



UNIVERSIDAD NACIONAL
AVENIDA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
UNIDAD ACADÉMICA DE LOS CICLOS
PROFESIONALES Y POSGRADO DEL
COLEGIO DE CIENCIAS Y HUMANIDADES
INSTITUTO DE INVESTIGACIONES EN
MATEMÁTICAS APLICADAS

MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

PROGRAMA SUPERVISOR PARA UNA RED DE
CONTROL DISTRIBUIDO

T E S I S

Que para obtener el grado de:

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

p r e s e n t a

Sergio Fernando López Caloca

MÉXICO, D. F.

TESIS CON
FALLA DE ORIGEN

1989



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

TESIS PROGRAMA SUPERVISOR PARA UNA RED DE CONTROL DISTRIBUIDO.

RESUMEN

La aplicación de esquemas y estrategias de control en la industria, han permitido supervisar las plantas industriales teniendo como consecuencia inmediata un mejoramiento en la producción. Sin embargo, los esquemas actuales de control no pueden implantarse sin el uso de una computadora digital. Esto ha originado el desarrollo de sistemas supervisores capaces de tomar decisiones para regular el funcionamiento de las plantas.

En el presente trabajo se muestra el diseño de un supervisor para un esquema de control distribuido, que contempla los siguientes aspectos: la aplicación de técnicas de programación de sistemas al control de procesos industriales, la creación de una herramienta de trabajo que permita aplicar las nuevas técnicas desarrolladas por los investigadores en el área de control y la aplicación del sistema a nivel industrial.

Dentro de los logros obtenidos se pueden mencionar los siguientes: La definición de una arquitectura de sistema de control distribuido, que permite comprender su estructura y funcionamiento y la definición de la programación de un supervisor con base en la arquitectura del sistema de control distribuido junto con las características básicas para realizar la observación de una planta industrial. Dicho supervisor tiene la capacidad de ser aplicado en diferentes tipos de plantas industriales al definirle un proceso en particular, mediante un mecanismo de configuración.

Los capítulos más importantes de la tesis son el segundo y cuarto. En el segundo se define la arquitectura del supervisor, para el esquema de control distribuido con el conjunto de necesidades básicas que impone la operación de control y en el cuarto la correspondiente a la programación, con base en la arquitectura definida para el supervisor.

He llegado a esta etapa de mi vida lleno de entusiasmo y satisfacción. Este trabajo representa un paso más en la realización de las metas que me he fijado. Metas que he ido logrando gracias al apoyo y solidaridad de las personas que han recorrido conmigo generosamente el camino.

A ellas quiero expresar mi agradecimiento

A la Dra. Cristina Verde Rodarte y al MI. Luis Alvarez Icaza por su apoyo profesional

A la Dra. Hanna Oktaba y al Dr. Victor Germán Sánchez por sus ideas acertadas

A la Sra. Teresa Radilla Ludwig por su colaboración en la corrección y cotejo de la tesis.

CAPÍTULO

1. INTRODUCCIÓN	1
2. CARACTERÍSTICAS GENERALES DEL SISTEMA	5
2.1 Definiciones generales	5
2.2 Supervisor	9
2.3 Tipos de controladores	10
2.3.1 Controladores físicos	13
2.3.2 Controladores virtuales	14
2.4 Comunicación entre controladores	17
3. MEDIOS PARA LA IMPLANTACIÓN	20
4. DIVISIÓN DE TAREAS E IMPLANTACIÓN DEL SUPERVISOR	23
4.1 Descripción funcional del sistema	23
4.2 Enlazador de actividades	25
4.3 Base de datos	26
4.3.1 Almacenamiento y consulta de las variables de entrada o salida de un controlador	29
4.3.2 Almacenamiento y consulta de las características de un controlador	30
4.3.3 Consulta de la ley de control de un controlador virtual	30
4.3.4 Generación de las llaves en los árboles B	31
4.4 Configuración	33
4.5 Lista de descriptores	35
4.5.1 Descriptores de los controladores físicos	35
4.5.2 Descriptores de controladores virtuales	38
4.5.3 Lista de alarmas	39
4.6 Coordinación en tiempo real	40
4.7 Comunicación entre el supervisor y los controladores físicos	44
CONCLUSIONES	46
REFERENCIAS	48

Apéndice 1	49
Apéndice 2	52
Apéndice 3	55
Apéndice 4	60
Apéndice 5	68

INTRODUCCIÓN

En la década de los cincuenta el control de los procesos industriales se llevaba a cabo por medio de dispositivos analógicos de tipo neumático, hidráulico o electromecánico. Sin embargo, debido al desarrollo de la tecnología de circuitos integrados a gran escala, el panorama descrito se modificó considerablemente. Actualmente, la regulación y/o supervisión de plantas industriales se realiza mediante sistemas digitales con base en microprocesadores. Esto ha permitido implantar y modificar estrategias de supervisión y control, que permiten mejorar el desempeño global de estos sistemas.

Esta situación ha generado diferentes estructuras de control, entre las que destacan: el esquema de control digital directo y el de control distribuido. En el primer esquema (fig 1) se centralizan en una computadora todas las funciones de control y supervisión de un proceso (planta). Sin embargo, esto resulta desventajoso ya que una falla en la computadora, deja al proceso sin control. Además, si se desean hacer cambios y/o expansiones del sistema se requiere que se suspenda el control del proceso. En cambio, la filosofía de un control distribuido (fig 2), emplea un conjunto de unidades de control llamadas controladores, que pueden contener uno o más microprocesadores y están supeditados a un ordenador de funciones comúnmente llamado supervisor. Cada dispositivo o unidad, tiene como función controlar localmente varios lazos y su comunicación se realiza mediante una red a la computadora central que lo supervisa. Así, si una unidad de control falla, el control del proceso se pierde parcialmente y cuando se realizan cambios y expansiones no necesariamente se suspende el control del proceso.

A la fecha, el esquema de control distribuido tiene mayores aplicaciones en los procesos industriales y existe un gran número de sistemas comerciales que permiten realizar la supervisión y control de un proceso llevado a cabo en pequeñas y grandes industrias [ref 9]. Estos sistemas comerciales son poco flexibles para relizar investigación ya que que no permiten la posibilidad de aplicar las nuevas técnicas propuestas por los investigadores en el área de control [ref 6].

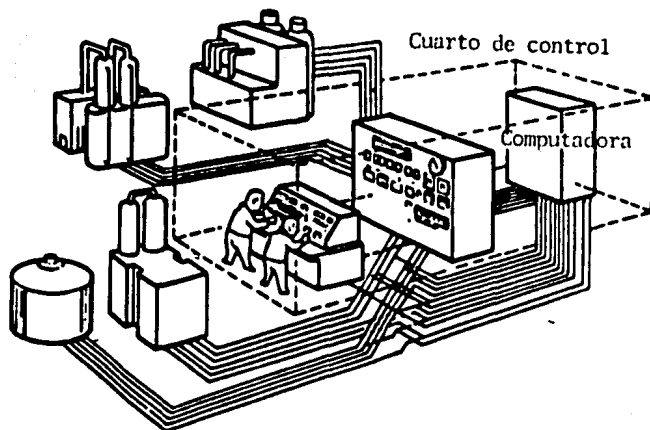


Fig 1. Esquema de control digital directo

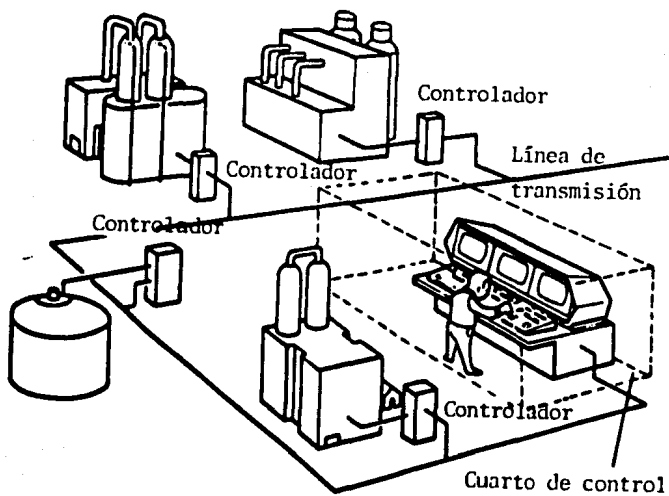


Fig 2. Esquema de control distribuido

En la sección de Automatización, del Instituto de Ingeniería de la UNAM, se trabaja en el diseño e implantación de una red supervisora para el control de plantas industriales bajo la filosofía del control distribuido, con fines académicos y con la posibilidad de aplicarla a nivel industrial. Para su realización, el esquema del control distribuido, se dividió en tres partes: el desarrollo de la electrónica para los controladores, el diseño de la red de comunicación y los programas para la computadora de supervisión.

Las compañías dedicadas al desarrollo de sistemas aplicados al control sólo proporcionan información del funcionamiento de su sistema sin reportar los aspectos relacionados con la programación [ref 6]; por otro lado, en la literatura de control por computadora, se le da mayor importancia a los algoritmos de control, dejando marginados los problemas de programación. Se presentan normalmente soluciones para algunas aplicaciones de control, pero sin reportar un análisis profundo de la programación desarrollada [refs 3, 12 y 15].

El propósito del presente trabajo es el diseño de un supervisor para un esquema de control distribuido con los siguientes objetivos:

- Estudiar y aplicar técnicas de programación de sistemas al control de procesos industriales.
- Ofrecer una herramienta flexible que permita aplicar las nuevas técnicas desarrolladas por los investigadores en el área de control.
- Proponer una arquitectura que ofrezca perspectivas de aplicación práctica en diferentes tipos de procesos industriales, tomando en cuenta que se le desea implantar en una computadora personal (PC).
- Integrar el supervisor al esquema de control distribuido que se desarrolla en el Instituto de Ingeniería.

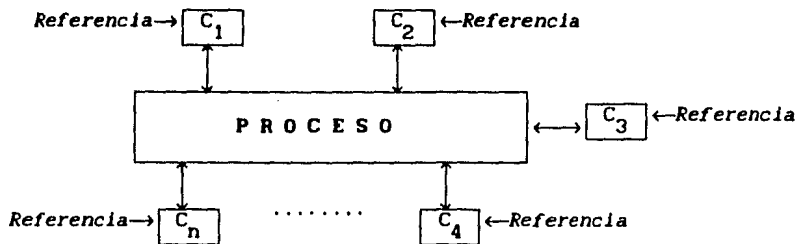
En el segundo capítulo se especifican las características de la red de controladores que pueden ser operados por el supervisor y la comunicación que se puede establecer entre ellos; en el tercero, se examina y seleccionan las herramientas utilizadas para la elaboración del programa supervisor; en el último, se describen las actividades del supervisor y

la forma en que fueron divididas, así como el diseño e implantación de la base de datos utilizada por los programas que integran el programa supervisor.

2. CARACTERÍSTICAS GENERALES DEL SISTEMA

2.1 DEFINICIONES GENERALES.

Se define como sistema de control distribuido aquel donde las labores de control se reparten entre un conjunto de dispositivos independientes, llamados controladores y que realizan la función de gobernar de manera coordinada el comportamiento de un proceso físico en tiempo real. El esquema de un sistema como el que se describió se presenta en la siguiente figura:



Donde C_1, C_2, \dots, C_n , son los controladores y la *Referencia* es el valor deseado para la variable de control

Fig 3. Esquema de control distribuido.

Se entiende por proceso al sistema físico que se desea gobernar de manera automática.

Se define como controlador la unidad elemental (más primitiva o esclava) de un sistema distribuido. Físicamente, corresponde a un dispositivo que acopla en forma retroalimentada una parte del proceso a controlar. Las tareas que puede ejecutar por lo general son muy particulares, limitándose a calcular acciones que dependen de su ley de control, de las órdenes o entradas de referencia y de las variables del proceso.

Lo deseable sería el caso que todos los controladores pudiesen realizar su trabajo adecuadamente, aun cuando existan perturbaciones eléctricas, electrónicas, mecánicas, etc, en el proceso. Como en la realidad este

caso ideal no se cumple, es común hoy en día agregar a los sistemas de control un elemento más denominado supervisor, cuyas funciones son:

- a) Permitir configurar esquemas distribuidos de control.
- b) Adquirir y recopilar información del estado del proceso a través de los controladores para mostrársela al operador y realizar funciones de muestreo y estadística.
- c) Establecer enlaces entre el proceso y el operador para que éste tome decisiones sobre el proceso.

El elemento supervisor es generalmente un programa que se ejecuta en una computadora y al ser agregado al esquema de control distribuido toma la siguiente forma:

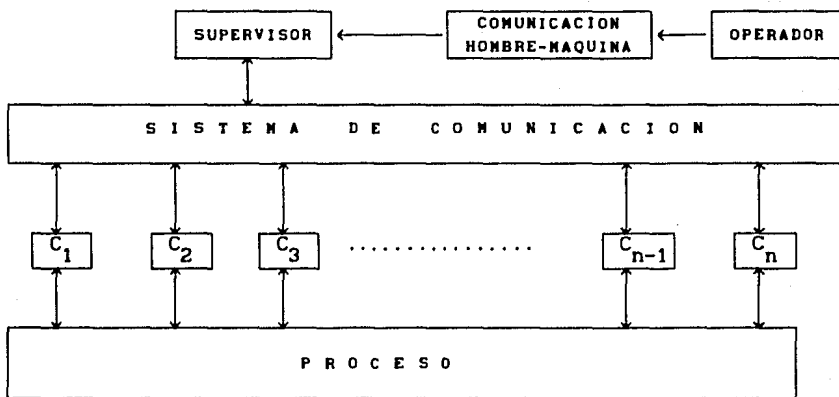


Fig 4. Supervisor de control distribuido

Para que el supervisor pueda realizar sus funciones, es necesario que existan medios para la transmisión de información entre los controladores locales, el operador y el supervisor. Esto se logra mediante dos sistemas de comunicaciones: uno que maneje la información con el operador y el otro con los controladores.

Debido a que desde el punto de vista de la programación del supervisor, el sistema de comunicación para los controladores locales es una unidad

independiente del supervisor, en el presente trabajo no se profundiza sobre dicho sistema. En el Apéndice 4 se describen los protocolos usados en el Instituto de Ingeniería.

