

DIVISION DE ESTUDIOS DE POSGRADO
FACULTAD DE INGENIERIA
INGENIERIA MECANICA

SINTESIS DE MECANISMOS RSSR GENERADORES
DE FUNCION CON VENTAJA MECANICA OPTIMA

TRABAJO DESARROLLADO POR:

ANGEL A. ROJAS SALGADO

Síntesis de mecanismos RSSR generadores de función con ventaja mecánica óptima

En el trabajo que a continuación se expone se sintetiza un mecanismo RSSR para una función específica, pero el proceso es adaptable a cualquier otra función.

Enunciado del problema.

Determinar las dimensiones de un mecanismo RSSR para generación de función, de tal manera que la ventaja mecánica sea la máxima posible y que el eslabón de entrada dé vuelta completa, para generar la siguiente tabla de valores, en que ϕ y ψ son la entrada y la salida, respectivamente

ϕ (grados)	ψ (grados)
30	45
45	60
60	90
90	60
150	30

Para determinar las dimensiones de diversos mecanismos se recurre a los procedimientos de síntesis; para generación de función se propone [1]*: "Dada una función $f = f(x_i)$ ($i = 1, 2, \dots, n$) definida sobre un conjunto discreto de valores, calcular las dimensiones relevantes de un mecanismo de una topología dada, para producir una relación entrada-salida que coincida con la función f en el conjunto discreto de valores especificados.

La relación entrada-salida necesaria para el mecanismo RSSR se determinará empleando un proceso análogo al

* números en parentesis cuadrados designan referencias en la bibliografía.

descrito en [2]. Sea el mecanismo ilustrado en la figura 1, se establecerá un sistema de ejes coordenados en los pares de rotación de entrada y salida (R_{12} y R_{14}) de tal forma que el eje "x", en ambos casos coincida con el eje de rotación, "z" coincidiendo con IO, "y" se elige de manera que se forme un sistema coordenado derecho en cada caso. Se definen, además, los siguientes vectores:

- \underline{a} = al segmento dirigido que va de B a A
- \underline{b} = al segmento dirigido que va de A a D
- \underline{c} = al segmento dirigido que va de B a C
- \underline{d} = al segmento dirigido que va de C a D

mostrados en la figura 2. De acuerdo a esta última figura, se tiene el siguiente polígono vectorial:

$$\underline{b} = \underline{c} + \underline{d} - \underline{a} \quad (1)$$

Sean los vectores $\underline{a}_j, \underline{b}_j$ y \underline{d}_j ($j=1, 2, \dots, n$), definidos en una posición cualquiera, \underline{Q}_j y \underline{R}_j matrices de rotación

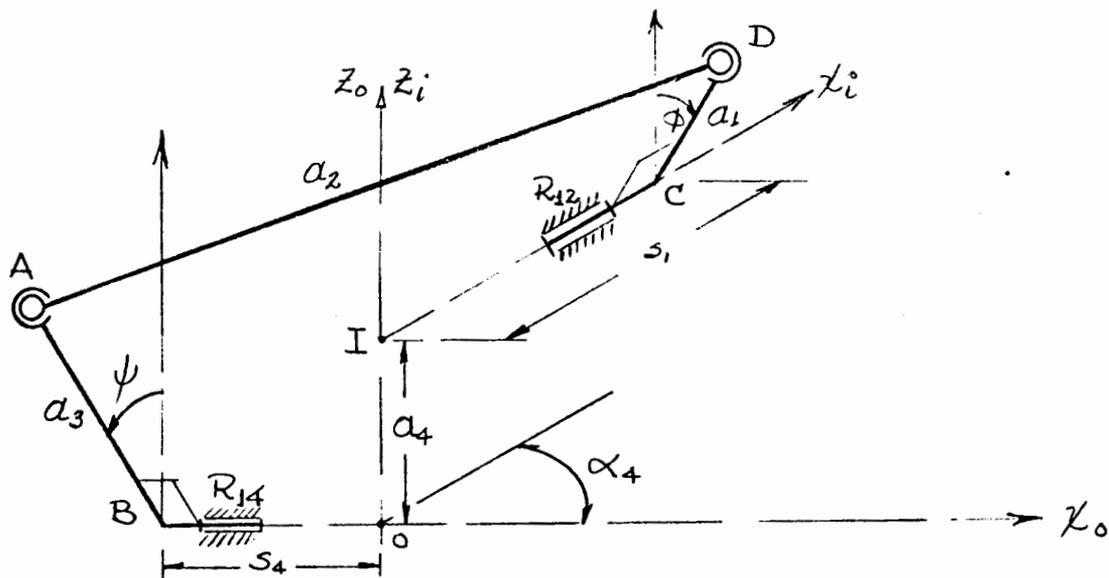


Figura 1.- Mecanismo RSSR ilustrando sus parámetros

que transforman \underline{a} y \underline{d} de la configuración de referencia \underline{a}_0 y \underline{d}_0 a su posición actual, por lo tanto:

$$\underline{a}_j = \underline{Q}_j \underline{a}_0 \quad (2)$$

$$\underline{d}_j = \underline{R}_j \underline{d}_0 \quad (3)$$

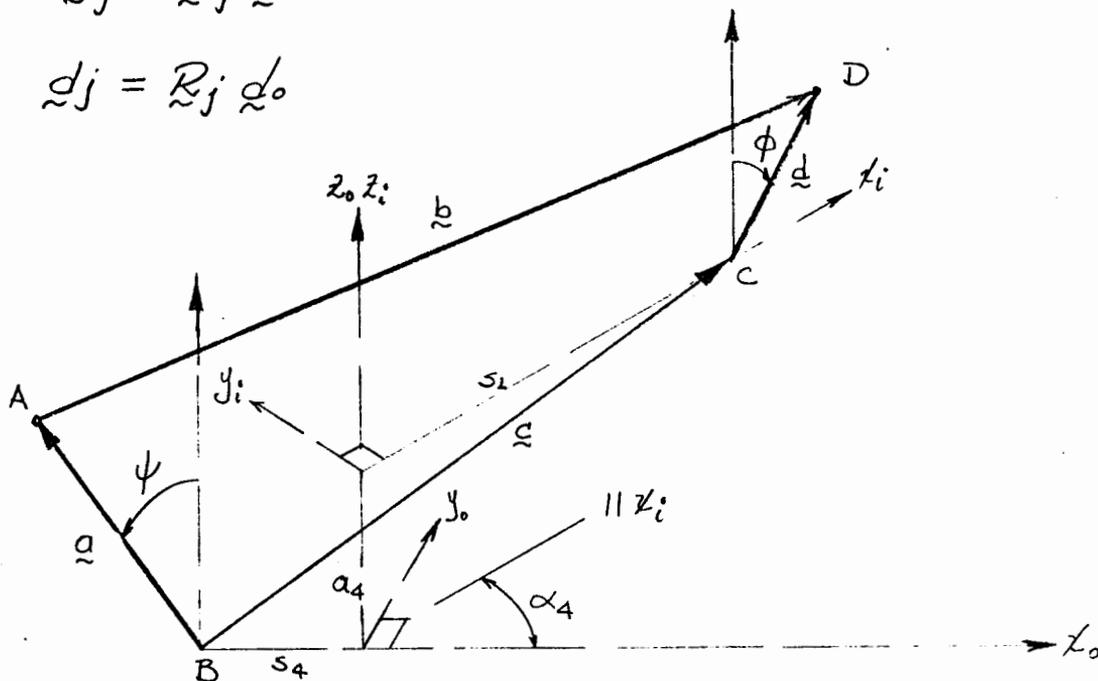


Figura 2.- Mecanismo RSSR mostrando sistemas coordenados.

El eslabón de magnitud a_2 se elegirá como apoyo en el desarrollo, debido a que es constante a través del movimiento y acopla la barra de entrada con la de salida. Así

$$\begin{aligned} \|\underline{b}_j\|^2 &= \underline{b}_j^T \underline{b}_j = (\underline{c} + \underline{d}_j - \underline{a}_j)^T (\underline{c} + \underline{d}_j - \underline{a}_j) \\ &= \underline{c}^T \underline{c} + \underline{d}_j^T \underline{d}_j + \underline{a}_j^T \underline{a}_j + 2\underline{c}^T \underline{d}_j - 2\underline{c}^T \underline{a}_j - 2\underline{d}_j^T \underline{a}_j \quad (4) \end{aligned}$$

De acuerdo a la figura 2, el vector \underline{c} puede expresarse como:

$$\underline{c} = \underline{B}\underline{o} + \underline{O}\underline{I} + \underline{I}\underline{c}$$

referido al sistema coordenado i , quedará expresado como:

$$[\underline{c}]_i = (s_4 \cos \alpha_4 + s_1, -s_4 \sin \alpha_4, a_4)^T \quad (5)$$

de aquí

$$\begin{aligned} \underline{c}^T \underline{c} &= s_4^2 \cos^2 \alpha_4 + s_1^2 + 2 s_4 s_1 \cos \alpha_4 + s_4^2 \operatorname{sen}^2 \alpha_4 + d_4^2 \\ &= s_4^2 + s_1^2 + d_4^2 + 2 s_4 s_1 \cos \alpha_4 \end{aligned} \quad (6)$$

