DISEÑO CINEMATICO DE MAQUINARIA 1984

echa	Temario	Horario	Profesor
unio 25	Definiciones y Conceptos Básicos	9 a 13:30 h	Dr.Jorge Angeles Alvare
	Clasificación de Mecanismos	15 a 17:30 h.	
unio 26	Análisis de Mecanismos de Pares inferiores	9 a 13:30 h	Dr. Jorge Angeles Alvar
	Introducción a la Robótica	15 a 17:30 h	
unio 27	Síntesis de Mecanismos de Pares Inferiores	9 a13:30 h	Dr. Jorge Angeles Alvar
	Optimación de Mecanismos	15 a 17:30 h	Dr. Jorge Angeles Alvare
unio 28	Mecanismos de levas	9 a 13:30 h	Dr. Carlos López Cajún
		15 a 17:30 h	
unio 29	Mecanismos de engranes	9 a 13:30 h	M. en I. Angel A. Rojas Salgado
	Ejemplos de aplicación	15 a 17:30 h	

EVALUACION DEL PERSONAL DOCENTE

CURSO: DISEÑO CINEMATICO DE MAQUINARIA

FECHA: Del 25 al 29 de junio de 1984.

	CONFERENCISTA	DOMINIO DEL TEMA	EFICIENCIA EN EL USO DE AYUDAS AUDIOVISUALES	MANTENIMIENTO DEL INTERES. (COMUNICACION CON LOS ASISTENTES, AMENIDAD, FACILIDAD DE EXPRESION).	PUNTUALIDAD	
1.	Dr. Jorge Angeles Alvarez					
2.	Dr. Carlos López Cajún					
3.	M. en I. Angel A. Rojas Salgado					
4.						
5.						
6.						
7.						
8.						
9.						

ESCALA DE EVALUACION: 1 a 10

EVALUACION DE LA ENSEÑANZA

SU EVALUACION SINCERA NOS AYUDARA A MEJORAR LOS PROGRAMAS POSTERIORES QUE DISEÑAREMOS PARA USTED.

TEMA	ORGANIZACION Y DESARROLLO DEL TEMA	GRADO DE PROFUNDIDAD LOGRADO EN EL TEMA	GRADO DE ACTUALIZACION LOGRADO EN EL TEMA	UTILIDAD PRACTICA DEL TEMA	
Definiciones y conceptos básicos					
Clasificación de mecanismos					
Análisis de mecanismos de pares inf.					
Introducción a la robótica					
Síntesis de mecanismos de pares inf.					
Optimización de mecanismos					
Mecnismos de levas					
Ejemplos de aplicación					

ESCALA DE EVALUACION: 1 a 10

EVALUACION DEL CURSO

	CONCEPTO	EVALUACION
1.	APLICACION INMEDIATA DE LOS CONCEPTOS EXPUESTOS	
2.	CLARIDAD CON QUE SE EXPUSIERON LOS TEMAS	
3.	GRADO DE ACTUALIZACION LOGRADO CON EL CURSO	
4.	CUMPLIMIENTO DE LOS OBJETIVOS DEL CURSO	
5.	CONTINUIDAD EN LOS TEMAS DEL CURSO	
6.	CALIDAD DE LAS NOTAS DEL CURSO	
7.	GRADO DE MOTIVACION LOGRADO CON EL CURSO	

<u>ESCALA DE EVALUACION DE 1 A 10</u>

1. ¿Qué le pareció el ambiente en la División de Educación Continua?

MUY AGRADABLE	AGRADABLE	DESAGRADABLE

2. Medio de comunicación por el que se enteró del curso:

PERIODICO EXCELSIOR ANUNCIO TITULADO DIVISION DE EDUCACION CONTINUA	PERIODICO NOVEDADES ANUNCIO TITULADO DIVISION DE EDUCACION CONTINUA	FOLLETO DEL CURSO

CARTEL MENSUAL	RADIO UNIVERSIDAD	COMUNICACION CARTA, TELEFONO, VERBAL, ETC.

REVISTAS TECNICAS	FOLLETO ANUAL	CARTELERA UNAM "LOS UNIVERSITARIOS HOY"	GACETA UNAM

3. Medio de transporte utilizado para venir al Palacio de Minería:

AUTOMOVIL PARTICULAR	METRO	OTRO MEDIO

4. ¿Qué cambios haría usted en el programa para tratar de perfeccionar el curso?

____

____

____

5. ¿Recomendaría el curso a otras personas?

SI	NO

6. ¿Qué cursos le gustaría que ofreciera la División de Educación Continua?

____

____

7. La coordinación académica fue:

EXCELENTE	BUENA	REGULAR	MALA

8. Si está interesado en tomar algún curso <u>intensivo</u> ¿Cuál es el horario más conveniente para usted?

LUNES A VIERNES DE 9 A 13 H. Y DE 14 A 18 H. (CON COMIDAS)	LUNES A VIERNES DE 17 A 21 H.	LUNES, MIERCOLES Y VIERNES DE 18 A 21 H.	MARTES Y JUEVES DE 18 A 21 H.

VIERNES DE 17 A 21 H. SABADOS DE 9 A 14 H.	VIERNES DE 17 A 21 H. SABADOS DE 9 A 13 Y DE 14 a 18 H.	OTRO

9. ¿Qué servicios adicionales desearía que tuviese la División de Educación Continua, para los asistentes? ____________________

____

10. Otras sugerencias:

____

____

____

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO CINEMATICO DE MAQUINARIA

SOFTWARE PARA EL ANALISIS DIGITAL DE SISTEMAS MECANICOS

JORGE ANGELES ALVAREZ
MANUEL CALLEJAS CASTRO

JUNIO, 1984

"SOFTWARE PARA EL ANALISIS DIGITAL DE SISTEMAS MECANICOS"

Jorge Angeles Alvarez
Profesor
División de Estudios de Posgrado
Facultad de Ingeniería
Apartado Postal 70-256
México 20, D.F. MEXICO

Manuel Callejas Castro
Ayudante de Profesor
Departamento de Ingeniería Mecánica
Facultad de Ingeniería, UNAM.

Abstract

Subprograms for digital computer are presented that allow, by their coupling within a main program, the analysis of multiple-loop linkages. Each subprogram analyzes kinematically a single-loop-single-degree-of-freedom linkage. The subprogram developed thus far analyze plane RRRR, RPRR and RRRP linkages. As an example, the analysis of the driving linkage of a mechanical shovel is included.

Resumen

Se presentan subprogramas de computadora digital que permiten, mediante su acoplamiento en un programa principal, el análisis de mecanismos de eslabones rígidos de malla múltiple. Cada subprograma analiza cinemáticamente un mecanismo plano de grado de libertad simple y de una sola malla. Los subprogramas hasta ahora desarrollados analizan mecanismos RRRR, RPRR y RRRP. Como ejemplo, se incluye el análisis del mecanismo accionador de una pala mecánica.

Introducción

El análisis cinemático de sistemas mecánicos mediante computadora digital cobra importancia en el proceso de diseño o de rediseño de tales sistemas, pues a través de este análisis es posible evaluar su operación sin necesidad de construirlo. En efecto, mediante este análisis puede determinarse la evolución de variables tales como máxima aceleración (lineal o angular) de partes críticas, la ventaja mecánica de la transmisión, o bien detectarse situaciones adversas tales como interferencias. Adicionalmente, cabe señalar que el análisis mediante computadora digital permite examinar la operación de sistemas mecánicos con todo el detalle deseado y con tanta precisión como sea necesario, sin tener que construir costosos prototipos, con el ahorro consecuente en economía y en cuanto a tiempo.

Los sistemas mecánicos a los que es aplicable el "software" aquí presentado consisten de acoplamientos de mecanismos planos de grado de libertad simple y de una sola malla.

Existen programas tales como el KAPKA o el IMP [1] que sirven para analizar cinemática y dinámicamente sistemas mecánicos de mallas múltiples y de múltiple grado de libertad. Inclusive, en [2] se consigna ampliamente el software disponible para el propósito mencionado

El objeto que se persigue al desarrollar el software aquí presentado es múltiple. Por un lado, orientarlo hacia la aplicación de técnicas interactivas de graficación. Por otro lado, desarrollarlo en forma modular, lo que permitirá su utilización en forma más eficiente, pues así el diseñador puede contar con una programoteca de la que puede seleccionar sólo los subprogramas que él requiera, sin tener que utilizar memoria de computadora que tendría ociosa, en caso de recurrir a un programa de propósito general. Finalmente, el desarrollo de un software propio es deseable, pues esto contribuye a la creación de una tecnología propia.

Descripción del software

Los subprogramas que se describen a continuación, RESCI1 (RESpuesta CInemática), RESCI2 y RESCI3, sirven para el análisis de mecanismos de los tipos RRRR, RPRR y RRRP, respectivamente. Estos mecanismos se muestran en las Figs. 1-3. Cada uno de esos subprogramas se describe a continuación.

RESCI1. Algoritmo de cálculo

Se supone que se conoce perfectamente los valores $a_1,\ldots,a_5$ y α de la Fig 1, así como la excitación del mecanismo, $\psi=\psi(t)$; así, se conocen también $\dot{\psi}(t)$, $\ddot{\psi}(t)$ y derivadas de orden superior de esta función. El análisis comprende la obtención de la respuesta cinemática del mecanismo, que incluye $\phi(t)$, $\dot{\phi}(t)$, $\ddot{\phi}(t)$, $\theta(t)$, $\dot{\theta}(t)$, $\ddot{\theta}(t)$, $x(t)$, $\dot{x}(t)$, $\ddot{x}(t)$, $y(t)$, $\dot{y}(t)$ $\ddot{y}(t)$, $\mu(t)$, para valores de $\psi(t)$ comprendidos entre $\psi_{mín}$ y $\psi_{máx}$ que, en general, son diferentes y sólo coinciden si el eslabón de entrada, el de longitud a_2, gira vuelta completa.

El ángulo ϕ se obtiene de la ecuación de Freudenstein [3]:

$$k_1-k_2\cos\phi+k_3\cos\psi+\cos(\phi-\psi)=0 \tag{1}$$

donde

$$k_1=\frac{a_3^2-a_1^2-a_2^2-a_4^2}{2a_2a_4}\ ,\ k_2=\frac{a_1}{a_2}\ ,\ k_3=\frac{a_1}{a_4} \tag{2}$$

Escribiendo la ec. (1) en la forma

$$k_1-k_2\cos\phi+k_3\cos\psi=-\cos\phi\cos\psi-\operatorname{sen}\psi\operatorname{sen}\phi \qquad (1')$$

y sustituyendo las siguientes identidades trigonométricas:

$$\operatorname{sen}\phi=\frac{2\tan(\phi/2)}{1+\tan^2(\phi/2)}\ ,\ \cos\phi=\frac{1-\tan^2(\phi/2)}{1+\tan^2(\phi/2)} \qquad (3)$$

la ec. (1') se transforma en

$$A\tan^2(\phi/2)+B\tan(\phi/2)+C=0 \qquad (4)$$

donde

$$A=k_1+k_2-(1-k_3)\cos\psi \qquad (5a)$$

$$B=2\operatorname{sen}\psi \qquad (5b)$$

$$C=k_1-k_2+(1+k_3)\cos\psi \qquad (5c)$$

Así la ec. (4) es cuadrática en $\tan(\phi/2)$, y su solución es, sencillamente,

$$\tan(\phi_{1,2}/2)=\frac{-B\pm\sqrt{B^2-4AC}}{2A} \qquad (6a)$$

de donde

$$\phi_{1,2}=2\tan^{-1}\left(\frac{-B\pm\sqrt{B^2-4AC}}{2A}\right) \qquad (6b)$$

La solución (6a) a la cuadrática (4) es correcta desde el punto de vista algebraico. Sin embargo, numéricamente puede causar dificultades catastróficas, como se apunta en (4). En efecto, si $B^2 >> 4AC$, la primera raíz (tomando el signo positivo en la solución (6a)) se anula a causa del error de redondeo. Para evitar esta situación, se calcula en este caso la primera raíz como

$$\tan(\phi_1/2)=-\frac{B+\sqrt{B^2-4AC}\ \operatorname{sgn}(B)}{2A} \qquad (7a)$$

y la segunda, como

$$\tan(\phi_2/2)=\frac{C}{A\tan(\phi_1/2)} \qquad (7b)$$

Las dos raíces anteriores corresponden a las posiciones conjugadas del mecanismo. En un punto muerto del eslabón de entrada, claramente las dos posiciones conjugadas se reducen a una sola, lo cual sucede cuando se anula el radical. Si este radical no se anula para ningún valor de ϕ, el eslabón de entrada no tiene ninguna posición de punto muerto, esto es, constituye una manivela. En estas condiciones, conviene en primer lugar determinar si este eslabón es del tipo balancín o del tipo manivela. Esto se puede determinar del criterio de Grashoff (5); pero como en este caso interesa conocer, siempre que se trate de un balancín, sus configuraciones extremas, se procede de la siguiente forma: Ya que el mecanismo se encuentra en una configuración extrema al anularse el radical, se debe determinar para qué valores de ψ sucede esto, es decir, es necesario despejar ψ de la ecuación

$$r(\psi)=B^2-4AC=0 \qquad (8)$$

Al sustituir los valores (5a-c) en la ec. (8) se tiene

$$\cos^2\psi+2b\cos\psi+c=0 \qquad (9)$$

donde

$$b=\frac{k_2+k_1k_3}{k_3^2} \qquad (10a)$$

$$c=\frac{k_1^2-k_2^2-1}{k_3^2} \qquad (10b)$$

Las raíces de la ec. (9) son, entonces,

$$\cos\psi_{1,2}=-b\pm\sqrt{b^2-c} \qquad (11)$$

Haciendo la sustitución de b y c en la ecuación (11) en términos de k_1, k_2 y k_3 y en seguida sustituyendo éstas por a_1, a_2, a_3 y a_4 se tiene

$$\cos\psi_{1,2}=\frac{a_1^2+a_2^2-(a_3\mp a_4)^2}{2a_1a_2} \qquad (12)$$

que, como se ve, da lugar a raíces reales. Se tiene, entonces, las siguientes situaciones posibles

i) El valor absoluto de ambas raíces es menor que 1.

ii) Sólo una raíz tiene valor absoluto menor que 1.

iii) Ambas raíces tienen valor absoluto mayor que 1.

En el caso i) el eslabón de entrada es del tipo balancín, estando dadas sus 2 configuraciones extremas por las raíces (12). En el caso ii) este eslabón también es balancín y, suponiendo que la primera raíz tenga valor absoluto menor que 1, las configuraciones extremas están dadas por esta raíz y son simétricas, esto es, $\psi_2=-\psi_1$. En el caso iii), el eslabón es del tipo manivela. Una vez determinadas las configuraciones extremas, se definen $\psi_{mín}$ y $\psi_{máx}$ (para eslabones del tipo balancín) de acuerdo con los valores que adquiera $\psi''(\phi)$ en esas configuraciones, según el criterio conocido de la segunda derivada, esto es, ψ es mínima donde $\psi'(\phi)$ se anula y $\psi''(\phi)>0$; ψ es máxima donde $\psi'(\phi)$ se anula y $\psi''(\phi)<0$.

Los variables $\dot\phi(t)$ y $\ddot\phi(t)$ se obtienen de las fórmulas

$$\dot\phi(t)=\frac{d\phi}{d\psi}\dot\psi \qquad (13)$$

y

$$\ddot\phi(t)=\frac{d^2\phi}{d\psi^2}\dot\psi^2+\frac{d\phi}{d\psi}\ddot\psi \qquad (14)$$

donde las derivadas con respecto a ψ se calculan de la ecuación de Freudenstein (1) que define, bajo las condiciones de existencia del caso (6), a ϕ como función implícita de ψ. En efecto, la ec. (1) se puede escribir como

$$f(\psi,\phi)=0 \tag{15}$$

de donde

$$\frac{d\phi}{d\psi}=-\frac{\partial f/\partial\psi}{\partial f/\partial\phi}=-\frac{N}{D} \tag{16}$$

y

$$\frac{d^2\phi}{d\psi^2}=-\frac{1}{D}\left(\frac{\partial N}{\partial\phi}+\frac{d\phi}{d\psi}\ \frac{\partial D}{\partial\phi}\right)\frac{d\phi}{d\psi}-\frac{\partial}{\partial\psi}\ \frac{N}{D} \tag{17}$$

Las variables $\theta(t)$, $\dot{\theta}(t)$ y $\ddot{\theta}(t)$ se calculan de una ecuación semejante a la de Freudenstein (7):

$$L_1+L_2\cos\psi+L_3\cos\theta-\cos(\phi-\theta)=0 \tag{18}$$

El ángulo $\theta(\psi)$ se calcula en forma análoga a como se calculó $\phi(\psi)$ de las ecs. (7a y b), obteniéndose así dos posiciones conjugadas. Las variables $\dot{\theta}(t)$ y $\ddot{\theta}(t)$ se obtienen, análogamente, como

$$\dot{\theta}=\frac{d\theta}{d\psi}\dot{\psi} \tag{19}$$

y

$$\ddot{\theta}=\frac{d^2\theta}{d\psi^2}\dot{\psi}^2+\frac{d\theta}{d\psi}\ddot{\psi}. \tag{20}$$

donde las derivadas parciales con respecto a ψ se obtienen de la ec. (12) que define análogamente a θ como función implícita de ψ. En efecto, escribiendo la ec. (18) como

$$h(\psi,\theta)=0 \tag{21}$$

se obtiene

$$\frac{d\theta}{d\psi}=-\frac{\partial h/\partial\psi}{\partial h/\partial\theta}=-\frac{N'}{D'} \tag{22}$$

y

$$\frac{d^2\theta}{d\psi^2}=-\frac{1}{D}\left(\frac{\partial N'}{\partial\theta}+\frac{d\theta}{d\psi}\ \frac{\partial D'}{\partial\theta}\right)-\frac{\partial}{\partial\psi}\ \frac{N'}{D'} \tag{23}$$

Las variables $x(t)$, $y(t)$ se calculan de la geometría de la Fig 1, sencillamente como

$$x=a_2\cos\psi+a_5\cos(\theta+\alpha) \tag{24a}$$

$$y=a_2\operatorname{sen}\psi+a_5\operatorname{sen}(\theta+\psi) \tag{24b}$$

calculándose sus derivadas por derivación directa de estas ecuaciones. Las fórmulas correspondientes son fácilmente obtenibles y por la falta de espacio no se incluyen. Finalmente, el ángulo de transmisión se calcula sencillamente de

$$\mu=\phi-\theta \tag{25}$$

RESCI2. Algoritmo de cálculo

Se supone que se conocen perfectamente los valores a_1 y a_2, así como la excitación $s(t)$ y sus derivadas. Se desea determinar $\theta(t)$, $\dot{\theta}(t)$, $\ddot{\theta}(t)$, $\phi(t)$, $\dot{\phi}(t)$ y $\ddot{\phi}(t)$. De la Fig 2 se obtiene

$$s^2=a_1^2+a_2^2+2a_1a_2\cos\phi \tag{26}$$

de donde

$$\phi=\cos^{-1}\left(\frac{s^2-a_1^2-a_2^2}{2a_1a_2}\right) \tag{27}$$

adicionalmente,

$$\theta=\tan^{-1}\left(\frac{a_2\operatorname{sen}\phi}{a_2\cos\phi+a_1}\right) \tag{28}$$

Las derivadas se calculan derivando directamente las relaciones:

$$s\cos\theta=a_2\cos\phi+a_1 \tag{29a}$$

$$s\operatorname{sen}\theta=a_2\operatorname{sen}\phi \tag{29b}$$

Así, se obtiene

$$\dot{\phi}=\frac{-\dot{s}}{a_2\operatorname{sen}(\phi-\theta)} \tag{30}$$

$$\dot{\theta}=\frac{-\dot{s}}{s\tan(\phi-\theta)} \tag{31}$$

$$\ddot{\phi}=\frac{-(s-a_2\cos(\phi-\theta))}{s^2\tan(\phi-\theta)\operatorname{sen}^2(\phi-\theta)}\dot{s}^2-\frac{1}{a_2\operatorname{sen}(\phi-\theta)}\ddot{s} \tag{32}$$

$$\ddot{\theta}=\frac{a_2\cos(\phi-\theta)\left(1+\cos^2(\phi-\theta)\right)-s}{a_2\ s^2\operatorname{sen}^3(\phi-\theta)}\dot{s}^2-\frac{\ddot{s}}{s\tan(\phi-\theta)} \tag{33}$$

RESCI3. Algoritmo de cálculo

Se supone que se conocen perfectamente las dimensiones a_1, a_2, a_3 y a_4, así como el ángulo α del mecanismo RRRP de la Fig 3, además de la excitación $\psi(t)$ y sus derivadas. Se desea calcular $s(t)$, $\dot{s}(t)$, $\ddot{s}(t)$, $\phi(t)$, $\dot{\phi}(t)$, $\ddot{\phi}(t)$, $x(t)$, $\dot{x}(t)$, $\ddot{x}(t)$, $y(t)$, $\dot{y}(t)$ y $\ddot{y}(t)$. De la geometría de la Fig 3,

$$s=a_2\cos\psi+a_3\cos\phi \tag{34a}$$

$$a_1=-a_3\operatorname{sen}\phi-a_2\operatorname{sen}\psi \tag{34b}$$

De la ec. (34b),

$$\phi=\operatorname{sen}^{-1}\left(\frac{-a_1-a_2\operatorname{sen}\psi}{a_3}\right) \tag{35}$$

Si $a_1+a_2\geq a_3$, la manivela oscila de ψ_1 a ψ_2, donde

$$\psi_1=\operatorname{sen}^{-1}\left(\frac{a_3-a_1}{a_2}\right) \tag{36a}$$

y

$$\psi_2=-(\psi_1+180°) \tag{36b}$$

Derivando (34b) con respecto al tiempo,

$$\dot{\phi}=\frac{-a_2\cos\psi}{a_3\cos\phi}\dot{\psi} \tag{37a}$$

Derivando (34a) con respecto al tiempo y sustituyendo (37a) en la expresión así obtenida

$$\dot{s}=-(a_2\operatorname{sen}\psi+a_3\operatorname{sen}\phi)\frac{a_2\cos\psi}{a_3\cos\phi}\dot{\psi} \tag{37b}$$

Derivando las expresiones (37a) y (37b) con respecto al tiempo se obtiene

$$\ddot{\phi} = \frac{a_2^2\cos\psi\,\mathrm{sen}\phi - a_2 a_3^2\,\mathrm{sen}\psi\cos^3\phi}{a_3\cos^3\phi}\,\dot{\psi}^2\;\frac{a_2\cos\psi}{a_3\cos\phi}\,\ddot{\psi} \quad (38a)$$

$$\ddot{s} = \frac{a_2\,\mathrm{sen}\psi\,\mathrm{sen}\phi(a_3^2\cos^2\phi - a_2^2\cos^2\psi) - a_2^2 a_3(\cos^2\psi + \cos^4\phi)}{a_3^2\cos^3\phi}\,\dot{\psi}^2 - \frac{a_2\cos\psi(a_2\,\mathrm{sen}\psi + a_3\,\mathrm{sen}\phi)}{a_3\cos\phi}\,\ddot{\psi} \quad (38b)$$

Las variables $x(t)$, $y(t)$ se calculan de la geometría de la Fig 3, sencillamente como

$$x = a_2\cos\psi + a_4\cos(\phi+\alpha) \quad (39a)$$

$$y = a_2\,\mathrm{sen}\psi + a_4\,\mathrm{sen}(\phi+\alpha) \quad (39b)$$

Calculándose sus derivadas por derivación directa de estas ecuaciones. Las fórmulas correspondientes son facilmente obtenibles y por falta de espacio no se incluyen.

Ejemplo:

Determine $\phi(t)$, $\dot{\phi}(t)$, $\ddot{\phi}(t)$ y $\mu(t)$ del mecanismo accionador de la pala mecanica de la Fig 4. El modelo cinemático y la definición de las variables anteriores se indican en la Fig 5. La entrada del mecanismo es $a(t)=3+0.5\ \mathrm{sen}^2\ \pi t$. Se acoplaron las subrutinas RESC11 y RESC12 en un programa principal y se obtuviron las curvas de la Fig 6.

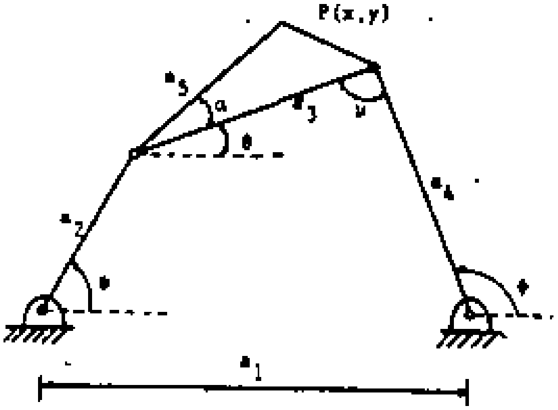


Fig 1. Mecanismo RRRR plano.

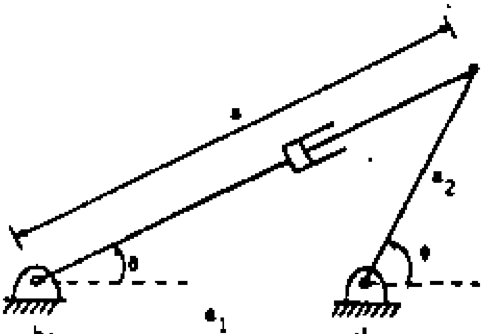


Fig 2. Mecanismo RPRR plano

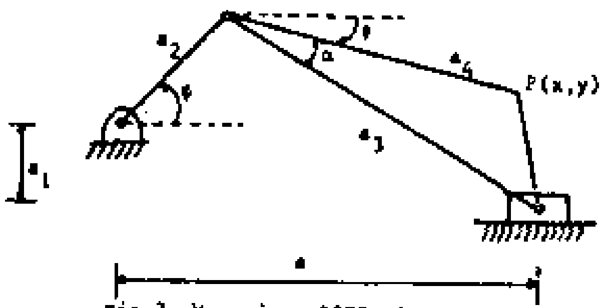


Fig 3. Mecanismo RRRP plano.

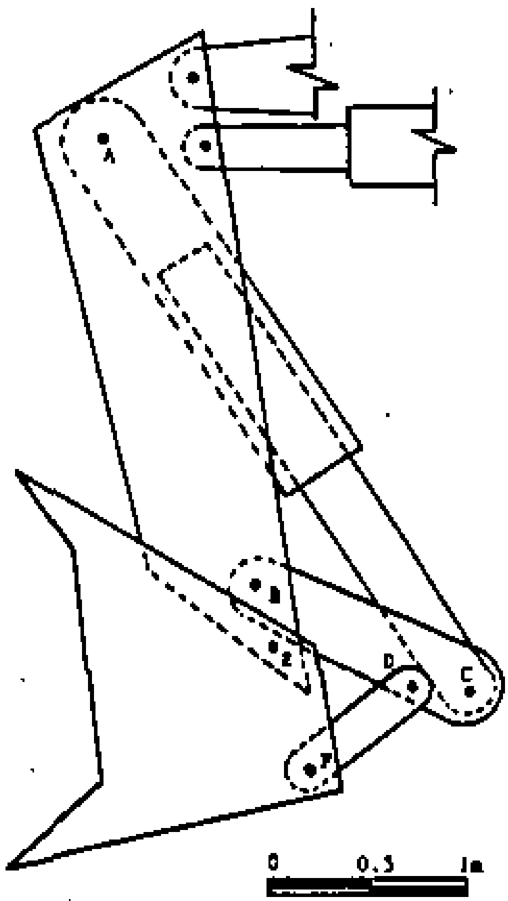


Fig 4. Mecanismo accionador de una pala mecánica.