Con las propiedades asignadas al controlador local, éste puede esquematizarse de la siguiente forma:

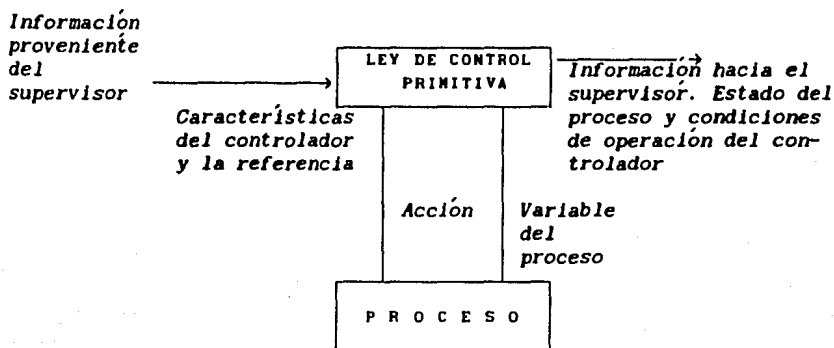


Fig 5. Flujo de información de un controlador

En los esquemas presentados en las figuras 4 y 5, se puede observar que el supervisor conoce la información de los controladores locales. Con la información de un conjunto de controladores locales se puede realizar un control más global si se utiliza una nueva ley de control que tome en cuenta dicha información. Esta nueva ley de control permite definir un nuevo nivel de leyes de acción, las cuales se denominarán controladores virtuales. Los controladores virtuales deberán configurarse dentro del sistema supervisor. Lo anterior permite incrementar las posibilidades para implantar controles más generales y eficientes.

En particular, se consideró adecuado permitir que los controladores virtuales ejecuten las siguientes funciones

- a) Tomar información de los controladores locales (como dispositivos físicos).
- b) Evaluar acciones de acuerdo con leyes de control general.
- c) Transmitir órdenes a los controladores locales para que éstos

realicen directamente las acciones sobre el proceso.

El concepto del controlador virtual requiere modificar nuevamente el esquema de control distribuido, aun cuando estos no existan como dispositivos físicos, ya que únicamente sus leyes de acción se realizan por programas. Una forma de visualizar el nuevo esquema de control distribuido se presenta en la fig 6.

En dicha figura se observa que los controladores virtuales reciben la información de los controladores físicos y la envían al supervisor y a los controladores físicos. El hecho de transmitir la información al supervisor se debe a que este tipo de controladores están conceptualizados para tomar decisiones y reportar condiciones especiales al operador en función del comportamiento del proceso a controlar.

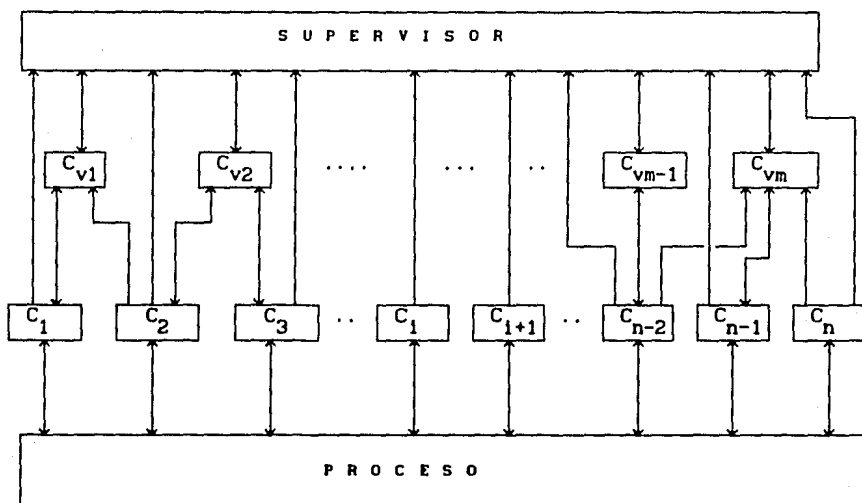


Fig 6. Sistema distribuido con jerarquía

Nótese que en este último esquema no se presentan los sistemas de comunicación entre los controladores locales (c_i) y los virtuales (c_{vi}) y entre el operador y el supervisor, con objeto de facilitar la concepción del esquema. Estos se explicarán en el capítulo 4.

A lo largo del presente trabajo, se identificarán como controladores físicos a los dispositivos locales y como virtuales a los implantados por programa.

Con el fin de dar una mejor descripción tanto de las tareas asignadas, como de las normas de operación del esquema supervisor (fig 6), se presentan por separado en los siguientes incisos.

2.2 SUPERVISOR

Para realizar un supervisor que permita gobernar cualquier proceso, es necesario dotarlo de un mecanismo de configuración mediante el cual un operador pueda definir un proceso en particular. Esta nueva tarea del supervisor debe permitir al operador configurar redes de controladores (físicos y virtuales) con las características asociadas a cada uno de ellos.

Para no perder la configuración del sistema como en el caso de fallas en el suministro de energía eléctrica, se considera necesario respaldarla en una base de datos.

Una vez configurada la red y puesta en operación, el supervisor tiene la tarea de obtener en tiempo real la información del estado del proceso mediante el sistema de comunicación.

Tal información se debe almacenar en una base de datos para diferentes usos, como cálculos de acciones de los controladores virtuales, despliegues, cálculos estadísticos, impresión de informes, etc. Por otro lado, cuando el supervisor está trabajando, pueden presentarse fallas externas a éste que impidan su buen funcionamiento.

Este hecho obliga, por tanto, a crear dentro del supervisor detectores de fallas que envíen mensajes al operador. Las anomalías consideradas más frecuentes y que deberán ser reportadas al operador son:

- a) El controlador físico no es capaz de compensar perturbaciones en el proceso y en consecuencia éste se encuentra en condiciones de alarma o peligro.
- b) El controlador físico no responde cuando se le solicita la información.
- c) No se recibe información de ningún controlador cuando se le solicita.

Resumiendo, el supervisor tiene la obligación de realizar las siguientes tareas:

- 1) Configurar el sistema (controladores físicos y virtuales).
- 2) Obtener la información de los controladores físicos que actúan sobre el proceso mediante el sistema de comunicación.
- 3) Almacenar la información de los controladores físicos para su uso posterior.
- 4) Detectar y reportar los estados de alarma de los controladores físicos.
- 5) Detectar y reportar las fallas de comunicación entre el sistema de comunicación y los controladores físicos.
- 6) Vigilar la buena operación de los controladores virtuales.
- 7) Desplegar la información recabada en la red, o bien, la solicitada por el operador.
- 8) Presentar reportes impresos de la situación del sistema en tiempos predefinidos, o bien, a solicitud del operador.

2.3 TIPOS DE CONTROLADORES

En forma aislada un controlador virtual puede verse como una unidad con n entradas (información proveniente de los controladores físicos); m salidas (información que se puede transferir a los controladores físicos) y una ley de control diseñada con base en la teoría del control, la cual puede verse como una transformación de espacio de entradas de dimensión de \mathbb{R}^n al espacio \mathbb{R}^m de las salidas.

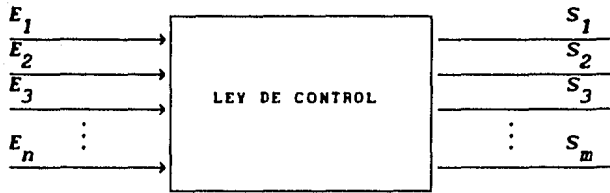


Fig 7

El concepto de controlador virtual -aquí presentado- se puede emplear para crear esquemas de control con un mayor número de niveles de Jerarquía; donde los físicos c_1 dependen de los virtuales del primer nivel c_{v1} , los cuales a su vez podrían recibir órdenes de otros virtuales del segundo nivel y así sucesivamente. Por razones prácticas, el presente trabajo considerará únicamente controladores virtuales del primer nivel.

En el Instituto de Ingeniería se construye actualmente un controlador, el cual sólo puede recibir información de una variable física y enviar una acción al proceso. Empero, esto no significa que no puedan construirse controladores que tengan más de una variable y más de una acción a controlar.

Este tipo de controlador industrial se identifica como controlador monovariante ya que sólo puede regular una variable. En caso de tener más de una acción o variable a controlar, se habla de controladores multivariantes. Conceptualmente, un conjunto de controladores monovariantes puede verse como un solo multivariante donde cada una de las acciones depende únicamente de una variable del proceso (sistema desacoplado).

Tomando en cuenta el hecho de que un controlador físico es monovariante, se adopta la convención de que su entrada de referencia sólo puede provenir de un controlador virtual, mas no así su salida (información de las condiciones del proceso), que puede ser usada por varios controladores virtuales para la toma de decisiones.

A pesar de las restricciones impuestas al esquema de control, éste conserva su capacidad de supervisión sobre el sistema distribuido, ya que se mantiene la comunicación directa entre los controladores físicos virtuales y el operador.

Las funciones especificadas hasta ahora para los controladores son muy generales, puesto que no se ha determinado el tipo de acciones que generan, ni la clase de variables que pueden manejar. Esto quiere decir, que falta por definir el dominio, la imagen y tipo de mapeo de un controlador. El dominio D está dado por el conjunto de órdenes o referencias y variables del proceso; la imagen I por el conjunto de acciones posibles sobre el proceso y el mapeo Γ al conjunto de leyes que pueden ser implantadas. La fig 8 muestra la representación usada en la literatura de teoría de control para un controlador, donde t , representa la variable independiente que es el tiempo [ref 2].



Fig 8. Variables asociadas a un controlador

De la misma manera que los controladores, los procesos físicos pueden ser representados matemáticamente por mapeos Γ' , con dominio en el conjunto de acciones posibles $u(t)$ e imagen en el conjunto de variables de salida del proceso $y(t)$.

La fig 9 muestra su representación convencional.



Fig 9. Variables asociadas a un proceso

Como la salida o salidas del proceso $y(t)$ forman parte del dominio del controlador, el acoplamiento de variables puede verse como un mapeo Γ''

con dominio en el conjunto de referencias $r(t)$ posibles y rango en el conjunto de salidas del proceso $y(t)$ posibles. Este tipo de mapeo corresponde al de un sistema retroalimentado. Una representación muy general de él se muestra a continuación.

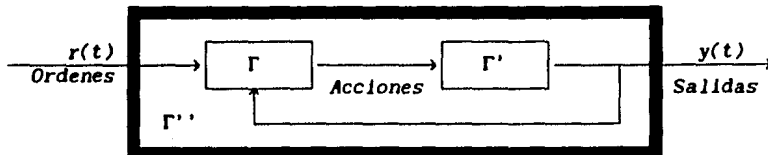


Fig 10. Sistema de control retroalimentado

2.3.1 Controladores físicos

En el controlador físico diseñado en el Instituto de Ingeniería se ha limitado el mapeo Γ a funciones lineales invariantes en el tiempo y con dominio e imagen en \mathbb{R} . En particular, se considera la ley de acción:

$$u(t) = K_p e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{d e(t)}{dt} \quad (I)$$

donde $e(t) = r(t) - y(t)$ se denomina la señal de error; $r(t) \in \mathbb{R}$ corresponde a la señal de referencia u órdenes; $y(t) \in \mathbb{R}$ es la variable de salida del proceso y t la variable independiente. Los parámetros constantes K_p , T_i y T_d se ajustan en función del comportamiento deseado del proceso [ref 8].

En la industria este tipo de controlador físico se denomina PID. Esto se debe a que las acciones que ejecutan son: proporcional (primer término de la ecuación), integral (segundo término) y derivativo (tercer término). En la práctica el 90 por ciento o más de los controladores industriales opera bajo este tipo de ley.

Una de las ventajas de un supervisor es la posibilidad de considerar en su esquema la detección de valores máximos y mínimos en las variables del

proceso. Esto implica, para el caso particular de los controladores, la definición de tolerancias r^+ y r^- para la variable de referencia $r(t)$ (entrada). De este modo, el supervisor tiene la posibilidad de producir alarmas por saturación de la señal de error $e(t) = r(t) - y(t)$.

Con el objeto de manejar dentro del controlador físico las variables en un mismo intervalo de valores (normalizadas), el dispositivo sólo puede enviar y recibir variables normalizadas en el intervalo de operación entre 0 y 1, donde 0 es el valor mínimo de operación y 1 el máximo. La fig 11 muestra los intervalos esquemáticamente.

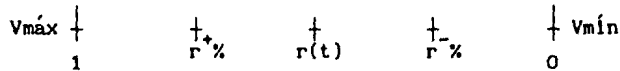


Fig 11. Intervalo de tolerancia y operación del controlador físico

Por tanto, el conjunto formado por los parámetros de la ecuación (I), el intervalo de operación V_{\max} y V_{\min} , el intervalo de tolerancia r^+ y r^- y las variables $y(t)$ y $e(t)$ caracterizan al controlador físico.

A manera de ejemplo se presenta el siguiente caso: Supóngase que el intervalo de la temperatura varía entre 10°C y 20°C y en un instante en particular se desea mantener la temperatura en 15°C con una tolerancia de $\pm 0.5\%$; esto implica que $V_{\max} = 20^{\circ}\text{C}$, $V_{\min} = 10^{\circ}\text{C}$, $r(t) = 15^{\circ}\text{C}$, $r^+ = 15.075^{\circ}\text{C}$ y $r^- = 14.925^{\circ}\text{C}$.

2.3.2 Controladores virtuales

Con el objeto de implantar leyes de acción más complejas y con diferentes tipos de variables, se proponen tres clases de funciones o transformaciones que pueden ser manejadas por los controladores virtuales. Dichas funciones deberán ser evaluadas en instantes de tiempo fijos T_i , los cuales se especifican junto con las acciones.

