{2}, t_k + \frac{h}{2}\right) \\ K_4 &= hf(X_k + K_3, t_k + h) \end{aligned} \quad (15)$$

Para evaluar la respuesta de la planta en cada intervalo de muestreo, el módulo principal envía a la función RUNGE() los parámetros de la función de transferencia G(S) y la señal de error o respuesta del bloque de controles. En ella se hace una nueva llamada a la función EVALUA(), donde se traslada la función G(S) a su representación en el espacio de estado de la forma canónica controlable. Allí se evalúa las ecuaciones (3) considerando condiciones iniciales nulas.

Estas ecuaciones son desarrolladas en la función EVALUA(), de

la siguiente forma:

```

DESDE  i = 1  HASTA  n - 1
      dXdti = Xi+1;
FIN DEL LAZO
SUM = 0;
DESDE  i = 1  HASTA  n
      SUM = SUM + (ai/an) * Xi+1;
FIN DEL LAZO
dXdtn = U/an - SUM;

```

El cálculo de la respuesta del proceso Y, es evaluada en la función RUNGE(). La que corresponde a una combinación de sus estados y se desarrolla de la siguiente forma:

```

Y = 0;
DESDE  i = 1  HASTA  m
      Y = Y + bi * Xi+1;
FIN DEL LAZO

```

Un importante parámetro del algoritmo de Runge-Kutta es el paso de integración  $h$ , el cual debe ser mucho menor que el período de muestreo  $T_{\text{muestreo}}$ . En este caso se adopta la siguiente relación:

$$h = \frac{T_{\text{muestreo}}}{10} \quad (16)$$

De esta forma, para evaluar la respuesta de la planta en cada período de muestreo se realizan un total 10 iteraciones.

Adicionalmente, en este módulo se incluyen las ecuaciones para simular el cambio de parámetros de la planta.

Cuando se ingresan los datos de la función de transferencia de la planta a través de la interfaz de usuario, se indica si se desea variar los parámetros del proceso. Si la contestación es afirmativa se debe ingresar además el factor de variación con el cual se altera uno de los coeficientes de  $G(S)$ .

A continuación el módulo CEXPER.C fija banderas y variables para que la señal de referencia con la que se alimenta al sistema también varíe en el tiempo, como se indica a continuación:

```

SI      desea cambiar parámetros   Y      80 ≤ contador < 120
ENTONCES      Set point = 0
DE OTRO MODO  Set point = Referencia

```

Para el cambio de parámetros en la función RUNGE() se incluye la siguiente instrucción:

```

SI      desea cambiar parámetros   Y      contador = 120
ENTONCES       $a_{n-1} = fac * a_{n-1}$ 

```

#### 4.4.2.2 TIEMPO DE RETARDO.

El tiempo de retardo o tiempo muerto es un elemento muy común en procesos industriales, ya que se presenta en sistemas donde se realiza transporte de material o en procesos donde existen sensores situados a cierta distancia de la planta, causando de esta forma retardos en la medición de variables. [Ref. 28]

Desde el punto de vista del control, el tiempo de retardo es un factor de fase no mínima altamente indeseable, ya que contribuye a una mala operación del controlador que se aplique al proceso. Además si el tiempo de retardo es considerable, la planta puede llegar a tener un funcionamiento inestable.

Para efectos de simulación, el tiempo de retardo  $T_r$  que se ingresa como parte de la función de transferencia  $G(S)$ , es evaluado posteriormente en el módulo CEXPER.C, para representarlo como un múltiplo entero del período de muestreo. Como se indica en la ecuación (17).

$$T_{\text{RETARDO}} = \text{ENTERO} \left( \frac{T_r}{T_{\text{MUESTREO}}} \right) \quad (17)$$

La simulación del tiempo de retardo se realiza en la función RETARDO(), donde se define un arreglo que almacena las señales del controlador y dos puntero IP1 e IP2, los que están



separados entre sí por el valor del TRETARDO. [Ref. 8]

El valor de TRETARDO y señal de control, son enviados desde la función RUNGE() a RETARDO(), donde el valor del control es guardado en la localidad indicada por IP1, mientras que el puntero IP2 señala la localidad que contiene el dato que se debe utilizar para evaluar la respuesta de la planta. Finalmente la función retorna el valor señalado por IP2.

Arreglo con datos del control

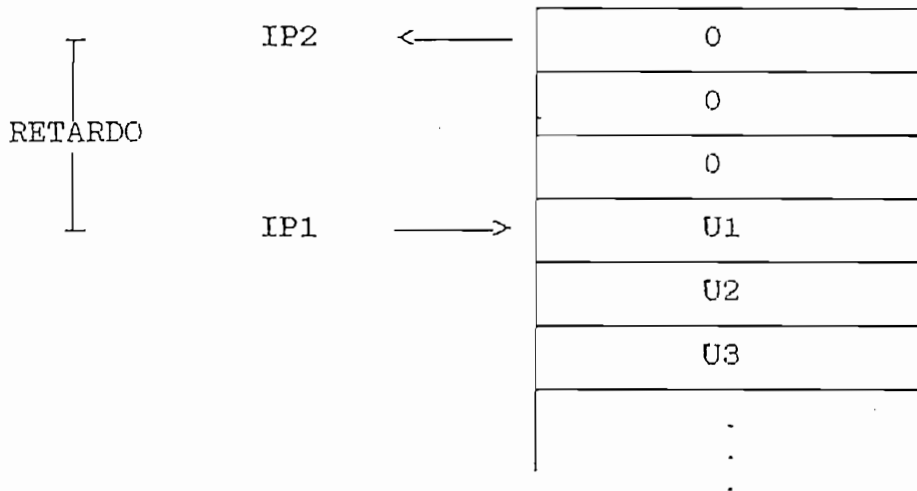


Fig. 4.8: Representación de la función RETARDO().

#### 4.4.2.3 ACTUADOR.

La mayoría de los sistemas de control tienen componentes que pueden variar su comportamiento sólo dentro de ciertos rangos. Por lo tanto para complementar la simulación del proceso se desarrolla la función ACTUADOR(), donde se utiliza la

alinealidad de saturación que permite modelar estos elementos tanto en su región lineal como en sus límites. La simulación del actuador de la planta se realiza de la siguiente forma:

SI	<i>señal de control</i> $\geq$ <i>límite del actuador</i>
ENTONCES	<i>señal de control</i> = <i>límite del actuador</i>
DE OTRO MODO SI	<i>señal de control</i> $\leq$ $-$ <i>límite del actuador</i>
ENTONCES	<i>señal de control</i> = $-$ <i>límite del actuador</i>
DE OTRO MODO	<i>utilice la misma señal de control</i>

#### 4.4.3 DISCRETIZACION DEL SISTEMA CONTINUO.

Para poder utilizar los algoritmos de los controles DEADBEAT y SELF TUNING, es necesario disponer de la función de transferencia discreta  $G(Z)$ .

$$G(Z) = \frac{b_1 * Z^{-1} + b_2 * Z^{-2} + b_3 * Z^{-3} + \dots + b_n * Z^{-n}}{1 + a_1 * Z^{-1} + a_2 * Z^{-2} + \dots + a_n * Z^{-n}} * Z^{-d} \quad (18)$$

Donde se cumple :

$$d = \frac{T_{RETARDO}}{T_{MUESTREO}} \quad (19)$$

Con este objeto se desarrolla el módulo DISCRETO.C, el que permite realizar la discretización del sistema continuo como se indica a continuación:

Definido un sistema continuo por las ecuaciones de estado :

$$\begin{aligned} \dot{X}(t) &= A * X(t) + B * U(t) \\ Y(t) &= C * X(t) \end{aligned} \quad (20)$$

Se debe obtener su representación discreta en el espacio de estado definido por:

$$\begin{aligned} X(k+1) &= A_D * X(k) + B_D * U(k) \\ Y(k) &= C_D * X(k) \end{aligned} \quad (21)$$

Lo que se consigue utilizando las siguientes ecuaciones:

[Ref. 27]

$$A_D = e^{A * T_{\text{MUESTRO}}} \quad (22)$$

$$B_D = \int_0^{T_{\text{MUESTRO}}} e^{A\alpha} d\alpha B \quad (23)$$

$$C_D = C \quad (24)$$

Para la resolver estas ecuaciones se emplea el siguiente algoritmo, donde I es la matriz identidad.

$$A_D = I + A * T_{MUESTREO} * \Psi \quad (25)$$

$$\Psi = I + \frac{A * T_{MUESTREO}}{2!} + \frac{A^2 * T_{MUESTREO}^2}{3!} + \dots \quad (26)$$

$$B_D = \Psi * T_{MUESTREO} * B \quad (27)$$

Considerando la ecuación (28), obtenemos la función de transferencia  $G(Z)$  incluyendo un elemento de retención de orden cero (zero order hold ZOH):

$$G(Z) = C_D * (ZI - A_D)^{-1} * B_D \quad (28)$$

Se aplica el algoritmo de Fadeer [Ref. 29] para obtener el término  $(ZI - A_D)^{-1}$ , utilizando las siguientes ecuaciones:

$$(ZI - A_D)^{-1} = \frac{Adj(ZI - A_D)}{|ZI - A_D|} \quad (29)$$

$$Adj(ZI - A_D) = F_1 * Z^{-1} + F_2 * Z^{-2} + \dots + F_n * Z^{-n} \quad (30)$$

$$|ZI - A| = 1 + \theta_1 * Z^{-1} + \dots + \theta_n * Z^{-n} \quad (31)$$

$$F_1 = I ; F_k = A_D * F_{k-1} + \theta_{k-1} * I \quad (32)$$

$$\theta_k = -\frac{tI(A_D * F_k)}{k} \quad (33)$$

$$tI(A_D) = \sum_1^n a_{11} \quad (34)$$

Reemplazando (29), (30) y (31) en (28) tenemos:

$$G(Z) = C_D * \frac{F_1 * Z^{-1} + F_2 * Z^{-2} + \dots + F_n * Z^{-n}}{1 + \theta_1 * Z^{-1} + \dots + \theta_n * Z^{-n}} * B_D \quad (35)$$

#### 4.5 ALGORITMOS DE CONTROL.

El bloque controlador está formado por los algoritmos de los controles: PID, DEADBEAT y SELF TUNING, cada uno de los cuales constituye una función desarrollada en el módulo CONTROL.C. La Fig. 4.9 indica la estructura empleada en este módulo.

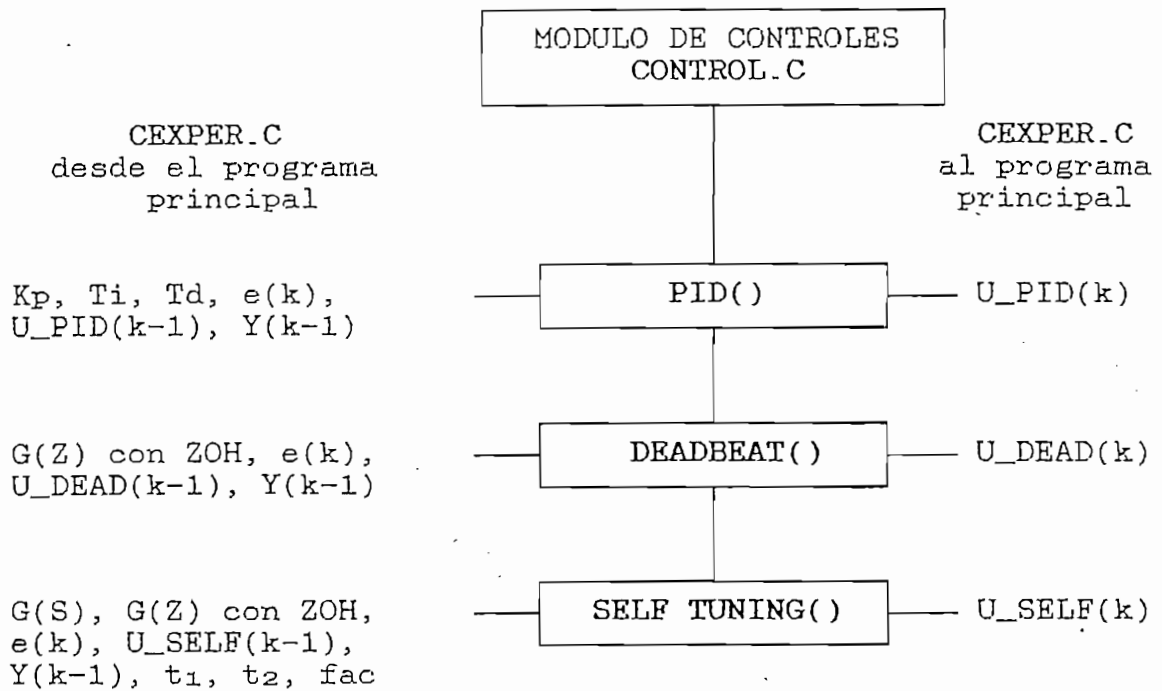


Fig. 4.9: Funciones del módulo CONTROL.C.

Donde:

U\_PID = Señal del control PID

U\_DEAD = Señal del control DEADBEAT

U\_SELF = Señal del control SELF TUNING

e = Error del sistema

Y = Respuesta del sistema

t<sub>1</sub> y t<sub>2</sub> = Polos asignados para el control SELF TUNING

fac = Factor de variación de a<sub>n-1</sub>

#### 4.5.1 CONTROL PID.

En la función PID() se desarrolla un controlador ampliamente usado a nivel industrial, el control PID. Este controlador proporciona un funcionamiento satisfactorio para una gran cantidad de procesos, considerando además que es muy conocido y fácil de implantarlo utilizando tecnología analógica o digital. El clásico algoritmo del control PID es:

$$u(t) = K_p * \left[ e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de}{dt} \right] \quad (36)$$

y su correspondiente transformada de Laplace es:

$$U(S) = K_p * \left[ 1 + \frac{1}{ST_i} + ST_d \right] E(S) \quad (37)$$

Donde:

$U$  es la acción de control

$E$  es el error (set point - salida)

$K_P$  es la ganancia proporcional

$T_i$  es el tiempo integral

$T_d$  es el tiempo derivativo

Este controlador tiene un integrador el cual elimina el error en estado estable del sistema en un tiempo proporcional a  $T_i$ . Desafortunadamente, el uso de un control integral en forma individual, lleva al sistema frecuentemente a la inestabilidad. Sin embargo la combinación de la acción proporcional e integral garantiza estabilidad en lazo cerrado cuando el valor de  $T_i$  es apropiadamente escogido.

La acción derivativa puede ser usada cuando se tiene un excesivo retardo en el proceso a controlar. En la práctica, este control es mucho menos empleado que otros ya que el ruido del proceso es indeseablemente amplificado en altas frecuencias.

En procesos donde el sobreimpulso debe ser evitado, es favorable el uso de la acción derivativa, la que actúa anticipadamente logrando que el valor de la señal de control llegue a ser negativa antes de que la respuesta alcance el valor de referencia.

Existen diferentes formas para desarrollar el controlador PID

tanto en tiempo continuo como discreto. En este caso se ha utilizado la configuración del control PID no interactuante con componente derivativa filtrada, que se muestra en la Fig. 4.10.

[Ref. 30]

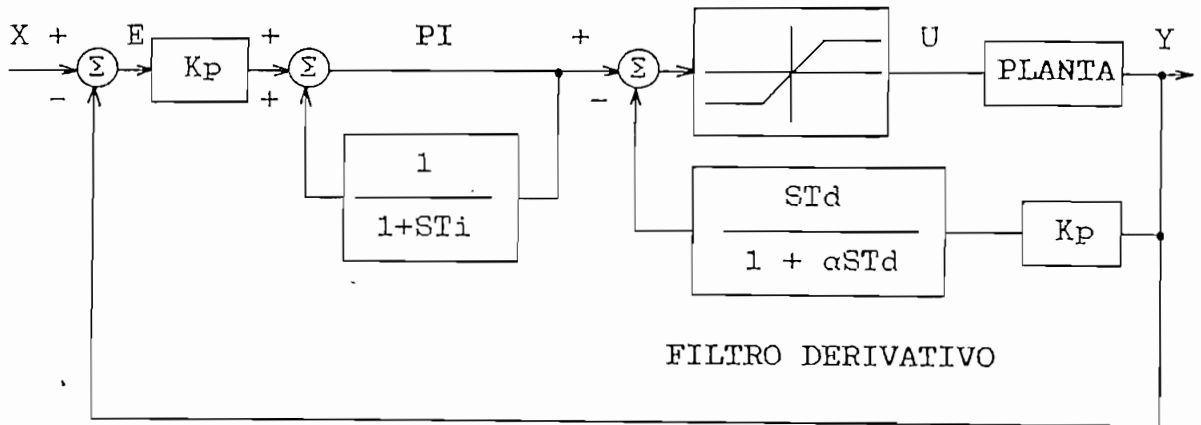


Fig. 4.10: Configuración del control PID no interactuante.

La señal de control se define como:

$$U(S) = K_p * \left[ \left( 1 + \frac{1}{ST_1} \right) E(S) - \frac{T_d}{1 + \alpha ST_d} Y(S) \right] \quad (38)$$

Donde:

$$\alpha = 0.1$$

Para aplicar el algoritmo del controlador PID no interactuante, se traslada la ecuación (38) a su correspondiente



representación discreta, utilizando el método de integración trapezoidal. Se aplica la siguiente relación conocida como transformada bilineal.

$$S = \frac{2(1-Z^{-1})}{T_{\text{MUESTREO}}(1+Z^{-1})} \quad (39)$$

Empleando la transformada bilineal al término proporcional-integral se tiene:

$$U_{PI}(Z^{-1}) = K' * \left[ 1 + \frac{(1-\beta)Z^{-1}}{1-Z^{-1}} \right] E(Z^{-1}) \quad (40)$$

o:

$$U_{PI}(Z^{-1}) = K' * E(Z^{-1}) + \frac{(1-\beta)}{1-\beta Z^{-1}} * U_{PI}(Z^{-1}) \quad (41)$$

Donde:

$$K' = K_D * \left( 1 + \frac{T_{\text{MUESTREO}}}{2T_i} \right) \quad (42)$$

$$\beta = \frac{2T_i - T_{\text{MUESTREO}}}{2T_i + T_{\text{MUESTREO}}} \quad (43)$$

Considerando:

$$INTEGRAL = \frac{(1-\beta)Z^{-1}}{1-\beta Z^{-1}} * U_{PI}(Z^{-1}) \quad (44)$$

Tenemos:

$$INTEGRAL(k) = \beta * INTEGRAL(k-1) + (1-\beta) * U_{PI}(k-1) \quad (45)$$

Reemplazando (44) en (41), se tiene que la salida para el control PI es:

$$U_{PI}(k) = K' * E(k) + INTEGRAL(k) \quad (46)$$

El término de la acción derivativa esta representada por:

$$D(S) = \frac{ST_d}{1 + \alpha ST_d} * Y(S) \quad (47)$$

Usando la transformación bilineal tenemos:

$$D(Z^{-1}) = \frac{2\Gamma(1-Z^{-1})}{1 + 0.2\Gamma + (1-0.2\Gamma)Z^{-1}} * Y(Z^{-1}) \quad (48)$$

Donde:

$$\Gamma = \frac{T_d}{T_{\text{MUESTREO}}} \quad (49)$$

La ecuación de diferencias que se obtiene de la ecuación (48) es:

$$(50) \quad D(k) = \frac{0.2 * \Gamma}{1 + 0.2 * \Gamma} * D(k-1) + \frac{2 * \Gamma}{1 + 0.2 * \Gamma} * [Y(k) - Y(k-1)]$$

Finalmente la respuesta del controlador PID es:

$$U_{\text{PID}}(k) = U_{\text{PI}}(k) - K' * D(k) \quad (51)$$

Para obtener los parámetros  $K_p$ ,  $T_i$  y  $T_d$  del controlador se puede aplicar las técnicas de ajuste de Ziegler-Nichols (1942) o las reglas de margen de fase de Åström. [Ref. 30]

Es importante señalar que al utilizar una aproximación discreta en el tiempo de un controlador continuo, el intervalo de muestreo es un factor que debe cumplir ciertas condiciones adicionales.

Generalmente, se trata que el valor del  $T_{\text{MUESTREO}}$  sea lo más

pequeño posible, pero pueden ocurrir algunas dificultades numéricas que afectan el comportamiento del controlador. En muchos casos, intervalos de muestreo muy pequeños proporcionan resultados aceptables, particularmente si se utiliza el control PI. Sin embargo, si la acción derivativa es utilizada se requiere que el valor de  $T_d$  mantenga cierta relación con el  $T_{MUESTREO}$  para igualar las constantes de tiempo de la planta. Para el efecto se sugiere utilizar la siguiente regla :

[Ref. 33]

$$\frac{T_{MUESTREO}}{T_d} \approx 0.1 - 0.5 \quad (52)$$

El pseudocódigo del algoritmo PID desarrollado en Quick C es el siguiente:

```

FUNCION: PID()
  Uin = U(k-1);           Desde el programa principal
  error = E(k-1);        Desde el programa principal
  Yin = Y(k-1);           Desde el programa principal
  Kr = KP * (1 + TMUESTREO/(2 * Ti));
  β = (2 * Ti - TMUESTREO)/(2 * Ti + TMUESTREO);
  INTEGRAL = β * INTEGRAL + (1 - β) * Uin;
  Upi = Kr * error + INTEGRAL;
  Γ = Td/TMUESTREO;
  D1 = (0.2 * Γ - 1)/(1 + 0.2 * Γ);
  D2 = (2 * Γ)/(1 + 0.2 * Γ);

```

```

DERIVATIVO = D1 * DERIVATIVO + D2 * (Yin - Yold);
Upid = Upi - K' * DERIVATIVO;
Yold = Yin;
RETORNA Upid;           Al programa principal

```

#### 4.5.2 CONTROL DEADBEAT.

En la función DEADBEAT() se desarrolla una estrategia de control, que tiene la propiedad de llevar todos los estados del sistema a cero en  $n$  pasos cuando existe un cambio en la referencia, donde  $n$  es el orden del sistema. Este tipo de controlador se denomina Control Deadbeat. [Ref. 33]

El algoritmo del controlador deadbeat tiene solamente un parámetro de diseño, el período de muestreo. Si el error del sistema se hace cero en  $n$  períodos de muestreo, el tiempo de establecimiento es por lo tanto igual a  $n(T_{\text{MUESTREO}})$ .

Sin embargo el valor de  $T_{\text{MUESTREO}}$  podría influenciar notablemente en la magnitud de la señal de control, ya que cuando se selecciona períodos de muestreo muy pequeños la variable manipulada podría ser extremadamente grande. Por lo tanto, es importante seleccionar cuidadosamente el valor del muestreo cuando se usa este controlador.

El control deadbeat es considerado un controlador de cancelación, ya que ubica los polos del sistema en el origen

del plano  $z$ .

La ecuación característica para el sistema en lazo cerrado es:

$$1 + G_c(Z)G_p(Z) = Z^n = 0 \quad (53)$$

Esta propiedad hace que el controlador deadbeat sea empleado solamente en sistemas de control digital. Siendo imposible desarrollar controladores deadbeat continuos, ya que la cancelación de polos es un algoritmo matemático que no se puede conseguir con la implantación de un sistema analógico.

La Fig. 4.11 muestra el sistema de control ha desarrollar:

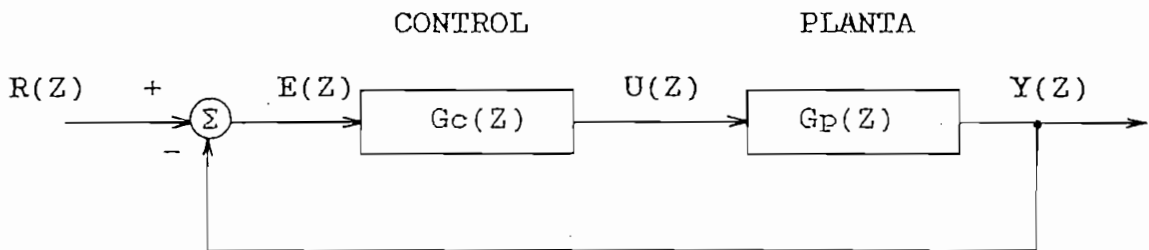


Fig. 4.11: Diagrama de bloques del lazo de control deadbeat.

Donde:

$G_p(Z)$  es la función de transferencia discreta del proceso a controlar incluyendo un ZOH.

$G_c(Z)$  es el controlador de cancelación.

$$G_p(Z) = \frac{b_1 * Z^{-1} + b_2 * Z^{-2} + b_3 * Z^{-3} + \dots + b_n * Z^{-n}}{1 + a_1 * Z^{-1} + a_2 * Z^{-2} + \dots + a_n * Z^{-n}} * Z^{-d} \quad (54)$$

$$d = \frac{T_{\text{RETARDO}}}{T_{\text{MUESTREO}}}$$

Se considera una entrada paso unitaria  $R(Z)$ , su correspondiente transformada  $Z$  es:  $R(k) = 1$  para  $k = 0, 1, 2, \dots$

$$R(Z) = \frac{1}{1 - Z^{-1}} \quad (55)$$

Para un retardo  $d = 0$ , los requisitos para un tiempo de ajuste mínimo es:

$$Y(k) = R(k) = 1 \quad \text{para } k \geq n \quad (56)$$

$$U(k) = U(n) \quad \text{para } k \geq n$$

Si no existe retardo:  $d \neq 0$

$$Y(k) = R(k) = 1 \quad \text{para } k \geq n \quad (57)$$

$$U(k) = U(n) \quad \text{para } k \geq n$$

La función de transferencia en lazo cerrado de la Fig. 4.11 es:

$$G(Z) = \frac{Y(Z)}{R(Z)} = \frac{G_c(Z) G_p(Z)}{1 + G_c(Z) G_p(Z)} \quad (58)$$

El controlador de cancelación está representado por:

$$G_c(Z) = \frac{1}{G_p(Z)} + \frac{G(Z)}{1 - G(Z)} \quad (59)$$

$$G_c(Z) = \frac{U(Z)}{E(Z)} = \frac{q_0 + q_1 * Z^{-1} + q_2 * Z^{-2} + \dots + q_n * Z^{-n}}{1 - p_1 * Z^{-(1+d)} - \dots - p_n * Z^{-(n+d)}} \quad (60)$$

y la ecuación característica llega a ser:

$$Z^{(n+d)} = 0 \quad (61)$$

Los parámetros de este controlador son obtenidos a partir de los coeficientes de la función de transferencia discreta de la planta.

$$q_0 = \frac{1}{b_1 + b_2 + \dots + b_n} = \frac{1}{\sum b_i} = U(0) \quad (62)$$



$$\begin{aligned}
 q_1 &= a_1 * q_0 ; p_1 = b_1 * q_0 \\
 q_2 &= a_2 * q_0 ; p_2 = b_2 * q_0 \\
 &\dots \qquad \qquad \dots \\
 &\dots \qquad \qquad \dots \\
 q_n &= a_n * q_0 ; p_n = b_n * q_0
 \end{aligned}
 \tag{63}$$

La ecuación de diferencias del controlador que se obtiene de la ecuación (60) es:

$$\begin{aligned}
 U(k) &= p_1 * U(k-1-d) + p_2 * U(k-2-d) + \dots + p_n * U(k-n-d) \\
 &\quad + q_0 * e(k) + q_1 * e(k-1) + \dots + q_n * e(k-n)
 \end{aligned}
 \tag{64}$$

Desarrollando estas ecuaciones para trabajar dentro del ambiente del Quick C tenemos el siguiente pseudocódigo:

```

FUNCION: DEADBEAT()
    U(k-1) = Uin; Desde el programa principal
    E(k) = error; Desde el programa principal
    DESDE i = 1 HASTA n
        q0 = 1/Σ bi;
    FIN DEL LAZO
    SUM = 0;
    DESDE i = 1 HASTA n
        qi = ai * q0;

```

```

    pi = bi * q0;
    SUM = SUM + pi * U(k-i-d) + qi * E(k-i);
FIN DEL LAZO
U(k) = SUM + q0 * E(k);
RETORNA U(k); Al programa principal

```

#### 4.5.3 CONTROL SELF TUNING.

En los últimos años las técnicas de control adaptivo han tenido gran impacto industrial, por lo que se considera de significativo beneficio incluir un controlador con propiedades de ajuste automático en el módulo CONTROL.C. Se selecciona con este objeto al control SELF TUNING, el que se desarrolla en la función SELF().

El propósito principal del control o regulador SELF TUNING, es automatizar los procedimientos de identificación y diseño de la ley de control del sistema. El diagrama de bloques del regulador SELF TUNING utilizado se muestra en la Fig. 4.12.

[Ref. 33]

Este regulador que se denomina SELF TUNING por facilitar el ajuste de sus propios parámetros, está formado por dos lazos como se observa en la Fig. 4.12: un lazo interno que consiste de la planta y un controlador lineal con realimentación y un lazo para el ajuste de parámetros.

El ajuste de parámetros está compuesto de dos bloques: un bloque que realiza la estimación de los coeficientes de un modelo del proceso y otro bloque que calcula los parámetros del regulador a partir de su modelo.

Para la identificación del proceso existen algunas técnicas, siendo la estimación de los mínimos cuadrados recursivos una de las más comunes. De igual forma el regulador SELF TUNING es muy flexible con respecto al procedimiento empleado para el diseño de la ley de control. En este caso se selecciona el método de asignación de polos de la función de transferencia en lazo cerrado, con lo que se asegura la estabilidad y una adecuada respuesta transitoria.

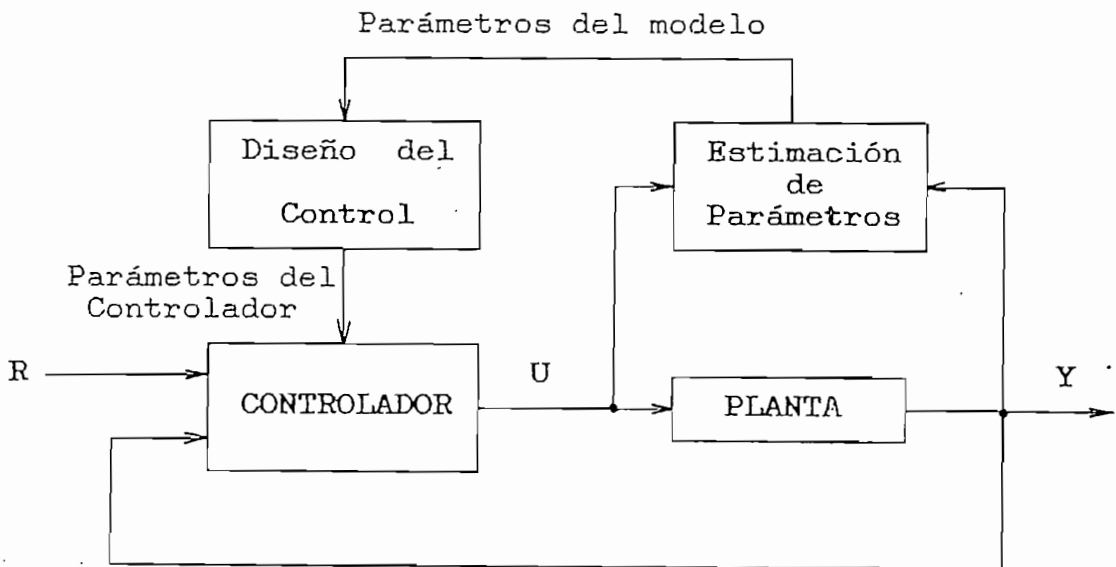


Fig. 4.12: Diagrama de bloques del control SELF TUNING.

## 4.5.3.1 IDENTIFICACION DEL SISTEMA.

Se considera que el proceso a controlar puede ser modelado por su función de transferencia discreta, expresada de la siguiente forma:

$$Y(k) = \frac{B(Z^{-1}) \cdot Z^{-d}}{1+A(Z^{-1})} U(k) \quad (65)$$

Donde  $U(k)$ ,  $Y(k)$  para  $k = 0, \pm 1, \pm 2, \dots$  son la salida del controlador y la respuesta del sistema muestreado respectivamente.