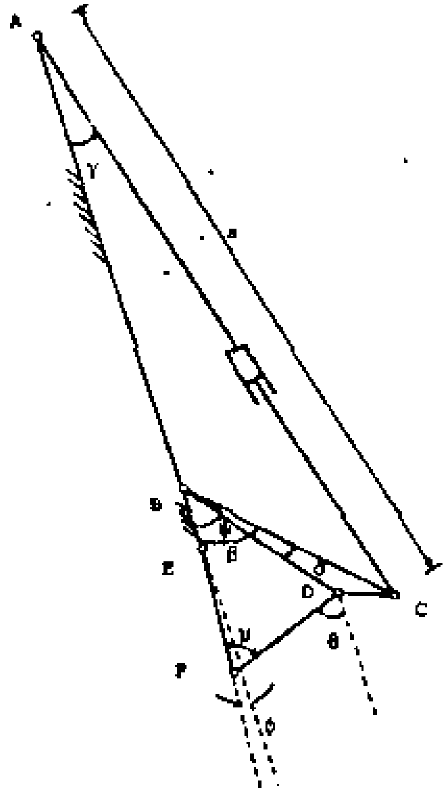


Fig 5. Modelo cinemático del sistema mecánico de la Fig 4.

Referencias

1. Sheth P.N. y Uicker, Jr. J.A., "IMP (Integrated Mechanisms Program): a computer-aided design analysis system for mechanisms and linkages, Trans. ASME, 94, Serie B, J. Eng. Ind., 1972, pp. 454-464

2. Paul B., "Analytical dynamics of mechanisms-A computer oriented overview", Mech. Mach. Theory, Vol. 10, 1975, pp. 481-507.

3. Denavit J. y Hartenberg R.S., Kinematic Synthesis of Linkages, McGraw-Hill Book Co., N. York, 1964, p. 297.

4. Forsythe C.E., Malcolm M.A. y Moler C.B., Computer Method for Mathematical Computations, Prentice-Hall, Englewood Cliffs, N.J., 1977, pp. 20-22.

5. Shigley J.E. y Uicker, Jr. J.J., Theory of Machines and Mechanisms, McGraw-Hill Book Co., N. York, 1980, pp. 16-18.

6. Brand L., Advanced Calculus, John Wiley & Sons, Inc., N. York, 1955, pp. 165-169.

7. Angeles J., Análisis y Síntesis Cinemáticos de Sistemas Mecánicos, Limusa, S.A., C. de México, 1978, pp. 53-60.

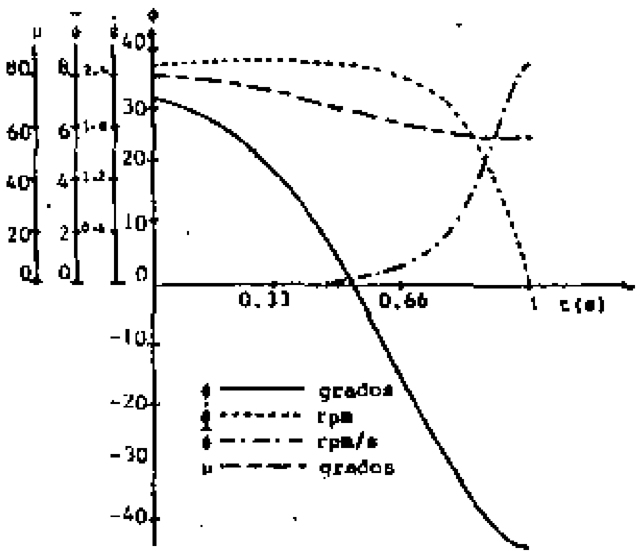


Fig 6. Respuesta cinemática del sistema mecánico de la Fig 4.

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO CINEMATICO DE MAQUINARIA

AN ALGEBRAIC FORMULATION OF GRASHOF'S MOBILITY CRITERIA WITH APPLICATION TO LINKAGE OPTIMIZATION USING GRADIENT-DEPENDENT METHODS

JORGE ANGELES
MANUEL CALLEJAS

JUNIO, 1984

Palacio de Minería Calle de Tacuba 5 primer piso Deleg. Cuauhtémoc 06000 México, D.F. Tel.: 521-40-20 Apdo. Postal M-2285

AN ALGEBRAIC FORMULATION OF GRASHOF'S MOBILITY CRITERIA WITH APPLICATION TO LINKAGE OPTIMIZATION USING GRADIENT-DEPENDENT METHODS

Jorge Angeles, Professor[1] (ASME Member 79)

Manuel Callejas, Research Assistant[1]

Abstract

Sets of constraints on analytic functions of linkage parameters, equivalent to those of Grashof's mobility criteria are derived. These inequalities represent necessary and sufficient mobility conditions for the input and the output links of planar 4R linkages, as well as for its coupler link. The application of the foregoing constraint relations to linkage optimization using gradient -dependent methods is shown with an example that is fully solved resorting to Newton-Raphson's method and an interior penalty function.

[1] Universidad Nacional Autónoma de México
Apdo. Postal 70-256
C. Universitaria
04510 México, D.F. MEXICO.

Introduction

Grashof's mobility criteria for RRRR plane linkages establish the conditions on the relative magnitudes of the links for the existence of double-crank, crank-rocker and double-rocker linkages. The proof of such criteria can be seen in either [1] or [2]. A recent reassessment of such criteria was given by Paul [3], who proved necessity and sufficiency of those. Paul also showed that two types of double-crank linkages should be distinguished, namely those with fully-revolving couplers and those with oscillating ones. The mobility of the coupler link is also analyzed in the present paper. Indeed, necessary and sufficient conditions for a fully-revolving and for an oscillating coupler are derived. Litvin [4] has established general conditions in algebraic form for the existence of cranks in closed kinematic chains of any type. He does not provide, however, specific formulae for specific types of linkages.

Grashof's conditions for the existence of the aforementioned types of linkages take the form of inequalities in which the lengths of the different links appear linearly. In synthesizing RRRR plane linkages for function generation, however, a system of linear equations not on the linkage lengths, but on a different set of parameters, arising from Freudenstein's equations, is to be solved. Given the RRRR plane linkage of Fig 1, let

$$k_1 = \frac{a_1^2 + a_2^2 - a_3^2 + a_4^2}{2a_2a_4}\ , \ k_2 = \frac{a_1}{a_2}\ , \ k_3 = \frac{a_1}{a_4} \qquad (1)$$

The problem of linkage synthesis for function generation consists of finding a set of values $\{a_1, a_2, a_3, a_4\}$ for the link lengths that produce a prescribed set of input-output pairs $\{(\psi_i, \phi_i)\}_1^n$, where ψ and ϕ are the input and the output angles, respectively.

Freudenstein's equation [5;6, p 297] allows one to compute the set $\{k_1, k_2, k_3\}$ for the prescribed input-putput values. This equation is the following:

$$k_1 + k_2 \cos\phi_i - k_3 \cos\psi_i = \cos(\psi_i - \phi_i), \quad i = 1, \ldots, n \tag{2}$$

Since the problem contains three unknowns, three input-output values can be prescribed, eq (2) thus leading to a linear algebraic system of the form

$$Ak = b \tag{3}$$

where A is a 3x3 matrix, k and b being the 3-dimensional vectors given next:

$$A = \begin{bmatrix} 1 & \cos\phi_1 & -\cos\psi_1 \\ 1 & \cos\phi_2 & -\cos\psi_2 \\ 1 & \cos\phi_3 & -\cos\psi_3 \end{bmatrix}, \quad k = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix}, \quad b = \begin{bmatrix} \cos(\psi_1-\phi_1) \\ \cos(\psi_2-\phi_2) \\ \cos(\psi_3-\phi_3) \end{bmatrix} \tag{4}$$

By inverting the system of equations (3) one can obtain the unique set of values $\{k_1, k_2, k_3\}$ that solve the proposed problem. Given the simple structure of matrix A and the low number of equations, matrix A can be inverted[2] explicitly. In fact, formulae are available in the literature [6, p 298] for the computation of k from (3). Once the value of k has been computed, the lengths a_i can be computed by inversion of eqs (1) as:

[2] Here it is assumed that the synthesis problem leading to eqs (1) is well posed, i e cases rendering either matrix $\underset{\sim}{A}$ singular or vector $\underset{\sim}{b}$ zero are discarded.

$$a_1 = 1,\ a_2 = \frac{1}{k_2},\ a_3 = \pm \frac{[k_2^2 + k_3^2 + k_2^2 k_3^2 - 2k_1k_2k_3]^{1/2}}{k_2k_3},\ a_4 = \frac{1}{k_3} \quad (5)$$

In the above discussion nothing prevents k_2 and k_3 from resulting negative, thus producing negative values for either a_2 or a_4, which situation is next dealt with; a_1 is arbitrarily chosen unity, for a scaling of all lengths by the same factor does not alter the input-output relationship, whereas a_3 can always be made positive by a proper choice of the sign of the square root. Negative values of either a_2 or a_4 indicate that the angle ψ or, correspondingly, ϕ , should be measured to an extension of either the input or the output link as shown in Fig 2. If it is necessary to verify Grashof's conditions within the synthesis process, then a_2 and a_4 should be computed using the absolute values of k_2 and k_3. Introducing such absolute values, however, removes the smoothness of one side of Grashof's inequalities, which might be undesirable if the foregoing computations are to be performed within an optimization procedure requiring the computation of the gradient of that side of the inequalities. An alternative approach [7] consists of determining lower and upper bounds for the length of the coupler link if either the input or the output link, or both, is to be a crank. For RRRR plane linkages, however, this approach leads to Grashof's unsmooth inequalities. Waldron [8] and Waldron and Stevensen [9] have proposed alternate approaches based on graphical methods, whereas Gupta has proposed sufficient algebraic smooth inequalities [10, 11] guaranteeing the existence of an input link of the crank type. Necessary and sufficient algebraic smooth inequalities have been proposed for the existence of an input crank [12].

In what follows, a set of smooth inequalities is obtained, that is necessary and sufficient to produce an input, an output or a coupler link of either type, crank or rocker. This set of inequalities is meant to be adjoined to optimization programs to produce linkages whose links be of a given type, while meeting prescribed functional requirements for the classical problems of linkage synthesis, namely rigid-body guidance, path generation and function generation. By incorporating further requirements on the transmission angle, additionally, as discussed in [13], the aforementioned problems of linkage synthesis can be treated as nonlinear programming problems, which can be readily solved using standard optimization packages, normally available in any program library.

Derivation of mobility conditions

First a set of mobility conditions for the input link is derived. To do this, indices are dropped from Freudenstein's equation for simplicity. Next, the following identities are introduced:

$$\cos\phi = \frac{1 - \tan^2(\phi/2)}{1 + \tan^2(\phi/2)}, \quad \sin\phi = \frac{2\tan(\phi/2)}{1 + \tan^2(\phi/2)} \tag{6}$$

Freudenstein's equation is thus transformed into

$$A \tan^2(\phi/2) - 2B \tan(\phi/2) + C = 0 \tag{7}$$

where

$$A = k_1 - k_2 + (1-k_3)\cos\psi \tag{8a}$$

$$B = \sin\psi \tag{8b}$$

$$C = k_1 + k_2 - (1+k_3)\cos\psi \tag{8c}$$

Eq (7), then, defines a quadratic equation in $\tan(\phi/2)$ for a given value of ψ. If none of its coefficients vanishes, this equation produces two values of ϕ, given by

$$\phi_{1,2} = 2\tan^{-1}\left[\frac{B \pm (B^2 - AC)^{1/2}}{A}\right] \tag{9a}$$

The two values of $\phi(\psi)$ given above can be: i) both complex, in which case the input link is of the rocker type, the corresponding value of ψ lying outside of the mobility range of this link, ii) both real and distinct, thus corresponding to the two conjugate configurations of the linkage, or iii) both real and identical to each other, in which case ψ attains an extremum value, i.e. the output link attains a dead-point position.

Now, if $A(\psi)=0$ and $B(\psi)\neq 0$, the set $\{k_1,k_2,k_3\}$ must observe the following relationship:

$$|k_1 - k_2| \leq |1 - k_3|$$

In this case, the left-hand side of eq (7) degenerates into a line. Viewed this line as a particular case of the general quadratic function, its *two intersections* with the $\tan(\phi/2)$ - axis can be thought of lying at $\tan(\phi/2) = C/2B$ and at infinity, which thus produces the two following values for corresponding conjugate configurations:

$$\phi_1 = \tan^{-1}(C/2B), \quad \phi_2 = \pi \tag{9b}$$

The two values of ϕ appearing in (9b) can also be derived formally by taking the limit, as $A \to 0$, of both values appearing in (9a).

Should both A and B vanish simultaneously, then C would necessarily vanish, as well, in which case, for this particular value of ψ, eq (7) would hold identically, *for any value of* ϕ. The zeroing of B implies, from eq (8b), $\psi = 0$ or π, which then yields $k_1 = k_3$ and $k_2 = 1$, or $k_1 = -k_3$ and $k_2 = -1$, respectively, in view of eqs (8a & c). In either case $a_2^2 = a_1^2$ and $a_4^2 = a_3^2$, i.e. the linkage is of the change-point type and either at $\psi = 0$ or $\psi = \pi$, the input and the fixed links are coincident, as well as the coupler and the output links. The linkage thus degenerates into a two-link open chain, for which ϕ can attain any real value, as predicted by eq (7).

Furthermore, if the discriminant of eq (7), i.e. the radical of expression (9a) is negative for all real values of ψ, then the four lengths a_1, a_2, a_3 & a_4 *do not define a closed quadrilateron.*

In establishing Grashof's mobility criteria, it is always assumed that the linkage lengths define a quadrilateron, that is to say, each length is smaller than the sum of the remaining ones. Such closure condition is next derived in terms of the set $\{k_1, k_2, k_3\}$. This is more easily done if the non-closing condition is established, instead. The set $\{k_1, k_2, k_3\}$ will define a set of link lengths that does not constitute a quadrilateron if the radical of eq (9a) is negative for all real values of ψ, as already remarked. A rigorous analysis of this situation, that is not presented here due to space limitations, leads to the following set of inequalities:

$$(k_2 - k_1k_3)^2 \geq k_3^4 \;,\; (k_1^2 - k_2^2 + k_3^2 - 1)^2 > 4(k_2 - k_1k_3)^2 \;,\; k_1^2 - k_2^2 - 1 > k_3^2 \qquad (10a)$$

$$(k_2 - k_1k_3)^2 \leq k_3^4 \;,\; (k_2 - k_1k_3)^2 < k_3^2(k_1^2 - k_2^2 - 1) \qquad (10b)$$

The following has then been proved: *If the set $\{k_1, k_2, k_3\}$, as given by eqs (1), verifies either set of inequalities, (10a) or (10b), then the link lengths $a_i (i = 1, \ldots, 4)$ do not define a linkage.*

It can be readily verified that if the second relation (10b) holds, then a_3, as given by eq (5), turns to be imaginary. The non-existence of a linkage can thus be due to either of two possibilities: i) One of its links turns to have an imaginary length (if all three k's in (5) are real, only a_3 can have an imaginary value); ii) Even though all four lengths are real, they do not meet the closure condition.

Now, if the k's are obtained as the solution of eqs (2) with n = 3, then the said k's will yield a closing real quadrilateron, i.e.neither of relations (10) will hold. Although this is not formally proved here, this fact can be realized from the continuity of both sides of Freudenstein's equation, the real-valuedness of $\{(\psi_i, \phi_i)\}_1^3$ and the (assumed) nonsingularity of matrix A appearing in (4).

The application of formulae (9) to obtain the real values ϕ_1, ϕ_2 corresponding to one single value of ψ should be made taking into account possible cancellations due to round-off errors. This can be taken care of if formula (9) is rewritten in a form that is more suitable for numerical stability, as indicated in [14]. This topic, however, is not further discussed here for it falls without the scope of the paper.

The two real values obtained from eq (9) are distinct, except at dead-point positions of the input link. Those distinct values correspond to the two conjugate configurations of the linkage. Next, conditions for the full rotatability of the input link are obtained. To this end, the discriminant of eq (9a) is expanded and then zeroed. This leads to

$$\cos^2\psi + 2b\cos\psi + c = 0 \tag{11a}$$

with

$$b = \frac{k_2 - k_1 k_3}{k_3^2}, \quad c = \frac{k_1^2 - k_2^2 - 1}{k_3^2} \tag{11b}$$

The real roots $\cos\psi_1$, $\cos\psi_2$, of eq (11a) then yield the two extremal values of ψ. These two roots are always real if a_3 is, but nothing prevents them from having absolute values larger than unity. The roots of eq (11a) are, in fact,

$$\cos\psi_{1,2} = -b \pm (b^2 - c)^{1/2} \tag{12a}$$

which, in terms of the link lengths, produce

$$\cos\psi_{1,2} = \frac{a_1^2 + a_2^2 - (a_3 \pm a_4)^2}{2a_1 2a_2} \tag{12b}$$

from which it is clear that both roots are real if a_3 is. Now, if the input link is to be of the crank type, then these two roots should yield complex values of ψ_1 and ψ_2, which is only possible if both roots (12a) have absolute values larger than unity. This is exactly the necessary and sufficient condition for an input link to be of the crank type, provided the k's were obtained from a well-posed synthesis problem. This condition is, then

$$|-b \pm (b^2 - c)^{1/2}| > 1 \tag{13}$$

which, if squared, still holds, for both sides of (13) are positive. Thus, it is equivalent to the following two inequalities:

$$2b^2 - c - 1 > 2b\ (b^2 - c)^{1/2}, 2b^2 - c - 1 > -2b\ (b^2 - c)^{1/2} \qquad (14)$$

which are in turn equivalent to the single one given next:

$$2b^2 - c - 1 > 2|b|(b^2 - c)^{1/2} \qquad (15)$$

which requires that its left-hand side be positive, i e

$$2b^2 - c - 1 > 0 \qquad (16)$$

Now, given inequality (16), relation (15) still holds if its both sides are squared. This leads to

$$(c + 1)^2 - 4b^2 > 0 \qquad (17)$$

If definitions (11b) are recalled and substituted into relations (16) and (17), the next two inequalities are obtained, in the space of k_1, k_2, k_3:

$$2\ (k_2 - k_1k_3)^2 - k_3^2\ (k_1^2 - k_2^2 + k_3^2 - 1) > 0 \qquad (18\ a)$$

$$[(k_1 - k_3)^2 - (k_2 - 1)^2]\ [(k_1 + k_3)^2 - (k_2 + 1)^2] > 0 \qquad (18\ b)$$

Summarizing, then, one has proved that: *The necessary and sufficient conditions for the synthesis problem leading to eqs (2) to produce an input crank is that both inequalities (18 a and b) hold.*

The full rotatability conditions for the output crank can be obtained analogously. It is far simpler, however, in view of the symmetry of definitions (11b) with respect to a_2 and a_4 in Freudenstein's equation, to exchange the roles of variables k_2 and $-k_3$ in inequalities (18 a and b). This yields

$$2\ (-k_3 + k_1k_2)^2 - k_2^2\ (k_1^2 + k_2^2 - k_3^2 - 1) > 0 \qquad (19\ a)$$

$$[(k_1 + k_2)^2 - (-k_3 - 1)^2]\ [(k_1 - k_2)^2 - (-k_3 + 1)^2] > 0 \qquad (19\ b)$$

One then has: *The necessary and sufficient conditions for the synthesis problem leading to eqs (2) to produce an output crank is that both inequalities (19 a and b) hold.*

Conditions for the existence of rockers are next derived. One possible way of establishing them is saying that "an input link is of the rocker type if inequalities 18 a and b) do not hold simultaneously, whereas an output link is of the rocker type of inequalities (19 a and b) do not hold simultaneously". Since the violation of the said inequalities presents various alternatives, it does not guarantee the existence of a rocker link. An alternative approach, specifying the extremal values of the variable of interest, either ψ or ϕ, is presented next.

If an input rocker is required to have mobility only within the range $\psi_1 \leq \psi \leq \psi_2$, then the discriminant of eq (9a) is zeroed at these values, i.e.

$$\cos^2 \psi_i + 2 b \cos \psi_i + c = 0, \quad i = 1,2 \tag{20}$$

The two given roots of eq (20) satisfy

$$b = -\frac{1}{2} (\cos\psi_1 + \cos\psi_2), \quad c = \cos\psi_1 \quad \cos\psi_2 \tag{21}$$

In terms of k_1, k_2 and k_3, eqs (21) lead to

$$2(k_2 - k_1 k_3) + k_3^2 (\cos\psi_1 + \cos\psi_2) = 0 \tag{22a}$$

and

$$k_1^2 - k_2^2 - 1 - k_3^2 \cos\psi_1 \cos\psi_2 = 0 \tag{22b}$$

Eqs (22) constitute a nonlinear algebraic system in two equations and three unknowns, i.e. it is underdetermined. These are subject to the second order constraints guaranteeing that ψ_1 be a minimum and ψ_2 be a maximum, which are

$$\psi''(\phi_1) \geq 0, \ \psi''(\phi_2) \leq 0 \tag{23}$$

where, ϕ_i , defined as $\phi(\psi_i)$ can be obtained by substitution in eq (7). At a stationary value of ψ, ψ' vanishes, $\psi''(\phi)$ reducing to

$$\psi''(\phi) = \frac{(A^2 + B^2)^2 A}{2 G^2} \tag{24a}$$

with A and B given as in definitions (8), and

$$G = [B^2 (1 - k_3) - A^2 (1 + k_3)] \sin \psi + 2 AB \cos \psi \tag{24b}$$

Relations (23) then can be expressed as

$$A (\psi_1) \geq 0, \ \ A (\psi_2) \leq 0 \tag{25a}$$

or, from definition (8a), in terms of k_1, k_2 and k_3, as

$$k_1 - k_2 + (1 - k_3) \cos\psi_1 \geq 0 \tag{25b}$$

and

$$k_1 - k_2 + (1 - k_3) \cos\psi_2 \leq 0 \tag{25c}$$

Relations (22a and b) and (25b & c) alone do not allow the computation of k. These should be incorporated into an optimization problem, e.g. one minimizing a norm of the structural error or maximizing a norm of the mechanical advantage within the range of motion.

Now, if an output rocker is required, whose motion be defined in the interval $\phi_1 \leq \phi \leq \phi_2$, this can be accomplished paralleling the foregoing procedure. It is far simpler, however, to derive the corresponding relations by exchanging the roles of k_2 and $-k_3$ in relations (23a & b) and (25b & c). This produces

$$2(k_1k_2 - k_3) + k_2^2 (\cos\phi_1 + \cos\phi_2) = 0 \tag{26a}$$

$$k_1^2 - k_3^2 - 1 - k_2^2 \cos\phi_1 \cos\phi_2 = 0 \tag{26b}$$

and

$$k_1 + k_3 + (1 + k_2) \cos\phi_1 \geq 0 \tag{27a}$$

$$k_1 + k_3 + (1 + k_2) \cos\phi_2 \leq 0 \tag{27b}$$

Mobility conditions for the coupler are next derived. An analysis similar to the one leading to Freudenstein's equation yields

$$m_1 + m_2 \cos\theta + m_3 \cos\psi = \cos(\psi - \theta) \tag{28}$$

with

$$m_1 = \frac{a_4^2 - a_1^2 - a_2^2 - a_3^2}{2\, a_2\, a_3}\ , \quad m_2 = \frac{a_1}{a_2}\ , \quad m_3 = \frac{a_1}{a_3} \tag{29}$$

Substitution of identities (6) for angle Ψ in the latter equation yields

$$J \tan^2(\psi/2) \quad - 2K \tan(\Psi/2) + L = 0 \tag{30}$$

with

$$J = m_1 + m_2\cos\theta-(m_3-\cos\theta) \tag{31a}$$

$$K = \sin\theta \tag{31b}$$

$$L = m_1 + m_2\cos\theta+ m_3 -\cos\theta \tag{31c}$$

One then has, for a given value of θ, from eq (30),

$$\psi_{1,2} = 2 \tan^{-1} \left[\frac{K \pm (K^2 - JL)^{1/2}}{J} \right] \tag{32}$$

Eq (32) thus yields two different values of ψ, corresponding to the conjugate configurations of the linkage, except at extremal positions of the coupler, where the radical vanishes. If the coupler is to have full rotatability, the radical should not vanish for real values of θ. The conditions under which this happens are derived paralleling the procedure leading to relations (18a & b), which produces the following set of inequalities:

$$1 + m_1^2 - m_2^2 + m_3^2 > 0 \tag{32a}$$

$$(1 - m_1^2 - m_2^2 + m_3^2)^2 - 4m_1^2 > 0 \tag{32b}$$

One then has proved: *The well-posed synthesis problem producing* m_1, m_2 *and* m_3 *from eq (28) yields a coupler link possessing full rotatability if, and only if, relations (32a & b) hold.*

Mobility conditions for the coupler, considering its motion with respect to the output link, are derived analogously. These are obtained from the equation

$$n_1 + n_2\cos\theta - n_3\cos\phi = \cos(\phi - \theta) \tag{33}$$

with

$$n_1 = \frac{a_1^2 - a_2^2 + a_3^2 + a_4^2}{2a_3a_4}, \qquad n_2 = \frac{a_1}{a_4}, \quad n_3 = \frac{a_1}{a_3} \tag{34}$$

Eq (33) leads to

$$M \tan^2(\psi/2) - 2N \tan(\psi/2) + P = 0 \tag{34}$$

where

where

$$M = n_1 + n_2\cos\theta + n_3 + \cos\theta \tag{35a}$$

$$N = \sin\theta \tag{35b}$$

$$P = n_1 + n_2\cos\theta - (n_3 + \cos\theta) \tag{35c}$$

The roots of eq (34) are, thus,

$$\psi_{1,2} = 2\tan^{-1}\left[\frac{N \pm (N^2 - MP)^{1/2}}{M}\right] \tag{36}$$

The conditions sought are then derived from the zeroing of the radical of eq (36). This leads to the following set of inequalities:

$$1 + n_1^2 - n_2^2 + n_3^2 > 0 \tag{37a}$$

$$(1 - n_1^2 - n_2^2 + n_3^2)^2 - 4n_1^2 > 0 \tag{37b}$$

That is: *The coupler link of a linkage whose lengths are derived from eq (33) has full rotatability with respect to the output link if, and only if, its lengths satisfy relations (37a & b).*

What is meant in the last paragraph under full rotatability of the coupler with respect to the output link is that $\theta'(\phi)$ does not vanish, for any real value of ϕ. This does not mean, however, that this is equivalent to full rotatability of the coupler link. In fact, if the output link does not possess full rotatability, then $\phi'(\psi)$ vanishes for two distinct values of ψ. Hence, even if $\theta'(\phi)$ does not vanish, $\theta'(\psi) = \theta'(\phi)\phi'(\psi)$ does, the coupler link thus lacking full rotatability.