Funciones recursivas, se definen como el conjunto de sistemas lineales con espacio de dimensión finita y descritos por una ecuación de

diferencia del tipo

$$\begin{aligned} X(kT+T) &= A \cdot X(kT) + B \cdot U(kT) \\ Y(kT+T) &= C \cdot X(kT) + D \cdot U(kT) \end{aligned} \quad (1)$$

Donde X es el estado, U la entrada y Y la salida del sistema lineal; con $X \in \mathbb{R}^n$, $U \in \mathbb{R}^m$, $Y \in \mathbb{R}^p$, A , B , C , D matrices de dimensiones apropiadas, T el periodo de muestreo o intervalo de evaluación de la función Y elemento de los enteros. La fig 12 muestra una representación de la ec 1. El vector U normalmente contiene información de las variables de los controladores físicos.

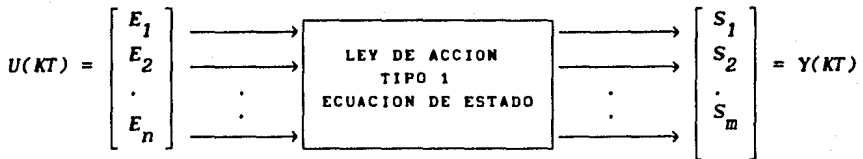


Fig 12. Modelo de estado multivariable

Las funciones estáticas, se definen como el conjunto de funciones no lineales sin memoria, del tipo

$$y = f(e_1, e_2) \quad (2)$$

con e_1, e_2, \dots, e_n escalares que corresponden con la información proveniente de los controladores físicos y los valores constantes. El Apéndice 1 muestra las reglas que rigen estas funciones.

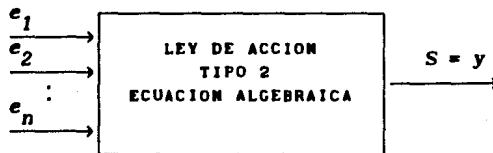


Fig 13

Las funciones lógicas, se definen como el conjunto de funciones lógicas generadas a partir de las operaciones básicas de conjunción, disyunción y negación (-), sin operadores relacionales. Del siguiente tipo:

$$y = f_1 (e_1, e_2, \dots, e_n) \quad (3)$$

con e_1, e_2, \dots, e_n , que representan las variables que corresponden con la información proveniente de los controladores físicos. El Apéndice 2 muestra las reglas que rigen estas funciones

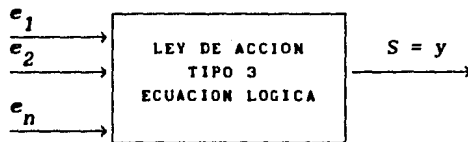


Fig 14

A manera de ejemplo se presenta el siguiente caso: Supóngase que se tiene una medición de temperatura que debe mantenerse en un intervalo entre 10°C y 20°C y una presión que debe regularse entre 2 y 5 gr/cm^2 ; en caso de que alguna de las dos variables estén fuera de los límites especificados, la variable de salida y deberá tener un valor positivo. La ley de acción con la que se puede lograr esta acción es:

$$y = \text{temperatura} * \text{presión}$$

donde el símbolo $*$ es la conjunción.

Para definir el estado lógico de una variable se aplica la siguiente regla. Si la variable está dentro del intervalo cerrado, toma el valor lógico 0; en caso contrario 1; por lo que, si la temperatura tiene el valor de 15°C y la presión 3.5 gr/cm^2 , la función y tomará el valor lógico 0.

2.4 COMUNICACIÓN ENTRE CONTROLADORES.

Al igual que los controladores físicos, los virtuales también deben ser definidos mediante el programa de configuración para que el supervisor pueda reconocerlos.

Para realizar la comunicación entre los controladores físicos y virtuales, es necesario considerar la restricción impuesta en el inciso 2.3, donde se dice que un controlador virtual tiene capacidad para acceder información de varios físicos, pero un físico sólo puede recibir información de una de las salidas de algún controlador virtual. Esta restricción no evita la presencia de casos en que la salida de un físico pueda ser usada en varios virtuales (ver fig 6), o que las salidas de un controlador virtual no estén asignadas a uno físico.

A continuación se presentan algunos ejemplos de acoplamiento entre los dos tipos de controladores. Estas gráficas no corresponden a ningún proceso físico en particular y servirán para indicar la relación entre las entradas y salidas.

Ejemplo 1.

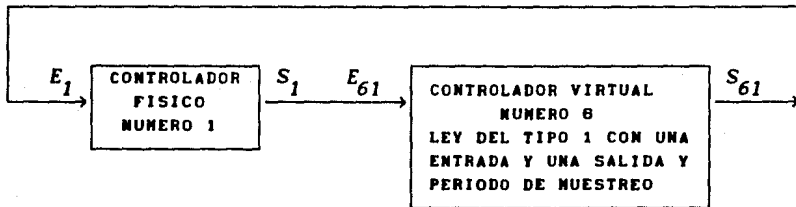


Fig 15. Acoplamiento entre un controlador virtual y uno físico con función recursiva

Ejemplo 2.

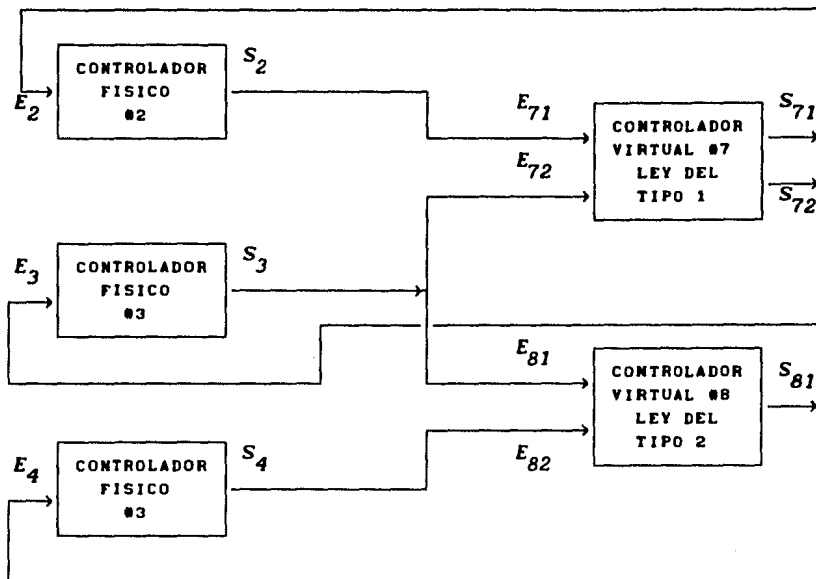


Fig 16. Acoplamiento entre controladores físicos y virtuales con función recursiva y no lineal

Ejemplo 3.



Fig 17. Acoplamiento entre un controlador virtual y uno físico con función lógica

Debido a que las salidas de los controladores virtuales tipo 3 son variables lógicas, éstas no pueden ser empleadas como entradas de controladores físicos del tipo PID que se utilizan en la red. Esta es una limitación del sistema actual, pero se está diseñando en el Instituto de Ingeniería, un controlador físico capaz de manejar variables lógicas.

Es importante señalar que los controladores con leyes del tipo 1

(funciones recursivas), tienen explícita la frecuencia de evaluación ($1/T_1$); los controladores del tipo 2 (no lineales) y del 3 (lógicos), por el contrario, carecen de un tiempo explícito; en este caso es necesario fijar la frecuencia de evaluación al definir la función en particular.

Analizando las convenciones establecidas para el acoplamiento entre controladores (físicos y virtuales), se puede ver que para establecer la comunicación entre ellos debe primeramente adquirirse la información proveniente de los controladores físicos, mediante el sistema de comunicación, y después evaluar las funciones de las leyes virtuales. En el otro caso, cuando la información de los virtuales deba ser enviada a los físicos, ésta se lleva a cabo después de evaluar las salidas de los virtuales.

Hasta el momento no se han tomado en cuenta las fallas en el envío de mensajes entre controladores físicos y virtuales, que pueden presentarse debido a retrasos en las llegadas de la información. Ésto implica que además de las tareas asignadas al supervisor (inciso 2.2), se le debe adicionar la tarea de generar estados de alarma indicándoselos al operador cuando la información no llegue en los tiempos previstos. Es claro que este problema rompe con el orden de evaluación propuesto (adquisición de los datos del proceso, evaluación de las leyes de control y respuesta de los controladores virtuales), para darle prioridad a las alarmas.

3. MEDIOS PARA LA IMPLANTACIÓN

En el capítulo anterior se describieron las especificaciones que debe cubrir el sistema supervisor para la red de control distribuido. A continuación se mencionan los argumentos que deben ser considerados para obtener una solución práctica. Además, se presentan las herramientas y metodología adoptadas en el caso particular del presente trabajo.

V V Lipaev [ref 10], reporta una amplia gama de aspectos que influyen en el diseño de paquetes genéricos para supervisar procesos industriales en tiempo real. Los más importantes de son:

a) Contar con personal especializado en las diferentes áreas que cubren el diseño y la programación (ingenieros de software, comunicaciones, procesos industriales, bases de datos, graficación, etc).

b) Seleccionar adecuadamente las herramientas de trabajo, como son computadora, sistema operativo, compilador, paquetes de comunicaciones, paquetes gráficos, manejadores de base de datos, etc.

Una característica de estos programas es el tamaño. Se estima que contienen del orden de cien mil instrucciones y su desarrollo puede tomar entre dos y cinco años.

Por otro lado, independientemente del tipo de sistema a diseñar, se sabe que la elección adecuada tanto del software como del hardware, requiere consideraciones globales de ambas partes. Puede presentarse el caso de que la computadora seleccionada sea correcta pero una inadecuada selección del lenguaje o sistema operativo, produzca desempeños mediocres.

En el caso del software de reciente creación, la falta de experiencia e información y quizá la novedad, hacen difícil garantizar la ausencia de problemas graves en la etapa de desarrollo, como pueden ser pérdidas excesivas de tiempo y complejidad no previstas en la programación.

Cuando no se tiene posibilidad de elegir el tipo de computadora donde se desea instalar el supervisor, como es el caso del que se trata, el problema de la selección de las herramientas restantes es aún más crítico. El hecho de restringir la computadora a una IBM PC o compatible limitó la selección del soporte de programación.

A continuación se describen los argumentos que se tomaron en cuenta para la selección del sistema operativo y de lenguaje.

Existen una gran cantidad de lenguajes para las PC, pero los que pueden realizar tareas en forma concurrente para resolver los problemas del supervisor, son pocos; de entre ellos destacan Modula-2 y ADA.

Con Modula-2 [ref 11], se realizaron pruebas, creando un despachador en tiempo real que conviviera con las corrutinas mediante la interrupción del reloj de la computadora y aprovechando las facilidades del lenguaje para hacer programación concurrente.

Los resultados mostraron que no es posible quitarle el control a una corrutina por tiempo, porque se crean desórdenes en la concurrencia que ocasionan la pérdida del control en el sistema operativo. Además, como se sabe el lenguaje cuenta con instrucciones primitivas que podrían ser usadas para hacer un despachador de procesos [refs 7 y 17]; sin embargo, esto implica que la programación debe iniciarse a un nivel muy básico, por lo que se estimó conveniente no utilizarlo.

El lenguaje ADA, ofrece muchas ventajas para el supervisor ya que su programación concurrente la realiza mediante el envío de mensajes. Pero como está en etapa de desarrollo para las PC, algunas de las versiones accesibles que existen en el país, presentan limitaciones prácticas. En particular, el compilador disponible era incompleto [ref 1]; requería de siete pasadas y una computadora IBM-AT con 2Mb de memoria como mínimo. Estas limitaciones hicieron su empleo poco atractivo.

Otra forma de solucionar el problema, aunque parcialmente, es utilizar como sistema operativo a UNIX, donde las alarmas podrían ser consideradas como procesos de alta prioridad con control absoluto del CPU.

Dentro de UNIX, el sistema operativo puede dar alta prioridad a ciertos procesos con periodos grandes de tiempo para uso de CPU; sin embargo, hay que tomar en cuenta que cuando se termina el tiempo asignado al proceso, éste es suspendido y se inicia la ejecución de otro. Esto significa que puede darse el caso de que una alarma no sea atendida completamente [ref 5].

Los resultados de las pruebas con el sistema operativo XENIX fueron desalentadores. Se registraron tiempos de respuesta del orden de segundos, implantado en una computadora PC compatible XT, por lo que se decidió no usarlo, ya que es necesario contar con versiones más optimizadas del sistema operativo y computadoras más rápidas.

Con base en las observaciones arriba presentadas, se adoptó MSDOS como sistema operativo.

Para la selección del lenguaje, dentro de los no concurrentes, se estimó al C como el más adecuado. A continuación se presentan las razones por las cuales se tomó esta decisión

- 1) La programación desarrollada puede trasladarse fácilmente si se sustituye MSDOS por UNIX.
- 2) Los paquetes comerciales como manejadores de bases de datos y graficadores cuentan con interfaces para que éstos puedan ser operados desde el lenguaje C.
- 3) Se tiene experiencia y se conoce la potencialidad del lenguaje como herramienta orientada a la programación de sistemas.

4. DIVISIÓN DE TAREAS E IMPLANTACIÓN DEL SUPERVISOR.

4.1 DESCRIPCIÓN FUNCIONAL DEL SISTEMA

Este capítulo describe la forma operativa propuesta para la programación del sistema, junto con su manejador de base de datos. Esto último se implantó seleccionando el modelo de base de datos tipo red y es la parte medular del mismo.

Con base en las funciones principales asignadas al supervisor y la incapacidad del lenguaje C para manejar concurrencia, se estimó conveniente dotar al sistema de un programa *ENLAZADOR DE FUNCIONES* con capacidad para vincular las actividades en una forma ordenada con MSDOS. Además, el *ENLAZADOR* estará residente en memoria y será el primer programa en ejecutarse en forma automática al instalarse MSDOS.

Por otro lado, a diferencia de las demás tareas asignadas al supervisor (inciso 2.3), la configuración de la red junto con la instalación del propio supervisor no requiere realización al mismo tiempo (en línea), por lo que se subdividió el *ENLAZADOR DE FUNCIONES (EF)* en dos unidades: la de *CONFIGURACION (C)* operando fuera de línea y la de *COORDINACION EN TIEMPO REAL (CTR)*.