La ecuación (65) que representa a la misma función de transferencia  $G(Z)$  expresada en la ecuación (18), está caracterizada por los polinomios  $A(Z^{-1})$  y  $B(Z^{-1})$  que representan la dinámica de entrada/salida del sistema a controlar.

$$B(Z^{-1}) = b_1 * Z^{-1} + \dots + b_n * Z^{-n} \quad (66)$$

$$A(Z^{-1}) = a_1 * Z^{-1} + \dots + a_n * Z^{-n}$$

La identificación del proceso supone determinar los coeficientes  $(a_1, b_1)$  de los polinomios  $A(Z^{-1})$  y  $B(Z^{-1})$  en cada período de muestreo. Esta tarea es realizada en el módulo IDENTIFI.C, en el que se desarrolla el algoritmo de los mínimos

cuadrados recursivos que se describe a continuación: [Ref. 35]

Sea la ecuación de diferencias que describe la dinámica del sistema:

$$Y(k) = -a_1*Y(k-1) - \dots - a_n*Y(k-n) + b_1*U(k-1-d) + \dots + b_n*U(k-n-d) \quad (67)$$

La que puede ser escrita como:

$$Y(k) = X(k) * \theta^T \quad (68)$$

Donde:

$X(k)$  es una secuencia de mediciones de entrada y salida a la planta

$\theta^T$  son los parámetros a identificar

Descritos por:

$$X(k) = [-Y(k-1), \dots, -Y(k-n); U(k-1-d), \dots, U(k-n-d)] \quad (69)$$

$$\theta^T = [a_1, \dots, a_n; b_1, \dots, b_n] \quad (70)$$

El algoritmo de los mínimos cuadrados recursivos estima los valores de  $\bar{\theta}$  en el instante  $k$ , aplicando las siguientes ecuaciones:

$$\bar{\theta}(k) = \bar{\theta}(k-1) + L(k) * e(k) \quad (71)$$

$$e(k) = Y(k) - X(k) * \bar{\theta}(k-1) \quad (72)$$

$$L(k) = \frac{P(k-1) * X^T(k)}{1 + X(k) * P(k-1) * X^T(k)} \quad (73)$$

$$P(k) = [I - L(k) * X(k)] * P(k-1) \quad (74)$$

Cuando  $k = 1$  los valores de  $P(0)$  y  $\bar{\theta}(0)$ , deben ser especificados. Así para  $\bar{\theta}(0)$  se puede asignar valores iniciales iguales a cero, mientras que la matriz  $P(0)$  debe ser una matriz diagonal con valores suficientemente grandes, para asegurar una convergencia rápida a los parámetros estimados.

El valor de  $P(0)$  generalmente se escoge como:

$$P(0) = \alpha I \quad (75)$$

Donde  $\alpha$  puede ser igual a 1000, 10000, 100000.

Posteriormente, para  $k > 1$  el algoritmo se aproxima lentamente hacia los parámetros, al dividir el valor de  $P(k-1)$  por un

factor conocido como Factor de olvido ( $\delta$ ).

$$P(k-1) = \frac{P(k-1)}{\delta} \quad (76)$$

Donde  $\delta$  puede tener valores entre 0 y 1. El factor de olvido controla la velocidad con el cual los parámetros estimados pueden adaptarse a cambios en el sistema. Si  $\delta$  disminuye la velocidad de adaptación aumenta perdiendo certeza en los resultados, además valores menores a 0.95 son raramente usados.

#### 4.5.3.2 DISEÑO DEL CONTROLADOR.

La técnica empleada para el diseño de la ley de control, se basa en una forma particular de operación del control SELF TUNING que es la asignación de polos.

Mediante este procedimiento se fijan los polos de la función de transferencia en lazo cerrado, los que a través de un algoritmo permiten calcular la variable manipulada que controlará el proceso.

La Fig. 4.13 muestra el diagrama de bloques bajo el cual se realiza el cálculo del controlador. [Ref. 35]

## CONTROL

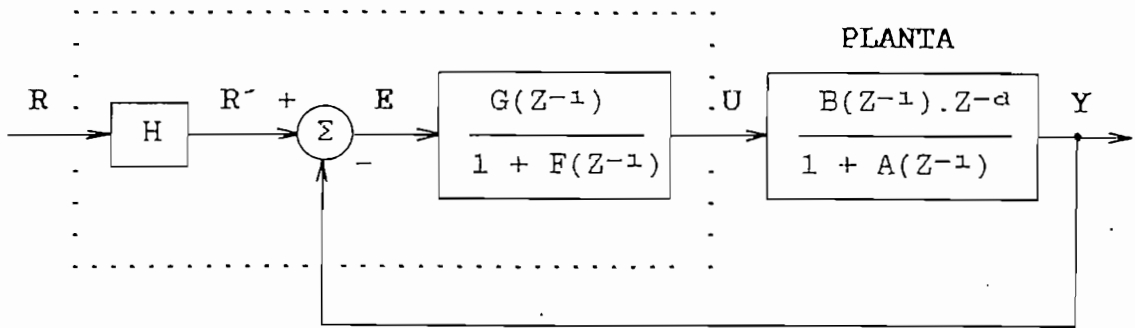


Fig. 4.13: Configuración del controlador a diseñar.

Donde:

H es un precompensador que lleva la respuesta del sistema al set point y  $F(Z^{-1})$ ,  $G(Z^{-1})$  están definidos por:

$$F(Z^{-1}) = f_1 * Z^{-1} + \dots + f_{n_f} * Z^{-n_f} \quad (77)$$

$$G(Z^{-1}) = g_0 + g_1 * Z^{-1} + \dots + g_{n_g} * Z^{-n_g} \quad (78)$$

En cada período de muestreo, los coeficientes del modelo de la planta ( $a_i$ ,  $b_i$ ) son estimados y transferidos al bloque de diseño del control. En este bloque se calcula los coeficientes del regulador ( $f_i$ ,  $g_i$ ), los que posteriormente son enviados al controlador para calcular la ley de control U.



Aplicando las reglas de reducción la Fig. 4.13 puede ser representada por:

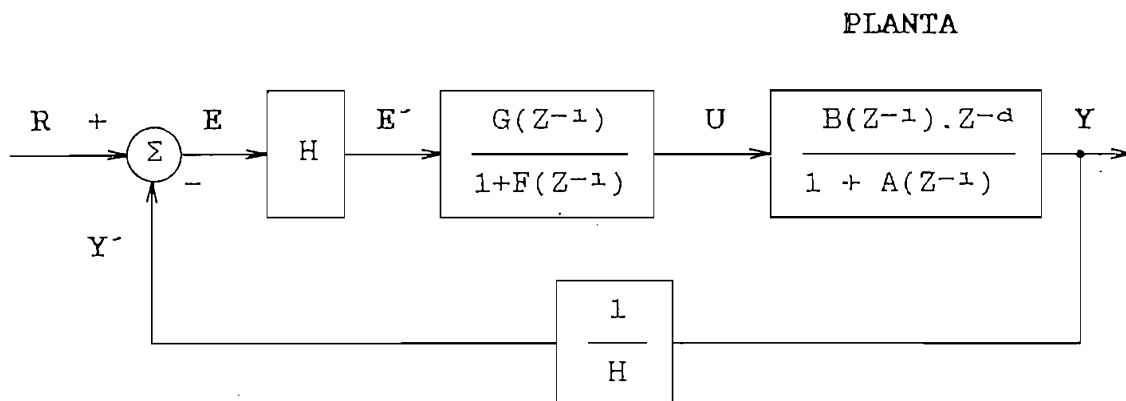


Fig. 4.14: Configuración equivalente del controlador.

La ecuación de salida se puede expresar como:

$$Y(k) = \frac{B(Z^{-1}) \cdot Z^{-d}}{1+A(Z^{-1})} * U(k) \quad (79)$$

Y la señal de control es:

$$(80) \quad U(k) = \frac{G(Z^{-1})}{1+F(Z^{-1})} * E'(k)$$

Donde:

$$E'(k) = H * E(k) \quad (81)$$

$$E(k) = R(k) - \frac{Y(k)}{H} \quad (82)$$

Sustituyendo las ecuaciones (80), (81), (82) en (79) y simplificando la notación tenemos:

$$[(1+A)(1+F) + Z^{-d} * B * G] Y(k) = [Z^{-d} * B * G * H] R(k) \quad (83)$$

Para especificar la posición de los polos del sistema en lazo cerrado, se iguala la ecuación (83) a  $T(Z^{-1})$ , que representa el polinomio característico definido en el plano  $Z$  cuyas raíces son los polos asignados al sistema.

$$T(Z^{-1}) = 1 + t_1 * Z^{-1} + t_2 * Z^{-2} \quad (84)$$

Con lo que se obtiene la siguiente ecuación:

$$[(1+A)(1+F) + Z^{-d} B G] = T \quad (85)$$

Desarrollando la ecuación (85) se genera un sistema de ecuaciones lineales, que es expresado en la siguiente forma matricial:

$$\begin{bmatrix} 1 & 0 & \dots & 0 & | & b_1 & 0 & \dots & 0 \\ a_1 & 1 & \dots & 0 & | & b_2 & b_1 & \dots & 0 \\ a_2 & a_1 & \dots & 0 & | & b_3 & b_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & | & \vdots & \vdots & \dots & \vdots \\ a_n & a_{n-1} & \dots & \dots & | & b_n & b_{n-1} & \dots & \dots \\ 0 & a_n & \dots & \dots & | & 0 & b_n & \dots & \dots \\ 0 & 0 & \dots & a_n & | & 0 & 0 & \dots & b_n \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n_f} \\ g_0 \\ g_1 \\ \vdots \\ g_{n_g} \end{bmatrix} = \begin{bmatrix} t_1 - a_1 \\ t_2 - a_2 \\ \vdots \\ \vdots \\ -a_n \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (86)$$

Al resolver este sistema de ecuaciones se obtiene los coeficientes de los polinomios  $F(Z^{-1})$  y  $G(Z^{-1})$ . Pero para garantizar una única solución, es necesario que el orden  $n_f$  y  $n_g$  cumplan la siguiente condición:

$$n_f = n + d - 1 \quad (87)$$

$$n_g = n - 1 \quad (88)$$

Bajo esta condición el orden del sistema de ecuaciones (M) es:

$$M = n_f + n_g + 1 \quad (89)$$

Es importante señalar que debido a limitaciones de programación el número máximo de ecuaciones a resolver es 10. Por lo tanto se debe considerar:

$$M_{\max} = 10 \quad (90)$$

Sustituyendo la ecuación (90) en (89) se tiene una nueva condición que se debe respetar al momento de ingresar los datos de la función de transferencia  $G(S)$ .

$$2n+d \leq 11 \quad (91)$$

Para resolver el sistema de ecuaciones (86), se emplea el método de reducción de Gauss que se desarrolla en la función ECLINR().

Una vez calculados los parámetros del controlador, se procede a evaluar la ley de control correspondiente. Reemplazando la ecuación (81) en (80) se tiene:

$$U(k) = \frac{G(Z^{-1})}{1+F(Z^{-1})} * H * E(k) \quad (92)$$

La ecuación de diferencias que permite obtener la señal de control es:

$$U(k) = -\sum_{i=1}^{n_f} f_i * U(k-i) + \sum_{i=0}^{n_g} g_i * H * E(k-i) \quad (93)$$

El valor del precompensador H, se calcula considerando que la respuesta del proceso en estado estable debe seguir a la señal de referencia o set point.

La función de transferencia en lazo cerrado de la Fig. 4.14 es:

$$FT(k) = \frac{H \cdot G(Z^{-1}) B(Z^{-1})}{[1 + F(Z^{-1})] [1 + A(Z^{-1})] + G(Z^{-1}) B(Z^{-1})} \quad (94)$$

El teorema del valor final definido en el plano Z dice:

$$Y(\infty) = \lim_{z \rightarrow 1} [(z-1) \cdot R(z^{-1}) \cdot FT(z^{-1})] \quad (95)$$

Aplicando (95) y simplificando notación tenemos:

$$Y(\infty) = \lim_{z \rightarrow 1} \left[ \frac{G \cdot B \cdot SET\_POINT \cdot H \cdot Z}{(1+F)(1+A) + G \cdot B} \right] \quad (96)$$

Pero se desea que:

$$Y(\infty) = SET\_POINT \quad (97)$$

Igualando (97) a (96) y despejando la variable de interés H, se obtiene:

$$H = \lim_{z \rightarrow 1} \left[ \frac{(1+F) * (1+A) + G * B}{G * B} \right] \quad (98)$$

Finalmente el valor de H se obtiene al aplicar la siguiente ecuación:

$$H = \frac{\left(1 + \sum_{i=1}^{n_f} f_i\right) * \left(1 + \sum_{i=1}^n a_i\right) + \sum_{i=1}^{n_g} g_i * \sum_{i=1}^n b_i}{\sum_{i=1}^{n_g} g_i * \sum_{i=1}^n b_i} \quad (99)$$

#### 4.5.3.3 ASIGNACION DE POLOS.

Para el diseño del controlador es necesario indicar la localización de los polos en lazo cerrado, lo que se consigue especificando los coeficientes del polinomio  $T(Z^{-1})$  de la ecuación (84). [Ref. 35]

En la práctica un polinomio  $T(Z^{-1})$  de primer o segundo orden, permite determinar las características globales con las que se espera responda el sistema. Además se considera las características del proceso a controlar para seleccionar el tipo de respuesta al aplicar el control SELF TUNING.

Si se trata de un sistema de primer orden, el polinomio  $T(Z^{-1})$  es:

$$T(Z^{-1}) = 1 + t_1 * Z^{-1} \quad (100)$$

Donde:

$$t_1 = -e^{-\frac{T_{MOMENTARIO}}{\tau}} \quad (101)$$

Donde:

$\tau$  es la constante de tiempo deseada en lazo cerrado.

Para seleccionar su valor se debe recordar que el tiempo de establecimiento  $t_s$  para sistemas de primer orden es aproximadamente:

$$t_s \approx 4\tau \quad (102)$$

Asignando un valor conveniente a  $t_s$ , de la ecuación (102) se despeja el parámetro  $\tau$  y aplicando (101) se determina el coeficiente  $t_1$ .

Si el sistema es de segundo u orden superior, el polinomio  $T(Z^{-1})$  queda determinado por:

$$T(Z^{-1}) = 1 + t_1 * Z^{-1} + t_2 * Z^{-2} \quad (103)$$

Donde:

$$t_1 = -2 * e^{-\epsilon W_n T_{\text{MUESTREO}}} * \cos(W_n T_{\text{MUESTREO}} \sqrt{1 - \epsilon^2}) \quad (104)$$

$$t_2 = e^{-2\epsilon W_n T_{\text{MUESTREO}}} \quad (105)$$

Estas ecuaciones se obtienen al convertir los polos asignados en el plano S, a su respectiva representación en el plano Z.

Los parámetros:

$\epsilon$  : coeficiente de amortiguamiento

$W_n$  : frecuencia natural

Caracterizan la respuesta transitoria para sistemas de segundo orden. Estos parámetros deben ser escogidos de tal forma que garanticen una respuesta rápida con un sobreimpulso aceptable.

El valor del sobreimpulso  $M_p$  de la señal de salida está dado por:

$$M_p\% = 100 * e^{-\frac{\epsilon\pi}{\sqrt{1-\epsilon^2}}} \quad (106)$$

A partir de esta expresión se determina que el valor de  $\epsilon$  puede variar entre 0.5 a 0.8, para mantener una respuesta del sistema aceptable y un sobreimpulso  $M_p\% < 20\%$ .

Para determinar el valor de  $W_n$  se considera la ecuación (107),



donde se especifica el tiempo de establecimiento con el que se desea responda la planta.

$$ts = \frac{4}{eW_n} ; \text{ (Criterio del 2\%)} \quad (107)$$

#### 4.5.3.4 PSEUDOCODIGO DEL ALGORITMO SELF TUNING.

El pseudocódigo del algoritmo SELF TUNING que define la operación de la función SELF(), es el siguiente:

```

FUNCION: SELF()
  U(k-1) = Uin;           Desde el programa principal
  E(k) = H * error;      Desde el programa principal
  nf = n + d - 1;
  ng = n - 1;
  M = nf + ng + 1;
  SI   k = 120   Y   cambiar parámetros
  ENTONCES   an-1 = fac * an-1;
              LLAMAR  Función DISC();   Discretizar nuevo sistema
              LLAMAR  Función IDENT();   Identificar modelo del proceso
              LLAMAR  Función MATRIZ();   Formar sistema de ecuaciones que
                                              definen los coeficientes del
                                              controlador
              LLAMAR  Función ECLINR();  Resolver sistema de ecuaciones
  aux1 = 0;
  aux2 = 0;

```

```

aux3 = 0;
aux4 = 0;
DESDE i = 1 HASTA nf
    aux1 = aux1 + fi;
FIN DEL LAZO
DESDE i = 1 HASTA n
    aux2 = aux2 + an;
FIN DEL LAZO
DESDE i = 1 HASTA ng
    aux3 = aux3 + fi;
FIN DEL LAZO
DESDE i = 1 HASTA n
    aux4 = aux4 + bn;
FIN DEL LAZO
H = [(1 + aux1) * (1 + aux2) + aux3 * aux4] / (aux3 * aux4);
SUM1 = 0;
DESDE i = 1 HASTA nf
    SUM1 = SUM1 - fi * U(k-i);
FIN DEL LAZO
SUM2 = 0;
DESDE i = 1 HASTA ng
    SUM2 = SUM2 + gi * E(k-i);
FIN DEL LAZO
U(k) = SUM1 + SUM2;
RETORNA U(k);           Al programa principal

```

#### 4.6 CONTROLADOR EXPERTO.

El sistema de control experto basado en conocimiento, es coordinado por el programa principal CEXPER.C. Este realiza llamadas a las respectivas funciones para calcular las señales correspondientes del bloque controlador (PID, DEADBEAT, SELF TUNING) y utilizando la base de conocimiento desarrollada en el módulo BASE.C, decide la acción de control que actuará sobre el proceso en cada período de muestreo.

El conocimiento está representado por dos funciones EXPERTO1() y EXPERTO2(), como se indicó en la sección 3.3. Donde la función EXPERTO1() reúne las reglas de producción que permiten escoger la mejor respuesta del bloque controlador mientras que en la función EXPERTO2(), se definen las reglas para asignar factores de ponderación a cada señal de control y así calcular una acción de control resultante.

En la Fig. 4.15 se indica el diagrama de bloques bajo el cual trabaja el controlador experto, donde:

U\_PID es la salida del control PID

U\_DEAD es la salida del control DEADBEAT

U\_SELF es la salida del control SELF TUNING

e es el error

$\delta e$  es la derivada del error

SET\_POINT es la referencia que ingresa al sistema

n es el orden del sistema

Programa principal

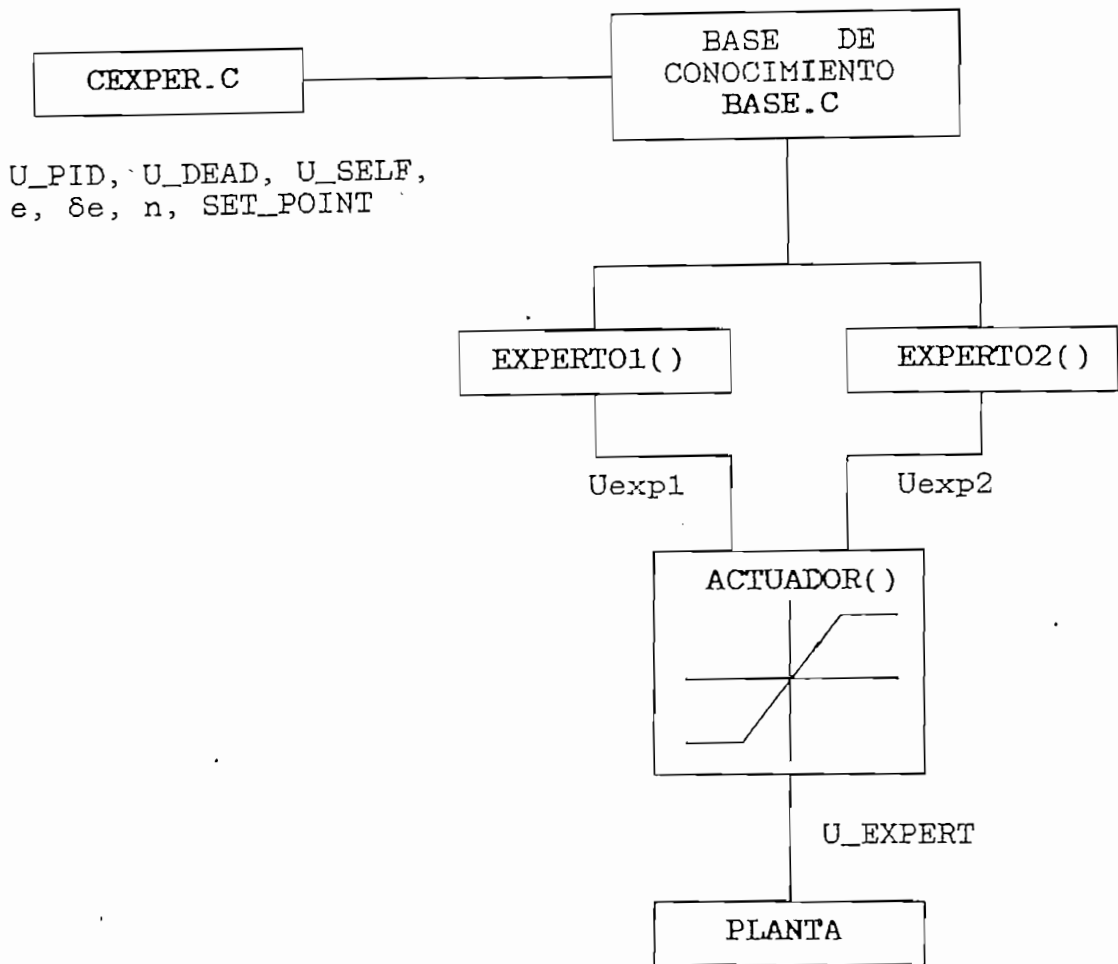


Fig. 4.15: Funciones y módulos utilizados por el controlador experto.

## CAPITULO V

## CASOS DE ESTUDIO

## 5.1 CONTROL DE TEMPERATURA CON RETARDO DE TRANSPORTE.

La Fig 5.1 muestra el sistema térmico a controlar en el que circula aire caliente para mantener constante la temperatura de la cámara. [Ref. 38]

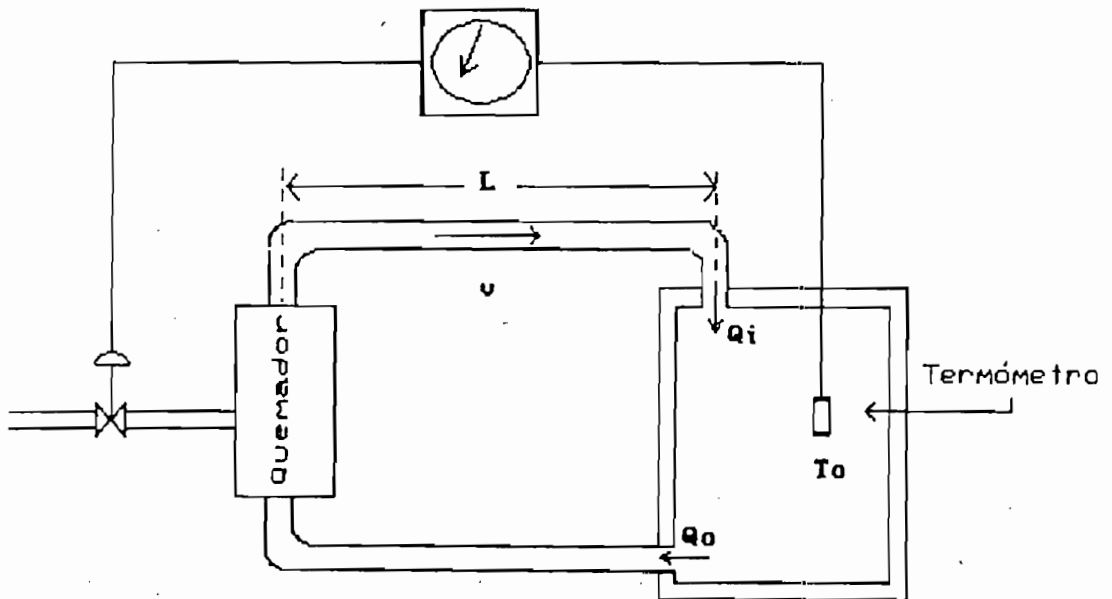


Fig. 5.1: Sistema térmico con retardo de transporte.

En este sistema el elemento de medida está ubicado en el retorno del flujo a una distancia  $L$  (m) de la caldera, la velocidad del aire es  $v$  (m/seg) y han de transcurrir  $T_r = L/v$  (seg) antes que cualquier modificación en la temperatura de la caldera sea registrada por el termómetro. Este retardo producido entre la medición y la operación del actuador es conocido como tiempo muerto o retardo de transporte ( $T_r$ ).

La ecuación diferencial que describe la dinámica de la planta es:

$$\tau * T_o'(t) + T_o(t) = T_i(t - T_r)$$

Donde:

$T_i$  = Temperatura del aire que ingresa a la cámara

$T_o$  = Temperatura del aire que sale de la cámara

$\tau$  = Constante de tiempo del sistema

La función de transferencia del sistema térmico está representada por:

$$G_1(S) = \frac{T_o(S)}{T_i(S)} = \frac{e^{-T_r s}}{\tau S + 1}$$

Los parámetros de  $G_1(S)$  se obtienen a partir de la [Ref. 37].

El sistema térmico con estas características existe en el laboratorio de Sistemas de Control de la Facultad de Ingeniería Eléctrica en la EPN.

$$\tau = 121 \text{ (seg)}$$

$$T_r = 6 \text{ (seg)}$$

$$T_{\text{MUESTREO}} = 2 \text{ (seg)}$$

$$\text{LIMITE} = 10$$

Reemplazando estos datos la función de transferencia correspondiente es:

$$G_1(s) = \frac{e^{-6s}}{121s+1}$$

### 5.1.1 CONTROLES.

Para obtener los parámetros del control PID se utilizó las técnicas de ajuste de Zeigler-Nichols, las reglas del margen de fase de Åström y las técnicas de respuesta de frecuencia; mientras que para el controlador SELF TUNING se especificó el tipo de respuesta y sus características globales.

Los parámetros obtenidos fueron los siguientes:

CONTROL PID:

$$K_P = 8$$

$$T_1 = 40$$

$$T_d = 5$$

#### CONTROL SELF TUNING:

Tipo de respuesta: Primer orden

$$\tau = 2$$

#### 5.1.2 RESULTADOS.

El objetivo de los controladores es mantener constante la temperatura  $T_0$  de la cámara, para el efecto se ingresa los parámetros calculados anteriormente a través de la interfaz de usuario, seleccionando además una entrada paso unitaria de temperatura  $T_1$  que active al sistema.

El resultado obtenido al aplicar las diferentes acciones de control y descripción del funcionamiento de la planta en lazo cerrado, se resume en la siguiente tabla.

Donde:

$M_p(\%)$  = porcentaje de máximo sobreimpulso

$t_p$  = tiempo pico

$t_r$  = tiempo de crecimiento

$t_s$  = tiempo de establecimiento (criterio 5% y 2%)

$E_{ap}$  = error en estado estable



	LAZO CERRADO	PID	DEAD BEAT	SELF TUNING	EXPERTO 1	EXPERTO 2
Mp(%)	-	10.25	-	-	-	0.72
tp(s)	-	56	-	-	-	34
tr(s)	138	34	8	10	18	28
ts(s)5%	176	98	8	12	20	20
ts(s)2%	228	126	8	14	22	22
Eap	0.5	0	0	0	0	0

Tabla 5.1: Funcionamiento del sistema térmico con retardo de transporte.

A continuación se muestran los resultados gráficos obtenidos de la simulación del proceso al aplicar las diferentes acciones de control.

**Fig. 5.1.a y Fig. 5.1.b**

Ilustra el comportamiento del sistema térmico con retardo de transporte al cual no se ha aplicado ninguna acción de control. La respuesta del sistema en lazo abierto y cerrado, crece exponencialmente antes de alcanzar el valor final. La respuesta del sistema en lazo cerrado mantiene un error en estado estable igual a 0.5.

**Fig. 5.1.c a Fig. 5.1.g**

Se muestran las diferentes respuestas del sistema al aplicar

cada uno de los controles.

Fig. 5.1.h

Presenta un gráfico comparativo de todas las respuestas del sistema dadas en las figuras anteriores, pero en este caso considerando sólo la respuesta al control EXPERTO 1. Esta última señal se caracteriza por no presentar sobreimpulso y poseer un bajo tiempo de establecimiento.

Fig. 5.1.i

Muestra las señales de error de cada controlador incluyendo el error del control EXPERTO 1.

Fig. 5.1.j

En esta figura se presenta la señal de control que es aplicada al sistema. Donde la variable manipulada posee el máximo valor al inicio de la simulación, lo que produce una saturación del actuador como resultado de utilizar como primera acción de control la respuesta del controlador DEADBEAT.

Fig. 5.1.k

Presenta el funcionamiento del sistema al utilizar cada uno de los controladores y la acción de control EXPERTO 2. Esta última, al igual que la respuesta del control EXPERTO 1, posee un bajo tiempo de establecimiento igual a 20 (seg) utilizando el criterio del 5%. La rapidez de la respuesta se debe a la aplicación del controlador DEADBEAT cuando se inicia la simulación. Además la respuesta presenta un pequeño

sobreimpulso de 0.72%.

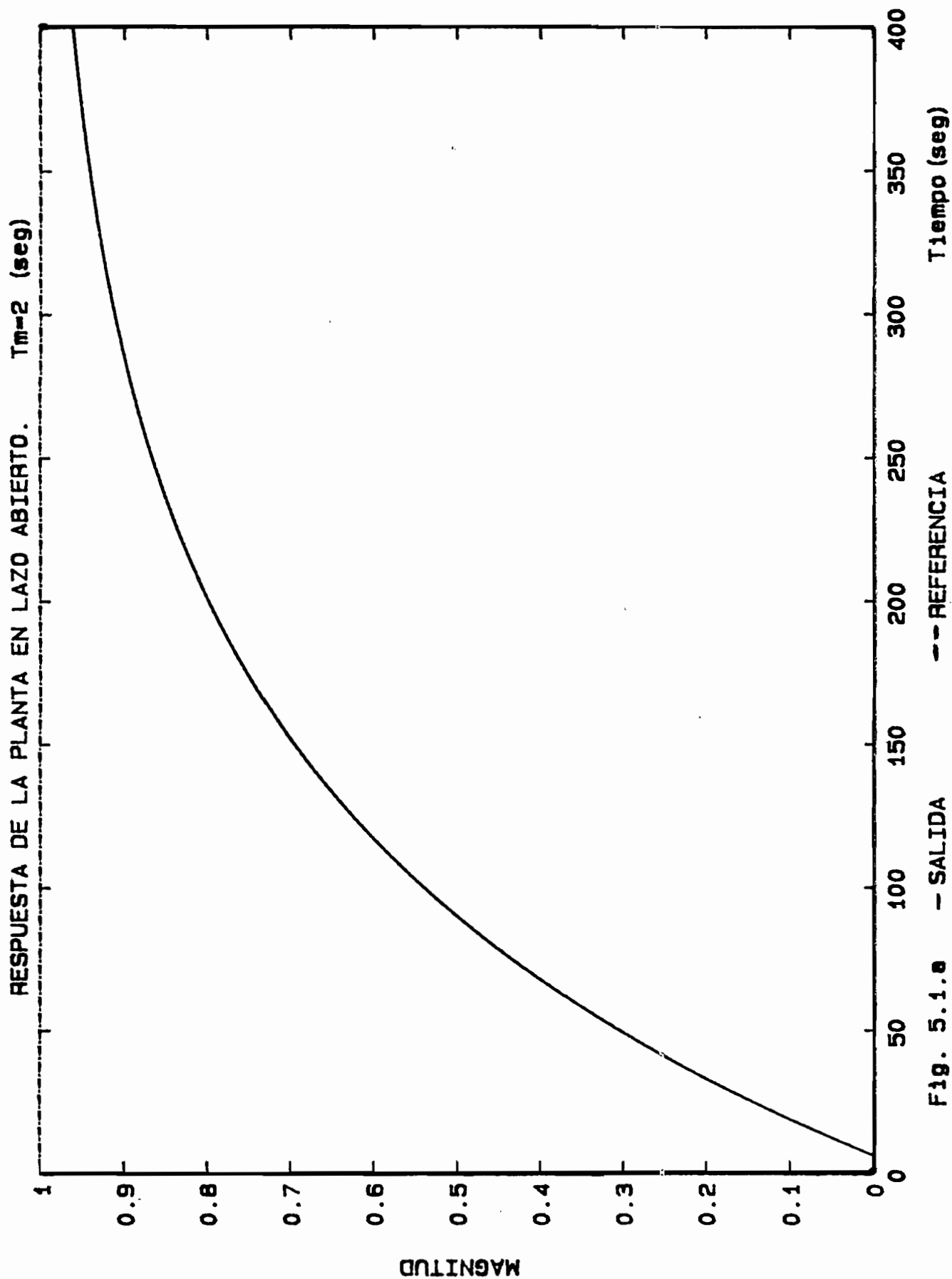
Fig. 5.1.1

En esta figura se muestra las señales de error del sistema al actuar los tres tipos de controles incluyedo el error del controlador EXPERTO 2.

