

ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

OPTIMIZACION JERARQUICA DE SISTEMAS LINEALES CON FUNCION
DE COSTO CUADRATICA

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

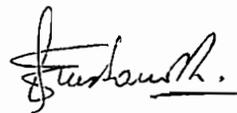
Marco Vinicio Ortiz Tirado

Mayo 1986



CERTIFICACION

Certifico que el presente trabajo
ha sido elaborado en su totalidad
por el señor Marco Vinicio Ortíz
Tirado.



Ing. Patricio Burbano
DIRECTOR DE TESIS

RECONOCIMIENTO

Este trabajo representa la síntesis del esfuerzo realizado durante mi vida en la Escuela Politécnica Nacional. Para llevarla a un feliz término tuve el apoyo decidido de innumerables personas y amigos que me alentaron a terminar una de las principales metas en mi vida.

Debo manifestar un sincero agradecimiento para el Ing. Patricio Burbano quien con su gran experiencia supo guiarme en la elaboración de esta tesis. Un reconocimiento especial al Ing. Marco Barragán, quien base a su conocimiento contribuyó para una mejor comprensión y elaboración del tema desarrollado.

Finalmente, un agradecimiento para la Universidad Central del Ecuador, y en especial al Ing. Rodrigo Tirado S., Decano de la Facultad de Ingeniería en Geología, Minas y Petróleos, quien me facilitó el uso del computador para la elaboración del programa presentado en este trabajo así como también a la Srta. María Elena Moya por su paciencia y por su magnífico trabajo en la transcripción de esta tesis.

3.1	Algoritmo de Optimización de sistemas lineales con función de costo cuadrática	59
3.2	Programa maestro y entrada de datos	64
3.3	Implementación Computacional del Algoritmo de Tamura	67
3.4	Salida de resultados	71
CAPITULO IV: RESULTADOS Y CONCLUSIONES		75
4.1	Resultados	76
4.2	Conclusiones	86
4.3	Recomendaciones	91
BIBLIOGRAFÍA		93
APENDICE A: MANUAL DE USO DEL PROGRAMA		94
APENDICE B: LISTADO DEL PROGRAMA		100
APENDICE C: CONDICIONES NECESARIAS PARA UN EXTREMO LOCAL		118

CAPITULO I

INTRODUCCION

- 1.1 INTRODUCCION
- 1.2 OPTIMIZACION DE SISTEMAS LINEALES CON FUNCIÓN
DE COSTO CUADRATICO
- 1.3 EJEMPLOS DE APLICACION
- 1.4 DISCUSION DE LA SOLUCION AL PROBLEMA DEL
REGULADOR CUADRATICO LINEAL
- 1.5 CRITERIOS GENERALES Y METODOS DE OPTIMIZACION

1.1 INTRODUCCION.- El problema de la optimización nace con el objetivo de ahorrar ciertos "costos" en funcionamiento de un sistema.

En la medida que los problemas de control involucran un mayor número de variables, la necesidad de encontrar soluciones más confiables a estos problemas condujeron a desarrollar técnicas involucradas con el control óptimo de sistemas.

El problema de optimización conduce a la necesidad de definir un "índice de funcionamiento", el mismo que representará la característica fundamental sobre la que se quiera controlar y optimizar un determinado sistema. Este "índice de funcionamiento" se lo conoce generalmente como "función objetiva" - o como "función de costo".

El problema de control óptimo tiene que ver con la optimización de una función de costo lo que se logra encontrando su mínimo valor con respecto a una determinada variable que generalmente es la del control del sistema.

En el control clásico, el controlador es diseñado en función de ciertos parámetros del sistema mediante ensayo y error. También se pueden satisfacer condiciones de invariancia, cambios en los parámetros y rechazo de perturbaciones.

El control óptimo estudia las técnicas de optimización matemática y la forma de aplicarlas al control mediante un método de síntesis.

Generalmente, la variable controlada en un sistema de control es una medida

del error del sistema. El objetivo del control óptimo tenderá a minimizar el área bajo la curva que representa la variación del error en función del tiempo. Esta condición puede expresarse por la siguiente integral:

$$J = \int_{t_0}^{t_f} e(t) dt$$

donde $e(t)$ es el error del sistema y J es la función de costo.

En la mayor parte de los casos prácticos, el comportamiento de un sistema se optimiza eligiendo el sistema de control, de manera tal que lleva a un mínimo (o un máximo) la función de costo. El control resultante puede ser lineal, - no lineal, estacionario o variable en el tiempo, y depende del índice de funcionamiento. El diseñador es el encargado de formular la función de costo basándose en los requisitos del problema, lo cual supone un conocimiento profundo del sistema que se va a controlar.

Para el presente trabajo presentaremos en un primer trabajo las técnicas centralizadas para el control óptimo de sistemas lineales. En una segunda parte realizaremos un análisis del control descentralizado para finalmente obtener - las conclusiones respectivas acerca de las ventajas y desventajas que implica la utilización del control jerárquico en el diseño de un control óptimo de un - determinado sistema.

$\underline{P}, \underline{Q}, \underline{R}$ = son matrices de ponderación

\underline{X} = estados del sistema

\underline{U} = controles del sistema

Esta forma cuadrática permite obtener ecuaciones sencillas al encontrar los mínimos o máximos de estas funciones de costo.

Utilicemos la función de costo así definida para resolver el problema del regulador cuadrático lineal, cuya configuración es la siguiente:

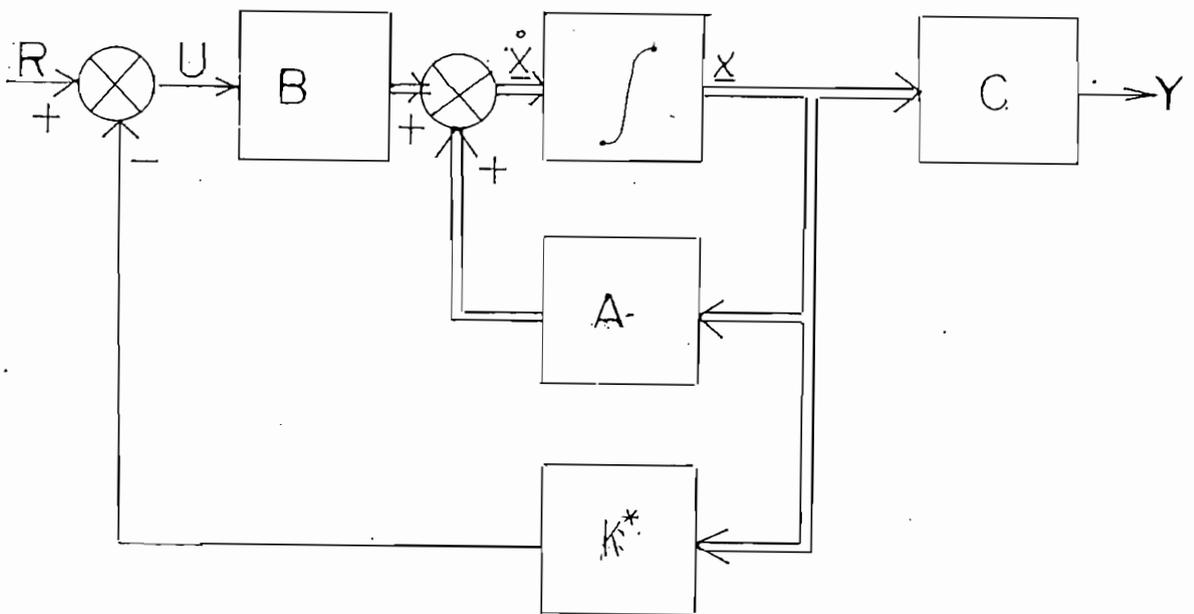


Fig. 1.1 Regulador cuadrático lineal. Diagrama de bloques.

En este caso, lo que se desea calcular es el valor óptimo de la ganancia de realimentación K^* tomando en consideración el índice de funcionamiento presentado anteriormente.

En este sistema se cumple:

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

y que es la ecuación de estado.

Recordemos que:

$$J = \frac{1}{2} \|\underline{X}(t_f)\|_{\underline{P}}^2 + \frac{1}{2} \int_{t_0}^{t_f} (\|\underline{X}(t)\|_{\underline{Q}}^2 + \|\underline{u}(t)\|_{\underline{R}}^2) dt$$

es la función de costo. Nuestro propósito es minimizar J pero tomando en cuenta que está sujeto a la restricción que representa la ecuación de estado del sistema.

El problema de control óptimo trata de encontrar los valores óptimos tanto de control como de estado que minimicen J .

Para resolver lo planteado debemos construir una función ampliada que contenga restricciones de tal forma que la minimización se realice como si se tratase de un problema que no tenga restricciones.

Construyamos esta función y que llamaremos el hamiltoniano H

$$H = \frac{1}{2} \|\underline{X}(t)\|_{\underline{Q}}^2 + \frac{1}{2} \|\underline{u}(t)\|_{\underline{R}}^2 + \underline{\lambda}^T (\underline{A} \underline{X}(t) + \underline{B} \underline{u}(t))$$

\underline{J}_x^* se define como $\underline{K}(t) \underline{X}(t)$

Si deseamos encontrar un mínimo con respecto a las variables debemos realizar lo siguiente:

$$\frac{\partial H}{\partial \underline{u}} = 0$$

$$\frac{\partial H}{\partial \underline{u}} = \underline{R} \underline{u}(t) + \underline{B}^T \underline{J}_x^*$$

$$0 = \underline{R} \underline{u}(t) + \underline{B}^T \underline{J}_x^*$$

$$\therefore \underline{u}^* = -\underline{R}^{-1} \underline{B}^T \underline{J}_x^*$$

donde debe darse que \underline{R} debe ser una matriz definida positiva.

Reemplacemos el valor de \underline{u}^* en H

$$H^* = \frac{1}{2} \left\| \underline{X}(t) \right\|_Q^2 + \frac{1}{2} \underline{u}^{*T} \underline{R} \underline{u}^* + \underline{J}_x^{*T} (\underline{A} \underline{X} + \underline{B} \underline{u}^*)$$

$$H^* = \frac{1}{2} \left\| \underline{X} \right\|_Q^2 - \frac{1}{2} \underline{J}_x^{*T} \underline{B} \underline{R}^{-1} \underline{B}^T \underline{J}_x^* + \underline{J}_x^{*T} \underline{A} \underline{X}$$

Por el valor se tiene:

$$\underline{J}^*(tf) = \frac{1}{2} \underline{X}^T(tf) \underline{P} \underline{X}(tf)$$

$$J^*(t) = \frac{1}{2} \underline{X}(t)^T \underline{K}(t) \underline{X}(t)$$

Entonces:

$$\frac{\Delta J^*}{\Delta X} = \underline{J}_X^*(t) = \underline{K}(t) \underline{X}(t)$$

$$\underline{J}(t) = \frac{1}{2} \underline{X}^T \underline{K}^{\circ}(t) \underline{X}$$

por lo tanto:

$$\underline{u}^* = -\underline{R}^{-1} \underline{B}^T \underline{K}(t) \underline{X}$$

$$\underline{u}^* = -\underline{K}^* \underline{X}$$

donde:

$$\underline{K}^* = \underline{R}^{-1} \underline{B}^T \underline{K}(t)$$

y que representará nuestra ganancia óptima.

Por lo tanto:

$$\frac{1}{2} \underline{X}^T \underline{K}^{\circ} \underline{X} + \frac{1}{2} \underline{X}^T \underline{Q} \underline{X} - \frac{1}{2} \underline{X}^T \underline{K} \underline{B} \underline{R}^{-1} \underline{B}^T \underline{K} \underline{X} + \underline{X}^T \underline{K} \underline{A} \underline{X} = 0$$

Usemos el siguiente artificio:

$$KA = \frac{1}{2} (KA + (KA)^T) + \frac{1}{2} (KA - (KA)^T)$$

que representa la suma de una matriz simétrica más una que no lo es.

Pero:

$$KA - (KA)^T = 0$$

pues es un escalar y entonces:

$$\frac{1}{2} X^T (\dot{K} + Q - KBR^{-1}B^TK + KA + A^TK) X = 0$$

Como:

$$X(t) \neq 0$$

Entonces:

$$\dot{K} + Q - KBR^{-1}B^TK + KA + A^TK = 0$$

que es una ecuación implícita de primer orden y que se conoce como la Ecuación Diferencial de Ricatti y que generalmente se representa como:

$$\dot{K} = -Q + KBR^{-1}B^TK - KA - A^TK = f \quad \text{Ecuación 1.1}$$

cuando $\dot{K} = 0$, K es una constante y se cumple que $f(K) = 0$

y que se le conoce como Ecuación Algebraica de Ricatti.

Existen muchos métodos para solucionar la ecuación de Ricatti entre los cuales mencionaremos los siguientes:

METODO DE EULER: donde se realiza la siguiente aproximación:

$$\dot{K} \approx \frac{K(t) - K(t - \Delta t)}{\Delta t} = f(K)$$

despejando $K(t)$ tenemos:

$$K(t) = f(K) \Delta t + K(t - \Delta t)$$

y si despejamos $K(t - \Delta t)$ tenemos

$$K(t - \Delta t) = K(t) - f(K) \Delta t$$

y haciendo que $K(t_f) = H$

Cuando Δt tiende a cero el método se hace más exacto. el método mejora calculando en forma iterativa el valor de K . En la solución puede ocurrir ciertos problemas de simetría y que pueden evitarse haciendo lo siguiente:

$$\underline{K}(t) = \frac{1}{2} (K(t) + K^T(t))$$

METODO DE RUNGE KUTTA: Este método es usado cuando deseamos alta precisión y se asume la siguiente configuración:

$$K(t - \Delta t) = K(t) - \frac{\Delta t}{6} (K_0 + 2K_1 + 2K_2 + K_3)$$

donde: $K_0 = f(K)$

$$K_1 = F\left(K(t) - \frac{K_0 \Delta t}{2}, t - \frac{\Delta t}{2}\right)$$

$$K_2 = F\left(K(t) - \frac{K_1 \Delta t}{2}, t - \frac{\Delta t}{2}\right)$$

$$K_3 = F(K(t) - K_2 \Delta t)$$

la precisión que se logra con este método es del orden de Δt^5 .

Si: $\left| \begin{array}{cc} K_1 & - K_2 \\ K_0 & - K_1 \end{array} \right| >> 0.01$

debe disminuirse el Δt escogido.

Para reducir los errores de redondeo se recurre a la regla de Gril que dice:

$$K(t - \Delta t) = K(t) - \frac{\Delta t}{6} (K_0 + (2 - \sqrt{2}) K_1 + (2 + \sqrt{2}) K_2 + K_3)$$

donde:

K_0 y K_1 tienen la misma forma descrita anteriormente.

$$K_2 = F\left(K(t) - (\sqrt{2} - 1) \frac{K_0 \Delta t}{2} - (2 - \sqrt{2}) \frac{K_1 \Delta t}{2}, t - \frac{\Delta t}{2}\right)$$

$$K_3 = F\left(K(t) + \frac{\sqrt{2}}{2} K_1 \Delta t - (1 + \frac{\sqrt{2}}{2}) K_2 \Delta t, t - \Delta t\right)$$

1.3 EJEMPLOS DE APLICACION.-

Ejemplo 1

Sea el siguiente regulador de tiempo descrito por:

$$\dot{X}_1 = -X_1 + U_1$$

y sea:

$$J = \int_0^{\infty} (X_1^2 + u_1^2) dt$$

podemos observar que $A=1, B=1, Q=1, R=1$ de tal forma que la ecuación de Ricatti toma la forma de:

$$0 = -2K + K^2 - 1 = 0$$

donde:

$$K = \frac{2 \pm \sqrt{4+4}}{2} = \frac{2 \pm 2\sqrt{2}}{2} = 1 \pm \sqrt{2}$$

si K se desea positivo entonces $K = 2.41$ y el control óptimo se hace

$$u_1(t) = -2.4 X_1(t)$$

lo que hace que:

$$\dot{X}_1 = -1.4 X_1(t)$$

cuya solución es:

$$X_1(t) = e^{-1.4 t} X_1(0) + \text{constante}$$

Ejemplo 2

Sea el sistema de la Figura 1.2

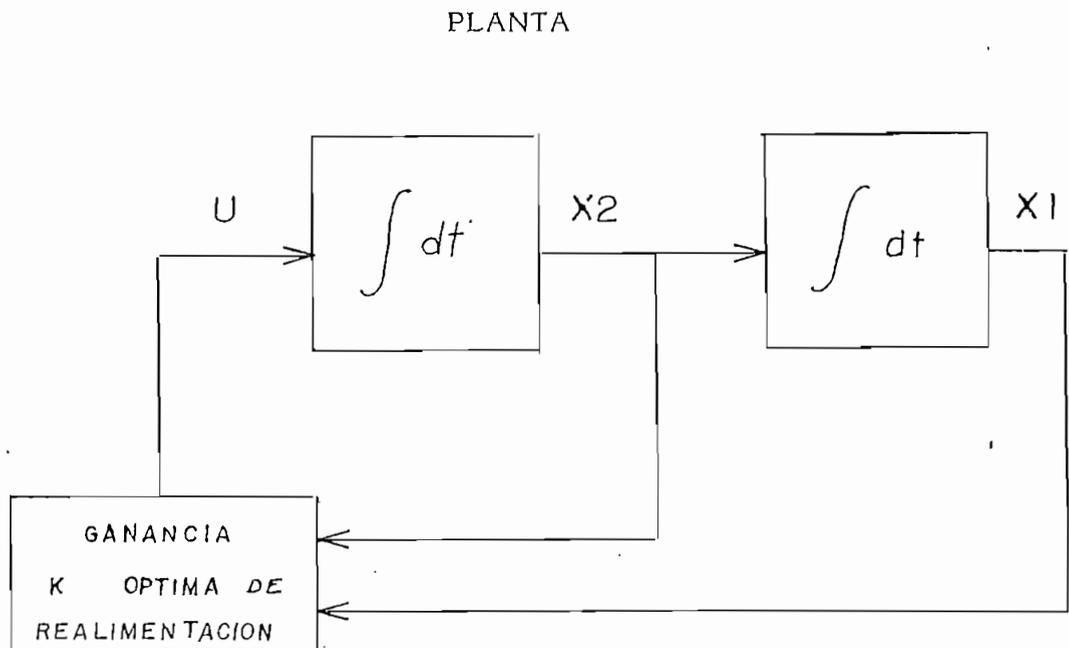


Fig. 1.2 Realimentación óptima de estado.

Si se desea encontrar K de forma que minimice a

$$J = \int_0^{\infty} (\|X\|_Q^2 + \|u\|_R^2) dt$$

donde:

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & u \end{pmatrix} \quad \text{con } u \geq 0$$

y lo que debemos encontrar es:

$$u = -K X$$

La ecuación de la planta es:

$$\begin{pmatrix} \dot{X}_1 \\ \dot{X}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

usando la ecuación 1.1 de Ricatti para cuando $f(K) = 0$

se llega a la siguiente ecuación:

$$A^T P + P A + P B R^{-1} B^T P + Q = 0$$

y realizando todas las simplificaciones se obtiene:

$$1 - p_{12}^2 = 0$$

$$p_{11} - p_{12} p_{22} = 0$$

$$u + 2 p_{12} - p_{22}^2 = 0$$

y haciendo que P sea positiva definida:

$$P = \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} = \begin{pmatrix} \sqrt{u+2} & 1 \\ 1 & \sqrt{u+2} \end{pmatrix}$$

entonces;

$$\begin{aligned}
 K &= R^{-1} B^T P \\
 &= (1) (0 \quad 1) \begin{pmatrix} \sqrt{u+2} & 1 \\ 1 & \sqrt{u+2} \end{pmatrix} \\
 &= \begin{pmatrix} 1 & \sqrt{u+1} \end{pmatrix}
 \end{aligned}$$

de donde u óptimo = u^* es:

$$u^* = -KX = -X_1 - (u+2)^{1/2} X_2$$

1.4 DISCUSION DE LA SOLUCION AL PROBLEMA DEL REGULADOR CUADRÁTICO LINEAL.-

El problema de la optimización sugiere el uso de un programa de computador que calcule la forma iterativa los valores óptimos del control y del estado de tal forma que minimice la función de costo.

En nuestro problema consideramos función de costo cuadrática. La programación con funciones cuadráticas es un caso particular de la programación no lineal. De alguna forma el problema de la programación cuadrática se reduce en un problema de programación lineal lo que simplifica el método a ser utilizado. Debido a este circunstancia los problemas de programación son considerados como una clase aparte.

Los problemas computacionales cuando se usa programación no lineal son más complicados. Desafortunadamente para la programación cuadrática y lo mismo para la programación lineal no existe un método que garantice la solución para un número finito de iteraciones. Se han desarrollado algoritmos aislados y al-

gunos de ellos utilizan las técnicas de aproximación mediante el gradiente de la función en la que se está trabajando. A continuación detallamos esta técnica.

Consideremos el siguiente problema:

Tratemos de maximizar la siguiente función f

$$\max (f (\underline{X})) / \underline{AX} < \underline{b} : \underline{X}_i \geq 0 \quad i = 1, \dots, n)$$

donde:

\underline{X} = vector de dimensión n

\underline{A} = una matriz constante A

\underline{b} = un vector constante de dimensión m

$f(\underline{x})$ = función de costo no lineal de \underline{X} variables

Presentamos un ejemplo de dos dimensiones para ilustrar el problema. La región sombreada de la figura 1.3 presenta la región donde los valores de \underline{X} satisfacen las condiciones del problema. Las curvas desde F_0 hasta F_5 representan el lugar de los valores constantes de la función de costo en un orden ascendente: $F_0 < F_1 < F_2 < F_3 < F_4 < F_5$. Los puntos E_1, E_2, E_3, E_4 y e_5 son los puntos extremos de la región permisible. De la figura se puede ver que E_3 representa la solución más óptima y que condice al valor máximo de $f(\underline{X})$. Por lo tanto:

$$f_{\max} (\underline{X}) = F_4$$

Mientras se satisfacen las condiciones, la pregunta es: ¿Cómo implementar un

algoritmo que conduzca a esta solución y sabiendo que $f(\underline{X})$ no es lineal?.

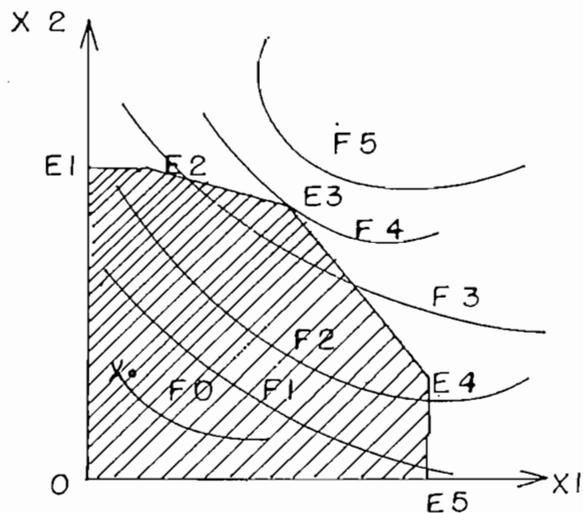


Fig. 1.3 Variación de la función f dentro de la región permisible.

Empecemos con el punto X_0 que pertenece a la región permisible. Nuestro problema es avanzar desde X_0 en la dirección del incremento de $f(\underline{X})$. Aún más necesitamos realizar este avance en la forma más rápida posible. El gradiente de $f(\underline{X})$.

$$\nabla f(\underline{X}) = \left(\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \dots \quad \frac{\partial f}{\partial x_n} \right)$$

es un vector que señala la dirección más rápida de incremento en el valor de esta función. La trayectoria debería avanzar desde el punto inicial de X_0 en la dirección del gradiente. Por lo tanto ya que $f(\underline{X})$ es no lineal, la dirección de su gradiente cambiará a lo largo de la trayectoria. Por lo tanto la dirección de este avance debe ser revisado de tiempo en tiempo.

Uno de los problemas cruciales en la aproximación mediante gradiente es el escoger el tamaño del pase de avance. Existen varias maneras de establecer el tamaño de este paso. Como se señaló anteriormente el tamaño es dependiente del gradiente pero no es necesariamente colineal con esta función.

Veamos una manera de definir el tamaño del paso:

Introduzcamos una notación apropiada que nos permita graficar esta explicación:

Sea: X_i = un punto situado en la región permisible después del paso i
 r_i = vector unitario en la dirección del paso a partir de X_i

Como vector unitario:

$$r_i^T r_i = 1$$

d_i = un escalar cuyo valor es la longitud del paso tomando desde X_i en la dirección de r_i .

Por lo tanto:

$$X_{i+1} = X_i + d_i r_i$$

la Dependencia de la nueva dirección de r_i viene dada por:

$$r_i = H_i \nabla f_i$$

H_i es una matriz $n \times n$ llamada matriz directriz o métrica.

Para especificar H_i existen varias técnicas pero cuando:

$$H_i = \frac{1}{\|\nabla f_i\|}$$

a dirección del nuevo paso es colineal con ∇f_i desde X_i . Esta técnica es denominada el método más rápido.

De acuerdo con esto:

$$f(X_{i+1}) = f(X_i + d_i \cdot \nabla f_i)$$

Para maximizar la función debe cumplirse que:

$$\frac{df(X_{i+1})}{d(d_i)} = \nabla f_i^T \cdot \nabla f_{i+1} = 0$$

debe asegurarse que X_{i+1} se encuentre dentro de la región permisible. La última ecuación implica que los dos puntos sucesivos son ortogonales, lo que quiere decir que la trayectoria de la solución óptima se da en pasos zigzagueantes que forma un ángulo recto entre sí.

En la mayoría de los casos esta técnica del ascenso o descenso más pronunciado son llamadas tácticas del gradiente de paso pequeño y son muy lentos en su convergencia.

d_i es la longitud del paso y el algoritmo de cálculo será dado en el capítulo III en todo caso será un valor que se encontrará entre 0 y 1 como límites inferior y superior respectivamente. Para unos casos el valor del paso será variable, haciendo que se llegue a obtener un óptimo a medida que la convergencia se vaya produciendo.

1.5 CRITERIOS GENERALES Y METODOS DE OPTIMIZACION.-

En la mayoría de los casos el índice de funcionamiento de un determinado sistema depende en un gran número de decisiones y en realidad el objetivo nuestro será reconocer como tomar el mejor grupo de decisiones que nos permita solucionar los problemas.

Por lo tanto, nosotros requerimos de lo siguiente:

1. Una función de costo o índice de funcionamiento que nos permita cuantificar los efectos de cualquier decisión.
2. Un modelo que nos permita predecir los efectos de cualquier decisión.
3. Un conocimiento de todos los factores del medio tanto pasados, presentes como futuros.

Se han desarrollado algunos métodos para resolver el problema de la optimización, es decir, determinar los controles que satisfagan el conjunto de puntos - que optimiza a la función de costo y que satisface además las restricciones.

Cuando se da la formulación matemática del problema, las técnicas usadas - para la solución vienen de teorías clásicas y pueden ser clasificadas dentro - de las tres categorías siguientes:

- a) La programación matemática
- b) La programación dinámica
- c) Cálculo variacional: principio del máximo y cálculo variacional clásico.