Tomando en consideración que \underline{a}_0 y \underline{d}_0 pueden estar colocados a un ángulo arbitrario ψ_0 y ϕ_0 con respecto al eje "z", cada vector quedará expresado en función de sus componentes como:

$$[\underline{d}_0]_i = (0, -d_1 \operatorname{sen} \phi_0, d_1 \cos \phi_0)^T \quad (7)$$

$$[\underline{a}_0]_0 = (0, -a_3 \operatorname{sen} \psi_0, a_3 \cos \psi_0)^T$$

considerados en los sistemas i y 0 respectivamente. Las matrices de rotación tienen la siguiente representación:

$$[Q_j]_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi_j & -\operatorname{sen} \psi_j \\ 0 & \operatorname{sen} \psi_j & \cos \psi_j \end{bmatrix} \quad [R_j]_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_j & -\operatorname{sen} \phi_j \\ 0 & \operatorname{sen} \phi_j & \cos \phi_j \end{bmatrix}$$

sea además \underline{s} la matriz de rotación que sobrepone el sistema 0 con el i ; quedando las coordenadas de 0 referidas en el sistema i .

$$[\underline{s}]_i = \begin{bmatrix} \cos \alpha_4 & \operatorname{sen} \alpha_4 & 0 \\ -\operatorname{sen} \alpha_4 & \cos \alpha_4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Efectuando las operaciones indicadas en la ecuación (4):

$$\begin{aligned} [\underline{c}^T \underline{a}_j]_i &= [\underline{c}]_i^T [\underline{s}]_i [Q_j]_i \underline{a}_0 \\ &= d_3 [a_4 \cos(\psi_j + \psi_0) - s_1 \operatorname{sen} \alpha_4 \operatorname{sen}(\psi_j + \psi_0)] \end{aligned}$$

$$[\underline{a}_j^T \underline{a}_j]_i = [\underline{R}_j \underline{d}_j]_i^T [\underline{s}]_i [\underline{Q}_j \underline{a}_0]_i$$

$$= \alpha_1 \alpha_3 [\cos \alpha_4 \sin(\psi_j + \psi_0) \sin(\phi_j + \phi_0) + \cos(\phi_j + \phi_0) \cos(\psi_j + \psi_0)]$$

$$[\underline{c}^T \underline{d}_j]_i = \alpha_1 [s_4 \sin \alpha_4 \sin(\phi_j + \phi_0) + a_4 \cos(\phi_j + \phi_0)]$$

Considerando $\phi_i = \phi_j + \phi_0$ y $\psi_i = \psi_j + \psi_0$ además de las expresiones anteriores, la ecuación (4) queda como:

$$\alpha_2^2 = s_4^2 + s_1^2 + a_4^2 + 2s_4 s_1 \cos \alpha_4 + a_1^2 + a_3^2 + 2\alpha_1 [s_4 \sin \alpha_4 \sin \phi_i + a_4 \cos \phi_i] - 2\alpha_3 [a_4 \cos \psi_i - s_1 \sin \alpha_4 \sin \psi_i] - 2\alpha_1 \alpha_3 [\cos \alpha_4 \sin \psi_i \sin \phi_i + \cos \phi_i \cos \psi_i] \quad (8)$$

la cual es semejante a la obtenida por Denavit y Hartenberg y expresada en [3]. Introduciendo las siguientes constantes en la expresión obtenida:

$$k_1 = \frac{s_1 \tan \psi_0 \sin \alpha_4 - a_4}{\alpha_1 \cos \phi_0 (1 + \tan \phi_0 \tan \psi_0 \cos \alpha_4)}$$

$$k_2 = \frac{a_4 \tan \psi_0 + s_1 \sin \alpha_4}{\alpha_1 \cos \phi_0 (1 + \tan \phi_0 \tan \psi_0 \cos \alpha_4)}$$

$$k_3 = \frac{a_4 + s_4 \tan \phi_0 \sin \alpha_4}{\alpha_3 \cos \psi_0 (1 + \tan \phi_0 \tan \psi_0 \cos \alpha_4)}$$

$$k_4 = \frac{s_4 \sin \alpha_4 - a_4 \tan \phi_0}{\alpha_3 \cos \psi_0 (1 + \tan \phi_0 \tan \psi_0 \cos \alpha_4)}$$

$$k_5 = \frac{\tan \phi_0 \cos \alpha_4 - \tan \psi_0}{1 + \tan \phi_0 \tan \psi_0 \cos \alpha_4}$$

$$k_6 = \frac{\tan \psi_0 \cos \alpha_4 - \tan \phi_0}{1 + \tan \phi_0 \tan \psi_0 \cos \alpha_4}$$

$$k_7 = \frac{\cos \alpha_4 + \tan \phi_0 \tan \psi_0}{1 + \tan \phi_0 \tan \psi_0 \cos \alpha_4}$$

$$k_8 = \frac{a_1^2 - a_2^2 + a_3^2 + a_4^2 + s_1^2 + s_4^2 + 2s_1s_4 \cos \alpha_4}{(2a_1a_3 \cos \phi_0 \cos \psi_0)(1 + \tan \phi_0 \tan \psi_0 \cos \alpha_4)}$$

quedando la ecuación (8) expresada como:

$$k_1 \cos \psi_i + k_2 \sin \psi_i + k_3 \cos \phi_i + k_4 \sin \phi_i - k_5 \cos \phi_i \sin \psi_i - k_6 \cos \psi_i \sin \phi_i - k_7 \sin \phi_i \sin \psi_i + k_8 = \cos \psi_i \cos \phi_i \quad (9)$$

en la cual se sustituyen los diferentes valores de ϕ y ψ ; si estos son ocho el sistema es lineal en ocho incógnitas y si son diferentes se forma un sistema indeterminado, el cual requiere para la solución aproximada algún método de optimización tal como se expresa en [1], lo anterior no ocurrirá si se especifican las incógnitas faltantes.

Como se plantea que la ventaja mecánica sea óptima se procederá a describir esta, así como los requerimientos para tener movilidad completa del eslabón de entrada.

Despreciando los efectos dinámicos y fricciones, la potencia instantánea de entrada y salida en un mecanismo sin ramificar (o sea, una entrada y una salida unidas por una ualla simple) es igual al "flujo de potencia" en cualquier otro punto del mecanismo. En un mecanismo de un solo grado de libertad sin ramas, el "flujo de potencia" en cualquier punto es el producto de la fuerza \underline{F} en ese punto y la velocidad \underline{v} en la dirección de la fuerza; de aquí,

$$\underline{F}^T \underline{v} = cte = \underline{M}^T \underline{\omega} \quad (10)$$

La ventaja mecánica se define como la relación de la fuerza (o par) en la salida a la fuerza (o par) en la entrada [1,4], o sea:

$$m = \frac{F_{\psi}}{F_{\phi}} = \frac{M_{\psi}}{M_{\phi}} \quad (11)$$

de acuerdo con la expresión (10) se transforma en:

$$m = \frac{M_{\psi}}{M_{\phi}} = \frac{\dot{\phi}}{\dot{\psi}} = \frac{d\phi/dt}{d\psi/dt} = \frac{1}{d\psi/d\phi} \quad (12)$$

La ecuación (9) es una función continua que relaciona la entrada con la salida y es útil para obtener una expresión para la ventaja mecánica, obteniéndose de esta expresión:

$$m = \frac{d\phi}{d\psi} = \frac{\sin\psi(k_1 - k_6 \sin\phi - \cos\phi) + \cos\psi(k_5 \cos\phi + k_7 \sin\phi - k_2)}{\sin\psi(\cos\phi - k_3 + k_5 \sin\psi) + \cos\psi(k_4 - k_6 \cos\psi - k_7 \sin\psi)} \quad (13)$$

Como puede apreciarse la ventaja mecánica es función de la entrada ϕ , puesto que ψ se relaciona con esto por medio de la ecuación (9)

En aplicaciones prácticas, la mayoría de los mecanismos requieren que el eslabón conductor gire 360° en forma continua, ello conduce a cumplir las restricciones necesarias para este fin. Para cada posición diferente del ángulo de entrada se tienen distintas longitudes de la barra acopladora que cumplen con esa posición, en particular interesa la magnitud máxima y mínima que permitan el ensamble del mecanismo. Al girar una revolución completa el eslabón de entrada se genera un conjunto de estas magnitudes, dentro del cual se encuentran las que conducen a la movilidad en la entrada.

Procediendo en forma semejante a [5], en un mecanismo

RSSR como el mostrado en la figura 1, la junta esférica A gira en un plano Π_B perpendicular al eje fijo Z_0 y pasa por el punto fijo B, la ecuación del plano es:

$$\underline{u}_B^T \underline{a} = 0 \quad \text{donde} \quad \underline{u}_B = (1, 0, 0)$$

$$\underline{a} = \underline{Q} \underline{a}_0$$

La longitud del eslabón de salida es constante y expresada por:

$$a_3^2 = \underline{a}_0^T \underline{Q}^T \underline{Q} \underline{a}_0 = \underline{a}^T \underline{a}$$