Fig. 5.1.m

Presenta la señal de control EXPERTO 2 que es aplicada al sistema. El valor inicial de la señal satura al actuador del proceso, como resultado de conectar a la planta una acción de control resultante, donde su principal componente es la señal obtenida del controlador DEADBEAT.

En general las respuestas obtenidas con el controlador EXPERTO 1 y EXPERTO 2, presentan mejores características en la respuesta del sistema.



RESPUESTA DE LA PLANTA EN LAZO CERRADO.  $T_m=2$  (seg)

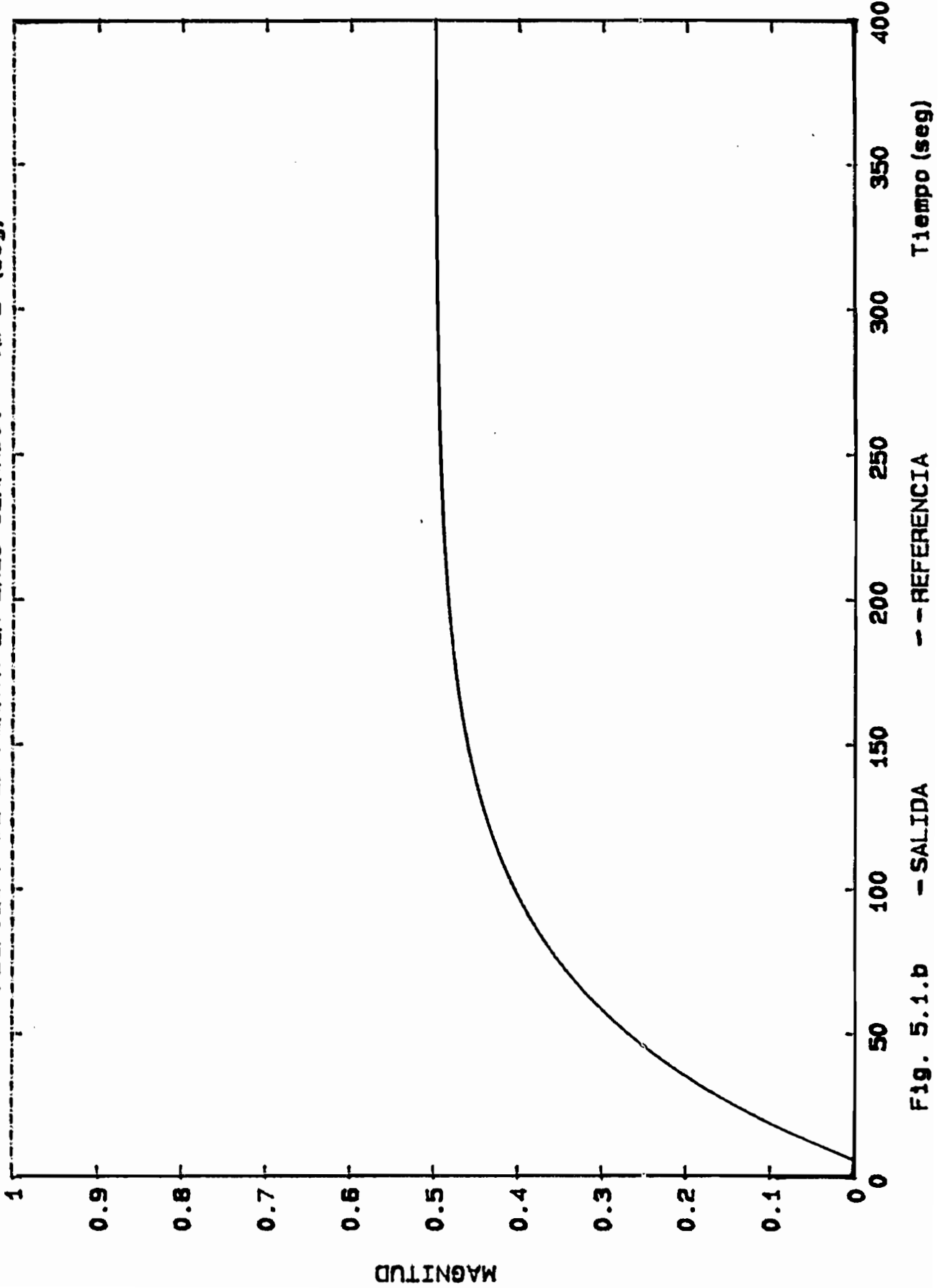
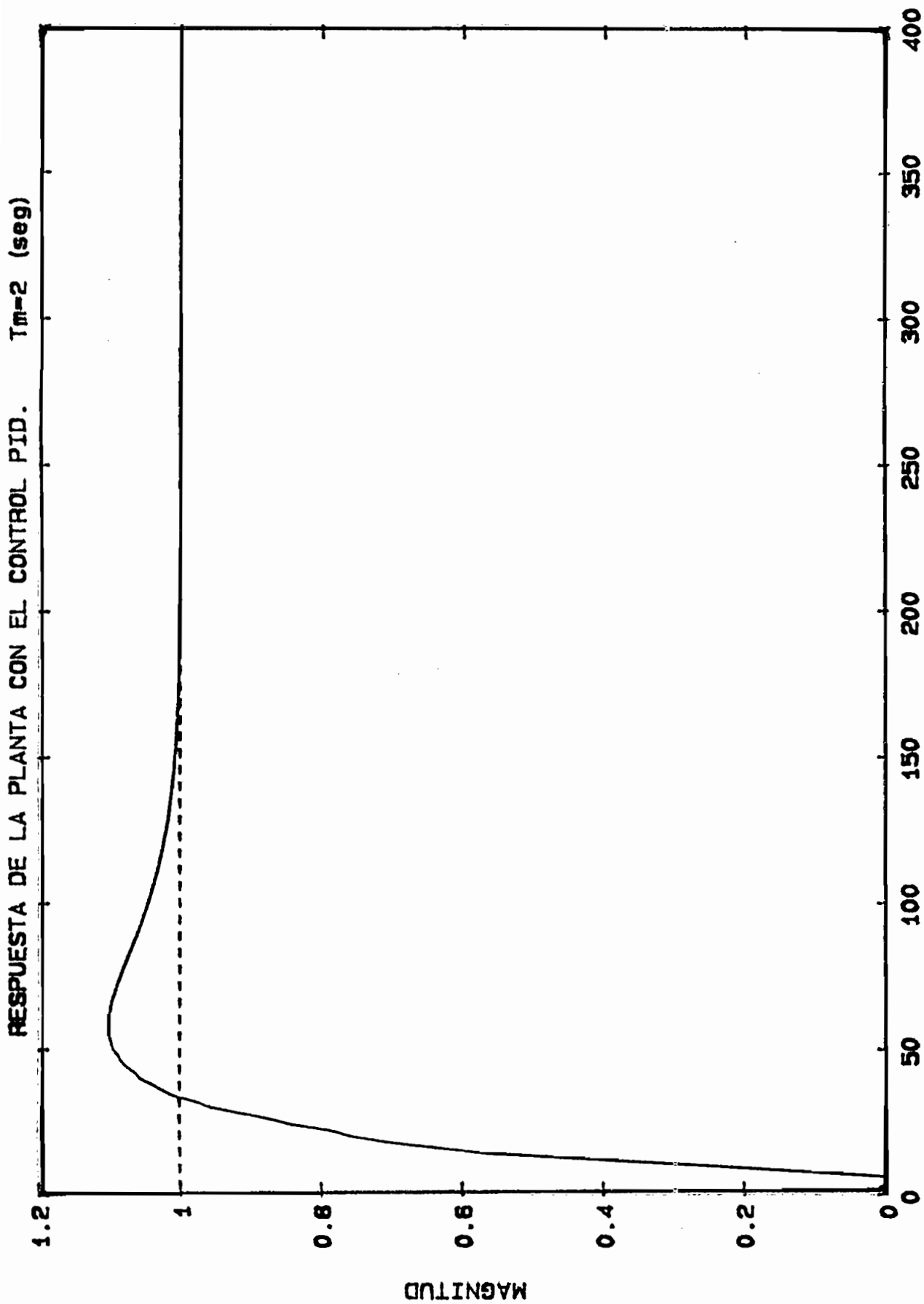


Fig. 5.1.b - SALIDA -- REFERENCIA



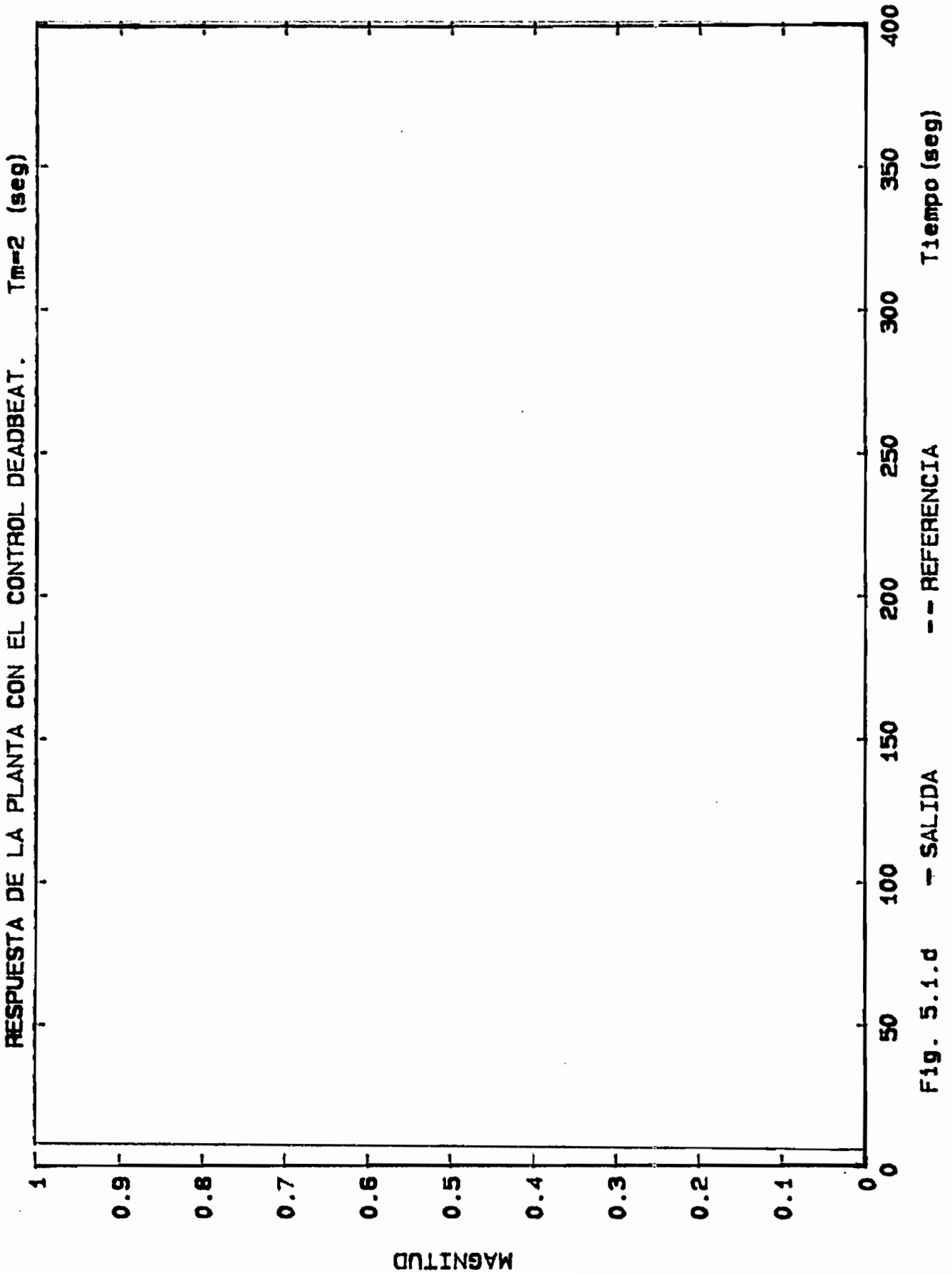


Fig. 5.1.d

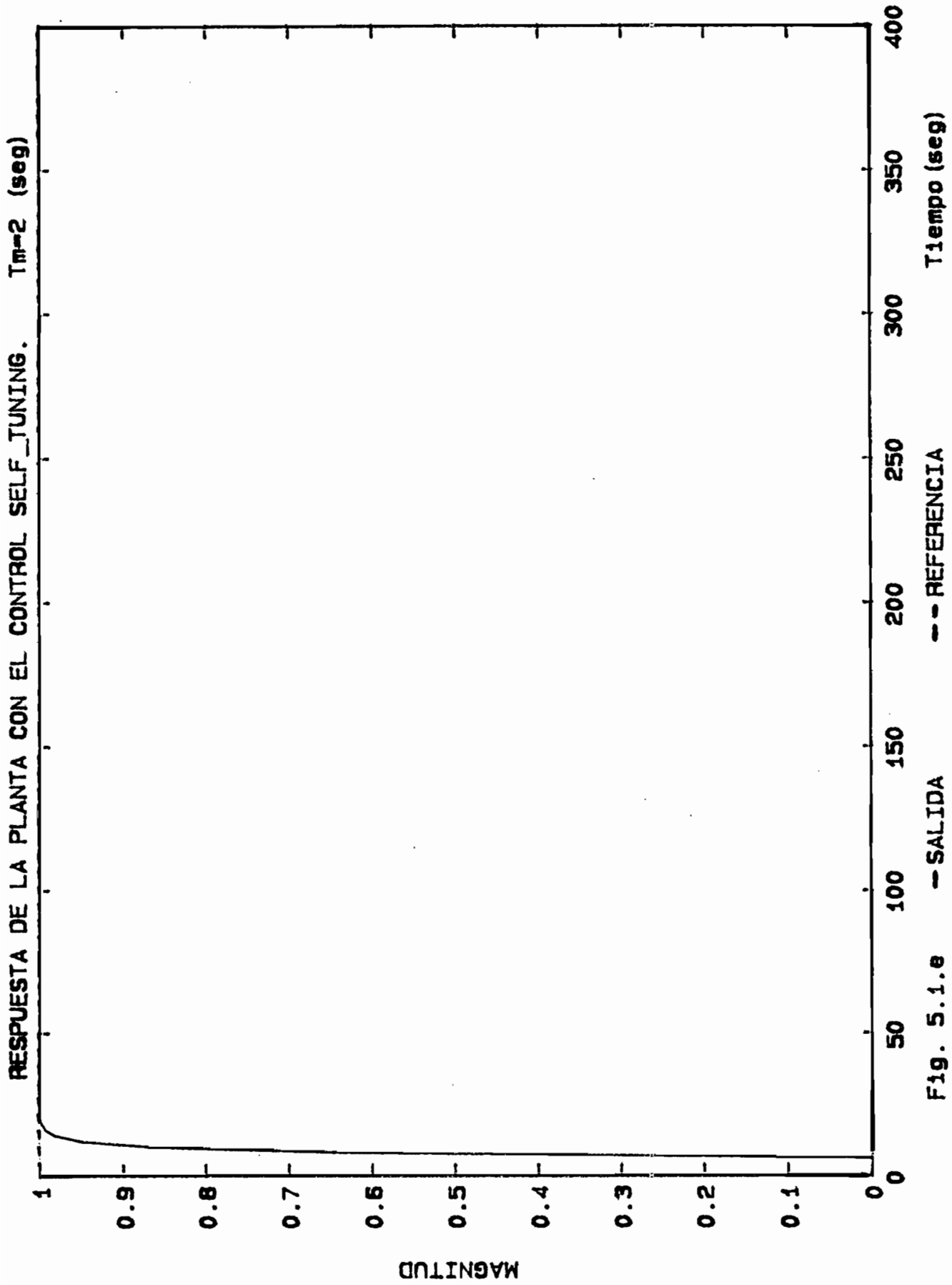


Fig. 5.1.e



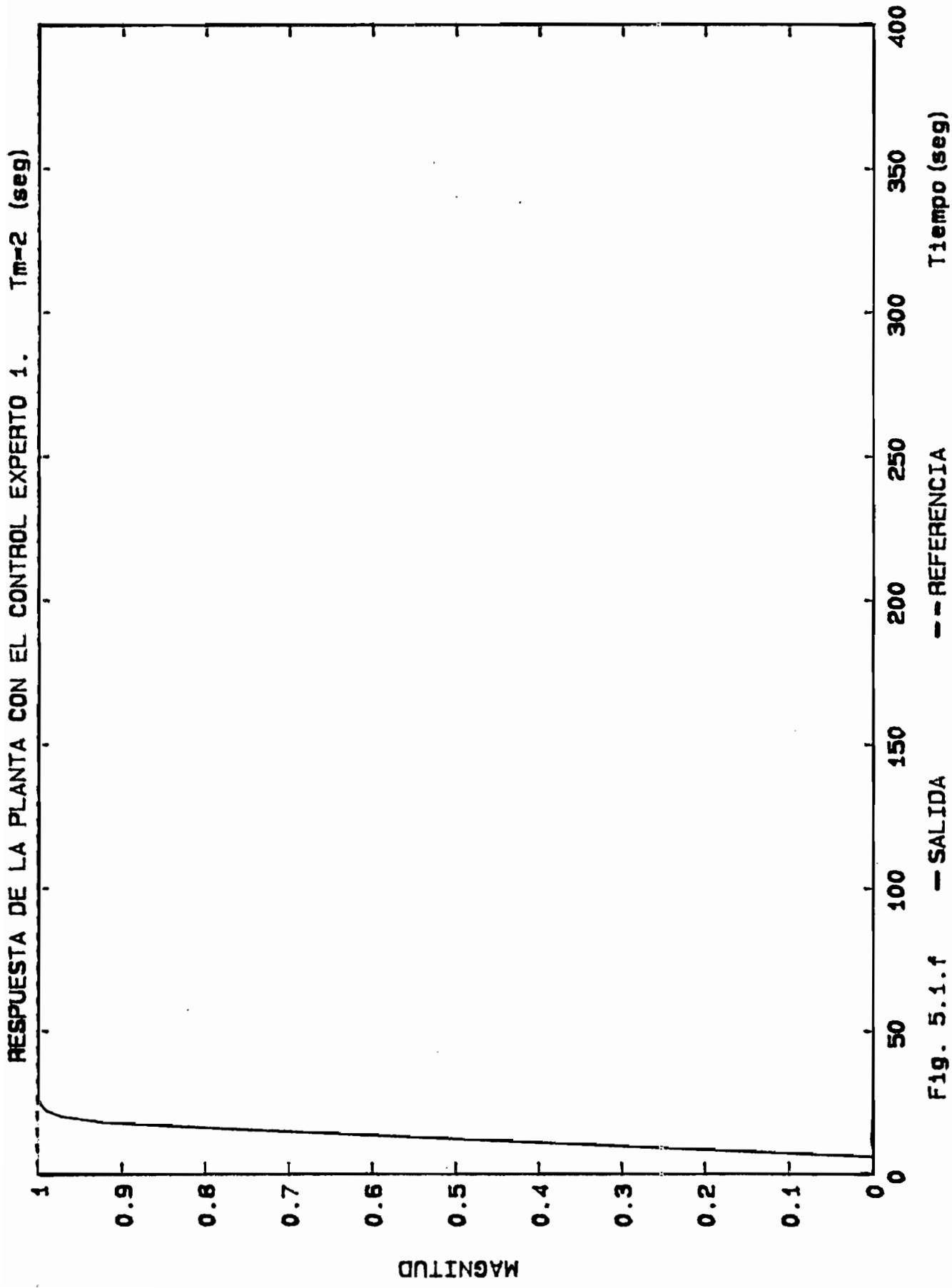


Fig. 5.1.1

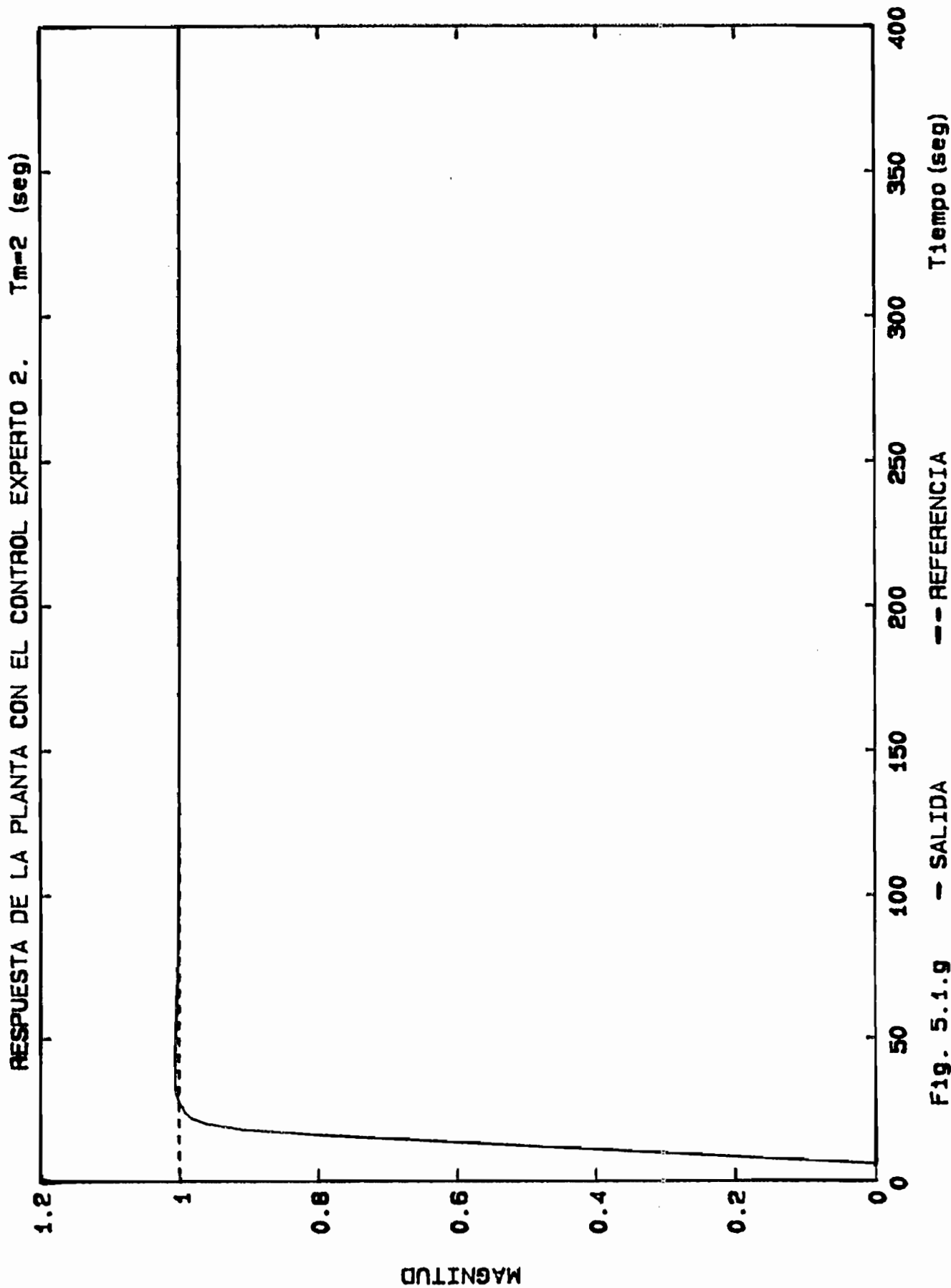
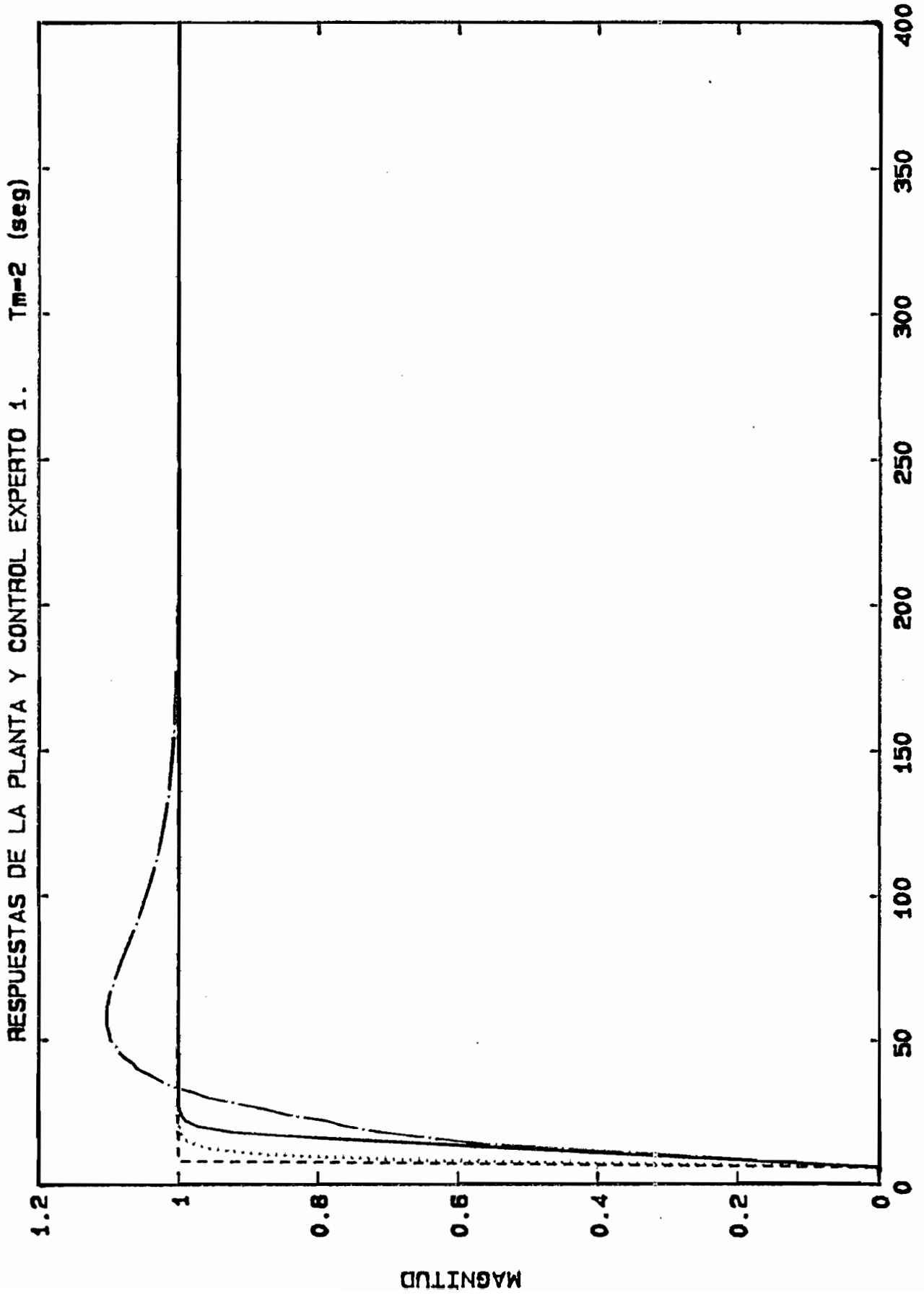