Por otro lado, cuando no se da una modelación matemática nos vemos obligados a usar métodos directos de optimización.

A continuación se presenta un resumen esquemático del método correspondiente a cada formulación del problema:

<u>Formulación del problema</u>	<u>Método</u>
1. Procesos estáticos	Teoría ordinaria sobre máximos y mínimos
1.1 Problemas sin restricciones	
$\max_{\underline{X} \text{ vector}} J = J(\underline{X})$	
1.2 Problemas con restricciones de igualdad	Método de Lagrangiano
$\max J = J(\underline{X}) \text{ sujeto a}$ $g(\underline{X}) = 0$	
1.3 Problemas con restricciones de no igualdad	Método de Kun - Tucker
$\max J = J(\underline{X}) \text{ sujeto a}$ $\underline{h}(\underline{X}) \leq 0$	
1.4 Problemas lineales	Método Simplificado
$\max J =$	

$$i = 1, \dots, N$$

1.5 Problemas generales no lineales

$$\max J = J(X) \text{ sujeto a } g_i^{(X)} (\leq = \geq) b_j$$

$$j = 1, \dots, m$$

Programación no lineal

Teoria de Kun - Tucker en particular

2. Sistemas dinámicos

2.1 Problemas lineales

$$J = J(X, u, t) \text{ sujeto a}$$

$$\dot{X} = A(t)X - B(t)u \\ = 0$$

- Métodos variacionales

- Programación dinámica

- Método del gradiente

2.2 Problemas no lineales

$$J = J(X, u, t) \text{ sujeto a}$$

$$\dot{X} = g(X, u, t) = 0$$

J, g no lineales

Cuasilinealización

3. Sistemas dinámicos o estáticos complejos

Método de descomposición y coordinación vistos en programación matemática y cálculo variacional, control jerárquico y en optimización.

Para escoger un determinado método debemos realizar las siguientes considerau

ciones:

1. ¿Es un problema de sistemas dinámicos o constantes?.
2. ¿Es un problema de naturaleza lineal o no lineal?.
3. ¿Es un caso monovariable o multivariable?.
4. ¿Existen restricciones?.

El método que escojamos para la resolución de tales problemas requerirá dos clases de preguntas adicionales:

a) Sobre el nivel matemático

1. ¿Existe solución?.
2. ¿Existe una única solución?.
3. ¿Son necesarias condiciones de optimalidad?.
4. ¿Existen suficientes condiciones de optimalidad?.
5. ¿Cuál es la naturaleza de un extremo?.

b) Sobre el nivel computacional:

1. ¿Existe una solución numérica al respecto?.
2. ¿Se definió el computador para el trabajo?.
3. ¿Si es que se usa un método iterativo, existe convergencia?.
4. ¿Cuál es el tiempo de cálculo?.

En la práctica se deben realizar todas estas formulaciones pues cada una de ellas tiene su importancia dentro del cálculo computacional.

Para resumir lo anteriormente descrito realizaremos el siguiente diagrama de

En el siguiente capítulo se analiza en forma completa una de las técnicas de optimización jerárquica.

flujo que define los diferentes estados en la formulación y solución de problemas de optimización.

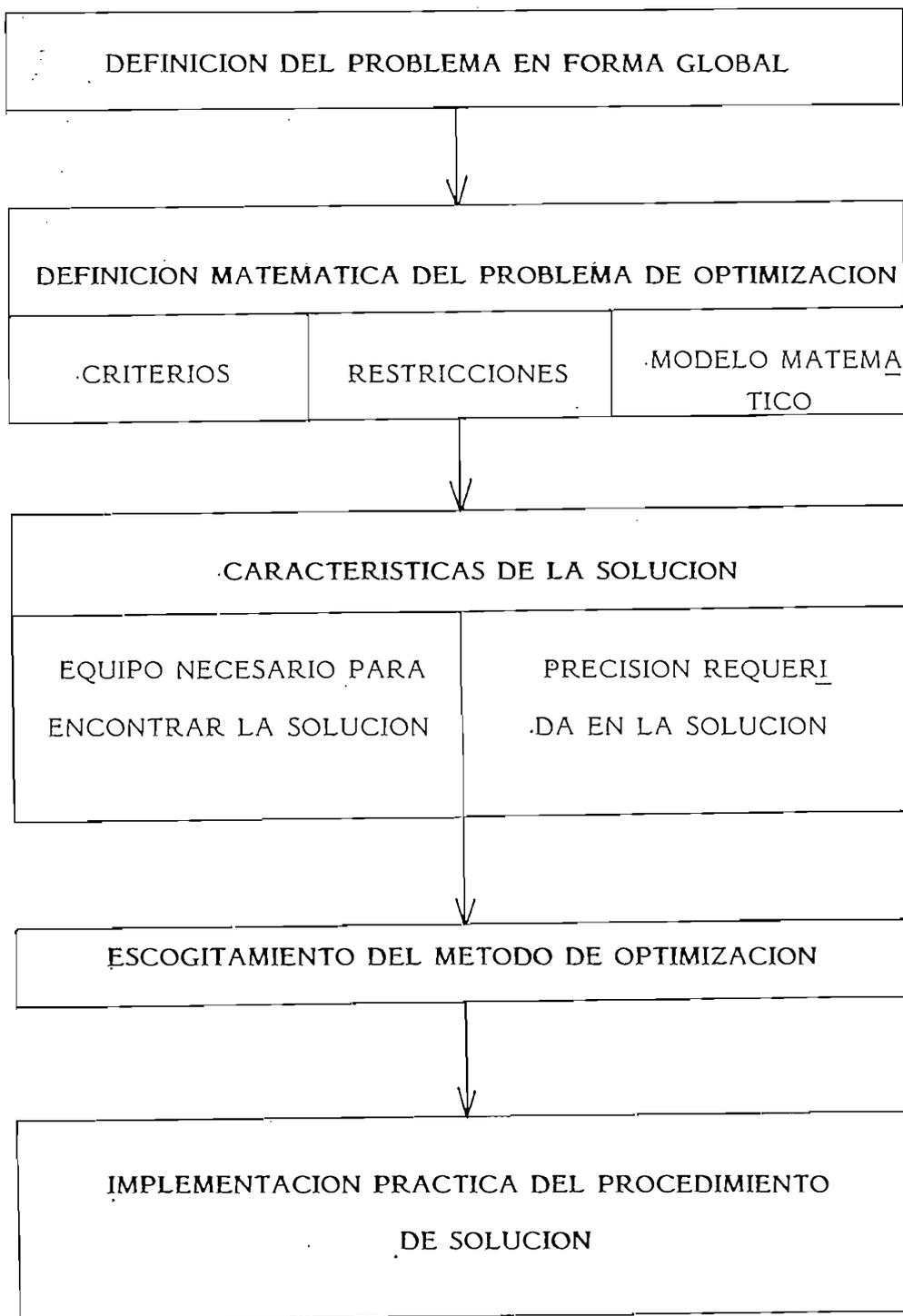


Fig. 1.4 Diagrama de flujo para la formulación y solución de los problemas de optimización.

CAPITULO II

OPTIMIZACION JERARQUICA

- 2.1 INTRODUCCION
- 2.2 METODO GENERAL DE COORDINACION
- 2.3 METODO DE TAMURA
- 2.4 OPTIMIZACION USANDO LA DESCOMPOSICION
DE SUBSISTEMAS CON EL CRITERIO DE
TAMURA



2.1 INTRODUCCION.- Como pudimos observar en el anterior capítulo, los sistemas dinámicos se encuentran descritos como un conjunto de ecuaciones diferenciales multidimensionales lineales o no lineales de primer orden.

La solución a estas ecuaciones son múltiples y uno de los métodos descritos es la solución por medio de las ecuaciones de Ricatti.

Generalmente, los modelos de sistemas reales a menudo comprenden ecuaciones diferenciales de un alto orden, suponiendo además que estas ecuaciones involucran ciertos retardos propios del sistema.

Las técnicas de solución que podemos encontrar y que se encuentran mejor descritas en el curso de control óptimo incluyen: La Programación Dinámica (DP), las técnicas de aproximaciones sucesivas, el cálculo de variaciones, etc.

Las ecuaciones de Ricatti son ecuaciones diferenciales (para sistemas continuos) o ecuaciones de diferencias (para sistemas discretos) no lineales con un número de elementos $\frac{n(n+1)}{2}$ para un sistema cuyo orden podemos denominar como n .

La desventaja principal tanto de las ecuaciones de Ricatti como de la Programación Dinámica para resolver los problemas de Control Óptimo de sistemas lineales multivariables radica en la gran cantidad de memoria de computador que es necesario emplear. El incremento de esta memoria está en función directa del orden del sistema. Como una desventaja adicional se contempla la acumulación de errores debido al error de redondeo que se produce en cada -

iteración. Esta última desventaja puede dar lugar a que se produzca una inestabilidad en el cálculo numérico.

A continuación detallamos el método de aproximación jerárquica para la resolución de estos problemas.

Formulemos inicialmente cual es el problema que vamos a resolver, tanto con el problema de optimización como el de control de sistemas interconectados - dinámicamente. Se trata de descomponer un sistema dinámico en N Subsistemas que interaccionan entre sí, y cuya iteración se representa mediante nuevas entradas que representan la influencia de los otros estados sobre un subsistema en particular.

Consideremos que el problema se halla conformado por N Subsistemas conectados, por ejemplo, como indica la figura 2.1

Para cualquier subsistema i X_i es el vector de dimensión N_i . U_i es un vector de Control de dimensión M_i . Z_i es el vector de entradas de dimensión r_i y que será generado por los estados de los otros subsistemas y representa la influencia de los otros estados.

Asumamos entonces que los sistemas por sí mismos pueden ser descritos por ecuaciones diferenciales lineales, es decir:

$$\dot{\underline{X}}_i(t) = \underline{A}_i \underline{X}_i(t) + \underline{B}_i \underline{u}_i(t) + \underline{C}_i \underline{Z}_i(t) \quad \text{Ecs. 2.1}$$

donde: $\underline{X}_i(0) = \underline{X}_{i,0}$ es decir tenemos como dato $\underline{X}_{i,0}$

Asumamos también que el vector de entradas \underline{Z}_i es una combinación lineal de los estados de los demás subsistemas. Es decir, \underline{Z}_i tendrá la siguiente representación:

$$\underline{Z}_i = \sum_{j=1}^N L_{ij} \underline{X}_j \quad \text{Esc. 2.2}$$

donde: L_{ij} es la ponderación de la influencia del estado X_j sobre el nuevo estado Z_i .

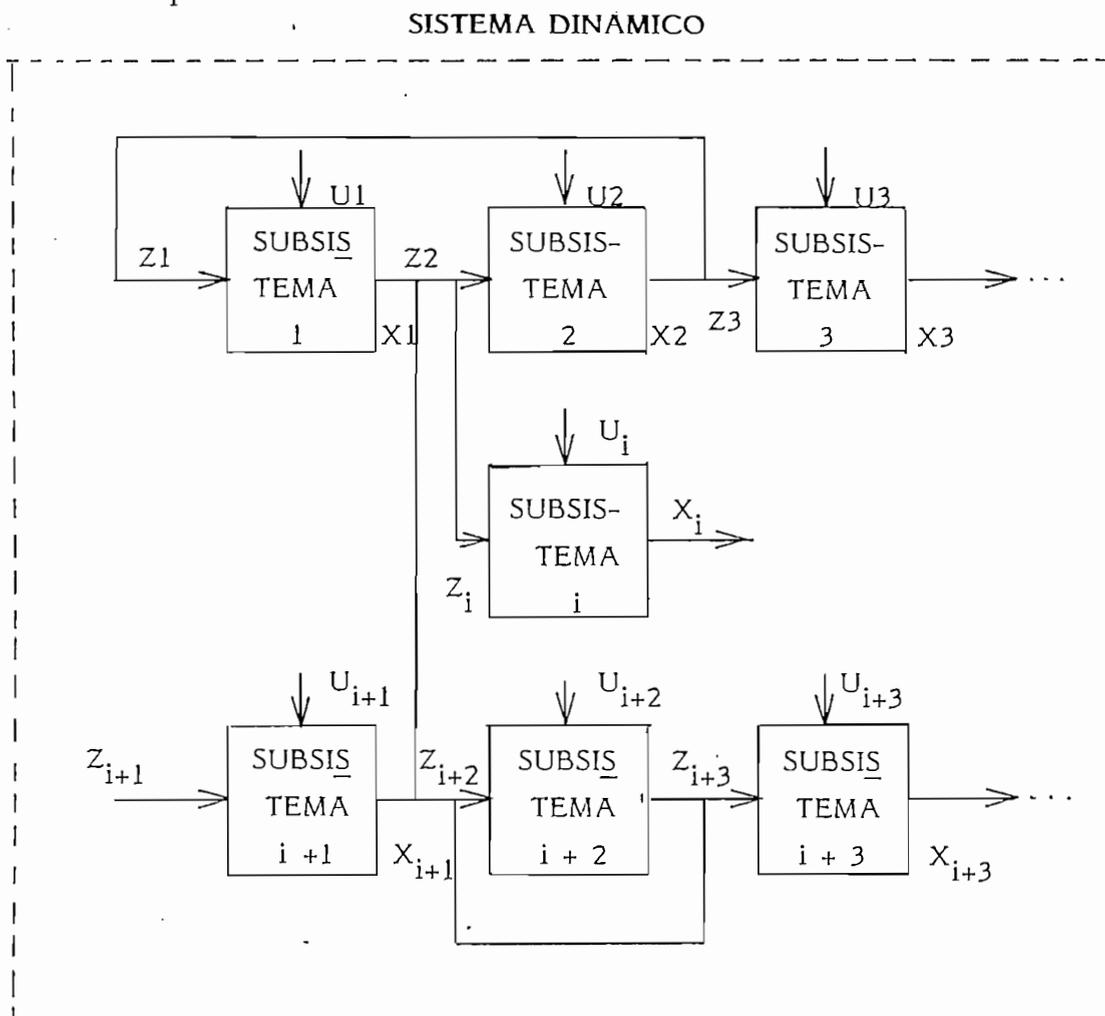


Fig. 2.1 Ejemplo de un sistema interconectado dinámicamente.

Para nuestros propósitos debemos definir adecuadamente la función de costo del sistema y coherencia con la descomposición del sistema en N Subsistemas. Esta función de costo como lo mencionamos en el anterior capítulo representa el índice de funcionamiento de dicho sistema y en realidad el problema de control óptimo será encontrar los controles óptimos de cada subsistema, que produzcan los estados óptimos y las trayectorias óptimas con el propósito de minimizar la función de costo que tendrá la siguiente forma:

$$J = \sum_{i=1}^N \frac{1}{2} \|\underline{X}_i(tf)\|_{\underline{P}_i}^2 + \int_0^{tf} \left(\frac{1}{2} \|\underline{X}_i(t)\|_{\underline{Q}_i}^2 + \frac{1}{2} \|\underline{U}_i(t)\|_{\underline{R}_i}^2 + \frac{1}{2} \|\underline{Z}_i(t)\|_{\underline{S}_i}^2 \right) dt$$

que en realidad es la forma cuadrática y describe el funcionamiento de un sistema lineal formado por N Subsistemas interconectados en forma dinámica. Este sistema además es continuo:

Esta función de costo tiene las siguientes restricciones:

$$1) \quad \dot{\underline{X}}_i(t) = \underline{A}_i \underline{X}_i(t) + \underline{B}_i \underline{U}_i(t) + \underline{C}_i \underline{Z}_i(t)$$

$$2) \quad \underline{Z}_i = \sum_{j=1}^N L_{ij} \underline{X}_j$$

Debemos indicar además que \underline{Q}_i y \underline{P}_i son matrices semidefinidas y además

R_i y S_i son matrices definidas positivas.

Las matrices Q_i están definidas semipositivamente cuando la forma cuadrática F asociada: $F = V^T Q V \geq 0$ donde V es vector de dimensiones compatibles al producto. De manera similar la matriz R_i está definida positivamente si $F > 0$.

Nosotros usaremos la siguiente notación en el presente trabajo:

$$\|X\|_L^2 = X^T L X$$

que es una forma cuadrática.

Si la relación de interconexión dada por la ecuación 2.2 es substituida en la ecuación 2.1 es posible encontrar una descripción global de la forma comúnmente conocida para cualquier sistema. Es decir:

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

donde:

$$\underline{X} = \begin{pmatrix} \underline{X}_1 \\ \underline{X}_2 \\ \vdots \\ \underline{X}_n \end{pmatrix} \quad \text{y} \quad \underline{u} = \begin{pmatrix} \underline{u}_1 \\ \underline{u}_2 \\ \vdots \\ \underline{u}_n \end{pmatrix}$$

Para el caso que estamos analizando es difícil dar una interpretación física al término cuadrático $\|z_i\|_{S_i}^2$ ya que siempre puede ser combinado dentro del término $\|X\|_Q^2$ para el caso global. Sin embargo, es imperiosa la ne-

cesidad de usar este término, ya que este "pseudo", control \underline{Z}_i , al constituir el Hamiltoniano para nuestra solución, permite descomponer el problema de optimización del sistema en N Subsistemas de optimización para cada subsistema, según se analizará en el método de General de Coordinación en el siguiente numeral.

Dado a que va reducirse el problema global de optimización a N problemas de optimización a N subsistemas conviene analizar con más detalle lo que es un subsistema.

La definición de un subsistema, lo que conduce a la definición de un subproblema de optimización puede ser hecha en base al conocimiento físico del sistema o en general en base a una conveniente descomposición analítica (matemática) del sistema, conveniente desde el punto de vista de su solución.

Puesto que el objetivo de esta tesis no es el estudio de modelos descentralizados vamos a realizar la descomposición en forma analítica.

Podemos asumir que en nuestro estudio, los sistemas se hallan formados por N subsistemas que tienen la siguiente representación:

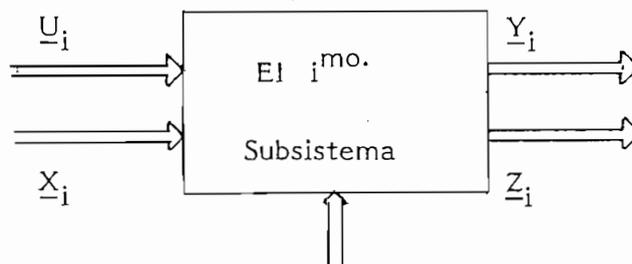


Fig. 2.2 El i^{mo} subsistema.

U_i, X_i, M_i, Y_i, Z_i son vectores que tienen respectivamente:

$MU_i, MX_i, MM_i, MY_i, MZ_i$ componentes.

Estos vectores representan lo siguiente:

U_i Entradas tanto al sistema global como al i^{mo} subsistema (perturbaciones o ruido sobre el que no tenemos ningún control).

X_i Entradas intermedias que provienen de otros subsistemas.

M_i Variables de control para el i^{mo} subsistema.

Y_i Salidas, tanto para los subsistemas como para el sistema en forma global. Podrían por ejemplo representar el producto final de un sistema de producción.

Z_i Salidas para el subsistema i y que representa la alimentación para otros subsistemas. Para el caso de un sistema de producción representarían los productos intermedios.

Para un vector global de entradas, U , el subsistema está completamente descrito en el estado estable por las siguientes ecuaciones vectoriales:

$$\underline{Z}_i = \underline{T}_i (\underline{M}_i, \underline{X}_i)$$

$$\underline{Y}_i = \underline{S}_i (\underline{M}_i, \underline{X}_i)$$

T_i, S_i son respectivamente de dimensión M_{zi} y M_{yi} .

Podemos interpretar la interconexión como:

$$X_i = \sum_{j=1}^N L_{ij} \cdot Z_j$$

$$V_i = 1 \text{ hasta } N$$

Y es aquí justamente que estamos asumiendo la descomposición en N Sub-sistemas. L_{ij} es la matriz de interconexión con M_{xi} filas y M_{zi} en - columnas y provee un acoplamiento lineal entre los subsistemas.

Lo que en realidad estamos alcanzando es hacer que la función de costo u ob- jetiva del sistema sea una función aditivamente separable de la siguiente for- ma:

$$J = \sum_i J_i (M_{ij}, X_j)$$

Analicemos el método general de coordinación para el caso del control jerár- quico o descentralizado para poder realizar un estudio más detenido de la des- composición de problema de control óptimo en N subproblemas de control - con cada índice J_i de funcionamiento. Debe recalcar que en lo sucesivo, que es objeto y esencia de la presente tesis, no se trabajará con la modela- ción de sistemas descentralizados (descomposición de un sistema en N subsis- temas) sino que se analizará la parte del control solamente que implica la des

composición de un problema de optimización en N subproblemas de optimización además debe considerarse que no se usarán las entradas de control M_i puesto que se restringe el trabajo de tesis para métodos de lazo abierto y finalmente por limitaciones tecnológicas y por escapar del nivel de esta tesis - este problema será abordado con el uso de un solo computador y no bajo el criterio de control computacional distribuido.

2.2 METODO GENERAL DE COORDINACION.-

Nuestro propósito fundamental es en realidad tratar de minimizar J. Mediante diversas técnicas, esta minimización puede efectuarse realizando las diferentes etapas que para ello amerita.

Sobre esta base, el método general de coordinación sostiene que un problema en general de minimización puede convertirse en un problema de maximización simple, en una estructura de dos niveles más sencillos todavía de ser resueltos.

Parece extraño que esto pueda acontecer pero en realidad es posible esta afirmación.

En control óptimo se utiliza la ayuda de los multiplicadores de Lagrange para resolver nuestro problema de minimización. Un resumen de estos conceptos se da en el apéndice "C". Usemos estos mismos conceptos para construir una función "dual" la misma que será la que vamos a maximizar con respecto a algún parámetro que en realidad equivaldría a minimizar nuestra función de costo.

Definiendo nuestra función de costo como:

$$J = \sum_{i=1}^N \left[\frac{1}{2} \| X_i(T_f) \|^2_{P_i} + \frac{1}{2} \int_0^{T_f} (\| X(t) \|^2_{Q_i} + \| u_i(t) \|^2_{R_i} + \| Z_i(t) \|^2_{S_i}) dt \right]$$

La función Dual $\phi(\lambda)$ será :

$$\Phi(\lambda) = \underset{x, u, z}{\text{Min}} (L(x, u, z, \lambda)) \quad \text{Ecs. 2.2}$$

Donde L es el Lagrangiano que se forma usando los multiplicadores de Lagrange de la forma siguiente:

$$\text{Partiendo de } J = f(x(T), x(t), u(t), z(t), t)$$

sujeta a las siguiente restricciones:

$$1. \quad \dot{X}(t) = A X(t) + B u(t) + C z(t)$$

$$2. \quad Z_i = \sum_{j=1}^N L_{ij} X_j$$

Introduciendo las restricciones dentro de J tenemos:

$$L(x, U, Z, \lambda) = \sum_{i=1}^N \left\{ \frac{1}{2} \|X(T)\|_{P_i}^2 + \int_0^T \left[\frac{1}{2} (\|X_i\|_{Q_i}^2 + \|U_i\|_{R_i}^2 + \|Z_i\|_{S_i}^2) + \lambda_i^T (Z_i - \sum_{j=1}^N L_{ij} X_j) \right] dt \right\}$$

Donde: $\underline{\lambda}_i$: es vector de dimensión r de los multiplicadores de Lagrange.

L : Lagrangiano que ha sido formado usando los multiplicadores de Lagrange

La dualidad de Lagrange asegura para casos como el presente que todas las constantes utilizadas son lineales y además que las funciones de costo son cuadráticas es decir el problema es convergente y podemos afirmar que:

$$\max_{\underline{\lambda}} \phi(\underline{\lambda}) = \min_{\underline{u}} J$$

Es decir, una forma equivalente de resolver el problema de minimizar J sujeto a las restricciones antes anotadas es maximizar $\phi(\underline{\lambda})$ con respecto a $\underline{\lambda}$.

Este problema podemos resolverlo dentro de una estructura de dos niveles de la siguiente forma:

$$\text{para una } \underline{\lambda} = \underline{\lambda}^* \text{ dada:}$$

tenemos que:

$$L(x, u, z, \lambda) = \sum_{i=1}^N \left[\frac{1}{2} \|X(t_f)\|_p^2 + \int_0^{t_f} \left(\frac{1}{2} (X_Q^2 + U_R^2 + Z_S^2) + \lambda_i^T z_i - \lambda_{ij}^T L_{ij} X_j \right) dt \right] = \sum_{i=1}^N L_i$$

lo que quiere decir que el Lagrangiano es una función lineal aditivamente separable y por lo tanto puede ser descompuesta dentro de N independientes sublagrangianos, uno para cada subsistema.

Por lo tanto para un $\lambda = \lambda^*$ dada que puede ser tratado como una trayectoria conocida, es posible la minimización del Sublangrangiano:

$$L = 1/2 \|X_i(t_f)\|_P^2 + \int_0^{t_f} \left[1/2 \|X_i(t)\|_Q^2 + \|U_i(t)\|_R^2 + \|Z_i(t)\|_S^2 + kZ_i - \sum_{j=1}^N k_{ij} L_{ij} X_j \right] dt$$

en forma totalmente independiente para los N Subsistemas y en donde la minimización de cada subsistema se encuentra sujeto a las restricciones dinámicas dadas por las ecuaciones diferenciales que definen el comportamiento del sistema.

Esto nos permite a nosotros obtener $\phi^*(\lambda)$ de la ecuación 2.2

$\phi(\lambda^*)$ Puede ser mejorado sucesivamente mediante un intercambio iterativo de información con el segundo nivel lo cual mejora $\phi^*(\lambda)$ usando las N independientes minimizaciones del primer nivel.

En el capítulo anterior hicimos una introducción respecto al método de optimización que vamos a usar. Este método era el de Gradiente. Para nuestro caso es fácil obtener una expresión simple de lo que será el Gradiente en término de las soluciones para el primer nivel. El gradiente viene dado en términos del error en la relación de interconexión.

$$\nabla \phi(\lambda) = \begin{pmatrix} z_i - \sum_{j=1}^N L_{ij} X_j \\ \vdots \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ e_i \\ \vdots \\ \vdots \end{pmatrix} = \underline{e}$$

Y si usamos estos valores para verificar la convergencia del proceso iterativo podemos concebir la siguiente estructura jerárquica en dos niveles como mues

tra la figura 2.3

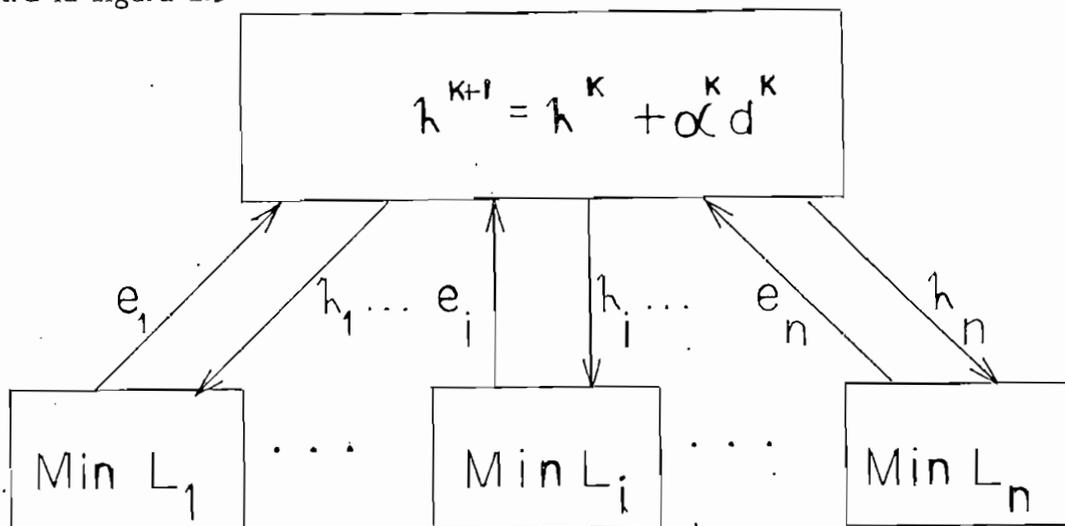


Fig. 2.3 Estructura de la optimización jerárquica en dos niveles.