On the other side, conditions for oscillating couplers can be obtained paralleling the procedure followed for the input and the output links. The obtention of such conditions is straightforward from the foregoing analysis, for which reason the subject is not further discussed here. Finally, change-point mechanisms are characterized within this context as those for which the left-hand side of at least one of inequalities (18a) or (18b) (or, equivalently (19a) or (19b)) vanishes.

The inequalities derived in this paper are now used as constraints of an optimization problem of linkage synthesis. This is solved using Newton-Raphson's method, which requires not only first, but also second derivatives of both the objective function and the constraints. The procedure is illustrated with one fully solved example.

Example

Synthesize a RRRR plane linkage, as the one shown in Fig 1, to produce the input-output relation appearing in Table 1. This linkage should approximate the synthesis equations with the least possible r.m.s. error, while its input link should be a crank.

This problem was solved in [15] without considereing the crank-type restriction. The least-square error linkage thus obtained turned to be of the rocker-rocker type.

The synthesis problem at hand is formulated as follows: The synthesis equations are of the type of eq (3), except that A is now a 5 x 3-matrix, whereas vector b is 5-dimensional. The i^{th} row of matrix A and the i^{th} component of vector b are, respectively,

$$A_i = [\ 1,\ \cos\phi_i,\ -\cos\psi_i],\quad b_i = \cos(\psi_i - \phi_i) \tag{38}$$

This problem thus leads to an overdetermined system of linear equations which, in general, has no exact solution. In this case, a vector k is sought that minimizes a norm of the error e, defined as

$$e = Ak - b \tag{39}$$

which is clearly a 5-dimensional vector. If the Euclidean norm is to be minimized, without imposing any further constraint, then the minimizing value of k can be expressed explicitly in terms of the Moore-Penrose generalized inverse [16], which can be computed very efficiently using Householder reflections as already shown in [17, 18]. Should a further constraint be imposed on the synthesis problem, then the above-mentioned generalized inverse is not applicable any more. A possible way of solving this problem is via a penalty function [19], which is next introduced. The problem is now formulated as: "Minimize, under k, the objective function z given as

$$z = \frac{1}{2}(Ak - b)^T(Ak - b) \tag{40}$$

i.e. half the square of the Euclidean norm of the error, subject to inequalities (18a & b), whose holding is necessary and sufficient for a crank-type input link".

Solution:

Let f_1 and f_2 represent the left-hand sides of inequalities (18a & b), respectively. Now assume a feasible linkage is given, i.e. one defined by a particular vector k^0 that satisfies both given inequalities, which in general is not optimal. This linkage can be improved by minimizing the new objetive function

$$\psi(k ; r_1 = z(k) + r_1\left(\frac{1}{f_1} + \frac{1}{f_2}\right) \tag{41}$$

subject to no further constraints.

The second term of the right-hand side of eq (41) is referred to as the penalty term. This is the product of the positive weighting factor r_1 times the sum in parenthesis, referred to as the penalty function. Criteria for selecting a suitable value of r_1 are given in [19, pp 156-196], but the simplest one is to choose it so as to render the penalty term a given fraction of $z(k^0)$. Next, the value of k minimizing ψ, k^1, is used as a "guess" value of k to minimize a new objective function with a new penalizing factor, r_2, a fraction of r_1. The procedure is repeated a few number of times, say p, which produces an equal number of pairs $\{(k^i , r_i)\}_1^p$ with $r_i > r_{i+1} > 0$; these can then be fitted to a suitable function, as shown next, the solution to the original constrained problem being obtained by extrapolation, with $r \to 0$.

Since both function z and the penalty term are infinitely many times differentiable, each unconstrained optimization problem meant to minimize $\psi_i = \psi(k ; r_i)$, for $i = 1, \ldots, p$ can be solved using a gradient method or even the Newton-Raphson method [20, pp 249-251].In any instance, the roots of the gradient of ψ_i with respect to k, are to be computed. The said gradient is given as

$$\nabla\psi_i = A^T(Ak - b) - r_i\left(\frac{\nabla f_1}{f_1^2} + \frac{\nabla f_2}{f_2^2} \right); \quad i = 1, \ldots, p \tag{42}$$

with

$$\nabla f_1 = 2 \begin{bmatrix} (k_1k_3 - 2k_2)k_3 \\ 2(k_2 - k_1k_3) + k_2k_3^2 \\ (k_1^2 + k_2^2 - 2k_3^2 + 1)k_3 - 2k_1k_2 \end{bmatrix} \tag{43}$$

$$\nabla f_2 = 2 \left(\begin{bmatrix} k_1 - k_3 \\ 1 - k_2 \\ k_3 - k_1 \end{bmatrix} [(k_1 + k_3)^2 - (k_2 + 1)^2] + \right.$$

$$\left. + \begin{bmatrix} k_1 + k_3 \\ -k_2 - 1 \\ k_1 + k_3 \end{bmatrix} \cdot [(k_1 - k_3)^2 - (k_2 - 1)^2] \right) \tag{44}$$

The application of Newton-Raphson's method to the computation of the roots of $\nabla\psi_i$ requires computing the Jacobian matrix J of $\nabla\psi_i$, i.e. $\nabla^2\psi_i$, with respect to k. This is readily computed as

$$\nabla^2\psi_i = A^TA - r_i \left[\frac{f_1^2 \nabla^2 f_1 - 2f_1\nabla f_1(\nabla f_1)^T}{f_1^4} + \frac{f_2^2 \nabla^2 f_2 - 2f_2\nabla f_2(\nabla f_2)^T}{f_2^4} \right] =$$

$$= A^TA - r_i \left[\frac{\nabla^2 f_1}{f_1^2} + \frac{\nabla^2 f_2}{f_2^2} - 2\left(\frac{\nabla f_1(\nabla f_1)^T}{f_1^3} + \frac{\nabla f_2(\nabla f_2)^T}{f_2^3}\right)\right] \tag{45}$$

with

$$\nabla^2 f_1 = 2 \begin{bmatrix} k_3^2 & -2\,k_3 & 2\,(k_1k_3 - k_2) \\ & 2 + k_3^2 & 2\,(k_2k_3 - k_1) \\ \text{sym} & & (k_1^2 + k_2^2 - 6\,k_3^2 + 1) \end{bmatrix} \tag{46}$$

$$\nabla^2 f_2 = 4\begin{bmatrix} 3k_1^2 - k_2^2 - k_3^2 - 1 & 2(k_3 - k_1k_2) & 2(k_2 - k_1k_3) \\ & -k_1^2 + 3k_2^2 - k_3^2 - 1 & 2(k_1 - k_2k_3) \\ \text{sym} & & -k_1^2 - k_2^2 + k_3^2 - 1 \end{bmatrix} \quad (47)$$

The Newton-Raphson method with damping, implemented with subroutine NRDAMP [18, pp 39-50] was used to solve the foregoing problem. The results obtained are shown in Table 2, where k^0 was chosen randomly so as to produce a linkage of the input-crank type. The successive values of r_i $(i=1,2,3)$ employed were 0.1, 0.01 and 0.001.

Table 1

ψ =	140°	130°	110°	100°	90°
ϕ =	80°	74°	64°	58°	50°

Table 2

$$k^1 = \begin{bmatrix} 0.205592 \\ 0.577607 \\ 0.195602 \end{bmatrix}, \quad k^2 = \begin{bmatrix} 0.206277 \\ 0.793621 \\ 0.220257 \end{bmatrix}, \quad k^3 = \begin{bmatrix} 0.195852 \\ 0.843946 \\ 0.208259 \end{bmatrix}$$

The values of Table 2 were interpolated to the curve

$$k(r) = \alpha + \beta r^{1/2} + \gamma r$$

which produced the following:

$$\alpha = \begin{bmatrix} 0.189301 \\ 0.918406 \\ 0.199494 \end{bmatrix}, \quad \beta = \begin{bmatrix} 0.224439 \\ -2.496524 \\ 0.309335 \end{bmatrix}, \quad \gamma = \begin{bmatrix} -0.546832 \\ 4.486709 \\ -1.017054 \end{bmatrix}$$

Thus the optimizing value k^* was obtained as

$$k^* = k(0) = \alpha$$

which produced the linkage

$$a_1 = 1, \quad a_2 = 1.088843 \quad , a_3 = 5.024554 \quad , a_4 = 5.012682$$

for which the least-square error is

$$e = [0.001602, 0.011487, -0.034524, -0.032521, 0.013597]^T ; \|e\| = 0.050685$$

Conclusions

Mobility conditions for RRRR planar linkages have been derived, that are equivalent to Grashof's mobility criteria. The conditions presented here differ from the usual ones in that they are established as relations on analytic functions of variables that are nonlinear combinations of the link lengths, rather than on unsmooth (because of the absolute-value function appearing there) functions of the link lengths. The incorporation of the conditions derived here as inequality constraints of optimization problems allows their solution via gradient-dependent methods, as shown with an example of constrained least-square approximate synthesis. This problem was solved using the Newton-Raphson method, for both the approximation error and the constraints are readily differentiable infinitely many times. Hence the computation of second partial derivatives, as required by the Newton-Raphson method, is quickly executed. Finally, the quadratic-convergence property of the said method, close to a solution, was made apparent by the quick convergence of each of the three nonlinear systems of equations that were solved in the example. In fact, each solution was obtained after at most three iterations.

Acknowledgements

The research work reported here was totally supported by the Graduate Division of the Faculty of Engineering (DEPFI)-UNAM and was performed at the CAD Laboratory of DEPFI-UNAM. Professor J M Hervé of the Ecole Centrale des Arts et Manufactures, Paris (France), brought the proof of Grashof's mobility criteria, given by Briccard, to the attention of the first author, for which he is gratefully acknowledged.

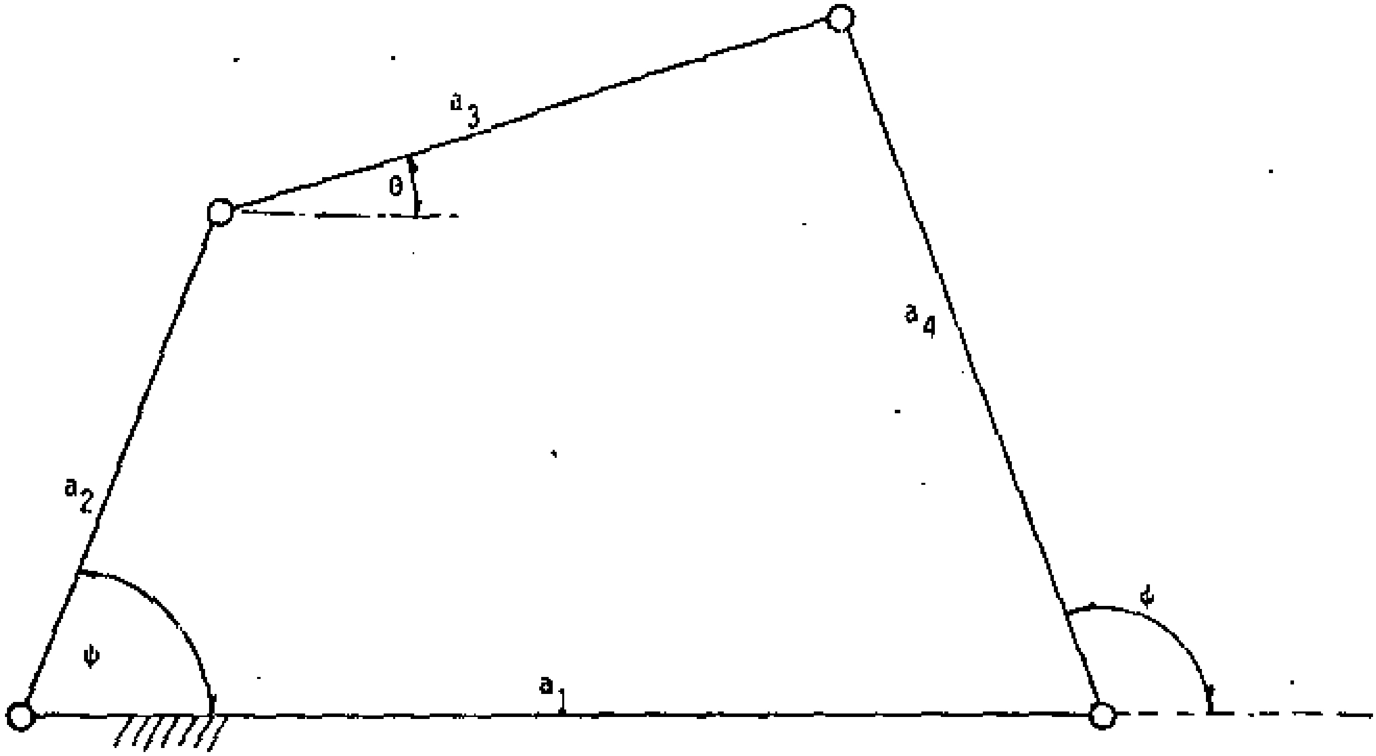


Fig 1 RRRR planarlinkage

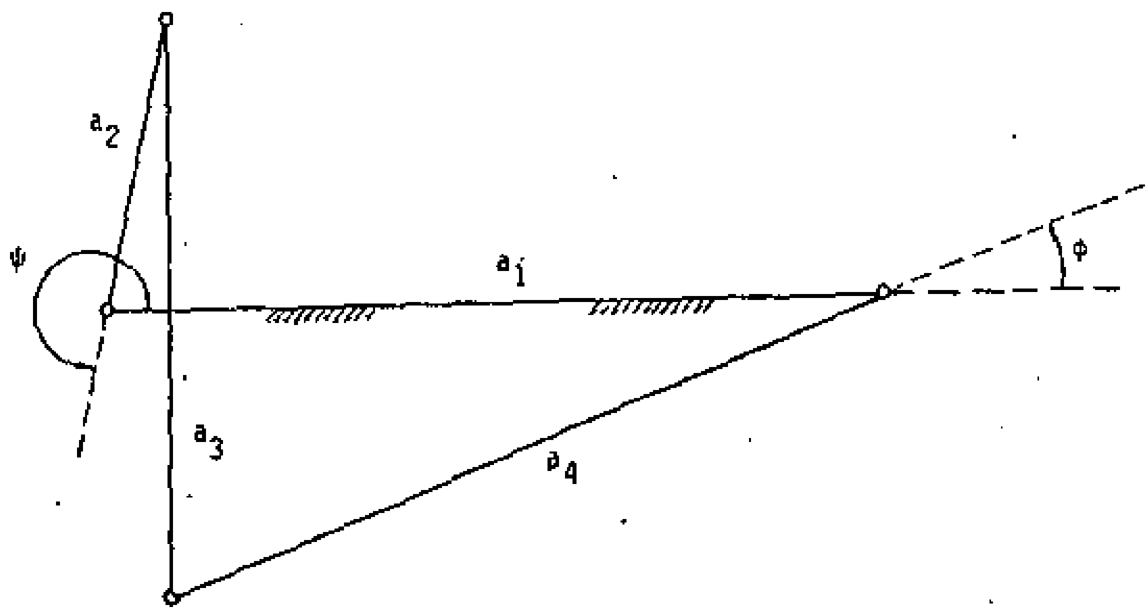


Fig 2 Synthesized RRRR planar linkage with a_2 & a_4 < 0

References

1. Briccard R, Leçons de Cinématique, Gauthier-Villars, Paris, 1927, pp 151-155

2. Nieto J, Síntesis de Mecanismos, Editorial AC, Madrid, 1978, pp 29-31

3. Paul B, "A reassessment of Grashof's criterion", Journal of Mechanical Design, Trans ASME, Vol. 101, July 1979, pp 515-518

4. Litvin F L, "Application of theorem of implicit function existence for analysis and synthesis of linkages", Mechanism and Machine Theory, Vol. 15, 1980, pp 115-125

5. Freudenstein F, "Approximate synthesis of four-bar linkages", Trans. ASME, Vol. 77, August 1955

6. Hartenberg R S and Denavit J, Kinematic Synthesis of Linkages, McGraw-Hill Book Co., New York, 1964

7. Gupta V K and Radcliffe C W, "Mobility analysis of plane and spatial mechanisms", Journal of Engineering for Industry, Trans ASME, Vol. 93, 1971, pp 125-130

8. Waldron K J, "Location of Burmester synthesis solution with fully rotatable cranks", Mechanism and Machine Theory, Vol. 13, 1978, pp 125-137

9. Waldron K J and Stevensen, Jr E N, "Elimination of branch, Grashof and order defects in path-angle and function generation synthesis", Journal of Mechanical Design, Trans ASME, Vol. 101, 1979, pp 428-437

10. Gupta K C, "A general theory for synthesizing crank-type four-bar function generators with transmission angle control", Trans ASME Journal of Applied Mechanics, Vol. 45, No 2, 1978, pp 415-421.

11. Gupta K C, "Synthesis of position, path and function generating 4-bar mechanisms with completely rotatable driving links", Mechanism and Machine Theory, Vol. 15, 1980, pp 93-101

12. Kazerounian S M K and Gupta K C, "Synthesis of position generating crank-rocker or drag-link mechanisms", Mechanism and Machine Theory,

13. Angeles J and Rojas A, "An optimisation approach to the branching problem of plane-linkage synthesis", Proc VI IFToMM Congress on Theory of Machines and Machinisms, pp 120-123 , Dec 15-20, New Delhi, India

14. Forsythe G E, Malcolm M A and Moler C B, Computer Methods for Mathematical Computations, Prentice-Hall, Inc, Englewood Cliffs, N.J. 1977, pp 20-23

15. Wilde D J, "Error linearization in the least-squares design of function generating mechanisms", J Mech Des, Trans ASME, Vol. 104, 1982, pp 881-884

16. Soderstrom T and Stewart G W, "On the numerical properties of an iterative method for computing the Moore-Penrose generalized inverse", SIAM J Num Anal, Vol. II, No 1, 1974.

17. Angeles J, "Optimal synthesis of linkages using Householder reflections", Proc V World Congress on TMM, Montreal, July 8-13, 1979, pp 111-114

18. Angeles J, Spatial Kinematic Chains. Analysis, Synthesis and Optimization, Springer-Verlag, Berlin, 1982, pp 300-304

19. Fiacco A V and McCormick, Nonlinear Programming, Sequential Unconstrained Minimization Techniques, John Wiley and Sons, Inc., New York, 1968.

20. Dahlquist G and Björck Å , Numerical Methods, Prentice-Hall, Inc., Englewood Cliffs, N J, 1974

'ion.

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO CINEMATICO DE MAQUINARIA

OPTIMAL SYNTHESIS OF LINKAGES USING HOUSEHOLDER REFLECTIONS

JORGE ANGELES

JUNIO 1984

Palacio de Minería Calle de Tacuba 5 primer piso Deleg. Cuauhtémoc 06000 México, D.F. Tel.: 521-40-20 Apdo. Postal M-2285

Proceedings of the Fifth World Congress on Theory of Machines and Mechanisms – 1979
Published by The American Society of Mechanical Engineers

OPTIMAL SYNTHESIS OF LINKAGES USING HOUSEHOLDER REFLECTIONS

J. Angeles, Professor
National University of Mexico (UNAM)
Mexico, D. F. Mexico

ABSTRACT

The unconstrained overdetermined problem of kinematic linkage synthesis is solved in an efficient way using Householder reflections. The problem formulation leads to a system of either linear or nonlinear equations in more equations than unknowns. The linear problem is solved directly by application of a finite number of successive reflections to the space of unknowns with the purpose of taking the system of equations into upper triangular form, which allows for the computation of the unknowns by back substitution. The nonlinear problem is solved via the Newton-Raphson method which computes, at each iteration, the correction to the vector of unknowns as the least-square solution to an overdetermined linear system in exactly the same way as described before for linear problems. Introduction of the said method produces accurate results in relatively short processing times, as shown in the examples presented.

ZUSAMMENFASSUNG

Das uneingeschränkte und überbestimmte Problem der kinematischen Getriebesynthese wird effizient gelöst mit Hilfe der Householder-Spiegelungen. Die Problemstellung leitet zu einem System von entweder linearen oder nichtlinearen Gleichungen mit mehr Gleichungen als Unbekannten. Das lineare Problem wird direkt gelöst mittels Anwendung einer finiten Zahl aufeinanderfolgenden Spiegelungen zum Raum der Unbekannten mit dem Ziel des Übertragens des Gleichungssystems zu einer höheren dreieckigen Form, welche die Rechnung der Unbekannten durch Rückersetzung erlaubt. Das nicht-lineare Problem wird mitteles der Newton-Raphson-Methode gelöst, die zu jeder Iteration die Besserungen der Unbekannten aus der wenigsten Quadraten-Lösung zu einem überbestimmten linearen Gleichungssystem errechnet, auf der gleichen Weise wie bei der Methode für lineare Systeme schon beschrieben wurde. Die Einführung dieser Methode führt zu deutlichen Erfolgen in relativ kurzen Prozessierzeiten, wie mittels der eingeschloßenen Beispielen gezeigt wird.

NOMENCLATURE

$\underline{A}$: upper-case underlined character, an $m\times n$ matrix.
$\underline{A}^{-1}$: the inverse of $\underline{A}$, when $\underline{A}$ is square and nonsingular
$\underline{A}^T$: the transpose of $\underline{A}$
$\underline{a}$: lower-case underlined latin character, an m-dimensional vector
$|a|$: the absolute value of a, when a is real; the modulus of a, when a is complex.
$\|\underline{a}\|$: the Euclidean norm of vector $\underline{a}$, i.e. the square root of the sum of the squares of its components
$\det\underline{A}$: the determinant of the square matrix $\underline{A}$
$\underline{f}(\underline{x})$: an m-dimensional vector function of the n-dimensional vector argument $\underline{x}$
$\underline{f}'(\underline{x})$: the Jacobian $m\times n$ matrix of $\underline{f}$ with respect to $\underline{x}$

PROBLEM FORMULATION

The equations arising in the realm of kinematic synthesis of linkages constitute either linear or nonlinear algebraic[1] systems (1,2), whose unknowns are the geometric parameters (lengths and angles) of the linkage. If these paramenters are arranged within the n-dimensional vector $\underline{x}$, the said equations are of the form

$$\underline{A}\underline{x}=\underline{b} \tag{1}$$

where $\underline{A}$ and $\underline{b}$ are a known $m\times n$ matrix and an m-dimensional known vector, respectively, when the system is linear. If it is nonlinear, then the synthesis equations are of the form

$$\underline{f}(\underline{x})=\underline{0} \tag{2}$$

$\underline{f}$ being an m-dimensional vector containing a set of m scalar functions $f_i(\underline{x})$ whose arguments are the unknown parameters of the linkage. When the number of specified conditions to be met by the linkage matches that of the unknowns, matrix $\underline{A}$ in (1) is square and vector $\underline{f}$ is of dimension n. In most technical problems, however, the number of prescribed conditions surpasses that of geometric paramenters available, the linkage synthesis problem thus leading to an overdetermined system of equations. This class of systems in general does not admit an exact solution, but it is possible to find a vector $\underline{x}_0$ that renders the quadratic error a minimum. Thus, the least-squares problem can be stated as:

"Find the value of $\underline{x}$ that minimizes the Euclidean norm[2] of either $\underline{A}\underline{x}-\underline{b}$, or that of $\underline{f}(\underline{x})$, depending on whether the system is linear or nonlinear".

The linear overdetermined system (1) admits a unique solution $\underline{x}_0$ that renders $\|\underline{A}\underline{x}-\underline{b}\|$ a minimum, provided $\underline{A}$ is of full rank, i.e. if rank$\underline{A}=n$. This value is given as (3)

$$\underline{x}_1=(\underline{A}^T\underline{A})^{-1}\underline{A}^T\underline{b} \tag{3}$$

where $(\underline{A}^T\underline{A})^{-1}\underline{A}^T$ is called a "Moore-Penrose generalized

---

[1] Algebraic as opposed to differential or integral equations.

[2] See the nomenclature for the definition of this term.

inverse of A". An extensive treatment of the linear least-squres problem is found in (4).

The nonlinear problem may admit multiple local minima; these can be found by application of the Newton-Raphson method (5), which at each iteration, computes the correction vector Δx_k as the least-square solution to the overdetermined linear system

$$f'(x_k)\Delta x_k = -f(x_k) \qquad (4)$$

This is a system like that appearing in eq.(1). Thus, its least-square solution is

$$\Delta x_k = -[f'(x_k)^T f'(x_k)]^{-1} f'(x_k)^T f(x_k) \qquad (5)$$

The new value of the unknown vector is then

$$x_{k+1} = x_k + \Delta x_k \qquad (6)$$

The procedure is stopped when the Euclidean norm of the correction vector is sufficiently small within the imposed accuracy, i.e. when

$$||\Delta x_k|| \le \epsilon \qquad (6)$$

ϵ being a "small" real positive number. The problem thus, whether linear or nonlinear, reduces to compute the minimizing value x_o given by eq. (3). An efficient way of computing this value, outlined next, does not require to invert any matrix. The computation is done by application of Householder reflections.

HOUSEHOLDER REFLECTIONS

An extensive account of this topic can be found in the specialized literature (4,6). For this reason, this theory is not treated here. A Householder relfection is a linear, improper orthogonal and symmetric transformation, i.e., if H is its mxm matrix representation, then

$$H = H^T = H^{-1}, \det H = -1 \qquad (8)$$

When n such tranformations are defined suitably, bly, their effect on matrix A appearing in eq. (1) is to take it into upper triangular form. This way, the transformed equations are equivalent to the following

$$Ux_o = c \qquad (9)$$

$$Ox_o = d \qquad (10)$$

where U is an upper triangular nxn matrix and O is the (m-n)xn zero matrix, c and d being n-and (m-n)-dimensional vectors, with $d \neq 0$. Thus, eq. (9) is determined and can readily be solved by back substitution, its solution x_o being the least-square solution to the overdetermined system. Eq. (10) is inconsistent and $||d||$ represents the Euclidean norm of the error in the approximation. Since the original system (1) is transformed into (9),(10) via a succession of orthogonal transformations, the error in the transformed coordinates, d, has the same Euclidean norm as that in the original coordinates. Hence, $||d||$ is the error associated with the original system.

APPLICATIONS TO KINEMATIC LINKAGE SYNTHESIS

Although in many practical applications the problems of linkage synthesis involve inequality constraints, still a considerably large class of synthesis problems are unconstrained. Moreover, efficient optimization techniques exist that handle inequality constraints by introducing suitable penalty functions (7), thus turning the problem an unconstrained one. For these reasons, the study of unconstrained optimization problems is of substantial technical interest. Applications to linkage synthesis problems are next illustrated with two examples.