Una perpendicular al plano Π_B es paralela a \underline{u}_B , la distancia t mostrada en la figura 3 está dada por:

$$t = \underline{u}_B^T (\underline{c} + \underline{d})$$

y la distancia BD' se calcula de:

$$(BD')^2 = (\underline{x} + \underline{c})^T (\underline{x} + \underline{c})$$

donde

$$\underline{x} = \underline{d} - t \underline{u}_B$$

Las longitudes máxima y mínima de la barra accionada para una posición específica de ϕ serán:

$$h_{\min}(\phi) = L = \sqrt{(BD' - BA)^2 + t^2} \quad (14)$$

$$h_{\max}(\phi) = L = \sqrt{(BD' + BA)^2 + t^2}$$

Lo anterior será referido al sistema i , mostrado en la figura 3

$$\underline{u}_B = (\cos \alpha_4, -\sin \alpha_4, 0)^T$$

$$\underline{c} = (s_4 \cos \alpha_4 + s_1, -s_4 \sin \alpha_4, a_4)^T$$

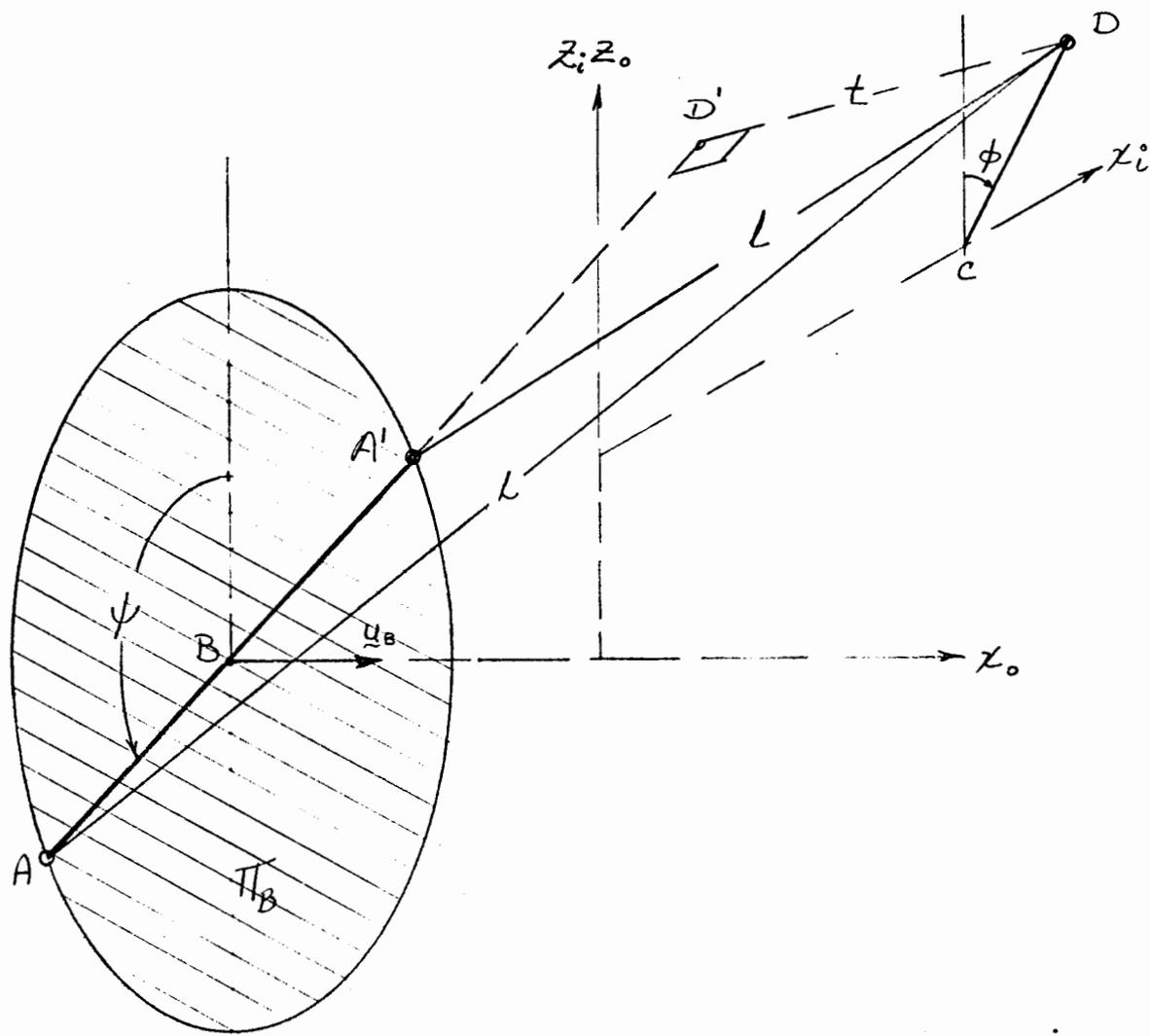


Figura 3.- Mecanismo RSSR mostrando una posición de la entrada y el plano de rotación en la salida.

$$\underline{d} = (0, -a_1 \sin \phi, a_1 \cos \phi)^T$$

$$\underline{a} = (-a_3 \sin \alpha_4 \sin \phi, -a_3 \cos \alpha_4 \sin \phi, a_3 \cos \phi)^T$$

$$\underline{c} + \underline{d} = (s_4 \cos \alpha_4 + s_1, -s_4 \sin \alpha_4 - a_1 \sin \phi, a_4 + a_1 \cos \phi)^T$$

$$t = s_4 + s_1 \cos \alpha_4 + a_1 \sin \alpha_4 \sin \phi$$

$$\underline{z} = (-t \cos \alpha_4, -a_1 \sin \phi + t \sin \alpha_4, a_1 \cos \phi)^T$$

$$\underline{z} + \underline{c} = [\cos \alpha_4 (s_4 - t) + s_1, -\sin \alpha_4 (s_4 - t) - a_1 \sin \phi, a_4 + a_1 \cos \phi]^T$$

$$(BD')^2 = a_1^2 + s_1^2 + a_4^2 + 2a_4 a_1 \cos \phi - (s_1 \cos \alpha_4 + a_1 \sin \alpha_4 \sin \phi)^2$$

$$h_{\min}(\phi) = L = \left\{ \left[\sqrt{a_1^2 + s_1^2 + a_4^2 + 2a_4 a_1 \cos \phi - (s_1 \cos \alpha_4 + a_1 \sin \alpha_4 \sin \phi)^2} - a_3 \right]^2 + (s_4 + s_1 \cos \alpha_4 + a_1 \sin \alpha_4 \sin \phi)^2 \right\}^{1/2} \quad (15)$$

$$h_{\max}(\phi) = L = \left\{ \left[\sqrt{a_1^2 + s_1^2 + a_4^2 + 2a_4 a_1 \cos \phi - (s_1 \cos \alpha_4 + a_1 \sin \alpha_4 \sin \phi)^2} + a_3 \right]^2 + (s_4 + s_1 \cos \alpha_4 + a_1 \sin \alpha_4 \sin \phi)^2 \right\}^{1/2} \quad (16)$$

Los valores extremos de las longitudes para cualquier posición de ϕ se calcularán a partir de:

$$\frac{dh}{d\phi} = 0 \quad (17)$$

Involucrando las ecuaciones (15) y (16) en la (17) considerando además que:

$$u = a_1^2 + s_1^2 + a_4^2 + 2a_4 a_1 \cos \phi - (s_1 \cos \alpha_4 + a_1 \sin \alpha_4 \sin \phi)^2$$

la expresión (17) quedará como:

$$\frac{dh}{d\phi} = \frac{1}{2} \left\{ (u^{1/2} \mp a_3)^2 + t^2 \right\}^{-1/2} \left\{ 2(u^{1/2} \mp a_3) \frac{1}{2} u^{-1/2} \frac{du}{d\phi} + 2t \frac{dt}{d\phi} \right\} = 0$$

pero

$$\frac{du}{d\phi} = -2a_4 a_1 \sin \phi - 2(s_1 \cos \alpha_4 + a_1 \sin \alpha_4 \sin \phi) a_1 \sin \alpha_4 \cos \phi$$

$$\frac{dt}{d\phi} = a_1 \sin \alpha_4 \cos \phi$$

Al realizar las sustituciones y elevar al cuadrado se obtiene:

$$[a_1^2 + s_1^2 + a_4^2 + 2a_4 a_1 \cos \phi - (s_1 \cos \alpha_4 + a_1 \sin \alpha_4 \sin \phi)^2] (s_4 \sin \alpha_4 \cos \phi - a_4 \sin \phi)^2 = a_3^2 [a_4 \sin \phi + \sin \alpha_4 \cos \phi (s_1 \cos \alpha_4 + a_1 \sin \alpha_4 \sin \phi)]^2$$

al desarrollar los productos indicados y agrupar términos en las potencias de $\cos \phi$, la expresión anterior queda en la siguiente forma:

$$b_4 \cos^4 \phi + b_3 \cos^3 \phi + b_2 \cos^2 \phi + b_1 \cos \phi + b_0 = \sin \phi (c_3 \cos^2 \phi + c_2 \cos \phi + c_1 \cos \phi + c_0) \quad (18)$$

$$\text{donde: } b_4 = a_1^2 \sin^2 \alpha_4 (a_4^2 - a_3^2 \sin^2 \alpha_4 - s_4^2 \sin^2 \alpha_4)$$

$$b_3 = 2a_4^3 a_1 + \sin^2 \alpha_4 (4s_4 a_4 s_1 a_1 \cos \alpha_4 - 2a_4 a_3^2 a_1 - 2a_4 a_1 s_4^2)$$

$$b_2 = \sin^4 \alpha_4 [(a_3^2 + s_4^2)(a_1^2 - s_1^2)] + \sin^2 \alpha_4 [a_4^2 (s_1^2 - s_4^2 - 2a_1^2) + a_3^2 s_1^2 - a_1^2 s_4^2] + a_4^2 (a_1^2 - a_3^2 + a_4^2)$$

$$b_1 = \sin^2 \alpha_4 (2a_4 a_3^2 a_1 - 4s_4 a_4 s_1 a_1 \cos \alpha_4) - 2a_4^3 a_1$$

$$b_0 = a_4^2 (a_3^2 - a_1^2 - s_1^2 - a_4^2 + s_1^2 \cos^2 \alpha_4 + a_1^2 \sin^2 \alpha_4)$$