Sobre el nivel 1 para $\underline{h} = \underline{h}^*$ dado, que se obtiene del segundo nivel, L_i se minimiza tomando en cuenta todas las restricciones, luego que se obtienen los estados X_i y los controles U_i de cada subsistema se los compara y si es necesario se los substituye para formar el vector de errores. Si se cumple con algún criterio de error preestablecido el proceso termina, caso contrario se regresa al nivel dos para actualizar el valor de \underline{h} con alguna técnica del gradiente y nuevamente alimentar el nivel 1 hasta cumplir con el criterio anteriormente expuesto.

Para actualizar \underline{h} se pueden usar diferentes métodos, como por ejemplo, el método del gradiente

$$\underline{h}(t) = \underline{h}(t) + \alpha \underline{d}(t) \quad 0 \leq t \leq t_f$$

donde:

α = longitud o tamaño del paso

d^k = dirección de búsqueda

Cuando se usa el método del descenso más pronunciado

$$d(t) = e(t) \quad \forall t \quad / \quad 0 \leq t \leq t_f$$

Si usamos el método del gradiente conjugado tenemos:

$$d(t) = e(t) + \beta \cdot d(t)$$

donde:

$$\beta^{k+1} = \frac{\int_0^{t_f} (\underline{e}^{k+1} \cdot \underline{e}^{k+1}) dt}{\int_0^{t_f} (\underline{e}^k \cdot \underline{e}^k) dt}$$

$$\underline{d}^0 = \underline{e}^0$$

El criterio de error usado dice que:

$$\underline{e}(t) \approx \underline{0}$$

Este método podríamos denominarlo como el de Coordinación de objetivos ya que existen objetivos plenamente definidos para descentralización. Esto es para el nivel inferior el de minimización para cada subproblema independiente - que está coordinado por el nivel superior y entre los cuales existe intercambio de información según se muestra en el esquema de la figura 2.3

Ejemplo de Optimización usando el método general de Coordinación.-

Consideremos el problema más simple de minimizar J .

$$J = \int_0^T (x_1^2 + x_2^2 + u_1^2 + u_2^2) dt$$

Con las siguientes restricciones:

$$\dot{x}_1 = x_1 - 2x_2 + u_1$$

$$\dot{x}_2 = -x_1 + x_2 + u_2$$

Sea: $z_1 = -2x_2$

$$z_2 = x_1$$

Entonces:

$$\dot{x}_1 = x_1 + u_1 + z_1$$

$$\dot{x}_2 = x_2 + u_2 - z_2$$

Construyamos el Lagrangiano como:

$$L = \int_0^T \left[x_1^2 + x_2^2 + u_1^2 + u_2^2 + z_1^2 + z_2^2 + p_1 (-\dot{x}_1 + x_1 + z_1 + u_1) + p_2 (-\dot{x}_2 + x_2 + u_2 - z_2) + \lambda_1 (z_1 + 2x_2) + \lambda_2 (z_2 - x_1) \right] dt$$

Pero puede el Lagrangiano separarse en:

$L = L_1 + L_2$ para λ_1, λ_2 dados donde:

$$L_1 = \int_0^T \left[x_1^2 + u_1^2 + z_1^2 + p_1 (-x_1 + x_1 + z_1 + u_1) + \lambda_1 z_1 - \lambda_2 x_1 \right] dt$$

$$L_2 = \int_0^T \left[x_2^2 + u_2^2 + z_2^2 + p_2 (-x_2 + x_2 - z_2 + u_2) + \lambda_2 z_2 + \lambda_1 x_2 \right] dt$$

Construyamos el Hamiltoniano para el primer subsistema:

$$H = X_1^2 + U_1^2 + Z_1^2 + \lambda_1 Z_1 - \lambda_2 X_1 + p_1 (X_1 + U_1 + Z_1)$$

$$\frac{\partial H}{\partial U_1} = 2U_1 + p_1 = 0$$

$$\frac{\partial H}{\partial Z_1} = 2Z_1 + \lambda_1 + p_1 = 0$$

$$\frac{\partial H}{\partial \lambda_1} = Z_1 = 0$$

$$\frac{\partial H}{\partial p_1} = X_1 + U_1 + Z_1 = 0$$

De igual modo se procede para el segundo subsistema.

Podemos entonces calcular los valores de x , u , z y actualizarlos mediante el ensayo del error dado por:

$$\nabla \phi(\lambda) = \begin{pmatrix} z_1 - 2x_2 \\ z_2 - x_1 \end{pmatrix}$$

Si el valor de este error no cumple con el criterio de convergencia es necesario actualizar el valor de λ_1 y λ_2 mediante el siguiente algoritmo:

$$\lambda^k(t) = \lambda^k(t) + \alpha^k d^k(t)$$

α^k = tamaño del paso

d^k = dirección de búsqueda

Cuando se utiliza el método del ascenso más pronunciado d^k se calcula con

$$\underline{d}^k(t) = \underline{e}^k(t) \quad (0 \leq t \leq T)$$

pero si se usa el Método del Gradiente conjugado

$$d^{k+1}(t) = e^{k+1}(t) + \beta^k d^k(t) \quad (0 \leq t \leq T).$$

donde

$$\beta^{k+1} = \frac{\int_0^T (e^{k+1}(t)^T e^{k+1}(t)) dt}{\int_0^T (e^k(t)^T e^k(t)) dt}$$

$$d^0 = e^0$$

2.3 METODO DE TAMURA.-

Como pudimos observar el método desarrollado nos presenta una estructura de dos niveles bien definidos. Lo expresado en el anterior numeral puede extenderse hacia el caso discreto y es en esta forma, como se lo realiza en un computador. El método desarrollado como lo mencionamos es el de coordinación de objetivos.

a continuación vamos a exponer los principales criterios y puntos de vista para desarrollar un método bastante atractivo que presenta una estructura en tres niveles para la optimización de sistemas lineales discretos que tienen un índice de funcionamiento con formas cuadráticas. Este método es conocido como el Método de Tamura ya que este autor fue el primero en desarrollarlo.

Empecemos desarrollando el Método de Coordinación de objetivos para sistemas discretos.

Nuestro problema en realidad es minimizar la función de costo: que sigue:

$$J = \sum_{i=1}^N \left\{ \frac{1}{2} \| X_i(K) \|_{P_i}^2 + \sum_{k=0}^{K-1} \left[\frac{1}{2} (\| X_i(k) \|_{Q_i}^2 + \| Z_i(k) \|_{S_i}^2 + \| U_i(k) \|_{R_i}^2) \right] \right\}$$

$\frac{1}{2} X_i^T P_i X_i$ representa el costo para el intervalo terminal. El sumatorio interno representa el costo en los demás intervalos de la secuencia de optimización.

De acuerdo a lo anteriormente expuesto esta optimización debe tener ciertas restricciones y en realidad estas son:

$$X_i(k+1) = A_i X_i(k) + B_i U_i(k) + C_i Z_i(k) \quad \text{Ecs. 2.3}$$

$$\forall i = 1, 2, 3, \dots, N; \quad k = 0, 1, 2, \dots, K-1$$

$$Z_i(k) = \sum_{j=1}^N L_{ij} X_j(k) \quad \text{Ecs. 2.4}$$

$$\forall k = 0, 1, 2, \dots, K-1; \quad \forall i = 1, 2, \dots, N$$

Ecs. 2.3 Representa el sistema en sí mismo y es una ecuación de diferencias que describe al subsistema i .

Ecs. 2.4 Representa la interacción con los otros subsistemas.

En este punto construyamos nuestra función aumentada que nos permitirá resolver el problema como si no existieran restricciones.

$$L(x, u, z, \lambda) = \sum_{i=1}^N \left\{ \frac{1}{2} \|X_i(K)\|_{P_i}^2 - \sum_{k=0}^{K-1} \left[\frac{1}{2} \left(\|X_i(k)\|_{Q_i}^2 - \|U_i(k)\|_{R_i}^2 - \|Z_i(k)\|_{S_i}^2 \right) - \lambda_i^T \left[Z_i(k) - \sum_{j=1}^N L_{ij} X_j(k) \right] \right] \right\}$$

$$L(x, u, z, \lambda) = \sum_{i=1}^N L_i \quad \text{Ecs. 2.5}$$

y nuestro criterio para minimizar es encontrar el gradiente de esta función y compararlo con un valor cercano a cero $\underline{0}$ en la misma forma que lo hicimos para el caso de funciones continuas.

Pero como podemos observar en Ecs. 2.5 solo hemos considerado la restricción 2.4 pero no está considerada la restricción 2.3, por lo tanto, debemos construir una función aumentada nueva y que representa la Modificación de Tamura, la misma que contempla esta restricción.

De lo expuesto entonces nuestra función será:

$$\bar{L} = \sum_{i=1}^N \left[\frac{1}{2} \|\underline{X}_i(k)\|^2 + \sum_{k=0}^{K-1} \left\{ \frac{1}{2} (\|\underline{X}_i(k)\|^2 + \|\underline{U}_i(k)\|^2 + \|\underline{Z}_i(k)\|^2) + \sum_{j=1}^T (\underline{Z}_i(k) - \sum_{j=1}^N L_{ij} \underline{X}_j(k)) + \underline{p}_i^T \left[-\underline{X}_i(k-1) + \underline{A}_i \underline{X}_i(k) + \underline{B}_i \underline{U}_i(k) + \underline{C}_i \underline{Z}_i(k) \right] \right\} \right]$$

donde \underline{p} son los nuevos multiplicadores de Lagrange para introducir las nuevas restricciones.

A donde el valor inicial de \underline{X} es conocido, es decir $\underline{X}_i(0)$ se dará como dato.

Ahora que el problema está dado apliquemos nuestro criterio de Coordinación. Inicialmente nuestro propósito es minimizar J con respecto a u , pero de acuerdo a lo que hemos mencionado, ahora debemos maximizar la función aumentada para resolver nuestro problema de control puesto que:

$$\text{Max } \bar{L}_p = \text{Min}_u J$$

Anteriormente encontrábamos el Gradiente como criterio de convergencia del problema iterativo para dos niveles pero veamos que sucede en este caso:

El Gradiente de \bar{L} para un p dado es:

$$\text{Grad. } \bar{L} = -\underline{X}_i(k+1) + \underline{A}_i \underline{X}_i(k) + \underline{B}_i \underline{U}_i(k) + \underline{C}_i \underline{Z}_i(k)$$

Para toda k entre 0 y $K-1$

Para toda i entre 1 y N

donde los valores de \underline{X}_i , y \underline{U}_i son las soluciones a nuestro problema de minimización de J con la restricción 2.3.

Con lo anteriormente expuesto en el método de Coordinación podemos encontrar el gradiente para un valor de $\underline{\lambda}$ y nuestro problema también contemplará la restricción 2.4 y habremos contemplado nuestra optimización.

El Langragiano en realidad puede ser descompuesto en N Sublangragianos uno para cada subsistema. Pero el Sublangragiano puede en sí mismo ser descompuesto en función del subíndice k lo que conduce a un nivel ínfimo para obtener una optimización paramétrica diferente a la optimización funcional que se venía realizando. En este caso será una descomposición mediante

subsistemas considerados anteriormente. El esquema de la figura 2.4 muestra esta optimización jerárquica discreta en tres niveles.

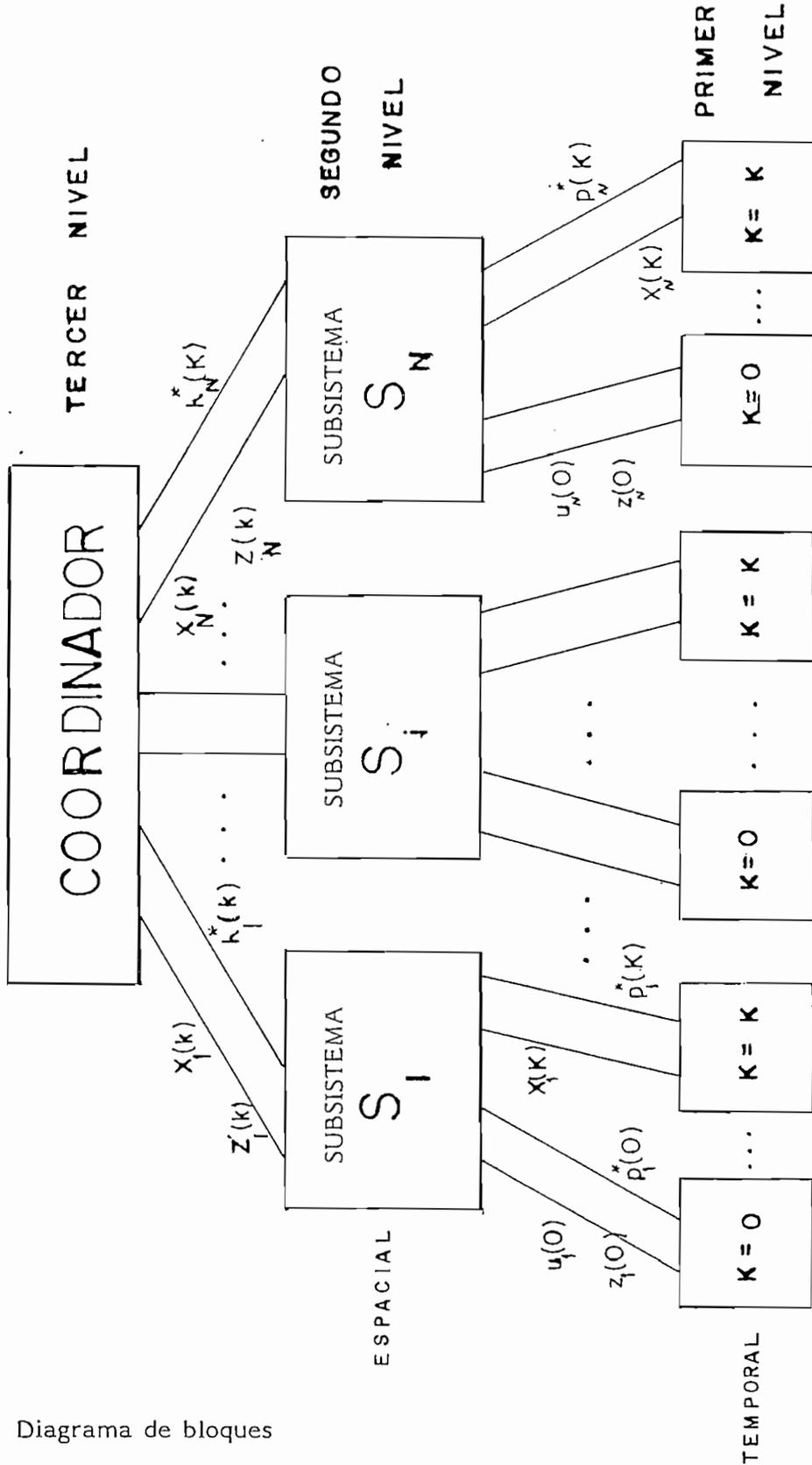


Fig. 2.4 Diagrama de bloques

que representa el Algoritmo de Tamura

2.4 OPTIMIZACION USANDO LA DESCOMPOSICION DE SUBSISTEMAS CON EL CRITERIO DE TAMURA.-

Para empezar nuestro problema planteado construyamos el hamiltoniano de dicho sistema.

El Hamiltoniano para el i^{mo} . subsistema será definido como:

$$H_i (X, U, Z, K) = \frac{1}{2} (\|X_i (k)\|^2_{Q_i} + \|U_i (k)\|^2_{R_i} + \|Z_i (k)\|^2_{S_i}) + \lambda_i^T (Z_i - \sum_{j=1}^N L_{ij} X_j (k)) + p_i^T (A_i X_i (k) + B_i U_i (k) + C_i Z_i (k))$$

y reemplazando en nuestra función aumentada tenemos

$$\bar{L} = \frac{1}{2} \|X_i (k)\|^2_{Q_i} + p_i^T (k-1) X_i (k) + \sum_{k=0}^{K-1} H_i - p_i^T (k-1) x_i (k)$$

y asumiremos que $p_i (-1)$ es 0

Aplicamos las condiciones necesarias para el instante $K=0$.

para $k = 0$

Debemos minimizar $H_i (x_i (0), u_i (0), z_i (0))$ dado el $x_i (0)$ con respecto a $u_i (0)$ y $z_i (0)$ y considerando un valor fijo de $p = p^*$

Aplicando la diferenciación vectorial tenemos:

$$\frac{\partial H_i}{\partial \underline{u}_i(0)} = \underline{R}_i \underline{u}_i(0) + \underline{B}_i^T \underline{p}_i^*(0) = 0$$

$$\underline{R}_i \underline{u}_i(0) = -\underline{B}_i^T \underline{p}_i^*(0)$$

$$\underline{u}_i(0) = -\underline{R}_i^{-1} \underline{B}_i^T \underline{p}_i^*(0)$$

$$\frac{\partial H_i}{\partial \underline{z}_i(0)} = \underline{S}_i \underline{z}_i(0) + \underline{\lambda}_i^*(0) + \underline{C}_i^T \underline{p}_i^*(0) = 0$$

$$\underline{S}_i \underline{z}_i(0) = -\underline{C}_i^T \underline{p}_i^*(0) - \underline{\lambda}_i^*(0)$$

$$\underline{z}_i(0) = -\underline{S}_i^{-1} (\underline{C}_i^T \underline{p}_i^*(0) + \underline{\lambda}_i^*(0))$$

Para $k = 1, 2, 3, \dots, K-1$ es necesario minimizar $H_i(\underline{x}_i(k), \underline{u}_i(k), \underline{z}_i(k),$

$k) - \underline{p}_i^{*T}(k-1) \underline{x}_i(k)$ con respecto a: $\underline{x}_i(k), \underline{u}_i(k), \underline{z}_i(k)$.

Entonces:

$$\underline{R}_i \underline{u}_i(k) + \underline{B}_i^T \underline{p}_i^*(k) = \frac{\partial H_i}{\partial \underline{u}_i} = 0$$

$$-\underline{R}_i \underline{u}_i(k) = \underline{B}_i^T \underline{p}_i^*(k)$$

$$\underline{u}_i(k) = -\underline{R}_i^{-1} \underline{B}_i^T \underline{p}_i^*(k)$$

$$\frac{\partial H_i}{\partial \underline{z}_i(k)} = 0 = \underline{S}_i \underline{z}_i(k) + \underline{\lambda}_i^*(k) + \underline{C}_i^T \underline{p}_i^*(k)$$

$$- \underline{S}_i \underline{z}_i(k) = \underline{\lambda}_i^*(k) + \underline{C}_i^T \underline{p}_i^*(k)$$

$$\underline{z}_i(k) = - \underline{S}_i^{-1} (\underline{C}_i^T \underline{p}_i^*(k) + \underline{\lambda}_i^*(k))$$

$$\frac{\partial H}{\partial \underline{x}_i(k)} = \underline{Q}_i \underline{x}_i(k) - \sum_{j=1}^N (\underline{\lambda}_j^{*T} \underline{L}_{ji})^T + \underline{A}_i^T \underline{p}_i^*(k) - \underline{p}_i^*(k-1) = 0$$

$$- \underline{Q}_i \underline{x}_i(k) = \underline{A}_i^T \underline{p}_i^*(k) - \underline{p}_i^*(k-1) - \sum_{j=1}^N (\underline{\lambda}_j^{*T} \underline{L}_{ji})^T$$

$$\underline{x}_i(k) = - \underline{Q}_i^{-1} (\underline{A}_i^T \underline{p}_i^*(k) - \underline{p}_i^*(k-1) - \sum_{j=1}^N (\underline{\lambda}_j^{*T} \underline{L}_{ji})^T)$$

Finalmente para $k = K$ es necesario minimizar:

$$\frac{1}{2} \Pi \underline{x}_i(K) \Pi^T \underline{p}_i^*(K-1) \underline{x}_i(K)$$

con respecto a $\underline{x}_i(K)$ y obtenemos para el instante final que:

$$\underline{p}_i \underline{x}_i(K) = \underline{p}_i^*(K-1)$$

$$\underline{x}_i(K) = \underline{p}_i^{-1} \underline{p}_i^*(K-1)$$

Por lo tanto nuestro algoritmo de solución será:

Para $K = 0$

$$\underline{u}_i(0) = - \underline{R}_i^{-1} \underline{B}_i^T \underline{p}_i^*(0)$$

$$\underline{z}_i(0) = -\underline{S}_i^{-1} (\underline{C}_i^T \underline{p}_i^*(0) + \underline{\lambda}_i^*(0))$$

Para $k = 1, 2, \dots, K-1$

$$\underline{u}_i(k) = -\underline{R}_i^{-1} \underline{B}_i^T \underline{p}_i^*(k)$$

$$\underline{z}_i(k) = -\underline{S}_i^{-1} (\underline{C}_i^T \underline{p}_i^*(k) + \underline{\lambda}_i^*(k))$$

$$\underline{x}_i(k) = -\underline{Q}_i^{-1} (\underline{A}_j^T \underline{p}_i^*(k) - \underline{p}_i^*(k-1) - \sum_{j=1}^N (\underline{\lambda}_j^{*T} \underline{L}_{ij})^T)$$

Para $k = K$

$$\underline{x}_i(K) = \underline{P}_i \underline{p}_i^*(K-1)$$

Por lo que el problema de minimizar J quedaría completamente resuelto - usando una estructura de tres niveles.

Para el primer nivel sustituimos los valores de $\underline{\lambda}^*(k)$ y $\underline{p}^*(k)$ dentro de las expresiones obtenidas anteriormente y calculamos los valores de $\underline{x}_i(k)$, $\underline{u}_i(k)$, y $\underline{z}_i(k)$; los valores serán óptimos y que permitirán para que - en un segundo nivel calcular el gradiente de la función ampliada para un va-
lor dado de \underline{p}^* , es decir $\nabla_{\underline{p}} M|_{\underline{p}=\underline{p}^*}$. Con este valor podemos mejorar el va-
lor de \underline{p}^* hasta optimizarlo.

Para el tercer nivel en función de \underline{p}^* optimizado podemos encontrar el gra-
diente de $\nabla_{\underline{\lambda}} M$ y mejorar $\underline{\lambda}^*$ para optimizarlo. Habremos encontrado los valores
óptimos cuando los dos gradientes se hagan cero.

Finalmente analicemos el método de Actualización de los valores de \underline{p}^* y $\underline{\lambda}^*$

Usando la técnica del gradiente conjugado \underline{p}^* y $\underline{\lambda}^*$ pueden ser actuali-
zados mediante el siguiente algoritmo

Sea i = número de iteraciones:

Entonces:

$$\underline{p}^{i+1} = \underline{p}^i + \alpha_i \cdot \underline{d}^i$$

donde: $\underline{d}^i = \underline{e}(p^i) + \beta_{i-1} \underline{d}^{i-1}$

$$\underline{d}^1 = \underline{e} (p^1)$$

$\underline{e} (p^i)$ = error o valor del gradiente para un \underline{p} dado en la $i^{\text{ésima}}$ iteración:

Además

$$\beta_{i-1} = \frac{\sum_{k=0}^{K-1} \underline{e} (p^i(k))^T * \underline{e} (p^i(k))}{\sum_{k=0}^{K-1} \underline{e} (p^{i-1}(k))^T * \underline{e} (p^{i-1}(k))}$$

Definido así el problema pasemos a su implantación computacional en el siguiente capítulo.-

Ejemplo del método de Optimización usando el algoritmo de Tamura.-

Considere el sistema dado por las siguientes ecuaciones de diferencias:

$$X_1 (K + 1) = X_1 (K) + U_1 (K) + Z_1 (K)$$

$$X_2 (K + 1) = - X_2 (K) + U_2 (K) + Z_2 (K)$$

$$Z_1 (K) = X_2 (K)$$

$$Z_2 (K) = X_1 (K)$$

La función de costo viene dada por:

$$J = \sum_{k=0}^3 \frac{1}{2} \left[X_1^2 (K) + X_2^2 (K) + U_1^2 (K) + M_2^2 (K) + Z_1^2 (K) + Z_2^2 (K) \right]$$

El Lagrangiano puede ser escrito como:

$$L(X, U, Z, \lambda) = \sum_{k=0}^3 \frac{1}{2} \left[(X_1^2(k) + X_2^2(k) + U_1^2(k) + U_2^2(k) + Z_1^2(k) + Z_2^2(k)) + \right.$$

$$\left. \lambda_1(k) (Z_1(k) - X_2(k)) + \lambda_2(k) (Z_2(k) - X_1(k)) \right]$$

Con las restricciones dadas por las ecuaciones diferenciales.

Podemos al Lagrangiano escribirlo de otra forma:

$$L = \sum_{k=0}^3 \frac{1}{2} \left[(X_1^2 + U_1^2 + Z_1^2) + (\lambda_1 Z_1 - \lambda_2 X_1) + \frac{1}{2} (X_2^2 + U_2^2 + Z_2^2) + \lambda_2 Z_2 - \lambda_1 X_2 \right]$$

que es igual a la suma de los lagrangianos y que están sujetos a restricciones dadas por las ecuaciones de estado. Aplicando el algoritmo de Tamura tenemos que para el primer Lagrangiano se tiene:

$$U_1(0) = -p_1(0)$$

$$Z_1(0) = -p_1(0) + \lambda_1(0)$$

$$X_1(1) = -(p_1(1) - p_1(0) - \lambda_2(1))$$

$$U_1(1) = -p_1(1)$$

$$Z_1 (1) = -(p_1 (1) + \lambda_1 (1))$$

$$X_1 (2) = - (p_1 (2) - p_1 (1) - \lambda_2 (2))$$

$$U_1 (2) = - p_1 (2)$$

$$Z_1 (2) = - (p_1 (2) + \lambda_1 (2))$$

$$X_1 (3) = p_1 (2)$$

Una vez calculados los X, U, Z, pasamos a calcular el gradiente dado por:

$$\text{grad.} = - X_1 (K+1) + X_1 (k) + U_1 (k) + Z_1 (k)$$

sino cumple debe actualizarse p , de tal forma que converja para un cierto valor de p^* óptimo.