Example 1. Synthesis of an RSSR function generator

The layout of an RSSR linkage, shown in Fig 1, indicates the different geometric parameters of this linkage: a_1, a_2, a_3, are the lengths of the output-, coupler-and input links, respectively; a_4 is the distance between the axes of the input and the output links; α_4 is the angle between the aforementioned axes, positive about ED; s_1 and s_4 are distances of points C and O along the axes of the output-and the input links, respectively. All over, the sign convention of Denavit and Hartenberg (1,pp.344-345) is observed. The input angle is ψ and the output angle is ϕ. For matching six pairs of input-output values (ψ_j, ϕ_j) with this linkage, Denavit and Hartenberg (1,pp.358-362) established the following relation

$$k_1\cos\phi_j + k_2\sin\phi_j + k_3\cos\psi_j + k_4\sin\psi_j + k_5(\sin\phi_j\cos\psi_j - \cos\alpha_4\sin\psi_j\cos\phi_j) + k_6 = \cos\phi_j\cos\psi_j + \cos\alpha_4\sin\psi_j\sin\phi_j \qquad (11)$$

where

$$k_1 = \frac{a_4 + s_4\sin\alpha_4\tan\phi_0}{a_3}, \quad k_2 = \frac{a_4\sin\alpha_4 - a_4\tan\phi_0}{a_3}$$

$$k_3 = \frac{a_4}{a_1\cos\phi_0}, \quad k_4 = \frac{s_1\sin\alpha_4}{a_1\cos\phi_0}; \quad k_5 = \tan\phi_0 \qquad (12)$$

$$k_6 = \frac{a_1^2 - a_2^2 + a_3^2 + a_4^2 + s_1^2 + s_4^2 + 2s_1s_4\cos\alpha_4}{2a_1a_3\cos\phi_0}$$

α_4 being assigned.

In the latter definitions, ϕ_0 measures the location of the zero of the output dial from the dotted line passing through C, parallel to line ED, as shown in Fig 1.

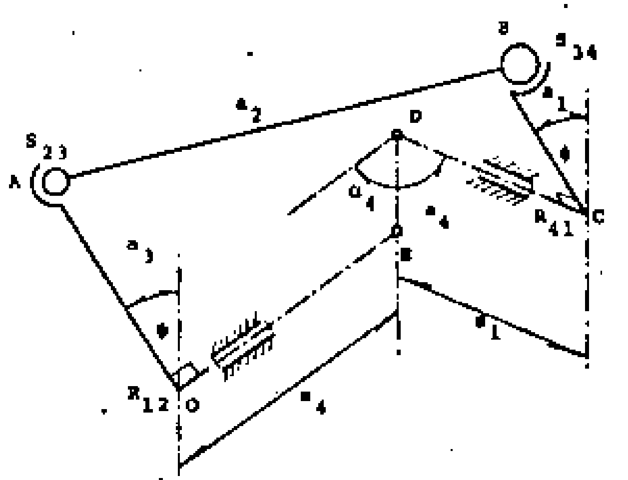


Fig 1 An RSSR linkage

For six precision-point synthesis, eqs. (11) yield a system of six linear equations in six unknowns which, when nonsingular, produces unique values $k_1, k_2, \ldots, k_6$. With these values known, the linkage parameters are computed from eqs. (12) for a given value of a_1. If more than six precision points are required, however, the system becomes overdetermined, in which case an efficient method to obtain its least-square solution is via Householder relfections.

In (8), Suh and Mecklenburg solve the over-determined uncostrained problem of this linkage with 19 prescribed input-output values. For comparison purposes, the solution developed in this example makes use of the same prescribed values. These are shown in Table 1

The method employed in (8) is that of Powell's (9), which does not require the computation of derivatives and leads quickly to convergence for quadratic functions of the independent variables. At this point, two remarks are in order: First, the derivatives of the synthesis equations are easily computed from either Devanit and Hartenberg's formulation, eqs. (11), or from Suh and Redcliffe's formulation (7), the first one being advantageous because of producing a linear system of equations. Second: The objective function of Suh and Mecklenburg's (8) is quadratic in the synthesis which, in turn, are quadratic in the independent variables; thus, their objective function is quartic in the independent variables, for which reason the quick convergence properties of Powell's method are not fully utilized. Furthermore, squaring the synthesis functions may introduce spurious local minima, as is apparent form the fact that three optimal solutions are reported in (8).

TABLE 1. Specified input-output pairs for the synthesis of the RSSR function generating linkage.

	ψ(degreees)	ϕ(degrees)
1	0.0	0.0
2	5.0	2.4
3	10.0	5.1
4	15.0	8.2
5	20.0	11.5
6	25.0	15.2
7	30.0	19.1
8	35.0	23.3
9	40.0	27.7
10	45.0	32.3
11	50.0	37.2
12	55.0	42.3
13	60.0	47.5
14	65.0	53.0
15	70.0	58.7
16	75.0	64.6
17	80.0	70.9
18	85.0	78.0
19	90.0	90.0

One advantage of using Householder reflections is that no explicit squaring is required, and the unique solution is obtained directly by the application of n(=6) reflections. Another advantage is that, since less computations are required, as compared to Powell's method, the round-off error is lowered. The approximation error obtained using each method is shown in Table 2.

The root mean square errors were essentially the same: that obtained by Powell's method was 0.00185269°, whereas the one obtained by Householder reflections, 0.00182254°. However, the differences in the resulting linkage parameters were more notorious. These are

Solution by Powell's method	Solution by Householder's method
a_1= 1.253803	a_1= 0.911269
a_2= 2.759566	a_2= 2.620568
a_3= 0.435003	a_3= 0.803577
s_1= 2.262110	s_1=-1.186240
s_4=-1.375270	s_4=-2.417556

In this problem, a_4 was set equal to 1, whereas α_4, equal to 90°.

TABLE 2. Approximation error in overdetermined RSSR linkage synthesis

	APROXIMATION ERROR USING POWELL'S METHOD (degrees)	APROXIMATION ERROR USING HOUSEHOLDER'S METHOD (degrees)
1	0.00000000	0.01420369
2	-.00120000	0.00100785
3	-.03060000	-.03515343
4	0.02330000	0.01590827
5	-.02680000	-.03451603
6	0.03260000	0.02596072
7	0.01600000	0.01079286
8	0.03830000	0.03440434
9	0.01520000	0.01196148
10	-.03790000	-.04106664
11	-.00640000	-.00968497
12	0.02270000	0.01930349
13	-.04220000	-.04497542
14	-.00020000	-.00155544
15	0.03350000	0.03434200
16	0.01600000	0.01324614
17	0.00020000	0.00139267
18	-.01250000	-.01970407
19	0.02980000	0.00412793

Example 2. Synthesis of the RR plane dyad for rigid-body guidance

A rigid body (shaded rectangle) appears in Fig 2, in "reference" configuration C_0 and in a different configuration C_j. Each configuration is defined by the position of a point, R, and angle, θ. In that figure, O represents the origin of the complex plane, and the arrows represent complex numbers associated with the location of the labelled points. The purpose of this class of synthesis problem is to locate point A whose reference and successive positions, A_0, A_j (j=1, ...,n) lie on a circumference centered at B, for which reason, A_0 and B are called, respectively, "circucular" and "central" points, within the Burmester Theory (10). Thus, AB_0 can constitute a rigid link to guide the rigid body. This is an RR plane dyad.

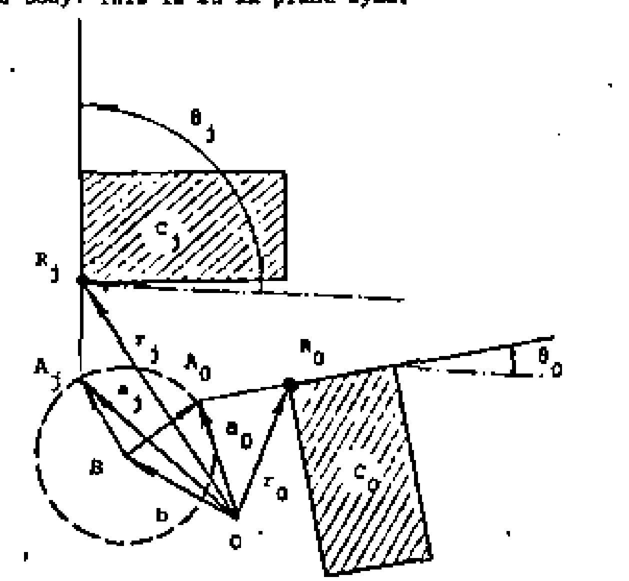


Fig 2 An RR plane dyad to guide a rigid body through n successive configurations

. The constancy of the length of line BA throughout its n configurations leads to

$$|e^{i\theta'_j}(a_0-r_0)+r_j-b|^2=|a_0-b|^2, j=1,\ldots,n \qquad (13)$$

where $\theta'_j \equiv \theta_j-\theta_0$. Eqs. (13) constitute the synthesis equations for this problem, a_0 and b being the unknowns. It is well known (2, p.146) that this problem allows to conduct a rigid body through five specified configurations. Some technical problems, however, may require to guide the body through more than five configurations, as shown in Table 3. Different syntheses were obtained for these, starting from the first 6 configurations, then adding the next ones, one at each time, until the 16 configurations were included

TABLE 3. Successive configurations of a rigid body

j	x_j (cm)	y_j (cm)	θ_j (degrees)
0	7.880	-0.260	313.720
1	8.490	-7.290	332.330
2	7.680	2.820	349.930
3	6.300	4.210	353.180
4	4.580	4.950	359.870
5	2.740	5.010	355.840
6	1.010	4.410	356.300
7	0.259	3.880	3.900
8	-0.400	-3.090	3.670
9	0.250	-3.760	3.690
10	1.000	-4.290	4.150
11	2.730	-4.890	5.120
12	4.560	-4.830	6.810
13	6.280	-4.090	10.000
14	7.660	-2.700	13.000
15	8.440	-0.610	18.000
16	7.790	-2.690	46.270

The procedure converged for all given inital guesses, produced by means of a random number generating subprogram, in less than 50 iterations (usually around 20). Contrary to the determined case (5 prescribed configurations), for which two different meaningful solutions exist, for the cases tried here the procedure converged always to the same single solution, except for 6 and 17 configurations, which produced two different solutions. The error in the approximation was normalized, to yield a dimensionless number, in the following way: Let

$$f_j=|e^{i\theta'_j}(a_0-r_0)+r_j-b|^2-|a_0-b|^2, j=1,\ldots,m \qquad (14)$$

If the synthesis were exact, then all f_j would be negligibly small. In approximate syntheses, however, these functions attain finite values. The kinematic meaning of these values is that they represent the difference between the length of the RR dyad in its initial configuration, and that in its jth configuration, i.e. $A_jB_j-A_0B_0$ if the synthesized linkage were to satisfy the prescribed conditions exactly. The dimensionless error in the approximation, e_j, associated with the jth configuration, is then

$$e_j=|f_j|/|a_0-b|^2, j=1,\ldots,m \qquad (15)$$

where a_0 and b are those obtained from the least-square solution to the nonlinear system of equations. Notice that the errors thus defined are quadratic. To obtain a representative value of the overall error, the average of the square roots of the m errors defined in (14) should be taken, i.e.

$$e=\sum_{1}^{m}\sqrt{|f_j|}\ /\ m|a_0-b|$$

Some of the results obtained are shown next.

TABLE 4. Overdetermined synthesis of the RR dyad for rigid-body guidance.

For 6 configurations,

First solution:	Second solution:
a_0=-0.961467-i2.826960	a_0=7.690390+i2.700030
b^0=-1.643590-i7.997190	b^0=0.748466-i0.609952
Error = 17.02%	Error = 13.49%

For 17 configurations,

First solution:	Second solutoon:
a_0=5.123750+i2.254620	a_0=1.443950-i6.704520
b^0=0.549476-i7.03377	b^0=6.370810-i9.315060
Error = 38.74%	Error = 60.77%

CONCLUSIONS

Householder reflections appear to be far more efficient in solving linear problems arising within the field of unconstrained optimal synthesis of linkages. As to nonlinear problems, the extension is straightforward. Regarding constrained problems, these could be handled using this method by introducing suitable slack variables and penalty functions. As to processor times, the first example consummed 11.8 sec, whereas the time reported (8) using Powell's method is 2.2 min, the method introduced here thus appearing to be more economical. With regard to the synthesis for rigid-body guidance, it is necessary to investigate whether for overdetermined problems, in general two different solutions can be expected, thus enabling the designer to synthesize RRRR plane linkages for overdetermined rigid-body guidance problems.

ACKNOWLEDGEMENTS

This research project was sponsored by the School of Enginnering at the National Autonomous University of Mexico. The computer programming was performed by Mr. Mario Siller. Householder's method was implemented via subroutines HECOMP and HOLVE, due to Moler (11)

REFERENCES

1. Denavit J. and Hartenberg R.S. Kinematic Synthesis of Linkages, Mc Graw-Hill, N. York, 1964
2. Suh C.H., and Radcliffe C.W., Kinematics and Mechanisms Design, Wiley, N. York, 1978
3. Ben-Israel A. and Greville T.N.E., Generalized Inverses: Theory and Applications, Wiley, N.York, 1974, pp.103-104
4. Stewart G.W., Introduction to Matrix Computations, Academic Press, N. York, 1973,pp.208-249
5. Björk Å. and Dahlquist G., Numerical Methods, Prentice-Hall, Englewood Cliffs, 1974,pp.4433-444.
6. Businger P. and Golub G.H., "Linear Least Squares Solutions by Householder Transformations", in Wilkinson J.H. and Reinsch C., eds., Handbook for Automatic Computation, Vol II, Springer-Verlag, N. York, 1971, pp. 111-118
7. Fox R.L and Gupta K.C., "Optimization Technology as Applied to Mechanism Design", Journal of Engineering for Industry,Trans. ASME, Series B.Vol. 95, May 1973, pp. 657-663
8. Suh C.H. and Mecklenburg A.W.,"Optimal Design of Mechanisms with the Use of Matrices and Least Squares", Mechanism and Machine Theory, Vol.8,pp. 479-495
9. Powell M.J.D., "An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives", Computer Journal, vol.7. No.4,1964,pp.303-307
10. Burmester R., Lehrbuch der Kinematik,Vol.1, „Die ebene Bewebung", Verlag von Arthur Felix, 1886
11. Moler C.B., Matrix Eigenvalue and Least-Square Computations, Computer Science Department, Stanford University, March 1973

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO CINEMATICO DE MAQUINARIA

UNA GENERALIZACION DEL TEOREMA DE ARONHOLD - KENNEDY

DR. JORGE ANGELES

JUNIO, 1984.

Palacio de Minería Calle de Tacuba 5 primer piso Deleg. Cuauhtémoc 06000 México, D.F. Tel.: 521-40-20 Apdo. Postal M-2285

Una generalización del teorema de Aronhold-Kennedy

Jorge Angeles Alvarez*

INTRODUCCION

En el presente trabajo se hace una generalización del teorema de Aronhold-Kennedy, comúnmente conocido en los textos elementales de mecanismos como *teorema de los tres centros.*

Este teorema establece que cuando tres cuerpos están en movimiento, tienen tres centros instantáneos de movimiento relativo, los cuales se encuentran alineados. Carece de precisión, pues en el caso más general de movimiento de un cuerpo rígido no es posible hablar de centros instantáneos de movimiento relativo, quedando reducido este término al caso particular de movimiento plano, que no es el único que se presenta en problemas de ingeniería; baste citar el caso del movimiento de una junta universal o el de un tren de engranes cónicos, hipoidales o corona sinfín.

Surge entonces la necesidad de hacer una generalización de este teorema que comprenda al movimiento tridimensional, por lo que en este trabajo se establece la condición necesaria y suficiente para que exista movimiento de rotación pura entre los elementos de un mecanismo, y al llevar el teorema a tres dimensiones, se habla, en términos más generales, de ejes instantáneos de rotación.

Como en general, al diseñar un mecanismo, se requiere que este sea lo más eficiente posible (hay casos en que no importa tanto la eficiencia como la ventaja mecánica), es necesario evitar el deslizamiento de las superficies de contacto, el que es nulo cuando hay movimiento de rotación pura.

Existe otra generalización del teorema, que garantiza la existencia de ejes de deslizamiento mínimo y que es útil en el diseño de mecanismos en los cuales es imposible evitar el deslizamiento, como sucede en los acoplamientos hipoidales y en los corona-sinfín. Esta generalización se puede consultar en libros avanzados de mecanismos (ref 1).

En su forma más general, el teorema de Aronhold-Kennedy establece que existen ejes de velocidad relativa mínima (llamados ejes de tornillo instantáneo) y que dichos ejes, para tres cuerpos rígidos en movimiento, son tres y tienen la propiedad de ser normales a un eje común.

La originalidad del presente trabajo estriba en que se llega a la forma general del teorema utilizando vectores cartesianos, en lugar de métodos matriciales.

* Profesor de la Facultad de Ingeniería, UNAM

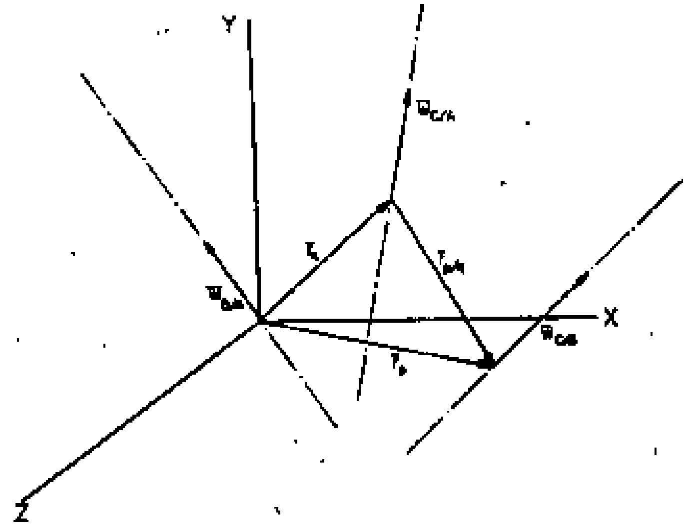


Fig 1

TEOREMA

Un cuerpo C, que gira con respecto a un cuerpo B, que a su vez está animado de un movimiento de rotación con respecto a un tercer cuerpo A, tiene un eje instantáneo de rotación con respecto a A si, y solo si, los ejes instantáneos de C con respecto a B, y de B con respecto a A son concurrentes; entonces, el eje instantáneo de C con respecto a A, también es concurrente con los dos anteriores.

Para su demostración, se dividirá el teorema en dos partes: en la primera se supondrá que existen los tres ejes, y se demostrará que son concurrentes; en la segunda, se considerará que los dos primeros ejes son concurrentes, y, como consecuencia de ello, se demostrará que existe el tercer eje, y que es concurrente con los dos anteriores.

Obsérvese que como lo que interesa es el movimiento relativo de los tres cuerpos, puede considerarse que uno de ellos tiene una velocidad arbitraria. Si por ejemplo, la velocidad angular del cuerpo A con respecto a un sistema de ejes inercial o newtoniano es nula, la velocidad angular de cualquier cuerpo referida a A es absoluta.

Según lo anterior, los ejes instantáneos de B y de C con respecto a A son conjuntos de puntos de velocidad absoluta nula.

Sean: x-y-z, un marco newtoniano al que está fijo A; $\overline{w}_{B/A}$ $\overline{w}_{C/B}$ y $\overline{w}_{C/A}$ las velocidades angulares de B con respecto a A, C con respecto a B, y C con respecto a A, respectivamente. En la fig 1, se muestran los tres ejes instantáneos.

El punto Q está alojado sobre el eje instantáneo de C con respecto a A; por lo tanto, puede considerarse como punto de C o de A indistintamente, y tiene, en consecuencia, velocidad absoluta nula. El punto P está alojado sobre el eje instantáneo de C con respecto a B, por lo que puede considerarse alojado en B o en C indistintamente.

Los vectores de posición de P y de Q son, respectivamente, $\overline{r}_p$ y $\overline{r}_q$.

Llamando $\overline{r}_{12}$ al vector de posición de la intersección de los ejes B con respecto a A y C con respecto a A, $\overline{r}_{13}$ al vector de posición de la intersección de los ejes B con respecto a A y C con respecto a B, y $\overline{r}_{23}$ a la intersección de los ejes C con respecto a A y C con respecto a B, la primera parte del teorema quedará demostrada al cumplirse las siguientes igualdades:

$$\overline{w}_{C/B}\ \overline{w}_{B/A}\ \overline{r}_p = 0$$

$$\overline{w}_{B/A}\ \overline{w}_{C/A}\ \overline{r}_q = 0$$

$$\overline{w}_{C/B}\ \overline{w}_{C/A}\ \overline{r}_{qp} = 0$$

$$\overline{r}_{12} = \overline{r}_{13} = \overline{r}_{23}$$

En efecto:

$$\overline{v}_q = \overline{v}_p + \overline{v}_{q/p}$$

Pero:

$$\overline{v}_p = \overline{w}_{B/A}\ \overline{r}_p$$

Y:

$$\overline{v}_{q/p} = \overline{w}_{C/B}\ (\overline{r}_q - \overline{r}_p) \tag{1}$$

Por lo tanto

$$\overline{v}_q = \overline{w}_{B/A}\ \overline{r}_p + \overline{w}_{C/B}\ \overline{r}_q - \overline{w}_{C/B}\ \overline{r}_p$$

Pero como Q es un punto de A, su velocidad absoluta es nula, por lo que

$$\overline{w}_{B/A}\ \overline{r}_p + \overline{w}_{C/B}\ \overline{r}_q - \overline{w}_{C/B}\ \overline{r}_p = 0$$

y

$$\overline{w}_{C/B}\ \overline{r}_p = \overline{w}_{B/A}\ \overline{r}_p + \overline{w}_{C/B}\ \overline{r}_q$$

Multiplicando ambos miembros por $\overline{w}_{B/A}$, se tiene

$$\overline{w}_{B/A}\ \overline{w}_{C/B}\ \overline{r}_p = \overline{w}_{B/A}\ \overline{w}_{B/A}\ \overline{r}_p + \overline{w}_{B/A}\ \overline{w}_{C/B}\ \overline{r}_q \tag{2}$$

El segundo término del miembro a la derecha de la última expresión es nulo, por lo que

$$\overline{w}_{B/A}\ \overline{w}_{C/B}\ \overline{r}_p = \overline{w}_{B/A}\ \overline{w}_{C/A}\ \overline{r}_q \tag{3}$$

Pero, de la ec 1: $\bar{v}_{q/p} = \bar{w}_{C/B}\,\bar{r}_{q/p}$

por lo que $\bar{v}_q = \bar{w}_{B/A}\,\bar{r}_p + \bar{w}_{C/B}\,\bar{r}_{q/p} = 0$

Es decir $\bar{w}_{B/A}\,\bar{r}_p = -\,\bar{w}_{C/B}\,\bar{r}_{q/p}$

Multiplicando la expresión anterior por $\bar{w}_{C/A}$, y tomando en cuenta que $\bar{w}_{C/A} = \bar{w}_{C/B} + \bar{w}_{B/A}$:

$$(\bar{w}_{C/B} + \bar{w}_{B/A})\,\bar{w}_{B/A}\,\bar{r}_p = -\,\bar{w}_{C/A}\,\bar{w}_{C/B}\,\bar{r}_{q/p}$$

Desarrollando el primer término, y eliminando el término que se anula

$$\bar{w}_{C/B}\,\bar{w}_{B/A}\,\bar{r}_p = -\,\bar{w}_{C/A}\,\bar{w}_{C/B}\,\bar{r}_{q/p}$$

Efectuando una rotación no cíclica con los factores del segundo miembro de esta expresión para eliminar el signo negativo, se tiene

$$\bar{w}_{C/B}\,\bar{w}_{B/A}\,\bar{r}_p = \bar{w}_{C/B}\,\bar{w}_{C/A}\,\bar{r}_{q/p} \tag{4}$$

Pero

$$\bar{w}_{C/B}\,\bar{w}_{C/A}\,\bar{r}_{q/p} = \bar{w}_{C/B}\,\bar{w}_{C/A}\,\bar{r}_q - \bar{w}_{C/B}\,\bar{w}_{C/A}\,\bar{r}_p$$

Sustituyendo $\bar{w}_{C/B}$ por su valor en función de las otras velocidades angulares, en el primer término del miembro a la derecha de la última expresión, se tiene

$$\bar{w}_{C/B}\,\bar{w}_{C/A}\,\bar{r}_{q/p} = (\bar{w}_{C/A} - \bar{w}_{B/A})\,\bar{w}_{C/A}\,\bar{r}_q - \bar{w}_{C/B}\,\bar{w}_{B/A}\,\bar{r}_p$$

Desarrollando el paréntesis, y eliminando el término que se anula, se tiene

$$\bar{w}_{C/B}\,\bar{w}_{C/A}\,\bar{r}_{q/p} = -\,\bar{w}_{B/A}\,\bar{w}_{C/A}\,\bar{r}_q - \bar{w}_{C/B}\,\bar{w}_{B/A}\,\bar{r}_p \tag{5}$$

Efectuando una rotación no cíclica con los factores del segundo término del miembro a la derecha de la ec 5, a fin de cambiar su signo,

$$\bar{w}_{C/B}\,\bar{w}_{C/A}\,\bar{r}_{q/p} = \bar{w}_{B/A}\,\bar{w}_{C/B}\,\bar{r}_p - \bar{w}_{B/A}\,\bar{w}_{C/A}\,\bar{r}_q$$

Según la ec 3 el miembro de la derecha de esta última expresión es cero, por lo que

$$\bar{w}_{C/B}\,\bar{w}_{C/A}\,\bar{r}_{q/p} = 0 \tag{6}$$

De la ec 4 resulta

$$\bar{w}_{C/B}\,\bar{w}_{B/A}\,\bar{r}_p = 0 \tag{7}$$

De la ec 2,

$$\bar{w}_{B/A}\,\bar{w}_{C/B}\,\bar{r}_q = 0$$

Sustituyendo en esta última expresión $\bar{w}_{C/B}$ por su valor $\bar{w}_{C/A} - \bar{w}_{B/A}$, se tiene

$$\bar{w}_{B/A}\,\bar{w}_{C/A}\,\bar{r}_q - \bar{w}_{B/A}\,\bar{w}_{B/A}\,\bar{r}_q = 0$$

Pero el segundo término de esta expresión se anula, por lo que

$$\bar{w}_{B/A}\,\bar{w}_{C/A}\,\bar{r}_q = 0 \tag{8}$$

De las ecs 6, 7 y 8, se deduce que los ejes se intersecan dos a dos. Falta ahora demostrar que se intersecan en un punto común. Para esto, basta verificar que el vector de posición es el mismo para las tres intersecciones; es decir, $\bar{r}_{12} = \bar{r}_{13} = \bar{r}_{23}$.

En la notación anterior, se sigue la misma nomenclatura que se expuso al enunciar el teorema.

Previamente se determinará la expresión que define a cada uno de los vectores de posición anteriores, para lo cual debe recurrirse a la fig 2.