La fig 18 muestra de manera esquemática las funciones del *EF*, así como las interacciones entre el *C* y el *CTR* con los dispositivos internos y externos del sistema distribuido. Dentro del esquema pueden identificarse los siguientes estados:

Estado inicial: El operador interactúa con el programa de configuración para definir la red del proceso y ésta se almacena en el *banco de datos*. Las tareas relacionadas con este estado trabajan bajo la responsabilidad de la unidad *configurador (C)*.

Estado uno: El *adquisidor* solicita información de los controladores físicos de la red (mediante el *sistema de comunicación*) para

almacenarla en el banco de datos. En el caso de condiciones de alarma, el *adquisidor* suspende temporalmente su tarea para suministrar un aviso al *desplegador*.

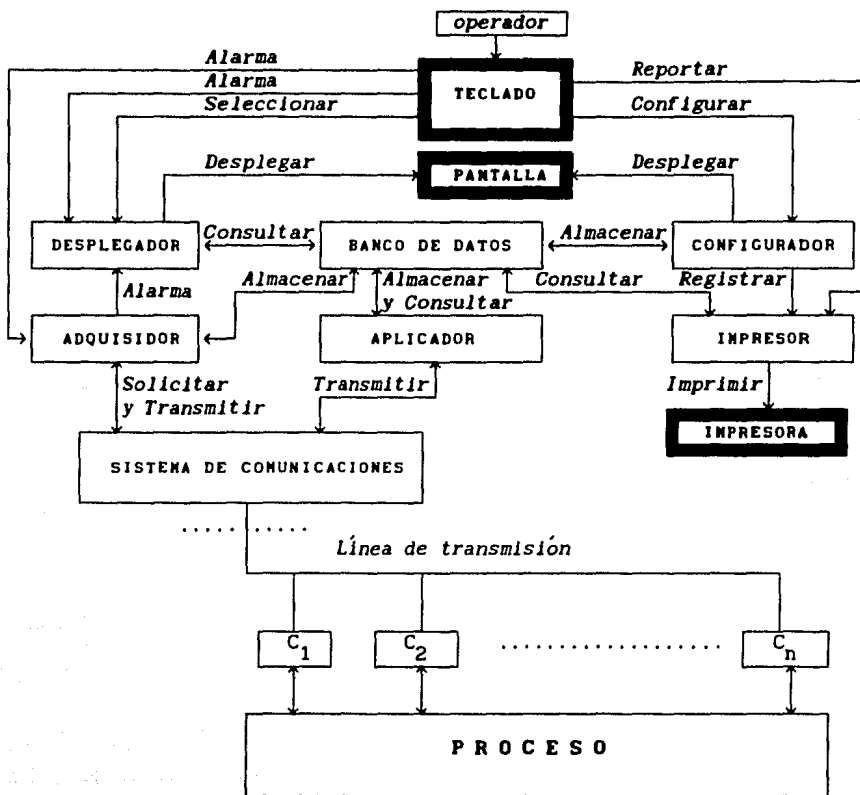


Fig 18. Diagrama operativo del supervisor

Estado dos: El *aplicador* de leyes de control toma de el *banco de datos* (las variables físicas y datos de la red), para evaluar las funciones de los controladores virtuales y los resultados los deposita en el *banco de datos*. En caso de tener acciones para los controladores físicos, éstas se transmiten automáticamente. Debido a que los tiempos para la evaluación de cada ley, por lo general, son diferentes, se debe vigilar que sus tiempos sean respetados.

Estado tres: En caso de alarmas el *desplegador* recibe directamente del *adquisidor* la información que debe aparecer en *pantalla*. Si existe una solicitud del operador para conocer el estado del proceso en la *pantalla*, el *desplegador* consulta el *banco de datos* y realiza el despliegue. La comunicación del operador con el sistema se logra mediante el *teclado* y la *pantalla*.

Estado cuatro: Si el intervalo de tiempo asignado para el reporte periódico del proceso se ha cumplido, se realiza la impresión. En caso de peticiones extemporáneas de reportes, el operador las solicita al *impresor*.

Estado cinco: Si existe una petición de cambio de configuración el *CTR* deja de operar y regresa al modo de configuración. En caso contrario, el ciclo de trabajo del *CTR* termina y regresa al estado uno.

4.2 ENLAZADOR DE ACTIVIDADES

De la descripción arriba mencionada, se observa que el *EF* puede modelarse bajo el siguiente algoritmo:

Paso 1: Revisa si está dada de alta la configuración del sistema; en caso afirmativo transfiere el control al paso 4, si no pasa al 2. En caso de una reinstalación de MSDOS el sistema se reinicia en este paso.

Paso 2: Prepara el medio ambiente para definir la configuración (creación de la base de datos si no existen, instalación del programa *CONFIGURADOR* en memoria, etc) y transfiere el control del procesador al *CONFIGURADOR*. Cuando el *CONFIGURADOR* termina su trabajo regresa el control al *EF*.

Paso 3: Desinstala el programa *CONFIGURADOR* junto con su medio ambiente.

Paso 4: Prepara el medio ambiente para el programa *CTR* y le transfiere el control del procesador . Cuando el *CTR* se interrumpe porque se desea reconfigurar la red, éste le regresa el control del procesador al *EF*.

Paso 5: Desinstala el programa *CTR* junto con su medio ambiente y regresa al paso 2.

4.3 BASES DE DATOS

De acuerdo con la fig 18, se puede observar que todos los elementos del sistema (desplegador, adquisidor, configurador y aplicador), requieren consultar la información del proceso continuamente que se encuentra en el banco de datos. Este hecho implica la necesidad de un mecanismo de acceso rápido a la información; para lograrlo se propone usar la memoria RAM como dispositivo de almacenamiento generando e instalando en ella una base de datos. Dicha base de datos debe contener la información de la configuración y la historia del proceso con su estado operativo.

Como en la memoria no se puede mantener toda la información que se recopila del proceso, éste se limitó a cierta cantidad de datos; esto implica respaldar en disco cada vez que el área prevista para el almacenamiento temporal (historia) se satura.

Los manejadores de bases de datos comerciales están diseñados para trabajar con el disco y no con la memoria. Una forma de crear una base de datos en memoria mediante los manejadores, es por medio de un disco virtual instalado en memoria. Esta solución tiene la desventaja de ser poco transportable, por lo que se propone resolver este problema diseñando una base de datos transportable con capacidad de instalación en disco y en memoria.

Tomando en cuenta el tipo de información, los canales de comunicación entre los componentes del supervisor y los modelos de base de datos más conocidos (jerárquico, redes y relacional [ref 13]), se puede concluir que el modelo de redes, es el que mejor se adapta a las características

del supervisor.

La fig 20 presenta el modelo de la red cuyos puntos de interés son:

- El registro de salida de los controladores físicos donde llegan los conjuntos Sal-CF (salida del controlador físico) y Ent-CV (entrada del controlador virtual)
- El registro de salida de los controladores virtuales donde llegan los conjuntos Ent-CF (entrada del controlador físico) y Sal-CV (salida del controlador virtual).

Estos conjuntos (vínculos), representan la comunicación entre controladores físicos y virtuales. La particularidad del modelo, es el registro con el sistema que sirve para representar un punto de entrada común a los dos tipos de controladores.

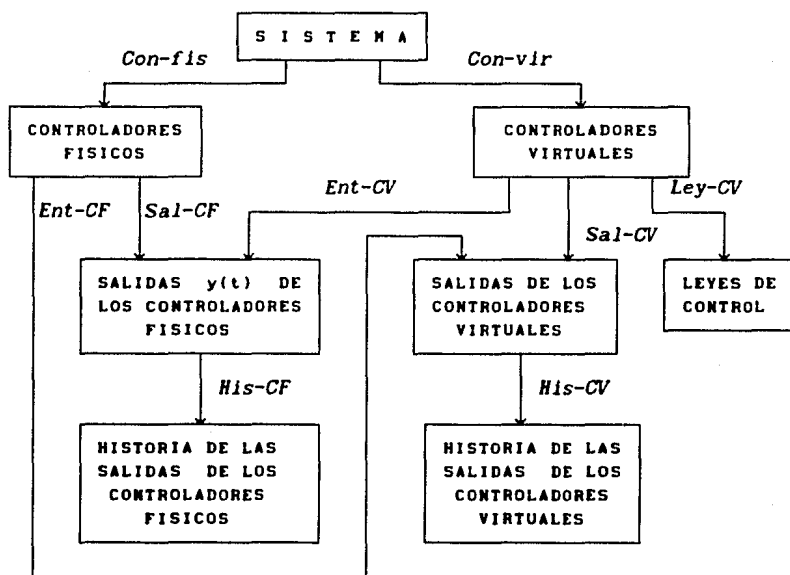


Fig 20. Modelo de la base de datos

En el modelo de redes propuesto no se consideró el registro de entrada de los controladores físicos debido a que en el caso en que la entrada no esté conectada a una salida de un controlador virtual, ésta se define en

la configuración y es una constante que sólo puede ser modificada por el operador; por el contrario, si la entrada está asociada a la salida de un controlador virtual se puede consultar mediante el conjunto *Ent_CF* (entrada del controlador físico).

Existen diferentes técnicas para implantar una base de datos [ref 14]. Entre éstas se encuentra la de los árboles *B* que permiten un buen manejo y organización de archivos; este hecho motivó seleccionar la estructura de árbol *B* [refs 4 y 16] para implantar el modelo de la base de datos.

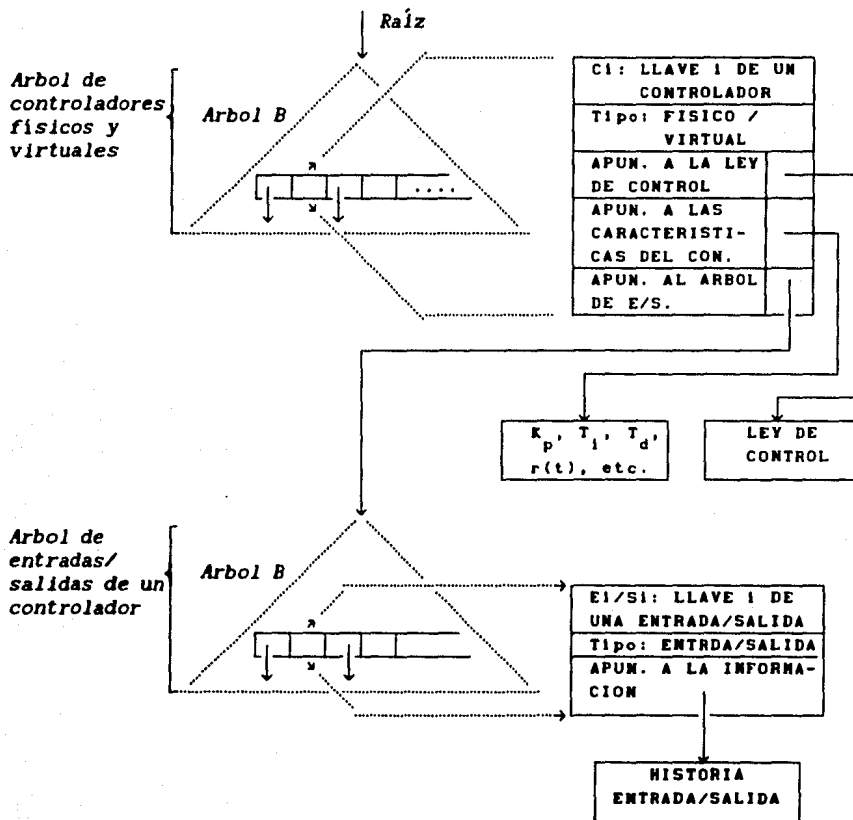


Fig 21. Arquitectura de la base de datos

La fig 21 muestra la arquitectura de la base de datos que se utilizó para

la implantación del modelo propuesto que será utilizada en el supervisor. En esta figura se tienen dos niveles de indexamiento; en el primero se tiene un árbol *B* con las llaves de todos los controladores físicos y virtuales y en el segundo, otro árbol *B* particular para cada controlador con las llaves de sus entradas y salidas.

De la fig 18 se deduce que la base de datos debe permitir realizar organizadamente las tareas de almacenar y consultar las variables de entrada y salida de un controlador, consultar sus características (incisos 2.3.1 y 2.3.2) y consultar las leyes de control para su aplicación.

Los siguientes algoritmos se proponen para lograr estas demandas.

4.3.1 Almacenamiento y consulta de las variables de entrada o salida de un controlador

Con base en el árbol de las variables (entradas y salidas) de un controlador, el algoritmo implantado para guardar o consultar éstas se reduce a:

Paso 1: Se consulta el árbol *B* del primer nivel que contiene las llaves de los controladores definidos en la configuración; si no existe el controlador, el algoritmo termina regresando un mensaje de error.

Paso 2: Se revisa que el controlador tenga definido un árbol *B* de entradas y salidas; si no existe éste, el algoritmo termina regresando un mensaje de error.

Paso 3: Se consulta el árbol *B* de las entradas y salidas para localizar la ubicación de la entrada/salida donde se desea almacenar o consultar el dato; si no existe la entrada/salida se termina el algoritmo regresando un mensaje de error.

Paso 4: Se deposita o consulta el dato en el área adecuada de

almacenamiento (entrada/salida) y se termina el algoritmo indicando éxito.

4.3.2 Almacenamiento y consulta de las características de un controlador

Cuando se desea examinar o modificar las características de un controlador (parámetros, tolerancia, referencia, tiempo de evaluación, etc) el procedimiento a seguir se concreta en:

Paso 1: Se consulta el árbol *B* del primer nivel que contiene las llaves de los controladores definidos en la configuración; si no existe el controlador se termina el algoritmo regresando error; en caso contrario se consulta la información del controlador y se termina al algoritmo indicando éxito.

4.3.3 Consulta de la ley de control de un controlador virtual

Para poder evaluar las funciones de los controladores virtuales es necesario obtener de la base de datos la expresión de la función en particular. Para ello se debe seguir el procedimiento.

Paso 1: Se consulta el árbol *B* del primer nivel que contiene las llaves de los controladores virtuales definidos en la configuración; si no existe el controlador se termina el algoritmo regresando error.