Luego realizaremos sobre el tercer nivel el cálculo del nuevo gradiente dado por:

$$\text{grad.} = Z_1 (k) - \sum_{j=1}^N L_{ij} X_j (k)$$

Es necesario anotar que el valor dado por p se almacenó ya y el gradiente entre el 2do. y 3er. nivel lo que hace es mejorar los valores de X, U, Z, actualizando el valor de los multiplicadores de Lagrange $\lambda^* (K)$.

La actualización de los multiplicadores se lo hace mediante cualquiera de las

técnicas ya mencionadas en el algoritmo de Coordinación.

CAPITULO III

SIMULACION DIGITAL PARA OPTIMIZACION JERARQUICA

- 3.1 ALGORITMO DE OPTIMIZACION DE SISTEMAS LINEALES DISCRETOS CON FUNCION DE COSTO CUADRATICO
- 3.2 PROGRAMA MAESTRO Y ENTRADA DE DATOS
- 3.3 IMPLEMENTACION COMPUTACIONAL DEL ALGORITMO DE TAMURA
- 3.4 SALIDA DE RESULTADOS

3.1 ALGORITMO DE OPTIMIZACION DE SISTEMAS LINEALES DISCRETOS CON FUNCION DE COSTO CUADRATICO.-

En el capítulo anterior realizamos todo el análisis del Algoritmo de Tamura, para optimizar sistemas lineales con función de costo cuadrático.

En este capítulo empezaremos por enunciar nuestro problema, a continuación pondremos el Algoritmo por medio del cual resolveremos numericamente el problema e implementaremos un programa de computador que permita correr ciertos ejemplos típicos.

Debemos tener en cuenta que la solución del problema numérico ameritó el uso de un computador para su solución pero además de esto, la necesidad del control jerárquico obligó a la implementación del problema usando el algoritmo aquí presentado.

Nuestro sistema se halla descrito mediante ecuaciones de diferencias (sistema dinámico) y para determinado subsistema tenemos:

$$\underline{B}_i (k + 1) = \underline{A}_i X_i (k) + \underline{B}_i u_i (k) + C_i Z_i (k)$$

donde el término $C_i Z_i (k)$ aparece luego de haberse realizado el desacoplamiento y en realidad $Z_i (k)$ representa la combinación lineal de los estados de los demás subsistemas, y que puede expresarse mediante el siguiente sumatorio:

$$\underline{Z}_i (k) = \sum_{j = 1}^N L_{ij} X_j (k)$$

con i variando entre 1 y n

con $k = 0, 1, 2, \dots, K-1$

El valor de K viene dado por la función de costo que tiene la siguiente expresión:

$$J = \sum_{i=1}^N \left[\frac{1}{2} \left\| \underline{X}_i(k) \right\|_{\underline{P}_i}^2 + \frac{1}{2} \sum_{k=0}^{K-1} \left\{ \left\| \underline{X}_i(k) \right\|_{\underline{Q}_i}^2 + \left\| \underline{U}_i(k) \right\|_{\underline{R}_i}^2 \right\} \right]$$

Para cuando k varia entre 0 y k y para i entre 1 - n .

Aquí podemos observar que todos los sistemas se hallan descritos mediante ecuaciones que describen sistemas discretos y si se nos presentara un sistema continuo, es necesario discretizarlo para poder aplicar en él, las ecuaciones que se hallan desarrolladas en el presente trabajo. Para discretizar un sistema continuo podemos recurrir a cualquier método que para el efecto existe.

De acuerdo al capítulo anterior nuestro algoritmo es el siguiente:

Para el nivel 1:

Dados los multiplicadores de Lagrange $\underline{\lambda}_i^*(k)$, $\underline{p}_i^*(k)$ calcular:

Para $k = 0$:

$$\underline{U}_i(0) = -\underline{R}_i^{-1} \underline{B}_i^T \underline{p}_i^*(0)$$

$$\underline{Z}_i(0) = -\underline{S}_i^{-1} (\underline{C}_i^T \underline{p}_i^*(0) + \underline{\lambda}_i^*(0))$$

para $k = 1$ hasta $K - 1$

$$\underline{U}_i(k) = -\underline{R}_i^{-1} \underline{B}_i^T \underline{p}_i^*(k)$$

$$\underline{Z}_i(k) = -\underline{S}_i^{-1} (\underline{C}_i^T \underline{p}_i^*(k) + \underline{\lambda}_i^*(k))$$

$$\underline{X}_i(k) = -\underline{Q}_i^{-1} (\underline{A}_i^T \underline{p}_i^*(k) + \underline{p}_i^*(k-1) + \sum_{j=1}^N (\underline{\lambda}_j(k) \cdot \underline{L}_{ji})^T)$$

para $k = K$

$$\underline{X}_i(k) = \underline{P}_i^{-1} \underline{p}_i^*(k-1)$$

para $k = 0$ hasta K

para $i = 1$ hasta N (todos los subsistemas).-

Para el nivel 2:

Los $\underline{X}_i(k)$, $\underline{U}_i(k)$, $\underline{Z}_i(k)$ calculados son usados para verificar si $\nabla \underline{M}_p$ que representa el error, está dentro de los criterios de convergencia. Si todavía no se encuentra es necesario actualizar el valor de \underline{p} mediante alguna de las técnicas del gradiente y realizar nuevamente el cálculo de \underline{x} , \underline{u} , y \underline{z} hasta que $\nabla \underline{M}_p$ sea del valor deseado.

Para el nivel 3:

Una vez que se ha obtenido el ∇M_p deseado se guarda el valor de \underline{p} y el mismo que será usado para mejorar los valores de $\underline{\lambda}$ usando el otro gradiente: $\nabla \phi(\underline{\lambda})$

$$\nabla M_p \Big|_{\underline{p} = \underline{p}^*} = - \underline{X}_i (K+1) + \underline{A}_i \underline{X}_i (K) + \underline{B}_i \underline{U}_i (K) + \underline{C}_i \underline{Z}_i (K)$$

$$= 0, 1, 2, \dots, K$$

$$= 1, 2, 3, \dots, N$$

y representa el error entre los niveles 1 y 2

$$\nabla \phi(\underline{\lambda}) \Big|_{\underline{\lambda} = \underline{\lambda}^*} = \underline{Z}_i (K) - \sum_{j=1}^N \underline{L}_{ij} \underline{X}_j (K)$$

$$= 0, 1, 2, \dots, K$$

$$= 1, 2, 3, \dots, N$$

y representa el error entre los niveles 2 y 3.

La implantación del método de algoritmo de Tamura sugiere una estructura de tres niveles que gráficamente puede visualizarse mediante el gráfico de la figura 3.1

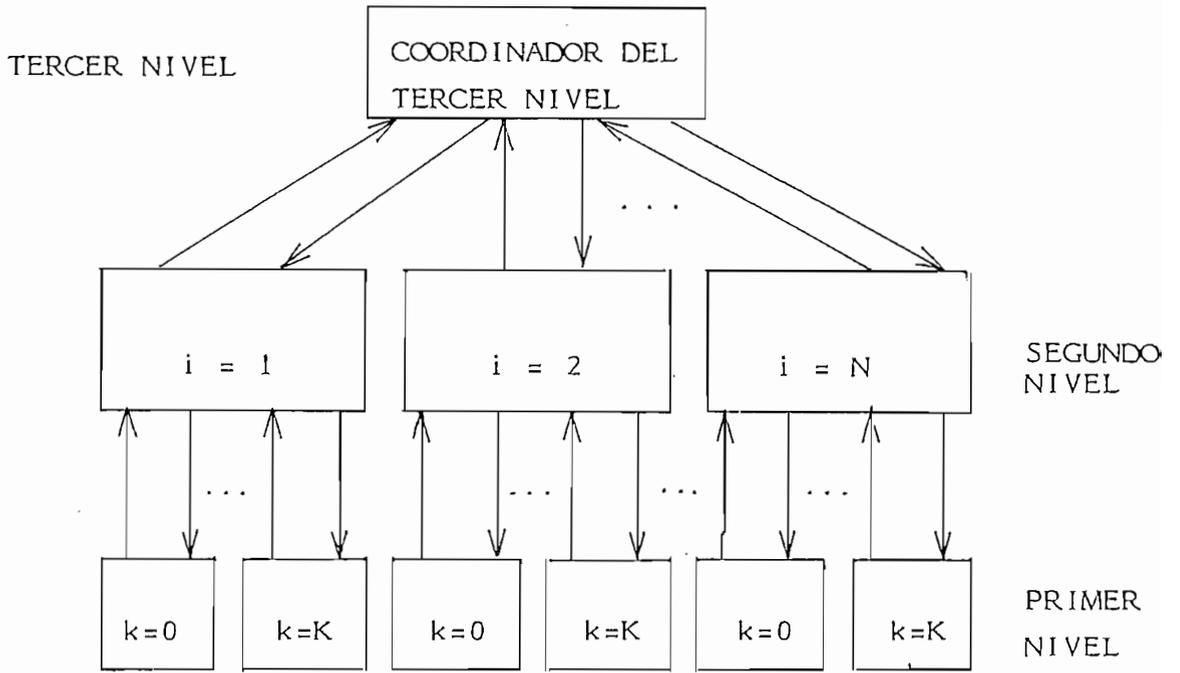


Fig. 3.1 Esquema de descomposición jerárquica dado por Tamura.

Los ejemplos fueron conocidos en el programa implementado en el computador DATAGENERAL 450 de la UNIVERSIDAD CENTRAL DEL ECUADOR, - usando el lenguaje intérprete BASIC.

3.2 PROGRAMA MAESTRO Y ENTRADA DE DATOS.-

El programa se compone de tres partes principales. En realidad dado que se usa el computador DATA GENERAL 450 no hace falta la segmentación del programa y en consecuencia no se usa un programa maestro. Así pues el programa está constituido por un solo conjunto de instrucciones divididas como muestra el diagrama de bloque de la figura 3.2

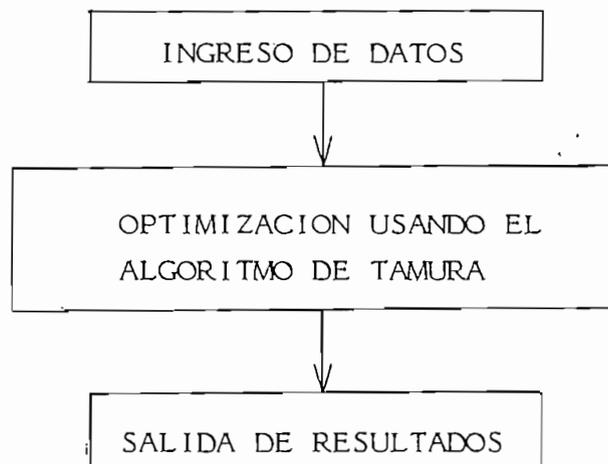


Fig. 3.2 Diagrama de bloques del programa de optimización jerárquica.-

Cada uno de los bloques puede estar desarrollado en la siguiente forma tomando en cuenta que nuestra implementación se lo hace con el Lenguaje intérprete BASIC.

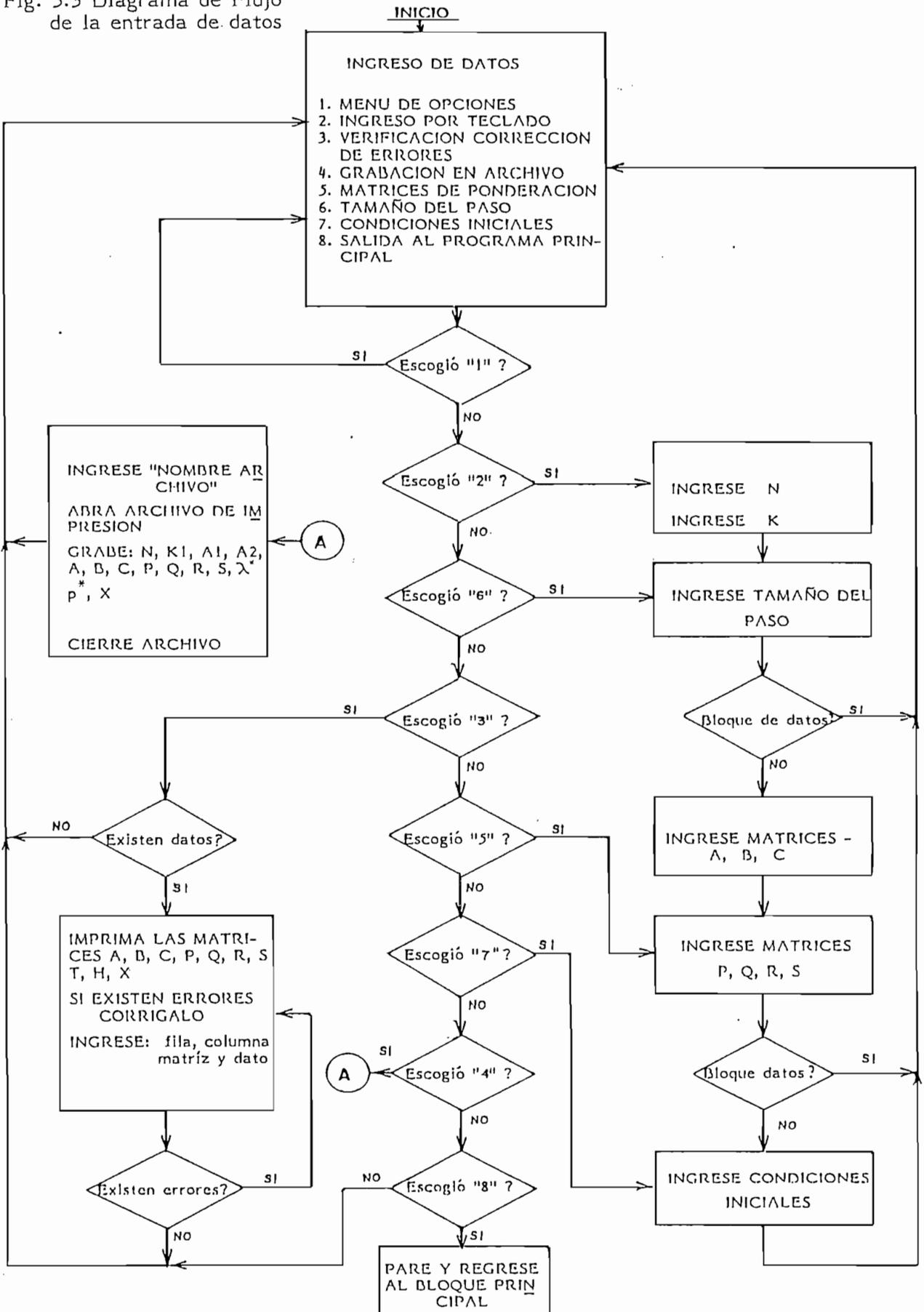
Entrada de datos.- De acuerdo al algoritmo presentado los datos necesarios para la implementación deben ser los siguientes: el orden

del sistema que lo representaremos por N ; el número de estados temporales de la función de costo que está representada por K ; las matrices de datos, A, B , del sistema dinámico; las matrices de ponderación de la función de costo P, Q, R y S ; así como los valores iniciales de los estados \underline{x}^* y \underline{p}^* de los multiplicadores de Lagrange

Es necesario anotar que la corrida de varios ejemplos típicos debería contemplar la posibilidad de cambiar ciertos valores de ponderación de los estados - para un mismo sistema, también es posible que los datos de información queden almacenados en forma permanente dentro de un archivo de datos.

Dentro del algoritmo es necesario considerar el tamaño del paso en el mejoramiento o actualización de los multiplicadores de Lagrange para lo cual se introduce cierto porcentaje de cambio Δ . Todo esto puede observarse en - el diagrama de flujo de la figura 3.3

Fig. 3.3 Diagrama de Flujo de la entrada de datos



3.3 IMPLEMENTACION COMPUTACIONAL DEL ALGORITMO DE TAMURA

Una vez ingresados los datos, la implementación computacional del algoritmo se lo hace fundamentalmente en función del método presentado y analizado el capítulo II e introducido como algoritmo en este capítulo. Cabe anotar que la estructura del programa permite observar claramente como la optimización jerárquica computacional contempla los tres niveles anteriormente expuestos. Dado que el BASIC del Computador usado contempla operaciones matriciales como: ingreso de matrices, inversión de matrices, multiplicación de matrices y asignación de valores de una matriz, se han evitado muchas subrutinas de tal forma que con una sola instrucción se han obviado todo un grupo extenso de instrucciones teniéndose como resultado un programa bastante compacto y resumido. La disposición física del programa hace fácil observar los diferentes bloques que están de acuerdo con la estructura jerárquica planteada en este trabajo según se muestra en la figura 3.4

Es necesario recalcar que por limitaciones computacionales el programa solo calcula los valores óptimos realizando un dimensionamiento unitario de tal forma que los "subsistemas" así considerados son sistemas que tienen en sus ecuaciones de diferencias tan solo coeficientes numéricos y no vectoriales.

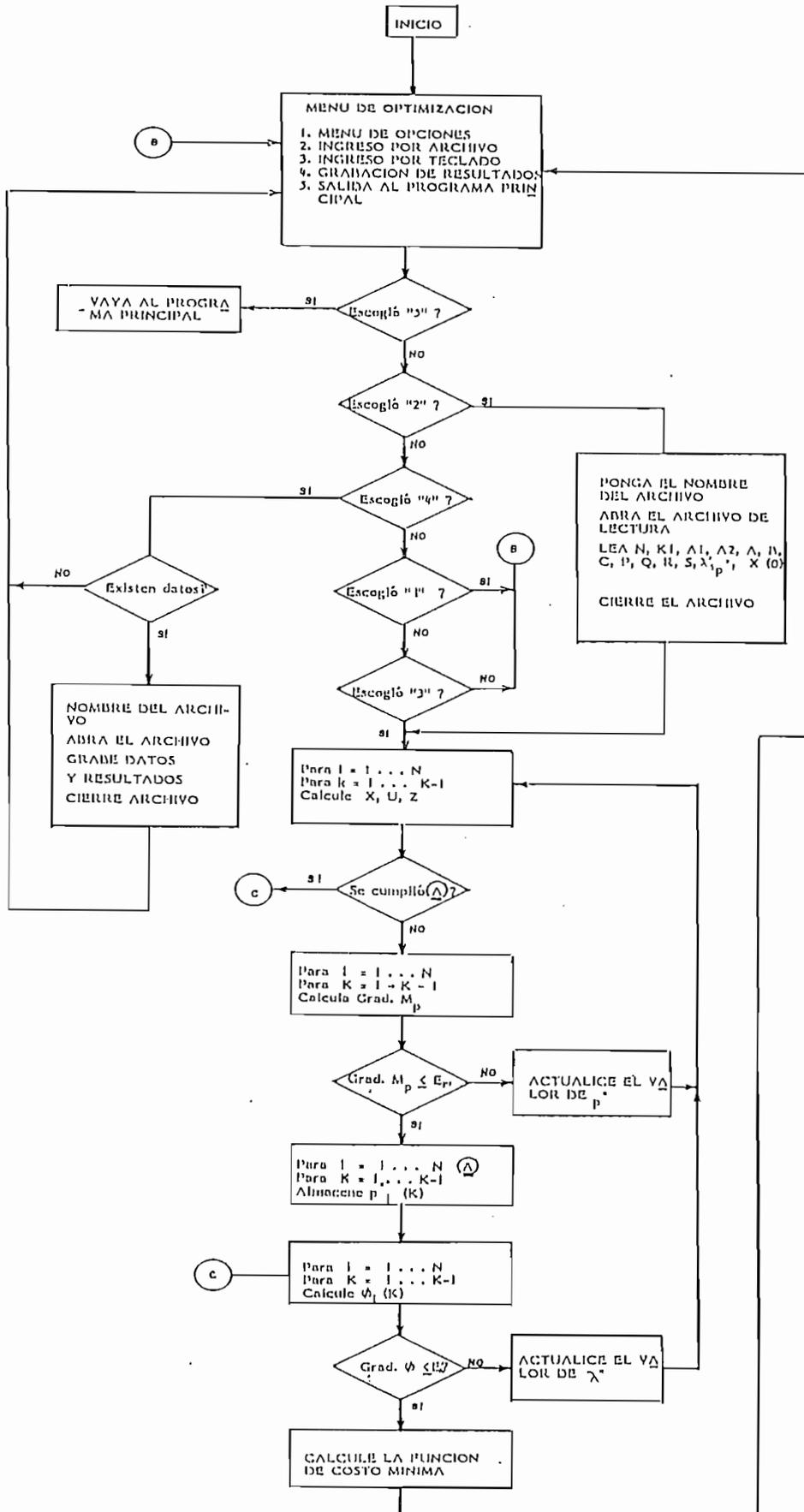


Fig. 3.4 Diagrama de flujo de la Optimización usando el Algoritmo de Tamura

Para actualizar el valor de los multiplicadores de Lagrange p^* y λ^* se lo hace mediante la técnica de la conjugada del gradiente que usa una ponderación de los valores ya calculados para encontrar el valor óptimo de estos multiplicadores.

Algoritmo:

Cálculase $p_i (K^*)$ de la siguiente forma:

$$p_{i+1} = p_i + \alpha_i d_i$$

donde α_i es el tamaño del paso para la iteración i y que se calcula como:

$$\alpha_i = \frac{\frac{q}{100} |J^i|}{\left\| \frac{\partial H^i}{\partial u} \right\|^2}$$

donde $|J^i|$ es el valor absoluto de la función de costo para la iteración i .

q es el porcentaje de variación que deseamos de J

H es el hamiltoniano del sistema

u es el vector de control

d_i es la dirección de búsqueda en la iteración i y que determina hacia donde debe calcularse el valor de p en función de la pendiente del gradiente que se está calculando.

$$d_i = \nabla_{p=p_i} M p_i + \beta_{i-1} \nabla_{p=p_{i-1}} M p_{i-1}$$

$$\beta_{i-1} = \sum_{l=1}^N \frac{\| \nabla_{p=p_l} M p_l \|^2}{\| \nabla_{p=p_{l-1}} M p_{l-1} \|^2}$$

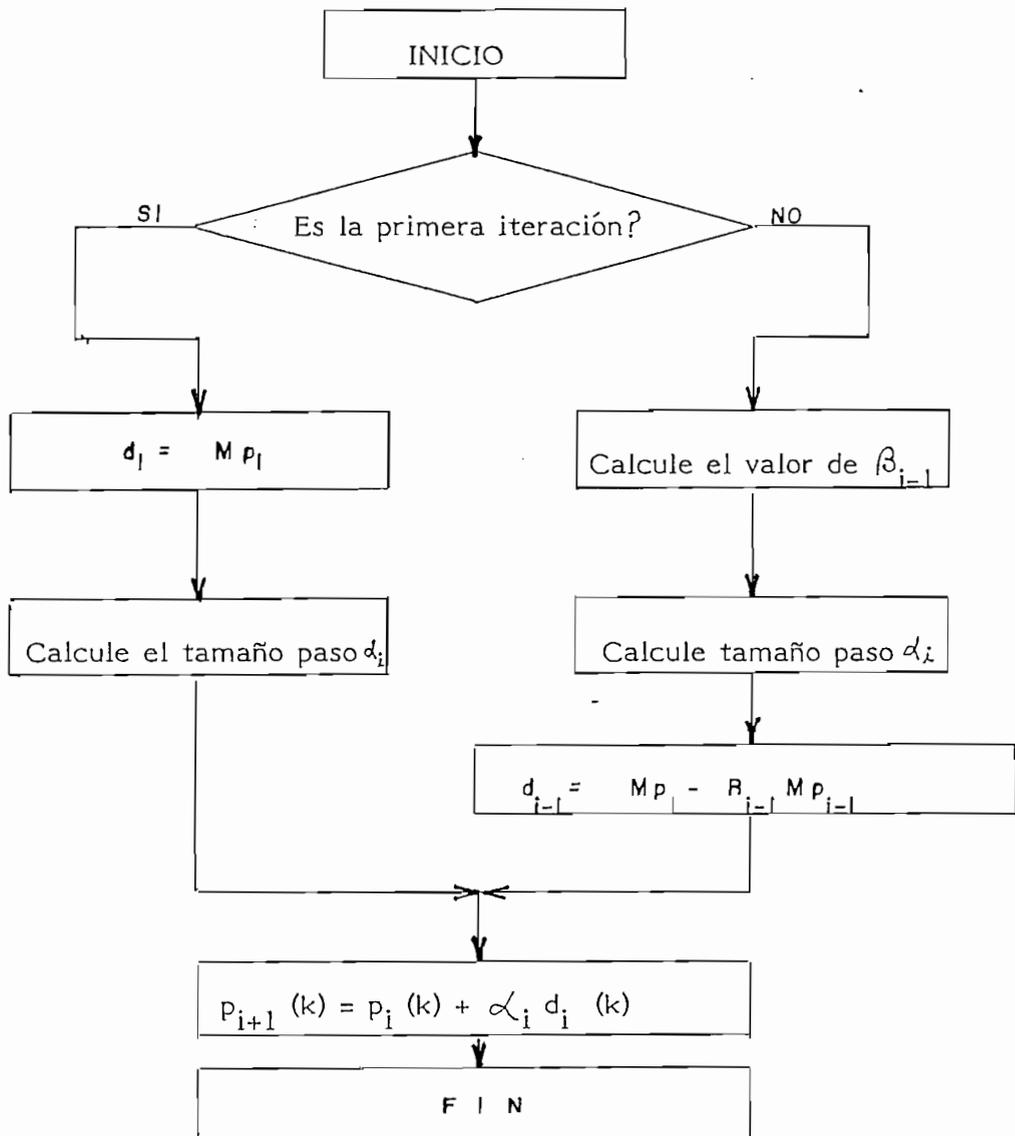
$$d_i = \nabla M p / p = p_i$$

Por lo tanto el diagrama de flujo es el siguiente y que varía en el valor de $\nabla M p$ por $\nabla \phi \lambda$ para cuando se desee actualizar el valor de p^* o λ^* respectivamente así:

$$\nabla M p = -X_f(k-1) - A_i X_i(k) - B_i U_i(k) - C_i Z_i(k)$$

$$\nabla \phi \lambda = Z_i(k) - \sum_{j=1}^N L_{ij} X_j(k)$$

Por lo tanto se tiene la siguiente figura.



Para calcular el valor del tamaño del paso se usa la siguiente expresión:

$$\alpha_i = \frac{q/100 \cdot |J^i|}{\| \partial H / \partial u \|_i^2}$$

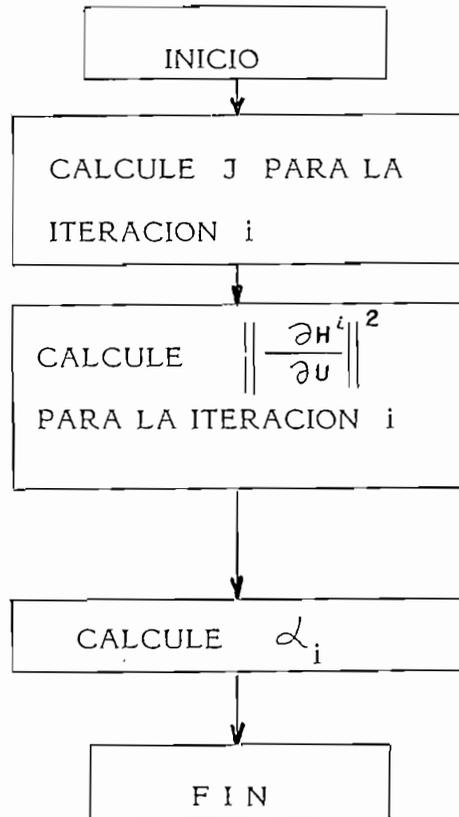


Fig. 3.6 Diagrama de flujo del cálculo del paso.