$$c_3 = -2s_4 a_4 a_1^2 \sin^3 \alpha_4$$

$$c_2 = 2a_1 \sin \alpha_4 [s_1 a_4^2 \cos \alpha_4 - s_1 (a_3^2 + s_4^2) (\sin^2 \alpha_4 \cos \alpha_4) - 2a_4^2 s_4]$$

$$c_1 = 2s_4 a_4 \sin^3 \alpha_4 (a_1^2 - s_1^2) - 2a_4 a_3^2 s_1 \sin \alpha_4 \cos \alpha_4 - 2s_4 a_4 \sin \alpha_4 (a_1^2 + a_4^2)$$

$$c_0 = -2s_1 a_1 a_4^2 \cos \alpha_4 \sin \alpha_4$$

Elevarlo al cuadrado y simplificando la ecuación (18) se llega a:

$$A_9 \cos^8 \phi + A_8 \cos^7 \phi + A_7 \cos^6 \phi + A_6 \cos^5 \phi + A_5 \cos^4 \phi + A_4 \cos^3 \phi + A_3 \cos^2 \phi + A_2 \cos \phi + A_1 = 0 \quad (19)$$

donde: $A_9 = b_4^2 + c_3^2$

$$A_8 = 2b_4 b_3 + 2c_3 c_2$$

$$A_7 = b_3^2 + 2b_4 b_2 - c_3^2 + c_2^2 + 2c_3 c_1$$

$$A_6 = 2b_4 b_1 + 2b_3 b_2 - 2c_3 c_2 + 2c_3 c_0 + 2c_2 c_1$$

$$A_5 = b_2^2 + \underline{2b_4 b_0} + \underline{2b_3 b_1} - c_2^2 - \underline{2c_3 c_1} + c_1^2 + \underline{2c_2 c_0}$$

$$A_4 = 2b_3 b_0 + 2b_2 b_1 - 2c_3 c_0 - 2c_2 c_1 + 2c_1 c_0$$

$$A_3 = b_1^2 + \underline{2b_2 b_0} - c_1^2 - \underline{2c_2 c_0} + c_0^2$$

$$A_2 = 2b_1 b_0 - 2c_1 c_0$$

$$A_1 = b_0^2 - c_0^2$$

Las raíces resultantes se analizarán de tal forma que:

$$\min_{\phi} L \leq \sigma_2 \leq \max_{\phi} L$$

para que exista movilidad completa en la entrada.

Las raíces se determinarán empleando la subrutina POLRT

cuyo listado se encuentra en la siguiente página, esta subrutina se basa en el método iterativo de Newton-Raphson, el cual, requiere la evaluación del polinomio y la derivada de este, para obtener una mejor aproximación a la raíz. Su proceso se bosqueja a continuación:

Sea $z = x + iy$ un valor inicial para determinar una raíz de

$$f(z) = \sum_{n=0}^N a_n z^n \quad (i)$$

donde $z^n = (x + iy)^n \quad (ii)$

Definiendo u_n como los términos reales de la expansión de (ii) y v_n como los términos imaginarios, para $n=0: u_0 = 1$ y $v_0 = 0.0$ y para $n > 0$:

$$u_n = x \cdot u_{n-1} - y \cdot v_{n-1} \quad (iii)$$

$$v_n = x \cdot v_{n-1} + y \cdot u_{n-1} \quad (iv)$$

Sean además U y V los términos reales e imaginarios de (i)

$$U = a_0 + \sum_{n=1}^N a_n u_n \quad (v)$$

$$V = \sum_{n=1}^N a_n v_n \quad (vi)$$

$$\frac{\partial U}{\partial x} = \sum_{n=1}^N n \cdot u_{n-1} \cdot a_n \quad (vii)$$

$$\frac{\partial V}{\partial y} = - \sum_{n=1}^N n \cdot v_{n-1} \cdot a_n \quad (viii)$$

Las ecuaciones de la (iii) a la (viii) pueden ser evaluadas iterativamente para $n=1$ a N almacenando u_{n-1} y v_{n-1} . Para calcular Δx y Δy empleando el método de Newton-Raphson el resultado es:

```

SUBROUTINE POLRT(XCOF,COF,M,ROOTR,ROOTI,IER)
REAL XCOF(1),COF(1),ROOTR(1),ROOTI(1)
DOUBLE PRECISION XD,YD,X,Y,XPR,YPR,DX,DY,V,X1,Y1,U,X12,Y12,SUFSU
,DX,DY,TEMP,ALPHA

```

```

000024
000024
000024
000024

```

```

C SUBROUTINE POLRT

```

```

000024

```

```

C PURPOSE

```

```

000024

```

```

COMPUTES THE REAL AND COMPLEX ROOTS OF A REAL POLYNOMIAL

```

```

000025

```

```

C DESCRIPTION OF PARAMETERS

```

```

000025

```

```

XCOF .VECTOR OF M+1 COEFFICIENTS OF THE POLYNOMIAL
ORDERED FROM SMALLEST TO LARGEST POWER

```

```

000025
000025

```

```

COF .WORKING VECTOR OF LENGTH M+1
M .ORDER OF POLYNOMIAL

```

```

000025
000025

```

```

ROOTR .RESULTANT VECTOR OF LENGTH M CONTAINING REAL ROOTS OF
THE POLYNOMIAL

```

```

000025
000025

```

```

ROOTI .RESULTANT VECTOR OF LENGTH M CONTAINING THE
CORRESPONDING IMAGINARY ROOTS OF THE POLYNOMIAL

```

```

000025
000026

```

```

IER .ERROR CODE WHERE

```

```

000026

```

```

IER=0 NO ERROR
IER=1 M LESS THAN ONE
IER=2 M GREATER THAN 36
IER=3 M UNABLE TO DETERMINE ROOT WITH 500 ITERATIONS
ON 5 STARTING VALUES
IER=4 HIGH ORDER COEFFICIENT IS ZERO

```

```

000026
000026
000026
000026
000026
000026

```

```

C REMARKS

```

```

000026

```

```

LIMITED TO 36TH ORDER POLYNOMIAL OR LESS
FLOATING POINT OVERFLOW MAY OCCUR FOR HIGH ORDER
POLYNOMIALS BUT WILL NOT AFFECT THE ACCURACY OF THE RESULTS

```

```

000026
000027
000027

```

```

C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

```

```

000027

```

```

NONE

```

```

000027

```

```

C METHOD

```

```

000027

```

```

NEWTON RAPHSON ITERATIVE TECHNIQUE. THE FINAL ITERATIONS
ON EACH ROOT ARE PERFORMED USING THE ORIGINAL POLYNOMIAL
RATHER THAN THE REDUCED POLYNOMIAL TO AVOID ACCUMULATED
ERRORS IN THE REDUCED POLYNOMIAL.

```

```

000027
000027
000027
000028

```

```

IFIT=0

```

```

000028

```

```

M=M

```

```

000028

```

```

IER=0

```

```

000028

```

```

IF(XCOF(M+1).EQ. 0.0) GO TO 16

```

```

000028

```

```

10 IF(0) 12,12,20

```

```

000028

```

```

C SET ERROR CODE TO 1

```

```

000028

```

```

12 IER=1

```

```

000028

```

```

14 RETURN

```

```

000028

```

```

C SET ERROR CODE TO 4

```

```

000028

```

```

16 IER=4

```

```

000029

```

```

GO TO 14

```

```

000029

```

```

C SET ERROR CODE TO 2

```

```

000029

```

```

18 IER=2

```

```

000029

```

```

GO TO 14

```

```

000029

```

```

20 IF(0-36) 22,22,18

```

```

000029

```

```

22 X=X

```

```

000029

```

```

X=X+1

```

```

000029

```

```

X=X+1

```

```

000029

```

```

X=X+1

```

```

000029

```

```

IF(X-36) 22,22,18

```

```

000029

```

```

IF(0-36)

```

```

000029

```

24	COF(M)=XCOF(L)	000030
C	SET INITIAL VALUES	000030
25	XU=0.00500101	000030
	YU=0.01000101	000030
C	ZERO INITIAL VALUE COUNTER	000030
	IN=0	000030
28	X=XU	000030
	XU=-10.0*YU	000030
	YU=-10.0*X	000031
C	SET X AND Y TO CURRENT VALUE	000031
	X=XU	000031
	Y=YU	000031
	IN=IN+1	000031
	GO TO 32	000031
30	IF I=1	000031
	XPR=X	000031
	YPR=Y	000031
C	EVALUATE POLYNOMIAL AND DERIVATIVES	000031
32	ICT=0	000032
34	UX=0.0	000032
	UY=0.0	000032
	V=0.0	000032
	XT=1.0	000032
	YT=0.0	000032
	U=COF(N+1)	000032
	IF(U) 36,36,36	000032
36	DO 38 I=1,N	000032
	L=N-I+1	000032
	TEMP=COF(L)	000033
	XT2=X*XT-Y*YI	000033
	YT2=X*YI+Y*XI	000033
	U=U+TEMP*XT2	000033
	V=V+TEMP*YT2	000033
	F1=1	000033
	UX=UX+F1*XT*TEMP	000033
	UY=UY-F1*YT*TEMP	000033
	XT=XT2	000033
38	YT=YT2	000033
	SU-SU=UX*UX+UY*UY	000034
	IF(SU) 40,56,40	000034
40	DX=(V*UY-U*UX)/SU	000034
	X=X+DX	000034
	DY=(-U*UY+V*UX)/SU	000034
	Y=Y+DY	000034
42	IF(DABS(DY)+DABS(DX)-1.0-05) 52,44,44	000034
44	ICT=ICT+1	000034
	IF(ICT-500) 34,46,46	000034
46	IF(I=1) 52,48,52	000034
48	IF(I=5) 28,50,50	000035
C	SET ERROR CODE TO 3	000035
50	IEF=3	000035
	GO TO 18	000035
52	DO 54 L=1,N	000035
	MT=N+1-L	000035
	TEMP=COF(MT)	000035
	XCOF(MT)=COF(L)	000035
54	COF(L)=TEMP	000035
	IEF=IEF	000035
	IF(IEF)	000035
	IF(IEF)	000035