Fig. 5.1.9



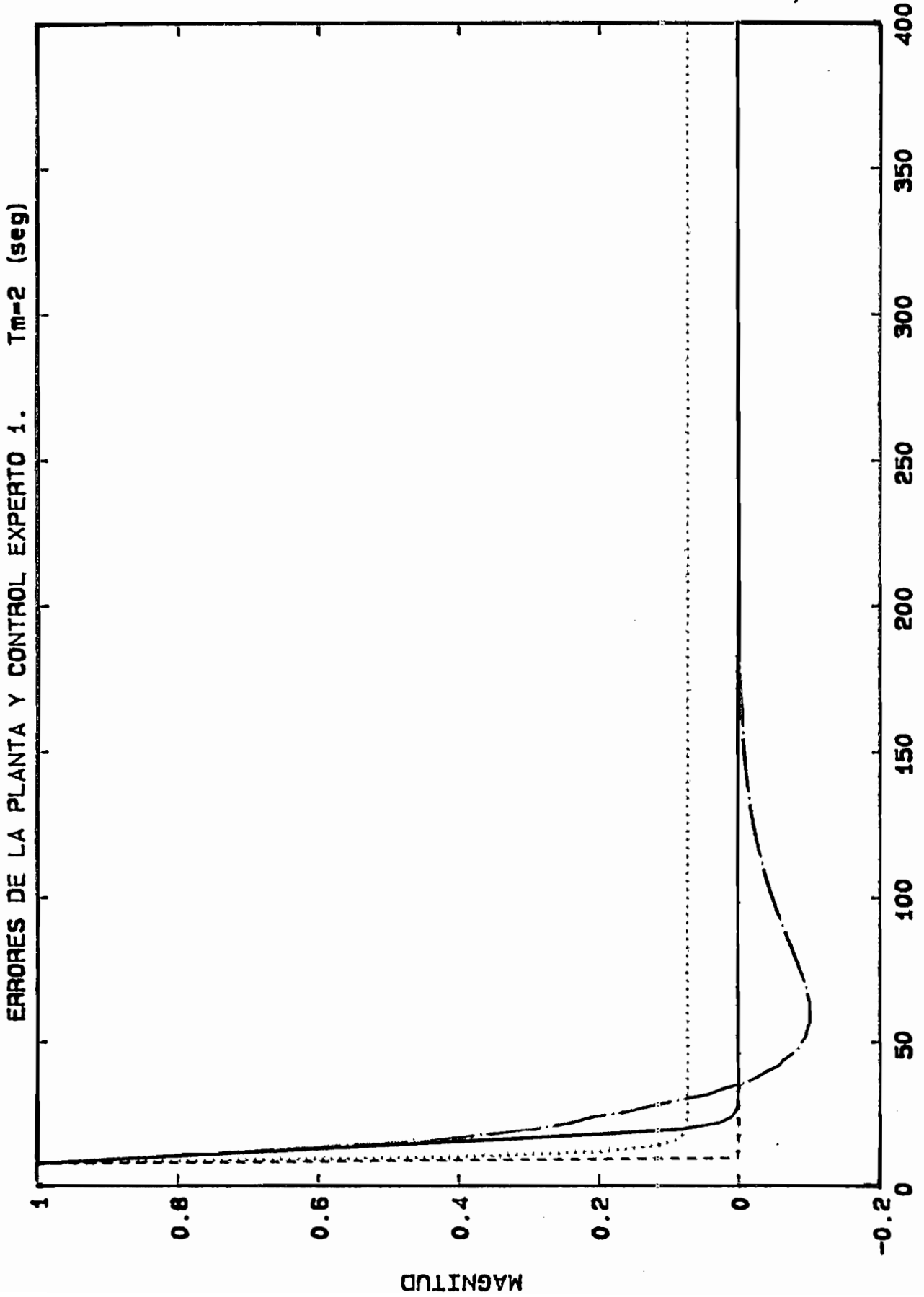


Fig. 5.1.1 --- PID -.- DEADBEAT ... SELF-TUNING - EXPERTO 1

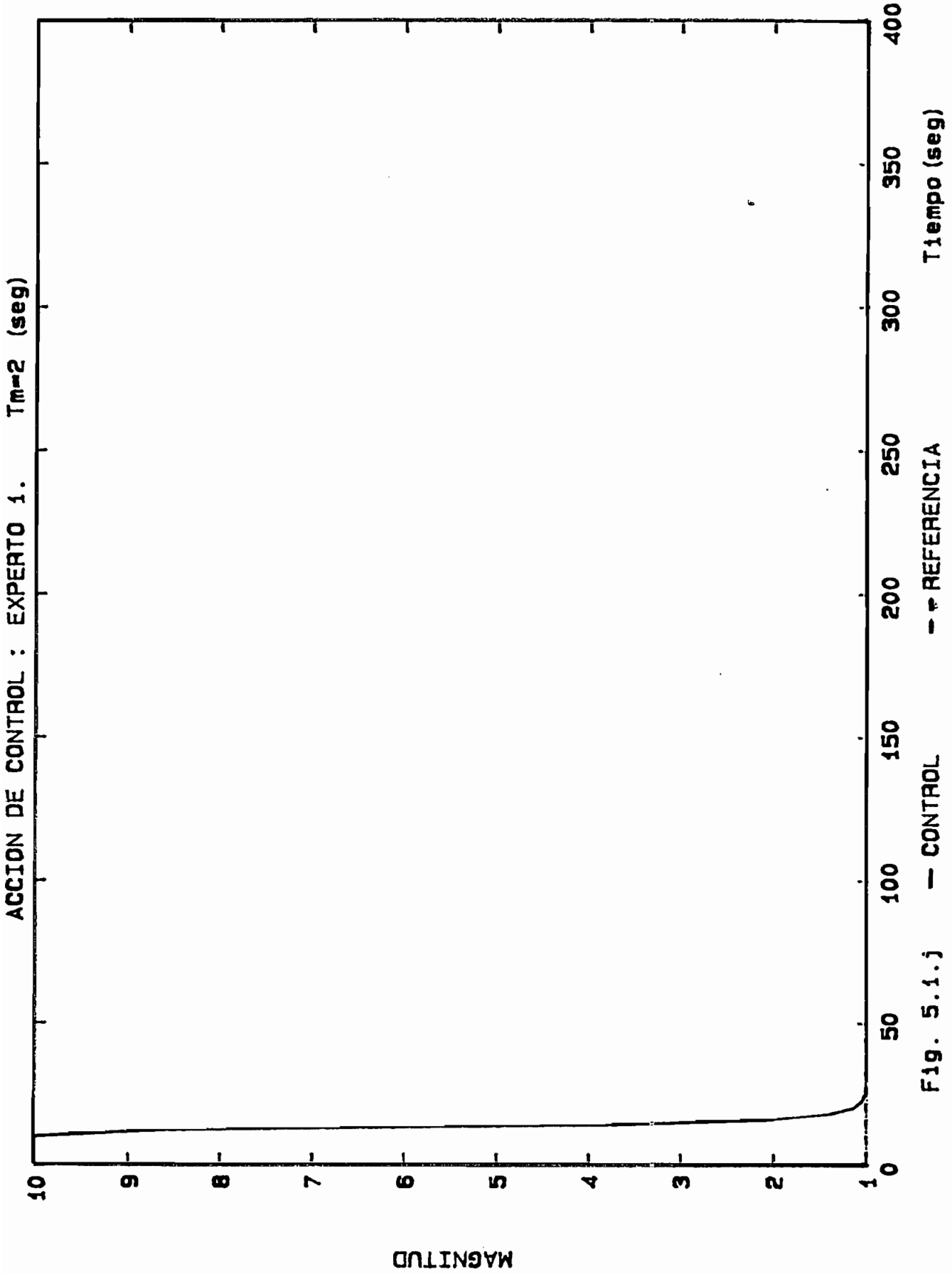


Fig. 5.1.1

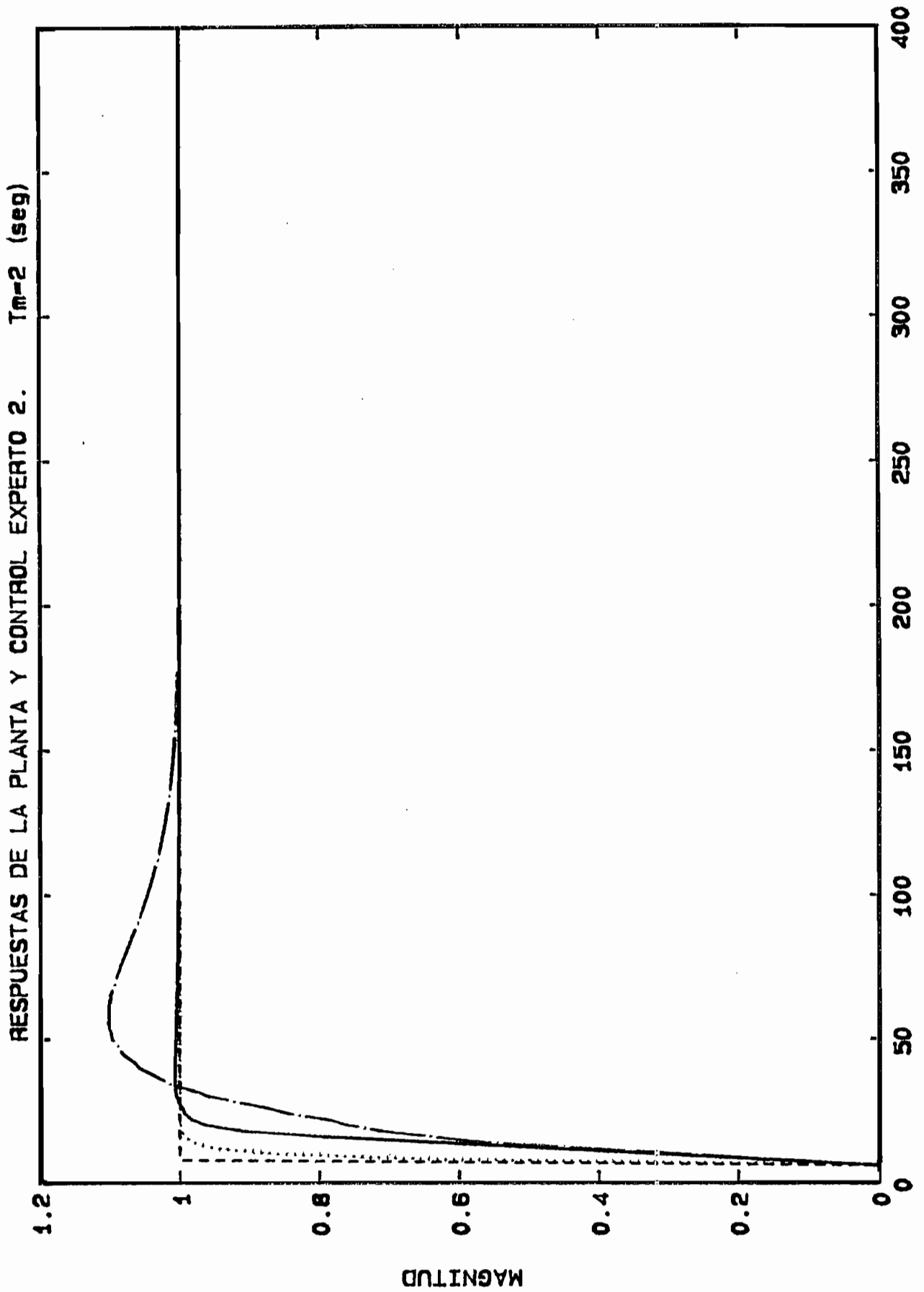
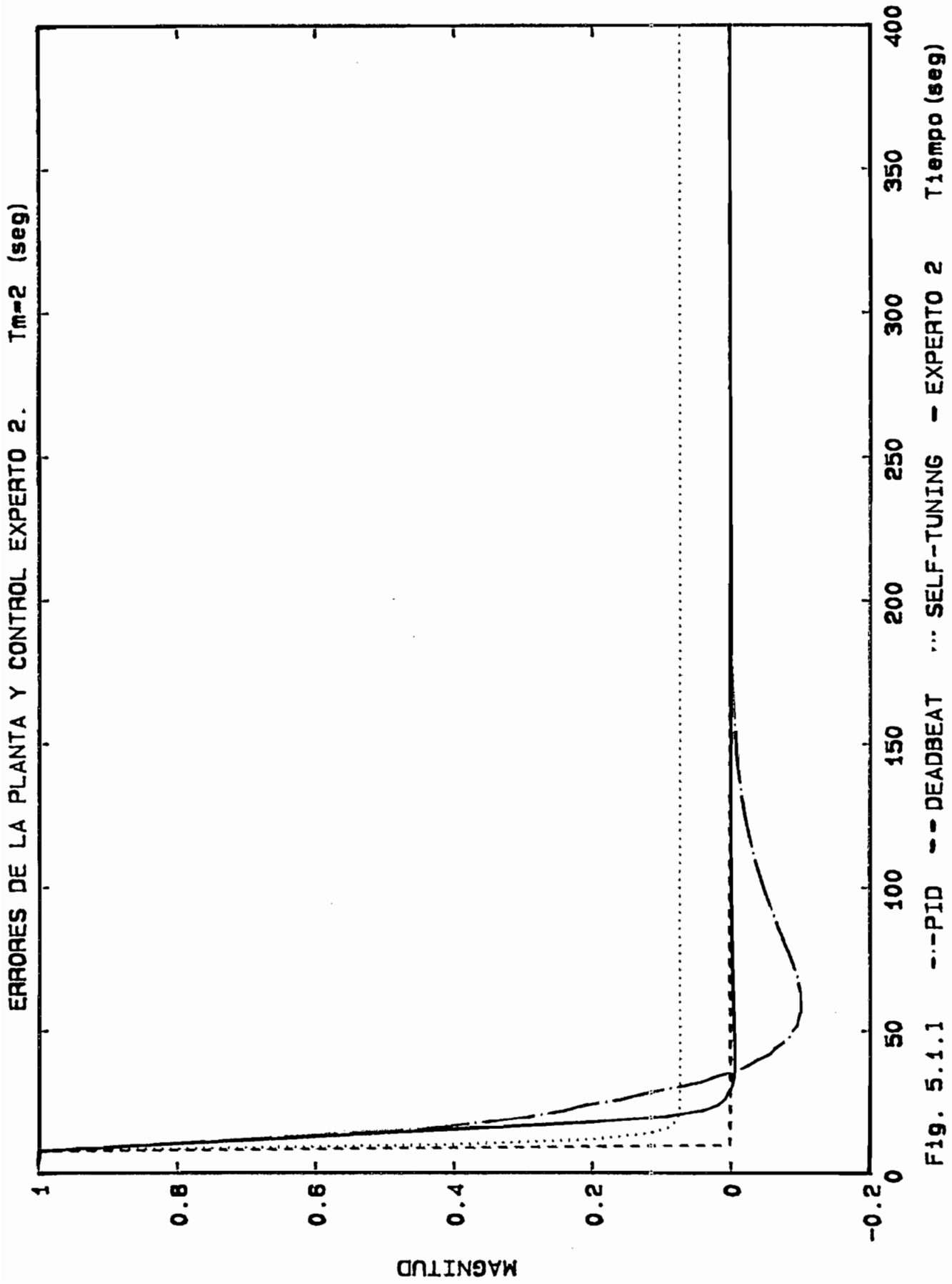


Fig. 5.1.1.k - - - PID - - - DEADBEAT ... SELF-TUNING - EXPERTO 2 Tiempo (seg)



## 5.2 CONTROL DE VELOCIDAD Y POSICION DE UN SERVOMOTOR.

En muchas aplicaciones prácticas se requiere el control de pequeños motores de corriente continua, así por ejemplo en los robots industriales, manipuladores médicos, entre otros. Por esta razón se considera importante analizar el comportamiento de un motor DC controlado por armadura.

La Fig. 5.2 presenta el diagrama de bloques del servomotor a controlar. [Ref. 39]

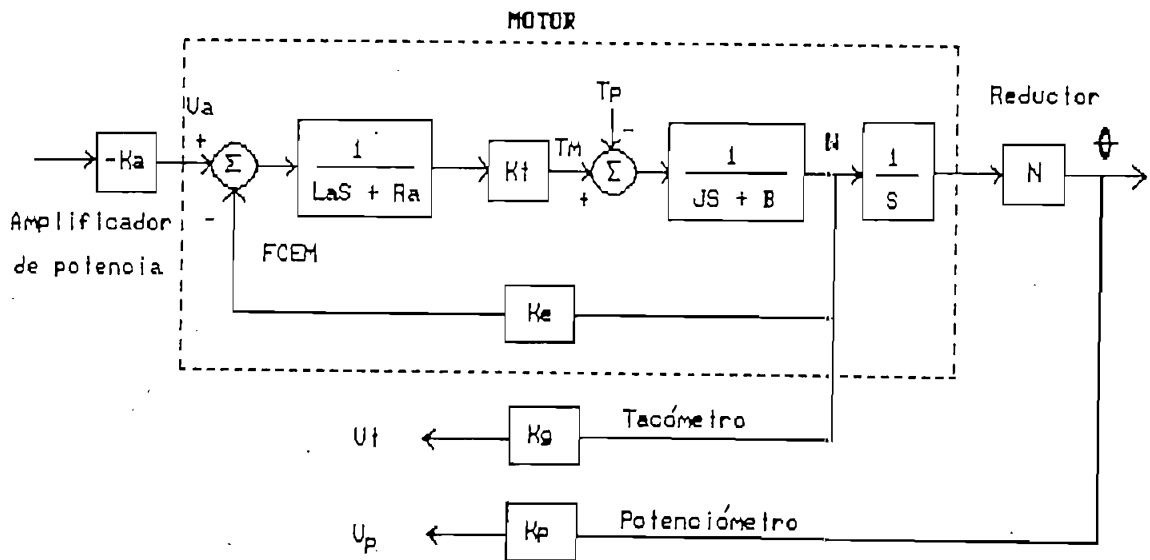


Fig. 5.2: Diagrama de bloques del servomotor.



Donde:

$K_a$	=	Ganancia del amplificador de potencia
$V_a$	=	Voltaje de armadura
$F_{CEM}$	=	Fuerza contra electromotriz
$T_m$	=	Torque del motor
$T_p$	=	Torque de perturbación en la carga
$W$	=	Velocidad del motor
$\theta$	=	Posición angular
$V_t$	=	Voltaje a la salida del tacómetro
$V_p$	=	Voltaje a la salida del sensor de posición
$K_g$	=	Constante del tacómetro
$K_p$	=	Constante del potenciómetro
$K_e$	=	Constante de velocidad
$K_t$	=	Constante de torque
$J$	=	Momento de inercia del motor
$B$	=	Amortiguamiento o fricción viscosa
$R_a$	=	Resistencia del motor.
$L_a$	=	Inductancia del motor
$N$	=	Relación de transformación del reductor
$V_e$	=	Voltaje de error

Para propósitos de control, se utiliza un diagrama de bloques simplificado como el que se muestra en la Fig. 5.3.

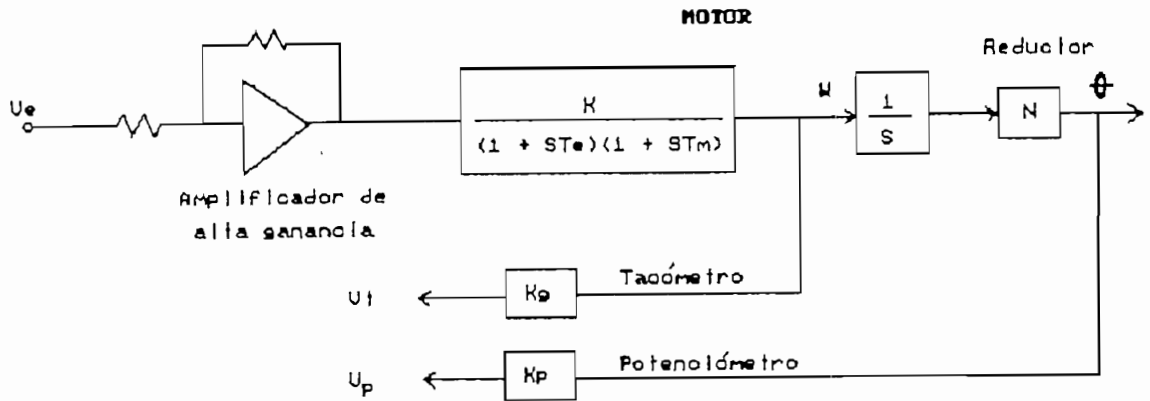


Fig. 5.3: Diagrama de bloques simplificado del servomotor.

Donde:

$$\begin{aligned}
 K &= 110 \text{ [rad/V*s]} \\
 T_m &= (11.5)^{-1} \text{ [s]} \\
 K_g &= 0.148 \text{ [V*s/rad]} \\
 T_e &= (1623)^{-1} \text{ [s]} \\
 K_p &= 6.74 \text{ [V/rad]}
 \end{aligned}$$

Estos parámetros corresponden al sistema MOTOMATIC MSL 100, existente en el laboratorio de Sistemas de Control de la Facultad de Ingeniería Eléctrica de la EPN.

### 5.2.1 CONTROL DE VELOCIDAD.

Para realizar el control de velocidad del servomotor se considera el diagrama de bloques que se presenta en la Fig. 5.4.

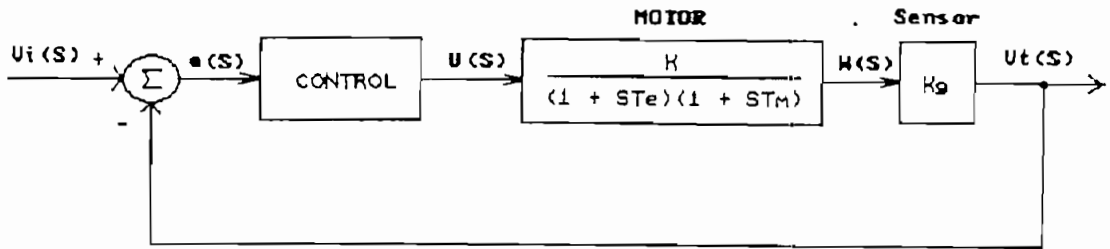


Fig. 5.4: Control de velocidad del Servomotor.

Reemplazando los valores indicados anteriormente, la función de transferencia del motor es:

$$\frac{W(S)}{U(S)} = \frac{2053095}{(S+1623)(S+11.5)}$$

Simplificando el polo lejano y haciendo un ajuste de ganancia la función de transferencia del motor está dada por:

$$\frac{W(S)}{U(S)} = \frac{1265}{S+11.5}$$

Considerando la ganancia del sensor, la función de transferencia que describe la dinámica del servomotor para control de velocidad, está dada por:

$$G_2(S) = \frac{187.22}{S+11.5}$$

Otras constantes que intervienen en la simulación del proceso son:

TMUESTREO = 0.0005 (seg)  
 LIMITE = 10

#### 5.2.1.1 CONTROLES.

Utilizando los mismos métodos indicados en el caso anterior, los parámetros que definen el bloque de control son:

CONTROL PID:

$K_P$  = 1.6  
 $T_i$  = 0.01  
 $T_d$  = 0.0025

CONTROL SELF TUNING:

Tipo de respuesta: Primer orden  
 $\tau$  = 0.002

#### 5.1.2 RESULTADOS.

Se ingresan al programa la función de transferencia del sistema

$G_2(S)$  y los parámetros de los controles, definiendo además una entrada paso unitaria para simular el voltaje de entrada  $V_1$ . La respuesta del sistema está representada por el voltaje  $V_\tau$ , el cual es proporcional a la velocidad angular a la que gira el motor.

En la siguiente tabla se resume el funcionamiento del control de velocidad del motor en lazo cerrado, y utilizando cada uno de los controles:

	LAZO CERRADO	PID	DEAD BEAT	SELF TUNING	EXPERTO 1	EXPERTO 2
$M_p(\%)$	-	7.52	-	-	-	1.58
$t_p(s)$	-	0.0195	-	-	-	0.0110
$t_r(s)$	0.0110	0.0090	0.0005	0.0045	0.0005	0.0065
$t_s(s)5\%$	0.0145	0.0285	0.0005	0.0060	0.0015	0.0400
$t_s(s)2\%$	0.0190	0.0370	0.0005	0.0080	0.0030	0.0500
$E_{ap}$	0.0600	0	0	0	0	0

Tabla 5.2: Funcionamiento del control de velocidad del servomotor.

A continuación se muestran los resultados gráficos obtenidos de la simulación del proceso al aplicar las diferentes acciones de control.

**Fig. 5.2.1.a y Fig. 5.2.1.b**

Ilustra el comportamiento del servomotor al control de velocidad en lazo abierto y lazo cerrado respectivamente. El sistema se comporta como un típico sistema de primer orden donde su respuesta en lazo abierto y cerrado, crece exponencialmente antes de alcanzar el valor final. La respuesta del sistema en lazo cerrado mantiene un error en estado estable igual a 0.06.

**Fig. 5.2.1.c a Fig. 5.2.1.g**

Muestran las diferentes respuestas del sistema al aplicar cada uno de los controles.

**Fig. 5.2.1.h**

Presenta un gráfico comparativo de todas las respuestas del sistema al bloque de controles, añadiéndose la respuesta al control EXPERTO 1. Esta última señal se caracteriza por no presentar sobreimpulso y poseer un mínimo tiempo de establecimiento, igual a 0.006 (seg).

**Fig. 5.2.1.i**

Muestra las señales de error de cada controlador incluyendo el error del control EXPERTO 1.

**Fig. 5.2.1.j**

Presenta la señal de control que es aplicada al sistema. Donde la variable manipulada posee el máximo valor al inicio de la simulación, lo que produce una saturación del actuador como

resultado de utilizar como primera acción de control la respuesta del controlador DEADBEAT.

Fig. 5.2.1.k

Describe el funcionamiento del sistema al utilizar cada uno de los controladores y la acción de control EXPERTO 2. Esta última presenta un pequeño sobreimpulso igual a 1.58% y un bajo tiempo de establecimiento. La rapidez con la que responde el proceso se debe a la aplicación del controlador DEADBEAT cuando existe un cambio de la referencia.

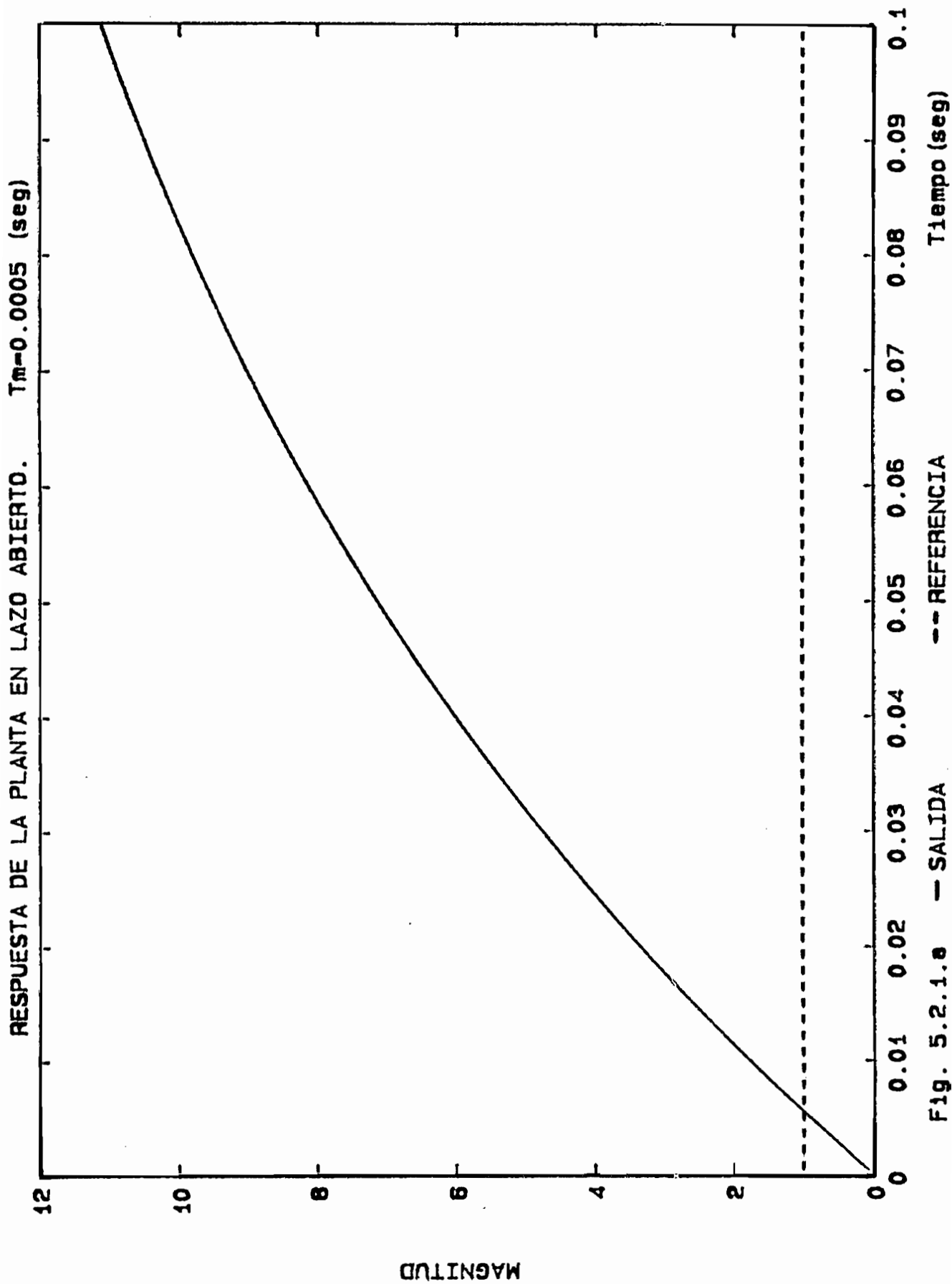
Fig. 5.2.1.l

En esta figura se muestra las señales de error del sistema al actuar los tres tipos de controles incluido el error del controlador EXPERTO 2.

Fig. 5.2.1.m

Presenta la señal de control EXPERTO 2 que es aplicada al sistema. En este caso, el valor inicial de la señal no satura al actuador, como resultado de conectar al motor una acción de control resultante, donde se asigna valores de ponderación a cada uno de los controles. Aunque su principal componente es la señal obtenida del controlador DEADBEAT, el resultado generado no sobrepasa el valor límite del actuador.

En general las respuestas obtenidas con el controlador EXPERTO 1 y EXPERTO 2, presentan mejores características en la respuesta del sistema.