Paso 2: Se revisa que el controlador virtual tenga definida una ley de control. Si no existe, se termina el algoritmo regresando error; en caso contrario, se puede consultar la ley de control y terminar el algoritmo indicando éxito.

Se hace notar que las leyes de control del tipo 2 y 3 cuando se les define en la configuración y se almacenan en la base de datos en disco, están descritas como cadenas de caracteres y es necesario transformarlas para que puedan ser evaluadas. Esta tarea se le asignó al ENLAZADOR DE ACTIVIDADES ya que sólo se requiere ejecutar al instalar el COORDINADOR

EN TIEMPO REAL. A continuación se describe el algoritmo propuesto para la transformación.

1. Genera un árbol de la expresión (leyes del tipo 2 y 3).
2. Recorre el árbol en posorden y genera al mismo tiempo una pila con la expresión en notación polaca posfija.
3. Deposita la pila generada en la base de datos en memoria.

Por el contrario las leyes de control tipo 1 pueden transferirse a la base de datos en memoria sin problema, ya que su representación en la base de datos en disco es similar (matricial y vectorial).

4.3.4 Generación de las llaves en los árboles B

Para tener acceso a las características y variables de los controladores dentro de los árboles B (ver fig 21), se utilizan llaves que se generan empleando los criterios:

- a) Se fijó un número máximo de 100 controladores (físicos y virtuales) para ser operados por el sistema de control distribuido, por lo que las llaves C_i para el árbol de los controladores, toman los valores entre 0 y 99.
- b) Como el número de entradas/salidas (E_i/S_i) varía en función del tipo de controlador se tomó la representación XCCYY para los árboles de las entradas/salidas con el siguiente código:

letra	intervalo	significado
X	0 - 1	0 = entrada, 1 = salida
CC	0 - 99	número del controlador al que pertenece la entrada/salida
YY	0 - 99	número de la entrada/salida

Para implantar los conjuntos Ent-CF y Ent-CV y evitar el manejo redundante de información dentro de los árboles B (fig 21), se propone: copiar en los árboles de entradas/salidas, las llaves que vinculan los elementos de Ent-CF y Ent-CV en los registros de las salidas de los

controladores *VIRTUALES* Y físicos, respectivamente. Esto no representa ningún problema para tener acceso a la información de los registros, porque en la propia llave (XCCYY) se incluye el número del controlador y su salida.

Se puede observar que en el código C_1 (ver fig 21), no se puede diferenciar cuándo se trata de un controlador físico o cuándo de un virtual. Además, para el caso particular de los controladores que establecen comunicación, no se puede distinguir en el código XCCYY, cuándo se trata de una entrada y cuándo de una salida. Por lo que es necesario utilizar un campo extra que permita realizar esta distinción. El código utilizado para éste, es denominado *tipo* y se describe a continuación:

<i>tipo</i>	significado
0	controlador físico
1	controlador virtual
2	entrada de un controlador físico
3	salida de un controlador físico
5	salida de un controlador virtual
1CF	entrada de un controlador virtual con salida del controlador físico CF

El *tipo* 1CF se escogió para codificar en los dos caracteres CF, el número del controlador físico de quien recibe información el controlador virtual. Esto se hizo para dar rapidez a la búsqueda de la información.

Se hace la observación que las bases de datos de disco y memoria están construidas con la misma técnica; sin embargo, existen dos detalles que las distinguen. La base de datos en memoria tiene restringida la capacidad para almacenar las variables de entradas/salidas de los controladores, estimándose conveniente limitarla a 25 datos, ya que las frecuencias de muestreo en un proceso industrial son del orden de decenas de minutos. Una posible alternativa que puede implantarse es ajustar la capacidad en función de la memoria; sin embargo, esta solución no está implantada actualmente.

En la realización de la base de datos en disco se estimó conveniente -con el objeto de acelerar las búsquedas dentro de los árboles B- crear un

área de almacenamiento en memoria para depositar los nodos de los árboles B. Esto permite tener los nodos de los árboles que se usan con más frecuencia en memoria y agiliza las búsquedas.

4.4 CONFIGURACIÓN

Con base en la división de tareas del supervisor, descritas en el inciso 4.1, se puede decir que la unidad de *CONFIGURACION* tiene como actividad principal permitir al operador especificar las características de la red de controladores físicos y virtuales que se desea supervisar.

Para evitar inconsistencias tanto dentro de la base datos como en el sistema en general, la unidad de *CONFIGURACION* fue diseñada con base en las siguientes operaciones:

a) Definir los controladores físicos junto con sus características propias; es decir, asignar los valores de sus parámetros K_p , T_i y T_d , su referencias $r(t)$, su tolerancia ($r^+%$ y $r^-%$), su intervalo de trabajo ($V_{máx}$ y $V_{mín}$), su periodo de adquisición de datos, sus nombres y fechas de creación.

b) Definir los controladores virtuales junto con sus características propias. En este caso, se requiere establecer el tipo de ley de control, el número de entradas y salidas y el intervalo de evaluación.

c) Asignar la función de un controlador virtual de manera consistente con las características definidas en el punto b. Es decir, una sola ecuación recursiva, algebraica o lógica con el número de entradas y salidas definidas de antemano.

Junto con las operaciones arriba definidas, la unidad de *CONFIGURACION* está dotada de un mecanismo que prohíbe la violación de las reglas.

a) Deben cumplirse las reglas de sintaxis de acuerdo con el tipo de función asignada a cada ley de control.

- b) Debe conectarse sólo una salida de los controladores virtuales a la entrada de uno físico.
- c) Debe existir al menos una salida de un controlador físico en la descripción de las leyes de control.
- d) Deben existir todos los controladores físicos asociados a las leyes de control.
- e) Debe deshabilitarse primero un controlador virtual antes que el físico cuando existe acoplamiento entre ellos.
- f) Deben deshabilitarse primero las leyes de acción de los controladores virtuales antes de darlos de baja.
- g) Deben cumplirse las relaciones: $r^+ \% > r^- \%$ y $V_{\text{máx}} > r(t) > V_{\text{mín}}$.

Para asegurarse de que el inciso a) no se viola, se vigilan -al describir la ley de control- las reglas de gramática asociadas a cada una de ellas. En el caso particular de las leyes del tipo 2 y 3 (no lineales y lógicas), las reglas que las rigen se describen en los Apéndices 1 y 2 respectivamente. Para las leyes de tipo 1, se verifican únicamente las dimensiones de las matrices de la ecuación recursiva (ec 1 del inciso 2.3.2).

Una vez definida la red, de acuerdo a las reglas arriba establecidas, ésta se almacena en la base de datos. Además se puede solicitar por medio de la impresora, un reporte de la red almacenada. Esto último puede servir como respaldo en papel la configuración de la red.

Se puede observar, que es una ventaja vigilar las especificaciones de la red cuando ésta se configura, ya que reduce las tareas que deben ejecutarse en tiempo real.

Para facilitar la definición de la red al operador, la unidad de *CONFIGURACION* cuenta con un manejador de ventanas donde aparece texto. Esto permite operar un conjunto de ventanas almacenadas en disco, con información predefinida, del tipo menús y comentarios. En el Apéndice 3 se muestran algunas ventanas utilizadas en la *CONFIGURACION*.

4.5 LISTA DE DESCRIPTORES

En los incisos anteriores se presentó la forma en que se maneja la configuración en memoria y en disco; sin embargo, no se ha hablado de los mecanismos para manipular los datos dinámicos de la red (los procedentes de los controladores físicos y los generados al aplicar las leyes de control). Para realizar esta tarea se hace uso del concepto de descriptor o vector de estado, que se define como una estructura constituida por un conjunto de datos que especifican el estado en curso de los controladores físicos y virtuales.

Debido a que se tienen dos tipos diferentes de controladores (físicos y virtuales), con características especiales cada uno de ellos, se propone emplear dos listas de descriptores. Estas listas se instalarán en memoria RAM para agilizar su consulta.

4.5.1 Descriptores de los controladores físicos.

En el capítulo 2 se detallan las características de un controlador físico formado por seis parámetros (K_p , T_i , T_d , T_{mp} , $r^+ \%$, $r^- \%$); dos variables ($r(t)$ y $y(t)$) y su condición de falla. Esta información junto con sus posibles llaves de vinculación con la red y los estados de acuse de recibo, implica que se requiere un descriptor con trece casillas por controlador.

A fin de simplificar la búsqueda de controladores con un estado o condición de operación igual, (referencia o parámetros transmitidos, alarma, cambio de referencia), se emplea la casilla reservada al estado dentro del descriptor como apuntador hacia el siguiente, con igual condición. De este modo, se tiene una lista de controladores en el mismo estado.

A continuación se desglosa la información contenida dentro de cada casilla del descriptor:

1. Apuntador al siguiente descriptor.

2. Identificador del descriptor que corresponde con el número de controlador físico instalado en la red y con la llave definida para el controlador en la base de datos en memoria.
3. Prioridad del controlador físico definida en el momento de configurar el sistema. Esta se emplea primeramente para atender las alarmas con mayor jerarquía.
4. Llave de la salida del controlador virtual, manejada por la base de datos en memoria, que se comunica con el controlador físico. Esta llave se utiliza para poner una marca de falla u operación del controlador físico en la lista de controladores físicos, manejada por el descriptor del controlador virtual con el que se establece comunicación (véase fig 23).
5. Llaves de la entrada y salida del controlador físico manejada por la base de datos en memoria. Estas permiten depositar con rapidez la información del controlador físico en la base de datos en memoria.
6. Apuntador al siguiente descriptor asociado con un controlador en estado de alarma que no ha sido atendido.
7. Apuntador al siguiente descriptor asociado con un controlador en estado de modificación de sus parámetros (K_p , T_i , T_d) por el operador y que no han sido transmitidos.
8. Apuntador al siguiente descriptor asociado con un controlador, al cual ya se le transmitió la orden de cambio de parámetros y se espera el acuse de recibo del controlador físico.
9. Apuntador al siguiente descriptor asociado con un controlador en estado de modificación de su referencia y tolerancia ($r(t)$, $r^{+}\%$, $r^{-}\%$), y que no han sido transmitidas.
10. Apuntador al siguiente descriptor asociado con un controlador, al cual ya se le transmitió su cambio de referencia y se espera el acuse de recibo del controlador físico.
11. Espacio donde se codifica el tipo de alarma presente. El tipo de alarma está ligado con el apuntador del estado de alarma.
12. Parámetros K_p , T_i y T_d , que deben ser enviados. Estos están ligados con el apuntador de cambio de parámetros.
13. Tres espacios donde se almacenan la referencia $r(t)$ y su tolerancia $r^{+}\%$ y $r^{-}\%$, que serán enviadas. Estas están ligadas con el apuntador de cambio de referencia.

Como se puede ver, en el descriptor se manejan seis listas, por lo que es necesario el uso de seis apuntadores que sirvan de cabezas para cada una. Dentro de las listas, la de los descriptors es circular y las restantes lineales. La fig 22 muestra un ejemplo de un posible estado de los controladores físicos.

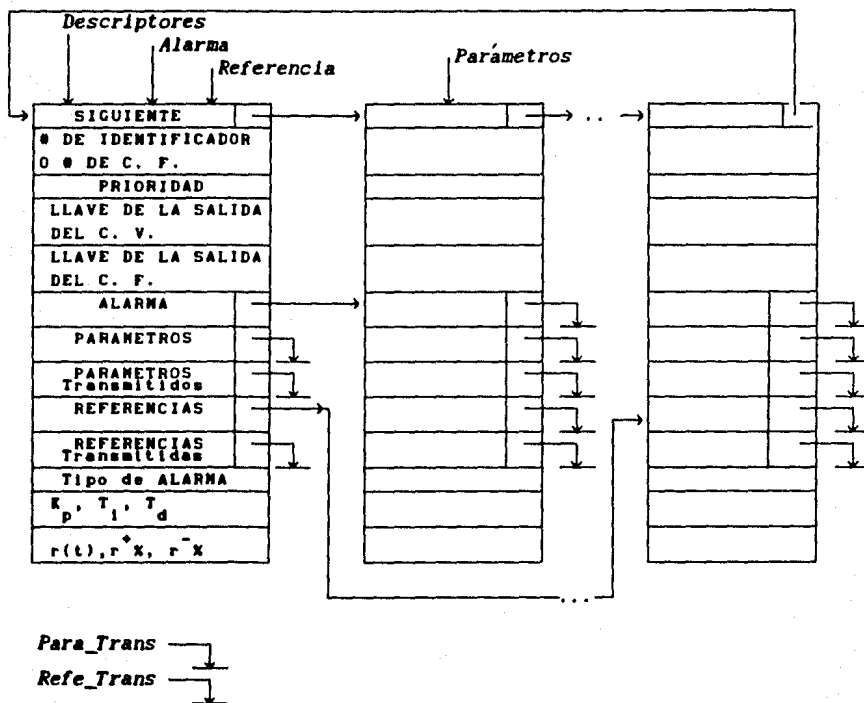


Fig 22. Lista de descriptors de controladores físicos

4.5.2 Descriptores de controladores virtuales

Las tareas que están relacionadas con los controladores virtuales son la aplicación de las leyes de control, el almacenamiento de los resultados en la base de datos en memoria y la transmisión de los resultados a los controladores físicos con los que tienen comunicación. Tomando como base estas tres tareas los descriptores de los controladores virtuales contienen la siguiente información en cada una de sus casillas.

1. Apuntador al siguiente descriptor.
2. Identificador del descriptor correspondiendo con el número de controlador y con la clave definida para éste en la base de datos en memoria.
3. Apuntador al siguiente descriptor con una ley de control que debe ser evaluada.
4. Dos espacios donde se almacena el periodo de tiempo para la evaluación de la ley de control y el tiempo transcurrido desde su última evaluación.
5. Apuntador a la lista de controladores físicos a quienes se les deben enviar los resultados de la aplicación de la ley de control.

Cabe señalar que en este caso los descriptores manejan únicamente dos listas, por lo cual resulta suficiente tener dos apuntadores que sean las cabezas de cada una de ellas. Una de las listas es circular y la otra lineal. La fig 23 muestra un ejemplo de un posible estado de los controladores virtuales.