Al término del cálculo se preguntará si desea imprimir resultados o no. Si se desea imprimir resultados se pasará al bloque de salida de resultados volviendo al Menú principal o en caso contrario los valores obtenidos serán guardados en archivo.

3.4 IMPRESION DE RESULTADOS.-

Una vez que se ha terminado el cálculo de los valores óptimos se regresará al programa principal, en cuyo menú existe la impresión de resultados.

El menú de impresión de resultados contempla la posibilidad de realizar la impresión de resultados si estos se encuentran en la memoria real del computador o si se encuentran grabados en disco en algún archivo de resultados.

Por lo mismo debe contemplar la posibilidad de imprimir el nombre del archivo; el sistema calculado, los valores iniciales, los valores óptimos de las trayectorias de los controles, de los estados, así como también el valor mínimo de la función de costo. De esta forma deben considerarse los siguientes aspectos:

1. Averiguar el sistema que va a imprimirse
2. Averiguar si el periférico de salida está listo para recibir la información de los resultados.
3. Imprimir los resultados.

A continuación se muestra el diagrama de bloques de la impresión de resultados.

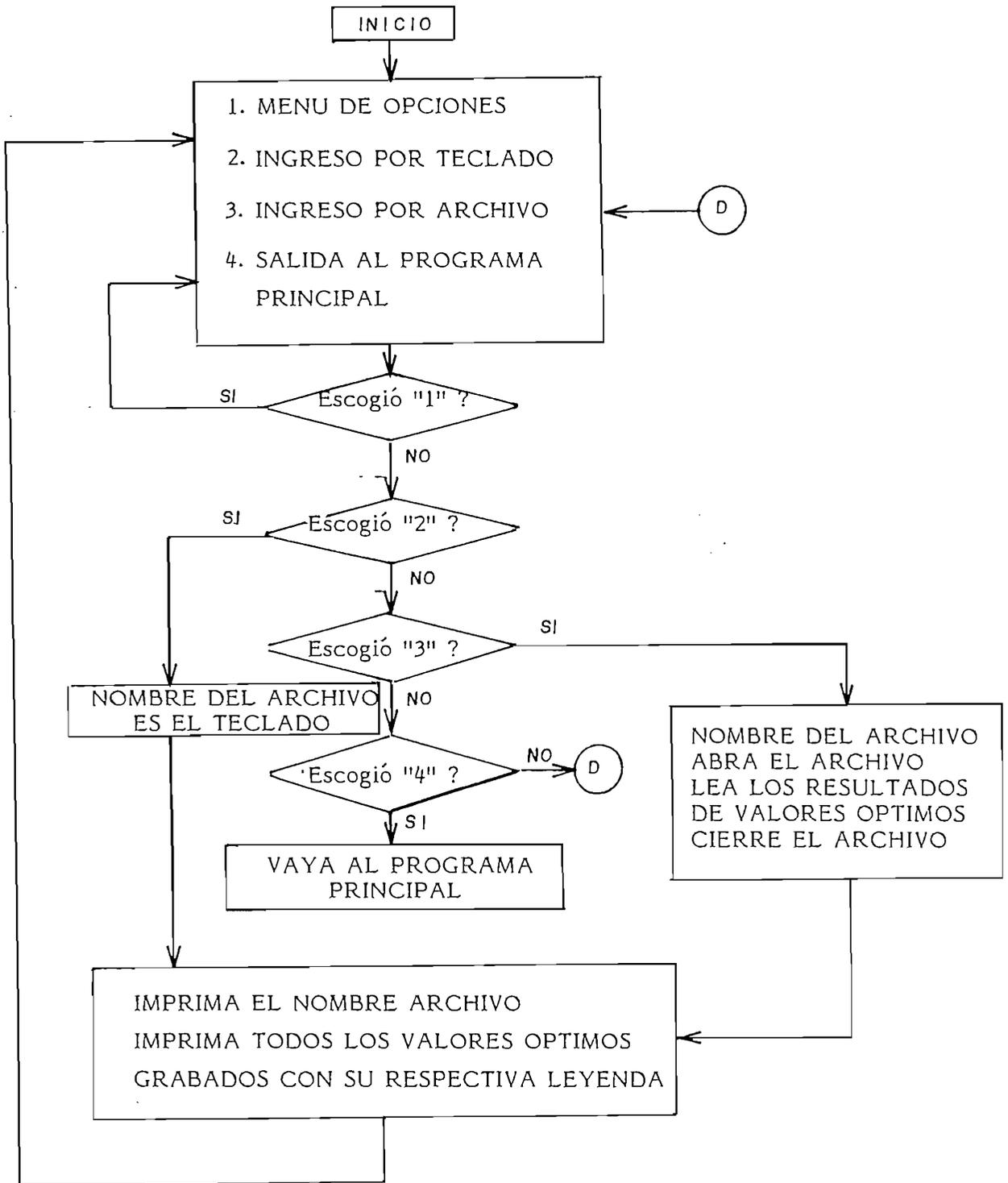


Fig. 3.5 Impresión resultados Diagrama de Flujo

La Fig. 3.6 Muestra el diagrama de flujo de la posibilidad de imprimir los datos total o parcial del problema, para casos en los que se corrie-

ra el mismo sistema con diferentes valores en las matrices de ponderación, en las condiciones iniciales, en el tamaño del paso, etc.

El Menú de impresión de resultados debe contemplar las siguientes opciones:

- Menú de impresión de resultados.-
- Nombre del archivo de resultados
 - Lista de opciones
 - Impresión del sistema:
Matrices A, B, C
 - Matrices de ponderación P, Q, R, S,
 - Condiciones iniciales
 - Trayectorias óptimas
 - Controles óptimos
 - Estados de realimentación Z_i óptimos
 - Función de costo mínimo
 - Salida al Menú de impresión

fig.3.6

Lista de opciones: Menú de impresión

En el capítulo IV se presentarán ejemplos típicos que muestran como se elaboró la implementación digital del algoritmo de Tamura.

CAPITULO IV

RESULTADOS Y CONCLUSIONES

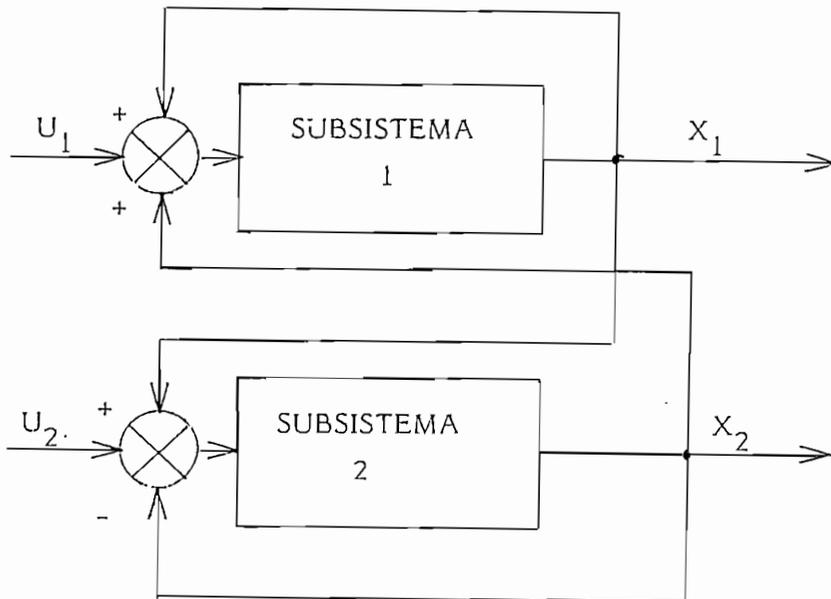
- 4.1 RESULTADOS
- 4.2 CONCLUSIONES
- 4.3 RECOMENDACIONES

4.1 RESULTADOS.-

Para ilustrar la bondad y eficiencia de los programas desarrollados se presentan los resultados de algunos ejemplos de optimización mediante la descomposición jerárquica usando el algoritmo de Tamura. Como no es nuestro objetivo la modelación de sistemas descentralizados, en los ejemplos que presentamos a continuación se parte ya del modelo matemático de los sistemas considerados. Los resultados obtenidos se ensayan con diferentes alternativas como son: Cambio en los valores iniciales de los estados del sistema, cambio en el tamaño del paso, cambio en los valores de las matrices de ponderación, etc.

EJEMPLO 1

Sea el sistema de la figura 4.1 cuyo sistema de ecuaciones de diferencias es el siguiente:



$$X_1 (K + 1) = X_1 (K) + U_1 (K) + X_2 (K)$$

$$X_2 (K + 2) = X_1 (K) + U_2 (K) - X_2 (K)$$

Sea la función de costo:

$$J = \sum_{K=0}^3 \frac{1}{2} \left[X_1^2 (K) + X_2^2 (K) + U_1^2 (K) + U_2^2 (K) + Z_1^2 (K) Z_2^2 (K) \right]$$

Agrupando los datos y poniéndoles en forma matricial se tiene:

MATRIZ A

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

MATRIZ B

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

MATRIZ P

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

MATRIZ Q

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

MATRIZ R

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

MATRIZ S

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Para el sistema se dan como valores iniciales:

MÁTRIZ λ	MATRIZ p^*	MATRIZ X
$\begin{vmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{vmatrix}$	$\begin{vmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{vmatrix}$	$\begin{vmatrix} 0.5 & 0.7 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{vmatrix}$

Los valores iniciales del tamaño del paso son: $A1 = 0.2$
 $A2 = 0.3$

Los resultados podemos observar en las siguientes hojas del computador.

El resultado converge para 24 iteraciones entre el nivel 1 y 2 y para 15 entre el nivel 2 y 3.

Existen ciertas leyendas del menú que resultaron difícil de discriminar especialmente para estos primeros ejemplos. En ejemplos sucesivos se hará una impresión del sistema en su totalidad.

CONDICIONES INICIALES DEL SISTEMA-----> RES1
 TAMANIO DEL PASO G1 = .2
 TAMANIO DEL PASO G2 = .3
 TRAYECTORIAS INICIALES XI(0)

SISTEMA -----> 1

X10 = (.5)

SISTEMA -----> 2

X10 = (.7)

MATRIZ P*
 0.1829- 0.2561
 0.0488 0.0683
 0.0122- 0.0171

MATRIZ H*
 - 0.8828- 0.2439
 0.0695- 0.1012
 - 0.0317- 0.0098

MATRIZ 7
 0.6999 0.5000
 0.0000 0.0000
 0.0000 0.0000

MATRIZ X
 0.5000 0.7000
 0.0329- 0.1183
 0.0268 0.0195
 0.0122- 0.0171

MATRIZ II
 - 0.1829 0.2561
 - 0.0488- 0.0683
 - 0.0122 0.0171

$J = 1 (X1(K)^2 * PI + ((X1(K)^2 * Q1 + U1)^2 * R1 + Z1^2 * S1)) / 2$
 LA FUNCION DE COSTO ES = 1,2826288

NUMERO DE ITERACIONES ENTRE NIVEL 1 Y 2 = ----> 24

NUMERO DE ITERACIONES ENTRE NIVEL 2 Y 3 = ----> 15

MATRIZ A

1.0000 1.0000
1.0000- 1.0000

MATRIZ B

1.0000 0.0000
0.0000 1.0000

MATRIZ C

1.0000 0.0000
0.0000 1.0000

RES1

MATRIZ P

1.0000 0.0000
0.0000 1.0000

MATRIZ Q

1.0000 0.0000
0.0000 1.0000

MATRIZ R

1.0000 0.0000
0.0000 1.0000

MATRIZ S

1.0000 0.0000
0.0000 1.0000

EJEMPLO 2

Sea el sistema descrito por las mismas ecuaciones del ejemplo 1, pero lo que en realidad vamos a cambiar es el tamaño del paso para poder observar la influencia de este elemento en el cálculo de los valores óptimos.

Por lo tanto:

$$a1 = 0.1$$

$$a2 = 0.1$$

Los resultados de la simulación digital de este ejemplo podemos observarlos en las hojas del computador presentadas a continuación.

Como podemos observar los resultados de la variación fundamental se presenta en el número de iteraciones. Como el tamaño del paso disminuyó, el número de iteraciones aumentó así para el nivel 1 - 2 aumentó de 24 a 53 y para el nivel 2 - 3 aumentó de 15 a 46. En todo caso el número de iteraciones sigue siendo menor para el segundo nivel.

MATRIZ A
 1.0000 1.0000
 0.0000 1.0000

MATRIZ B
 1.0000 0.0000
 0.0000 1.0000

MATRIZ C
 1.0000 0.0000
 0.0000 1.0000

MATRIZ P
 1.0000 0.0000
 0.0000 1.0000

MATRIZ Q
 1.0000 0.0000
 0.0000 1.0000

MATRIZ R
 1.0000 0.0000
 0.0000 1.0000

MATRIZ S
 1.0000 0.0000
 0.0000 1.0000

CONDICIONES INICIALES DEL SISTEMA-----> TECLADO

TAMANIO DEL PASO Q1 = .1

TAMANIO DEL PASO Q2 = .1

TRAYECTORIAS INICIALES XI(0)

SISTEMA -----> 1

XI0 = (.5)

SISTEMA -----> 2

XI0 = (.7)

MATRIZ	P*
0.1830-	0.2562
0.0490	0.0686
0.0131-	0.0184
0.0035	0.0049
0.0009-	0.0012

MATRIZ	X
0.5000	0.7000
0.0327-	0.1183
0.0271	0.0186
0.0024-	0.0085
0.0019	0.0014
0.0009-	0.0012

MATRIZ	H*
- 0.8829-	0.2438
0.0693-	0.1013
- 0.0317-	0.0088
0.0050-	0.0073
- 0.0023-	0.0007

MATRIZ	II
- 0.1830	0.2562
- 0.0490-	0.0686
- 0.0131	0.0184
- 0.0035-	0.0049
- 0.0009	0.0012

MATRIZ	Z
0.6999	0.5000
- 0.1183	0.0327
0.0186	0.0271
- 0.0085	0.0024
0.0014	0.0019

$$J = [(X1(K))^2 \times P1 + (X1(K))^2 Q1 + U1^2 R1 + Z1^2 S1] / 2$$

LA FUNCION DE COSTO ES =

1.3087315

MATRIZ A

1.0000	1.0000
1.0000-	1.0000

MATRIZ B

1.0000	0.0000
0.0000	1.0000

MATRIZ C

1.0000	0.0000
0.0000	1.0000

MATRIZ P

1.0000	0.0000
0.0000	1.0000

MATRIZ Q

1.0000	0.0000
0.0000	1.0000

MATRIZ R

1.0000	0.0000
0.0000	1.0000

MATRIZ S

1.0000	0.0000
0.0000	1.0000

ESCOJA OPCION DE IMPRESION---->4.

CONDICIONES INICIALES DEL SISTEMA----> TECLADO

TAMANIO DEL PASO G1 = .2

TAMANIO DEL PASO G2 = .1

TRAYECTORIAS INICIALES XI(0)

SISTEMA -----> 1

XI0 = (4.2)

SISTEMA -----> 2

XI0 = (-2.5)

MATRIZ P*

1.5366	0.9146
0.4098-	0.2439
0.1024	0.0610

MATRIZ H*

0.9634-	5.1145
0.5402-	0.4415
0.0402-	0.1841

MATRIZ X

4.2000-	2.5000
0.6854	0.1305
0.1232-	0.1427
0.1024	0.0610

MATRIZ U

- 1.5366-	0.9146
- 0.4098	0.2439
- 0.1024-	0.0610

MATRIZ Z

- 2.5000	4.1999
- 0.1305	0.6854
- 0.1427	0.1232

$$J = [(X1(K))^2 * P1 + ((Y1(K))^2 * W1 + (U1)^2 * R1 + (Z1)^2 * S1)] / 2$$

LA FUNCION DE COSTO ES =

1368.8903

EJEMPLO 4

Sea el sistema descrito por las siguientes ecuaciones de diferencias:

$$\begin{pmatrix} X_1 (K + 1) \\ X_2 (K + 1) \\ X_3 (K + 1) \end{pmatrix} = \begin{pmatrix} -0.1 & 0.05 & 0.05 \\ 0 & -0.2 & 0 \\ 0.2 & -0.1 & -3 \end{pmatrix} \begin{pmatrix} X_1(K) \\ X_2 (K) \\ X_3 (K) \end{pmatrix} + \begin{pmatrix} 0.025 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_1(K) \\ u_2(K) \end{pmatrix}$$

con la siguiente función de costo:

$$J = \sum_{l=1}^3 1/2 \|X\|_{P_l}^2 + \sum_{k=0}^9 1/2 (\|X\|_Q^2 + \|U\|_{R_l}^2 + \|Z\|_{S_l}^2)$$

Donde las matrices de ponderación son las siguientes:

MATRIZ P

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

MATRIZ Q

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

MATRIZ R

$$\begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{pmatrix}$$

MATRIZ S

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Tenemos como valores iniciales de las trayectorias los siguientes valores:

$$X(0)^T = (\quad 2 \quad -3 \quad 4)$$

Tomemos como condiciones iniciales de los multiplicadores la Lagrange:

Los valores de las matrices cuyos elementos son cero es decir: p^* , λ^*

$$\underline{p}^* = \underline{0} ; \quad \underline{\lambda}^* = \underline{0}$$

Los resultados de la simulación digital podemos observar a continuación en las hojas del computador obtenidas.

Tomamos como valores iniciales del paso para los dos niveles 0.2 las trayectorias X convergen para los tres sistemas a cero y para un número relativamente pequeño de iteraciones pues para el nivel 1 - 2 el número de veces es de - 90 y para el nivel 2 - 3 es 39. Estos resultados son muy similares a los presentados por Jamshidi según muestra la copia del gráfico de las trayectorias obtenidas por este autor en un computador Hewlett Packard 968 (4).

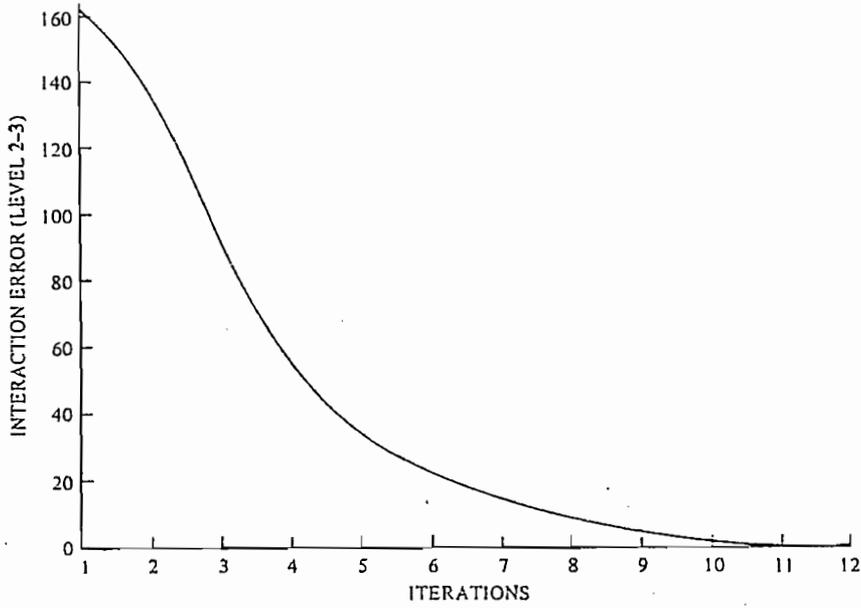
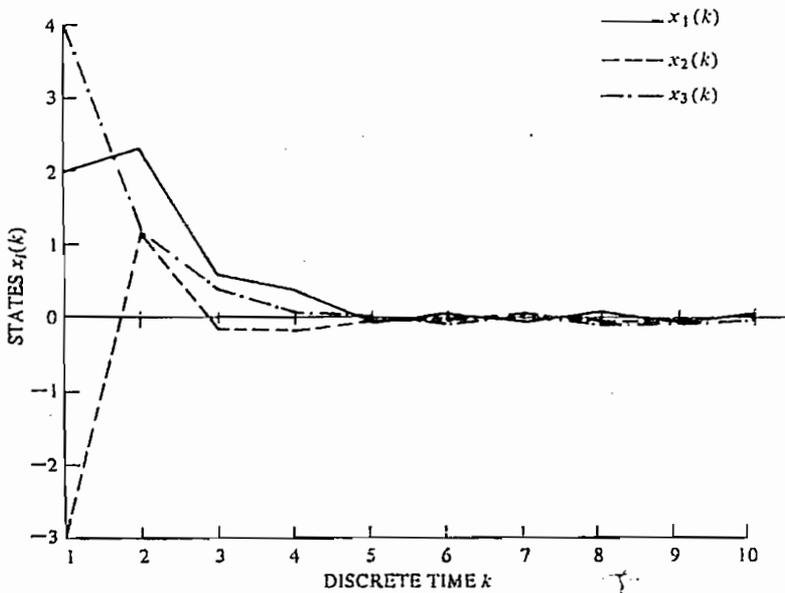


Figure 4.16 Typical interaction errors of Example 4.5.1.

Figure 4.17 Typical state trajectories for the third-order system of Example 4.5.1.



MATRIZ A

-	0.1000	0.0500	0.0500
	0.0000-	0.2000	0.0000
	0.2000-	0.1000-	3.0000

MATRIZ B

	0.2500	0.0000	0.0000
	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000

MATRIZ C

	1.0000	0.0000	0.0000
	0.0000	1.0000	0.0000
	0.0000	0.0000	1.0000

SISIFMA

KFS4

MATRIZ P

	2.0000	0.0000	0.0000
	0.0000	2.0000	0.0000
	0.0000	0.0000	2.0000

MATRIZ Q

	2.0000	0.0000	0.0000
	0.0000	2.0000	0.0000
	0.0000	0.0000	2.0000

MATRIZ R

	0.5000	0.0000	0.0000
	0.0000	0.5000	0.0000
	0.0000	0.0000	0.5000

MATRIZ S

	1.0000	0.0000	0.0000
	0.0000	1.0000	0.0000
	0.0000	0.0000	1.0000

```

CONDICIONES INICIALES DEL SISTEMA-----> RES4
TAMANO DEL PASO      G1 =                      .2
TAMANO DEL PASO      G2 =                      .2
                TRAYECTORIAS INICIALES      AT(0)
SISTEMA ----->                1
X10 = ( 2 )
SISTEMA ----->                2
X10 = (-3 )
SISTEMA ----->                3
X10 = ( 4 )
    
```

MATRIZ X

```

                2.0000-    3.0000    4.0000
-    0.132E    0.1982-    1.0691
    0.0213-    0.0131    0.2859
-    0.0053    0.0009-    0.0766
    0.0014-    0.0001    0.0205
-    0.0004    0.0000-    0.0055
    0.0001-    0.0000    0.0015
-    0.0000    0.0000-    0.0004
    0.0000-    0.0000    0.0001
-    0.0000    0.0000-    0.0000
    0.0000-    0.0000    0.0000
    
```

MATRIZ U

```

                0.0616    0.0000    0.0000
-    0.0019    0.0000    0.0000
    0.0001    0.0000    0.0000
-    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000
-    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000
-    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000
-    0.0000    0.0000    0.0000
    
```

MATRIZ Z

	0.0500-	0.0000	0.7001
-	0.0435	0.0000-	0.0464
	0.0130-	0.0000	0.0056
-	0.0038	0.0000-	0.0011
	0.0010-	0.0000	0.0003
-	0.0003	0.0000-	0.0001
	0.0001-	0.0000	0.0000
-	0.0000	0.0000-	0.0000
	0.0000-	0.0000	0.0000
-	0.0000	0.0000-	0.0000

MATRIZ F*

-	0.1232	0.4018-	10.9281
	0.0038-	0.0265	2.9282
-	0.0001	0.0010-	0.7846
	0.0000-	0.0001	0.2102
-	0.0000	0.0000-	0.0563
	0.0000-	0.0000	0.0151
-	0.0000	0.0000-	0.0040
	0.0000-	0.0000	0.0011
-	0.0000	0.0000-	0.0003
	0.0000-	0.0000	0.0001

MATRIZ H*

	0.0732-	0.4018	10.2280
	0.0398	0.0265-	2.8218
-	0.0135-	0.0010	0.7790
	0.0038	0.0001-	0.2091
-	0.0010-	0.0000	0.0560
	0.0003	0.0000-	0.0150
-	0.0001-	0.0000	0.0040
	0.0000	0.0000-	0.0011
-	0.0000	0.0000-	0.0003
	0.0000	0.0000-	0.0001

LA FUNCION DE COSTO ES = 3730.3151

NUMERO DE ITERACIONES ENTRE NIVEL 1 Y 2 = ---> 90

NUMERO DE ITERACIONES ENTRE NIVEL 2 Y 3 = ---> 39

EJEMPLO 5

Sea el sistema descrito por el ejemplo 4 la variación que vamos a realizar es cambiar el valor del paso de la siguiente forma:

$$A1 = 0.1$$

$$A2 = 0.1$$

Los resultados podemos observarlos en las siguientes hojas del computador - presentadas a continuación.

Al cambiar el tamaño del paso efectivamente el número de iteraciones aumenta, pues para el nivel 1-2 aumenta de 90 - 146 y para el nivel 2 - 3 aumenta de 39 a 80. El valor de la función de costo se mantiene aproximadamente - igual. Los valores de las trayectorias siguen convergiendo en igual forma que se observa en la copia del gráfico.