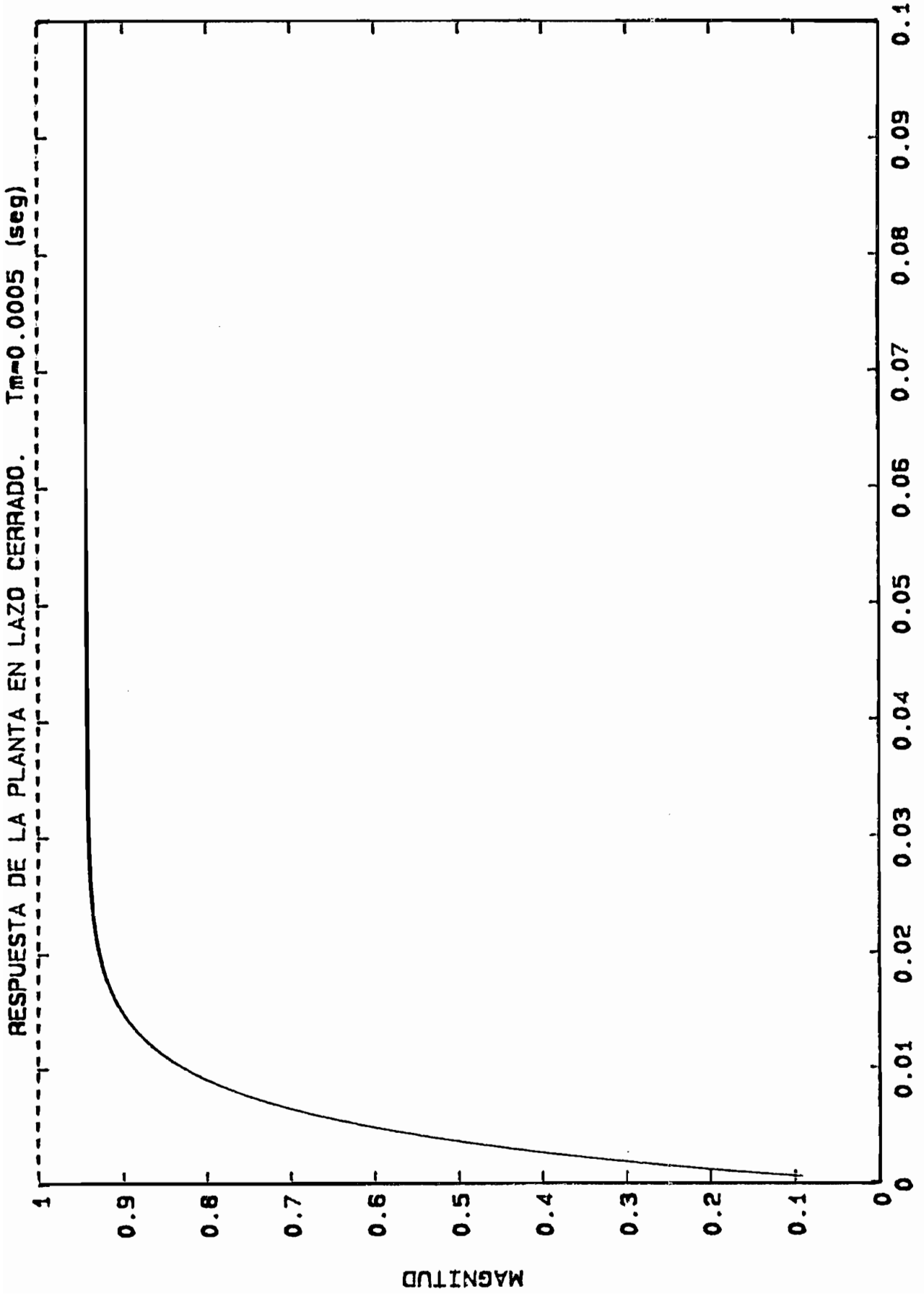


Fig. 5.2.1.b

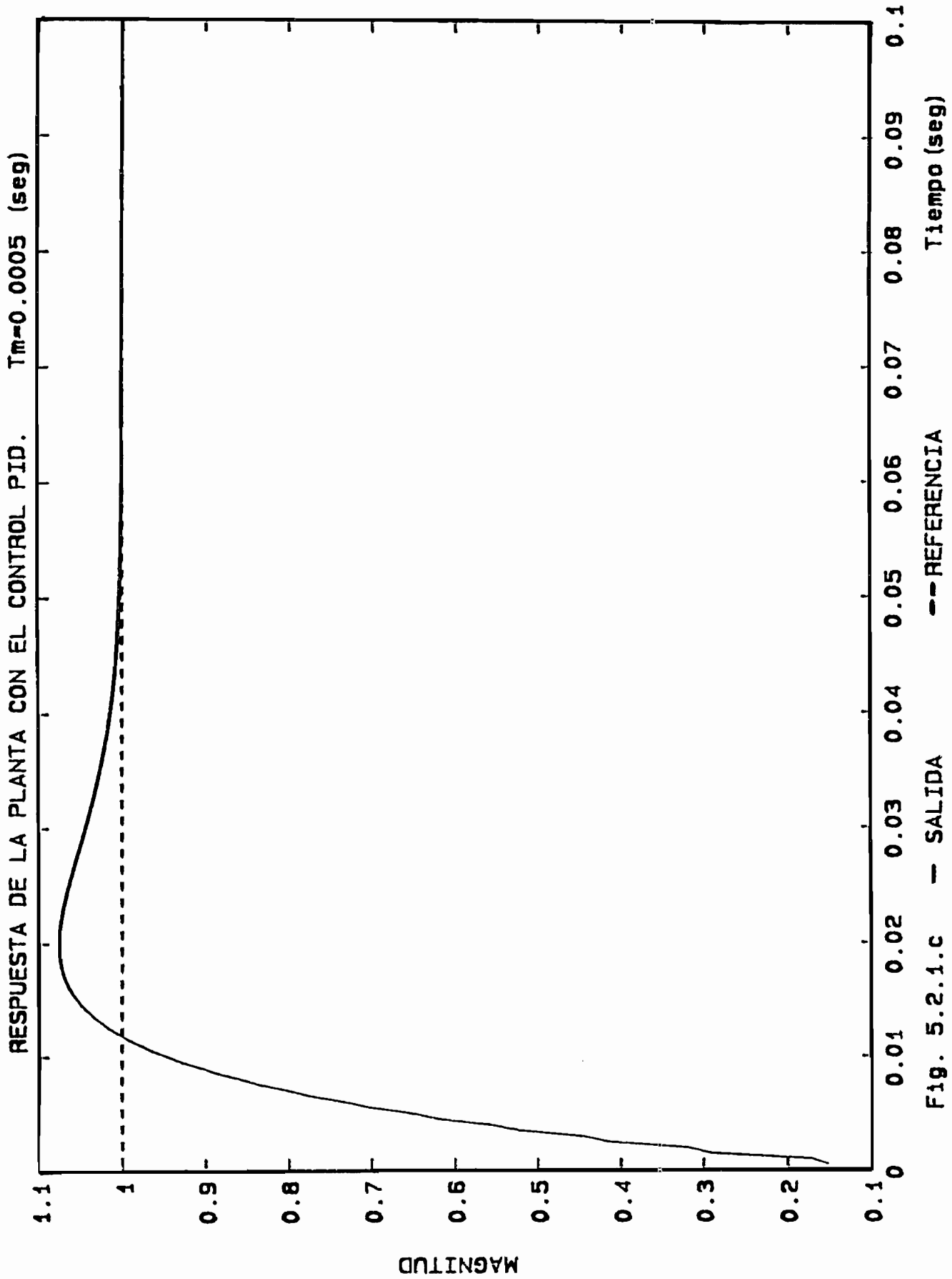


Fig. 5.2.1.c

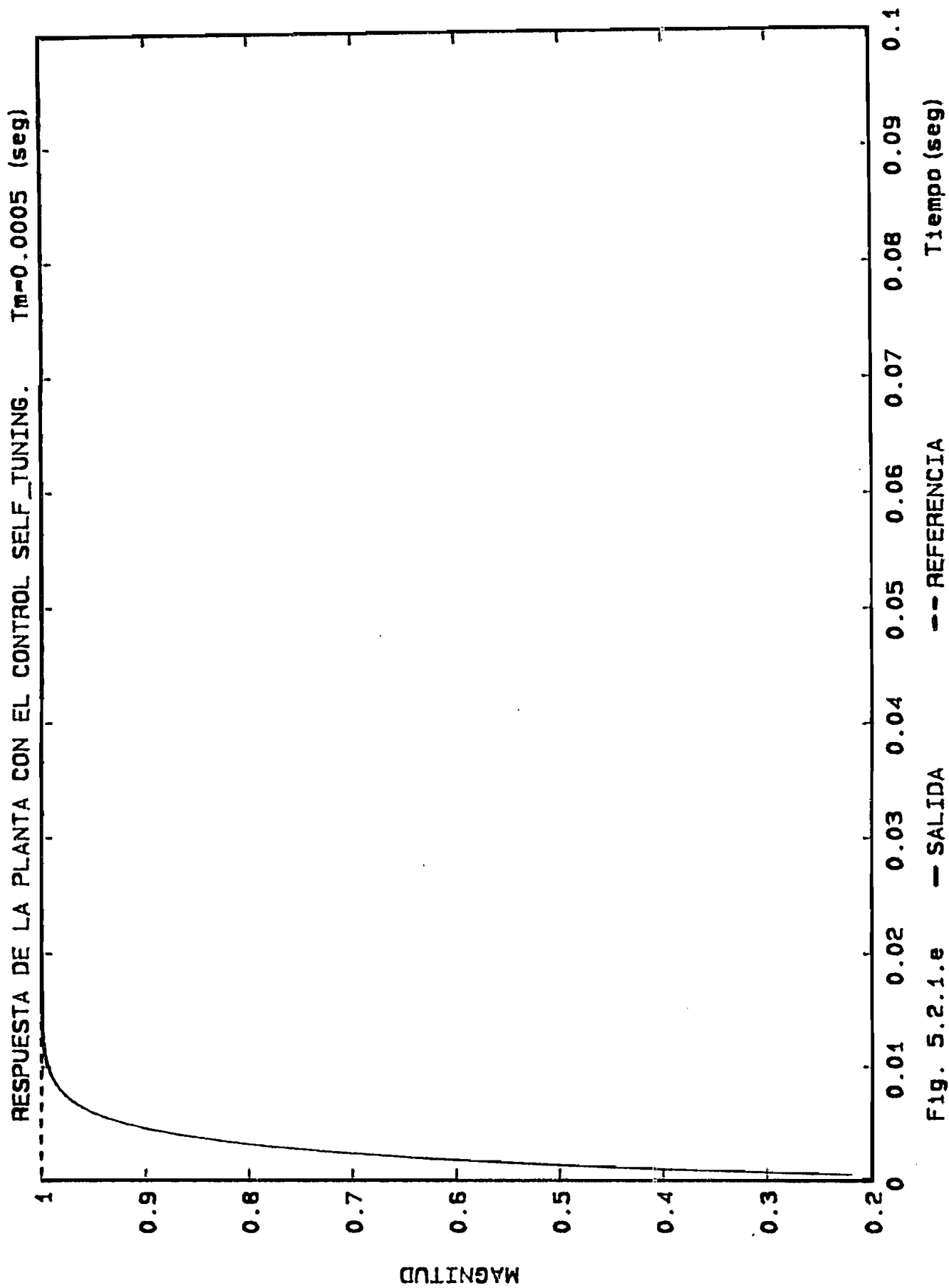
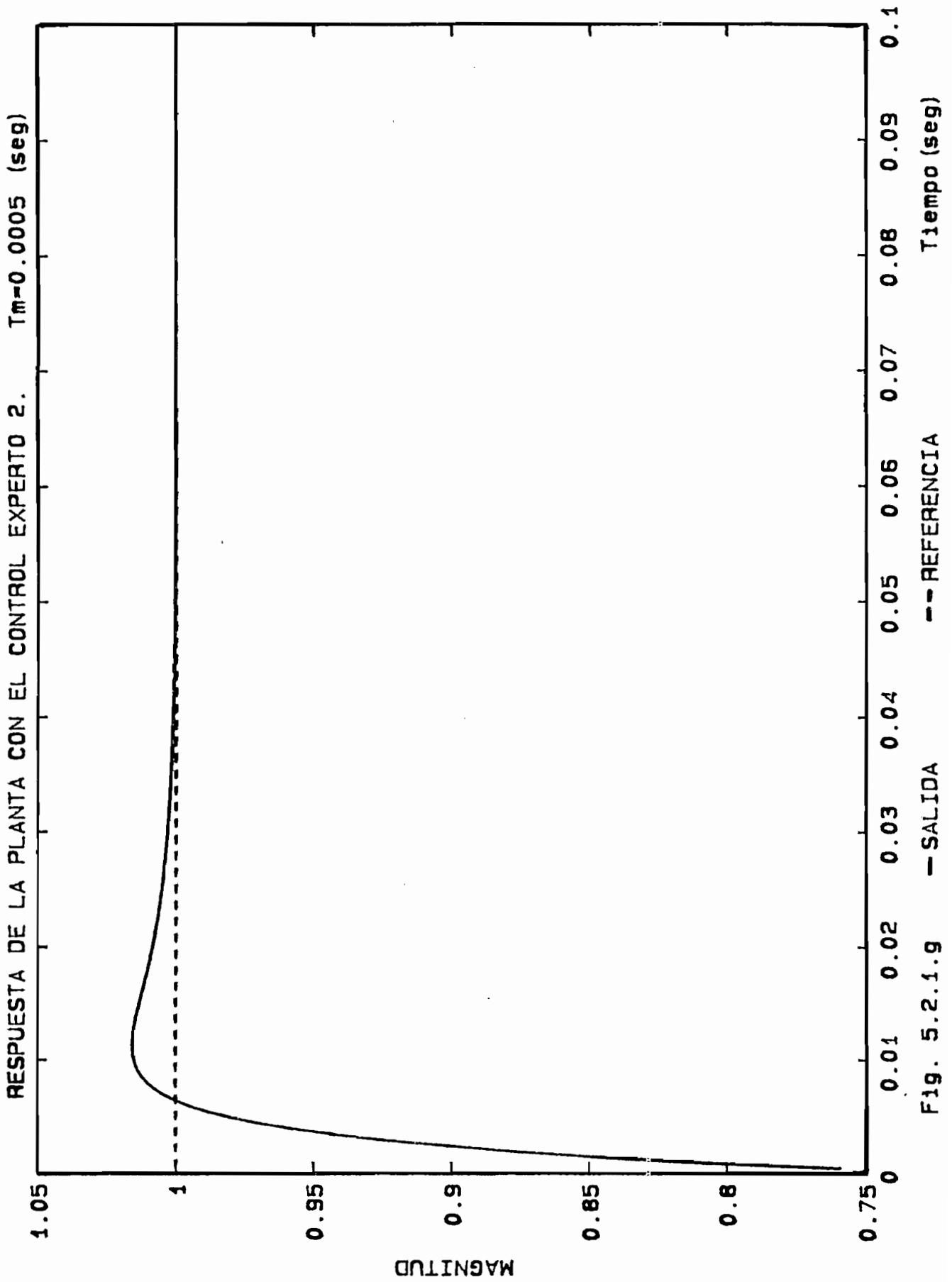


Fig. 5.2.1.e



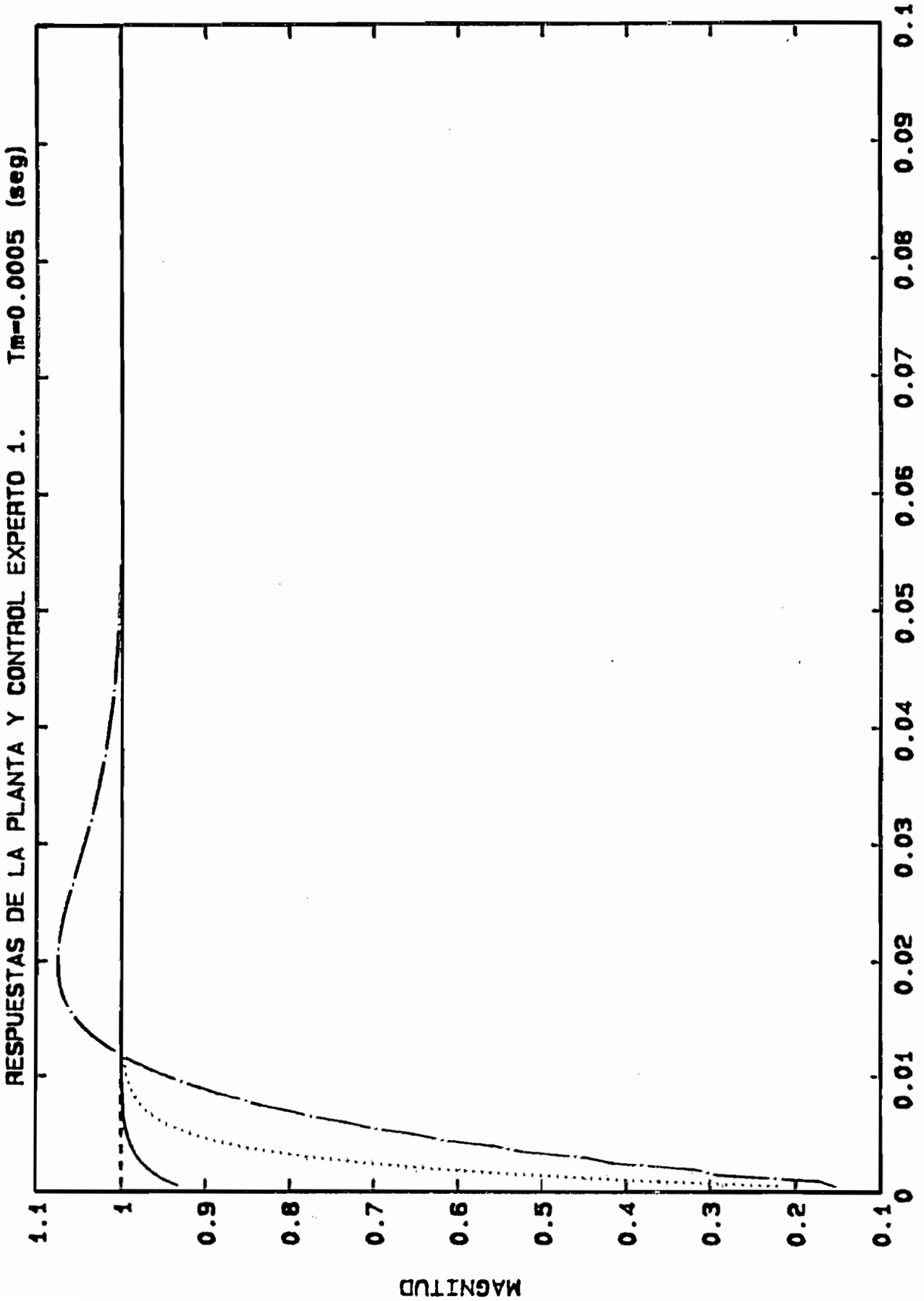


Fig. 5.2.1.h

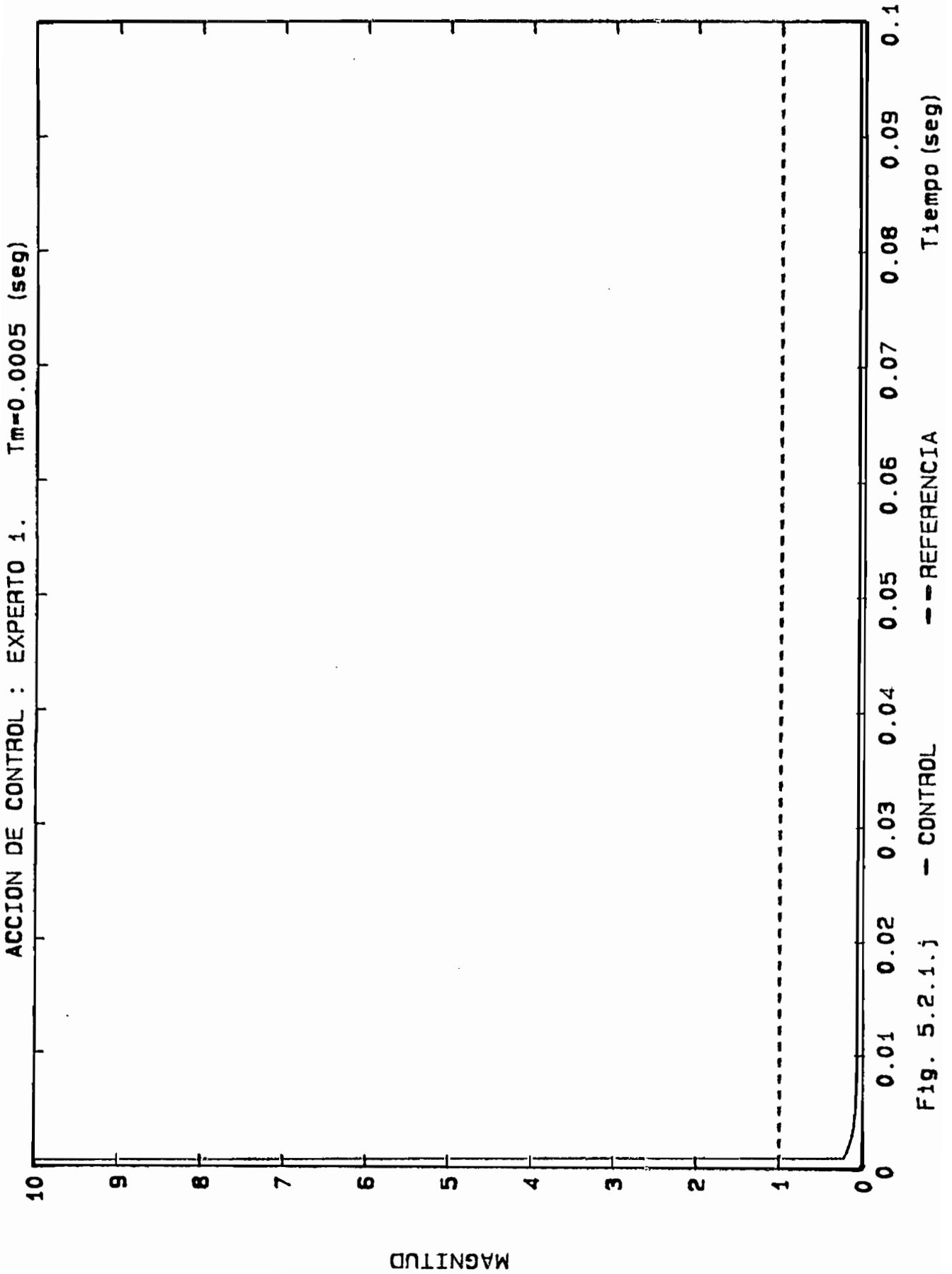
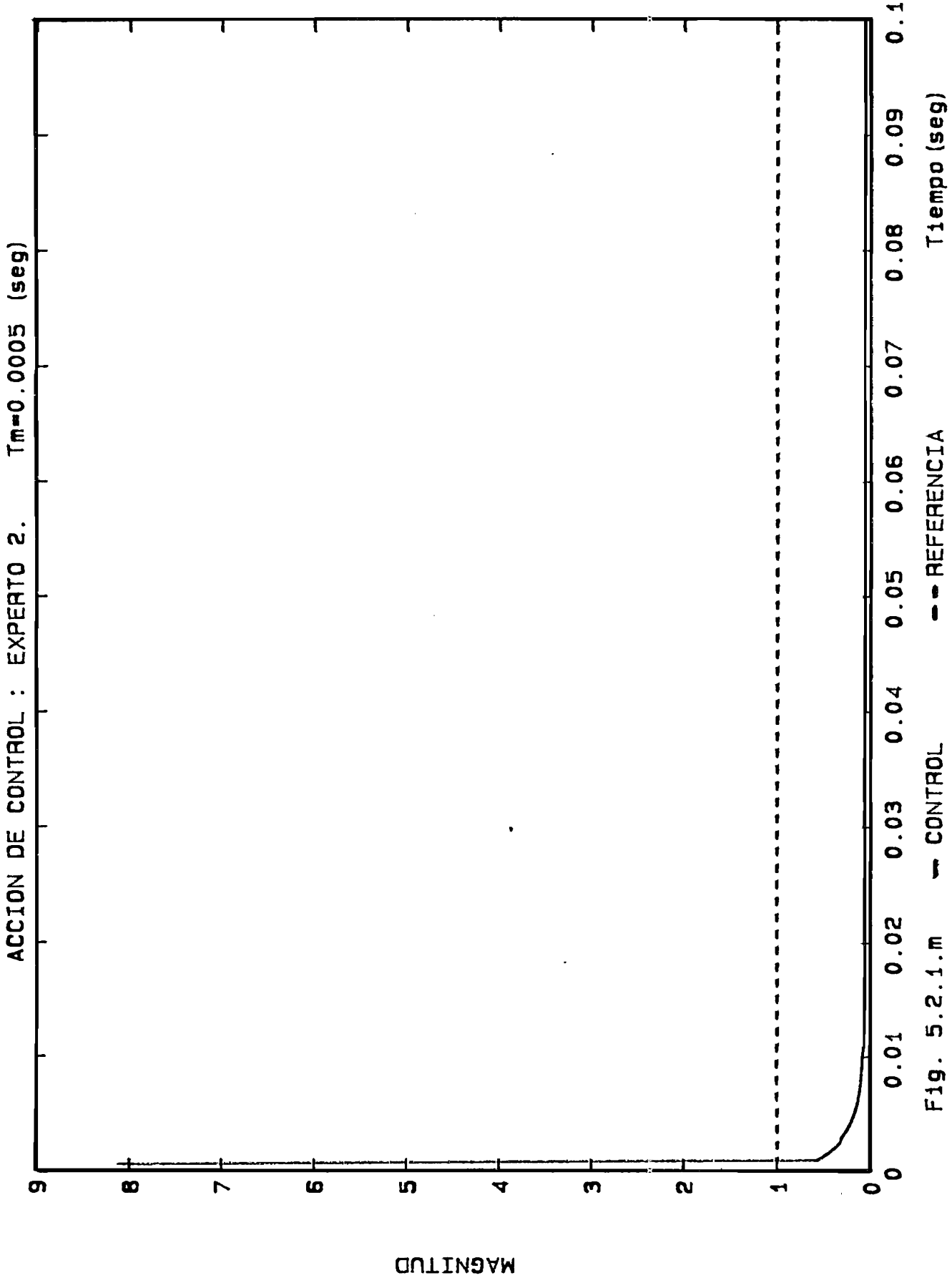


Fig. 5.2.1.j



## 5.2.2 CONTROL DE POSICION.

El control de posición del servomotor se basa en el diagrama de bloques que se muestra en la Fig.5.5 :

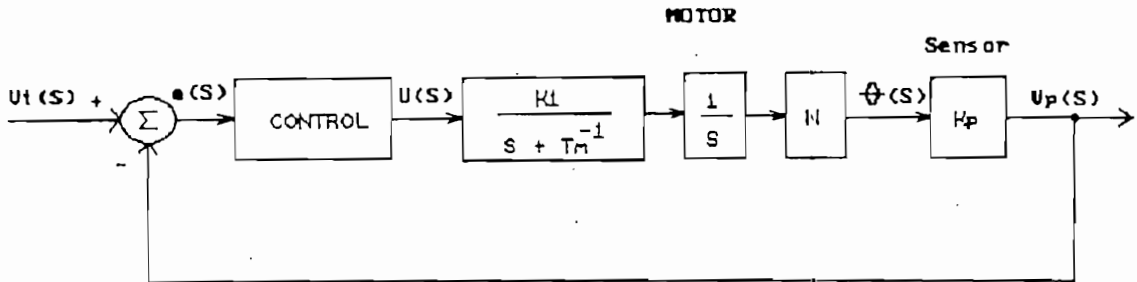


Fig. 5.5: Control de posición del servomotor.

Donde se ha considerado la función de transferencia simplificada del motor, que se indicó en el caso anterior.

Los parámetros que definen el sistema son:

$$\begin{aligned}
 K_1 &= 1265 \\
 T_m^{-1} &= 11.5 \\
 N &= 1/9 \\
 K_p &= 6.74
 \end{aligned}$$

Considerando la ganancia del sensor, la función de transferencia que describe la dinámica del servomotor para el



control de posición es:

$$G_3(s) = \frac{947.37}{s(s+11.5)}$$

Otras constantes que intervienen en simulación del proceso son:

TMUESTREO = 0.005 (seg)

LIMITE = 10

#### 5.2.2.1 CONTROLES.

Utilizando los mismos métodos indicados en los casos anteriores, los parámetros que definen el bloque de control son:

CONTROL PID:

$K_P$  = 2.5

$T_i$  = 0.2

$T_d$  = 0.025

CONTROL SELF TUNING:

Tipo de respuesta: Segundo orden

$\epsilon$  = 0.8

$\omega_n$  = 40

## 5.2.2.2 RESULTADOS.

Se ingresan al programa la función de transferencia del sistema  $G_3(S)$  y los parámetros de los controles, definiendo además una entrada paso unitaria para simular el voltaje de entrada  $V_1$ . La respuesta del sistema está representada por el voltaje  $V_P$ , el cual es proporcional a la posición angular del motor.

Los resultados obtenidos de la simulación del control de posición del servomotor y respuestas a cada uno de los controladores se resume en la siguiente tabla:

	LAZO CERRADO	PID	DEAD BEAT	SELF TUNING	EXPERTO 1	EXPERTO 2
$M_p(\%)$	62.39	4.68	-	13.47	6.21	7.65
$t_p(s)$	0.105	0.075	-	0.060	0.060	0.060
$t_r(s)$	0.060	0.060	0.010	0.035	0.040	0.045
$t_s(s)5\%$	0.640	0.080	0.010	0.110	0.075	0.080
$t_s(s)2\%$	0.665	0.110	0.010	0.130	0.100	0.105
$E_{ap}$	0	0	0	0	0	0

Tabla 5.3: Funcionamiento del control de posición del servomotor.

A continuación se muestran los resultados gráficos obtenidos de la simulación del proceso al aplicar las diferentes acciones de control.

**Fig. 5.2.2.a y Fig. 5.2.2.b**

Ilustra el comportamiento del servomotor al control de posición en lazo abierto y lazo cerrado respectivamente. El sistema se comporta como un típico sistema de segundo orden donde su respuesta en lazo cerrado, alcanza el valor de referencia pero después de un largo tiempo de establecimiento igual a 0.665 (seg), además presenta un sobreimpulso considerable del 62.39%.

**Fig. 5.2.2.c a Fig. 5.2.2.g**

Muestran las diferentes respuestas del sistema al aplicar cada uno de los controles.

**Fig. 5.2.2.h**

Presenta un gráfico comparativo de todas las respuestas del sistema al bloque de controles. En este caso se añade la respuesta del sistema al control EXPERTO 1, la cual permite obtener la mejor operación al control de posición del servomotor. Esta señal se caracteriza por presentar un sobreimpulso del 6.21% y un bajo tiempo de establecimiento igual a 0.075 (seg) con criterio del 5%.

**Fig. 5.2.2.i**

Muestra las señales de error de cada controlador incluyendo el error del control EXPERTO 1, donde se obtiene un comportamiento satisfactorio ya que la respuesta es rápida y el sobreimpulso es bajo.

**Fig. 5.2.2.j**

Presenta la señal de control que es aplicada al proceso. Esta señal posee el máximo valor al inicio de la simulación, debido a la operación del controlador DEADBEAT que es empleado para acelerar la respuesta de la planta. En este momento la acción de control satura el actuador por lo que se debe cambiar a otro controlador que mantenga la velocidad de la respuesta y no sature el sistema, para el efecto se conecta la acción de control SELF TUNING.

**Fig. 5.2.2.k**

Describe el funcionamiento del sistema al utilizar cada uno de los controladores y la acción de control EXPERTO 2. La respuesta obtenida con este último controlador, presenta mejores características que los demás, excepto de la respuesta obtenida con el control EXPERTO 1. Posee un pequeño sobreimpulso igual a 7.65% y un bajo tiempo de establecimiento de 0.08 (seg).

**Fig. 5.2.2.l**

En esta figura se muestra las señales de error del sistema al actuar los tres tipos de controles incluido el error del controlador EXPERTO 2.

**Fig. 5.2.2.m**

Presenta la señal de control EXPERTO 2 que es aplicada al sistema. Esta señal es muy parecida a la acción producida por el control EXPERTO 1. De igual forma el valor generado por el

controlador al cambiar la referencia, sobrepasa el límite del actuador produciendo la saturación del mismo.

En general las respuestas obtenidas por los controles EXPERTO 1 y EXPERTO 2, presentan mejores características manteniendo un tipo de señal rápida con un mínimo sobreimpulso.