	IF(IF11) 60,30,60	000030
56	IF(IF11) 58,28,58	000030
58	X=XPR	000030
	Y=YPR	000030
60	IF11=0	000030
62	IF(DABS(Y)-1.0-04*DABS(X)) 68,64,64	000030
64	ALPHA=X+X	000030
	SUMSQ=X*X+Y*Y	000030
	N=N-2	000037
	GO TO 70	000037
66	X=0.0	000037
	NX=NX-1	000037
	FXX=NX-1	000037
68	Y=0.0	000037
	SUMSQ=0.0	000037
	ALPHA=X	000037
	L=N-1	000037
70	CUF(2)=CUF(2)+ALPHA*CUF(1)	000037
72	DO 74 L=2,N	000038
74	CUF(L+1)=CUF(L+1)+ALPHA*CUF(L)-SUMSQ*CUF(L-1)	000038
76	ROUT1(N2) = Y	000038
	ROUTR(N2) = X	000038
	N2=N2+1	000030
	IF(SUMSQ) 78,80,78	000038
78	Y=-Y	000038
	SUMSQ=0.0	000038
	GO TO 76	000038
80	IF(N) 14,14,26	000038
	END	000039

$$\Delta x = (v \frac{\partial U}{\partial y} - u \frac{\partial U}{\partial x}) / [(\frac{\partial U}{\partial x})^2 + (\frac{\partial U}{\partial y})^2]$$

$$\Delta y = -(u \frac{\partial U}{\partial y} + v \frac{\partial U}{\partial x}) / [(\frac{\partial U}{\partial x})^2 + (\frac{\partial U}{\partial y})^2]$$

obtenido después de emplear las ecuaciones de Cauchy-Riemann las cuales se expresan como:

$$\frac{\partial U}{\partial x} = \frac{\partial V}{\partial y} \quad \text{y} \quad \frac{\partial U}{\partial y} = -\frac{\partial V}{\partial x}$$

para la siguiente aproximación a la raíz se tendrá:

$$x_{k+1} = x_k + \Delta x$$

$$y_{k+1} = y_k + \Delta y$$

Función a optimizar:

Al obtener la ventaja mecánica como se expresó en la ecuación (12), donde se consideraron despreciables los efectos de inercia y fricción, se tiene un "flujo de potencia" instantánea, que será máximo cuando coincidan el par y la velocidad angular para una revolución completa del eslabón de entrada. Del mismo modo; para este caso se dan generalmente estas condiciones. Se determinará el máximo valor acumulado del par en la salida de la siguiente manera:

$$F(\alpha_4, \phi_0, \psi_0) = \frac{1}{2\pi} \int_0^{2\pi} m^2 d\phi \quad (20)$$

con esta forma se evitan cancelaciones de valores positivos y negativos al oscilar la barra de salida.

El integrando de la expresión (20) queda como:

$$u^2 = N^2/D^2$$

donde

$$N^2 = \sin^2\psi (k_1 - k_6 \sin\phi - \cos\phi)^2 + \cos^2\psi (k_5 \cos\phi + k_7 \sin\phi - k_2)^2 + 2 \sin\psi \cos\psi (k_1 - k_6 \sin\phi - \cos\phi)(k_5 \cos\phi + k_7 \sin\phi - k_2)$$

$$D^2 = \sin^2\psi (\cos\psi - k_3 + k_5 \sin\psi)^2 + \cos^2\psi (k_4 - k_6 \cos\psi - k_7 \sin\psi)^2 + 2 \sin\psi \cos\psi (\cos\psi - k_3 + k_5 \sin\psi)(k_4 - k_6 \cos\psi - k_7 \sin\psi)$$

En estas últimas expresiones se aprecia claramente la dependencia de la integral (20) de los ángulos de entrada y salida, pero debido a la dependencia del ángulo de salida del ángulo de entrada, la integral no puede realizarse en forma analítica, se procederá a su realización mediante algún método numérico. Para este fin se empleará el método de Romberg [7] adaptado en la subrutina ROMBER, cuyo listado se encuentra en la siguiente página.

Para facilidad de los cálculos, necesarios en la ecuación (20), la ecuación (9) se modificará expresando el ángulo de salida en forma explícita, mediante la introducción de:

$$\sin\psi = \frac{2 \tan(\psi/2)}{1 + \tan^2(\psi/2)}$$

$$\cos\psi = \frac{1 - \tan^2(\psi/2)}{1 + \tan^2(\psi/2)}$$

obteniéndose

$$\psi = 2 \arctan \left[\frac{F \pm (F^2 - 4EG)^{1/2}}{2E} \right] \quad (21)$$

donde

$$E = s_4^2 \pm s_1^2 + a_4^2 + a_3^2 - a_2^2 + a_1^2 + 2s_1 s_4 \cos\alpha_4 + 2a_1 (s_4 \sin\alpha_4 \sin\phi + a_4 \cos\phi) + 2a_3 a_4 + 2a_1 a_3 \cos\phi$$

$$F = 4(a_1 a_3 \cos\alpha_4 \sin\phi - a_3 s_1 \sin\alpha_4)$$

	RINI=RUM(1,1)-RUM(1,1-1)	00001
	RINI=RINI/RUM(1,1)	00001
	IF (ABS(RINI).GT.PRESI) GO TO 10	00001
	KONVER=.TRUE.	00001
	WRITE(6,300)(RUM(L,L),L=1,1)	00001
	 RINI=RUM(1,1)	00001
	RETURN	00001
80	WRITE(6,100) RUM(1,1)	00001
	RETURN	00001
90	WRITE(6,200)	00001
	RETURN	00001
100	FORMAT(/3X,'NO HAY CONVERGENCIA. EL ULTIMO VALOR CALCULADO',	00001
	/3X,'DE LA INTEGRAL ES :',E13.5/)	00001
200	FORMAT(/3X,'N NO DEBE SER MENOR QUE (2**NAX+1)+1)'/)	00001
300	FORMAT(/3X,5HRUM =6E13.5/)	00001
	END	00001

$$G = F - 4(a_3 a_4 - a_1 a_3 \cos \phi)$$

Esta misma ecuación aparece en [2, 6] en forma semejante pero obtenida por un proceso algo diferente. Del análisis de esta ecuación se puede concluir que tiene solución imaginaria cuando F^2 sea menor que $4EG$, cuando se presenta este caso el mecanismo no se puede ensamblar. cuando F es cero la salida será de 180° .

Para el cálculo de la integral (20) solo se tomará un signo del radical de la ecuación (21), el otro signo representa la configuración conjugada del mecanismo, es decir se empleará

$$\psi = 2 \text{ áng } \tan \left[\frac{F + (F^2 - 4EG)^{1/2}}{2E} \right]$$

o bien

$$\psi = 2 \text{ áng } \tan \left[\frac{F - (F^2 - 4EG)^{1/2}}{2E} \right]$$

El método de Romberg para calcular (20) está basado en la extrapolación al límite o extrapolación de Richardson sobre la regla trapezoidal de integración.

$$\text{Sea } T_N(f) = h \sum_{i=1}^{N-1} f_i + \frac{h(f_0 + f_N)}{2} \quad (a)$$

$$\text{donde } h = \frac{b-a}{N} \quad \text{y } f_i = f_{i,N} = f(a+ih) \quad (i=0, 1, \dots, N)$$

la aproximación al número

$$I(f) = \int_a^b f(x) dx \quad (b)$$

si $f(x)$ es diferenciable tres veces

$$I(f) = T_N(f) + c_1 h^2 + O(h^4) \quad (c)$$

c_1 en este caso es independiente de h , su valor está dado por:

$$c_1 = \frac{f'(a) - f'(b)}{12}$$

por lo que se puede aplicar extrapolación al límite, obteniéndose que [8]:

$$T_{N,q} = T_N(f) + \frac{T_N(f) - T_{N/q}(f)}{q^2 - 1}$$

es una aproximación de $O(h^4)$ a $I(f)$, mientras que $T_N(f)$ tiene un error de $O(h^2)$. Generalmente se elige $q=2$ de modo que N debe ser par; por lo tanto

$$T_N(f) = h \sum_{i=1}^{N/2} f[a + (2i-1)h] + h \sum_{i=1}^{N/2-1} f[a + 2ih] + \frac{h[f(a) + f(b)]}{2}$$

la primera suma se extiende sobre los puntos "impares" y la segunda sobre los "pares". Los últimos dos términos de la expresión anterior se pueden escribir como:

$$\left\{ 2h \sum_{j=1}^{N/2-1} f[a + j2h] + \frac{2h[f(a) + f(b)]}{2} \right\} / 2$$

puesto que $2h = 2(b-a)/N = \frac{b-a}{N/2}$

estas dos últimas sumas se suman hasta $T_{N/2}(f)/2$; obteniéndose:

$$T_N(f) = \frac{T_{N/2}(f)}{2} + h \sum_{i=1}^{N/2} f[a + (2i-1)h] \quad (d)$$

Esta expresión es útil para calcular T_N calculados $T_{N/q}$. Si el integrando de (b) tiene $2k+2$ derivadas continuas