Aparecen dos ejes que se intersecan en el punto I. Estos ejes están determinados por los puntos L y M, dados a su vez por sus vectores de posición $\bar{r}_L$ y $\bar{r}_M$ respectivamente, y por los vectores unitarios que dan su dirección, $\bar{l}$ y $\bar{m}$. El problema es determinar el vector $\bar{r}_I$.

$$\bar{r}_I = \bar{r}_L - \bar{r}_{LI}$$

$\bar{r}_L$ es dato; $\bar{r}_{LI} = -\,\mathrm{LI}\,\bar{I}$

El segmento LI se determina por el teorema de los senos, a partir de la fig

$$\frac{\mathrm{LI}}{\mathrm{sen\ B}} = \frac{\mathrm{LM}}{\mathrm{sen\ A}}\ ;\ \text{de donde}\ \mathrm{LI} = \frac{\mathrm{sen\ B}}{\mathrm{sen\ A}}\ \mathrm{LM}$$

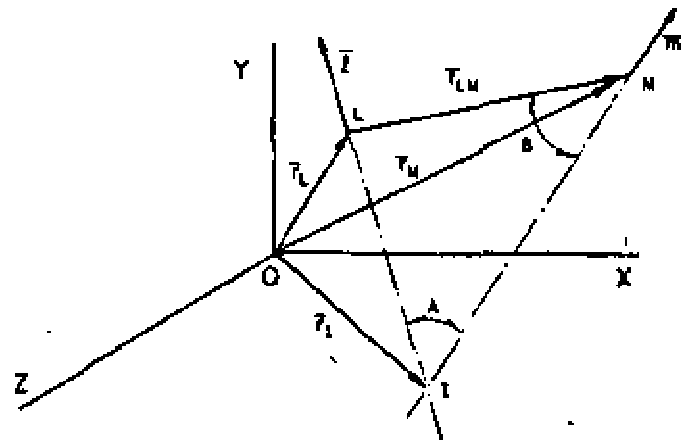


Fig 2

La longitud del segmento LM es dato, pues está dada por los vectores de posición de L y de M, que también son datos; los senos de A y de B se determinan de la siguiente manera:

$$\mathrm{sen\ B} = \frac{|\bar{r}_{LM}\ \bar{m}|}{|\bar{r}_{LM}|}\ ;\ \mathrm{sen\ A} = |\bar{I}\ \bar{m}|$$

$$\text{Por lo tanto } \mathrm{LI} = |\bar{r}_{LM}|\ \frac{|\bar{r}_{LM}\ \ \bar{m}|}{|\bar{r}_{LM}|\ \ |\bar{I}\ \ \bar{m}|} = \frac{|\bar{r}_{LM}\ \ \bar{m}|}{|\bar{I}\ \ \bar{m}|}$$

$$\bar{r}_{LI} = -\frac{|\bar{r}_{LM}\ \ \bar{m}|}{|\bar{I}\ \ \bar{m}|}\ \bar{I}$$

$$\text{De donde } \bar{r}_I = \bar{r}_L + \frac{|\bar{r}_{LM}\ \ \bar{m}|}{|\bar{I}\ \ \bar{m}|}\ \bar{I}$$

Esta expresión servirá para determinar el vector de posición de cada intersección.

En la fig 3 aparece nuevamente la disposición de los tres ejes instantáneos a que se hace mención, mostrándose las intersecciones de cada par de ejes y los vectores de posición de cada una de estas.

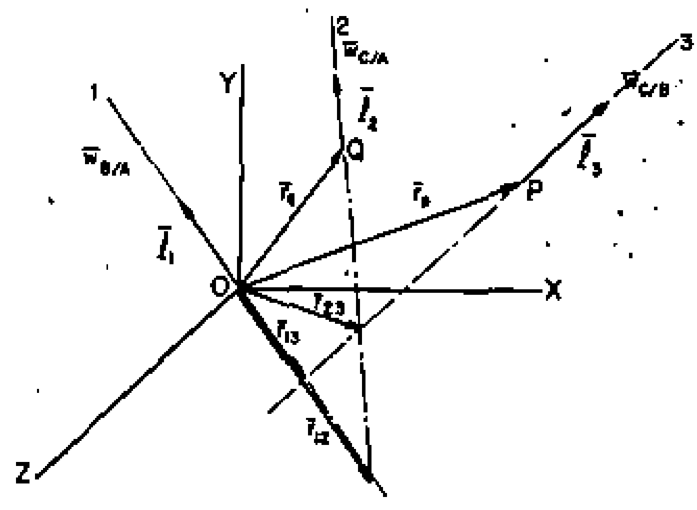


Fig 3

Los vectores unitarios $\bar{I}_1$, $\bar{I}_2$ y $\bar{I}_3$ están dados por

$$\frac{\bar{w}_{B/A}}{|\bar{w}_{B/A}|}\ ,\ \frac{\bar{w}_{C/A}}{|\bar{w}_{C/A}|}\ \ y\ \ \frac{\bar{w}_{C/B}}{|\bar{w}_{C/B}|}\quad \text{respectivamente.}$$

De la expresión que da r_I, se tiene:

$$\bar{r}_{12} = -\frac{|\bar{r}_q\ \bar{I}_2|}{|\bar{I}_1\ \bar{I}_2|}\ \bar{I}_1 = -\frac{|\bar{r}_q\ \ \bar{w}_{C/A}|}{|\bar{w}_{C/A}|\ \dfrac{|\bar{w}_{B/A}\ \bar{w}_{C/A}|}{|\bar{w}_{B/A}||\bar{w}_{C/A}|}}\ \frac{\bar{w}_{B/A}}{|\bar{w}_{B/A}|}$$

Simplificando

$$\bar{r}_{12} = -\frac{\left|\bar{r}_q \quad \bar{W}_{C/A}\right|}{\left|\bar{W}_{B/A}\ \bar{W}_{C/A}\right|}\ \bar{W}_{B/A} \tag{9}$$

$$\bar{r}_{13} = -\frac{\left|\bar{r}_p\ \bar{T}_3\right|}{\left|\bar{T}_1\ \ \bar{T}_3\right|}\ \bar{T}_1 = -\frac{\left|\bar{r}_p\ \ \bar{W}_{C/B}\right|}{\left|\bar{W}_{C/B}\right|\dfrac{\left|\bar{W}_{B/A}\ \bar{W}_{C/B}\right|}{\left|\bar{W}_{B/A}\right|\left|\bar{W}_{C/B}\right|}}\ \frac{\bar{W}_{B/A}}{\left|\bar{W}_{B/A}\right|}$$

Simplificando

$$\bar{r}_{13} = -\frac{\left|\bar{r}_p \quad \bar{W}_{C/B}\right|}{\left|\bar{W}_{B/A}\ \bar{W}_{C/B}\right|}\ \bar{W}_{B/A} \tag{10}$$

$$\bar{r}_{23} = \bar{r}_q - \frac{\left|\bar{r}_{qp}\ \bar{T}_3\right|}{\left|\bar{T}_2\ \ \bar{T}_3\right|}\ \bar{T}_2 = \bar{r}_q - \frac{\left|\bar{r}_{qp}\ \bar{W}_{C/B}\right|}{\left|\bar{W}_{C/B}\right|\dfrac{\left|\bar{W}_{C/A}\ \bar{W}_{C/B}\right|}{\left|\bar{W}_{C/A}\right|\left|\bar{W}_{C/B}\right|}}\ \frac{\bar{W}_{C/A}}{\left|\bar{W}_{C/A}\right|}$$

de donde

$$\bar{r}_{23} = \bar{r}_q - \frac{\left|\bar{r}_{qp}\ \ \bar{W}_{C/B}\right|}{\left|\bar{W}_{C/A}\ \bar{W}_{C/B}\right|}\ \ \bar{W}_{C/A} \tag{11}$$

Primero se demostrará que $\bar{r}_{12} = \bar{r}_{13}$

En efecto $\bar{r}_q\ \bar{W}_{C/A} = \bar{W}_{C/A}\ \bar{r}_q$, y

$$\bar{r}_p\ \bar{W}_{C/B} = \bar{W}_{C/B}\ \bar{r}_p$$

En estas expresiones, por comodidad se ha invertido el sentido de los factores de los productos vectoriales, inversión que resulta de un cambio de signo al producto; pero como se manejan módulos, el cambio de signo no afecta a éstos y la manipulación es justificada.

Entonces

$$\bar{W}_{C/B}\ \bar{r}_p = (\bar{W}_{C/A} - \bar{W}_{B/A})\ \ (\bar{r}_q + \bar{r}_{qp}) =$$

$$= \bar{W}_{C/A}\ \bar{r}_q + \bar{W}_{C/A}\ \bar{r}_{qp} - \bar{W}_{B/A}\ \bar{r}_q - \bar{W}_{B/A}\ \bar{r}_{qp} =$$

$$= \bar{W}_{C/A}\ \bar{r}_q + \bar{v}_p - \bar{v}_q - \bar{W}_{B/A}\ \bar{r}_{qp}$$

Pero $\bar{v}_q = 0$, como ya se vio en un principio. De donde

$$\bar{W}_{C/B}\ \bar{r}_p = \bar{W}_{C/A}\ \bar{r}_q - \bar{v}_p - \bar{W}_{B/A}\ (\bar{r}_p - \bar{r}_q)$$

$$= \bar{W}_{C/A}\ \bar{r}_q - \bar{v}_p - \bar{v}_q$$

Obsérvese que $\bar{W}_{B/A}\ \bar{r}_p = \bar{v}_p$, y que $\bar{W}_{B/A}\ \bar{r}_q = \bar{v}_q = 0$

Simplificando, se obtiene

$\bar{W}_{C/B}\ \bar{r}_p = \bar{W}_{C/A}\ \bar{r}_q$, y al tomar módulos

$$\bar{W}_{C/B}\ \bar{r}_p = \bar{W}_{C/A}\ \bar{r}_q \tag{12}$$

Por otra parte $\bar{W}_{B/A}\ \bar{W}_{C/B} = \bar{W}_{B/A}\ (\bar{W}_{C/A} - \bar{W}_{B/A})$

$$\text{de donde } \bar{W}_{B/A}\ \bar{W}_{C/B} = \bar{W}_{B/A}\ \bar{W}_{C/A} \tag{13}$$

Dividiendo (12) entre (13), resulta

$\dfrac{\bar{r}_q \quad \bar{W}_{C/A}}{\bar{W}_{B/A}\ \bar{W}_{C/A}} = \dfrac{\bar{r}_p \quad \bar{W}_{C/B}}{\bar{W}_{B/A}\ \bar{W}_{C/B}}$ con lo que se demuestra que

$$\bar{r}_{12} = \bar{r}_{13}$$

Llamando I a esta intersección común, se tiene que I está en los ejes 1 y 2, al mismo tiempo que en los ejes 1 y 3, por lo que, I está en los ejes 2 y 3; por lo tanto

$$\bar{r}_{23} = \bar{r}_{12} = \bar{r}_{13}$$

De lo anterior se concluye que si existen los tres ejes, estos concurren en un punto común, con lo cual se ha demostrado la primera parte del teorema.

Falta demostrar la segunda parte, en la que se supone que dos de los ejes se intersecan en un punto, y que como consecuencia de ello existe el tercer eje, demostrándose además, que este es concurrente con los dos anteriores.

Supóngase que los ejes de B con respecto a A, y de C con respecto a B, se intersecan en un punto que, por comodidad, se considerará como el origen de coordenadas. Sea x-y-z un marco newtoniano, con origen en la intersección de los dos ejes. Debe demostrarse que existe un conjunto de puntos con velocidad absoluta nula que están en el cuerpo C y que en consecuencia, constituyen el eje instantáneo de C con respecto a A.

Sea P un punto cualquiera de C. Su velocidad absoluta con respecto a la velocidad de un punto de B, como Q por ejemplo, será

$$\bar{v}_p = \bar{v}_q + \bar{v}_{p/q}$$

$$\bar{v}_q = \bar{W}_{B/A}\ \bar{r}_q, \text{ y}$$

$$\bar{v}_{p/q} = \bar{W}_{C/B}\ \bar{r}_{qp}$$

Entonces $\bar{v}_p = \bar{W}_{B/A}\ \bar{r}_q - \bar{W}_{C/B}\ \bar{r}_{qp}$

$$\bar{v}_p = \bar{W}_{B/A}\ \bar{r}_q + \bar{W}_{C/B}\ \bar{r}_{qp} = \bar{W}_{B/A}\ \bar{r}_q + \bar{W}_{C/B}\ (\bar{r}_p - \bar{r}_q)$$

$$\bar{v}_p = \bar{W}_{B/A}\ \bar{r}_q + \bar{W}_{C/B}\ \bar{r}_p - \bar{W}_{C/B}\ \bar{r}_q$$

Pero de la fig 4, $\bar{r}_q = m\ \bar{W}_{C/B}$, en que m es un escalar cualquiera que hace que el módulo de $\bar{r}_q$ sea m veces el de $\bar{W}_{C/B}$.

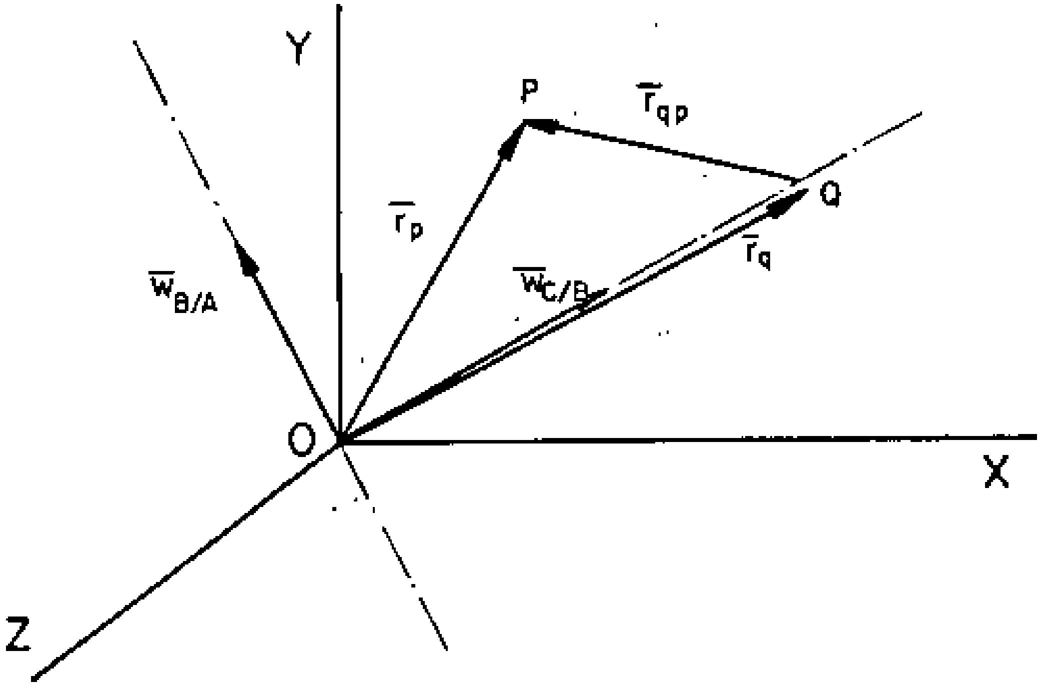


Fig 4

Entonces $\overline{w}_{C/B}\ \overline{r}_q = \overline{w}_{C/B}\ m\ \overline{w}_{C/B}$, que es evidentemente cero. Por tanto

$$\overline{v}_p = \overline{w}_{B/A}\ \overline{r}_q + \overline{w}_{C/B}\ \overline{r}_p$$

Para que $\overline{v}_p$ sea cero como condición para que sea eje instantáneo de rotación de C con respecto a A, se requiere que

$$\overline{w}_{B/A}\ \overline{r}_q + \overline{w}_{C/B}\ \overline{r}_p = 0$$

Es decir, se requiere que

$$\overline{w}_{B/A}\ m\ \overline{w}_{C/B} + \overline{w}_{C/B}\ \overline{r}_p = 0$$

Esta última expresión puede ponerse en la forma

$$\overline{w}_{C/B}\ \overline{r}_p - m\ \overline{w}_{C/B}\ \overline{w}_{B/A} = 0$$

Sacando como factor común a $\overline{w}_{C/B}$:

$$\overline{w}_{C/B}\ (\overline{r}_p - m\ \overline{w}_{B/A}) = 0$$

igualdad que se cumple si $\overline{r}_p - m\ \overline{w}_{B/A} = 0$, o sea, si $\overline{r}_p = m\ \overline{w}_{B/A}$, lo cual quiere decir que P, punto de C, está alojado sobre el eje instantáneo de rotación de B con respecto a A. Esto sería un caso trivial, pues entonces C y A serían el mismo cuerpo. También se cumple la igualdad anterior si $\overline{r}_p - m\ w_{B/A} = n\ w_{C/B}$, en que n es un escalar. Se tendría entonces que

$$\overline{r}_p = m\ \overline{w}_{B/A} + n\ \overline{w}_{C/B}.$$

De esta manera, todos los puntos como P, en que sus vectores de posición $\overline{r}_p$ son linealmente dependientes con $\overline{w}_{B/A}$ y $\overline{w}_{C/B}$, cumplen con la condición de tener velocidad absoluta nula. Es decir, constituyen el eje instantáneo de rotación de C con respecto a A. Además se observa que cuando m y n son simultáneamente nulas, el lugar geométrico pasa por la intersección de los dos primeros ejes, que es lo que se quería demostrar.

COROLARIO.- Los tres ejes son coplanares y guardan la relación: $\overline{w}_{C/A} = \overline{w}_{C/B} + \overline{w}_{B/A}$, y como se demostró que los ejes concurren en un mismo punto, se desprende que son coplanares.

REFERENCIAS

1. J. S. Beggs "Advanced Mechanisms", The Macmillan Co., Nueva York.

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO CINEMATICO DE MAQUINARIA

AUTOMATIC COMPUTATION OF THE SREW PARAMETERS OF RIGID-BODY MOTIONS
PART. I: FINITELY - SEPARATED POSITIONS

JORGE ANGELES

JUNIO, 1984

AUTOMATIC COMPUTATION OF THE SCREW PARAMETERS OF RIGID-BODY MOTIONS.

PART. I: FINITELY - SEPARATED POSITIONS

Jorge Angeles [1]

Abstract

A novel approach, based on invariants, is introduced, that leads to efficient algorithms for computing the screw parameters of rigid-body motions. Both finitely and infinitesimally-separated positions are treated. The computer implementation of the algorithm allows the real-time computation of the parameters defining the position and orientation of a rigid body.

[1] Professor, (ASME Member), DEPFI-UNAM (Universidad Nacional Autónoma de México. Apdo. Postal 70-256. C. Universitaria. 04510 México,D.F. MEXICO.

Introduction

Robotics applications, calling for efficient algorithms for the determination of the position and orientation of a rigid body from a reduced set of measurements, have motivated current research on this topic [1-4] . Although this subject is well known from a theoretical standpoint [5,pp 1-25, 6, 7 & 8, pp 35-62] , the need of means for the efficient real-time computation of the parameters defining a rigid-body motion, commonly referred to as the *screw parameters*, has called for a revisitation of the underlying theoretical basis. In fact, as pointed out in [2, pp85-118]generally acepted formulae can fail to apply under special, though rather frequent, circumstances.

Presented in [2] are algorithms that take into account all possible particular cases. These algorithms, however, are rather lengthy and lack symmetry, in the sense of considering one particular point as a body-fixed reference. The latter item is disadvantageous in applications, as pointed out in [3]. The approach introduced in [3] solves the problem of lack of symmetry, but introduces spurious singularities; it is, additionally, limited to the infinitesimally-separated-positions case. Introduced in the present paper is an algorithm based upon invariant concepts that allow a fast and reliable computation of the screw parameters, for finitely-separated positions. Infinitesimally-separated positions are discussed in an accompanying paper. Given the reduced number of operations involved, this algorithm can be applied to the real-time computation of the said parameters, as required in robotics applications. The procedure, however, is based upon an exact knowledge of the coordinates of three noncollinear points in two distinct configurations of the rigid body which they belog to. In practice, such coordinates are known only up to random measurement errors. Thus, filtering off of these errors requires either taking measurements of over three points of the rigid body or perform redundant computations, as outlined in the paper. The principles presented here, nevertheless, can be applied even if the computations are based upon measurements of over three points. This subject, however, is not discussed here, but only proposed for further research.

Description of the algorithm

The motion associated with two finitely-separated positions of a rigid body is fully described by the following [2,pp 85-119] : a) the axis of the screw, given by the three coordinates of one of its points (preferably the one lying closest to the origin) and three direction cosines, b) the sliding of the screw along its axis, and c) the angle of rotation about the axis of the screw, supplied with sign, given a positive direction defined on the axis. The set of scalar screw parameters of the rigid-body motion is, thus, the following: the three components of a vector r_0, locating point R_0 of the screw axis L, that lies the closest to the origin; the three components of a unit vector e, paralel to L and defining the positive direction along L; two scalars, u and θ, representing the sliding along and the rotation about L. This gives 8 scalar components, which are subject to the following two scalar constraints:

$$e^T e = 1 \quad (1a)$$

$$r_0^T e = 0 \quad (1b)$$

the superscript $(^T)$ standing for transposition. Thus, the number of independent screw parameters is six. The computation is based upon that of the orthogonal matrix defining the rotation involved. The latter is based, in turn, upon the computation of the principal directions of the second-moment tensor of the three points about their centroid. This is equivalent to the moment of inertia of a rigid system composed of three unit-mass particles. Let p_i be the position vector of the i^{th} point, and c, that of their centroid. Hence, c is given by

$$c = \frac{1}{3} \sum_1^3 p_i \quad (2)$$

whereas its second-moment tensor with respect to its centroid, by [9,pp 393-397]

$$I = \sum_1^3 (\rho_i^2 1 - \rho_i \rho_i^T) \quad (3a)$$

with

$$\rho_i \equiv p_i - c_i \quad (3b)$$

The second-rank tensor I is invariant, symmetric and positive definite. The last two properties are obvious from definition (3a). Invariance, on its behalf, means that, under a change of coordinates the three proper values of I do not change, its proper vectors both in the original configuration, $\{e_1, e_2, e_3\}$, and those in the new one, $\{f_1, f_2, f_3\}$, being related by

$$f_i = Q\, e_i \quad (4)$$

where Q is the matrix associated with the rotation involved. The foregoing is illustrated in Fig 1. Moreover, if A denotes the 3 x 3 matrix containing the components of I in the original configuration, whereas

5

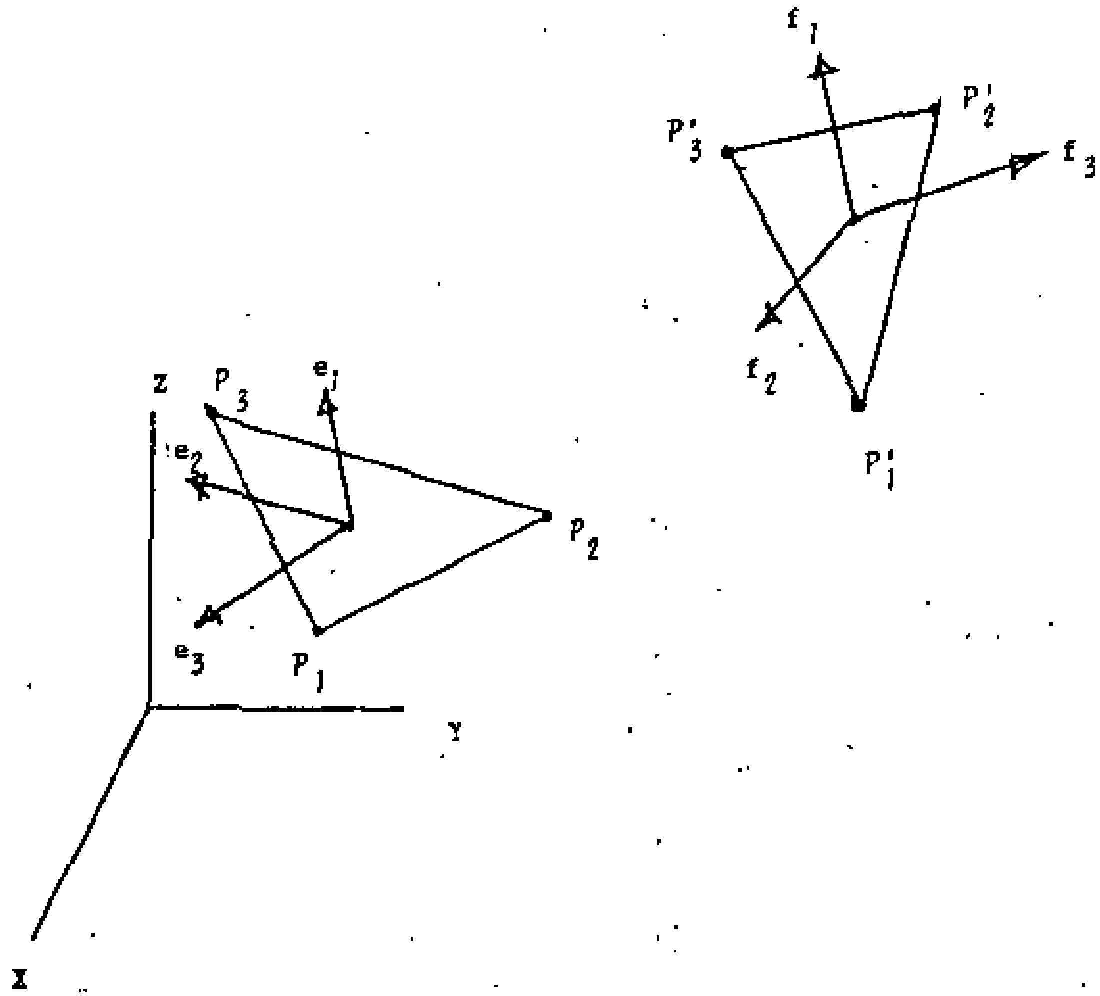


Fig 1 Proper vectors of the second-moment tensor of a three-point rigid system

B, those in the new one, then

$$B = Q A Q^T \qquad (5)$$

Clearly, all matrices appearing in (5) should be expressed in the same coordinate frame. Now, if the three given points are noncollinear, it is a simple matter to verify that tensor I, and hence either matrix A or B, is nonsingular. The aforementioned matrix Q is, of course, proper orthogonal.

The algorithm described next requires the computation of the three proper values and vectors of tensor I. Although this computation leads, in general, to the solution of a nonlinear problem which has to be solved iteratively, for the problem at hand a direct solution is possible, as shown next. Furthermore, once the foregoing eigenvalue problem has been solved, tensor I is expressed with respect to its proper vectors, indentified in what follow with the two orthonormal triplets $\{e_1, e_2, e_3\}$ and $\{f_1, f_2, f_3\}$, of eq (4). Given these two triplets, the computation of Q is a simple matter. The procedure to compute the three proper values and vectors of Q directly(as opposed to iteratively) is, thus, fundamental to this algorithm, for which reason the said procedure is described first.