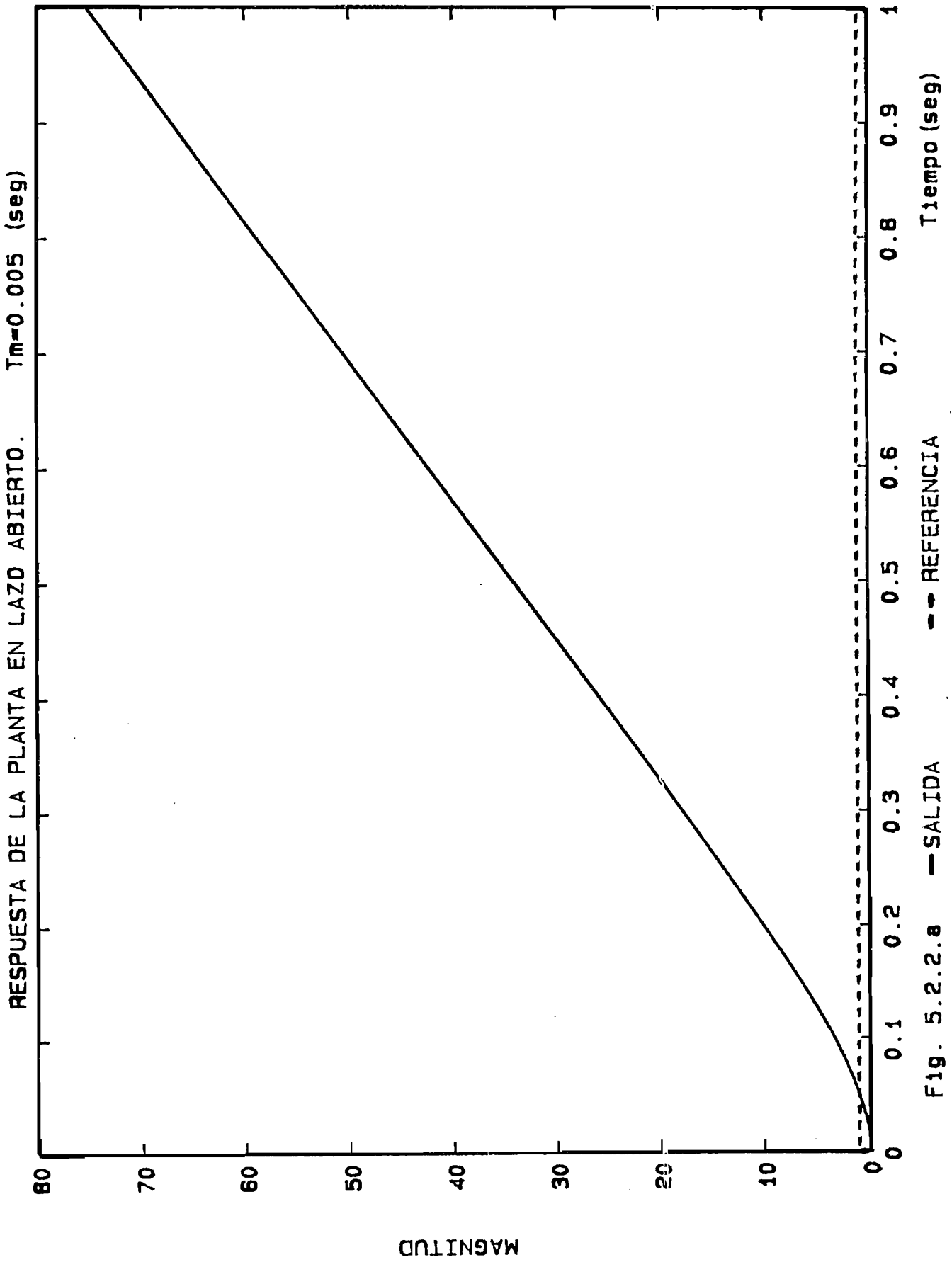


Fig. 5.2.2.8

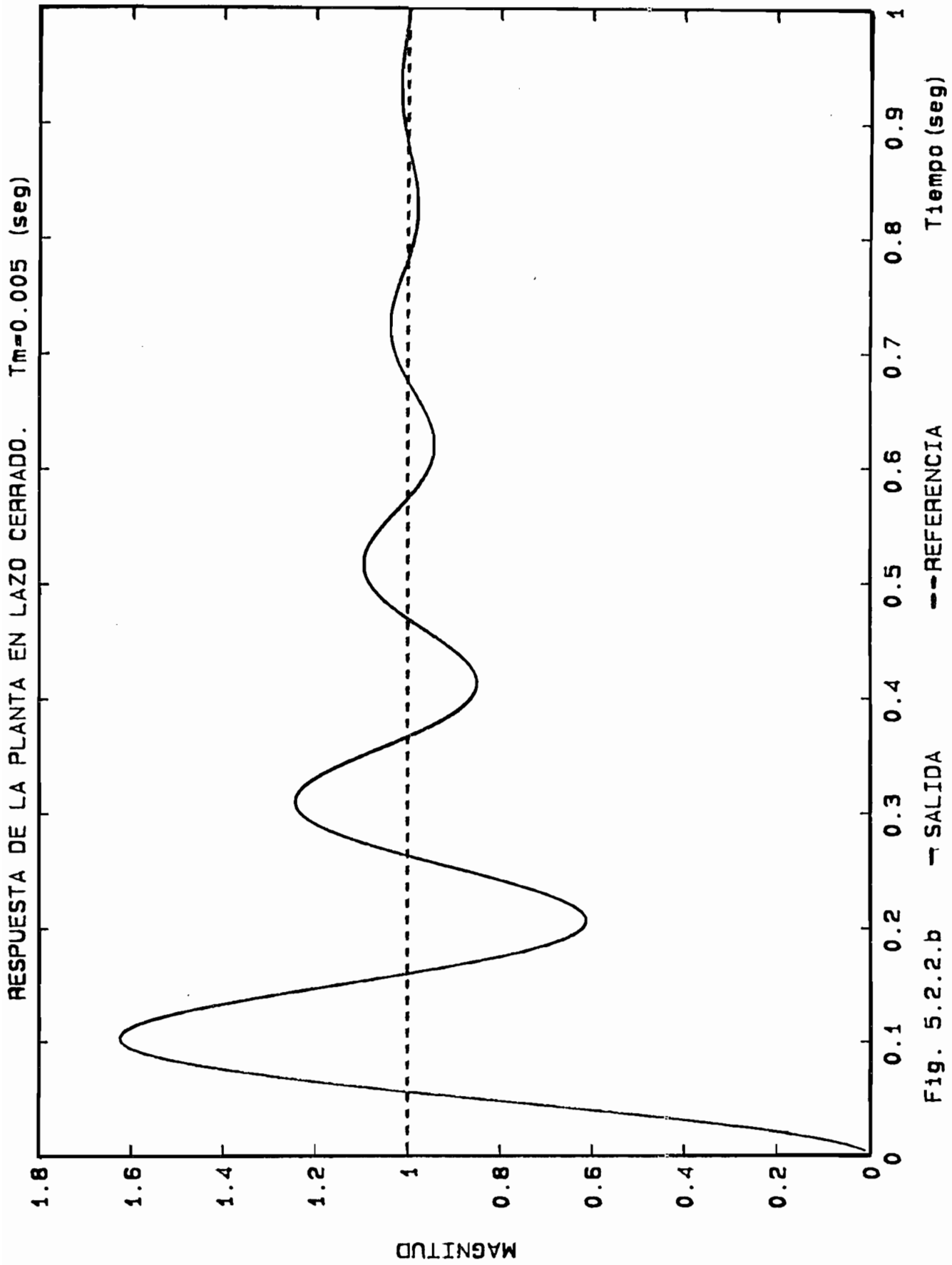


Fig. 5.2.2.b

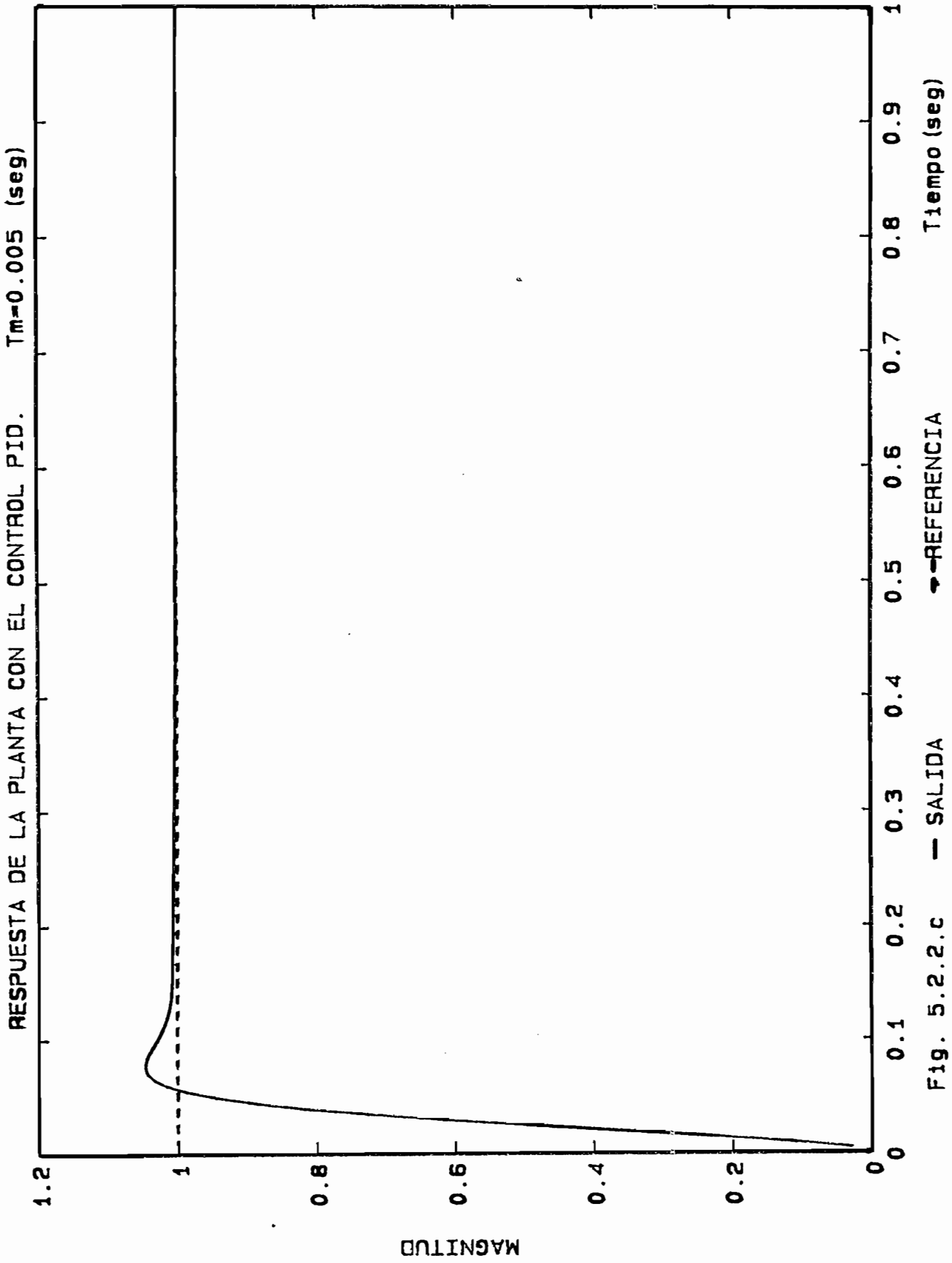


Fig. 5.2.2.c



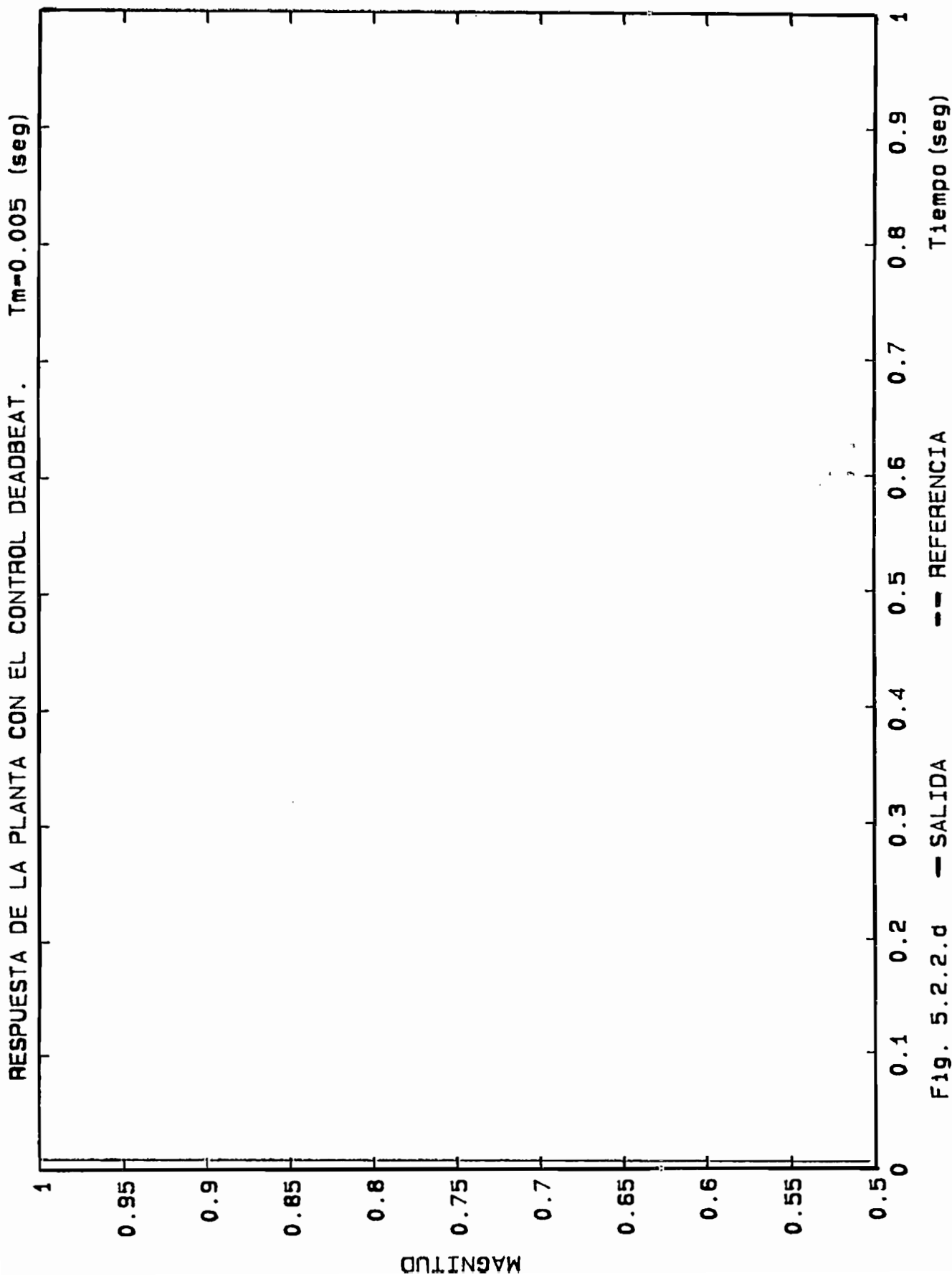
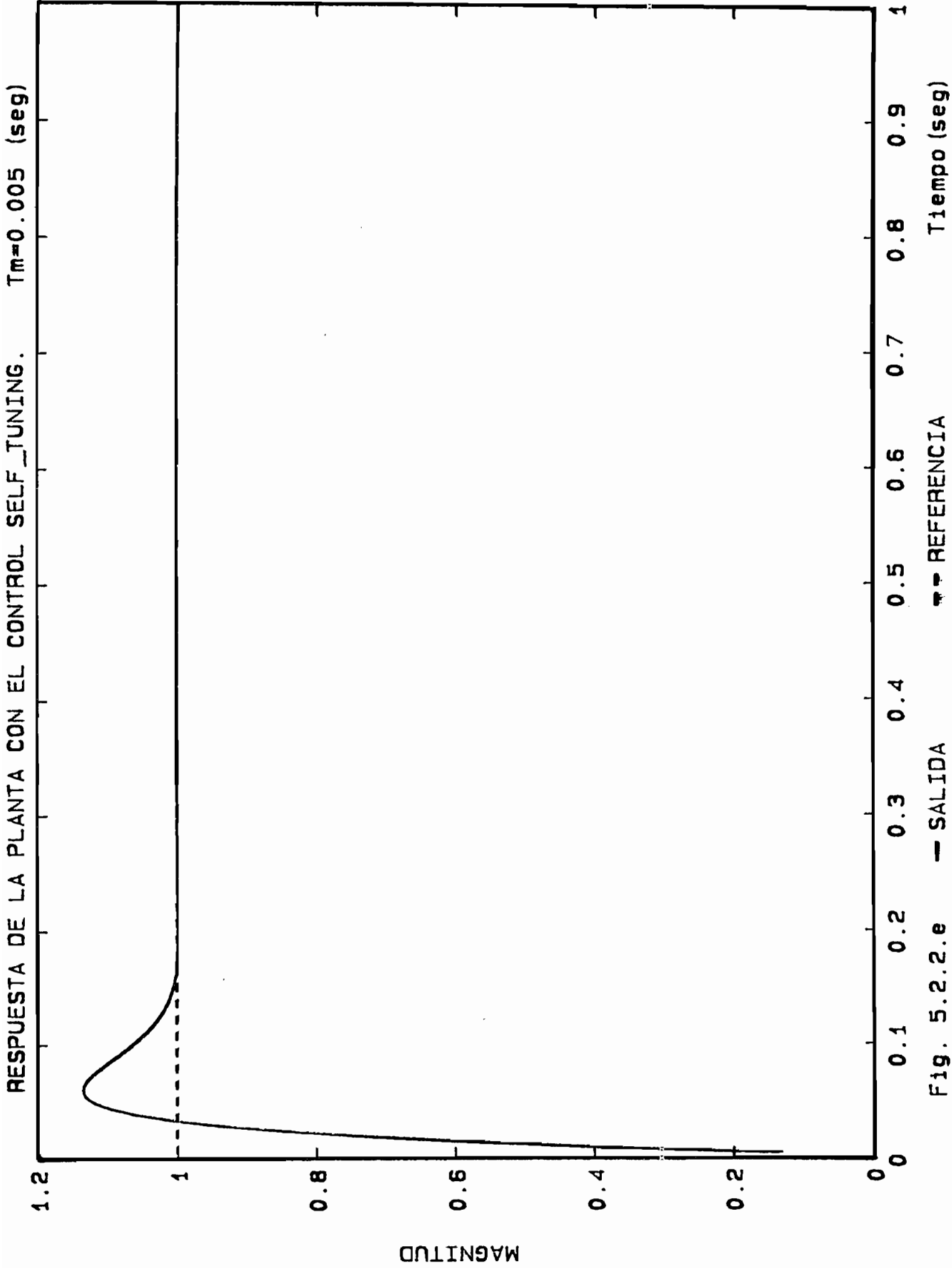


Fig. 5.2.2.d



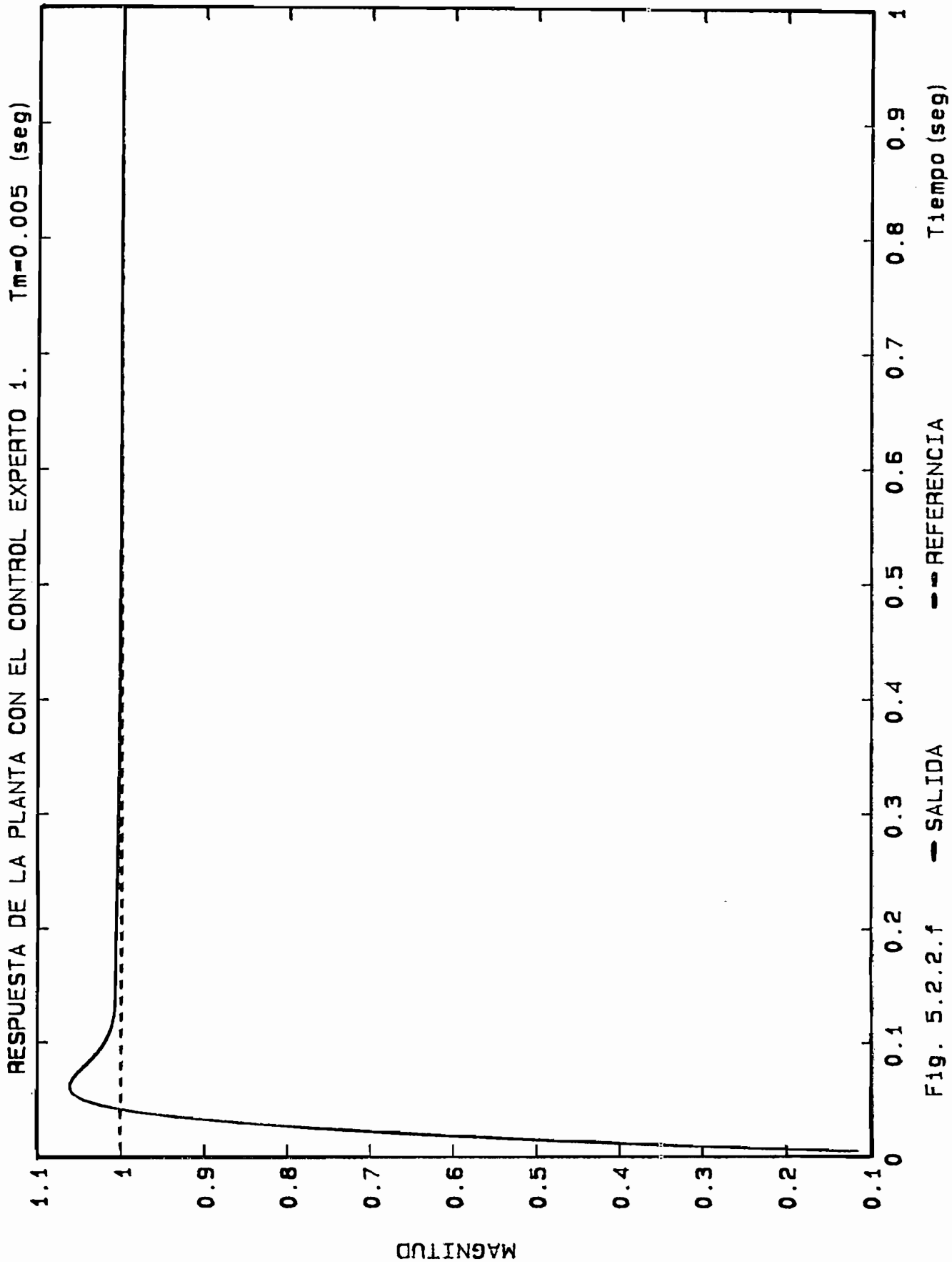


Fig. 5.2.2.f

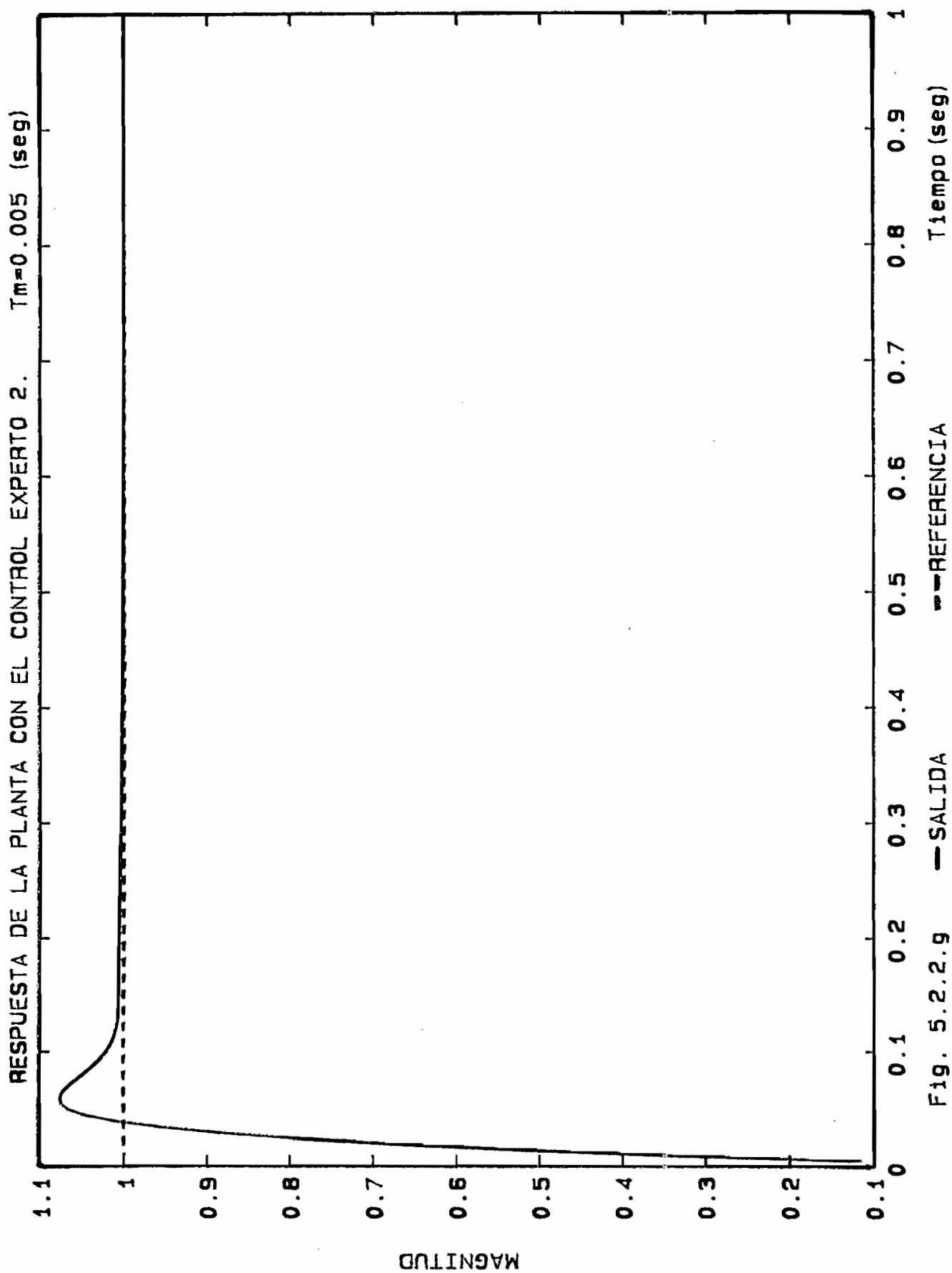


Fig. 5.2.2.9

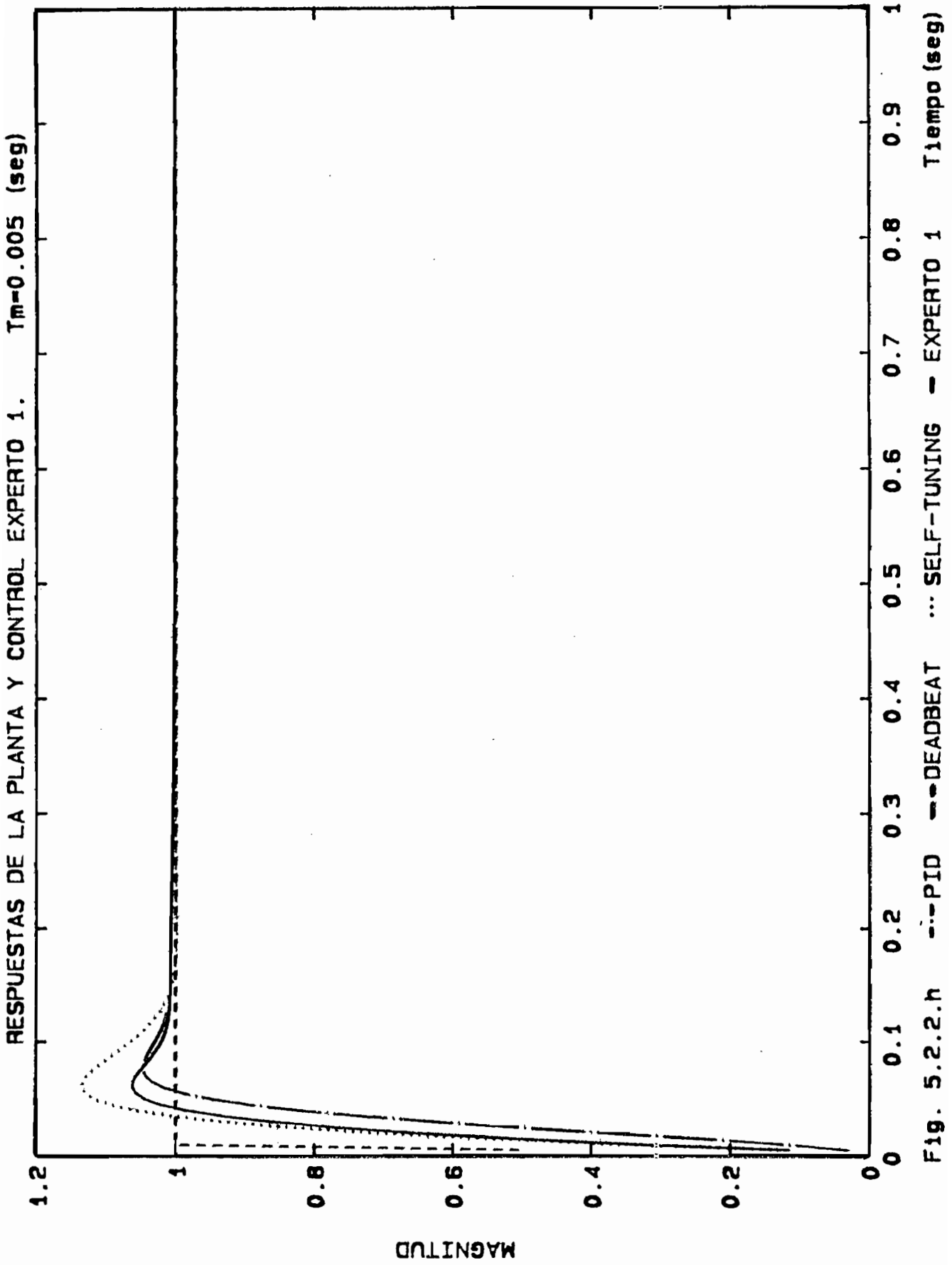
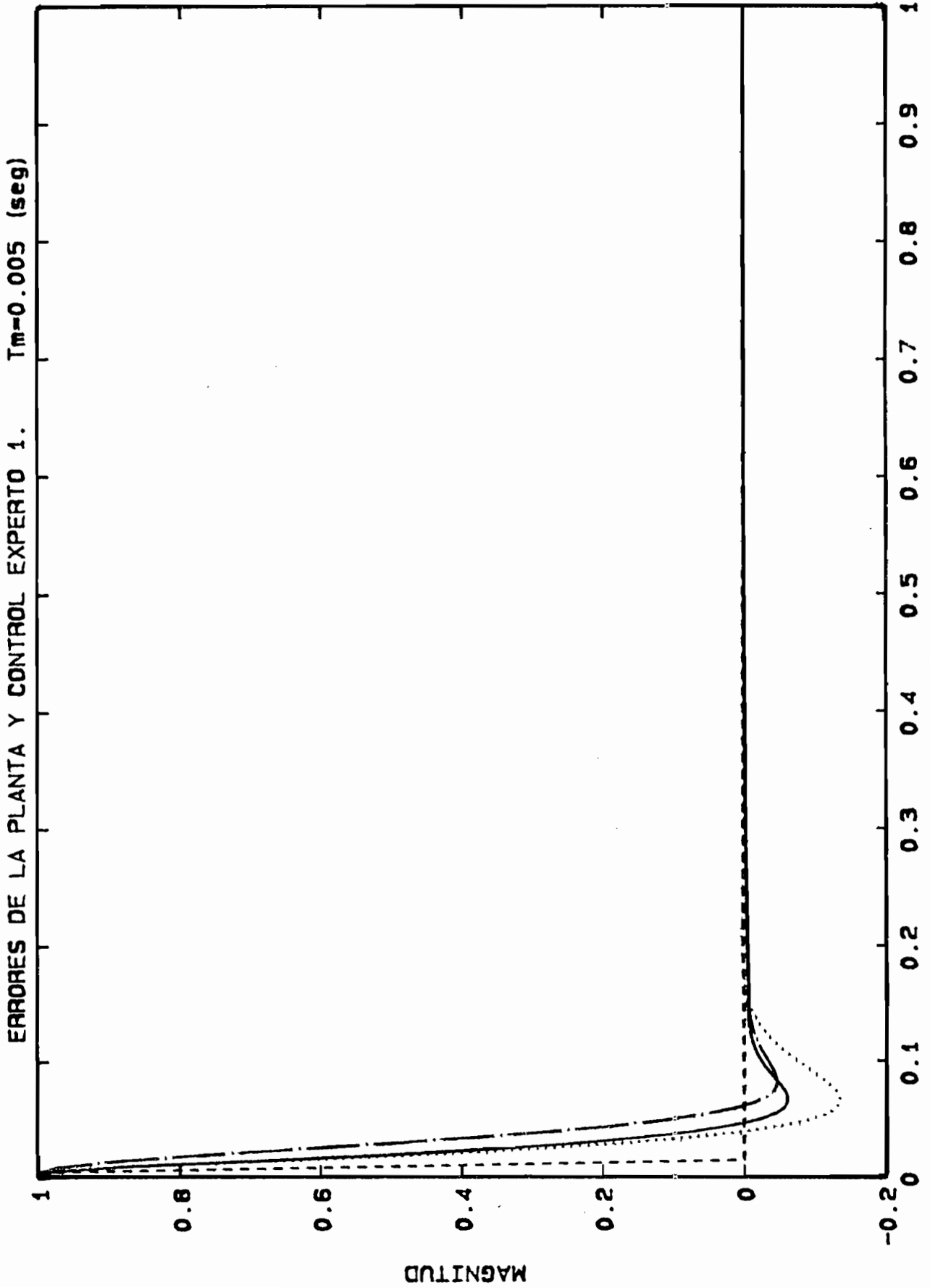
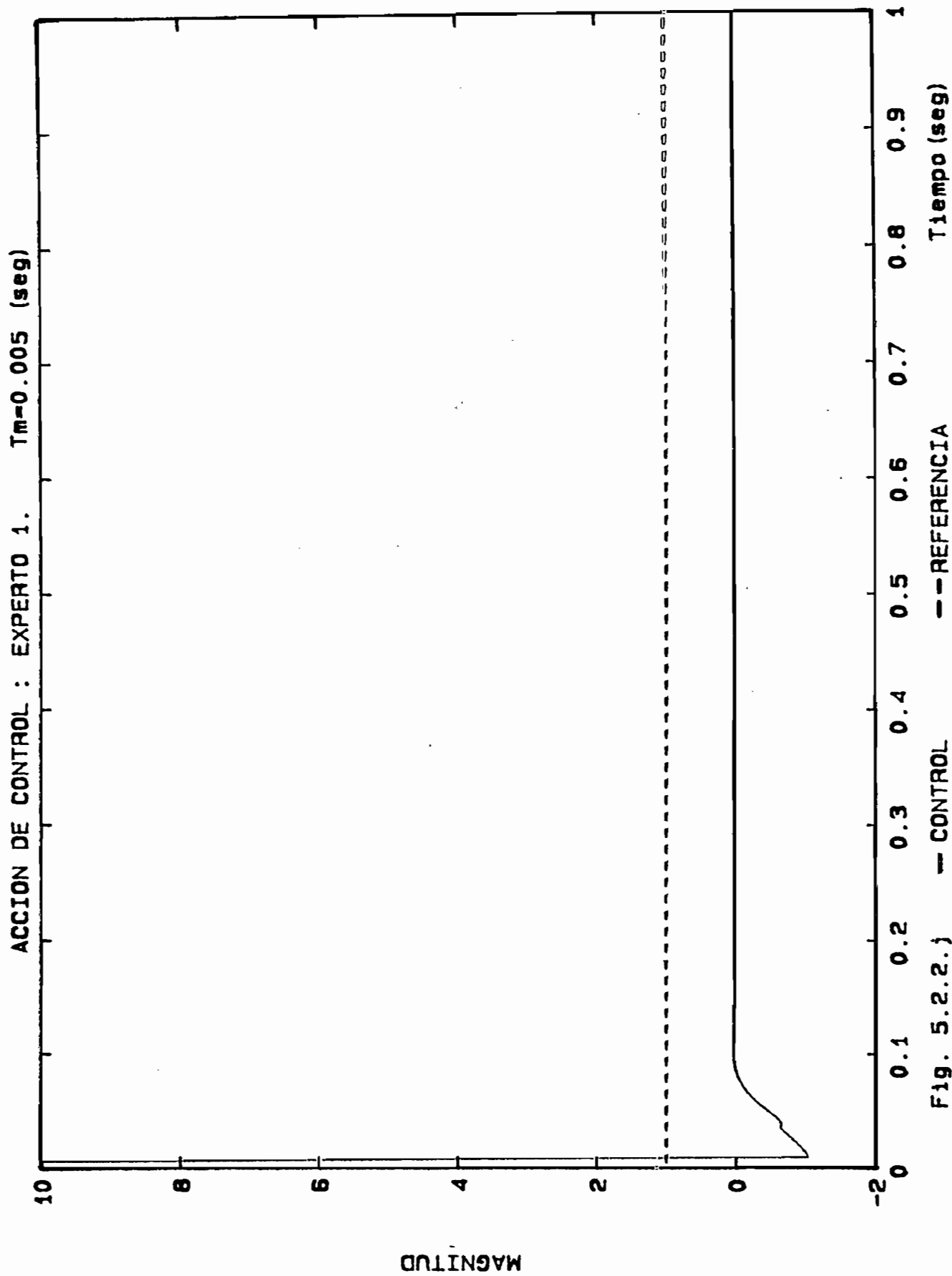