$$I(f) = T_N(f) + c_1 h^2 + c_2 h^4 + c_3 h^6 + \dots + c_k h^{2k} + O(h^{2k+2})$$

donde las constantes $c_1, c_2, c_3, \dots, c_k$ no dependen de h , o sea

$$T_N^1(f) = T_N(f) + \frac{T_N(f) - T_{N/2}(f)}{3}$$

(primera extrapolación), se obtiene:

$$I(f) = T_N^1(f) + c'_2 h^4 + c'_3 h^6 + \dots + c'_k h^{2k} + O(h^{2k+2})$$

en la cual, las constantes son independientes de h , extrapolando nuevamente en la siguiente forma:

$$T_N^2(f) = T_N^1(f) + \frac{T_N^1(f) - T_{N/2}^1(f)}{15}$$

teniendo que:

$$I(f) = T_N^2(f) + c''_3 h^6 + c''_4 h^8 + \dots + c''_k h^{2k} + O(h^{2k+2})$$

y en general para $m=1, 2, \dots, k$

$$T_N^m(f) = T_N^{m-1}(f) + \frac{T_N^{m-1}(f) - T_{N/2}^{m-1}(f)}{4^m - 1}$$

es una aproximación de $O(h^{2m+2})$ a $I(f)$. Para que T_N^m este definido se requiere que:

$$M = \frac{N}{2^m}$$

El algoritmo se establece como [8]:

Dada una función $f(x)$ definida sobre $[a, b]$ y un entero positivo M (normalmente 1):

tomar $h = \frac{b-a}{M}$

calcular $T_M^0 = h \sum_{i=1}^{M-1} f(a+ih) + h[f(a) + f(b)]/2$

para $k=1, 2, 3, \dots$

tomar $h = h/2$

calcular $T_{2^k M}^0 = \frac{1}{2} T_{2^{k-1} M}^0 + \frac{h}{2^k} \sum_{i=1}^{2^{k-1} M} f[a + (2i-1)h]$

para $m=1, 2, \dots, k$ hacer:

calcular $T_{2^k M}^m = T_{2^k M}^{m-1} + (T_{2^k M}^{m-1} - T_{2^{k-1} M}^{m-1}) / (4^m - 1)$

Si $f(x)$ tiene $2m+2$ derivadas continuas, entonces:

$$I(f) = \int_a^b f(x) dx = T_{2^k M}^m + O\left[\left(\frac{b-a}{2^k M}\right)^{2m+2}\right] \quad k=m, m+1, \dots$$

y si k es suficientemente grande

$$|I(f) - T_{2^k M}^m| < |T_{2^k M}^m - T_{2^{k-1} M}^{m-1}|$$

antes de depositar fé alguna en esta desigualdad debe comprobarse que:

$$R_k^{m-1} = \frac{T_{2^{k-2} M}^{m-1} - T_{2^{k-1} M}^{m-1}}{T_{2^k M}^{m-1} - T_{2^{k-1} M}^{m-1}} \approx 4^m \quad (e)$$

para tener la mejor aproximación al resultado; (e) indica que la parte de extrapolación que se toma es la dominante y es el valor adecuado para los cálculos de la integral (b).

Proceso de optimización

Al realizar este proceso se eligen valores aleatorios de α , ϕ_0 y ψ_0 , con estos, se calculan los valores de las constantes k_5 , k_6 y k_7 de la ecuación (9), quedando como incógnitas k_1 , k_2 , k_3 , k_4 y k_8 . Estas últimas se determinarán resolviendo el sistema que resulta al sustituir los valores propuestos de ϕ y ψ , así como k_5 , k_6 y k_7 en esta misma ecuación, el sistema resultante es de 5 ecuaciones con 5 incógnitas y se resolverá empleando las subrutinas DECOMP y SOLVE[11].

La subrutina DECOMP descompone la matriz A de coeficientes del sistema en dos matrices, una triangular superior y otra triangular inferior (descomposición LU , o sea, $A = \underline{L}\underline{U}$) mediante el método de Gauss, empleando en el proceso pivoteo parcial, es decir, no empleando un proceso fijo de descomposición, lo que conduciría a grandes errores.

La subrutina SOLVE encuentra la solución al sistema, siempre que este sea no singular, mediante sustitución hacia atrás, o sea $\underline{U}\underline{x} = \underline{L}^{-1}\underline{b}$ donde \underline{b} son los valores del lado derecho del sistema. Para este proceso no es necesario invertir la matriz triangular inferior \underline{L} pues en DECOMP se obtiene directamente al realizarse la descomposición y se almacena de este modo para usos ulteriores.

Determinados los valores de k_1, k_2, k_3, k_4 y k_8 , los parámetros del mecanismo se determinarán de:

$$s_4 = \frac{\frac{k_2}{k_4} \tan \phi_0 + 1}{(k_3/k_4 - \tan \phi_0) \sec \alpha_4}$$

$$a_3 = \frac{s_4 \sin \alpha_4 - \tan \phi_0}{k_4 \cos \phi_0 (1 + \tan \phi_0 \tan \psi_0 \cos \alpha_4)}$$

$$s_1 = \frac{k_2/k_1 + \tan \psi_0}{(k_2/k_1 \tan \psi_0 - 1) \sec \alpha_4}$$

$$a_1 = \frac{s_1 \tan \psi_0 \sin \alpha_4 - 1}{k_1 \cos \psi_0 (1 + \tan \phi_0 \tan \psi_0 \cos \alpha_4)}$$

$$a_2 = \left\{ a_1^2 + a_3^2 + 1 + s_1^2 + s_4^2 + 2s_1s_4 \cos \alpha_4 - k_8 (2a_1a_3 \cos \phi_0 \cos \psi_0) (1 + \tan \phi_0 \tan \psi_0 \cos \alpha_4) \right\}^{1/2}$$

donde se consideró $\alpha_4 = 1$



con estos valores se calculará ahora (19), obteniéndose a lo más 8 raíces diferentes de $\cos \phi$, de aquí, se tomará el ángulo ϕ que satisfaga además (18), puesto que el coseno tiene idénticos valores para ángulos positivos y negativos, si no cumple se le cambiará el signo al ángulo obteniéndose ángulos entre 0° y $\pm 180^\circ$. Determinados los valores de ϕ se calcularán las longitudes (15) y (16), ordenándose de mayor a menor y se compararán con la magnitud de la barra acopladora los valores extremos, si se cumple que:

$$a_2 \geq \text{máx } L$$

$$a_2 \leq \text{mín } L$$

se tendrá movilidad completa en la entrada.

Si lo anterior es satisfactorio se procede a calcular la función objetivo, ecuación (20). Si no, se proponen nuevos valores de α_4 , ϕ_0 y ψ_0 , procediendo de manera similar.

Para el proceso de optimización de la función objetivo se empleará el método de BOX [9], implementado en el paquete "OPTIM" cuyo listado se encuentra en la siguiente página & también puede emplearse el método de Hooke y Jeeves [10] con perturbación aleatoria del óptimo, debido a que puede haber un punto difícil (hang) o en algún óptimo relativo. Ambos métodos son de búsqueda directa (no emplean derivadas de la función objetivo). El primero realiza la búsqueda mediante un modelo, el cual puede ser un tetraedro en 3 dimensiones y se denomina "complex", cada uno de los vértices corresponde a un valor de la función objetivo, refleja por el centroide de la figura el vértice que tenga el peor valor, se continúa con este proceso hasta toparse tal vez con una frontera, si es una frontera implícita, es decir, una limitante no declarada, reduce la mitad de su reflexión y si es explícita toma su valor límite, el proceso prosigue hasta que se obtenga convergencia o se detiene por exceso de búsquedas.