Let all coordinates be given with respect to a *reference frame*, henceforth referred to as X, Y, Z. Moreover, let A and B be the matrices representing I with respect to the reference frame in the original and the new configuration of the rigid body, respectively. Since tensor I is associated with a plane body, the one defined by the three given points, two of its proper vectors lie in the plane of the body, the remaining one being perpendicular to this plane, as shown in elementary mathematics and mechanics texts. Furthermore, the proper value associated with the third vector, I_3, equals the sum of the proper values, I_1 and I_2, associated with the first two proper vectors. Thus,

$$I_3 = I_1 + I_2 \qquad (6)$$

Next the first two invariants of I, tr (I) and tr (I^2), also called the two first *moments* of I [20, p 67] are expressed both in terms of the proper values of I, and in terms of matrix A, and then equated, which leads to the following system of equations for I_1 and I_2:

$$2 (I_1 + I_2) = \mathrm{tr}\, A \qquad (7a)$$

$$I_1^2 + I_2^2 + (I_1 + I_2)^2 = \mathrm{tr}\, A^2 \qquad (7b)$$

with a similar set of equations for matrix B which, if no roundoff nor measurement errors were present, would be identical. If these are present, then both sets can be used to filter the said errors. Since (7a & b) represent a system of two independent equations for two unknowns, I_1 and I_2, they suffice to compute these.

Elimination of I_2 between eqs (7a & b) produces the following quadratic equation for I_1:

$$4I_1^2 - 2\,(\mathrm{tr}A)\, I_1 + (\mathrm{tr}^2 A - 2\,\mathrm{tr}\, A^2) = 0 \qquad (8)$$

whence

$$I_1 = \frac{\mathrm{tr}A \pm \sqrt{8\,\mathrm{tr}\, A^2 - 3\,\mathrm{tr}^2\, A}}{4}$$

Assuming that the three proper values of I are ordered such that

$$I_1 \leq I_2 < I_3 \qquad (9)$$

then

$$I_1 = \frac{\mathrm{tr}A - \sqrt{8\,\mathrm{tr}\, A^2 - 3\,\mathrm{tr}^2\, A}}{4} \qquad (10a)$$

From eq (7a) it is readily realized that the second root of eq (8) equals I_2, i.e.

$$I_2 = \frac{\mathrm{tr}\,A + \sqrt{8\,\mathrm{tr}\,A^2 - 3\,\mathrm{tr}^2\,A}}{4} \qquad (10b)$$

and hence

$$I_3 = \frac{1}{2}\ \mathrm{tr}\,A \qquad (10c)$$

In order to compute the proper vectors of I, two possibilities are considered, namely either $I_1 < I_2$ or

$I_1 = I_2$ (naturally, to machine precision).

In the first case, the null space of $A - I_i 1$, for $i = 1, 2, 3$, 1 equals the 3×3 identity matrix, is of dimension 1. Thus, any vector f the said space can be normalized to produce e_i, the problem thus reducing to the determination of this space. This is most efficiently done with the aid of Householder reflections [1, pp 111 - 118], which reduce matrix A to a row echelon form. That is to say, if H is the product of the three involved reflections, then

$$H\,(A - I_i 1) = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix} \qquad (11a)$$

with

$$a_1 = [a_{11},\ a_{12},\ a_{13}]^T \qquad (11b)$$

$$a_2 = [0,\ a_{21},\ a_{22}]^T \qquad (11c)$$

$$a_3 = [0,\ 0,\ 0]^T \qquad (11d)$$

Hence,

$$e_i = \frac{a_1 \times a_2}{||a_1 \times a_2||} \qquad (11e)$$

In practice the foregoing computations need be executed only for two proper vectors, the remaining one being computed simply as the cross product of the two previously computed ones.

Now, if $I_1 = I_2$, this means that the three-particle rigid system has a cylindrically symmetric inertia tensor. Since each particle has been assumed of unit mass, this can only happen if the three particles are located at the vertices of an equilateral triangle. Hence, any vector lying in the plane defined by the three points is a proper vector of I. This means that e_1 and e_2 can be chosen arbitrarily within that plane, though mutually orthogonal. In order to uniquely define these vectors, e_1 can be chosen, for instance, as

$$e_1 = \frac{p_1 - c}{\|p_1 - c\|} \tag{12a}$$

Since e_3 is uniquely defined perpendicular to the plane of the given points, e_2 can be readily computed as

$$e_2 = e_3 \times e_1 \tag{12b}$$

Notice that symmetry is not destroyed by the fact of defining arbitrarily vector e_1 as appearing in eq (12a), for the inertia tensor itself is cylindrically symmetric, as said previously.

The set $\{f_i\}_1^3$ should be computed correspondingly, i.e. for the case $\lambda_1 < \lambda_2 < \lambda_3$, f_i should correspond to λ_i; for the second case, $\lambda_1 = \lambda_2$, if e_1 and e_2 are computed as given by eqs (12 a &b), then

$$f_1 = \frac{p_1' - c'}{\|p_1' - c'\|} \tag{13a}$$

and

$$f_2 = f_3 \times f_1$$

p'_i $(i = 1,2,3)$ and c' being the position vectors of the given points and their centroid, respectively, in the new configuration.

The entries of matrix Q, q_{ij} $(i, j = 1,2,3)$, representing the rotation in the $\{e_i\}_1^3$ basis, are now computed by simply recalling the definition of the matrix representation of a linear transformation [12, p 65]. Thus, if both $\{e_i\}_1^3$ and $\{f_i\}_1^3$ are given in reference-frame coordinates, then

$$q_{ij} = e_i^T f_j \tag{14a}$$

The rotation expressed in reference-frame coordinates, referred to as matrix Q_R, is obtained as

$$Q_R = EQE^T \tag{14b}$$

with

$$E = [e_1 \vdots \; e_2 \vdots \; e_3] \tag{14c}$$

Parameters e, in reference-frame coordinates, and θ can now be obtained as

$$e \sin \theta = \text{vect}\,(Q_R), \; \cos\theta = \frac{1}{2}\, \text{tr}\,(Q) - 1 \tag{15a}$$

Now, denoting by q_i the i-th component of vect (Q), the foregoing invariants are given by

$$q_i = \frac{1}{2}\, \epsilon_{ijk}\, q_{kj}, \quad \text{tr}\,(Q) = q_{ii} \tag{15b}$$

and so

$$e \sin \theta = E\, q \tag{15c}$$

The index convention has been used in eq (15b), ϵ_{ijk} being the *alternating tensor* commonly used in tensor analysis. The remaining screw parameters are now computed: Line L is defined here by the position vector r_0 of one of its points, R_0, whose distance to the origin (of the reference frame previously introduced) is a minimum, and by vector e, giving its direction [2, pp 85-119]. Vector r_0 can be computed, in turn, using either a closed-form expression or the minimum-norm solution to an underdetermined linear algebraic system. In the first case, the formula is [2,p 91]:

$$r_0 = \frac{1}{2}\cot\frac{\theta}{2}\, e \times (p_1' - p_1) - \frac{1}{2} e \times e \times (p_1' + p_1) \qquad (16)$$

with similar expressions for vectors p_2, p_2', p_3 and p_3'. The three resulting formulae are, of course, redundant and, if no measurement nor roundoff errors were present, all three would yield one and the same vector r_0. Since such errors are always present, the said three formulae do in fact produce slightly different results. The involved error can be filtered by taking the mean of the three computed values. Alternatively, vector r_0 can be computed as the solution of a linear algebraic system. Indeed, any point R, of position vector r, lying on L satisfies the following equation [2, p 89]:

$$(Q - 1)^T (Q - 1)\, r = (Q - 1)^T (Qp_i - p_i') \qquad (17)$$

which is valid for $i = 1, 2, 3$. None of the three equtions (17) can be solved for r, however, for matrix Q - 1, and hence $(Q-1)^T(Q - 1)$, is singular. In fact, were this matrix nonsingular, then eq (17) would define not a set of points R of L, but one single point. Eq (17) can now be expressed in a more compact form as

$$A\,r = b \tag{18a}$$

[illegible]

$$A \equiv (Q-1)^T (Q-1), \quad b = (Q-1)^T (Qp_{\ell} - \bar{p}_{\ell}) \tag{18b}$$

The point of L whose distance to the origin is a minimum is now solved via the following minimization problem

$$\min_{r} \frac{1}{2} \| r \|^2, \text{ subject to eq (18a)} \tag{19}$$

The solution to problem (19) is well known and can be given in closed form if constraint (18a) is expressed through its two linearly independent equations

$$A' r = b' \tag{20}$$

with A' being a full-rank 2 x 3 matrix. Matrix A' can be readily obtained from (18a) if it is taken into a row echelon form, as discussed before, via Householder reflections. This would imply a corresponding transformation of vector b of eq (18a), for which reason, the right-hand side of eq (20) is changed to b'.

Now problem (19) should be rephrased exchanging eq (18a) for eq (20). The solution to this problem, r_0, is given in closed form by the pseudoinverse of matrix A', namely as

$$r_0 = (A')^{\dagger}\, b' \tag{21a}$$

with

$$(A')^{\dagger} = (A')^T [A' (A')^T]^{-1} \tag{21b}$$

Computing $(A')^{\dagger}$ explicitly, as given by eq (21b), however, is not recommended, for the matrix in brackets is usually ill-conditioned;

in fact, its condition number is the square of that of matrix A',which is a number larger than 1 [13, p 223] . A safe means of computing r_o from eq (20) is now outlined, as recommended by Lawson and Hanson [14,pp 74-76]. Let H be the product of the two Householder reflections reducing $(A')^T$ to row echelon form. Since matrix H is orthogonal, $H^TH = 1$, and hence, eq (20) can be rewritten as

$$A'H^THr = b' \tag{22}$$

Let

$$H(A')^T = T, \quad Hr = y \tag{23a}$$

with T defined as

$$T = \begin{array}{c} \left[\begin{array}{c} U \\ \hline 0 \end{array}\right] \begin{array}{l} \updownarrow 2 \\ \updownarrow 1 \end{array} \\ \leftrightarrow 2 \end{array} \quad , \quad y = \left[\begin{array}{c} y_1 \\ \hline y_2 \end{array}\right] \begin{array}{l} \updownarrow 2 \\ \updownarrow 1 \end{array} \tag{23b}$$

the 2 x 2 matrix U being upper triangular. Eq (22) can thus be expressed as

$$[U^T \; \vdots \; 0] \left[\begin{array}{c} y_1 \\ \hline y_2 \end{array}\right] = b' \tag{24a}$$

or, alternatively, as

$$U^Ty_1 + 0y_2 = b' \tag{24b}$$

Eq (24b) can be readily solved for y_1 by forward substitution, for U^T is lower triangular, y_2 being left undefined so far. However, since y_2 is being multiplied times a zero matrix, it can be given any real value. Since the minimum norm solution to eq (22) is being sought, then the proper choice of y_2 is

$$y_2 = 0 \qquad (25)$$

Thus, the minimum-norm solution to eq (24b), y_0, is given as

$$y_0 = \begin{bmatrix} U^{-T} b' \\ 0 \end{bmatrix} \qquad (26)$$

Vector r_0 is now computed simply as

$$r_0 = H^T y_0 \qquad (27)$$

Again, each value of i produces one corresponding system of equations (17), or, alternatively (18a). All three systems can be solved for r_0, which would produce three slightly different values for this unknown, out of which the mean value would be the one containing the minimum error.

It is pointed out here that the foregoing computations, simple as they are, are sought to be executed with the highest efficiency, for they are aimed at applications demanding real-time computations, while keeping the roundoff error low enough. Finally, the single parameter that is to be computed is u, the sliding of the screw. This is readily computed as the projection on L of the displacement undergone by any of the three given points, i. e. as

$$u = e^T (p_i' - p_i) \qquad 28)$$

which is valid, again, for all three points. Hence, the likeliest u can be taken as the mean value of those three, thereby completing the computation of the parameters sought.

In the foregoing discussion the only source of singularities is tensor I. In fact, if the three given points are collinear, then the orientation of the body about the line defined by the three points is undefined. Practical applications, rather frequent, call for the determination of the position and orientation of a line of a body, and not of the body itself. Such applications arise, for instance, in the positioning of axially-symmetric workpieces or tools. The problem of the determination of the screw parameters for such a situation and others, referred to as incompletely-specified motions, has been studied in [15], whereas the motion of a rigid line in two infinitesimally-separated positions only, has been treated in [16].

Apart from the singular cases mentioned, ill- conditioned problems should be considered, as well. These arise whenever the ratio I_2/I_1 becomes very large, as compared to unity. Such a ratio becomes large when the legs of the triangle defined by the three different points have very different lengths. Hence, the more such a triangle "approaches" an equilateral one, the better conditioned is the problem. Ill- conditioned problems may lead to cancellations in the computation of I_1, as given by eq (10a), for which reason the computation of I_1 and I_2 should be executed in the following order, according to [17,pp20-23]: I_2 is first computed with eq (10b); next I_1 is computed as

$$I_1 = \frac{tr^2 A - 2 tr\, A^2}{I_2} \quad \ldots \qquad (29)$$

An example is next included, that illustrates the foregoing procedure.

Example

Determine the screw parameters of the motion of a rigid body whose original and final configurations, referred to as C and C', respectively, are given by the coordinates of three of its points, P_1, P_2 and P_3 in C, and P_1', P_2 and P_3' in C'. The position vectors of these points are given as:

$$p_1 = [1, 0, 0]^T, \qquad p_1' = [2, 0, -1]^T$$
$$p_2 = [1, 1, 0]^T, \qquad p_2' = [2, 0, 0]^T$$
$$p_3 = [2, 1, -1]^T, \qquad p_3' = [3, -1, 0]^T$$

Hence,

$$c = \frac{1}{3}[4, 2, 1]^T, \quad c' = \frac{1}{3}[7, -1, -1]^T$$

c and c' being the position vectors of the centroids in and '', respectively. Vectors ρ_i, ρ_i', defined as $\rho_i = p_i - c$, and $\rho_i' = p_i' - c'$, $i = 1, 2, 3$, are thus

$$\rho_1 = \frac{1}{3}[-1,-2,1^T] \quad \rho_1' = \frac{1}{3}[-1, 1, 2]^T$$
$$\rho_2 = \frac{1}{3}[-1,1,1^T] \quad \rho_2' = \frac{1}{3}[-1, 1, 1]^T$$
$$\rho_3 = \frac{1}{3}[2,1,-2^T] \quad \rho_3' = \frac{1}{3}[2, -2, 1]^T$$

Matrices A and B are thus computed as

$$A = \frac{1}{3}\begin{bmatrix} 4 & -1 & 2 \\ -1 & 4 & 1 \\ 2 & 1 & 4 \end{bmatrix}, \quad B = \frac{1}{3}\begin{bmatrix} 4 & 2 & -1 \\ 2 & 4 & 1 \\ -1 & 1 & 4 \end{bmatrix}$$

Hence,

$$\mathrm{tr}A = 4, \quad \mathrm{tr}\,A^2 = 20/3$$

Eqs (10a - c) thus yield

$$I_1 = 1 - \frac{\sqrt{3}}{3}, \quad I_2 = 1 + \frac{\sqrt{3}}{3}, \quad I_3 = 2$$

Now e_3, the unit vector spanning the null space of $A - I_3 1$, is determined. This matrix is

$$A - I_3 1 = \frac{1}{3}\begin{bmatrix} -2 & -1 & 2 \\ -1 & -2 & 1 \\ 2 & 1 & -2 \end{bmatrix}$$

which can be taken to the following row echelon form via Householder reflections:

$$H (A - I_3 1) = \frac{1}{3}\begin{bmatrix} 3 & 2 & -3 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Thus, letting

$$a_1 = [3, 2, -3]^T, \quad a_2 = [0, \sqrt{2}, 0]^T$$

in eq (11c), one readily obtains

$$e_3 = [0.7071, 0, 0.7071]^T$$

Similarly,

$$e_2 = [0.3251, -0.888, -0.3251]^T$$

and

$$e_1 = e_2 \times e_3 = [-0.6280, -0.4597, 0.6280]^T$$

Similar results are obtained for matrix B as follows:

$f_1 = [-0.6280, 0.6280, -0.4597]^T$

$f_2 = [0.3251, -0.3251, -0.8880]^T$

$f_3 = [-0.7071, -0.7071, 0.0]^T$

Hence, the rotation matrix Q is obtained from formula (14) as

$$Q = \begin{bmatrix} -0.1833 & -0.6124 & 0.7691 \\ -0.6124 & 0.6833 & 0.3980 \\ \text{[illegible]} & -0.3980 & 0.5000 \end{bmatrix}$$

and

$$Q_R = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Thus,

$$e \sin\theta = \frac{1}{2}\,[\,1, -1, -1\,]^T, \quad \cos\theta = -\frac{1}{2}$$

i.e.

$$e = \frac{\sqrt{3}}{3}[\,-1,\ 1,\ 1\,]^T, \quad \theta = 4\pi/3$$

Eq (17) is now written for $i = 1$, which yields eq (18a) with

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & -1 \\ 1 & -1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$$

Householder reflections applied to A and b yield

$$A' = \begin{bmatrix} -\sqrt{6} & -\sqrt{6}/2 & -\sqrt{6}/2 \\ 0 & -3\sqrt{2}/2 & 3\sqrt{2}/2 \end{bmatrix}, \quad b' = \begin{bmatrix} -3\sqrt{6}/2 \\ -\sqrt{2}/2 \end{bmatrix}$$

Hence, the minimum-norm solution of eq (24a) is

$$y_0 = [-\frac{\sqrt{6}}{2},\ -\frac{\sqrt{2}}{6},\ 0]^T$$

and so,

$$x_0 = [1,\ \frac{2}{3},\ \frac{1}{3}]^T$$

Finally,

$$u = e^T (p_j' - p_j) = \frac{\sqrt{3}}{3}[-1, 1, 1] \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = -\frac{2\sqrt{3}}{3}$$

thereby completing the solution

Conclusions

An algorithm was presented, that allows the efficient computation of the screw parameters of the motion undergone by a rigid body between two finitely-separated positions. The efficiency of the algorithm refers to its low number of operations, as well as to its avoidance of large roundoff errors and of spurious singularities. Means of avoiding ill conditioning were discussed. Ways of filtering roundoff and/or measurement errors were outlined, but not treated in detail, since this fall outside the scope of the paper. The computation of the screw parameters for two infinitesimally-separated positions is discussed in an accompanying paper [18].

Acknowledgements

The research work reported here was totally supported by the Graduate Division of the Faculty of Engineering - National Autonomous University of Mexico (DEPFI-UNAM) and was completed at the CAD Laboratory of this Division.

References

1. Chen N-Y and Birk J.R, "Estimating workpiece pose using the feature points method", IEEE Trans Automatic Control, Vol AC-25, No. 6, Dec 1980, pp 1027-1041

2. Angeles J, Spatial Kinematic Chains. Analysis, Synthesis, Optimization, Springer-Verlag, Berlin, 1982

3. Lanb A J and Shiflett G R, "A linear algebra approach to the analysis of rigid body velocity from position and velocity data", Trans ASME J Dynamic Systems, Measurement, and Control, Vol 105, June 1983, pp 92-95

4. Foral R D and Seireg A A, "Fast tracking and handling of moving objects: A new method", CIME, Sept 1983, pp 32-38

5. Whittaker E T, A Treatise on the Analytical Dynamics of Particles and Rigid Bodies, Cambridge University Press, Cambridge, (Great Britain), 1961.

6. Rodrigues O, "Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés independamment des causes qui peuvent les produire", Journal des Mathématiques Pures et Appliquées, Vol 5, 1st Series, 1840, pp 380-440

7. Meyer P, "Zur Geometric affin-veränderlicher räumlicher Systeme", Mechanism and Machine Theory, Vol 10, 1975, pp 217-232

8. Bottema O and Roth B, Theoretical Kinematics, North-Holland Publishing Company, Amsterdam-New York-Oxford, 1979

9. Fox E A, Mechanics, Harper and Row, New York, 1967

10. Leigh D C, Nonlinear Continuum Mechanics, Mc Graw-Hill Book Co, New York, 1968

11. Businger P and Golub G H, "Linear least square solution by Householder transformations", in Wilkinson J H and Reinsch C (editors), Handbook for Automatic Computation. Vol II: Linear Algebra, Springer - Verlag, Berlin-Heidelberg - New York, 1971

12. Halmos P R, Finite-Dimensional Vector Spaces, Springer-Verlag, New York, 1970

13. Stewart G W, Introduction to Matrix Computations, Academic Press, Inc, London, 1973

14. Lawson C L and Hanson R J, Solving Least Squares Problems, Prentice-Hall, Inc, Englewood Cliff, N J, 1974

15. Tsai L-W and Roth B, "Incompletely specified displacements: Geometry and spatial linkage synthesis", ASME Paper No 72-Mech-13

16. Henderson J M and Meriam J L, "On the space rotation of a two-point link", Mechanism and Machine Theory, Vol 10, 1975, pp 347-354

17. Forsythe G E, Malcolm M A and Moler C B, Computer Methods for Mathematical Computations, Prentice-Hall, Inc, Englewood Cliff, N J, 1977.

18. Angeles J, "Automatic computation of the screw parameters of rigid-body motions. Part II: Infinitesimally-separated positions", submitted for publication to Trans ASME J Dynamic Systems Measurements, and Control

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO CINEMATICO DE MAQUINARIA

AUTOMATIC COMPUTATION OF THE SCREW PARAMETER OF RIGID-BODY MOTIONS

PART. II: INFINITESIMALLY - SEPARATED POSITIONS

DR. JORGE ANGELES ALVAREZ

JUNIO, 1984.

Palacio de Minería Calle de Tacuba 5 primer piso Deleg. Cuauhtémoc 06000 México, D.F. Tel.: 521-40-20 Apdo. Postal M-2285

AUTOMATIC COMPUTATION OF THE SCREW PARAMETERS OF RIGID-BODY MOTIONS. PART II: INFINITESIMALLY-SEPARATED POSITIONS.

Jorge Angeles[1]

Abstract

The approach introduced in an accompanying paper, aimed at the computation of the screw parameters of a rigid-body motion defined by two finitely-separated positions, is now applied to that defined by two infinitesimally-separated positions. Given the economy of computation of this algorithm, it should allow the real-time computation of the screw parameters under study. The algorithm assumes perfect knowledge of the position and the velocity of three noncollinear points of the body.

1 Professor, (ASME Member), DEPFI-UNAM (Universidad Nacional Autónoma de México. Apdo. Postal 70-256. C. Universitaria. 04510 México, DF., MEXICO).

Introduction

A novel approach, aimed at the computation of the screw parameters of a rigid-body motion defined by two finitely-separated positions, is introduced in an accompanying paper [1]. Many an application, either in robotics or in mechanism design, however, require the computation of the said parameters when the motion is defined by two not finitely-, but infinitesimally-separated positions. Moreover, since infinitesimally-separated positions give rise to linear problems, whereas finitely-separated positions, to nonlinear ones, the latter are solved frequently by first solving the former, and then performing a time integration.

The algorithm introduced here is based upon that presented in [2], but modifies it in the sense of eliminating the spurious singularity contained therein. Moreover, the computations are simplified and the procedure extended to the computation of all the independent parameters of the screw motion under study.

Description of the algorithm

The motion defined by two infinitesimally-separated positions of a rigid body is fully described by the following [3, pp 119-148]: a) the axis of the instantaneous screw, given by the position vector of one of its points (preferably the one lying the closest to the origin) and three direction cosines, b) the sliding of the screw along its axis, and c) the rate of rotation about the axis, supplied with sign, given a positive direction defined on the axis. The set of scalar screw parameters of the rigid body motion is thus, the following: the three components of a vector r_0, locating point R_0 of the screw axis L, whose distance to the origin is a minimum; the three components of a vector e, parallel to L and defining the positive direction along L; two scalars, $\dot{u}$ and $\dot{\theta}$, representing the sliding along and the rate of rotation about L. This gives 8 scalar components, which are subject to:

$$e^T e = 1$$

$$r_0^T e = 0$$

the superscript (^{T}) standing for transposition. The number of independent scalar parameters is, thus, six. The computation of the foregoing parameters has been treated in detail in [3], the algorithm implementing the corresponding procedure becoming lengthy and complicated. A simplified algorithm is presented in [2] that is symmetric in the sense of giving no preference to any of the three given points. It introduces, however, a spurious singularity, dependent upon the location of the origin of the coordinate frame.

.../...

To describe the present algorithm, let p_i ($i = 1, 2, 3$) be the position vectors of three noncollinear points P_i ($i = 1, 2, 3$) of a rigid body, their velocities being denoted by $\dot{p}_i$ ($i = 1, 2, 3$). Now the angular velocity ω is computed as follows:

Let c be the centroid of the three points, v being its velocity. These two vectors are clearly given by

$$c = \frac{1}{3} \sum_1^3 p_i, \quad v = \frac{1}{3} \sum_1^3 \dot{p}_i \tag{1}$$

The velocity of any point of the body can be expressed in terms of that of one point A, $\dot{a}$, whose position vector is represented by a, and the angular velocity ω of the body. If the angular velocity matrix Ω is used instead of vector ω, then the foregoing relation, written for each point P and C, is

$$\dot{p} = \dot{a} + \Omega(p_i - a), \; i = 1, 2, 3 \tag{2a}$$

$$v = \dot{a} + \Omega(c - a) \tag{2b}$$

The relation between ω and Ω is

$$\omega = \text{vect}\ (\Omega) \tag{3a}$$

or

$$\omega_i = \frac{1}{2} \varepsilon_{ijk}\Omega_{kj} \tag{3b}$$

where the standard index notation is being used, ε_{ijk} being the alternating tensor. Subtraction of eq (2b) from eq (2a) yields

$$\dot{p}_i - v = \Omega(p_i - c), \; i = 1, 2, 3 \tag{4}$$

.../...

Following the approach introduced in [2], matrices $\dot{P}$ and P are next defined as

$$\dot{P} = \left[\dot{p}_1 - v \;\vdots\; \dot{p}_2 - v \;\vdots\; \dot{p}_3 - v \right] \;, \; P = \left[p_1 - c \;\vdots\; p_2 - c \;\vdots\; p_3 - c \right] \tag{5}$$

All three relations (4) can thus be expressed as

$$\dot{P} = \Omega P \tag{6}$$

which is an equation not depending upon the location of the origin. Thus, the algorithm is not affected if two of the three given points turn to be collinear with the origin. The algorithm, moreover, does not depend upon the regularity of matrix P; indeed, this matrix, as given by definitions (5), is identically singular, but this is no drawback, as shown next.

Taking the vector of both sides of eq (6) produces [4]:

$$\frac{1}{2}\,(1\ \mathrm{tr}P - P)\omega = \mathrm{vect}(\dot{P}) \tag{7}$$

which can be solved for ω provided matrix $M = 1\ \mathrm{tr}\ P - P$ is invertible. If this is the case, then

$$\omega = 2\ (1\ \mathrm{tr}\ P - P)^{-1}\ \mathrm{vect}\ (\dot{P}) \tag{8}$$

matrix M becoming singular only if P becomes a rank-one matrix which, in turn, implies that the three given points are collinear. This fact is supported by the following

<u>Theorem 1</u>. The trace of matrix P, as defined in eq (5), is identically different from zero.