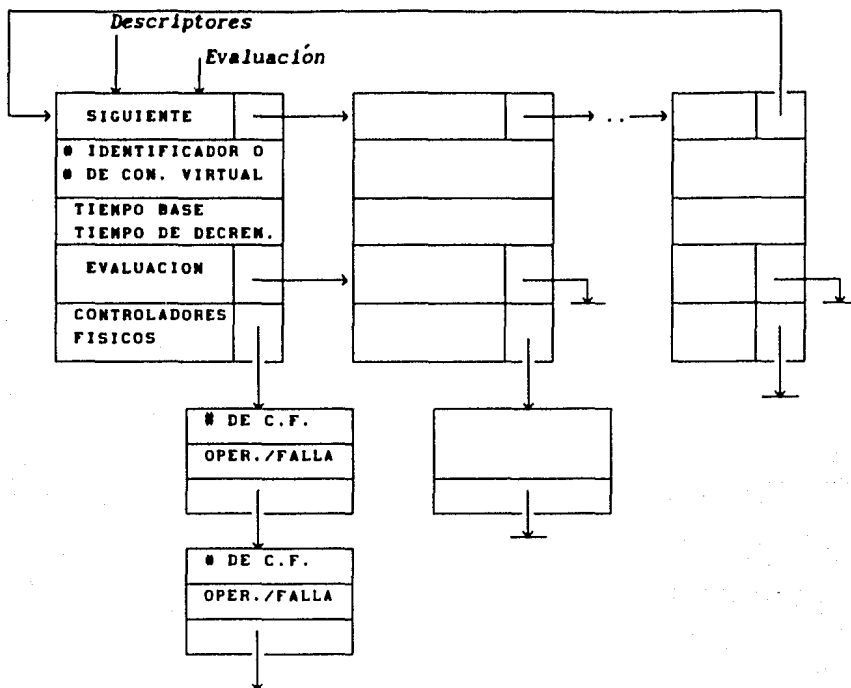


Fig 23. Lista de descriptores de los controladores virtuales

4.5.3 Lista de alarmas.

Como en los descriptores de los controladores virtuales no se puede detectar si alguna ley de control no ha sido evaluada porque no se recibe información de alguno o varios controladores físicos que están en estado de alarma, se utilizó una lista lineal denominada *Con_en_falla* que contiene todos los controladores físicos y virtuales en estado de alarma o falla. Esto permite que el operador pueda conocer rápidamente los estados de emergencia del proceso.

4.6 COORDINACION EN TIEMPO REAL.

De acuerdo con la asignación de tareas presentada en el inciso 4.1, la función principal de la unidad de *COORDINACION EN TIEMPO REAL*, es demandar el tiempo de proceso de la computadora para administrarlo y poder atender las siguientes actividades:

- a) Adquirir a través del sistema de comunicaciones, la información de los controladores físicos y almacenarla en la base de datos en memoria.
- b) Desplegar los estados de alarma reportados por los controladores físicos.
- c) Evaluar las leyes de control de los controladores virtuales y almacenar resultados en la base de datos de memoria.
- d) Transferir cuando se requiera, las acciones provenientes de los controladores virtuales a los físicos a través del sistema de comunicación.
- e) Copiar la base de datos de memoria a disco con una frecuencia preestablecida.
- f) Desplegar información del estado del sistema cuando el operador lo solicita.
- g) Establecer comunicación con el operador a través de una terminal, para realizar actividades específicas.
- h) Imprimir reportes del estado del proceso por cumplimiento de tiempo o a solicitud del operador.

Como se puede ver las actividades a, b, c y d, dependen de las variables y características de los controladores físicos y virtuales, por lo que para su manejo el *CTR* hace uso de las dos listas de descriptores, definidas en el inciso 4.5.

Respecto a las actividades restantes se observa que ellas requieren de la impresora y la terminal, por lo que al ejecutarse éstas toman el control y no se puede atender ninguna otra tarea bajo el sistema operativo *MSDOS*. Para resolver esta dificultad se planea implantar un sistema de comunicación hombre-máquina, capaz de imprimir reportes y establecer una

comunicación entre el sistema y el operador sin bloquear otras tareas del CTR. La implantación de este sistema se encuentra actualmente en desarrollo por personal del Instituto de Ingeniería, y en el presente trabajo no se profundiza sobre dicho sistema.

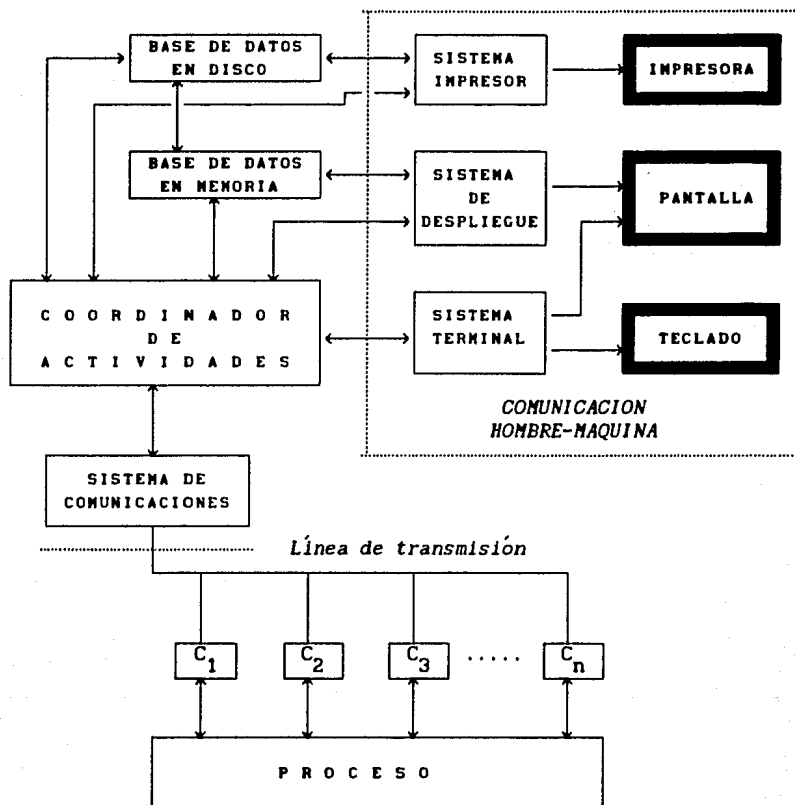


Fig 24. Supervisor en tiempo real

En la fig 24 se muestra cómo interactúa el CTR con el sistema de comunicación hombre-máquina.

Tomando como herramientas las listas de descriptores para los controladores físicos y virtuales, la lista de *Con-en-falla*, el sistema de comunicación hombre-máquina y el sistema de comunicación, se definió

el siguiente algoritmo para la unidad CTR:

Paso 1: Obtener información de los controladores físicos que hayan cumplido su periodo de muestreo mediante el sistema de *comunicación*. Si se presentara el caso de recibir información de un controlador que anteriormente estaba en la lista de *Con-en-falla*, éste se elimina.

Paso 2: Actualizar la base de datos en memoria con la información obtenida de los controladores físicos. En caso de no tener espacio disponible en memoria se transfieren los datos al disco para crear espacio y posteriormente depositar la información.

Paso 3: Capturar mediante el sistema de *comunicación* todas las alarmas de los controladores físicos y dar avisos por el monitor de acuerdo con sus prioridades y guardarla en la lista de *Con-en-falla*. Durante esta etapa se trabaja con las listas de alarma y la de *Con_en_falla*.

Paso 4: Actualizar los tiempos de evaluación de las leyes de control e insertar en la lista de aplicación los controladores virtuales a los cuales se les debe aplicar su ley de control en función de sus prioridades. Antes de colocarlo en la lista de aplicación se revisa que sus variables estén actualizadas y completas; en caso negativo se inserta en la lista de *Con-en-falla*. Si se presenta el caso de que estén sus entradas completas y actualizadas y el controlador se encuentra en la lista de *Con-en-falla*, se elimina de ésta y se inserta en la lista de aplicación.

Paso 5: Evaluar las leyes de control que están en la lista de aplicación.

Paso 6: Depositar los resultados de las leyes de control en la base de datos en memoria. En caso de no tener lugar disponible para almacenar los resultados, se transfieren los datos de memoria a disco para crear espacio y se guarda la nueva información en memoria.

Paso 7: Transmitir los resultados de las leyes de control a los

controladores físicos que lo requieran mediante el sistema de *comunicación*.

Paso 8: Atender mediante el sistema de *terminal*, la solicitud del operador de cambio de parámetros o referencia para un controlador físico y modificar los datos necesarios en la lista de descriptores de los controladores físicos.

Paso 9: Enviar al controlador físico un cambio de parámetros si existe dentro de la lista de solicitudes correspondientes e indicar el envío en la lista de parámetros transmitido donde permanecerá hasta obtener un acuse de recibo.

Paso 10: Enviar un cambio de referencia si existe dentro de la lista de solicitudes correspondientes e indicar el envío en la lista de referencias transmitidas donde permanecerá hasta obtener un acuse de recibo.

Paso 11: Obtener mediante el sistema de *terminal* los acuses de cambios de parámetros y referencias y actualizar las listas correspondientes.

Paso 12: Atender mediante el sistema *terminal*, cuando lo solicite el operador, el despliegue de las variables de un controlador (físico o virtual) y el modo en el que se desea visualizar la información.

Paso 13: Desplegar la información mediante el *sistema de despliegue*

Paso 14: Revisar si se cumplió el periodo para la impresión de un informe. En caso afirmativo se ejecuta la tarea por medio del sistema de *impresión*.

Paso 15: Atender a través del sistema de *terminal*, la solicitud del operador para imprimir un informe mediante el sistema de *impresión*.

Paso 16: Atender mediante el sistema de *terminal*, la solicitud de suspender la ejecución del *COORDINADOR EN TIEMPO REAL* para

reconfigurar el sistema. Si esta condición se cumple se guarda la información de la base de datos de memoria en disco y se interrumpe el ciclo del *CTR* transfiriendo el control al programa *ENLAZADOR*.

Paso 17: Regresar al paso 1.

Con respecto al paso 5 en donde se evalúan las leyes de control se realiza alguno de los dos procedimientos:

Para la ley tipo 1. Primeramente se obtienen los valores de las entradas que corresponden al vector $U(KT+T)$ de la ecuación recursiva; después se calcula el producto y suma de matrices y finalmente se almacena el resultado en la base de datos en memoria.

Para las leyes tipo 2 y 3. Primeramente se copia la pila que contiene la ley de control y se actualizan las variables (entradas del controlador) con respecto a los valores de la base de datos en memoria. Posteriormente, con ayuda de una segunda pila, se evalúa la expresión y el resultado se almacena en la base de datos en memoria.

4.7 COMUNICACIÓN ENTRE EL SUPERVISOR Y LOS CONTROLADORES FÍSICOS

A partir del capítulo 2 se ha hablado de la necesidad de un sistema de comunicación que vincule el mundo físico de los controladores con el programa supervisor. En el Apéndice 5 se describe este sistema que fue desarrollado por personal del Instituto de Ingeniería.

Para utilizar el sistema de comunicación, primeramente deben realizarse los siguientes pasos:

1. Cargar el programa del sistema de comunicación en memoria RAM. Esta tarea la realiza el *EF*.
2. Proporcionar la lista de controladores físicos instalados en la red al programa del sistema de comunicación, tarea que también realiza el *EF*.

Una vez instalado el sistema de comunicación comienza su trabajo con las siguientes actividades en forma automática:

- a) Se comunica con cada uno de los controladores físicos cuando se cumple su periodo de muestreo para solicitarle su salida ($y(t)$) y la almacena en una área temporal hasta que el *CTR* se la solicite.
- b) Se enlaza con cada uno de los controladores físicos en tiempos intermedios (cuando no se solicita las $y(t)$), para demandarles su estado operativo (alarma) y lo almacena en una área temporal hasta que el *CTR* se los solicite.

Cuando el *CTR* requiere transmitir un mensaje especial como cambio de parámetros, referencia o alguna salida de un controlador virtual, se lo indica al sistema de comunicación, proporcionándole el número de controlador y el mensaje que se va transmitir. El sistema de comunicación transmite los mensajes especiales en el orden en que se proporcionaron y los acuses de recibo de los controladores físicos se ponen en una área temporal hasta que el *CTR* las solicite.

En la comunicación entre el sistema de comunicación y los controladores físicos, existe un protocolo a nivel de paquete de información y dentro de este paquete se manejan otros tres protocolos que permiten conocer su salida, su estado de alarma y la transmisión de un mensaje especial. En el Apéndice 4 se explican con detalle los protocolos empleados en la comunicación con los controladores físicos.

CONCLUSIONES

Se ha definido una arquitectura para sistemas de control distribuido que permite comprender su estructura y funcionamiento. Entre las características más importantes que incorpora la arquitectura están:

- La posibilidad de manejar unidades elementales llamadas controladores físicos que operan sobre una variable de un proceso y tienen capacidad de comunicarse con un supervisor.
- La factibilidad de crear unidades complejas llamadas controladores virtuales que permiten realizar el control de un proceso de manera global, tomando en cuenta la información de varias variables proporcionadas por un conjunto de controladores físicos.
- La comunicación que se puede establecer entre los dos controladores (físicos y virtuales) en función de su tipo y de los requisitos del control que se desea realizar.

Se implantó un supervisor para sistemas de control distribuido, que considera por un lado la arquitectura definida y por el otro el conjunto de necesidades básicas que impone la operación de un sistema de control. La definición de la programación se basó en un esquema general de tareas y de flujo de información entre ellas, a partir del cual se identificaron dos bloques fundamentales de tareas: la configuración del proceso y la supervisión del mismo en tiempo real. El supervisor tiene la capacidad de ser aplicado en diferentes tipos de plantas industriales mediante el mecanismo de configuración. Una característica importante de la programación fue el diseño de una base de datos que satisface los requerimientos del sistema supervisor y que puede ser instalada en disco y en memoria.