Fig. 5.2.2.h





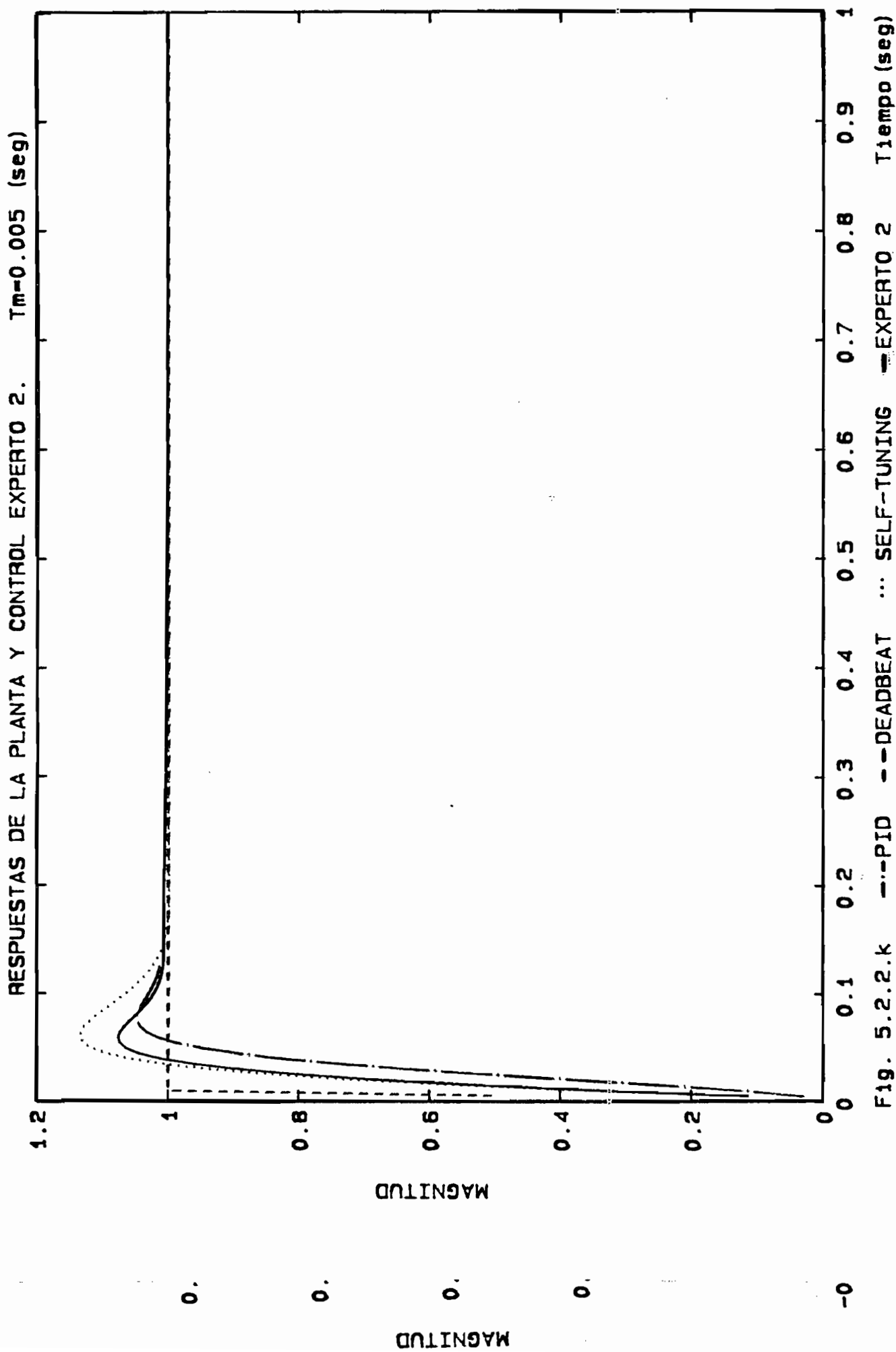


Fig. 5.2.2.k



### 5.3 CONTROL DE UN SISTEMA VARIANTE EN EL TIEMPO.

Muchos procesos industriales sufren variación de sus parámetros iniciales, como resultado de experimentar cambios en sus condiciones de operación, lo que podría degradar el comportamiento del sistema o producir la inestabilidad de los algoritmos de control.

Así por ejemplo un motor DC en una silla de ruedas eléctrica puede experimentar modificaciones en su carga inercial debido a variaciones en el peso y posición de sus pasajeros, considerando que se utilizaría un motor de iguales características y un mismo tipo de controlador en diferentes sillas.

Basados en estos antecedentes y con el objeto de estudiar lo que sucede cuando se altera uno de los parámetros que constituyen la función de transferencia del proceso, se ha procedido a analizar el control de posición del servomotor, descrito en el caso de estudio anterior.

Además se ha considerado un cambio de la señal de referencia que alimenta al sistema. La función de transferencia del proceso por analizar es:

$$G_1(S) = \frac{947.37}{S(S+11.5)}$$

Otras constantes que intervienen en la simulación del proceso son:

```

fac      =   - 0.1
TMUESTRO =   0.005 (seg)
LIMITE   =   10

```

### 5.3.1 CONTROLES.

Empleando los mismos parámetros para el bloque de controles, definidos en el caso anterior se tiene:

CONTROL PID:

```

Kp   =   2.5
Ti   =   0.2
Td   =   0.025

```

CONTROL SELF TUNING:

```

Tipo de respuesta: Segundo orden
ε      =   0.8
Wn   =   40

```

### 5.3.2 RESULTADOS.

Siguiendo el mismo procedimiento, se ingresan al programa la función de transferencia del sistema  $G_4(S)$ , considerando en este caso variación de los parámetros.

Es necesario además ingresar el factor de variación del coeficiente  $a_{n-1}$  de la función de transferencia de la planta. Al tratarse de un sistema de segundo orden el coeficiente que se altera es  $a_1$ , el que contiene el valor del coeficiente de amortiguamiento  $\zeta$  que caracteriza un sistema de segundo orden.

La señal de referencia  $V_1$ , es simulada por una señal de entrada paso unitaria, la que también varía en el tiempo entre valores de 0 a SET\_POINT, coincidiendo el cambio de referencia con la variación de  $a_1$ .

Los resultados obtenidos de la simulación del control de posición del servomotor con variación de su función de transferencia y señal de entrada se resume en la tabla 5.4. Donde se presenta los resultados obtenidos en el segundo ciclo, en el que se produce el cambio de parámetros.

Los resultados del primer ciclo son los mismos de la tabla 5.3.

	LAZO CERRADO	PID	DEAD BEAT	SELF TUNING	EXPERTO 1	EXPERTO 2
$M_p(\%)$	-	10.72	9.10	25.39	7.20	10.17
$t_p(s)$	-	0.060	0.015	0.045	0.055	0.050
$t_r(s)$	0.060	0.040	0.005	0.020	0.035	0.030
$t_s(s)5\%$	-	0.080	0.065	0.110	0.070	0.075
$t_s(s)2\%$	-	0.090	0.135	0.125	0.080	0.085

	LAZO CERRADO	PID	DEAD BEAT	SELF TUNING	EXPERTO 1	EXPERTO 2
Eap	$\infty$	0	0	0	0	0

Tabla 5.4: Funcionamiento del control de posición del servomotor a partir del cambio de la función de transferencia  $G_4(S)$  y señal de entrada.

A continuación se muestran los resultados gráficos obtenidos de la simulación del proceso al aplicar las diferentes acciones de control.

Fig. 5.3.a y Fig. 5.3.b

Ilustra el comportamiento del servomotor al control de posición en lazo abierto y lazo cerrado respectivamente, considerando la variación de uno de sus parámetros  $a_1$  y la señal de entrada.

La respuesta del sistema en lazo cerrado muestra un tipo de respuesta oscilante en el primer ciclo, mientras que en el segundo ciclo se presenta una respuesta oscilante de amplitud creciente, lo que caracteriza a un sistema inestable.

Este tipo de comportamiento es esperado ya que el factor de variación  $fac = -0.1$ , hace que la constante de amortiguamiento ( $\epsilon$ ) cambie su valor original.

Los parámetros originales del sistema de control de posición,

antes de intervenir en el factor de variación son:

$$\epsilon = 0.18$$

$$W_n = 30.77$$

Estos parámetros se los obtiene a partir de la función de transferencia  $G_4(S)$ , que describe la dinámica del proceso a controlar.

A partir del segundo ciclo de operación se da lugar al cambio de la constante de amortiguamiento, la que es afectada por el factor de variación fac. Desde este momento los parámetros que caracterizan al sistema son:

$$\epsilon = -0.018$$

$$W_n = 30.77$$

Con estos parámetros, la nueva función de transferencia del proceso es:

$$G'_4(S) = \frac{947.37}{S(S-1.15)}$$

$G'_4(S)$  describe la dinámica de un sistema inestable, donde las raíces del polinomio característico  $P(S)$  se ubican en el semiplano derecho del plano complejo  $S$ .

## Fig. 5.3.c a Fig. 5.3.g

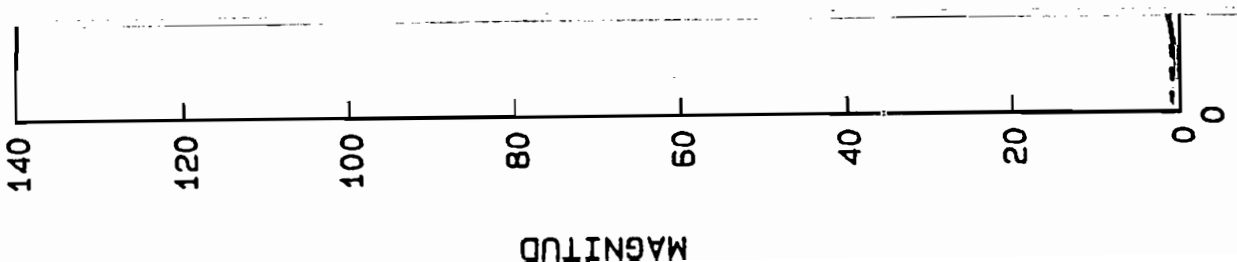
Muestran las diferentes respuestas del sistema al aplicar cada acción de control. En el segundo ciclo, el sobreimpulso de las respuestas a los controles PID y SELF TUNING aumentan con la variación de  $\epsilon$ .

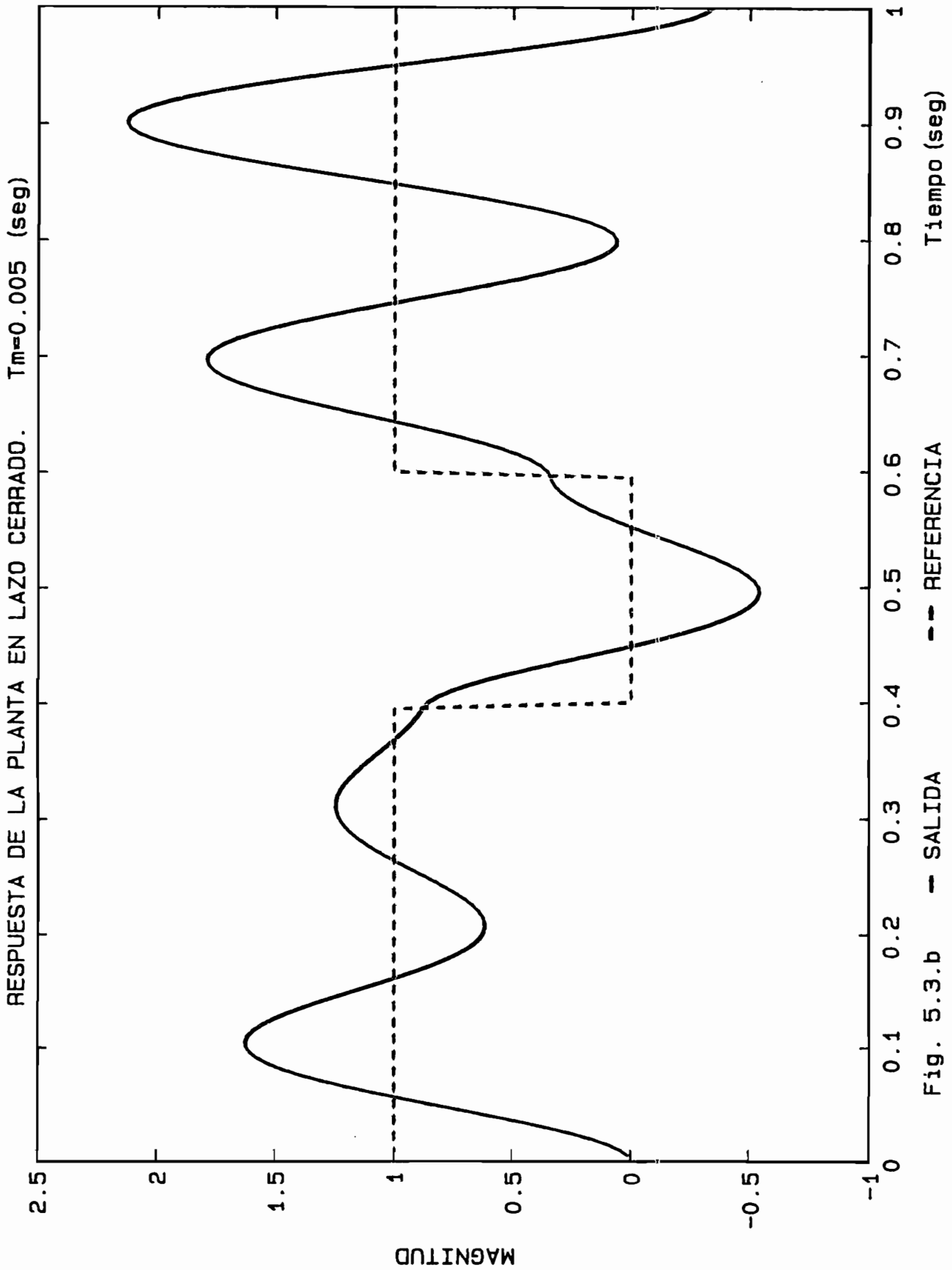
La respuesta del control DEADBEAT es el que presenta mayor distorsión, ya que inicialmente este controlador no poseía sobreimpulso. Este comportamiento se debe a que la señal de control es calculada a partir de los coeficientes de la función de transferencia en el plano Z, cuyos valores ya no son correctos debido al cambio de la función de transferencia  $G(S)$  de la planta.

## Fig. 5.3.h

Presenta un gráfico comparativo de todas las respuestas del sistema al bloque de controles. En este caso se añade la respuesta del sistema al control EXPERTO 1, el cual permite obtener una mejor operación del control de posición del servomotor con cambio de sus parámetros.

Esta señal se caracteriza por presentar el mínimo sobreimpulso en el segundo ciclo de operación, al conectar al proceso la señal obtenida del controlador PID. Los valores obtenidos son del 7.20% de sobreimpulso y un bajo tiempo de establecimiento igual a 0.070 (seg) utilizando el criterio del 5%. El primer ciclo se comporta exactamente igual al caso analizado anteriormente.





RESPUESTA DE LA PLANTA CON EL CONTROL PID.  $T_m=0.005$  (seg)

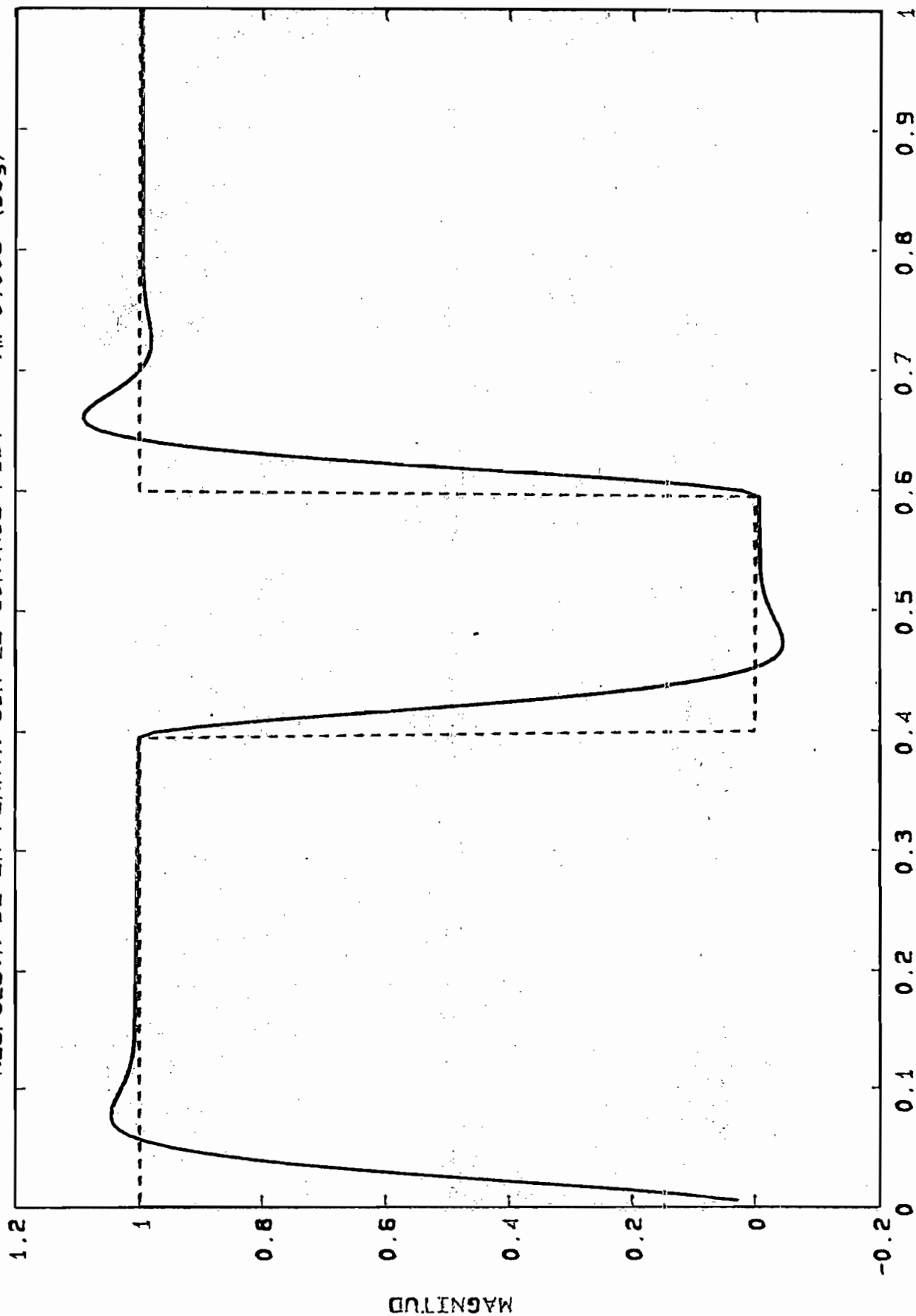


Fig. 5.3.c -- SALIDA -- REFERENCIA



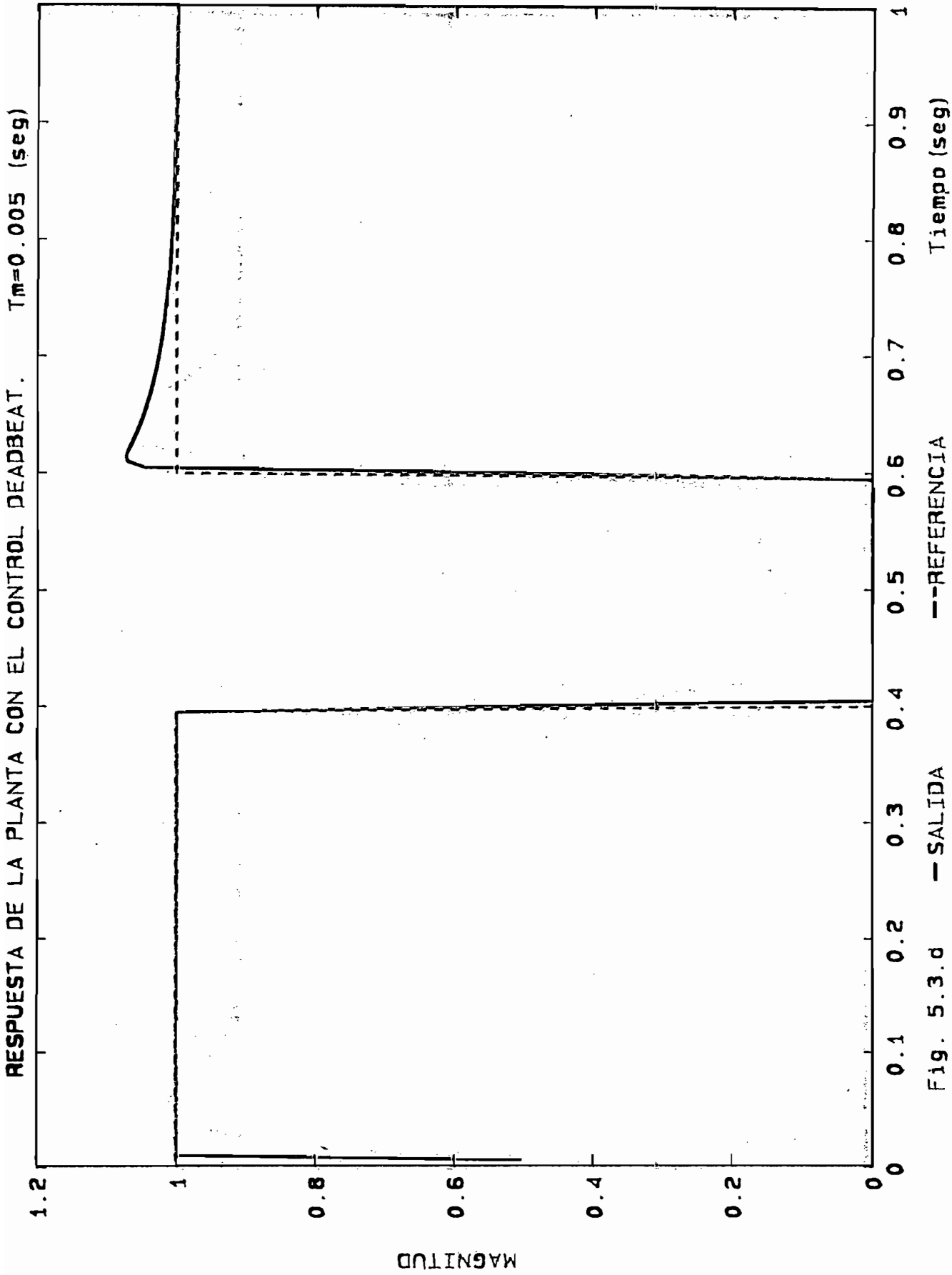
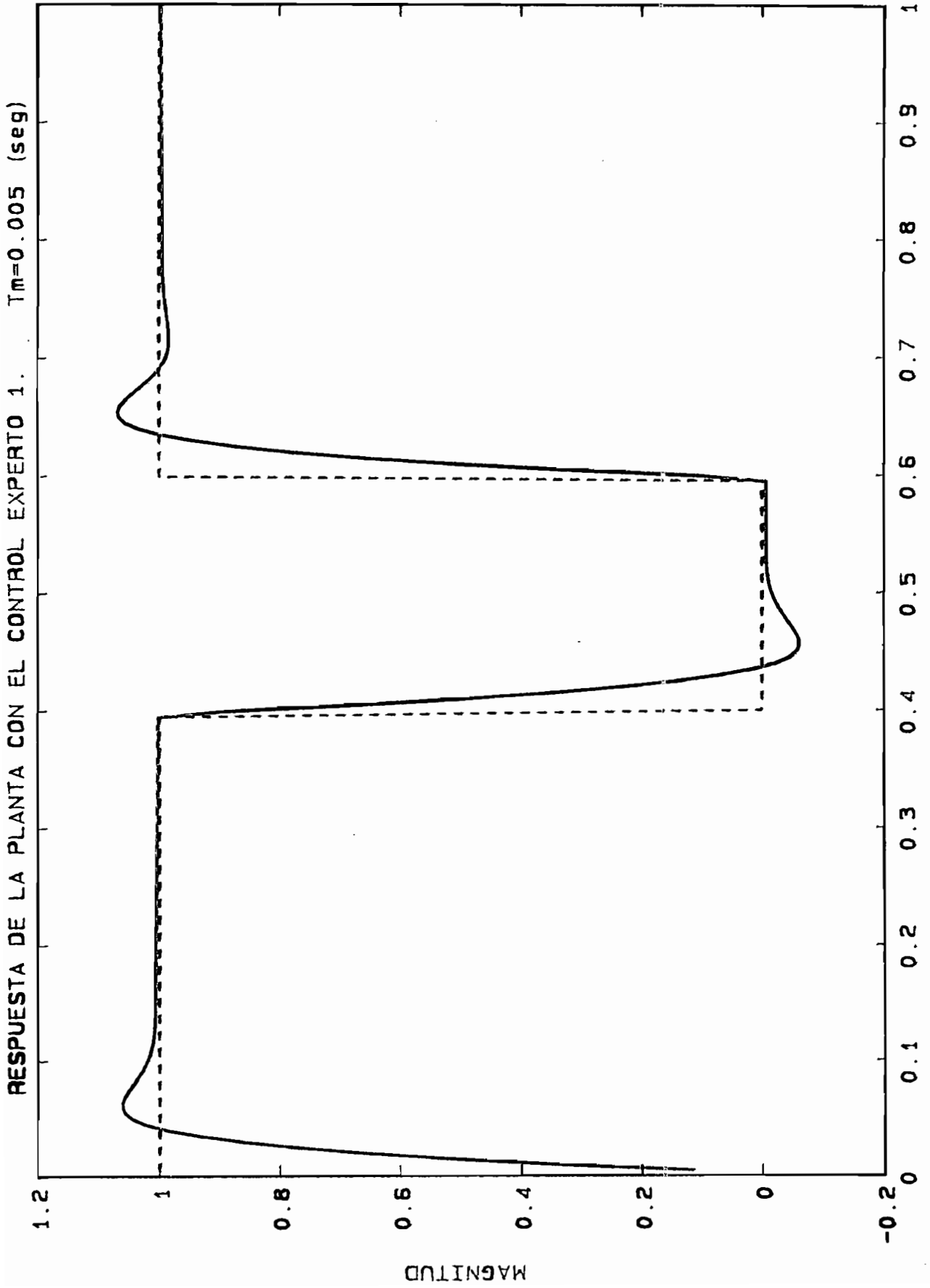