Proof:

With no loss of generality, axes X-Y-Z are assumed to be orientated so that Z is perpendicular to the plane defined by the three given points. Furthermore,

.../...

the origin is placed at C, whereas X is orientated parallel to line P_3P_1, so that $x_1 > 0$. Hence, $y_1 = y_3 < 0$, $y_2 = -2y_1 > 0$, the corresponding layout appearing in Fig 1

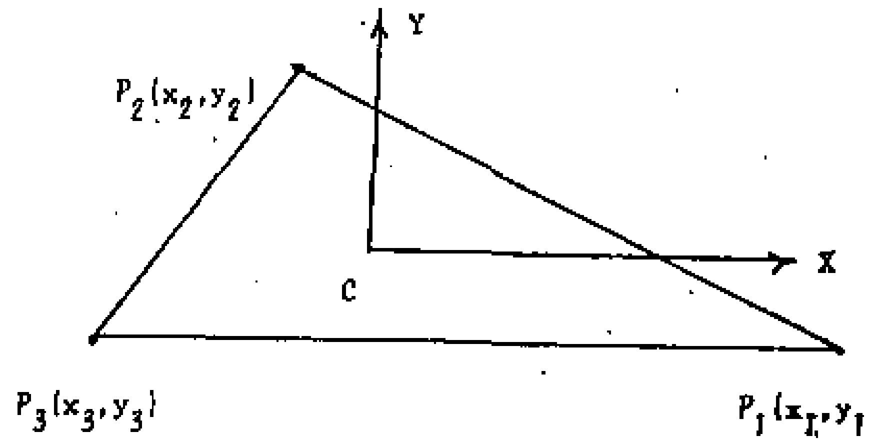


Fig 1   X-Y plane containing the three given points

Matrix P is th~~is~~us given by

$$P = \begin{bmatrix} x_1 & x_2 & -(x_1 + x_2) \\ y_1 & -2y_1 & y_1 \\ 0 & 0 & 0 \end{bmatrix}$$

Hence,

$$\mathrm{tr}\, P = x_1 - 2y_1$$

which, under the foregoing assumptions, is the sum of two positive real numbers, and hence, never vanishes, q. e. d.

<u>Theorem 2.</u> Matrix $1\,\mathrm{tr}P - P$, with P defined as in eq(5), is singular if, and only if the three given points P_i ($i = 1, 2, 3$) are collinear.

Proof:

Only necessity will be proved, for sufficiency can be proved by inverting the necessity proof given. To this end, let π_i and μ_i be the proper values of P and $M = 1\ \mathrm{tr}\ P - P$, respectively. Given the definition of M, these values are related by

$$\mu_i = \mathrm{tr}\ P - \pi_i\ ,\ i = 1,\ 2,\ 3 \tag{9}$$

However, the three columns of matrix P represent the components of three coplanar vectors, hence P is singular. Thus, at least one proper value of P vanishes. Let $\pi_3 = 0$. Hence,

$$\mathrm{tr}\ P = \pi_1 + \pi_2 \tag{10a}$$

and

$$\mu_1 = \mathrm{tr}\ P - \pi_1\ ,\ \mu_2 = \mathrm{tr}\ P - \pi_2\ ,\ \mu_3 = \mathrm{tr}\ P \tag{10b}$$

Therefore, the determinant of M, det(M), is given by

$$\det(M) = \mu_1\ \mu_2\ \mu_3 = \pi_1\ \pi_2\ \mathrm{tr}\ P\ . \tag{11}$$

where relations (10a & b) have been taken into account. From eq(11) and Theorem 1, it is clear that det(M) vanishes only if at least one of π_1 and π_2 does. However, both of them cannot vanish simultaneously, for this would imply $P_1 = P_2 = P_3$. The vanishing of, say π_1 , implies that matrix P has two vanishing proper values, i. e. that it is a rank-one matrix. This implies, in turn, that the three column vectors of P are parallel to one single vector, i. e. that the three points are collinear, thereby completing the proof.

Now, if the three points are given along the edges of a right-angled trihedron, so that their distances to the apex are identical, and vector c is defined now not as the centroid, but as the position vector of the apex, then a proper normalization renders matrix P orthogonal. In this case, matrix $1\ \mathrm{tr}\ P - P$ can be inverted explicitly as [4]:

.../...

$$(1 \operatorname{tr} P - P)^{-1} = \frac{1}{\operatorname{tr}^2 P - 1} (P \operatorname{tr} P + P^T) \tag{12}$$

In this case, matrix $1 \operatorname{tr} P - P$ turns to be singular if, and only if, P represents a rotation of the trihedron, with respect to the given reference frame, of an angle of rotation $\theta = \pi/2$, π or $3\pi/2$. This singularity can be readily removed by introducing a redefinition of the Cartesian axes attached to the reference frame, as discussed in [4].

In general, however, and particularly if c is defined as in eq (1), matrix P is not orthogonal. The computation of ω can be executed safely if matrix $1 \operatorname{tr} P - P$ is not only nonsingular, but also well conditioned. Ill conditioning of this matrix arises only when the three given points, though not collinear, are close to it. A measure of the closeness to collinearity can be given, thus, by the inverse of the condition number [5] of this matrix. Most linear-equation solvers supply the user with a good estimate of the said number.

The remaining parameters are now computed. Given ω, it is a simple matter to construct Ω. Symbolically they are related by

$$\Omega = 1 \times \omega \tag{13}$$

where $\times$ denotes the standard cross product, in dyadic notation.

Now, vector r_o defining the position of point R_o on the instantaneous screw axis, whose distance to the origin is a minimum, can be computed in two alternate ways, one resorting to an explicit formula, the second one, via a minimization problem. Both approaches are now discussed.

.../...

The formula giving r_o is [3, p 129]:

$$r_o = p_i + \frac{1}{\omega^2} \omega \times p_i - (\omega \cdot p_i)\omega \tag{14}$$

which is valid for $i = 1, 2, 3$. If no roundoff nor measurement errors were present, formula (14), as applied to all three points, would yield one and the same value for r_o. In practice, this is not the case; thus the said errors can be filtered if the formula is applied to the three given points, then defining r_o as the mean value of the three distinct values thus obtained.

Alternatively, r_o can be computed via the following optimization problem [3, pp 126-129]:

$$\text{Min} \ \frac{1}{2} r^T r \tag{15a}$$

subject to

$$\Omega[\dot{p}_i + \Omega(r - p_i)] = 0 \tag{15b}$$

Matrix Ω being 3 x 3 and skew-symmetric, is singular; hence r cannot be solved for from eq (15b). In fact, if r could be solved for from that equation, then the said equation would produce one single value of r, not a set, that defining the screw axis. Eq (15b) contains exactly 2 linearly independent equations, for rank $(\Omega) = 2$. These can be readily extracted from eq (15b) if Householder reflections are applied to both sides of eq (15b), as discussed in [1]. This would yield:

$$Ar = b \tag{16a}$$

.../...

with

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & 0 \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix}$$

Eq (16a) represents then an underdetermined linear algebraic system possessing infinitely many solutions, its minimum- (Euclidean) norm solution being given by the generalized inverse of the upper 2 x 3 submatrix of A, referred to as A_u. Analogously, let b_u be the upper 2-dimensional subvector of b. Thus,

$$r_o = A_u^+ b_u \tag{17a}$$

with

$$A_u^+ \equiv A_u^T (A_u A_u^T)^{-1} \tag{17b}$$

Given the frequent ill conditioning of matrix $A_u A_u^T$, it is not recommended to invert this matrix explicitly. In fact, r_o can be more efficiently computed resorting again to Householder reflections, as discussed in [1]. This would produce r_o as follows: let H be the product of Householder reflections rendering A_u^T upper triangular. Then

$$H A_u^T = T = \begin{array}{c} \leftarrow 2 \rightarrow \\ \begin{bmatrix} U \\ --- \\ 0 \end{bmatrix} \end{array} \begin{array}{c} 2 \\ 1 \end{array} \tag{18a}$$

U being upper triangular. Vector y_o is computed from

$$y_o = \begin{bmatrix} y_u \\ --- \\ 0 \end{bmatrix} \begin{array}{c} 2 \\ 1 \end{array}, \quad U^T y_u = b_u \tag{18b}$$

from which the 2-dimensional vector y_u can be readily computed, since U^T is lower triangular. Then

$$r_o = H^T y_o \tag{18c}$$

.../...

Clearly, each value of p_i and $\dot{p}_i$ (i = 1, 2, 3) would produce one pair (A, b), to form then three different minimization problems (15a & b). Again, if no roundoff nor measurement errors were present, then all three problems would produce one and the same value r_0. Since such errors are always present, the likeliest value of r_0 can be chosen as the mean value of the three values thus obtained.

Having determined ω, e and $\dot{\theta}$ can be readily computed from

$$\omega = e\,\dot{\theta} \tag{19}$$

with e defined as a unit vector. Thus, only parameter $\dot{u}$ need be computed. This is done simply by projecting any of $\dot{p}_i$ (i = 1, 2, 3) on e, thus obtaining

$$\dot{u} = e^T \dot{p}_i \tag{20}$$

which would again produce three slightly different values, out of which the likeliest would be their mean value.

Example

Given the three position vectors p_i of points P_i, as well as their corresponding velocities, $\dot{p}_i$, for i = 1, 2, 3, all vectors referred to the same coordinate frame X-Y-Z, and shown next, determine the parameters defining the instantaneous screw of the corresponding motion.

.../...

$$p_1 = [1, \quad 1, \quad 7]^T, \quad \dot{p}_1 = [7, \quad -5, \quad 1]^T$$
$$p_2 = [4, \quad 7, \quad 1]^T, \quad \dot{p}_2 = [-5, \quad 4, \quad 4]^T$$
$$p_3 = [7, \quad 10, \quad 10]^T, \quad \dot{p}_3 = [1, \quad -2, \quad 4]^T$$

The vectors given above are taken from Example 1 of [2] for comparison purposes. They define in fact the positions and the velocities of three points of a rigid body. Indeed, the following compatibility condition holds for the said vectors:

$$(\dot{p}_i - \dot{p}_j)^T(p_i - p_j) = 0, \ i = 1, 2, 3; \quad j = 1, 2, 3; \quad j \neq i$$

Now c and v are computed as given in eq (1):

$$c = [4, 6, 6]^T, \quad v = [1, -1, 3]^T$$

Matrices P and $\dot{P}$, defined in eq (5), are given next:

$$P = \begin{bmatrix} -3 & 0 & 3 \\ -5 & 1 & 4 \\ 1 & -5 & 4 \end{bmatrix}, \quad \dot{P} = \begin{bmatrix} 6 & -6 & 0 \\ -4 & 5 & -1 \\ -2 & 1 & 1 \end{bmatrix}$$

Hence,

$$(1 \text{ tr } P - P) = \begin{bmatrix} 5 & 0 & -3 \\ 5 & 1 & -4 \\ -1 & 5 & -2 \end{bmatrix}, \quad (1 \text{ tr } P - P)^{-1} = \frac{1}{12}\begin{bmatrix} 18 & -15 & 3 \\ 14 & -13 & 5 \\ 26 & -25 & 5 \end{bmatrix}$$

$$\text{vect}(\dot{P}) = [1, 1, 1]^T$$

.../...

Substitution of the foregoing matrices and vectors in eq (8) produces

$\omega = [1, 1, 1]^T$

thus obtaining the same value of ω as that reported in [2]. The remaining parameters are now obtained, To this end, matrix Ω is now computed. From eq (13),

$$\Omega = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 1 \end{bmatrix}$$

Now eq (15b) is expressed, for $i = 1$, as

$$\begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

x, y and z being the components of vector r of that equation, as yet to be determined. This is done as the solution to problem (15a & b), which is readily obtained by application of Householder reflections to both sides of the last equation, to render it into row echelon form. Letting H and A be the product of the two reflections required for the aforementioned transformation, and the matrix of the system involved, one has

$$HA = \begin{bmatrix} 2\alpha & -\alpha & -\alpha \\ 0 & \beta & -\beta \\ 0 & 0 & 0 \end{bmatrix}, \quad \alpha \equiv \frac{\sqrt{6}}{2}, \quad \beta = \frac{3\sqrt{2}}{2}$$

Hence, eq (15b) represents the null space of matrix HA given above. Any

.../...

vector r of that space has clearly the following components:

$$r = \| r \| [\, 1,\ 1,\ 1\,]^T$$

$\| r \|$ representing its magnitude. Hence the minimum-norm vector r_o, solving problem (15a & b) is simply

$$r_o = [\, 0,\ 0,\ 0\,]^T$$

for the axis of the instantaneous screw, given by vector r, as found above, passes through the origin.

Parameters $\dot{\theta}$ and e are readily obtained from vector ω, as

$$\dot{\theta} = \| \omega \| = \sqrt{3}, \quad e = \frac{\omega}{\dot{\theta}} = \frac{\sqrt{3}}{3} [\, 1,\ 1,\ 1\,]^T$$

Finally, parameter $\dot{u}$ is found simply as the projection of any of $\dot{p}$ on e. For instance, for $i = 1$,

$$\dot{u} = e^T \dot{p}_i = 0$$

and hence the motion is a pure rotation about the origin. The following remarks are now in order:

a) This problem turned to be very simple to solve, given that it reduced, in the last stage of computing r_o, to finding the null space of a rank-2 2 x 3 matrix. In the general case, it would have given rise to an under-determined linear system of 2 equations in 3 unknowns, whose minimum-norm solution would have been determined as outlined in [1].

b) The solution reported in [2] did not make evident that the motion of this example is a pure rotation about the origin.

Conclusions

A method was presented that allows the automatic computation of the screw parameters of a motion defined by two infinitesimally-separated positions. This method is simpler than previously reported ones, making use of a smaller number of operations. The latter feature can allow the real-time computation of the said parameters, which is essentially necessary in robotics applications. The method does not depend on the location of the origin of the coordinate reference frame; but its orientation can produce a spurious singularity, that can be readily removed. Furthermore, it gives no preference to any of the three given points. It fails only if the three given points are collinear, but in this case no method can provide the parameters sought, for they are undetermined. A possible source of numerical instability is contained only in matrix 1 tr P - P, this matrix becoming ill conditioned as the three given points approach collinearity. Hence, this matrix is better conditioned, the more triangle $P_1P_2P_3$ approaches an equilateral triangle.

Acknowledgements

The research work reported here was totally supported by the Graduate Division of the Faculty of Engineering - National Autonomous University of Mexico (DEPFI-UNAM) and was completed at the CAD Laboratory of this Division.

References

1. Angeles J, "Automatic computation of the screw parameters of rigid-body motions. Part I: Finitely-separated positions", submitted for publication to: Trans ASME J of Dynamic Systems, Measurement, and Control.

.../...

2. Laub A J and Shiflett G R, "A linear algebra approach to the analysis of rigid body velocity from position and velocity data", Trans ASME J of Dynamic Systems, Measurements, and Control, Vol 105, June 1983, pp 92-95.

3. Angeles J, Spatial Kinematic Chains: Analysis, Synthesis, Optimization, Springer-Verlag, Berlin, 1982.

4. Angeles J, "Software for the analysis of seven-link revolute-coupled kinematic chains", Internal Report, DEPFI-UNAM, Mexico City, 1984.

5. Forsythe G E and Moler C B, Computer Solution of Linear Algebraic Systems, Prentice-Hall, Inc, Englewood Cliffs, NJ, 1967, pp 20-26.

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISENO CINEMATICO DE MAQUINARIA

SUBPROGRAMAS PARA EL ANALISIS DE CADENAS CINEMATICAS CON 7 ESLABONES ACOPLADOS MEDIANTE PARES DE ROTACION

DR. JORGE ANGELES ALVAREZ

JUNIO, 1984.

Palacio de Minería Calle de Tacuba 5 primer piso Deleg. Cuauhtémoc 06000 México, D.F. Tel.: 521-40-20 Apdo. Postal M-2285

SUBPROGRAMAS PARA EL ANALISIS DE CADENAS CINEMATICAS CON 7 ESLABONES ACOPLADOS MEDIANTE PARES DE ROTACION

J. Angeles A.
A. A. Rojas S.
División de Estudios de Posgrado y de Ingeniería
Mecánica y Eléctrica
Facultad de Ingeniería, UNAM
Apdo. Postal 70-256
04510 MEXICO, D.F.

Abstract

Computer subprograms for the analysis of seven-link revolute-coupled kinematic chains, both closed and open are presented. The subprograms are based upon an algorithm that solves the displacement kinematic equations that were obtained from invariance concepts. Newton-Raphson's method was used to solve the displacement equations. These subprograms can be used in real time, thus allowing the control of robot manipulators.

Resumen

Se presentan subprogramas para el análisis de cadenas cinemáticas abiertas o cerradas, compuestas por 7 eslabones acoplados mediante pares de rotación. Los subprogramas están basados en un algoritmo que resuelve las ecuaciones cinemáticas que se obtuvieron utilizando conceptos de invariancia. Se emplea el método de Newton-Raphson para obtener la solución de las ecuaciones cinemáticas señaladas. Estos subprogramas pueden funcionar en tiempo real para el control de manipuladores.

Introducción

El problema del cálculo de los 6 ángulos desconocidos en una cadena cinemática compuesta por 7 eslabones acoplados mediante pares de rotación se conoce como *problema cinemático inverso* y ha atraído la atención de muchos investigadores, sobre todo en los últimos 10 años [1-4] por su aplicabilidad en robots manipuladores. En los trabajos existentes se sigue básicamente la misma tendencia, esto es, reducir las ecuaciones cinemáticas involucradas a un polinomio en una incógnita; esto ha dado lugar a que, para obtener soluciones, se tengan que hallar raíces de determinantes de matrices con elementos polinómicos. Por ejemplo, en [2] se trabaja con una matriz de 12x12, cuyos elementos son de 4° grado en la tangente de la mitad del ángulo de entrada, en tanto que en [4] se establece una matriz de 16x16 con elementos de 2° grado en el mismo argumento. Posteriormente, Albala [5] obtiene una matriz de 16x16 con elementos de 2° grado en la tangente de la mitad del ángulo de salida, lo cual consume demasiado tiempo para obtener una solución aceptable, pues esta formulación produce un excesivo error de redondeo. Los ángulos intermedios, $\theta_2, \ldots, \theta_6$, se calculan hallando las raíces de polinomios de grado 4° o menor. Sin embargo, en el proceso de eliminación de los ángulos intermedios se introducen raíces espurias, por lo que debe verificarse cada solución hallada [6].

El algoritmo que aquí se emplea se obtiene aplicando conceptos de invariancia que permiten un cálculo muy eficiente de las derivadas involucradas, lo cual, hasta la fecha, no ha sido aplicado por ningún investigador. Esto da lugar a una solución directa para una posición especificada [7]; con esto se puede lograr, con un número muy limitado de operaciones por iteración y con un error de redondeo sumamente bajo, el análisis de cadenas cinemáticas, abiertas o cerradas, compuestas de siete eslabones articulados, mediante pares de rotación. En cadena abierta, los datos del problema son las variables que definen el movimiento del órgano terminal (OT) en tanto que, en cadena cerrada, son las que definen el movimiento del eslabón motor o "de entrada".

Descripción del algoritmo

Las ecuaciones para el análisis de desplazamiento de una cadena cinemática de sexto grado de libertad pueden obtenerse de las condiciones de cerradura de desplazamiento y de rotación que se describen en seguida. De acuerdo con el método y la notación de Denavit y Hartenberg, [5, 8], los n eslabones de una cadena cinemática se numeran ordenadamente, de 1 a n y al i° eslabón se fija el sistema coordenado X_i, Y_i, Z_i. Así, $[Q_{i,i+1}]_i$ representa una matriz ortogonal, referida a los ejes X_i, Y_i, Z_i que gira éstos a una posición coincidente con los correspondientes $X_{i+1}, Y_{i+1}, Z_{i+1}$; por su parte, $[a_{i,i+1}]_i$ es el vector que une los orígenes O_i y O_{i+1} de los ejes anteriores, dirigido del primero al segundo y referido a ejes fijos al i° eslabón. Así, las condiciones de cerradura son (ver Fig 1):

$$[Q_{1,2}]_1 \, [Q_{2,3}]_2 \cdots [Q_{6,7}]_6 = [Q]_1 \qquad (1)$$

para rotación y

$$[a_{1,2}]_1 + [a_{2,3}]_1 + \cdots + [a_{6,7}]_1 = [r]_1 \qquad (2)$$

para desplazamiento.

Las ecuaciones (1) y (2) constituyen un sistema de 12 ecuaciones escalares, de las cuales sólo 6 son independientes. Se suponen conocidos $[Q]_1$, que representa la rotación para girar los ejes X_1, Y_1, Z_1 a una orientación idéntica a la de los ejes X_7, Y_7, Z_7, así como $[r]_1$, que es el vector de posición del punto P del OT (origen del 7° sistema coordenado). Aplicando la invariancia del vector axial [9] se obtiene:

$$\text{vect}\ (\underset{\sim}{Q}) = \underset{\sim}{e}\ \text{sen}\ \phi \qquad (3)$$

siendo $\underset{\sim}{e}$ el vector característico real asociado con el valor característico real +1 y ϕ, el ángulo de rotación. Combinado (1) y (3) se tiene:

$$\text{vect}\ (\left[\underset{\sim}{Q}_{1,2}\right]_1\left[\underset{\sim}{Q}_{2,3}\right]_2 \cdots \left[\underset{\sim}{Q}_{6,7}\right]_6) = \underset{\sim}{e}\ \text{sen}\ \phi \qquad (4)$$

Las ecuaciones (2) y (4) constituyen un sistema algebraico no lineal de sexto orden de la forma:

$$\underset{\sim}{f}(\underset{\sim}{\theta}) = \begin{bmatrix} \underset{\sim}{f}_1(\underset{\sim}{\theta}) \\ \underset{\sim}{f}_2(\underset{\sim}{\theta}) \end{bmatrix} = \underset{\sim}{0} \qquad (5a)$$

donde:

$$\underset{\sim}{\theta} = \left[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\right]^T$$

$$\underset{\sim}{f}_1(\underset{\sim}{\theta}) = 2\ \text{vect}\ \left(\left[\underset{\sim}{Q}_{1,2}\right]_1\left[\underset{\sim}{Q}_{2,3}\right]_2 \cdots \left[\underset{\sim}{Q}_{6,7}\right]_6\right)$$

$$- 2\ \underset{\sim}{e}\ \text{sen}\ \phi = \underset{\sim}{0} \qquad (5b)$$

$$\underset{\sim}{f}_2(\underset{\sim}{\theta}) = \left[\underset{\sim}{a}_{1,2}\right]_1 + \left[\underset{\sim}{a}_{2,3}\right]_1 + \cdots + \left[\underset{\sim}{a}_{6,7}\right]_1 - \left[\underset{\sim}{r}\right]_1 = \underset{\sim}{0} \qquad (5c)$$

La ecuación (5b) aparece multiplicada por 2 con el fin de evitar divisiones posteriores entre 2 que pueden incrementar el número de operaciones necesarias.

Se puede obtener la solución de (5a) aplicando el método de Newton-Raphson que, con un valor inicial próximo a una solución, converge cuadráticamente, requiriendo así poco tiempo para obtener esa solución [10].

El esquema iterativo de Newton-Raphson es el siguiente:

$$\underset{\sim}{\theta}^{k+1} = \underset{\sim}{\theta}^{k} + \Delta\underset{\sim}{\theta}^{k} \qquad (6a)$$

donde $\Delta\underset{\sim}{\theta}^{k}$, es la solución de

$$\underset{\sim}{J}(\underset{\sim}{\theta}^{k})\Delta\underset{\sim}{\theta}^{k} = -\ \underset{\sim}{f}(\underset{\sim}{\theta}^{k})$$

y $\underset{\sim}{J}(\underset{\sim}{\theta}^{k})$ es la matriz Jacobiana, evaluada en $\underset{\sim}{\theta} = \underset{\sim}{\theta}^{k}$, del sistema (5a), que se calcula como:

$$\underset{\sim}{J}(\underset{\sim}{\theta}) = \begin{bmatrix} \partial\underset{\sim}{f}_1(\underset{\sim}{\theta})/\partial\underset{\sim}{\theta} \\ \partial\underset{\sim}{f}_2(\underset{\sim}{\theta})/\partial\underset{\sim}{\theta} \end{bmatrix} \qquad (7)$$

donde

$$\frac{\partial\underset{\sim}{f}_1(\underset{\sim}{\theta})}{\partial\theta_i} = 2\ \text{vect}\ (\left[\underset{\sim}{Q}_{1,2}\right]_1\left[\underset{\sim}{Q}_{2,3}\right]_2 \cdots \left[\frac{\partial\underset{\sim}{Q}_{i,i+1}}{\partial\theta_i}\right]_i$$

$$\left[\underset{\sim}{Q}_{i+1,i+2}\right]_{i+1} \cdots \left[\underset{\sim}{Q}_{6,7}\right]_6) \qquad (8a)$$

Para el cálculo de $\partial\underset{\sim}{f}_2/\partial\underset{\sim}{\theta}$, defínase

$$x_1(\underset{\sim}{\theta}) = \left[\underset{\sim}{a}_{1,2}\right]_1 + \left[\underset{\sim}{Q}_{1,2}\right]_1\left[\underset{\sim}{a}_{2,3}\right]_2 + \cdots + \left[\underset{\sim}{Q}_{1,2}\right]_1\left[\underset{\sim}{Q}_{2,3}\right]_2 \cdots$$

$$\cdots\left[\underset{\sim}{Q}_{5,6}\right]_5\left[\underset{\sim}{a}_{6,7}\right]_6 \qquad (8b)$$

que puede calcularse mediante el algoritmo de Horner para evaluación de polinomios [10] como:

$$\underset{\sim}{x}_6 = \left[\underset{\sim}{a}_{6,7}\right]_6$$

$$\underset{\sim}{x}_k = \left[\underset{\sim}{a}_{k,k+1}\right]_k + \left[\underset{\sim}{Q}_{k,k+1}\right]_k \underset{\sim}{x}_{k+1} \qquad k = 5, 4, \ldots, 1 \qquad (8c)$$

Así

$$\frac{\partial\underset{\sim}{f}_2(\underset{\sim}{\theta})}{\partial\theta_i} = \frac{\partial\underset{\sim}{x}_1}{\partial\theta_i} \qquad (8d)$$

La ecuación (8a) puede simplificarse [7] en la forma

$$\frac{\partial\underset{\sim}{f}_1(\underset{\sim}{\theta})}{\partial\theta_i} = (\underset{\sim}{1}\ \text{tr}\ \underset{\sim}{P} - \underset{\sim}{P})\left[\underset{\sim}{Q}_{1,2}\right]_1 \cdots \left[\underset{\sim}{Q}_{i-1,i}\right]_{i-1}\underset{\sim}{e}_i \qquad (9)$$

siendo $\underset{\sim}{e}_i$ el vector unitario paralelo al eje de rotación del i° par cinemático de rotación, y

$$\underset{\sim}{P} = \left[\underset{\sim}{Q}_{1,2}\right]_1\left[\underset{\sim}{Q}_{2,3}\right]_2 \cdots \left[\underset{\sim}{Q}_{6,7}\right]_6 \qquad (10)$$

La velocidad se calcula a partir de la siguiente relación:

$$\text{vect}\ (\dot{\underset{\sim}{P}}\ \underset{\sim}{P}^T) = \underset{\sim}{\omega} \qquad (11)$$

donde $\underset{\sim}{\omega}$ es la velocidad angular especificada del OT, referida al sistema 1, en tanto que

$$\frac{\partial\underset{\sim}{x}_1(\underset{\sim}{\theta})}{\partial\underset{\sim}{\theta}}\dot{\underset{\sim}{\theta}} = \underset{\sim}{v} \qquad (12)$$

siendo $\underset{\sim}{v}$ la velocidad del punto p del OT referida al sistema 1. Las ecuaciones (11) y (12), después de algunas simplificaciones [7], quedan como:

$$\underset{\sim}{J}(\underset{\sim}{\theta})\ \dot{\underset{\sim}{\theta}} = \underset{\sim}{d} \qquad (13a)$$

$$\text{donde}\quad \underset{\sim}{d} = \begin{bmatrix} \underset{\sim}{\omega}' \\ \underset{\sim}{v} \end{bmatrix}\ \text{y}\ \underset{\sim}{\omega}' = (\underset{\sim}{1}\ \text{tr}\ \underset{\sim}{P} - \underset{\sim}{P})\ \underset{\sim}{\omega} \qquad (13b)$$

La aceleración se calcula a partir de las ecuaciones (11) y (12) como [7]:

$$\underset{\sim}{J}(\underset{\sim}{\theta})\ \ddot{\underset{\sim}{\theta}} = \underset{\sim}{g}$$

donde

$$\underset{\sim}{g} = \begin{bmatrix} \dot{\underset{\sim}{\omega}}' \\ \underset{\sim}{a}' \end{bmatrix} = \begin{bmatrix} (\underset{\sim}{1}\ \text{tr}\ \underset{\sim}{P} - \underset{\sim}{P})(\dot{\underset{\sim}{\omega}} - \frac{1}{2}\dot{\underset{\sim}{\theta}}^T \frac{\partial^2\underset{\sim}{\omega}}{\partial\underset{\sim}{\theta}^2}\dot{\underset{\sim}{\theta}}) \\ \underset{\sim}{a} - \frac{1}{2}\dot{\underset{\sim}{\theta}}^T \frac{\partial^2\underset{\sim}{v}}{\partial\dot{\underset{\sim}{\theta}}^2}\dot{\underset{\sim}{\theta}} \end{bmatrix} \qquad (14b)$$

en las cuales $\dot{\underset{\sim}{\omega}}$ es la aceleración angular del OT y $\underset{\sim}{a}$ es la aceleración del punto p del OT, ambas prescritas y referidas al sistema coordenado 1.