El sistema desarrollado satisface los objetivos que motivaron su realización: supervisa una red de controladores distribuidos y permite implantar algoritmos de control propuestos por investigadores del área de control del Instituto de Ingeniería. Al comparar este sistema con otros comercialmente disponibles se encuentra que permite el soporte de lazos individuales de control mediante mallas PID, como ocurre con la mayor parte de dichos sistemas. Por otro lado, también permite manejar

estructuras complejas de control a través de los controladores virtuales, lo que lo equipara con sistemas avanzados de control distribuido que basan su funcionamiento en diagramas de flujo.

El sistema supervisor que aquí se describe forma parte de un proyecto para implantar una red de control distribuido que se encuentra todavía en fase de desarrollo. Si se toman como base las figs 18 y 24, a la fecha se han cubierto las siguientes etapas: programación del configurador de procesos, bases de datos de disco y memoria. El coordinador de actividades está terminado, pero se encuentra en fase de prueba. Con respecto a la comunicación hombre-máquina, se están definiendo los protocolos de comunicación entre el hombre y la máquina, para lo cual se han tomado como marcos de referencias las diferentes técnicas usadas en los sistemas comerciales.

El sistema operativo MSDOS de la PC utilizado en la implantación del sistema supervisor, restringió grandemente el desarrollo del sistema en el uso de las herramientas y las técnicas de programación. Esta limitación se notó especialmente en la programación correspondiente en la comunicación entre controladores y la relacionada con el supervisor en tiempo real (adquisición de datos, tiempos de aplicación de leyes de control, manejo de alarmas, etc). Por tanto, se recomienda que en desarrollos futuros, se seleccione una computadora que tome en consideración los requisitos especificados en la arquitectura del sistema de control distribuido y sobre todo que se cuente con un sistema operativo diseñado para tiempo real.

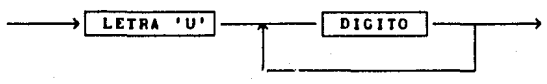
REFERENCIAS

1. Alsys, S A, Alsys PC AT Ada compiler, version 1.2 , Ed Alsys, SA, 1986.
2. Astron, Karl J. y Witternmark, Bjorn, Computer controlled systems, Ed Prentice-Hall, 1984.
3. Bennett, Stuart, Real time computer control: An introduction, Ed Prentice Hall, 1988.
4. Comer, Douglas, The ubiquitous B-tree, ACM Trans. Database Syst., (11) II (June, 1979), pp 121-137.
5. Cramer, Bill, Real-Time programs under UNIX, Dr. Dobb's Jurnal (June 1988).
6. Equinox Data Corporation, Loopworks for the IBM industrial computer system version 2.00, Ed Equinox Data Corporation, 1987.
7. Ford, Gary y Wiener, Richard, Modula-2 : A software development approach, Ed John Wiley & Sons, 1985.
8. Kuo, Benjamin C, Automatic Control Systems, Ed Prentice-Hall, 1975.
9. Kompass, E J, Reviewing PC-based software for control engineering, Control engineering, (35) XXXV (Novembre 1988), pp 57-60.
10. Lipaev, V V, Computer-aided design of software for control systems, Computer in control, II (February 1982), pp 212-221.
11. Logitech Inc, Modula-2/86, Ed Logitech Inc, 1986.
12. Murrill, Paul W, Fundamentals of process control theory, Ed Instrument Society of America, 1981.
13. Tsichritzis, Dionysios C y Lochovsky, Frederick H, Data basemanagement systems, Ed Academic Press, 1977.
14. Ullman, Jeffrey D, Principles of database systems, Ed ComputerScience Press, 1982.
15. Williams, Theodore J, The use of digital computers in processcontrol, Ed. Instrument Society of America, 1984.
16. Wirth, Niklaus, Algorithms + Data structures = Programs, Ed Prentice-Hall, 1976.
17. Wirth, Niklaus, Programming in Modula-2, Ed Springer Verlag, New York, 1982.

APÉNDICE 1

DIAGRAMAS SINÁCTICOS DE LAS LEYES DE CONTROL DEL TIPO 2.

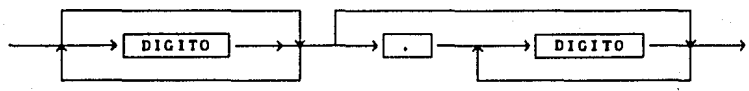
Identificador



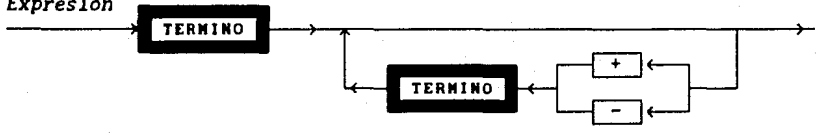
Identificador de funcion



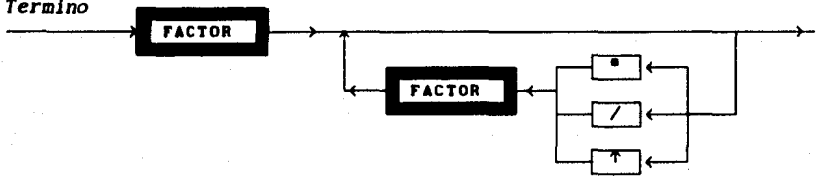
Real sin signo

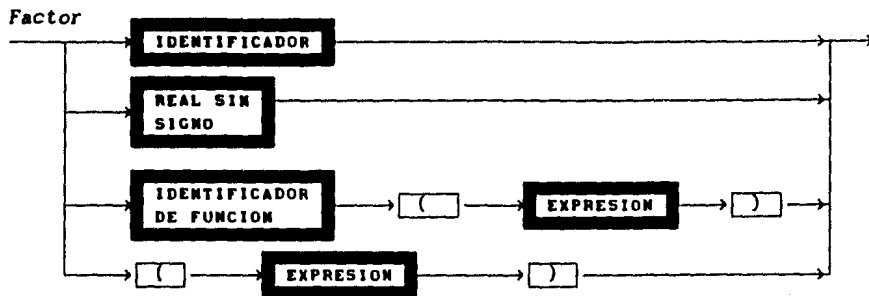


Expresión



Término





Ecuación algebraica



SINTÁXIS DE LAS LEYES DEL TIPO 2 EN FORMA BNF.

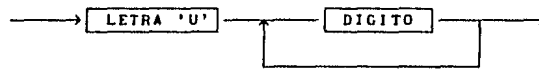
```

<identificador> ::= <U> {<dígito>}
<identificador de función> ::= <letra> {<letra>}
<real sin signo> ::= <dígito> {<dígito>} . <dígito> {<dígito>} | . <dígito>
{<dígito>} | <dígito> {<dígito>}
<signo> ::= +|-
<expresión> ::= <término> | <signo> <término>
<operadores> ::= * | / | ↑
<término> ::= <factor> | <operadores> <factor>
<designador de función> ::= <identificador de función> ( <expresión> )
<factor> ::= <identificador> | <real sin signo> | ( <expresión> ) |
<designador de función>
<ecuación algebraica> ::= <expresión>
  
```

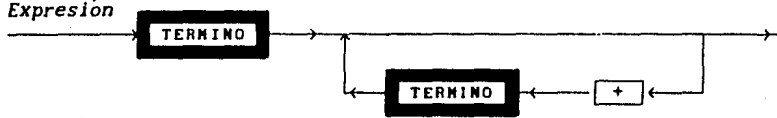
APÉNDICE 2

DIAGRAMAS SINTÁCTICOS DE LAS LEYES DE CONTROL DEL TIPO 3.

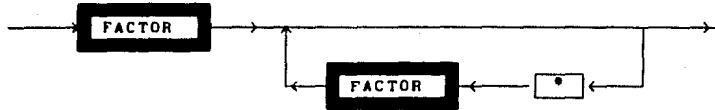
Identificador



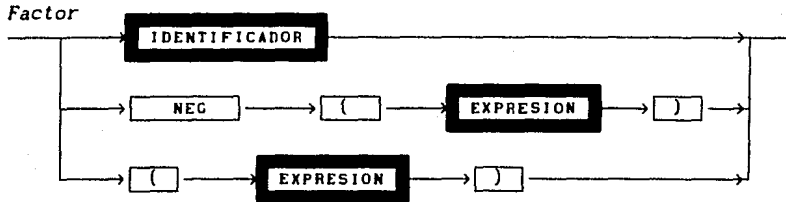
Expresión



Término



Factor



Ecuación lógica



SINTÁXIS DE LAS LEYES DEL TIPO 3 EN FORMA BNF.

<identificador> ::= <U> {<dígito>}

<disyunción> ::= +

<expresión> ::= <término> | <disyunción> <término>

<conjunción> ::= *

<término> ::= <factor> | <conjunción> <factor>

<designador de función> ::= NEG (<expresión>)

<factor> ::= <identificador> | (<expresión>) | <designador de función>

<ecuación lógica> ::= <expresión>

APÉNDICE 3

<p style="text-align: center;">CONTROLADORES</p> <p>a) Crear, revisar o modificar la información de un controlador</p> <p>b) Dar de baja un controlador</p> <p>c) Desplegar los controladores dados de alta</p>	<p style="text-align: center;">VARIABLES DE E/S</p> <p>d) Crear, revisar o modificar la información de una variable</p> <p>e) Dar de baja una variable</p> <p>f) Desplegar las variables de un controlador</p>
<p style="text-align: center;">ECUACION DE ESTADO</p> <p>g) Crear, revisar o modificar la ec. de estado de un controlador virtual</p> <p>h) Dar de baja una ecuación de estado</p>	<p style="text-align: center;">ECUACIONES ALGEBRAICAS O LOGICAS</p> <p>i) Crear, revisar o modificar la ecuación de una variable</p> <p>j) Dar de baja una ecuación</p>
<p>k) Imprimir la configuración</p>	<p>Esc) Terminar la configuración</p>

Opciones disponibles para configurar el sistema supervisor (menú).

<p>CONTROLADORES: Esc) Terminar</p> <p>00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24</p> <p>25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49</p> <p>50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74</p> <p>75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99</p>

Selección de controladores disponibles que pueden operar en la red de control distribuido.

CONTROLADOR FISICO: 5
 Nombre : CONTROLADOR FISICO # 5 Kp : 5.5
 Ti : 4.5 Td : 2.1
 Vmáx : 10 Vmín : 5
 Referenc.: 7.5 Error(+%) : .5
 Error(-%) : .5 Periodo : 5
 Prioridad: 10

Información característica de un controlador físico.

CONTROLADOR VIRTUAL: 1 EXISTE ECUACION DE ESTADO
 Nombre : CONTROLADOR VIRTUAL # 1 Tipo: 0
 Periodo: 5

Información característica de un controlador virtual con tipo de ley de control 1

CONTROLADOR VIRTUAL: 2
 Nombre : CONTROLADOR VIRTUAL # 2 Tipo: 1
 Periodo: 5

Información característica de un controlador virtual con tipo de ley de control 2.

CONTROLADOR VIRTUAL: 3
 Nombre : CONTROLADOR VIRTUAL # 3 Tipo: 2
 Periodo: 5

Información característica de un controlador virtual con tipo de ley de control 3.

VARIABLES DEL CONTROLADOR VIRTUAL: F1) Entradas F2) Salidas Esc) Terminar																												
ENTRADAS:																												
U00	U01	U02	U03	U04	U05	U06	U07																					
U08	U09	U10	U11	U12	U13	U14	U15																					
U16	U17	U18	U19	U20	U21	U22	U23																					
U24	U25	U26	U27	U28	U29	U30	U31																					
U32	U33	U34	U35	U36	U37	U38	U39																					
U40	U41	U42	U43	U44	U45	U46	U47																					
U48	U49																											
SALIDAS:																												
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24				
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49				

Selección de una entrada y/o salida que se puede crear en un controlador virtual.

INFORMACION DE LA VARIABLE: 1
 Nombre : Presión
 SQR (U0) + 10 * SIN (U0)

CONTROLADOR: 1
 Controlador fis. : 5

Información de una salida de un controlador virtual y ley de control del tipo 2.

DIMENSIONES: Ren. de A y B, y Col. de A y C (1..5): Ren. de C y D (1..5):
 Edición de los vectores (Uk,Yk): Col. de B y D (1..5):

MATRIZ A: [2,2]
 0.000 1.000
 -0.200 0.800

Uk[i]	Yk[i]
U00	Y01
---	---
---	---
---	---

MATRIZ B: [2,1]
 0.000
 1.000

MATRIZ C: [1,2]
 1.000 0.400

MATRIZ D: [1,1]
 0.000

Editor de la ecuación de estado.

Nota: la ecuación no debe exceder de 80 caracteres
 SQR (U0) + 10 * SIN (U0)

Editor de la ecuación algebraica o lógica.

APÉNDICE 4

DESCRIPCIÓN DE LOS MENSAJES QUE SE INTERCAMBIAN ENTRE EL CONTROLADOR MONOVARIABLE Y EL PUESTO CENTRAL DE CONTROL

Este documento presenta en forma detallada el contenido de todos los mensajes se transmiten entre el controlador monovariable y el supervisor.

ESTRUCTURA DE LOS MENSAJES

Los mensajes se componen de bloques de caracteres ASCII, que se pueden a su vez dividir en subconjuntos, cuya estructura general se muestra en la siguiente figura:



Fig. 1 Estructura general de los mensajes

donde

- INI - caracteres que indican el inicio de un mensaje.
- IDEN - identificación del controlador hacia el que se dirige la información, o del que proviene la misma.
- TIPO - tipo del mensaje en tránsito.
- MENSAJE - contenido del mensaje.
- FIN - caracteres de fin de mensaje.
- CHECK - paridad o suma vertical.

Los bloques de caracteres pueden tener longitud variable, siempre y cuando esta no exceda en total a los 128 caracteres.