(J A 8 3) C A N D E / C O D E 5 0 0 U N
 = = = = =

FILE 2=FILE5,UNIT=REMOTE 000000

FILE 3=FILE6,UNIT=PRINTER 000000

C <<< OPTIM MODIFICADO >>> 000000

DIMENSION XDEV(5),SX(5,10) 000000

COMMON F,X(5),Y(10),P(20), 000000

1 NX,NY,NP,XL(5),XU(5),YL(10),YU(10), 000000

2 XC(5),XX(5,10),YY(10,10),FF(10),JG,NIT, 000000

3 TITUL(13),FNAME,XNAME(5),YNAME(10),FNAME(20), 000000

4 ALPHA,BETA,KMAX,MAXIT,FR,FA,FDEV,FMIN,MFREQ 000000

C *** INICIA PARAMETROS DEL OPTIM *** 000000

DATA XDEV/5*0.0/,ALPHA,BETA,KMAX/1.30,0.5,10/ 000000

DATA((SX(N1,NK),NK=2,10),N1=1,5)/0.10480,0.22368,0.24130, 000000

1 0.42167,0.57370,0.77921,0.99562,0.96301,0.89579,0.85475, 000000

2 0.28918,0.63553,0.09429,0.10365,0.07119,0.51085,0.02368,0.01011 000000

3 ,0.52162,.07056,.486630,.54164,0.32639,0.29334,0.02488,0.15011, 000000

4 0.46573,.46360,0.93093,0.39975,0.06907,0.72935,0.91977,0.14342, 000000

5 0.36857,.696756,.40961,0.93969,0.61129,0.97336,0.12765,0.21382, 000000

6 0.54092,0.53926,0.97625/ 000000

C *** LEE DATOS BASICOS PARA LA CORRIIDA DE OPTIMIZACION *** 000000

21 CALL LEE(8133) 000000

CALL IMPRE(1) 000000

DO 24 I = 1,NX 000000

24 XC(I) = X(I) 000000

CALL BOX (N1,1,821,827,4) 000000

C *** ESTABLECE SIMPLEX INICIAL *** 000000

27 NIT = 0 000000

KMAX = 2*NX ; K = 1 000000

30 FF(K) = F 000000

DO 32 I = 1,NX 000000

32 XX(I,K) = X(I) 000000

IF (NY) 36,36,34 000000

34 DO 35 I = 1,NY 000000

35 YY(I,K) = Y(I) 000000

36 DO 37 I = 1,NX 000000

37 XC(I) = (XC(I)*(K-1)+X(I))/K 000000

IF (K-KMAX) 39,46,46 000000

39 K = K+1 000000

DO 42 I = 1,NX 000000

42 X(I) = XL(I) + SX(I,K)*(X(I) - XL(I)) 000000

CALL BOX (N1,KMAX1,821,830,1) 000000

C *** COMIENZA LA BUSQUEDA ITERATIVA DEL PUNTO OPTIMO *** 000000

C *** ESTABLECE CONTADOR PARA IMPRESION ITERATIVA *** 000000

46 IF (MFREQ) 47,47,49 000000

47 IPRT = MAXIT + 1 000001

GO TO 55 000001

49 IPRT = MFREQ 000001

WRITE (3,32) IT 000001

CALL IMPRE(3) 000001

C))) ESTABLECE VALOR Y TIPO DE BÚSQUEDA DEL SIMPLEX *** 000001

53 NIT = NIT + 1 ; FR = -1.E10 ; FA = 1.E10 ; IS = 1 ; JL = 1 000001

CALL BOX (3,32) IT 000001

IF (FF(K)-F(4)) 54,54,54 000001

```

60   JG = J ; FMAX = FF(J)                                000001
62   IF (FF(J) - FMIN) 63,65,65                          000001
63   FMIN = FF(J) ; JL = J                               000001
65   CONTINUE                                           000001
C   *** PRUEBA DE CONVERGENCIA ***                       000001
    FDEV = FMAX - FMIN                                   000001
    IF (FDEV - FR*ABS(FMIN) - FA) 71,71,76              000001
C   *** PRUEBA SATISFECHA, EL PROCESO CONVERGE ***      000001
71   CALL IMPRE(1)                                       000001
    WRITE (3,128) NIT                                    000001
    CALL IMPRE(2)                                       000001
    GO TO 21                                             000001
C   ))) PRUEBA NO SATISFECHA, SE PROCEDE A OTRA ITERACION ### 000001
76   IF (NIT - MAXIT) 86,86,78                          000001
C   ))) SE SOBREPASO EL NUMERO DE ITERACIONES PERMITIDO ### 000001
78   CALL IMPRE(1)                                       000001
    WRITE (3,130) NIT                                    000001
    CALL IMPRE(2)                                       000001
    CALL IMPRE(3)                                       000001
    CALL IMPRE(2)                                       000001
    GO TO 21                                             000001
C   ))) CALCULA EL CENTROIDE DE LOS PUNTOS EN SIMPLEX, 000001
C   EXCLUYENDO EL MAXIMO ###                             000001
86   DO 90 I = 1, NX ; XC(I) = 0.                       000001
    DO 89 J = 1, KMAX                                    000001
89   XC(I) = XC(I) + XX(1,J)                            000001
90   XC(I) = (XC(I) - XX(1,JG))/(KMAX - 1)             000001
C   ))) CALCULA NUEVO PUNTO DE PRUEBA RELEJANDO EL     000001
C   MAXIMO MEDIANTE EL CENTROIDE DE LOS PUNTOS RESTANTES ### 000001
    DO 101 I = 1, NX ; X(I) = XC(I) - ALPHA*(XX(1,JG) - XC(I)) 000001
C   ))) PRUEBA CADA VARIABLE EXPLICITA, SI VIOLA LOS LIMITES, 000001
C   SE INTRODUCE A LA R.D.F. A UNA PEQUEÑA DISTANCIA DEL LIMITE ### 000001
    IF (XU(I) - X(I)) 98,98,99                          000001
98   X(I) = XU(I) - XDEV(I)                             000001
99   IF (X(I) - XL(I)) 100,100,101                     000001
100  X(I) = XL(I) + XDEV(I)                             000001
101  CONTINUE                                           000001
C   ))) PRUEBA SI LAS VARIABLES IMPLICITAS VIOLAN LOS LIMITES ### 000001
    CALL BOX (N1, NMAX1, 821, 8106, 2)                 000001
C   ))) PRUEBA PARA VER SI EL PUNTO DE PRUEBA PRODUCE UN MAXIMO 000001
C   EN EL NUEVO SIMPLEX ###                             000001
106  DO 108 J = 1, KMAX                                  000001
    IF (J.NE.JG.AND.FF(J).GT.F) GO TO 114              000001
108  CONTINUE                                           000001
C   ))) YA QUE EL PUNTO DE PRUEBA PRODUCE UN MAXIMO, ADEVELO UNA 000001
C   DISTANCIA BEJA DEL CENTROIDE DE LOS PUNTOS RESTANTES ### 000001
    DO 112 I = 1, NX                                    000001
112  X(I) = XC(I) + BETA*(X(I) - XC(I))                 000001
C   ))) COLOCA EL PUNTO DE PRUEBA EN EL NUEVO SIMPLEX ### 000001
114  CALL BOX (N1, NMAX1, 821, 8115, 3)                 000001
115  DO 116 I = 1, NX                                    000001
116  XX(1,JG) = X(I)                                    000001
    IF (NY) 120,120,115                                000001
118  DO 119 I = 1, NX                                    000001
119  YY(1,JG) = Y(I)                                    000001
120  FF(JG) = F                                          000001
C   ))) IMPRESION INTERMEDIA SI SE REQUIERE ###        000001
    IF (U1 - IFF) 123,123,123                          000001
123  CALL IMPRE(4)                                       000001
    CALL IMPRE(5)                                       000001

```

```
IPRT = IPRT + NFREQ                                000002
GO TO 53                                           000002
128 FORMAT('DEL PROCESO CONVERGE EN ',J4,' ITERACIONES' / 000002
* 'OLA SOLUCION ES :')                             000002
130 FORMAT('DEL PROCESO NO CONVERGE EN ',J4,' ITERACIONES' / 000002
* "OLA SOLUCION DE PRUEBA Y EL SIMPLEX ACTUALES SON :") 000002
132 FORMAT('OITERACION ',J4)                      000002
133 CALL EXIT                                     000002
END                                                000002
```

```

SUBROUTINE IMPRE(1WUE)
  DIMENSION NINI(12)
  COMMON F,X(5),Y(10),P(20),
  1  NX,NY,NP,XL(5),XU(5),YL(10),YU(10),
  2  XC(5),XX(5,10),YY(10,10),FF(10),JG,NIT,
  *  TITUL(13),FNAME,XNAME(5),YNAME(10),PNAME(20),
  *  ALPHA,BETA,KMAX,MAXIT,FR,FA,FDEV,FMIN,NFREQ
  COMMON/UNILS/UNIX(5),UNLY(10),UNIP(20),UNIF
  DATA NINT/1,2,3,4,5,6,7,8,9,10,11,12/
  GO TO (144,147,167,174,180,180) 1WUE
C  ))) IMPRIME ENCABEZADO ###
144  WRITE (3,2000) TITUL
     RETURN
C  ))) IMPRIME SOLUCION DE PUEBA Y LIMITES ###
147  WRITE (3,2002)
     WRITE (3,2003)
     DO 150 I = 1,NX
150  WRITE (3,2001) XNAME(1),3,XL(1),XU(1),X(I),UNIX(1)
     IF (NY) 156,156,152
152  WRITE (3,2005)
     WRITE (3,2003)
     DO 155 I = 1,NY
155  WRITE (3,2001) YNAME(1),3,YL(1),YU(1),Y(I),UNLY(1)
156  IF (NP) 161,161,157
157  WRITE (3,2011)
     WRITE (3,2012)
     DO 160 I = 1,NP
160  WRITE (3,2001) PNAME(1),1,P(1),UNIP(1)
161  WRITE (3,2010)
     WRITE (3,2006)
     WRITE (3,2001) FNAME,3,F,FR,FA,UNIF
     RETURN
C  ))) IMP VALORES DE VARIABLES EN LOS VERTICES DEL SIMPLEX ACTUAL ###
167  KMAX1 = KMAX + 1
     WRITE (3,2007) KMAX1
     WRITE (3,2008) (NINT(I),I = 1,KMAX1)
     DO 172 I = 1,NX
     XX(I,KMAX1) = XC(I)
172  WRITE (3,2004) XNAME(1),(XX(I,K),K = 1,KMAX1)
     IF (NY) 176,176,174
174  DO 175 I = 1,NY
175  WRITE (3,2004) YNAME(1),(YY(I,K),K = 1,KMAX)
176  WRITE (3,2004) FNAME,(FF(K),K = 1,KMAX)
     RETURN
C  ))) IMPRIME LOS RESULTADOS DE LA ITERACION ACTUAL ###
179  WRITE (3,2009) NIT,JG,FDEV,FMIN
180  RETURN
2000  FORMAT ('1',13A6)
2001  FORMAT(A,45,'(3X,1P612.5),4X,A6/)
2002  FORMAT('VARIABLES INDEPENDIENTES')
2003  FORMAT('VARIABLES     LIM. INF.     LIM. SUP.     VALOR
  *  'UNIDADES')
2004  FORMAT(A,45,'(2X,1P612.5)/12X,1P612.5)
2005  FORMAT('VARIABLES DEPENDIENTES')
2006  FORMAT('VARIABLES     VALOR     UNIDADES     UNIDADES
  *  'UNIDADES')
2007  FORMAT('VARIABLES DEL SIMPLEX (DE INICIO = VERTICE 1,NS,1P)
2008  FORMAT('VERTICE'15X,115/19X,15)
2009  FORMAT('ITERACIONES'15X,'VALORES DE F'15X,'VALOR F'15X,