MATRIZ A

0.1000	0.0500	0.0500
0.0000-	0.2000	0.0000
0.2000-	0.1000-	3.0000

MATRIZ B

0.2500	0.0000	0.0000
0.0000	0.0000	0.0000
0.0000	0.0000	0.0000

MATRIZ C

1.0000	0.0000	0.0000
1.0000	1.0000	0.0000
0.0000	0.0000	1.0000

SISTEMA

KFS9

MATRIZ P

2.0000	0.0000	0.0000
0.0000	2.0000	0.0000
0.0000	0.0000	2.0000

MATRIZ Q

2.0000	0.0000	0.0000
0.0000	2.0000	0.0000
0.0000	0.0000	2.0000

MATRIZ R

0.5000	0.0000	0.0000
0.0000	0.5000	0.0000
0.0000	0.0000	0.5000

MATRIZ S

1.0000	0.0000	0.0000
0.0000	1.0000	0.0000
0.0000	0.0000	1.0000

```

CONDICIONES INICIALES DEL SISTEMA----->  PES9
TAMARCO DEL PASO      G1 =                .1
TAMARCO DEL PASO      G2 =                .1
          TRAYECTORIAS INICIALES          XT(0)
SISTEMA ----->                1
X10 = ( 2 )
SISTEMA ----->                2
X10 = (-3 )
SISTEMA ----->                3
X10 = ( 4 )

```

MATRIZ X

```

      2.0000-   3.0000   4.0000
-   0.1328    0.1982-   1.0091
      0.0213-   0.0131   0.2859
-   0.0053    0.0009-   0.0766
      0.0014-   0.0001   0.0205
-   0.0004    0.0000-   0.0055
      0.0001-   0.0000   0.0015
-   0.0000    0.0000-   0.0004
      0.0000-   0.0000   0.0001
-   0.0000    0.0000-   0.0000
      0.0000-   0.0000   0.0000

```

MATRIZ U

```

      0.0616    0.0000   0.0000
-   0.0019    0.0000   0.0000
      0.0001    0.0000   0.0000
-   0.0000    0.0000   0.0000
      0.0000    0.0000   0.0000
-   0.0000    0.0000   0.0000
      0.0000    0.0000   0.0000
-   0.0000    0.0000   0.0000
      0.0000    0.0000   0.0000
-   0.0000    0.0000   0.0000

```

	0.0500-	0.0000	0.7001
-	0.0435	0.0000-	0.0464
	0.0136-	0.0000	0.0056
-	0.0038	0.0000-	0.0011
	0.0010-	0.0000	0.0003
-	0.0003	0.0000-	0.0001
	0.0001-	0.0000	0.0000
-	0.0000	0.0000-	0.0000
	0.0000-	0.0000	0.0000
-	0.0000	0.0000-	0.0000

MATRIZ P*

-	0.1232	0.4018-	10.9283
	0.0038-	0.0265	2.9282
-	0.0001	0.0018-	0.7846
	0.0000-	0.0001	0.2102
-	0.0000	0.0000-	0.0563
	0.0000-	0.0000	0.0151
-	0.0000	0.0000-	0.0040
	0.0000-	0.0000	0.0011
-	0.0000	0.0000-	0.0003
	0.0000-	0.0000	0.0001

MATRIZ H*

	0.0732-	0.4018	10.2282
	0.0398	0.0265-	2.8818
	0.0135-	0.0018	0.7790
	0.0038	0.0001-	0.2091
-	0.0010-	0.0000	0.0560
	0.0003	0.0000-	0.0150
	0.0001-	0.0000	0.0040
	0.0000	0.0000-	0.0011
	0.0000-	0.0000	0.0003
	0.0000	0.0000-	0.0001

LA FUNCION DE COSTO ES =

3730.3163

NUMERO DE ITERACIONES ENTRE NIVEL 1 Y 2 = ---->

146

NUMERO DE ITERACIONES ENTRE NIVEL 2 Y 3 = ---->

80

EJEMPLO 6

Se ael sistema descrito por las ecuaciones del ejemplo 4 con la misma función de costo cuyas matrices de ponderación son las siguientes:

MATRIZ P

$$\begin{vmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{vmatrix}$$

MATRIZ Q

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

MATRIZ R

$$\begin{vmatrix} 0.7 & 0 & 0 \\ 0 & 0.7 & 0 \\ 0 & 0 & 0.7 \end{vmatrix}$$

MATRIZ S

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

... hojas del Computador

Los resultados conducen a un nuevo valor de la función de costo pues el valor de las matrices de ponderación cambiaron, lo que conduce a la solución de un problema totalmente distinto. Los valores de las trayectorias óptimas también convergen a cero.

MATRIZ A

0.1000	0.0500	0.0500
0.0000	0.2000	0.0000
0.2000	0.1000	3.0000

MATRIZ B

0.2500	0.0000	0.0000
0.0000	0.0000	0.0000
0.0000	0.0000	0.0000

MATRIZ C

1.0000	0.0000	0.0000
0.0000	1.0000	0.0000
0.0000	0.0000	1.0000

SISTEMA REFS

MATRIZ P

3.0000	0.0000	0.0000
0.0000	1.0000	0.0000
0.0000	0.0000	0.5000

MATRIZ Q

1.0000	0.0000	0.0000
0.0000	1.0000	0.0000
0.0000	0.0000	1.0000

MATRIZ R

0.7000	0.0000	0.0000
0.0000	0.7000	0.0000
0.0000	0.0000	0.7000

MATRIZ S

1.0000	0.0000	0.0000
0.0000	1.0000	0.0000
0.0000	0.0000	1.0000

```

CONDICIONES INICIALES DEL SISTEMA-----> RES5
TAMANO DEL PASO    G1 =                      .1
TAMANO DEL PASO    G2 =                      .1
                TRAYECTORIAS INICIALES      X(0)
SISTEMA ----->                               1
X(0) = ( 2 )
SISTEMA ----->                               2
X(0) = (-3 )
SISTEMA ----->                               3
X(0) = ( 4 )

```

MATRIZ X

	2.0000-	3.0000	4.0000
-	0.2504	0.2970-	1.1620
	0.0514-	0.0294	0.3494
-	0.0142	0.0029-	0.1035
	0.0042-	0.0003	0.0307
-	0.0012	0.0000-	0.0091
	0.0004-	0.0000	0.0027
-	0.0001	0.0000-	0.0008
	0.0000-	0.0000	0.0003
-	0.0000	0.0000-	0.0002
	0.0000-	0.0000	0.0003

MATRIZ H

	0.0343	0.0000	0.0000
-	0.0016	0.0000	0.0000
	0.0001	0.0000	0.0000
-	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
-	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
-	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
-	0.0000	0.0000	0.0000

MATRIZ 7

0.0500-	0.0000	0.7001
0.0443	0.0000-	0.0798
0.0160-	0.0000	0.0132
0.0050	0.0000-	0.0031
0.0015-	0.0000	0.0009
0.0005	0.0000-	0.0002
0.0001-	0.0000	0.0001
0.0000	0.0000-	0.0000
0.0000-	0.0000	0.0000
0.0000	0.0000-	0.0000

MATRIZ F*

0.0959	0.3030-	10.8129
0.0046-	0.0300	3.2087
0.0002	0.0030-	0.9521
0.0000-	0.0003	0.2825
0.0000	0.0000-	0.0238
0.0000-	0.0000	0.0249
0.0000	0.0000-	0.0074
0.0000-	0.0000	0.0022
0.0000	0.0000-	0.0006
0.0000-	0.0000	0.0002

MATRIZ H*

0.0459-	0.3030	10.1128
0.0397	0.0300-	3.1288
0.0158-	0.0030	0.9389
0.0050	0.0003-	0.2794
0.0015-	0.0000	0.0830
0.0005	0.0000-	0.0246
0.0001-	0.0000	0.0073
0.0000	0.0000-	0.0022
0.0000-	0.0000	0.0006
0.0000	0.0000-	0.0002

LA FUNCION DE COSTO ES = 243.17666
 NUMERO DE ITERACIONES ENTRE NIVEL 1 Y 2 = ---> 170
 NUMERO DE ITERACIONES ENTRE NIVEL 2 Y 3 = ---> 83

EJEMPLO 7

Sea el sistema del ejemplo 6 al que cambiamos el tamaño del paso. Los resultados se presentan a continuación:

Lo más destacado de esta solución es que el número de iteraciones varía. Conservándose la tendencia en los ejemplos anteriores en lo que se cambió el tamaño del paso.

MATRIZ A

0.1000	0.0500	0.0500
0.0000	0.2000	0.0000
0.2000	0.1000	3.0000

MATRIZ B

0.2500	0.0000	0.0000
0.0000	0.0000	0.0000
0.0000	0.0000	0.0000

MATRIZ C

1.0000	0.0000	0.0000
0.0000	1.0000	0.0000
0.0000	0.0000	1.0000

SISTEMA RES6

MATRIZ P

3.0000	0.0000	0.0000
0.0000	1.0000	0.0000
0.0000	0.0000	0.5000

MATRIZ D

1.0000	0.0000	0.0000
0.0000	1.0000	0.0000
0.0000	0.0000	1.0000

MATRIZ R

0.7000	0.0000	0.0000
0.0000	0.7000	0.0000
0.0000	0.0000	0.7000

MATRIZ S

1.0000	0.0000	0.0000
0.0000	1.0000	0.0000
0.0000	0.0000	1.0000

CONDICIONES INICIALES DEL SISTEMA-----> RES6
 TAMANIO DEL PASO G1 = .1
 TAMANIO DEL PASO G2 = .2
 TRAYECTORIAS INICIALES XI(0)

SISTEMA -----> 1

XI0 = (2)

SISTEMA -----> 2

XI0 = (-3)

SISTEMA -----> 3

XI0 = (4)

MATRIZ P*			MATRIZ H*				
-	0.0959	0.3030-	10.8131	0.0459-	0.3030	10.1130	
	0.0046-	0.0300	3.2087	0.0397	0.0300-	3.1289	
-	0.0002	0.0030-	0.9522	-	0.0150-	0.0030	0.9389
	0.0000-	0.0003	0.2825		0.0050	0.0003-	0.2794
-	0.0000	0.0000-	0.0838	-	0.0015-	0.0000	0.0830
	0.0000-	0.0000	0.0249		0.0005	0.0000-	0.0246
-	0.0000	0.0000-	0.0074	-	0.0001-	0.0000	0.0073
	0.0000-	0.0000	0.0022		0.0000	0.0000-	0.0022
-	0.0000	0.0000-	0.0006	-	0.0000-	0.0000	0.0006
	0.0000-	0.0000	0.0002		0.0000	0.0000-	0.0002

MATRIZ X

	2.0000-	3.0000	4.0000
-	0.2504	0.2970-	1.1820
	0.0514-	0.0294	0.3494
-	0.0142	0.0029-	0.1035
	0.0042-	0.0003	0.0307
-	0.0012	0.0000-	0.0091
	0.0004-	0.0000	0.0027
-	0.0001	0.0000-	0.0008
	0.0000-	0.0000	0.0003
-	0.0000	0.0000-	0.0002
	0.0000-	0.0000	0.0002

MAIRI7 U

	0.0347	0.0000	0.0000
-	0.0016	0.0000	0.0000
	0.0001	0.0000	0.0000
-	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
-	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
-	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
-	0.0000	0.0000	0.0000

MAIRI7 7

	0.0500-	0.0000	0.7001
-	0.0443	0.0000-	0.0798
	0.0160-	0.0000	0.0132
-	0.0050	0.0000-	0.0031
	0.0015-	0.0000	0.0009
-	0.0005	0.0000-	0.0002
	0.0001-	0.0000	0.0001
-	0.0000	0.0000-	0.0000
	0.0000-	0.0000	0.0000
-	0.0000	0.0000-	0.0000

LA FUNCION DE COSTO ES = 243.17653
 NUMERO DE ITERACIONES ENTRE NIVEL 1 Y 2 = ---> 110
 NUMERO DE ITERACIONES ENTRE NIVEL 2 Y 3 = ---> 40

EJEMPLO 8

Sea el sistema descrito por el ejemplo 6. Los valores que vamos a variar son las condiciones iniciales dadas por el vector de trayectorias $\underline{X}(K)$ para cuando $K = 0$. Entonces:

$$X(0)^T = (\quad 4 \quad -5 \quad 1.5)$$

Efectivamente el valor de J cambia y el número de iteraciones también, pero las trayectorias convergen al mismo valor presentados en los ejemplos 4, 5, 6, y 7.

Los resultados podemos observarlos a continuación en las hojas del computador que son presentadas.

MENU DE RESULTADOS

- 1.- LISTA DE OPCIONES
- 2.- SISTEMA EN MEMORIA DE LA CONSOLA
- 3.- SISTEMA EN ARCHIVO DE RESULTADOS
- 4.- SALIDA AL PROGRAMA PRINCIPAL

ESCOJA LA OPCION -----> 3

NOMBRE DEL ARCHIVO DE RESULTADOS ----> RES7
MENU DE IMPRESION DE RESULTADOS

SISTEMA EN EL ARCHIVO----> RES7

- 1.- LISTA DE OPCIONES
 - 2.- IMPRESION DEL SISTEMA Y MATRICES--> (A), (B), (C)
 - 3.- IMPRESION DE LAS MATRICES DE PONDERACION--> (P), (G), (M), (S)
 - 4.- CONDICIONES INICIALES
 - 5.- TRAYECTORIAS OPTIMAS
 - 6.- CONTROLES OPTIMOS
 - 7.- ESTADOS 2x1 OPTIMOS
 - 8.- FUNCION DE COSTO MINIMO
 - 9.- SALIDA AL MENU DE IMPRESION
- ESCOJA OPCION DE IMPRESION---->2
SISTEMA RES7

MATRIZ A

-	0.1000	0.0500	0.0500
	0.0000	0.2000	0.0000
	0.2000	0.1000	3.0000

MATRIZ B

	0.2500	0.0000	0.0000
	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000

MATRIZ C

	1.0000	0.0000	0.0000
	0.0000	1.0000	0.0000
	0.0000	0.0000	1.0000

SISTEMA RES7

MATRIZ P

	3.0000	0.0000	0.0000
	0.0000	1.0000	0.0000
	0.0000	0.0000	0.5000

MATRIZ D

	1.0000	1.0000	0.0000
	0.0000	1.0000	0.0000
	0.0000	0.0000	1.0000

MATRIZ R

	0.7000	0.0000	0.0000
	0.0000	0.7000	0.0000
	0.0000	0.0000	0.7000

MATRIZ S

	1.0000	0.0000	0.0000
	0.0000	1.0000	0.0000
	0.0000	0.0000	1.0000

CONDICIONES INICIALES DEL SISTEMA----> RES7

TAMANO DEL PASO G1 = .1

TAMANO DEL PASO G2 = .1

TRAYECTORIAS INICIALES XI(0)

SISTEMA ----> 1

X10 = (4)

SISTEMA ----> 2

X10 = (-5)

X10 = (4)
SISIFEA ----->

2

X10 = (-5)
SISIFEA ----->

3

X10 = (1.5)

PAIR17 X

	4.0000-	5.0000	1.5000
-	0.2437	0.4550-	0.4524
	0.0262-	0.0490	0.1210
-	0.0056	0.0045-	0.0289
	0.0010-	0.0005	0.0115
-	0.0005	0.0000-	0.0034
	0.0001-	0.0000	0.0010
-	0.0000	0.0000-	0.0003
	0.0000-	0.0000	0.0001
-	0.0000	0.0000-	0.0001
	0.0000-	0.0000	0.0001

PAIR17 H

	0.0000	0.0000	0.0000
-	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
-	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
-	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
-	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
-	0.0000	0.0000	0.0000

PAIR17 7

-	0.1150-	0.0000	1.3001
	0.0021	0.0000-	0.0483
	0.0041-	0.0000	0.0102
-	0.0017	0.0000-	0.0016
	0.0000-	0.0000	0.0004
-	0.0000	0.0000-	0.0001
	0.0001-	0.0000	0.0000
-	0.0000	0.0000-	0.0000
	0.0000-	0.0000	0.0000
-	0.0000	0.0000-	0.0000

PAIR17 F*

-	0.1519	0.5050-	4.0550
	0.0092-	0.0500	1.2033
-	0.0004	0.0045-	0.3571
	0.0000-	0.0005	0.1060
-	0.0000	0.0000-	0.0314
	0.0000-	0.0000	0.0093
-	0.0000	0.0000-	0.0028
	0.0000-	0.0000	0.0000
-	0.0000	0.0000-	0.0002
	0.0000-	0.0000	0.0001

PAIR17 H*

	0.3065-	0.5050	2.7549
-	0.0112	0.0500-	1.1050
	0.0037-	0.0045	0.3469
-	0.0017	0.0005-	0.1043
	0.0000-	0.0000	0.0311
-	0.0000	0.0000-	0.0092
	0.0001-	0.0000	0.0027
-	0.0000	0.0000-	0.0000
	0.0000-	0.0000	0.0000
-	0.0000	0.0000-	0.0000
	0.0000-	0.0000	0.0001

EJEMPLO 9

Sea el sistema descrito por las siguientes ecuaciones:

$$X_1 (K + 1) = - 0.1 X_1 (K) + 0.02 X_2 (K) + 0.1 X_3 (K) + 0.2 U_1$$

$$X_2 (K + 1) = 0.01 X_1 (K) - 0.2 X_2 (K)$$

$$X_3 (K + 1) = 0.1 X_1 (K) - X_3 (K) + 0.1 U_2 (K)$$

$$X_4 (K + 1) = - X_4 (K)$$

$$\text{Con: } J = \sum_{i=1}^4 \| X_i(5) \|_{P_i}^2 + \sum_{K=0}^4 \| X \|_{Q}^2 + \| U \|_{R}^2$$

Con los siguientes valores iniciales:

$$X_1 (0) = 0$$

$$X_2 (0) = 0$$

$$X_3 (0) = 0.5$$

$$X_4 (0) = 7$$

Los resultados podemos observar en las siguientes hojas.

TAMAÑO DEL PASO L1 = .2
 TAMAÑO DEL PASO L2 = .2
 TRAYECTORIAS INICIALES X(0)

SISTEMA -----> 1
 X(0) = (0)
 SISTEMA -----> 2
 X(0) = (0)
 SISTEMA -----> 3
 X(0) = (.5)
 SISTEMA -----> 4
 X(0) = (7)

PAIR 17		Y	
0.0000	0.0000	0.5000	7.0000
- 0.1651	0.0027-	0.1643-	2.6742
0.0629-	0.0010	0.0628	1.0225
- 0.0236	0.0004-	0.0242-	0.3932
0.0079-	0.0002	0.0047	0.1573
0.0000	0.0000-	0.0056-	0.0786
PAIR 12		U	
0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000
PAIR 17		Z	
0.7006-	0.0000	0.0000-	0.0001
- 0.2674-	0.0017-	0.0165-	0.0000
0.1022	0.0006	0.0003	0.0000
- 0.0393-	0.0002-	0.0024-	0.0000
0.0157	0.0001	0.0008	0.0000
MATRIX		PA	
0.0000	0.0000-	0.5090-	4.3252
0.0000	0.0000	0.1180	1.6516
0.0000	0.0000-	0.0449-	0.6292
0.0000	0.0000	0.0169	0.2355
0.0000	0.0000-	0.0056-	0.0786
PAIR 17		HA	
- 0.7000	0.0000	0.5090	4.3251
0.2674	0.0017-	0.1015-	1.6516
- 0.1022-	0.0006	0.0387	0.6291
0.0393	0.0002-	0.0145-	0.2355
- 0.0157-	0.0001	0.0048	0.0786

LA FUNCION DE COSTO ES = 1692.7314
 NUMERO DE ITERACIONES ENTRE NIVEL 1 Y 2 = ----> 13
 NUMERO DE ITERACIONES ENTRE NIVEL 2 Y 3 = ----> 30

4.2 CONCLUSIONES.-

En todos los resultados obtenidos en la simulación digital del método de Tamura podemos observar que efectivamente este método funciona, y mediante comparaciones realizadas con ejemplos típicos corridos en otros computadores podemos afirmar que los resultados son confiables.

La descentralización permite alcanzar el control óptimo de una manera más eficiente, y su efectividad aumenta para casos de ejemplos de sistemas cuyo orden es elevado, puesto que en muchas de las ocasiones los requerimientos de espacios de memoria así como de rapidez de ejecución conducirán necesariamente al uso de este algoritmo y de otros algoritmos de descentralización para la realización de control óptimo de aquellos sistemas.

Una de las acotaciones más importantes que debemos realizar en este trabajo es que la simulación digital se la hizo con el uso de un solo computador, por lo tanto, es necesario mencionar que no se está realizando un control óptimo descentralizado, si no que se hace la simulación del control óptimo distribuido.

La introducción del método de Coordinación tiene su razón de ser en función de entender porque se llega al algoritmo de Tamura. En realidad los métodos de coordinación pueden constituirse como cierto tipo de métodos clásicos para realizar el control óptimo.

Podemos afirmar que el método de Coordinación se presenta como un método de coordinación de objetivos ya que los mismos de cada uno de los subsistemas se encuentran coordinados en forma jerárquica por un -----

coordinador ubicado en un nivel superior al de los subsistemas.

Podemos decir que el método sirve para resolver problemas no lineales y lineales de sistemas de gran escala, es decir, de sistemas de un alto orden N , con términos cuadráticos, para facilitar la obtención de expresiones más sencillas de entender. Los requerimientos de almacenamiento no son un problema, y - más aún serían insignificantes si se tuviera la oportunidad de realizar el control óptimo distribuido, es decir usando varios computadores y microprocesadores, que realicen los cálculos de cada subsistema en forma individual.

Es más sencillo trabajar con ecuaciones algebraicas y de diferencias que con ecuaciones diferencias. En el método de Coordinación y mucho más en el método propuesto por Tamura, la resolución de las ecuaciones especialmente en el nivel más inferior, son sumamente sencillas. Si se pudiera hacer un estudio respecto al uso de las restricciones, así como también de las condiciones de borde, en realidad estas condiciones tampoco constituyen un obstáculo difícil de superar con estos métodos.

Sin embargo no podemos decir que el método del control jerárquico usando Tamura o el método de Coordinación tengan todas las ventajas ya que en realidad no sucede esto. Existen algunos tópicos no tratados en este trabajo puesto que es necesario realizar un estudio matemático profundo que nos permita entender el uso de ciertos términos introducidos tanto en la función de costo como en las restricciones respectivas. Uno de estos términos es:

$$Z^T S Z$$

que constituye un término cuadrático, y que el autor del algoritmo justifica

su utilización en base de la simulación digital realizada pero no colabora con un desarrollo matemática que justifique efectivamente su uso. En nuestros ejemplos, este término en algunas corridas fue eliminado y los resultados condujeron a una no convergencia del método, por lo tanto podemos afirmar que efectivamente este término es usado para evitar singularidades, es decir intervalos de no convergencia del método utilizado.

Uno de los problemas más importantes que tuve que resolver es el escogitamiento de la longitud del paso para la resolución de la actualización de los multiplicadores de Lagrange. En realidad no podemos aplicar un determinado método para calcular el tamaño del paso, ni siquiera el método presentado en el capítulo III es confiable, ya que en determinadas circunstancias, nos encontramos en intervalos de no convergencia, y el tamaño del paso no se ajustaba de ninguna forma para la resolución del problema. En realidad, debemos mencionar que en algunas ocasiones, fue preferible escoger un paso de valor constante, o a lo mucho que variara en forma linealmente descendente, para que un determinado ejemplo pudiera tener solución.

Se realizaron varias corridas con un paso determinado, y se escogió el que mejor cumplía con los requerimientos del problema. En los casos más generales, ningún valor en el tamaño del paso es confiable de ser utilizado para asegurar una convergencia en el método, y por lo mismo es necesario realizar algunas corridas del programa para encontrar el valor óptimo apropiado del valor de la longitud o tamaño del paso.

Otro de los problemas presentados en este trabajo, es el uso de un computador que utiliza terminales de tiempo compartido, y que es difícil determinar

ni el tiempo de ejecución fue largo en un determinado ejemplo. Muchas de las ocasiones, el tiempo de la ejecución fue largo, el número de iteraciones requeridas variaban con el tiempo de ejecución así como también con el tamaño del paso seleccionado inicialmente. Para sistemas de un alto orden, el tiempo de ejecución crecía sustancialmente.

El método utilizado no es confiable para efectos de control, puesto que los valores óptimos calculados del control, son usados al final de las iteraciones, puesto que es necesario determinar si la convergencia se produjo o no, y además - ver si cumple con la restricción que representa la combinación lineal de todos los estadores que inciden sobre un determinado subsistema.

Como habíamos mencionado, el término cuadrático

$$\| Z \|^2 = Z^T S Z$$

no representa una situación real.

Sin embargo a pesar de todas estas restricciones y anotaciones, en su forma conceptual, los métodos de descentralización son conceptualmente fáciles de comprender.

Todo lo que hemos mencionado respecto al método de Coordinación podemos hacerlo extensivo al método de Tamura, considerando que debemos tener en cuenta que existe un nivel adicional que guarda cierta analogía con el segundo nivel, hablando exclusivamente del escogitamiento del tamaño del paso en el

cálculo de la actualización de los multiplicadores de Lagrange.

Entre las ventajas del uso del algoritmo propuesto por Tamura, una de las principales es, que el nivel más inferior, el cálculo se reduce a la solución de ecuaciones triviales.

Uno de los problemas más importantes encontrados en la implementación de este algoritmo, es la falta de herramientas computacionales más poderosas que hubieran permitido realizar la implementación computacional en una forma más general, considerando la posibilidad de dimensionar los subsistemas en función de características físicas del sistema que representa una determinada planta, que sustituye una situación real física.

En los ejemplos corridos se consideraron varias alternativas, y es por esto que para ilustrar las bondades del método, se cambiaron los diferentes valores del tamaño del paso, de las condiciones iniciales, de los valores de las matrices de ponderación de la función de costo, etc. En algunas ocasiones el método no convergía por lo que fue necesario cambiar el tamaño del paso, para encontrar una solución aceptable al problema.

4.3 RECOMENDACIONES.-

Una vez que se ha completado con este trabajo que resulta pionero dentro de los trabajos de Control Óptimo descentralizado a nivel de la Facultad de Ingeniería Eléctrica, debemos decir que apenas es un "pequeño capítulo introductorio" de trabajos que serán realizados posteriormente.

Podríamos hacer una lista larga sobre los temas en los que será necesario iniciar una investigación, que permita el Área de Control y Sistemas de la Facultad tener una información actualizada de tópicos que se están tratando a nivel mundial y que tienen su aplicación en la solución de problemas que involucran un gran número de variables, un tiempo de ejecución mínimo, así como también soluciones confiables, que permita realizar un control también confiable, pero solo mencionaremos algunos temas que no se pudieron tratar en esta tesis entendiendo sobre todo al nivel de complejidad de la misma, y a su carácter de trabajo primerizo.

Entre estos tópicos que podríamos mencionar están:

- La implantación de un programa que considere no solamente la descentralización en forma diagonal y unidimensional, sino que permita la introducción de las dimensiones de los subsistemas a gusto y criterio del diseñador.
- El estudio detallado de otro tipo de algoritmos de descentralización para realizar control óptimo de sistemas lineales.

- La implementación del Control Computacional Distribuido, mediante el uso de varios computadores, o minicomputadores, realizando para ello un estudio de las comunicaciones entre computadores o minicomputadores trabajando en paralelo, para lo cual es recomendable conocer profundamente la arquitectura no solo de los computadores, sino de los microprocesadores.

Un estudio más profundo de algoritmos, de actualización de valores de parámetros, utilizando para ello toda la teoría que se pueda encontrar en métodos numéricos, tópicos tales como: rapidez de ejecución, estabilidad de los algoritmos, introducción de términos que permitan asegurar la solución de no singularidades o evitarlas si es que se produjera, etc.

BIBLIOGRAFIA

1. Sing and A. Titli; Systems: Descomposition, optimization and control.- Pergamon Press, Oxford England 1978.
2. Tamura Hiroyuki: Decentralized optimization for Distributed lag Models of Discrete Systems; IFACJ Automática 11:593-602
3. Sandell N.R. Jr.; Varaiya P., Athans, M.: and Salonoo. N.6. Survey of decentralized control methods for lange scale systems, IEEE T A C AC - 23:108-128
4. Jamshidi Mohammad. Large Scale Systems: Modeling and Control, North Holland Publishing Co. New York 1983
5. Singh M. 6. Decentralised Control Noth Holland Publishing Co. New York 1981
6. DATA GENERAL 450 System, Appendix E Statement, Command and Function Summary for Basic.