BLOQUES DE INICIO Y FIN

De estos seis subconjuntos, el de INI y FIN son siempre fijos. Cada uno se compone de dos caracteres ASCII no desplegables, según se describe a continuación:

DLE	STX
-----	-----

INI

DLE	ETX
-----	-----

FIN

Fig. 2 Contenido de los bloques INI y FIN

El caracter DLE equivale a 10H (H = hexadecimal), STX a 02H y ETX a 03H.
El significado de los nemónicos empleados es:

DLE - liga de datos habilitada (Data Link Enable)
STX - principio de texto (Start of TeXt)
ETX - fin de texto (End of TeXt)

BLOQUES DE IDENTIFICACION Y TIPO

Cada uno de estos dos subbloques tiene la misma estructura:



Fig. 3 Estructura de IDEN y TIPO

donde N1 y N2 son caracteres ASCII numéricos hexadecimales, es decir:

$$N1, N2 \in \{ "0", \dots, "F" \}$$

lo que da posibilidad de manejar hasta 256 controladores y el mismo número de tipo de mensajes. A pesar de lo anterior, en esta primera etapa de desarrollo del sistema de control distribuido se ha limitado el intervalo de validez del tipo de mensaje, de tal suerte que:

$$\text{TIPO} \in \{ "01", "02", "03" \}$$

El significado del mensaje según el tipo es:

TIPO = "01" → ORDEN
TIPO = "02" → SOLICITUD
TIPO = "03" → ALARMA

BLOQUE DE MENSAJE

El bloque de mensaje contiene la información a intercambiar; es el único de longitud variable, ya que su tamaño puede oscilar entre 0 y 118 caracteres ASCII alfanuméricos. Su estructura es:

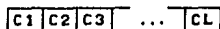


Fig. 4 Estructura del bloque MENSAJE

donde

$$C_i \in \{ "0", \dots, "9", "A", \dots, "F" \} \quad ; \quad i=1, \dots, L$$

$$L \in \{ 0, \dots, 118 \}$$

BLOQUE DE PARIDAD VERTICAL

Este bloque contiene también dos caracteres y sirve para verificar que no se han presentado errores durante la recepción de un mensaje. Su forma es:



Fig. 5 Estructura de CHECK

donde

$$P1, P2 \in \{ "0", \dots, "F" \}$$

P1 y P2 se calculan sumando el valor ASCII de los caracteres transmitidos desde INI hasta FIN, y obteniendo su módulo 256. Una vez hecho esto la cantidad restante se complementa a 256, se divide en dos caracteres ASCII hexadecimales que se transmiten como P1 y P2, donde P1 es el dígito más significativo y P2 el menos.

Si el mensaje tiene la siguiente estructura:

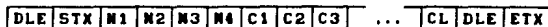


Fig. 6 Cálculo de la paridad vertical

la paridad vertical se calcula así:

$$SUMA = 256 - \left[\left(\sum_{i=1}^4 N_i + \sum_{i=1}^L C_i + 2 DLE + STX + ETX \right) \bmod 256 \right]$$

$$P1 = ASCII (SUMA \text{ DIV } 16)$$

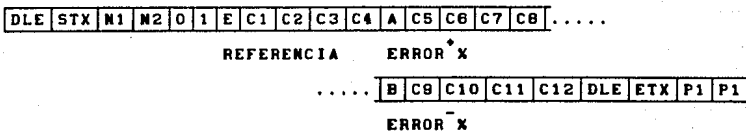
$$P2 = ASCII (SUMA \text{ MOD } 16)$$

ORDENES (MENSAJES TIPO = "01")

Las órdenes serán mensajes que transitan desde el puesto central hacia el controlador. Se distinguen dos tipos de órdenes:

- 1) Cambio de la referencia para la variable de entrada al controlador y de sus límites de validez alto y bajo.
- ii) Cambio de los parámetros del controlador.

La respuesta del controlador en todos los casos indicará únicamente que la información fue recibida.

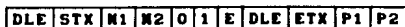
CAMBIO DE REFERENCIA Y LIMITES DE VALIDEZDirección Puesto Central-Controlador

donde

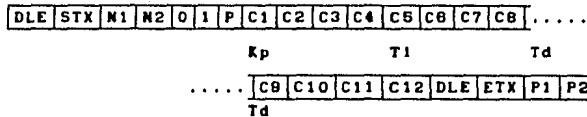
$C1 \in \{ "0", \dots, "9" \}$

$; i=1, \dots, 12$

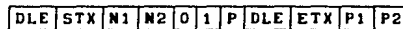
Las tres variables de este mensaje se forman con cuatro dígitos ASCII decimales cada una. El mensaje debe obligatoriamente contener la referencia, pero el límite alto y bajo son opcionales.

Dirección Controlador-Puesto Central

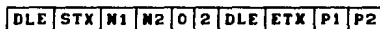
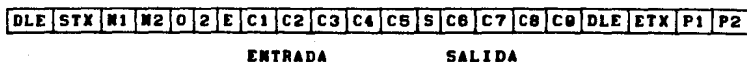
CAMBIO DE LOS PARAMETROS DEL CONTROLADOR

Dirección Puesto Central-Controlador

Los caracteres C1 se interpretan como en el caso anterior. Las tres variables en cuestión (Kp, T1 y Td) siempre se deben transmitir.

Dirección Controlador-Puesto Central**SOLICITUDES (MENSAJES TIPO = "02")**

Las peticiones de información del puesto central al controlador siempre se referirán al estado de la entrada y salida del controlador. El mensaje del puesto central es un mensaje nulo, con el tipo e identificador apropiados, y la respuesta del controlador siempre será un mensaje con la información solicitada.

Dirección Puesto Central-ControladorDirección Controlador-Puesto Central

ALARMAS (MENSAJES TIPO = "03")

En este caso las alarmas corresponden a solicitudes que tienen ratamiento especial. El formato de los mensajes se diseñó para inimizarse el flujo de información, pues se ha supuesto que será el tipo e mensaje que ocupe preferentemente el canal de comunicaciones. El puesto central enviará un mensaje sin información (longitud nula) y tendrá por respuesta un mensaje de a lo más un carácter.

Se han especificado siete tipos de respuesta, que corresponden con los posibles estados del controlador, visto desde el puesto central:

- i) Estado normal, no existe alteración o malfuncionamiento en el controlador.
- ii) Controlador muerto, indicará que la parte analógica del controlador está fuera de servicio.
- iii) Local, señalará que el controlador está en modo local y que no puede por tanto recibir órdenes del puesto central.
- iv) Reinicio, el controlador salió de servicio temporalmente y pudo haber perdido referencias y parámetros.
- v) Salida por arriba del máximo permitido.
- vi) Salida por abajo del mínimo permitido.
- vii) Señal de control saturada.

Los formatos de los mensajes se presentan a continuación.

Dirección Puesto Central-Controlador

DLE	STX	N1	N2	0	3	DLE	ETX	P1	P2
-----	-----	----	----	---	---	-----	-----	----	----

Dirección Controlador-Puesto Central

DLE	STX	N1	N2	0	3	DLE	ETX	P1	P2
-----	-----	----	----	---	---	-----	-----	----	----

ESTADO NORMAL

DLE	STX	N1	N2	0	3	M	DLE	ETX	P1	P2
-----	-----	----	----	---	---	---	-----	-----	----	----

CONTROLADOR MUERTO

DLE	STX	N1	N2	O	3	L	DLE	ETX	P1	P2
-----	-----	----	----	---	---	---	-----	-----	----	----

CONTROLADOR EN LOCAL

DLE	STX	N1	N2	O	3	R	DLE	ETX	P1	P2
-----	-----	----	----	---	---	---	-----	-----	----	----

REINICIO

DLE	STX	N1	N2	O	3	+	DLE	ETX	P1	P2
-----	-----	----	----	---	---	---	-----	-----	----	----

SALIDA ARRIBA MÁXIMO

DLE	STX	N1	N2	O	3	-	DLE	ETX	P1	P2
-----	-----	----	----	---	---	---	-----	-----	----	----

SALIDA ABAJO MÍNIMO

DLE	STX	N1	N2	O	3	S	DLE	ETX	P1	P2
-----	-----	----	----	---	---	---	-----	-----	----	----

SEÑAL DE CONTROL SATURADA

APÉNDICE 5

SISTEMA DE COMUNICACIÓN

El sistema de comunicación es el medio de enlace entre el supervisor y los controladores físicos. Su función es enviar y recibir los mensajes que permitan realizar las tareas de control y supervisión del proceso.

Para establecer la comunicación entre la computadora y los controladores, se utiliza un canal de comunicación lineal como se muestra en la siguiente figura:

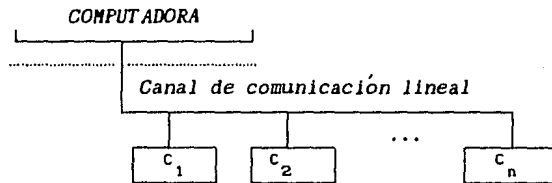


Fig 1. Canal de comunicaciones.

El canal de comunicación es del tipo bidireccional-simultáneo (full-duplex), que permite transmitir en ambos sentido. Por medio de este canal la computadora transmite los mensajes que pueden ser escuchados por todos los controladores y sólo responde aquel cuya clave corresponde con la que se insertó en el mensaje. En el Apéndice 4 se presentan los mensajes que pueden establecer entre la computadora y cada uno de los controladores.

El sistema de comunicación se desarrolló tomando el modelo ISO como referencia y consta de las capas: física, enlace, transporte y parte de la aplicación.

En la siguiente figura se presenta un esquema de la arquitectura utilizada en la implantación del sistema de comunicación, donde se pueden observar las diferentes capas que lo constituyen.

CAPA FÍSICA: Para realizar la capa física se utilizó el puerto serial de la computadora PC, el cual se programó para trabajar como full-duplex y con la capacidad de interrumpir cada vez que un carácter se transmite o recibe.

Las siguientes capas de enlace, transporte y la parte de aplicación, se llevan a cabo mediante un despachador de actividades que es activado periódicamente por medio de interrupciones al procesador, generadas por el reloj calendario de la computadora. Este despachador tiene la tarea de coordinar las actividades que se desarrollan en cada una de las capas y establecer los canales de comunicación entre ellas.

CAPA ENLACE: La capa de enlace está formada por un conjunto de programas desarrollados en ensamblador, con los cuales se realizan las actividades de transmisión y recepción de un mensaje.

Para transmitir un mensaje se sigue el algoritmo:

1. La capa de transporte transmite a la de enlace un mensaje que puede tener una longitud de hasta 118 caracteres.
2. La capa de enlace agrega al mensaje información de control para formar un paquete de 128 caracteres.
3. El paquete formado, es pasado carácter por carácter a la capa física, mediante una rutina de servicio de interrupción generada por el puerto serial cada vez que se transmite un carácter.

Para recibir un mensaje se sigue el algoritmo:

1. Mediante una rutina de servicio de interrupción que atiende al puerto serial cada vez que llega un carácter se va formando el paquete transmitido por el controlador.
2. Una vez completo el paquete se revisan sus caracteres de control para detectar errores en la transmisión.

3. Si no se encontró error en el paquete, se le quitan los caracteres de control y se pasa el mensaje a la capa de transporte. En caso contrario, cuando hay error, se le indica a la capa de transporte que se recibió un paquete con error.

CAPA DE TRANSPORTE: Está formada por un conjunto de programas en ensamblador con los que se manejan los diferentes tipos de mensajes utilizados en la red, así como el transporte de los mensajes con las capas de aplicación y enlace.

Para el manejo de los mensajes se utilizan seis listas, de las cuales tres son para las transmisiones y las restantes para las recepciones (Ver figura 2). A continuación se explicarán sus usos en función de la actividad de transmisión y recepción.

Las listas utilizadas para la transmisiones de mensajes se atienden con prioridad empezando con las de órdenes, de solicitudes y finalmente de alarmas.

Dado el tipo de canal de comunicación que se utiliza, sólo se puede enviar y recibir un mensaje a la vez, por lo que el procedimiento usado para enviar los mensajes considera esta posibilidad.

El algoritmo de transmisión y recepción de un mensaje es el siguiente: se busca en las listas de transmisión un mensaje en función de su prioridad, cuando se localiza uno pendiente en una lista, se toma y se le transfiere a la capa de enlace quedándose en estado de espera hasta que la capa de enlace regrese la respuesta. El mensaje de respuesta se coloca en la lista correspondiente al tipo de mensaje que se transmitió.

Para lograr la comunicación entre las capas de enlace y aplicación, se utilizan cuatro procedimientos. El primero permite realizar la transmisión de un mensaje de orden al colocarlo en la lista de órdenes; el segundo, sirve para transmitir una solicitud que es colocada en la lista de solicitudes; el tercero se usa para transmitir una solicitud del estado operativo del controlador que es colocada en la lista de alarmas y

el cuarto es un procedimiento especial para transferir los diferentes tipos de mensajes recibidos a la capa de aplicación .

CAPA DE APLICACIÓN: Una parte de la capa de aplicación la realiza el sistema de comunicación debido a limitaciones del sistema operativo MSDOS para realizar programación concurrente para lo cual se aprovechó la posibilidad de interrupción del reloj calendario de la computadora para que el sistema de comunicaciones pudiera solicitar la información de los controladores y su estado de alarma. Para lograrlo, el programa que utiliza el sistema de comunicación, hace uso de un procedimiento especial llamado *Alta*, mediante el cual le transfiere el identificador del controlador y su periodo de muestreo. Esta información se inserta en una lista circular como se muestra en la Fig 2 y con ella comienza su trabajo automáticamente el sistema de comunicación.

Existen otros dos medios de enlace que permiten la comunicación entre el programa de aplicación y el sistema de comunicaciones: cuando se desea transmitir un mensaje especial y cuando se requiere conocer las respuestas de los mensajes enviados.

Para transmitir un mensaje especial se utiliza un procedimiento llamado *Orden*, que le pasa a la capa de transporte el mensaje que se debe enviar a un controlador.

Cuando se requiere conocer las respuestas de los diferentes mensajes enviados a los controladores, el programa de aplicación utiliza un procedimiento llamado *Respuesta*, por medio del cual se le indica a la capa de transporte, el controlador, el tipo de mensaje que se requiere (orden, solicitud o alarma) y el área donde debe copiarlo.

Como el sistema de comunicaciones es un programa que debe ser instalado en memoria por el programa de aplicación, esto se realiza ejecutando el procedimiento *Inicial*. Para desinstalarlo el sistema de comunicación se ejecuta el programa *Terminar*.