Fig. 5.3.d



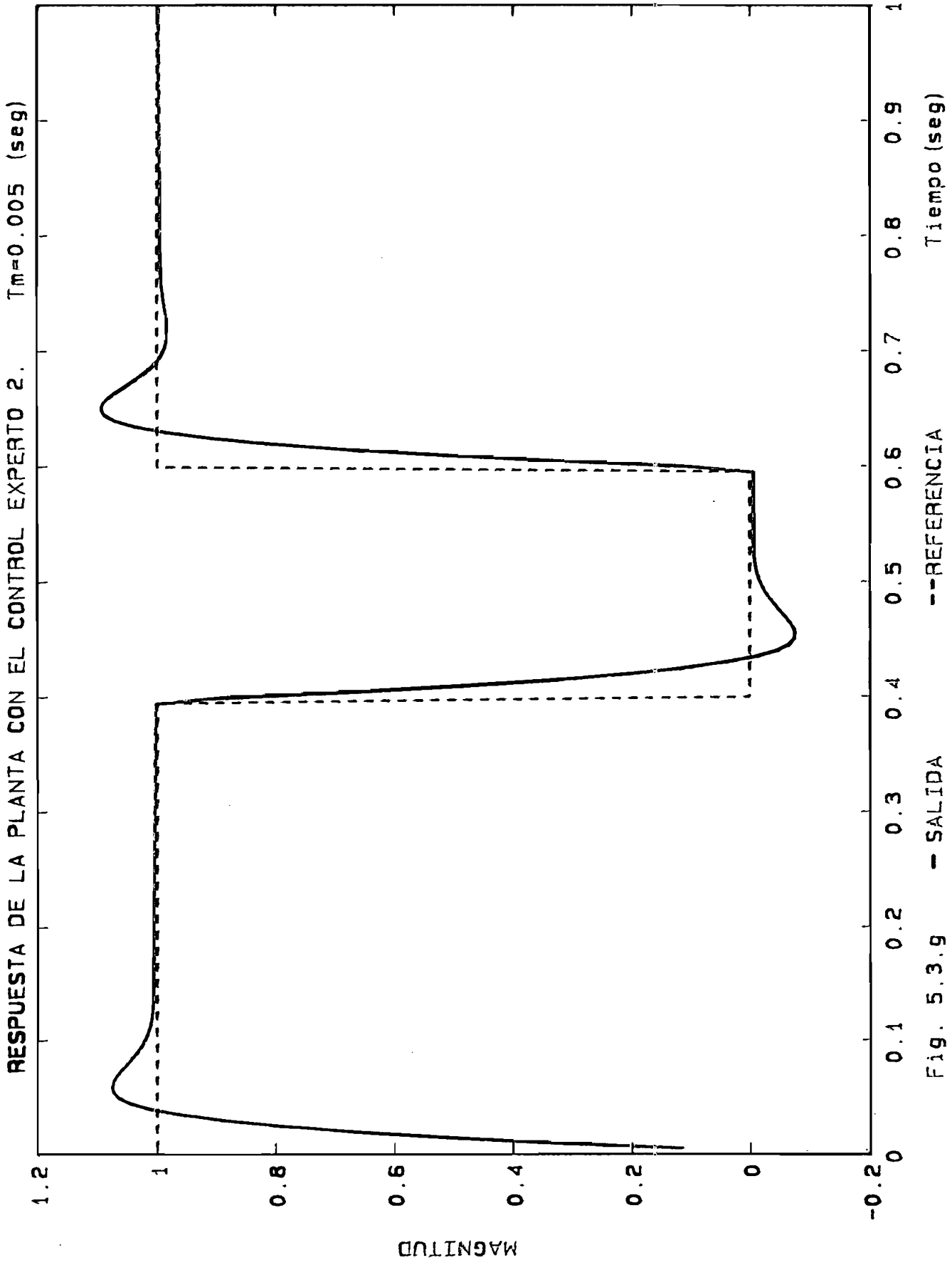
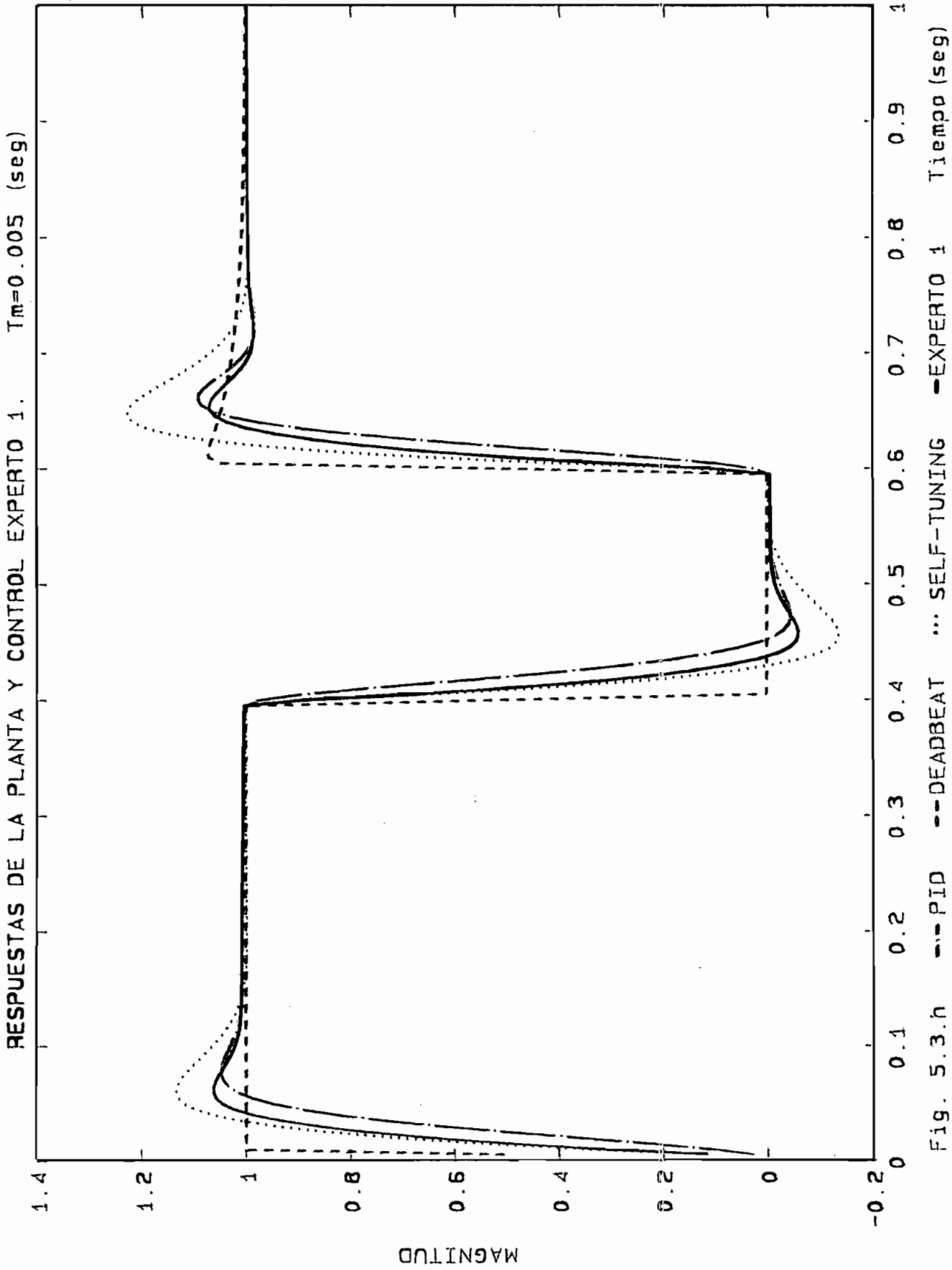


Fig. 5.3.9



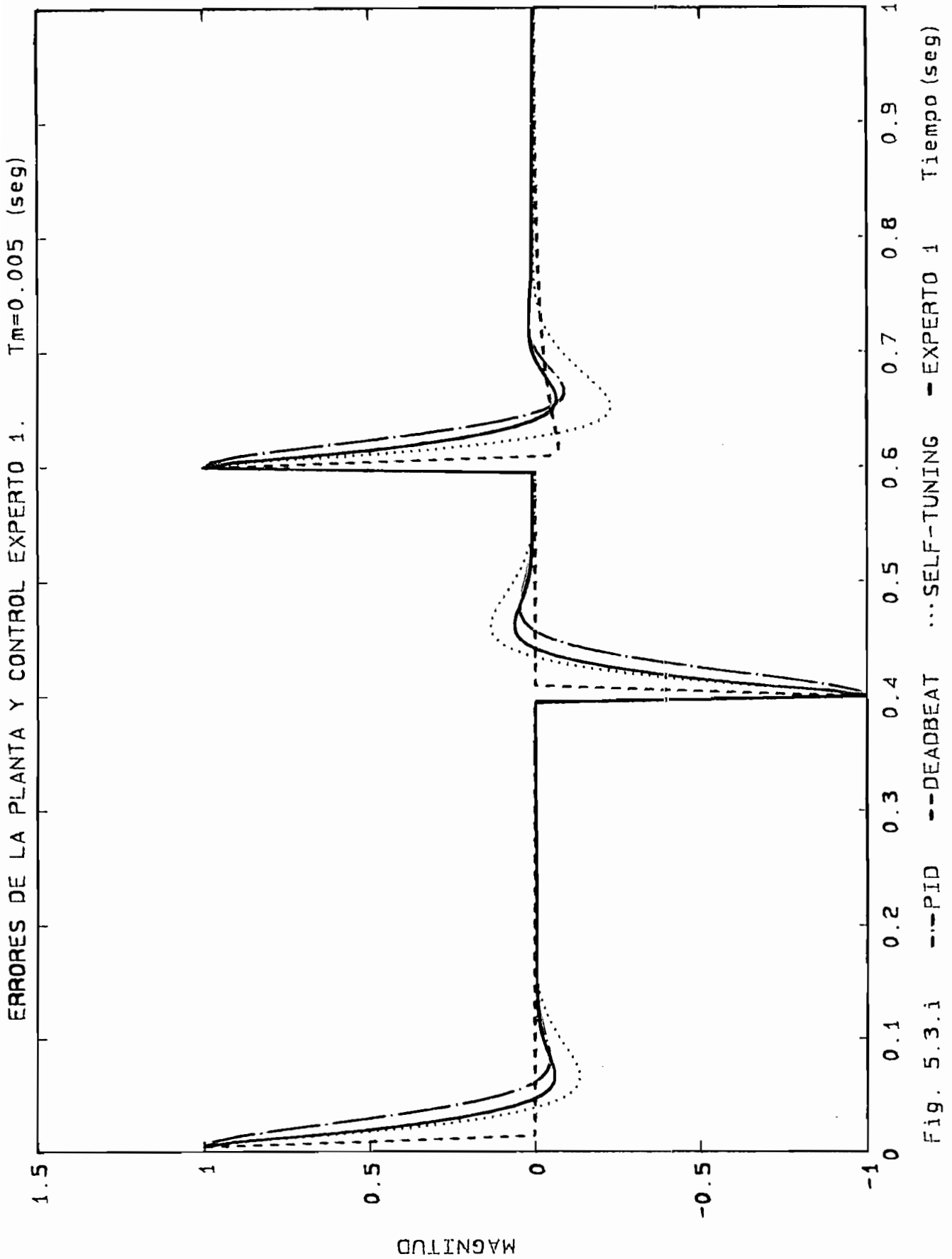


Fig. 5.3.1

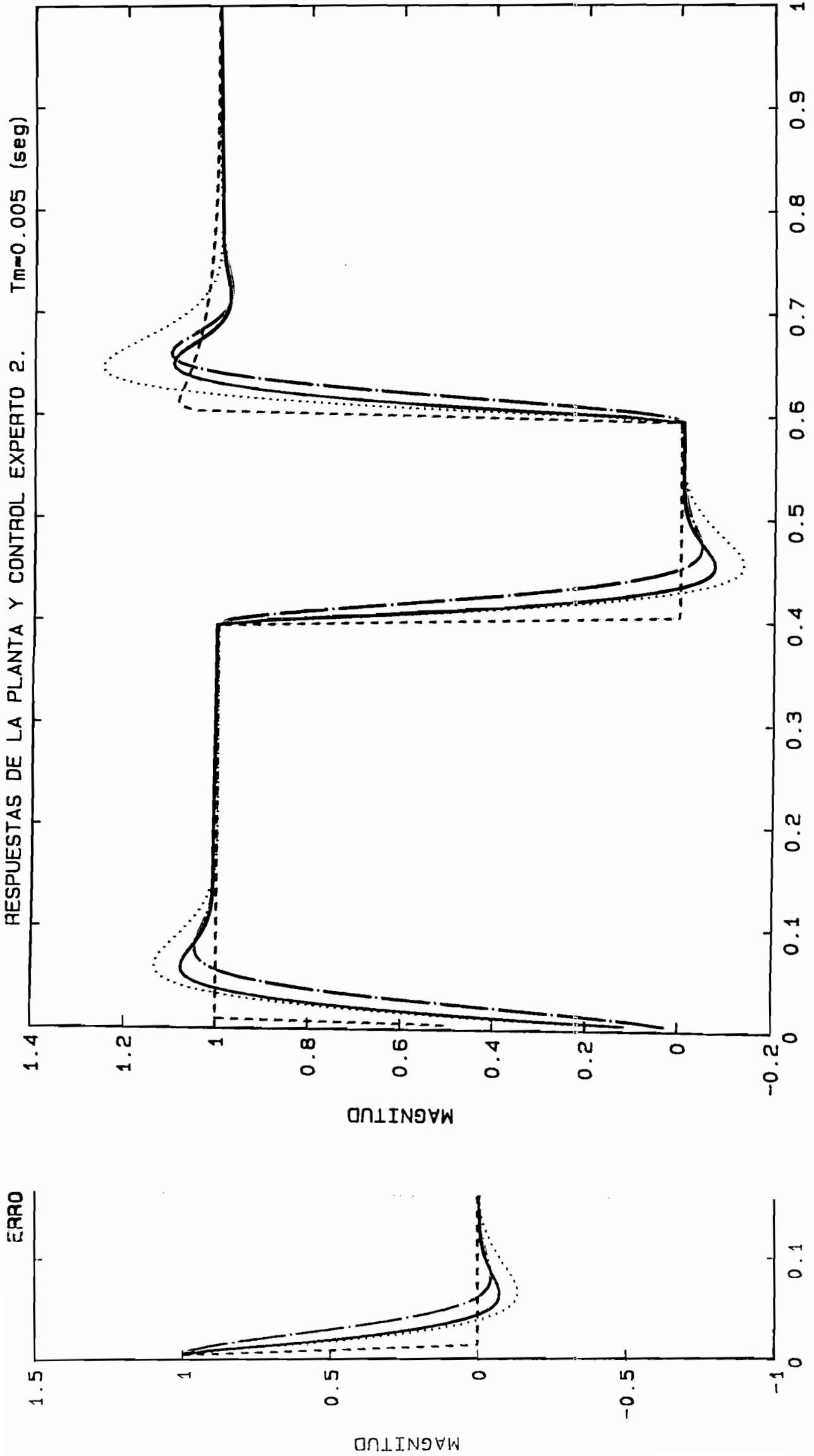


Fig. 5.3.k ---PID ---DEADBEAT ... SELF-TUNING - EXPERTO 2 Tiempo (seg)

Fig. 5.3.1

## CAPITULO VI

### CONCLUSIONES

#### 6.1 EVALUACION DE LA IMPLANTACION DEL CONTROLADOR EXPERTO.

El principal objetivo de este trabajo fue el desarrollo de un Sistema de control basado en conocimiento para controlar satisfactoriamente procesos lineales con y sin retardo de transporte o variantes en el tiempo. El Controlador Experto utiliza algoritmos de control y técnicas de sistemas expertos para producir un controlador robusto.

Para probar el funcionamiento del CE se consideró distintas condiciones de operación, por lo que se realizaron varias simulaciones con el objeto de probar los controles EXPERTO 1 y EXPERTO 2. En esta evaluación se utilizó diferentes parámetros como: orden del sistema, período de muestreo, tiempo de retardo, límite del actuador y algunos diseños para el bloque de controles.

Además se efectuó un análisis comparativo de la operación del sistema en lazo cerrado para diferentes situaciones como:

- Sistema en lazo cerrado con realimentación unitaria.
- Funcionamiento del proceso al utilizar los controles PID, DEADBEAT y SELF TUNING.
- Planta controlada por el control EXPERTO 1.
- Planta controlada por el control EXPERTO 2.

En todos los casos el sistema fue alimentado por una entrada paso unitaria.

Los resultados obtenidos en esta evaluación son:

#### Sistema térmico con retardo de transporte

El sistema en sí mismo no es capaz de seguir el cambio de la referencia, manteniendo un error de estado estable del 50%. Al utilizar los controles EXPERTO 1 y EXPERTO 2, la respuesta del sistema alcanza rápidamente el valor de referencia, reduciendo considerablemente el máximo sobreimpulso obtenido con el control PID.

El controlador DEADBEAT presenta la mejor respuesta del sistema, pero este tipo de control es inaceptable en muchos procesos prácticos. Por esta razón el CE1 y CE2 limitan la señal de control deadbeat, el cual es usado solamente para dar a la planta el impulso inicial al existir un cambio en la referencia.



La respuesta del control SELF TUNING se aproxima mucho a la señal obtenida con el control DEADBEAT, por lo que es utilizada por CE1 y CE2 para mantener la rapidez de la respuesta y eliminación del sobreimpulso.

Control de velocidad de un servomotor.

Se comporta como un sistema de primer orden cuya señal de salida mantiene un error de estado estable igual a 6.79%. Los controles EXPERTO 1 y EXPERTO 2 aplican las mismas reglas que el caso anterior, obteniéndose de igual forma una respuesta muy rápida con un mínimo sobreimpulso.

El control PID produce un comportamiento aceptable, ya que el máximo sobreimpulso es menor a 10% pero su respuesta es lenta. Mientras que el control SELF TUNING produce una respuesta con características de un sistema de primer orden, ya que crece exponencialmente hasta llegar a su valor final eliminando el error en estado estable.

Control de posición de un servomotor.

Corresponde a un típico sistema de segundo orden. La respuesta del sistema en lazo cerrado sin utilizar ningún tipo de control es altamente oscilante, con un elevado valor de sobreimpulso y un largo tiempo de establecimiento.

El control SELF TUNING genera una respuesta con el sobreimpulso de mayor valor que el resto de controles, debido a que el parámetro  $W_n$  fue seleccionado de tal forma que el tiempo de

establecimiento sea menor a 0.125 (seg), lo que hace crecer Mp%. La respuesta del control PID posee el menor sobreimpulso pero produce una señal más lenta. Los controles EXPERTO 1 y EXPERTO 2 permiten obtener la mejor respuesta del proceso. Las que se caracterizan por su rapidez y bajo sobreimpulso, debido a la utilización de la señal del control DEADBEAT para excitar el sistema y el control SELF TUNING para mantener la velocidad de la respuesta, pasando posteriormente al control PID para atenuar el sobreimpulso.

Control de un sistema variante en el tiempo.

En este caso se analiza el comportamiento de un sistema de segundo orden, al cambiar el valor original del coeficiente de amortiguamiento, coincidiendo esta variación de parámetros con el cambio de referencia. El diseño del bloque de controles mantiene los mismos datos con los que se trabajó en el caso anterior.

La respuesta de la planta en lazo cerrado sin utilizar un controlador se convierte en un sistema inestable al producirse el cambio de  $\zeta$  a un valor menor que cero. El control DEADBEAT distorciona la señal de salida del proceso en este ciclo, produciendo un máximo sobreimpulso de 9.10%.

El control SELF TUNING produce una respuesta cuyo sobreimpulso supera el 20% aumentando la rapidez de la señal de salida, esto se debe a que la dinámica del proceso ha cambiado al transformarse en un sistema inestable. Este controlador

identifica un nuevo modelo que caracteriza el sistema, bajo el cual calcula la acción de control que debe actuar sobre el mismo para mantener una respuesta satisfactoria.

La respuesta del control PID presenta una señal de características aceptables, las que son aprovechadas para disminuir el sobreimpulso que genera la acción de control SELF TUNING.

Usando CE1 es posible producir la mejor respuesta para controlar la posición de un servomotor, donde el sistema sufre variaciones en sus parámetros. El máximo sobreimpulso y rapidez de respuesta entre el primer y segundo ciclo presentan un ligero incremento, aunque el proceso haya sido alterado.

La señal de salida del CE2 en el segundo ciclo de operación, se mantiene dentro de las características deseadas aunque el valor del sobreimpulso sea mayor que el obtenido por el CE1, de todas formas es inferior al 20%, con un bajo tiempo de establecimiento.

Considerando los resultados evaluados en esta sección se puede concluir que el funcionamiento de los procesos al utilizar los controles EXPERTO 1 y EXPERTO 2, presentaron mejores características que la mejor respuesta obtenida al emplear un controlador en particular. El sistema experto extrae las mejores características de cada control para conectar al proceso en el momento adecuado, generando de esta forma un

controlador robusto que mantiene una correcta operación de la planta.

## 6.2 CONCLUSIONES.

La capacidad del controlador experto basado en la conmutación de diferentes algoritmos de control y asignando factores de ponderación a los controladores fueron estudiadas y comparadas con el funcionamiento que presenta cada controlador.

Las conclusiones generales que resultaron de esta evaluación son las siguientes:

- El sistema de control experto basado en conocimiento permitió obtener un controlador robusto, capaz de controlar adecuadamente procesos lineales, combinando técnicas de auto ajuste, supervisión y un conjunto de algoritmos de control. De esta forma se desarrolló una nueva generación de sistemas de control con resultados efectivos.
- La complejidad del sistema basado en conocimiento depende principalmente de la naturaleza y tipo de planta, el número de algoritmos que conformen el bloque de control y los requerimientos del sistema.
- La base de conocimiento bajo la cual trabaja el

controlador experto, puede estar sujeta a cambios y modificaciones. Esto se debe a que las reglas que la conforman fueron obtenidas principalmente como resultado de un estudio experimental, donde se simuló el funcionamiento de varios procesos y cada algoritmo de control. Por esta razón la información empleada para la elaboración de la base de conocimiento, no está libre de juicios subjetivos para seleccionar el tipo de control que debe aplicarse a la planta o los factores de ponderación que son asignados a los mismos.

- La operación del controlador experto no puede ser analizada matemáticamente, ya que el sistema experto no trabaja como resultado de aplicar un algoritmo en particular, sino seleccionando reglas de una base de conocimiento, que en este caso permitirá obtener una acción de control apropiada para conectar al proceso en cada período de muestreo.
- El controlador experto desarrollado genera un tipo de respuesta que se caracteriza por ser más rápida y presentar un menor sobreimpulso que los algoritmos que forman el bloque de control. Esta diferencia se debe a que el controlador experto CE1 y CE2 intenta capturar las mejores características de cada control.
- El diseño del controlador experto no es independiente de sus algoritmos numéricos. Ya que si el cálculo de los

parámetros de los controles a utilizarse no son correctos, el control experto basado en conocimiento fallará al aplicar acciones de control defectuosas.

- En el desarrollo del sistema de control experto se debe superar dos problemas principales para conseguir un controlador robusto que supervise un proceso. El conocimiento que es necesario adquirir y la manera conveniente de representar este conocimiento.
- El programa elaborado posee una estructura modular, que se divide en la interfaz de usuario, algoritmos de control, algoritmos de identificación, simulación del proceso y base de conocimiento. Esta estructura del sistema hizo posible desarrollar y probar individualmente cada módulo. De esta forma el programador puede cambiar y mejorar cualquier módulo, sin necesidad de recompilar o actualizar otros más.
- El sistema de control basado en conocimiento automatiza las capacidades de un competente ingeniero de control. Al razonar y seleccionar las adecuadas estrategias de control de acuerdo al funcionamiento del proceso.
- Para propósitos de un simple control, la técnica de encadenamiento hacia adelante es suficiente para implantar el motor de inferencia, que decide la regla que debe ser ejecutada por el sistema experto. Esta consideración

simplifica la tarea de programación.

- Con el objetivo de conseguir una verdadera automatización del proceso con un control robusto, es conveniente combinar las técnicas de control convencional y las técnicas de sistemas expertos.

### 6.3 RECOMENDACIONES.

El presente trabajo es una introducción a la utilización de técnicas de sistemas expertos para el control de procesos industriales, pero existen algunos aspectos por investigarse que podrían realizarse en futuros proyectos. Algunas sugerencias para futuras expansiones y mejoras de este trabajo se indican a continuación:

- Una de las limitaciones del presente estudio es la clase de sistema utilizado para controlar. Como se definió inicialmente, se trata de procesos lineales. Sin embargo, sistemas no lineales son muy comunes en el mundo real. La base de conocimiento puede necesitar algunas modificaciones y mejoras para controlar este tipo de procesos de una forma adecuada.
- La adquisición del conocimiento se basó en la señal de error del sistema como función del tiempo, pero otras técnicas pueden emplearse como por ejemplo la respuesta de

frecuencia.

- El funcionamiento del sistema de control basado en conocimiento podría ser estudiado no solamente para una entrada paso unitaria, sino también para otro tipo de excitaciones como: tren de pulsos, sinusoidal, rampa, etc.
- Otras características importantes podrían ser consideradas en futuros análisis, como son la presencia de ruido y perturbaciones que afectan a la mayoría de procesos en tiempo real.
- La arquitectura del controlador podría ser extendida para sistemas multivariables. Una librería para realizar tareas en forma concurrente podría incluirse para obtener una configuración en paralelo. Esta característica es importante especialmente cuando se trabaja en tiempo real.
- Se recomienda para futuras extensiones, orientar el presente trabajo con el objeto de controlar procesos en tiempo real. Donde el controlador experto supervise el correcto funcionamiento de la planta cuya información sea obtenida de sensores, operando continuamente y trabajando bajo restricciones impuestas por el proceso.
- Para aprovechar las técnicas de identificación incluidas en este proyecto, se considera importante que en futuras ampliaciones se utilice estos algoritmos para diseñar un



control PID adaptivo, cuyos parámetros de diseño se actualicen en cada período de muestreo. Esto permitiría aplicar acciones de control PID que se ajustan correctamente a los cambios y perturbaciones en un proceso industrial.

- Para efectos de estudio y adquirir la experiencia y conocimiento necesarios de un tipo de proceso industrial, se podría realizar la simulación de la planta en tiempo real. Esta simulación se ejecutaría en un computador, mientras que el bloque de controles y sistema experto serían implantados en otro computador. Esta técnica permitiría observar el comportamiento de dichos sistemas sin utilizar la planta real.

## BIBLIOGRAFIA

- [1] Mishkoff, Henry C., *Understanding Artificial Intelligence*, 2da ed., Howard W. Sams & Company, Indianapolis, 1988.
- [2] Rich, Elaine, *Artificial Intelligence*, Mc Graw Hill Book Company, New York, 1983.
- [3] Garnham, Alan, *Artificial Intelligence An Introduction*, Routledge & Kegan Paul, Londres, 1988.
- [4] Cura, A., López, F. y Navarro, C., *Introducción a la Inteligencia Artificial y Programación en Prolog*, Universidad Tecnológica Nacional, Córdoba, 1987.
- [5] Turban, Efraim, *Decision Support and Expert Systems*, Macmillan Publishing Company, New York, 1988.
- [6] Schildt, Herbert, *Utilización de C en inteligencia artificial*, McGraw-Hill, Madrid, 1990.
- [7] Sawyer, B., and Foster, D., *Programming Expert Systems in Pascal*, A Wiley Press Book, New York, 1986.
- [8] Fierro, Rafael, *An Expert System for Supervisory Process Control*, Msc. Dissertation, University of Bradford, 1990.

- [9] FUINCA, *Inteligencia Artificial para gestión de bases de datos*, Librograf, Madrid.
- [10] McGhee, J., Grimble, M. and Mowforth, P., *Knowledge - Based Systems for Industrial Control*, Peter Peregrinus, IEE, Londres.
- [11] Johannsen, G. and Alty, J., Knowledge Engineering for Industrial Expert Systems, *Automatica*, 27, No. 1, pp. 97-114, 1991.
- [12] Fierro, Rafael, *Sistemas Expertos aplicados a la industria*, Escuela Politécnica Nacional, Quito, 1992.
- [13] William, S., Applications of AI in Engineering, *Computer*, IEEE, pp. 17-27, julio 1986.
- [14] Hayes, Frederick, The Knowledge-Based Expert System: A Tutorial, *Computer*, IEEE, pp. 11-28, septiembre 1984.
- [15] Hong, S., Guest Editor's Introduction, *Computer*, IEEE, pp. 12-15, julio 1986.
- [16] Remmele, W. and Ueberreiter, B., Systematic Knowledge Acquisition in Expert Systems, *Siemens Review. R&D Special*, pp. 9-14, Spring 1991.
- [17] Doraiswami, R. and Jiang, J., Performance Monitoring in

Expert Control Systems, *Automatica*, 25, No. 6, pp. 799-811, 1989.

[18] Aström, K. J., Toward Intelligent Control, *IEEE Control Systems Magazine*, pp. 60-64, abril 1989.

[19] Arzen, K.-E., An Architecture for Expert System Based Feedback Control, *Automatica*, 25, No. 6, pp. 813-827, 1989.

[20] Efstathiou, J., Expert Systems, fuzzy logic and rule-based control explained at last, *Trans Inst MC*, 10, No. 4, pp. 198-206, 1988.

[21] Chantler, M. J., Real-time aspects of expert systems in process control, *Colloquium on Expert Systems in Process Control*, IEE, Savoy Place, Londres, 1988.

[22] Krijgsman, A. J., Verbruggen, R. B., and Bruijn. P. M., *Knowledge-based real-time control*, Delft University of Technology, Department of Electrical Engineering, pp. 7-13.

[23] Laffey, Thomas, The Real-Time Expert, *BYTE*, pp. 259-264, enero 1991.

[24] Bernard, J. A., Use of a Rule-Based Systems for Process Control, *IEEE Control Systems Magazine*, 8, No. 5, pp. 3-

13, octubre 1988.

- [25] Clark, J., *Microsoft Quick C Programmer's Toolbox*, Microsoft Press, Redmond, 1990.
- [26] Conte, S. D., Carl, B., *Análisis Numérico Elemental*, 2da ed., México, 1980.
- [27] Puga, P., *Simulación dinámica de una columna de destilación*, Tesis de Grado, Escuela Politécnica Nacional, Quito, 1991.
- [28] Nicholson, H., *Modelling of dynamical systems*, Peter Peregrinus, Londres, 1980.
- [29] Takahashi, *Control and Dynamics Systems*, Adison-Wesley, 1970.
- [30] Clarke, D. W., PID Algorithms and their computer implementation, *O.U.E.I. Report No. 1482/83*, University of Oxford, Departament of Engineering Sciencie, 1984.
- [31] Thompson, S., McCluskey E. G., An Expert Adaptive PID Controller, *IFAC Symp. on Adaptive Control and Signal Processing*, pp. 553-558, Glasgou, 1989.
- [32] Isermann, Rolf, *Digital Control Systems*, Springer Verlag, Berlin, 1981.

- [33] Åström, K. J. and Witternmark, *Computer Controlled Systems, Theory and Design*, pp. 174-196, Prentice-Hall, Englewood, 1984.
- [34] Hansen, P. D. and Kraus T. W., *Expert Systems and Model-Based Self-Tuning Controllers*, pp. 216-229,
- [35] Wellstead, P.E., Self-tuning digital control systems, *Computer Control of Industrial Processes*, IEE Control Engineering Series 21, Peter Peregrinus Ltd., Londres, 1982.
- [36] Harris, C.J. and Billings, S.A., *Self-Tuning and Adaptive Control: Theory and Applications*, 2da ed., Peter Peregrinus Ltd, IEE, Londres, 1985.
- [37] Guarderas, Humberto, *Controlador Industrial Programable*, Tesis de Grado, Escuela Politécnica Nacional, Quito, 1990.
- [38] Ogata, Katsuhiko, *Ingeniería de Control Moderna*, Prentice Hall S.A., México, 1974.
- [39] Bastidas, Jorge, *Estudio teórico-experimental de un servomecanismo de velocidad y posición*, Tesis de Grado, Escuela Politécnica Nacional, Quito, 1981.
- [40] Johnson, B. and Aylor, J., Design of an Adaptive Controller for Microcomputer Implementation, *IEEE*

*Transactions on industrial electronics*, Vol IE-33, No. 1,  
pp. 28-38, Febrero 1986.

# APENDICE

## ABREVIATURAS Y SIMBOLOGIA

IA:	Inteligencia Artificial.
PLN:	Procesamiento de Lenguaje Natural.
CE:	Controlador Experto.
CE1:	Controlador Experto 1.
CE2:	Controlador Experto 2.
CBC:	Control Basado en Conocimiento.
CBR:	Control Basado en Reglas.
$\mu$ :	Función de pertenencia.
$n$ :	Orden del sistema.
$G(S)$ :	Función de transferencia en el plano S.
$G(Z)$ :	Función de transferencia en el plano Z.
LIMITE:	Límite del actuador.
SET_POINT:	Valor de la referencia.
fac:	Factor de variación de $a_{n-1}$ en $G(S)$ .
TMUESTREO:	Período de muestreo.
$h$ :	Paso de integración.
$T_r$ :	Tiempo muerto o retardo de transporte.
$d$ :	Representación del tiempo muerto en la simulación.
$K_p$ :	Ganancia proporcional para el control PID.
$T_i$ :	Tiempo integral para el control PID.
$\epsilon$ :	<del>Coeficiente</del>
$\omega_n$ :	Frecuencia natural.



Ta: Tiempo derivativo para el control PID.

e y E: Señal de error.

Y: Respuesta del sistema.

U\_PID: Señal del control PID.

U\_DEAD: Señal del control DEADBEAT.

U\_SELF: Señal del control SELF TUNING.

Gc(S): Función de transferencia del controlador.

k: Número de iteración.

$\theta$ : Arreglo con los parámetros estimados del sistema.

$\delta$ : Factor de olvido.

H: Precompensador.

T(Z<sup>-1</sup>): Polinomio característico cuyas raíces definen los polos asignados.

t<sub>1</sub> y t<sub>2</sub>: Coeficientes del polinomio T(Z<sup>-1</sup>).

Mp%: Porcentaje de máximo sobreimpulso.

ts: Tiempo de establecimiento.

$\tau$ : Constante de tiempo de un sistema de primer orden.

$\epsilon$ : Coeficiente de amortiguamiento.

Wn: Frecuencia natural.