A P E N D I C E A

MANUAL DE USO DEL PROGRAMA

Manual de utilización del programa.-

La disposición física del programa permite una fácil identificación de todas las partes contempladas en el enunciado del problema y básicamente en la estructura del algoritmo. Los objetivos de la tesis fueron propuestos en el capítulo introductorio, y que fueron satisfechos en su totalidad en los capítulos II, III, y IV. El programa consta de tres partes, como se describió en el capítulo III, y la ejecución del mismo resulta fácil y sencillo.

Para utilizar el programa es necesario conocer la lista de variables utilizadas y que en la ejecución de ejemplos será necesario mencionar.

Estas variables son las siguientes:

N = número de estados o trayectorias del sistema.

También representa el número de Subsistemas.

K = número de estados temporales dada por la función de costo.

$A1$ = Tamaño del paso para actualizar los multiplicadores de Lagrange p^* .

$A2$ = Tamaño del paso para actualizar los multiplicadores de Lagrange λ^* .

A = Matriz de estados del sistema

B = Matriz de controles del sistema

C = Matriz de estados de interacción del sistema

P, Q, R, S = Matrices de ponderación de la función de costo.

H, P^* = Matrices de multiplicadores de Lagrange inicial.

X = Matriz de trayectorias del sistema

U = Matriz de controles del sistema

Z = Matriz de estado de interacción entre los subsistemas.

Una vez que se conoce las variables, se procede a la ejecución del programa.

Es necesario, debido a que se utiliza un computador de tiempo compartido, - pedir la asignación de tiempo de consola o terminal. Ya en el terminal se procede de la siguiente forma:

1. USERNAME Corresponde a la primera parte de su clave de entrada.
2. PASSWORD Corresponde a la segunda parte de su clave de entrada.
3. BASIC Ingresa a la edición del compilador BASIC del computador.

En este momento es necesario chequear los archivos que tiene almacenado en el disco de su directorio, para lo cual introduce el comando FILE.

Chequeado de esta forma el contenido de su directorio, es necesario cargar - a la memoria de imagen de la consola, el programa correspondiente a la simulación del algoritmo de Tamura, para lo cual se procede de la siguiente forma:

```
NEW "TAMURA" (NEWLINE)
```

A continuación se procede a correr el programa con la instrucción RUN.

El programa es de tan fácil manejo debido al listado de instrucciones claras que permiten una ejecución sin problemas.

Ejemplo:

NEW "TAMURA

RUN

MENU PRINCIPAL

1. LISTA DE OPCIONES
2. INGRESO DE DATOS
3. OPTIMIZACION USANDO EL ALGORITMO DE TAMURA
4. RESULTADOS

ESCOJA OPCION —————> 2

Aparece luego el listado de opciones del ingreso de datos que permiten una serie de alternativas fáciles de ejecutar:

INGRESO DE DATOS

1. LISTA DE OPCIONES
2. INGRESO POR TECLADO
3. VERIFICACION Y CORRECCION DE ERRORES
4. GRABACION EN ARCHIVO
5. NUEVOS VALORES EN LAS MATRICES DE PONDERACION
6. NUEVOS VALORES EN EL TAMAÑO DEL PASO
7. NUEVOS VALORES EN LAS CONDICIONES INICIALES
8. SALIDA AL PROGRAMA PRINCIPAL

ESCOJA OPCION —————> 2

Aparece inmediatamente las variables a ser ingresadas.

Debemos mencionar que cada ingreso tiene su mensaje en pantalla de tal forma que es difícil cometer errores en el ingreso, aunque es factible realizar tal ingreso y corrección de errores, volviendo al menú de ingreso de datos y proceder a corregir los errores mediante la opción.

3. Si se quiere grabar en archivo se coloca el nombre del archivo y se espera hasta que el sistema grave sus datos en la memoria del disco o de la cinta, según el caso; pero para ello necesariamente deberán existir datos en la memoria de la consola.

Si el ingreso de datos ha terminado, es necesario regresar al programa principal para poder continuar o escoger otras opciones presentes en el Menú.

Si la opción escogida tiene que ver con la ejecución del algoritmo, aparecerán otras opciones en la ejecución, y será necesario escoger alguna de ellas, como por ejemplo, si el sistema a calcularse está presente en la memoria de la consola o si se gravó en archivo.

Para obtener resultados es necesario ir al Menú de resultados, el mismo que presenta otra lista de opciones, que en una forma completa contempla los requerimientos de información del sistema.

En el siguiente apéndice se presentan los listados del programa.

APENDICE B

LISTADO DEL PROGRAMA

```

0001 REP
0002 REP
0003 PRINT "<12>"
0004 REP
0005 REP
0006 PRINT "*****"
0007 PRINT "*****"
0008 PRINT "*****"
0009 PRINT "*****"
0010 PRINT "***** ESCUELA POLITECNICA NACIONAL *****"
0011 PRINT "*****"
0012 PRINT "*****"
0013 PRINT "***** FACULTAD DE INGENIERIA ELECTRICA *****"
0014 PRINT "*****"
0015 PRINT "***** DEPARTAMENTO DE ELECTRONICA Y CONTROL *****"
0016 PRINT "*****"
0017 PRINT "*****"
0018 PRINT "*****"
0019 PRINT "*****"
0020 PRINT "*****"
0021 PRINT "*****"
0022 PRINT
0023 INPUT " PRESIONE ( NEW LINE ) TECLA PARA CONTINUAR,";CS
0024 IF CS<>"XXXXXX" THEN GOTO 0025
0025 PRINT
0026 PRINT "<12> OPTIMIZACION DE SISTEMAS LINEALES DISCRETOS DINAMICOS"
0027 PRINT
0028 PRINT " CON FUNCIONES DE COSTO CUADRATICO"
0029 PRINT
0030 PRINT " REALIZADO POR : MARCO VINICIO ORTIZ TIRADO ,"
0031 PRINT
0032 INPUT " PRESIONE ( NEW LINE ) TECLA PARA CONTINUAR,";CS
0033 PRINT "<12>"
0034 PRINT " A T E N C I O N"
0035 PRINT "ESTE PROGRAMA REALIZA EL CALCULO DE VALORES OPTIMOS TANTO DE CONTROL COMO DE ."
0036 PRINT "ESTADO QUE MINIMIZA LA FUNCION DE COSTO CUADRATICA -- J -- . PARA RESOLVER"
0037 PRINT "LOS PROBLEMAS SE PRESENTAN UNA SERIE DE ALTERNATIVAS QUE PODRAN SER ESCOGIDAS ."
0038 PRINT "DE ACUERDO CON EL CRITERIO DEL USUARIO DEL PROGRAMA, PARA LO CUAL ES NECESARIO ."
0039 PRINT "TENER EN CUENTA LAS SIGUIENTES INDICACIONES:"
0040 PRINT "1.- DEBE ESCOGERSE EL SISTEMA CUYO ORDEN N SEA EL MAS GRANDE DE TODOS LOS ."
0041 PRINT "SISTEMAS CARGADOS EN LA MEMORIA DEL COMPUTADOR. EVITAR EL ERROR DE REDIMENSIONAMIENTO MAS GRANDE"
0042 PRINT "2.- CUIDE QUE LOS VALORES EN LOS PORCENTAJES DE CAMBIO SEA DE UN VALOR COMPROMETIDO ENTRE 0 Y 1 ."
0043 PRINT "3.- DEBE TOMAR EN CUENTA QUE EL PROGRAMA REALIZA UNA OPCION A LA VEZ. SI DESEA ."
0044 PRINT "REALIZAR CUALQUIERA DE LAS TRES OPCIONES DEBE REGRESAR AL PROGRAMA PRINCIPAL ."
0045 PRINT "4.- DEBEN EXISTIR SIEMPRE DATOS PARA REALIZAR CUALQUIER OPCION. CASO CONTRARIO ."
0046 PRINT "DEBE REGRESAR AL MENU DE INGRESO DE DATOS DEL PROGRAMA PRINCIPAL ."
0047 PRINT "5.- SI DESEA IMPRIMIR VARIOS SISTEMAS Y ALGUNO DE ELLOS SOBREPASA EL DIMENSIONAMIENTO DE N=12 EL FORMATO DE IMPRESION NO ALCANZA NI EN LA PANTALLA ."
0048 PRINT "NI EN EL PAPEL, Y LO MEJOR SERA OBTENER LOS DATOS EN FORMA INDIVIDUAL ."
0049 PRINT "REALIZAR LAS SIGUIENTES OPERACIONES:"
0050 PRINT "A.- MAT PRINT A---> IMPRIME UNA MATRIZ <A> FILA POR FILA"
0051 PRINT "B.- PRINT X---> IMPRIME LA VARIABLE X"
0052 INPUT " PRESIONE ( NEW LINE ) TECLA PARA CONTINUAR,";CS
0053 IF CS<>"XXXXXX" THEN GOTO 0054
0054 PRINT
0055 PRINT "<12> M E N U P R I N C I P A L ."
0056 PRINT
0057 PRINT
0058 PRINT
0059 PRINT "1.- MENU PRINCIPAL"
0060 PRINT
0061 PRINT
0062 PRINT
0063 PRINT

```

```

LISTADO DEL PROGRAMA 102
0064 PRINT "2.- INGRESO DE DATOS"
0065 PRINT
0066 PRINT "3.- OPTIMIZACION USANDO ALGORITMO DE TABUFA"
0067 PRINT
0068 PRINT "4.- RESULTADOS"
0069 PRINT
0070 PRINT "5.- FINALIZACION ."
0071 PRINT
0072 INPUT " ESCOJA LA OPCION-----> ", B3
0073 IF B3="1" THEN GOTO 0059
0074 IF B3="2" THEN GOTO 0545
0075 IF B3="3" THEN GOTO 0545
0076 IF B3="4" THEN GOTO 0722
0077 IF B3="5" THEN GOTO 0980
0078 GOTO 0059
0079 REM
0080 REM      DESACOPLAMIENTO TOTAL
0081 REM
0082 REM
0083 REM SUBROUTINA DE DESACOPLAMIENTO TOTAL
0084 REM
0085 PRINT "<12>"
0086 INPUT " ORDEN DEL SISTEMA Y DEL NUMERO DE SUBSISTEMAS N= ", N
0087 PRINT
0088 PRINT
0089 INPUT " NUMERO DE ESTADOS DE CADA SUBSISTEMA .          K= ", K1
0090 PRINT
0091 PRINT
0092 INPUT "<12> PORCENTAJE DE VARIACION DE LA FUNCION DE COSTO Q1= ", A1
0093 PRINT
0094 PRINT
0095 INPUT "PORCENTAJE DE VARIACION DE LA FUNCION DE COSTO Q2= ", A2
0096 PRINT
0097 INPUT "SI ESTA EN EL MENU DE DATOS COLOQUE <SI> SI EMPIEZA LA INTRODUCCION APLASTE (REM LINE 1)",
0098 IF B5="SI" THEN GOTO 0545
0099 REM DIMENSIONAMIENTO DE LAS MATRICES DE DATOS
0100 DIM A(N,K), D(N,N), C(N,K), P(N,N), L(N,K), R(N,N), S(K,K), H(K1,K), T(K1,K)
0101 GOTO X(K1+1,N), H9(K1,N), H0(N,K1)
0102 PRINT "<12> INGRESE LAS MATRICES POR FILAS"
0103 PRINT
0104 MAT INPUT " MATRIZ A          ", A
0105 MAT INPUT " MATRIZ B          ", B
0106 MAT INPUT " MATRIZ C          ", C
0107 PRINT "<12>"
0108 MAT INPUT " MATRIZ P          ", P
0109 MAT INPUT " MATRIZ D          ", D
0110 MAT INPUT " MATRIZ R          ", R
0111 MAT INPUT " MATRIZ S          ", S
0112 INPUT "SI ESTA EN EL MENU DE DATOS COLOQUE <SI> SI EMPIEZA LA INTRODUCCION APLASTE (REM LINE 1)",
0113 IF B5="SI" THEN GOTO 0545
0114 GOTO 0214
0115 REM
0116 REM      SUBROUTINA DE CORRECCION DE ERRORES
0117 PRINT "<12>"
0118 PRINT
0119 PRINT "          MATRIZ  A"
0120 PRINT
0121 PRINT
0122 FOR I=1 TO N
0123   PRINT TAB(X9);
0124   FOR J=1 TO N
0125     PRINT USING "-####.####", A(I,J);
0126   NEXT J

```

LISTADO DEL PROGRAMA 10B

```
0127 PRINT
0128 PRINT
0129 NEXT I
0130 GOSUB 0473
0131 PRINT "<12>"
0132 PRINT
0133 PRINT "          MATRIZ B."
0134 PRINT
0135 PRINT
0136 FOR I=1 TO N
0137 PRINT TAB(X9);
0138 FOR J=1 TO N
0139 PRINT USING "-####.####",B(I,J);
0140 NEXT J
0141 PRINT
0142 PRINT
0143 NEXT I
0144 GOSUB 0473
0145 PRINT "<12>"
0146 PRINT
0147 PRINT "          MATRIZ C."
0148 PRINT
0149 PRINT
0150 FOR I=1 TO N
0151 PRINT TAB(X9);
0152 FOR J=1 TO N
0153 PRINT USING "-####.####",C(I,J);
0154 NEXT J
0155 PRINT
0156 PRINT
0157 NEXT I
0158 GOSUB 0473
0159 PRINT "<12>"
0160 PRINT
0161 PRINT "          MATRIZ P."
0162 PRINT
0163 FOR I=1 TO N
0164 PRINT TAB(X9);
0165 FOR J=1 TO N
0166 PRINT USING "-####.####",P(I,J);
0167 NEXT J
0168 PRINT
0169 PRINT
0170 NEXT I
0171 GOSUB 0473
0172 PRINT "<12>"
0173 PRINT "          MATRIZ Q."
0174 PRINT
0175 PRINT
0176 FOR I=1 TO N
0177 PRINT TAB(X9);
0178 FOR J=1 TO N
0179 PRINT USING "-####.####",Q(I,J);
0180 NEXT J
0181 PRINT
0182 PRINT
0183 NEXT I
0184 GOSUB 0473
0185 PRINT "<12>"
0186 PRINT
0187 PRINT "          MATRIZ R."
0188 PRINT
0189 PRINT
```

LISTADO DEL PROGRAMA 104

```

0190 FOR I=1 TO N
0191   PRINT TAB(X9);
0192   FOR J=1 TO N
0193     PRINT USING "-####.####",R(I,J);
0194   NEXT J
0195   PRINT
0196   PRINT
0197 NEXT I
0198 GOSUB 0473
0199 PRINT "<12>"
0200 PRINT
0201 PRINT "
0202 PRINT
0203 PRINT
0204 FOR I=1 TO N
0205   PRINT TAB(X9);
0206   FOR J=1 TO N
0207     PRINT USING "-####.####",S(I,J);
0208   NEXT J
0209   PRINT
0210   PRINT
0211 NEXT I
0212 GOSUB 0473
0213 GOTO 0545
0214 PRINT "<12>"
0215 MAT INPUT "MATRIZ DE LAMNDAS INICIALES " ,H
0216 MAT INPUT "MATRIZ DE MULTIPLICADORES P* " ,I
0217 PRINT "<12>"
0218 FOR Y=1 TO N
0219   PRINT
0220   PRINT " PARA EL SUBSISTEMA " ,I
0221   PRINT
0222   PRINT
0223   PRINT
0224   INPUT " X 0 = ? " ,X(1,I)
0225   PRINT
0226 NEXT I
0227 GOTO 0545
0228 PRINT "<12>"
0229 PRINT
0230 PRINT
0231 PRINT " ESTOY CALCULANDO LOS VALORES OPTIMOS DE LAS TRAYECTORIAS "
0232 PRINT
0233 PRINT " OPTIMAS ( X ); DE LOS CONTROLES ( U ) Y DE LAS REALIMENTA "
0234 PRINT
0235 PRINT " CTOMES DE LOS ESTADOS ( Z ). !!!!! FAVOR ESPERAR!!!!!"
0236 PRINT
0237 IF N<>0 THEN GOTO 0240
0238 INPUT "<12> NO EXISTEN DATOS PRESIONE (NEW LINE) PARA CONTINUAR " ,BS
0239 GOTO 0545
0240 REM
0241 REM DIFERENCIALAMIENTO DE L(I,J)
0242 REM
0243 DIM L(N,K),A9(N)
0244 FOR J=1 TO N
0245   FOR I=1 TO N
0246     IF I=J THEN GOTO 0249
0247     LET L(I,J)=A(1,J)
0248     GOTO 0251
0249     LET L(I,J)=0
0250     LET A9(I)=A(1,I)
0251   NEXT I
0252 NEXT J

```

LISTADO DEL PROGRAMA 105

```

0253 REM
0254 REM CALCULO DE LOS VALORES OPTIMOS DE ----> X ---- U ---- Z
0255 REM
0256 LET Y/=0
0257 LET Y8=0
0258 LET Y5=1
0259 LET I9=1
0260 LET I8=1
0261 FOR I=1 TO N
0262 REM
0263 REM PARA K= 0 CALCULO DE LOS U(I) Y Z(I)
0264 REM
0265 DIM Z(K1,N),U(K1,N)
0266 LET U(I,1)=-1*(1/R(I,I))*H(I,I)*I(I,1)
0267 LET Z(I,I)=-1*(1/S(I,I))*(C(I,I)*T(I,I)+H(I,I))
0268 REM
0269 REM PARA-K = 1 TO K-1
0270 REM
0271 FOR K=2 TO K1
0272 LET S9=0
0273 FOR J=1 TO N
0274 LET S9=S9+L(I,J)*H(K,J)
0275 RLX1 J
0276 LET X(K,I)=-1*(1/Q(I,I))*(A9(I)*T(K,I)-T(K-1,I)-S9)
0277 LET U(K,I)=-1*(1/R(I,I))*B(I,I)*I(K,I)
0278 LET Z(K,I)=-1*(1/S(I,I))*(C(I,I)*T(K,I)+H(K,I))
0279 NEXT K
0280 REM
0281 REM PARA K = K
0282 REM
0283 LET X(K+1,I)=(1/P(I,I))*T(K,I)
0284 NEXT I
0285 DIM M(K1,N)
0286 IF Y5=1 THEN GOTO 0291
0287 GOTO 0303
0288 REM
0289 REM CALCULO DEL GRADIENTE PARA UN P* DADO
0290 REM
0291 FOR I=1 TO N
0292 FOR K=1 TO K1
0293 LET F(K,I)=-1*X(K+1,I)+A9(I)*X(K,I)+B(I,I)*U(K,I)+C(I,I)*Z(K,I)
0294 RLX1 K
0295 FOR K=1 TO K1
0296 IF M(K,I)^2>.00000001 THEN GOTO 0393
0297 IF ABS(M(K,I))>1000 THEN GOTO 0534
0298 NEXT K
0299 LET I8=1
0300 NEXT I
0301 LET Y5=Y5+1
0302 REM
0303 REM CALCULO DEL GRADIENTE PARA UN LAMDA DADO
0304 REM
0305 DIM F1(K1,N)
0306 FOR I3=1 TO N
0307 FOR K=1 TO K1
0308 LET S8=0
0309 FOR J=1 TO N
0310 LET S8=S8+L(I3,J)*X(K,J)
0311 NEXT J
0312 LET F1(K,I3)=Z(K,I3)-S8
0313 NEXT K
0314 NEXT I3
0315 FOR I3=1 TO N

```

LISTADO DEL PROGRAMA 106

```

0316 FOR K4=1 TO K1
0317   IF M1(K4,I3)^2>.00000001 THEN GOTO 0421
0318   IF ABS(M1(K4,I3))>1000 THEN GOTO 0539
0319 NEXT K4
0320 LET I9=1
0321 NEXT I3
0322 GOTO 0575
0323 PRINT "<12>"
0324 PRINT
0325 PRINT "                                MATRIZ  X"
0326 PRINT
0327 PRINT
0328 FOR I=1 TO K1+1
0329   PRINT TAB(X9);
0330   FOR J=1 TO N
0331     PRINT USING "-####.####",X(I,J);
0332   NEXT J
0333   PRINT
0334   PRINT
0335 NEXT I
0336 INPUT "PRESIONE [ NEW LINE ] TECLA PARA CONTINUAR",AS
0337 IF AS<>"XXXX" THEN GOTO 0338
0338 GOTO 0749
0339 PRINT "<12>"
0340 PRINT "                                MATRIZ  U"
0341 PRINT
0342 FOR I=1 TO K1
0343   PRINT TAB(X9);
0344   FOR J=1 TO N
0345     PRINT USING "-####.####",U(I,J);
0346   NEXT J
0347   PRINT
0348   PRINT
0349 NEXT I
0350 INPUT "PRESIONE [ NEW LINE ] TECLA PARA CONTINUAR",AS
0351 IF AS<>"XXXX" THEN GOTO 0366
0352 GOTO 0749
0353 PRINT "<12>"
0354 PRINT "                                MATRIZ  Z"
0355 PRINT
0356 FOR I=1 TO K1
0357   PRINT TAB(X9);
0358   FOR J=1 TO N
0359     PRINT USING "-####.####",Z(I,J);
0360   NEXT J
0361   PRINT
0362   PRINT
0363 NEXT I
0364 INPUT "PRESIONE [ NEW LINE ] TECLA PARA CONTINUAR",AS
0365 IF AS<>"XXXX" THEN GOTO 0366
0366 GOTO 0749
0367 PRINT "<12>"
0368 PRINT "                                MATRIZ  P*"
0369 FOR I=1 TO K1
0370   PRINT TAB(X9);
0371   FOR J=1 TO N
0372     PRINT USING "-####.####",T(I,J);
0373   NEXT J
0374   PRINT
0375   PRINT
0376 NEXT I
0377 INPUT "PRESIONE [ NEW LINE ] TECLA PARA CONTINUAR",AS
0378 IF AS<>"XXXX" THEN GOTO 0379

```

LISTADO DEL PROGRAMA 107

```

0379 PRINT "<12>"
0380 PRINT "
0381 PRINT
0382 FOR I=1 TO K1
0383 PRINT TAB(X9);
0384 FOR J=1 TO N
0385 PRINT USING "-####.####",H(I,J);
0386 NEXT J
0387 PRINT
0388 PRINT
0389 NEXT I
0390 INPUT "PRESIONE [ NEW LINE ] TECLA PARA CONTINUAR",AS
0391 IF AS<>"XXXX" THEN GOTO 0392
0392 GOTO 0749
0393 REM
0394 REM          SUBROUTINA PARA EL MEJORAMIENTO DE P
0395 REM
0396 DIM D(K1),M2(K1,N)
0397 LET Y8=Y8+1
0398 IF J8=1 THEN GOTO 0400
0399 GOTO 0409
0400 FOR I4=1 TO N
0401 FOR K2=1 TO K1
0402 LET D(K2)=M(K2,I4)
0403 LET T(K2,I4)=T(K2,I4)+A1*D(K2)
0404 NEXT K2
0405 NEXT I4
0406 PA) M2=M
0407 LET I8=I8+1
0408 GOTO 0261
0409 LET S7=0
0410 LET S6=0
0411 FOR K2=1 TO K1
0412 LET S7=S7+M(K2,1)^2
0413 LET S6=S6+M2(K2,1)^2
0414 LET S5=S7/S6
0415 LET D(K2)=M(K2,1)+S5*M2(K2,1)
0416 GOSUB 0520
0417 LET T(K2,1)=I(K2,1)+A1*D(K2)
0418 LET M2(K2,J)=N(K2,I)
0419 NEXT K2
0420 GOTO 0261
0421 REM
0422 REM
0423 REM          SUBROUTINA PARA EL MEJORAMIENTO DE LAMDA
0424 REM
0425 LET Y7=Y7+1
0426 DIM D1(K1),M3(K1,N)
0427 IF I9=1 THEN GOTO 0429
0428 GOTO 0439
0429 FOR I2=1 TO N
0430 FOR K2=1 TO K1
0431 LET D1(K2)=M1(K2,I2)
0432 GOSUB 0527
0433 LET H(K2,I2)=A2*D1(K2)+H(K2,I2)
0434 NEXT K2
0435 NEXT I2
0436 LET I9=I9+1
0437 PA) M3=M1
0438 GOTO 0261
0439 FOR I5=1 TO N
0440 LET S4=0
0441 LET S3=0

```

PAIRIZ H*"

```

LISTADO DEL PROGRAMA 108
0442 FOR K2=1 TO K1
0443 LEI S4=S4+M1(K2,15)^2
0444 LEI S3=S3+M3(K2,15)^2
0445 LET S2=S4/S3
0446 LEI D1(K2)=M1(K2,15)+S2*M3(K2,15)
0447 LET H(K2,15)=H(K2,15)+A2*D1(K2)
0448 LET M3(K2,15)=M1(K2,15)
0449 NEXT K2
0450 NEXT I5
0451 GOTO 0261
0452 REM
0453 REM          CALCULO DE LA FUNCION DE COSTO CON LOS VALORES OPTIMOS
0454 REM
0455 PRINT "<12> J=( XI(K)^2*PI +((XI(K)^2*NI+UI^2*RI+Z1^2*SI))/2"
0456 FOR I=1 TO N
0457 FOR K=1 TO K1
0458 LEI L1=L1+X(K,I)^2*Q(I,I)+U(K,I)^2*R(I,I)+Z(K,I)^2*S(I,I)
0459 NEXT K
0460 LET L2=.5*(L1+X(K1+1,I))^2+P(I,I)
0461 NEXT I
0462 RETURN
0463 PRINT
0464 PRINT " LA FUNCION DE COSTO ES = ",L2
0465 PRINT
0466 PRINT "NUMERO DE ITERACIONES ENTRE NIVEL 1 Y 2 = ---> ",Y8
0467 PRINT
0468 PRINT "NUMERO DE ITERACIONES ENTRE NIVEL 2 Y 3 = ---> ",Y7
0469 PRINT
0470 INPUT "PRESTONE ( NEW LINE ) TECLA PARA CONTINUAR",AS
0471 IF AS<>"XXXX" THEN GOTO 0472
0472 GOTO 0749
0473 REM
0474 PRINT
0475 INPUT " EXISTE ALGUN ERROR EN LA MATRIZ ? (SI/ (NEW LINE) ) ",XS
0476 IF XS="SI" THEN GOTO 0478
0477 RETURN
0478 INPUT " INGRESE LA FILA ",L4
0479 INPUT " INGRESE LA COLUMNA ",L5
0480 INPUT " INGRESE LA MATRIZ A CONREGIRSE -----> ",B3
0481 IF B3="A" THEN GOTO 0488
0482 IF B3="B" THEN GOTO 0490
0483 IF B3="C" THEN GOTO 0492
0484 IF B3="P" THEN GOTO 0494
0485 IF B3="Q" THEN GOTO 0496
0486 IF B3="R" THEN GOTO 0498
0487 IF B3="S" THEN GOTO 0500
0488 INPUT " INGRESE EL VALOR DEL ELEMENTO = ",A(L4,L5)
0489 GOTO 0117
0490 INPUT " INGRESE EL VALOR DEL ELEMENTO = ",H(L4,L5)
0491 GOTO 0131
0492 INPUT " INGRESE EL VALOR DEL ELEMENTO = ",C(L4,L5)
0493 GOTO 0145
0494 INPUT " INGRESE EL VALOR DEL ELEMENTO = ",P(L4,L5)
0495 GOTO 0159
0496 INPUT " INGRESE EL VALOR DEL ELEMENTO = ",R(L4,L5)
0497 GOTO 0172
0498 INPUT " INGRESE EL VALOR DEL ELEMENTO = ",R(L4,L5)
0499 GOTO 0185
0500 INPUT " INGRESE EL VALOR DEL ELEMENTO = ",S(L4,L5)
0501 GOTO 0199
0502 REM
0503 REM          CALCULO DEL PASO EN FUNCION DEL HAMILTONIANO
0504 REM