```

```
* 1PG12.5, ' FMIN =', 1PG12.5)
2010 FORMAT('OFUNCION OBJETIVO')
2011 FORMAT('OPARAMETROS')
2012 FORMAT ('ONOMBRE VALOR
END
```

UNIDADES' /)

000003
000003
000003
000003
000003

	SUBROUTINE BOX(N,NMAX,*,*,NTIPO)	000004
C		000004
C))) EN ESTA SUBROUTINA SI LAS RESTRICCIONES IMPLICITAS SON VIOLADAS	000004
C	LA VARIABLES EXPLICITA DE DECISION SE MUEVE LA MITAD DEL	000004
C	CAMINO HACIA EL CENTROIDE ###	000004
C	N = NUMERO DE MOVIMIENTOS DEL PUNTO DE PUEBA HACIA EL CENTROIDE	000004
C	NMAX = NUMERO MAXIMO DE MOVIMIENTOS	000004
C		000004
	COMMON F,X(5),Y(10),P(20),	000004
	1 NX,NY,NP,XL(5),XU(5),YL(10),YU(10),	000004
	2 XC(5),XX(5,10),YY(10,10),FF(10),JG,N11,	000004
	3 TITUL(13),FNAME,XNAME(5),YNAME(10),POME(20),	000004
	4 ALPHA,BETA,NMAX,NAXIT,FR,FA,FDEV,FMIN,FMRES	000004
	N = 1	000004
C))) EVALUA FUNCION OBJETIVO Y VARIABLES DEPENDIENTES ###	000004
	232 CALL MODEL	000004
	IF(NY.LE.0) GO TO 254	000004
C))) PRUEBA SI SE VIOLA ALGUNA RESTRICCION IMPLICITA ###	000004
	DO 237 I = 1,NY	000004
	IF (Y(I).LT.YL(I).OR.YU(I).LT.Y(I)) GO TO 240	000004
	237 CONTINUE	000004
	GO TO 254	000004
C))) COMO EL PUNTO DE PUEBA VIOLA LAS RESTRICCIONES IMPLICITAS,	000004
C	MUEVELO UNA DISTANCIA BETA DEL CENTROIDE DE LOS PUNTOS RESTANTES	000004
	240 DO 242 I = 1,NX	000004
	242 X(I) = XC(I) + BETA*(X(I) - XC(I))	000004
	IF (N = NMAX) 244,248,248	000004
	244 N = N + 1	000004
	GO TO 232	000005
C) EL PUNTO DE PUEBA NO SATISFACE LA RESTRICCION IMPLICITA DESPUES	000005
C	DE NMAX MOVIMIENTOS HACIA EL CENTROIDE DE LOS PUNTOS RESTANTES #	000005
	248 IF(NTIPO.EQ.4) GO TO 251	000005
	CALL IMPRE (2)	000005
	WRITE (3,252)	000005
	250 RETURN 1	000005
	251 CALL FALLA (NTIPO,250)	000005
	252 FORMAT('DEL PUNTO INICIAL VIOLA ALGUNA RESTRICCION')	000005
	254 IF(NTIPO.EQ.4) CALL IMPRE (2)	000005
	RETURN 2	000005
	END	000005

	SUBROUTINE LEE (*)	000003
	COMMON F,X(5),Y(10),P(20),	000004
1	NX,NY,NP,XL(5),XU(5),YL(10),YU(10),	000005
2	XC(5),XX(5,10),YY(10,10),FF(10),JG,NIT,	000006
3	TITUL(13),FNAME,XNAME(5),YNAME(10),PNAME(20),	000007
4	ALPHA,BETA,KMAX,MAXII,FR,FA,FDEV,FMIN,WFREQ	000008
	COMMON/UNITS/UNIX(5),UNY(10),UNIP(20),UNIF	000009
C))) LEE DATOS BASICOS ***	000009
	READ (2,215) TITUL,NX,NY,NP,MAXII,WFREQ	000004
	IF(TITUL(1).NE.6HFIN DE) GO TO 205	000004
	WRITE (3,221)	000004
	RETURN 1	000004
205	NV = 3	000004
	DO 206 I = 1,NX	000004
206	READ (2,216) XNAME(I),NV,XL(I),XU(I),X(I),UNIX(I)	000004
	IF(NY.LE.0) GO TO 210	000004
	NV = 2	000004
	DO 209 I = 1,NY	000004
209	READ (2,216) YNAME(I),NV,YL(I),YU(I),UNY(I)	000004
210	IF(NP.LE.0) GO TO 213	000004
	NV = 1	000004
	DO 212 I = 1,NP	000004
212	READ (2,216) PNAME(I),NV,P(I),UNIP(I)	000004
213	NV = 2	000004
	READ (2,216) FNAME,NV,FR,FA,UNIF	000004
215	FORMAT (13A6/7I5)	000004
216	FORMAT(A5,4X,*(F10.5),A6)	000004
221	FORMAT(/ /20X,10(1H=),' FIN DEL PROBLEMA ',10(1H=))	000004
	RETURN	000004
	END	000004

```
SUBROUTINE FALLA(NTIPO,*)                                000005
  WRITE (3,262) NTIPO                                    000005
  CALL IMPRE(2)                                          000005
  CALL IMPRE(3)                                          000005
  RETURN 1                                              000005
262 *  FORMAT('OERROR EN OPTIMIZACION'/6H TIPO J3,' VIOLACION DE RES 000005
      TRICCION')                                        000005
      END                                              000005
```

El segundo método realiza una exploración sobre cada coordenada, si hay mejora, el nuevo punto será el inicio de la siguiente exploración, pero antes se probará si el doble de la longitud medida a partir del punto anterior, pasando por el actual es una mejora a la función objetivo, si es así, se toma y desecha el actual. Si no hay mejora en ninguna coordenada, el incremento de exploración se reduce a la mitad, continuándose hasta que este incremento sea menor que algún valor especificado, obteniéndose el óptimo.

Los métodos anteriormente descritos no requieren derivadas, son bastante simples en su implantación, pero son lentos para obtener convergencia, si es que esta puede obtenerse. Por la flexibilidad que presenta el método de BOX será el que se emplee.

COMENTARIOS

En este trabajo, la síntesis que se efectúa solo cumple con las posiciones especificadas en los eslabones de entrada y de salida, no siguiendo el orden en que estos son requeridos. Queda abierto el estudio de síntesis con transmisión óptima si se desea que los mecanismos sintetizados cumplan con posición y orden. Una forma de satisfacer este requerimiento es introduciendo condiciones de velocidad, es decir, dar posición y velocidad para cada punto que el mecanismo debe cumplir; cabe recordar que en la velocidad se introducen dos condiciones de posición arbitrariamente próximas.

Bibliografía

- 1.- J. Angeles A., "Matrix Methods in applied kinematics".
D.E.P.F.I. U.N.A.M. 1980 cap. V y I
- 2.- V. K. Gupta, "Kinematic Analysis of plane and spatial Mechanisms" *Journal of Engineering for Industry, Trans. A.S.M.E., series B.* Vol 95 May 1973 pp. 481-486
- 3.- A. V. M. Rao, G. N. Sandor, D. Kohli, A. H. Soni, "Closed form of spatial function generating mechanism for the maximum number of precision points" *Journal of Engineering for Industry, Trans. A.S.M.E., series B.* Vol 95 August 1973 pp. 725-736.
- 4.- F. Freudenstein and G. N. Sandor, "Kinematics of Mechanisms" *Mechanical Design and System Handbook*, H. A. Rothbart, Editor, McGraw-Hill New York 1964. pp 4-5 to 4-7
- 5.- V. K. Gupta and C. W. Radcliffe, "Mobility Analysis of Plane and Spatial Mechanisms", *Journal of Engineering for Industry, Trans. A.S.M.E., series B.* Vol 93 No 1, Feb 1971 pp 125-130
- 6.- V. K. Gupta, "Computer-Aided Synthesis of Mechanisms Using Non-linear Programming", *Journal of Engineering for Industry, Trans. A.S.M.E., series B,* Vol 95, Feb 1973 pp 339-344.

- 7.- G. Dahlquist and A. Björck, "Numerical Methods", Prentice-Hall, Englewood Cliffs, New Jersey 1974, translated by Ned Anderson pp 269-305
- 8.- S. D. Conte, C. de Boor, "Análisis Numérico", Mc Graw Hill, México D.F. 1974. traductor: H. Alfonso Castillo pag. 326-338
- 9.- M. A. Murray-Lasso, E. Chierel U., "Aplicaciones de Computación a la Ingeniería", Limusa, México D.F. 1975 Cap 8 y Apéndice 4.
- 10.- Beveridge y Shechter, "OPTIMIZATION, Theory and Practice", Mc Graw-Hill, International student edition., 1970 pp 363-390
- 11.- G. F. Forsythe y C. B. Moler, "Solución mediante computadoras de sistemas algebraicos lineales", Ed. Universitaria de Buenos Aires, Argentina 1973 pag. 33-94

F-DEPFI/Trabajo/[0001]



722095

F-DEPFI
Trabajo
[0001]