```

LISTADO DEL PROGRAMA 109

```
0505 DIM U9(K1,K),I9(K1,N)
0506 REM
0507 MAT U9=U*R
0508 MAT I9=I*B
0509 MAT H9=U9+I9
0510 MAT H8=TRN(H9)
0511 LET H5=0
0512 FOR J9=1 TO N
0513 REM
0514 FOR K8=1 TO K1
0515 LE1 H7=H7+H9(K8,J9)*H8(J9,K8)
0516 NEXT K8
0517 LET H5=H5+H7
0518 NEXT J9
0519 RETURN
0520 REM
0521 REM SUBROUTINA PARA EL CALCULO DEL PASO
0522 REM
0523 GOSUB 0456
0524 GOSUB 0502
0525 PRINT " A1=01/100*(L2/H5)"
0526 RETURN
0527 REM
0528 REM SUBROUTINA PARA EL CALCULO DEL PASO
0529 REM
0530 GOSUB 0456
0531 GOSUB 0502
0532 LE1 A2=02/100*L2/H5
0533 RETURN
0534 PRINT "<12>"
0535 PRINT
0536 PRINT " EL SISTEMA DIVERGE EN EL CALCULO DEL P *."
0537 PRINT
0538 STOP
0539 PRINT "<12>"
0540 PRINT
0541 PRINT " EL SISTEMA DIVERGE EN EL CALCULO DEL LAMDA."
0542 PRINT
0543 PRINT
0544 STOP
0545 PRINT "<12> INGRESO DE DATOS."
0546 PRINT
0547 PRINT
0548 PRINT " 1.- INDICE DE OPCIONES."
0549 PRINT
0550 PRINT " 2.- INGRESO POR TECLADO."
0551 PRINT
0552 PRINT " 3.- VERIFICACION Y CORRECCION DE ERRORES."
0553 PRINT
0554 PRINT " 4.- GRABACION EN ARCHIVO."
0555 PRINT
0556 PRINT " 5.- NUEVOS VALORES EN LAS MATRICES DE PONDERACION."
0557 PRINT
0558 PRINT " 6.- NUEVOS VALORES EN EL TAMAÑO DEL PASO."
0559 PRINT
0560 PRINT " 7.- NUEVOS VALORES EN LAS CONDICIONES INICIALES."
0561 PRINT
0562 PRINT " 8.- SALIDA AL MENU PRINCIPAL."
0563 PRINT
0564 LET X9=INT((80-10*K)/2)
0565 INPUT " ESCUJA LA OPCION ----> ",B3
0566 IF B3="1" THEN GOTO 0545
0567 IF B3="8" THEN GOTO 0059
```

LISTADO DEL PROGRAMA 110

```
0568 IF BS="2" THEN GOTO 0080
0569 IF BS="3" THEN GOTO 0115
0570 IF BS="4" THEN GOTO 0655
0571 IF BS="5" THEN GOTO 0107
0572 IF BS="6" THEN GOTO 0092
0573 IF BS="7" THEN GOTO 0214
0574 GOTO 0545
0575 REM
0576 REM      OPTIMIZACION USANDO EL ALGORITMO DE TAMURA
0577 REM
0578 PRINT "<12>          MENU DE OPTIMIZACION"
0579 PRINT
0580 PRINT " 1.- INDICE DE OPCIONES"
0581 PRINT
0582 PRINT " 2.- INGRESO DIRECTO POR TECLADO"
0583 PRINT
0584 PRINT " 3.- INGRESO DESDE UN ARCHIVO DE DATOS"
0585 PRINT
0586 PRINT " 4.- GRABACION DE RESULTADOS EN ARCHIVO?"
0587 PRINT
0588 PRINT " 5.- SALIDA AL PROGRAMA PRINCIPAL"
0589 PRINT
0590 INPUT "ESCOJA LA OPCION -----> ",BS
0591 IF BS="1" THEN GOTO 0578
0592 IF BS="2" THEN GOTO 0228
0593 IF BS="3" THEN GOTO 0598
0594 IF BS="4" THEN GOTO 0984
0595 IF BS="5" THEN GOTO 0059
0596 GOTO 0578
0597 GOTO 0578
0598 INPUT "<12>  NOMBRE DEL ARCHIVO EN QUE SE GRABO EL SISTEMA-----> ",CS
0599 OPEN FILE (4,3),CS
0600 INPUT FILE (4),N,K1,A1,A2
0601 DIM A(N,N),B(N,N),C(N,N),P(N,N),G(N,N),R(N,N),S(N,N),H(K1,N),T(K1,N)
0602 DIM X(K1+1,N),H9(K1,N),H8(N,K1)
0603 FOR I=1 TO N
0604   FOR J=1 TO N
0605     INPUT FILE (4),A(I,J)
0606   NEXT J
0607 NEXT I
0608 FOR I=1 TO N
0609   FOR J=1 TO N
0610     INPUT FILE (4),B(I,J)
0611   NEXT J
0612 NEXT I
0613 FOR I=1 TO N
0614   FOR J=1 TO N
0615     INPUT FILE (4),C(I,J)
0616   NEXT J
0617 NEXT I
0618 FOR I=1 TO N
0619   FOR J=1 TO N
0620     INPUT FILE (4),P(I,J)
0621   NEXT J
0622 NEXT I
0623 FOR I=1 TO N
0624   FOR J=1 TO N
0625     INPUT FILE (4),U(I,J)
0626   NEXT J
0627 NEXT I
0628 FOR I=1 TO N
0629   FOR J=1 TO N
0630     INPUT FILE (4),R(I,J)
```

LISTADO DEL PROGRAMA 111

```
0631 NEXT J
0632 NEXT I
0633 FOR I=1 TO N
0634 FOR J=1 TO N
0635 INPUT FILE (4),S(I,J)
0636 NEXT J
0637 NEXT I
0638 FOR I=1 TO K1+1
0639 FOR J=1 TO N
0640 INPUT FILE (4),X(I,J)
0641 NEXT J
0642 NEXT I
0643 FOR I=1 TO K1
0644 FOR J=1 TO N
0645 INPUT FILE (4),H(I,J)
0646 NEXT J
0647 NEXT I
0648 FOR I=1 TO K1
0649 FOR J=1 TO N
0650 INPUT FILE (4),T(I,J)
0651 NEXT J
0652 NEXT I
0653 CLOSE
0654 GO TO 0228
0655 REM
0656 REM GRABACION EN ARCHIVO
0657 REM
0658 IF N=0 THEN GO TO 0660
0659 GO TO 0664
0660 PRINT "<12> NO EXISTEN DATOS ----> "
0661 INPUT " PRESIONE (NEW LINE) PARA CONTINUAR",B$
0662 IF B$<>"XXXXXXXX" THEN GO TO 0059
0663 GO TO 0059
0664 INPUT "<12> NOMBRE DEL ARCHIVO EN EL QUE SE ESTA GRAHANDO EL SISTEMA----->",C$
0665 OPEN FILE (5,1),C$
0666 PRINT FILE (5),M
0667 PRINT FILE (5),K1
0668 PRINT FILE (5),A1
0669 PRINT FILE (5),A2
0670 FOR I=1 TO N
0671 FOR J=1 TO N
0672 PRINT FILE (5),A(I,J)
0673 NEXT J
0674 NEXT I
0675 FOR I=1 TO N
0676 FOR J=1 TO N
0677 PRINT FILE (5),H(I,J)
0678 NEXT J
0679 NEXT I
0680 FOR I=1 TO N
0681 FOR J=1 TO N
0682 PRINT FILE (5),C(I,J)
0683 NEXT J
0684 NEXT I
0685 FOR I=1 TO N
0686 FOR J=1 TO N
0687 PRINT FILE (5),P(I,J)
0688 NEXT J
0689 NEXT I
0690 FOR I=1 TO N
0691 FOR J=1 TO N
0692 PRINT FILE (5),Q(I,J)
0693 NEXT J
```

```

LISTADO DEL PROGRAMA 112
0694 NEXT I
0695 FOR I=1 TO N
0696   FOR J=1 TO N
0697     PRINT FILE (5),R(I,J)
0698   NEXT J
0699 NEXT I
0700 FOR I=1 TO N
0701   FOR J=1 TO N
0702     PRINT FILE (5),S(I,J)
0703   NEXT J
0704 NEXT I
0705 FOR I=1 TO K1+1
0706   FOR J=1 TO N
0707     PRINT FILE (5),X(I,J)
0708   NEXT J
0709 NEXT I
0710 FOR I=1 TO K1
0711   FOR J=1 TO N
0712     PRINT FILE (5),H(I,J)
0713   NEXT J
0714 NEXT I
0715 FOR I=1 TO K1
0716   FOR J=1 TO N
0717     PRINT FILE (5),T(I,J)
0718   NEXT J
0719 NEXT I
0720 CLOSE
0721 GOTO 0545
0722 REM
0723 REM           RESULTADOS
0724 REM
0725 INPUT " NOMBRE DE LA IMPRESION DE RESULTADOS PARA ESCRITURA EN PAPEL-->" ,DS
0726 AUDIT DS
0727 PRINT "<12>"
0728 PRINT "           MENU DE RESULTADOS,"
0729 PRINT
0730 PRINT " 1.- LISTA DE OPCIONES"
0731 PRINT " ."
0732 PRINT " 2.- SISTEMA EN MEMORIA DE LA CONSOLA"
0733 PRINT
0734 PRINT " 3.- SISTEMA EN ARCHIVO DE RESULTADOS"
0735 PRINT
0736 PRINT " 4.- SALIDA AL PROGRAMA PRINCIPAL"
0737 PRINT
0738 INPUT "   ESCOJA LA OPCION ----->   " ,BS
0739 IF BS="1" THEN GOTO 0727
0740 IF BS="2" THEN GOTO 0745
0741 IF BS="3" THEN GOTO 0907
0742 AUDIT
0743 IF BS="4" THEN GOTO 0059
0744 GOTO 0727
0745 REM
0746 REM   SISTEMA EN LA CONSOLA
0747 REM
0748 LET CS="TECLADO"
0749 PRINT "<12>           MENU DE IMPRESION DE RESULTADOS"
0750 PRINT
0751 PRINT "           SISTEMA EN EL ARCHIVO---> " ,CS
0752 PRINT
0753 PRINT "1.- LISTA DE OPCIONES,"
0754 PRINT
0755 PRINT "2.- IMPRESION DEL SISTEMA : MATRICES--> (A), (B), (C)"
0756 PRINT

```

```

LISTADO DEL PROGRAMA 113
0757 PRINT "3.- IMPRESION DE LAS MATRICES DE Ponderacion--> (P), (Q), (R), (S)."
```

0758 PRINT

```

0759 PRINT "4.- CONDICIONES INICIALES."
0760 PRINT
0761 PRINT "5.- TRAYECTORIAS OPTIMAS."
0762 PRINT
0763 PRINT "6.- CONTROLES OPTIMOS."
0764 PRINT
0765 PRINT "7.- ESTADOS      Z*I      OPTIMOS"
0766 PRINT
0767 PRINT "8.- FUNCION DE COSTO MINIMO."
0768 PRINT
0769 PRINT "9.- SALIDA AL MENU DE IMPRESION"
```

0770 PRINT

```

0771 LET X9=INT((120-10*N)/2)
0772 INPUT "ESCOJA OPCION DE IMPRESION---->,";B5
```

0773 IF B5="1." THEN GOTO 0749

0774 IF B5="2." THEN GOTO 0782

0775 IF B5="3." THEN GOTO 0826

0776 IF B5="4." THEN GOTO 0886

0777 IF B5="5." THEN GOTO 0323

0778 IF B5="6." THEN GOTO 0339

0779 IF B5="7." THEN GOTO 0353

0780 IF B5="8." THEN GOTO 0452

0781 IF B5="9." THEN GOTO 0722

```

0782 PRINT "<12>          SISTEMA          ",CS
0783 PRINT
0784 PRINT "
0785 PRINT
0786 PRINT
0787 FOR I=1 TO N
0788   PRINT TAB(X9);
0789   FOR J=1 TO M
0790     PRINT USING "-###.###",A(I,J);
0791   NEXT J
0792   PRINT
0793   PRINT
0794 NEXT I
0795 INPUT "PRESIONE [ NEW LINE ] TECLA PARA CONTINUAR",AS
0796 IF AS<>"XXX" THEN GOTO 0797
0797 PRINT "<12>"
0798 PRINT "
0799 PRINT
0800 PRINT
0801 FOR I=1 TO N
0802   PRINT TAB(X9);
0803   FOR J=1 TO M
0804     PRINT USING "-###.###",B(I,J);
0805   NEXT J
0806   PRINT
0807   PRINT
0808 NEXT I
0809 INPUT "PRESIONE [ NEW LINE ] TECLA PARA CONTINUAR",AS
0810 IF AS<>"XXX" THEN GOTO 0811
0811 PRINT "<12>"
0812 PRINT "
0813 PRINT
0814 PRINT
0815 FOR I=1 TO N
0816   PRINT TAB(X9);
0817   FOR J=1 TO M
0818     PRINT USING "-###.###",C(I,J);
0819   NEXT J
```

MATRIZ A"

MATRIZ B"

MATRIZ C"

LISTADO DEL PROGRAMA 116

```
0946 FOR I=1 TO N
0947   FOR J=1 TO N
0948     INPUT FILE (6),S(I,J)
0949     NEXT J
0950   NEXT I
0951 FOR I=1 TO K1+1
0952   FOR J=1 TO N
0953     INPUT FILE (6),X(I,J)
0954     NEXT J
0955   NEXT I
0956 FOR I=1 TO K1
0957   FOR J=1 TO N
0958     INPUT FILE (6),H(I,J)
0959     NEXT J
0960   NEXT I
0961 FOR I=1 TO K1
0962   FOR J=1 TO N
0963     INPUT FILE (6),T(I,J)
0964     NEXT J
0965   NEXT I
0966 FOR I=1 TO K1
0967   FOR J=1 TO N
0968     INPUT FILE (6),U(I,J)
0969     NEXT J
0970   NEXT I
0971 FOR I=1 TO K1
0972   FOR J=1 TO N
0973     INPUT FILE (6),Z(I,J)
0974     NEXT J
0975   NEXT I
0976 INPUT FILE (6),Y7
0977 INPUT FILE (6),Y8
0978 CLOSE
0979 GOTO 0749
0980 INPUT "<12> DESEA SALIR DEL PROGRAMA? (S/N) ",B5
0981 IF B5="S" THEN GOTO 0983
0982 GOTO 0001
0983 END
0984 PRINT "<12>"
0985 INPUT "INGRESE EL NOMBRE DEL ARCHIVO DE RESULTADOS----> ",E5
0986 OPEN FILE (7,1),E5
0987 PRINT FILE (7),N
0988 PRINT FILE (7),K1
0989 PRINT FILE (7),A1
0990 PRINT FILE (7),A2
0991 FOR J=1 TO N
0992   FOR I=1 TO N
0993     PRINT FILE (7),A(I,J)
0994   NEXT I
0995 NEXT J
0996 FOR I=1 TO N
0997   FOR J=1 TO N
0998     PRINT FILE (7),H(I,J)
0999   NEXT J
1000 NEXT I
1001 FOR I=1 TO N
1002   FOR J=1 TO N
1003     PRINT FILE (7),C(I,J)
1004   NEXT J
1005 NEXT I
1006 FOR I=1 TO N
1007   FOR J=1 TO M
1008     PRINT FILE (7),P(I,J)
```

A P E N D I C E . C

CONDICIONES NECESARIAS PARA UN
EXTREMO LOCAL

Sea la función $f(\underline{X}, \underline{u})$ sujeto de la restricción $g_i(\underline{X}, \underline{u}) = 0$

$$\forall_i = 1, 2, 3, \dots, n$$

Nuestro problema de minimización nos dice que:

$$\left. \frac{\partial f}{\partial x} \right|_0 = \left. \frac{\partial f}{\partial u} \right|_0 = 0$$

lo que constituye una condición necesaria y representa una condición estacionaria.

Encontramos ahora la condición suficiente desarrollando en series de Taylor.

Obteniendo la segunda derivada para obtener un mínimo necesitamos que:

$$h^2 f_{xx} + 2kh f_{xu} + k^2 f_{uu} > 0 \quad \text{Ecuación C 1}$$

lo que puede expresarse como:

$$\begin{bmatrix} h & k \end{bmatrix} \begin{bmatrix} f_{xx} & f_{xu} \\ f_{xu} & f_{uu} \end{bmatrix} \begin{bmatrix} h \\ k \end{bmatrix} > 0 \quad \text{Ecuación C2}$$

donde: h, k coeficientes > 0

$$f_{xx} = \frac{\partial^2 f}{\partial x^2}$$

$$f_{uu} = \frac{\partial^2 f}{\partial u^2}$$

$$f_{xu} = \frac{\partial^2 f}{\partial x \partial u}$$

$$\bar{J} = J + \underline{\lambda}^T (\dot{\underline{x}} - f(\underline{x}, \underline{u}, t))$$

donde:

$$J = \phi(\underline{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\underline{x}, \underline{u}, t) dt$$

Por lo tanto:

$$\bar{J} = \phi(\underline{x}(t_f), t_f) + \int_{t_0}^{t_f} \left[L(\underline{x}, \underline{u}, t) - \underline{\lambda}^T \{ \dot{\underline{x}} - f(\underline{x}, \underline{u}, t) \} \right] dt$$

Como ya definimos el Hamiltoniano podemos colocarlo en la integral:

$$\bar{J} = \phi(\underline{x}(t_f), t_f) + \int_{t_0}^{t_f} \left[H(\underline{x}, \underline{u}, \underline{\lambda}, t) - \underline{\lambda}^T \dot{\underline{x}}(t) \right] dt.$$

y es sobre esta función sobre la cual vamos a trabajar. La función así construida se denomina Lagrangiano ya que $\underline{\lambda}$ es el vector de los multiplicadores de Lagrange que ayudaron a construir una función aumentada, la misma que no tiene restricciones y que usaremos para encontrar los valores óptimos de los diferentes controles que minimicen la función de costo J .

Si asumimos que las restricciones impuestas son independientes y además si las funciones tanto de costo como la que representa los estados de los subsistemas son continuos y continuamente diferenciales de primer orden, deben satisfacerse para el Lagrangiano y para la función ampliada \bar{J} las condiciones estacionarias del Lagrangiano. Por lo tanto:

$$\frac{\partial H}{\partial \underline{u}} = 0 \quad \frac{\partial H}{\partial \underline{z}} = 0 \quad \frac{\partial H}{\partial \underline{x}} = 0 \quad \frac{\partial H}{\partial \underline{\lambda}} = 0$$

Sea la función $f(\underline{X}, \underline{u})$ sujeto de la restricción $g_i(\underline{X}, \underline{u}) = 0$

$$\forall_i = 1, 2, 3, \dots, n$$

Nuestro problema de minimización nos dice que:

$$\left. \frac{\partial f}{\partial x} \right|_0 = \left. \frac{\partial f}{\partial u} \right|_0 = 0$$

lo que constituye una condición necesaria y representa una condición estacionaria.

Encontramos ahora la condición suficiente desarrollando en series de Taylor.

Obteniendo la segunda derivada para obtener un mínimo necesitamos que:

$$h^2 f_{xx} + 2kh f_{xu} + k^2 f_{uu} > 0 \quad \text{Ecuación C 1}$$

lo que puede expresarse como:

$$\begin{bmatrix} h & k \end{bmatrix} \begin{bmatrix} f_{xx} & f_{xu} \\ f_{xu} & f_{uu} \end{bmatrix} \begin{bmatrix} h \\ k \end{bmatrix} > 0 \quad \text{Ecuación C2}$$

donde: h, k coeficientes > 0

$$f_{xx} = \frac{\partial^2 f}{\partial x^2}$$

$$f_{uu} = \frac{\partial^2 f}{\partial u^2}$$

$$f_{xu} = \frac{\partial^2 f}{\partial x \partial u}$$

De C^2 podemos decir que la matriz **Jacobiana** debe ser definida positiva para un mínimo local.

$$\begin{pmatrix} f_{xx} & f_{xu} \\ f_{xu} & f_{uu} \end{pmatrix} > 0$$

Veamos ahora ciertas restricciones de igualdad.

1. Empecemos con algún ejemplo:

Maximice la siguiente función:

$$f(x, y) = x + y$$

$$\text{sujeto a que } x^2 + y^2 = 1$$

a) Eliminación: $y = \pm \sqrt{1 - x^2}$
 $f(x, y) = g(x) = x \pm \sqrt{1 - x^2}$
que es una función sin restricción.

b) Substitución

$$\text{Sea } x = \cos \theta$$

$$y = \sin \theta$$

$$f(x, y) = h(\theta) = \cos \theta + \sin \theta$$

y vemos que reduce las variables en la restricción.

c) Usemos los multiplicadores de Lagrange.

Sea $f(x, y)$ sujeto a $g(x, y) = 0$, creamos una nueva función que será una función aumentada de la forma siguiente:

$$F(x, y, \lambda) = f(x, y) + \lambda g(x, y)$$

donde λ es un multiplicador de Lagrange, una variable adjunta que va a ser evaluada.

Obtengamos las condiciones necesarias para obtener un mínimo utilizando el método de los multiplicadores de Lagrange.

1. $F_x = f_x + \lambda g_x = 0$
2. $F_y = f_y + \lambda g_y = 0$
3. $F_\lambda = g(x, y) = 0$

De la restricción $g(x, y) = 0$ obtenemos la siguiente condición:

$$g_x dx + g_y dy = 0$$

donde: $\frac{dx}{dy} = -\frac{g_y}{g_x} = h$

Encontremos la condición suficiente para el máximo:

De la ecuación C1 se tiene que:

$$h^2 F_{xx} - 2hk F_{xy} + k^2 F_{yy} < 0 \quad (> 0 \text{ para el mínimo})$$

$$\text{Si } k = 1 \quad \begin{pmatrix} dx & dy \end{pmatrix} \begin{pmatrix} F_{xx} & F_{xy} \\ F_{xy} & F_{yy} \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} < 0 \quad \text{Ecs. C.3}$$

$$\begin{pmatrix} h & 1 \end{pmatrix} \begin{pmatrix} F_{xx} & F_{xy} \\ F_{xy} & F_{yy} \end{pmatrix} \begin{pmatrix} h \\ 1 \end{pmatrix} < 0$$

Trabajemos ahora con nuestras funciones vectoriales:

Sea: $f(x_1, x_2, \dots, x_n; u_1, \dots, u_m)$ sujeto a

$$g_i(x_1, \dots, x_n; u_1, u_2, \dots, u_m) = 0 \quad i = 1, \dots, n$$

Construyamos entonces nuestra función ampliada en función de los multiplicadores de Lagrange:

$$H(\underline{x}, \underline{u}, \underline{\lambda}) = f(\underline{x}, \underline{u}) + \underline{\lambda}^T \underline{g}(\underline{x}, \underline{u})$$

$$H = \underline{f} + \sum_{i=1}^N \underline{\lambda}_i^T \underline{g}_i \quad \text{y a esta función se la conoce}$$

como el Hamiltoniano.

La condición suficiente se expresa como: (aplicando la ecuación C.3)

$$\underline{H}_{uu} + \underline{H}_{ux} \underline{h} + \underline{h}^T \underline{H}_{xu} + \underline{h}^T \underline{H}_{xx} \underline{h} = 0$$

lo que se encuentra aplicando diferenciación vectorial.

Hasta ahora solo hemos visto para el caso de sistemas continuos estáticos, vamos a ver para el caso continuo y dinámico.

Cualquier sistema dinámico se encuentra descrito a variables de estado mediante:

$$\dot{\underline{x}}(t) = f(\underline{x}(t), \underline{u}(t), t) \quad \text{con } t_0 \leq t \leq t_f \quad \text{ecuación C.4}$$

$t_0, t_f, x(t_0)$ son valores conocidos

La ecuación C.4 representa la restricción en nuestro sistema.

Escojamos la función que deseamos minimizar y que representará en realidad la función de costo:

Sea entonces:

$$J = \Phi(\underline{x}(t_f), t_f) + \int_{t_0}^{t_f} \left[L(\underline{x}(t), \underline{u}(t), t) \right] dt$$

El Hamiltoniano en este caso podemos definirlo como:

$$H(\underline{x}, \underline{u}, \lambda, t) = L(\underline{x}, \underline{u}, t) - \lambda^T f(\underline{x}, \underline{u}, t)$$

y lo que en realidad se busca es el valor óptimo de $\underline{u}(t)$ dentro del intervalo $t \in (t_0, t_f)$ de forma que minimice la función, de costo en este caso, J .

Definamos entonces la función aumentada:

$$\bar{J} = J + \underline{\lambda}^T (\dot{\underline{x}} - f(\underline{x}, \underline{u}, t))$$

donde:

$$J = \phi(\underline{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\underline{x}, \underline{u}, t) dt$$

Por lo tanto:

$$\bar{J} = \phi(\underline{x}(t_f), t_f) + \int_{t_0}^{t_f} \left[L(\underline{x}, \underline{u}, t) - \underline{\lambda}^T \{ \dot{\underline{x}} - f(\underline{x}, \underline{u}, t) \} \right] dt$$

Como ya definimos el Hamiltoniano podemos colocarlo en la integral:

$$\bar{J} = \phi(\underline{x}(t_f), t_f) + \int_{t_0}^{t_f} \left[H(\underline{x}, \underline{u}, \underline{\lambda}, t) - \underline{\lambda}^T \dot{\underline{x}}(t) \right] dt.$$

Y es sobre esta función sobre la cual vamos a trabajar. La función así construida se denomina Lagrangiano ya que $\underline{\lambda}$ es el vector de los multiplicadores de Lagrange que ayudaron a construir una función aumentada, la misma que no tiene restricciones y que usaremos para encontrar los valores óptimos de los diferentes controles que minimicen la función de costo J .

Si asumimos que las restricciones impuestas son independientes y además si las funciones tanto de costo como la que representa los estados de los subsistemas son continuos y continuamente diferenciales de primer orden, deben satisfacerse para el Lagrangiano y para la función ampliada \bar{J} las condiciones estacionarias del Lagrangiano. Por lo tanto:

$$\frac{\partial H}{\partial \underline{u}} = 0 \quad \frac{\partial H}{\partial \underline{z}} = 0 \quad \frac{\partial H}{\partial \underline{x}} = 0 \quad \frac{\partial H}{\partial \underline{\lambda}} = 0$$