De acuerdo con el algoritmo expuesto, los subprogramas realizan la solución numérica de las ecuaciones (5a), (13a) y (14a), y producen los valores de $\underset{\sim}{\theta}$, $\dot{\underset{\sim}{\theta}}$ y $\ddot{\underset{\sim}{\theta}}$ correspondientes a los prescritos de $\underset{\sim}{Q}$ (o equivalentemente, de $\underset{\sim}{e}$ y ϕ), $\underset{\sim}{r}$, $\underset{\sim}{v}$, $\underset{\sim}{\omega}$, $\underset{\sim}{a}$ y $\dot{\underset{\sim}{\omega}}$. A continuación se describen las variables que intervienen en el cálculo, así como las subrutinas y sus argumentos.

Según la Fig 1, y la notación de Denavit y Hartenberg, se tiene, con $c(\ ) \equiv \cos(\ )$ y $s(\ ) \equiv \text{sen}(\ )$:

$$\left[a_{i,i+1}\right]_i = \left[a_i c\theta_i,\ a_i s\theta_i,\ b_i\right]^T \qquad (15)$$

donde a_i es la distancia entre los ejes Z_i y Z_{i+1}, luego siempre positiva, y b_i es la coordenada de la intersección de x_{i+1} con Z_i en el sistema X_i, Y_i, Z_i. Así, la matriz $\left[Q_{i,i+1}\right]_i$ se puede expresar como

$$\left[Q_{i,i+1}\right]_i = \begin{bmatrix} c\theta_i & -s\theta_i\, c\alpha_i & s\theta_i\, s\alpha_i \\ s\theta_i & c\theta_i\, c\alpha_i & -c\theta_i\, s\alpha_i \\ 0 & s\alpha_i & c\alpha_i \end{bmatrix} \qquad (16)$$

donde α_i es el ángulo entre Z_i y Z_{i+1}, medido en la dirección positiva de X_{i+1}.

Con lo anterior se pueden definir los arreglos THETA y P. Los primeros 32 elementos de P son datos y deben almacenarse como se indica a continuación:

I=1,...,6

THETA(I): contiene los ángulos asociados a cada par de la cadena cinemática, ecs (15) y (16)

P(I): almacena a_i de la ec (15)

P(I+6): almacena b_i de la ec (15)

P(I+12): almacena α_i y posteriormente, cos α_i de la ec (15)

P(I+18): almacena sen α_i de la ec (15)

P(25): almacena ϕ de la ec (3)

J=1,2,3

P(J+25): almacena $\underline{e}$ de la ec (3)

P(J+28): almacena las coordenadas del OT referidas al sistema 1 (lado derecho de la ec (2))

P(32): almacena 2π

De P(33) hasta P(122) se definen en la tabla 1.

En la Fig 2 se muestra el diagrama jerárquico del programa.

Las subrutinas que se emplean son:

NRDAMP(THETA,F,DF,IP,P,TOLX,TOLF,DAMP,N,ITER, MAX,KMAX)

Obtiene las raíces de un sistema algebraico no lineal de orden N, aplicando el método de Newton-Raphson con amortiguamiento (0<DAMP≤1) [8]. El amortiguamiento tiene por objeto acelerar la convergencia. En esta subrutina, F representa el miembro izquierdo de la ec (5a), que es de dimensión 6 y se calcula en FUN; DF es la matriz Jacobiana de la ec (7) y de dimensión 6x6 y se calcula en DFDX. La solución de (6b) se obtiene por medio de las subrutinas DECOMP y SOLVE que se describen posteriormente. TOLX es la tolerancia impuesta en la aproximación a la solución, en tanto que TOLF es la tolerancia que se acepta en la función F; MAX es el número máximo de iteraciones permitidas, ITER es el número de la iteración ejecutada y KMAX, el número máximo de amortiguamientos que se permiten por iteración; IP es un arreglo entero de dimensión 6, se obtiene en DECOMP y es el índice pivotal en la descomposición LU.

FUN(THETA,F,P,N)

Forma las ecuaciones de desplazamiento y de rotación que, al anularse, proporcionan el valor de los ángulos θ_i para una posición dada, ec (5a). Para este fin se requieren las subrutinas PROD y VECX. Estas requieren a su vez el arreglo TCS, definido como:

I=1,...,6

TCS(I) = THETA(I)

TCS(I+6) = COS(THETA(I))

TCS(I+12) = SIN(THETA (I))

con lo cual se conservan los valores de THETA, en cada iteración.

PROD(TCS,P)

Efectúa los productos:

$$\left[Q_{1,2}\right]_1$$

$$\left[Q_{1,2}\right]_1 \left[Q_{2,3}\right]_2$$

$$\left[Q_{1,2}\right]_1 \left[Q_{2,3}\right]_2 \left[Q_{3,4}\right]_3$$

$$\vdots$$

$$\left[Q_{1,2}\right]_1 \left[Q_{2,3}\right]_2 \cdots \left[Q_{6,7}\right]_6$$

y los almacena de acuerdo al arreglo mostrado en la Tabla 1. Para lograrlo evalúa (16) con los valores de θ_i y α_i para $i\,(i=2,\ldots,6)$ y ejecuta el producto matricial con el último arreglo almacenado. El resultado es guardado en memoria con la ubicación mostrada en la tabla 1.

VECX(TCS,P)

Calcula el vector $\underline{x}_1$, ecs (6b y 6c), así como su derivada con respecto a θ, ec (8d) y los almacena en el arreglo P, como se muestra en la Tabla 1. Esto se debe a que existen conjuntos de operaciones comunes para el cálculo de $\underline{x}_1$ y su derivada.

DFDX(THETA,DF,P,N)

Calcula la matriz Jacobiana definida en la ec (7) y la almacena en DF(6,6).

DECOMP(N,N,DF,COND,IP,WORK)

Descompone una matriz real (DF) en el producto LU y estima su condición (COND) [11], con lo que se conoce la amplificación del error de redondeo. Para efectuar esta estimación se emplea el arreglo WORK. IP es el vector pivote, que almacena información sobre los intercambios de renglones necesarios para evitar un excesivo error de redondeo.

SOLVE(N,N,DF,F,IP)

Resuelve el sistema $J\delta=-f$ empleando la matriz DF factorizada en DECOMP por sustitución regresiva, considerando los intercambios de renglones efectuados en DECOMP.

VELAC(N,DF,IP,P,V,A)

Calcula $\dot{\underline{\theta}}$ y $\ddot{\underline{\theta}}$ cuando se tiene convergencia en NRDAMP, retorna al programa principal y llega a ésta con la matriz Jacobiana descompuesta en el producto LU. En caso de que casualmente, o a propósito, se de a NRDAMP valores solución a las ecs (5a), esta subrutina no utiliza DECOMP, por lo que no se tiene la matriz Jacobiana descompuesta; en este caso, llama a DFDX y a DECOMP para entrar a VELAC. A esta subrutina se le suministran los valores de $\underline{\omega}$ y $\underline{v}$ en el arreglo V, $\dot{\underline{\omega}}$ y $\underline{a}$ en el arreglo A y regresa los resultados de $\dot{\underline{\theta}}$ y $\ddot{\underline{\theta}}$ en V y A respectivamente, o sea, obtiene la solución de las ecuaciones (13a) y (14a).

Ejemplo

Los subprogramas se aplicaron para obtener el análisis de una malla cerrada de 7 eslabones que se muestra en la Fig 3, cuyos parámetros son:

a_i=1 (u de long)	i=1,...,6
a_7=3 (u de long)	
b_i=0,	i=1,...,7
$\alpha_1=\alpha_3=\alpha_4=\alpha_6=90°$	$\alpha_2=\alpha_5=\alpha_7=0°$

Los resultados se muestran en la Fig 4 en la que se graficó sólo θ_2 vs θ_1, ya que por las simetrías de este mecanismo, $\theta_2=\theta_5=-\theta_3=-\theta_6$ y $\theta_4=2\theta_1$ [7]. El error que se obtuvo al comparar θ_2 con la solución analítica obtenida en [7] se muestra en la Fig 5; en las Figs 6 y 7 se muestran los errores de $\dot{\theta}_2$ y $\ddot{\theta}_2$, respectivamente, comparados con la solución analítica.

Conclusiones y extensiones

Los resultados hasta ahora obtenidos demuestran la potencialidad del algoritmo. En la solución del ejemplo se requirieron aproximadamente 17 segundos, para la rama superior o sea, para $120°\leq\theta_1\leq240°$ y $0°\leq\theta_2\leq120°$. Este tiempo puede disminuirse al emplear un lenguaje de máquina.

La subrutina VELAC no calcula $\dot{\underline{\theta}}$ ni $\ddot{\underline{\theta}}$ cuando la matriz Jacobiana es singular, lo cual sucede cuando se tienen configuraciones extremas de las cadenas cinemáticas, por lo que se continúa la investigación en este sentido.

Referencias

1. Duffy J. and Derby S., "Displacement analysis of a spatial 7R mechanism - a generalized lobster's arm", Journal of Mechanical Design Trans. ASME, Vol 101, april 1979, pp 224-231.

2. Albala H. and Angeles J. "Numerical solution to the input-output displacement equation of the general 7R spatial mechanism", Proc. V World Congress on Theory of Machines and Mechanisms", Montreal, Canada, july 8-12, 1979, pp 1008-1011.

3. Donati M, Morasso P. and Tagliasco V., "The inverse kinematic problem for antropomorphic manipulator arms" Journal of Dynamic Systems Measurement and Control. Trans, ASME, Vol. 104, march 1982, pp 110-113.

4. Duffy J and Crane C. "A displacement analysis of the general spatial 7-link, 7R mechanism" Mechanism and Machine Theory, Vol. 15, 1980. pp 153-169.

5. Albala H. "Displacement analysis of the general n-bar, single-loop, spatial linkage" Journal of Mechanical Design, Trans. ASME Vol. 104, april 1982, pp 501-525.

6. Angeles J. y Rojas A.A. "Programa para analizar digitalmente el desplazamiento del mecanismo espacial 7R general", Memoria del IV Congreso de la Academia Nacional de Ingeniería, A.C., Mérida, Yuc. 1978 pp 88-91.

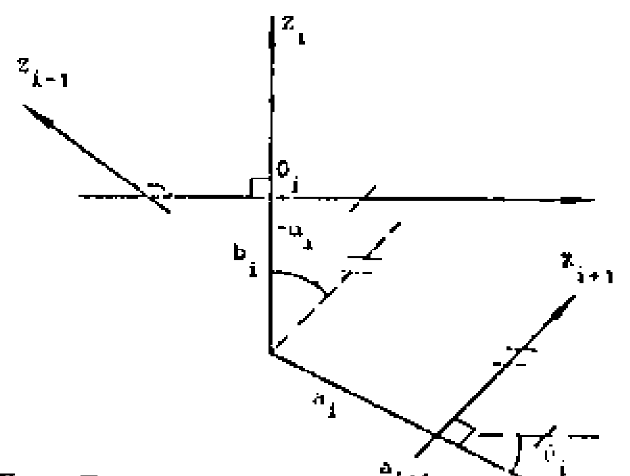
7. Angeles J. Analysis of General-Seven-Link Revolute Coupled Kinematic Chain, Informe Interno DEPFI, UNAM, 1984.

8. Angeles J. Spatial Kinematic Chains. Analysis, Synthesis and Optimization, Springer-Verlag, Berlín 1982.

9. Wrede R.C, Introduction to Vector and Tensor Analysis, Dover Publications, Inc. New York, 1972, pp. 158-159.

10. Ralston A. Introducción al Análisis Numérico, Limusa, México 1978, pp 349-362, pp 309.

11. Forsythe G.E., Malcom M.A. y Moler C.B. Computer Methods for Mathematical Computations, Prentice Hall Inc, Englewood Cliffs N.J. 1977, cap. 3.



$\left[\underline{a}_{i,i+1}\right]_i$ =vector que va de O_i a O_{i+1}

Fig 1 Parámetros que definen la arquitectura de una cadena cinemática

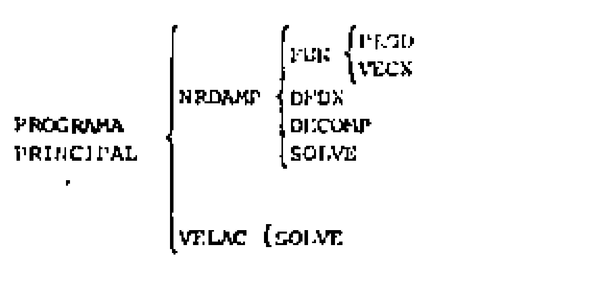


Fig 2 Diagrama jerárquico del programa

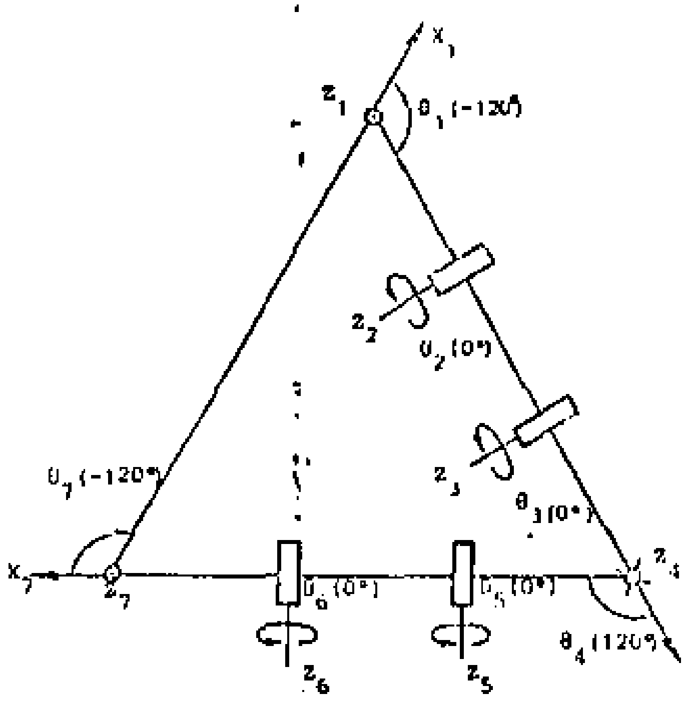


Fig 3 Cadena cinemática cerrada 7R

• Eje que apunta hacia afuera del plano de la figura.

X Eje que apunta hacia adentro del plano de la figura

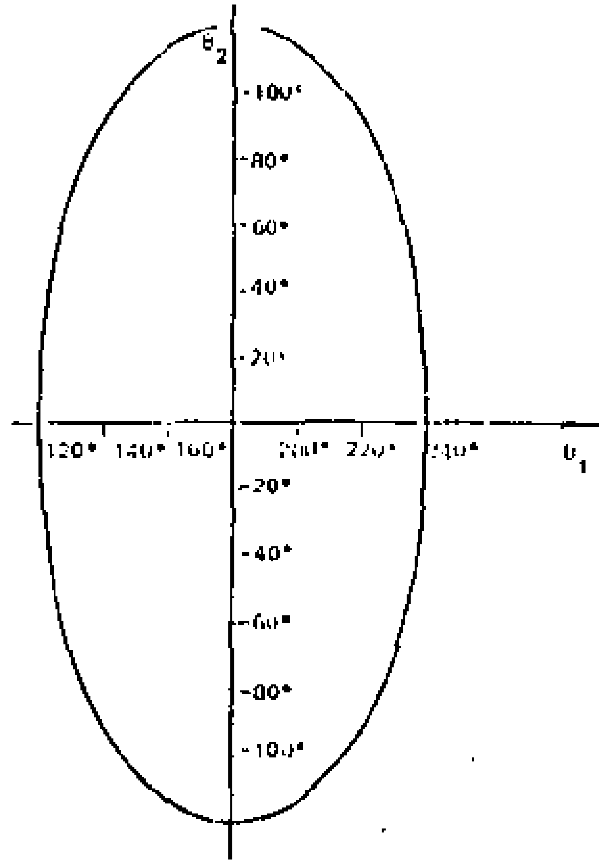


Fig 4 Respuesta cinemática del mecanismo 7R de la Fig 3.

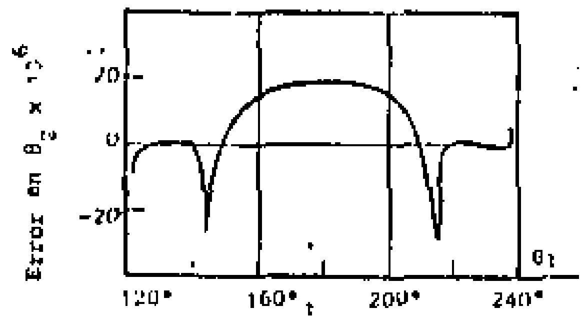


Fig 5 Error en el cálculo de θ_2 con $\theta_2 \geq 0$

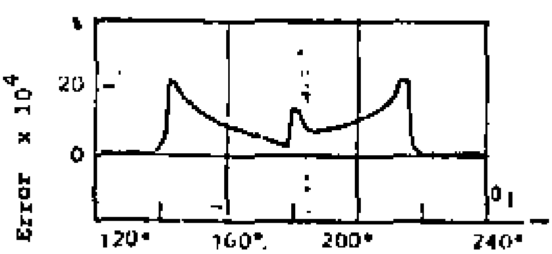


Fig 6 Error en el cálculo de $\theta_2'(\theta_1)$, $\theta_2 \geq 0$

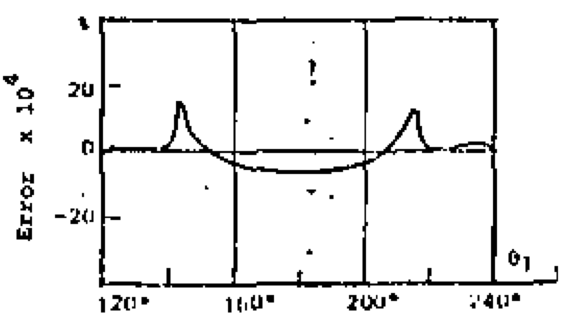


Fig. 7 Error en el cálculo de $\theta_2''(\theta_2)$, $\theta_2 \geq 0$

Tabla 1*

P_{33}	P_{36}	P_{39}	P_{42}	P_{45}	P_{48}	P_{51}	P_{54}	P_{57}
P_{34}	P_{37}	P_{40}	P_{43}	P_{46}	P_{49}	P_{52}	P_{55}	P_{58}
P_{35}	P_{38}	P_{41}	P_{44}	P_{47}	P_{50}	P_{53}	P_{56}	P_{59}
Q_1			Q_1Q_2			$Q_1Q_2Q_3$		

P_{60}	P_{63}	P_{66}	P_{69}	P_{72}	P_{75}	P_{78}	P_{81}	P_{84}
P_{61}	P_{64}	P_{67}	P_{70}	P_{73}	P_{76}	P_{79}	P_{82}	P_{85}
P_{62}	P_{65}	P_{68}	P_{71}	P_{74}	P_{77}	P_{80}	P_{83}	P_{86}
$Q_1Q_2Q_3Q_4$			$Q_1Q_2Q_3Q_4Q_5$			$Q_1Q_2Q_3Q_4Q_5Q_6$		

P_{87}	P_{90}	P_{93}	P_{96}	P_{99}	P_{102}
P_{88}	P_{91}	P_{94}	P_{97}	P_{100}	P_{103}
P_{89}	P_{92}	P_{95}	P_{98}	P_{101}	P_{104}
X_1	X_2	X_3	X_4	X_5	X_6

P_{105}	P_{108}	P_{111}	P_{114}	P_{117}	P_{120}
P_{106}	P_{109}	P_{112}	P_{115}	P_{118}	P_{121}
P_{107}	P_{110}	P_{113}	P_{116}	P_{119}	P_{122}
$\frac{\partial f_2}{\partial \theta_1}$	$\frac{\partial f_2}{\partial \theta_2}$	$\frac{\partial f_2}{\partial \theta_3}$	$\frac{\partial f_2}{\partial \theta_4}$	$\frac{\partial f_2}{\partial \theta_5}$	$\frac{\partial f_2}{\partial \theta_6}$

*$[Q_i, Q_{i+1}]_i$ se indica como Q_i

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISENO CINEMATICO DE MAQUINARIA

MODELIZACION DE CADENAS CINEMATICAS CON PARES DE ROTACION Y PRISMATICOS

A.A. ROJAS S.
DR. JORGE ANGELES ALVAREZ

JUNIO, 1984.

MODELIZACION DE CADENAS CINEMATICAS CON PARES DE ROTACION Y PRISMATICOS

A.A. ROJAS S.

J. ANGELES A.

Facultad de Ingeniería, UNAM.

México, D.F.

Abstract

A model of kinematic chains with prismatic and revolute pairs is presented. The model is based upon an algorithm using invariance concepts and solves the arising kinematic equations via the method of Newton-Raphson. The aplication is illustrated with a fully solved example, namely the analysis of a spatial RSRC linkage

Resumen

Se presenta un modelo de cadenas cinemáticas con pares de rotación y prismáticos. El modelo está basado en un algoritmo que emplea conceptos de invariancia y resuelve las ecuaciones cinemáticas resultantes mediante el método de Newton-Raphson. La aplicación se ilustra con un ejemplo que consiste en el análisis de un mecanismo espacial RSRC.

Introducción

El problema cinemático inverso consiste en determinar los ángulos o los desplazamientos en una cadena cinemática compuesta por n eslabones acoplados mediante pares inferiores, suponiendo conocidas las historias de posición, velocidad y aceleración de uno de los eslabones. Esta cadena puede ser abierta (manipuladores) o cerrada (mecanismos). Dada la importancia que tienen actualmente los manipuladores industriales y la complejidad del problema inverso, numerosos investigadores han propuesto diferentes soluciones, principalmente en forma cerrada, para casos particulares simples, empleando matrices de 4x4 que contienen rotación y desplazamiento simultáneamente [1,2,3], o bien reduciendo las ecuaciones cinemáticas involucradas a un polinomio en una incógnita. Este aparece expresado como un determinante de una matriz de 12x12 con elementos de 4º grado en la tangente de la mitad del ángulo de entrada [4], o bien de una de 16x16 con elementos de 2º grado en el mismo argumento [5,6]. Alizade y Duffy [7] proponen la relación entre los datos y las incógnitas a través de un conjunto de 30 ecuaciones, algunas de ellas dependientes de la topología, las cuales no son de fácil aplicación para el cálculo de velocidad y aceleración. Whitney [8] orienta su trabajo hacia la solución del problema inverso aplicando cambios diferenciales en la posición, con lo que se obtiene el jacobiano que relaciona los cambios en posición y orientación dados, como variables dependientes, con las variables de los pares considerados independientes, calculándose fácilmente sólo la velocidad. Un método para calcular la aceleración en cada par, conocidas la posición y la velocidad en cada uno de éstos, así como el momento o la fuerza de entrada, ha sido propuesto por Walker y Orin [9]. En este trabajo se presenta un modelo general aplicable en tiempo real a manipuladores con 7 eslabones con arquitectura arbitraria, conociendo las historias de posición, velocidad y aceleración del órgano terminal (OT) del manipulador. El modelo es una generalización del expuesto con anterioridad [10,11], el cual no tiene la flexibilidad de sustituir un par prismático por uno de rotación. El presente modelo es aplicable a cadenas con seis pares de rotación o bien a una combinación de cinco pares de rotación y un prismático, el cual puede estar ubicado en una posición arbitraria con respecto a los de rotación. Esta última topología se encuentra en una gran cantidad de manipuladores, tales como el brazo stanford (USA) [1], el VWR30 (RFA), el ACMA Cribier H-80 (Francia) y el Komatsu RCA70 (Japón) [12].

Descripción del algoritmo generalizado

De acuerdo con el método y la notación de Denavit y Hartenberg [13], que se resume en la Fig 1, los n eslabones de una cadena cinemática se numeran ordenadamente de 1 a n y al iº eslabón se fija el sistema coordenado X_i, Y_i, Z_i como sigue:

Z_i se dirige a lo largo del eje del par i , que une los eslabones i e i+1, si es de rotación, o alineado con la traslación del par i cuando éste es prismático.

X_i se define sobre la perpendicular común a Z_{i-1} y Z_i, dirigida de Z_{i-1} a Z_i.

Y_i completa el sistema coordenado dextrógiro del iº eslabón.

La posición relativa de dos eslabones adyacentes está completamente definida por los siguientes parámetros:

a_i, distancia entre los ejes Z_i y Z_{i+1}

α_i, ángulo entre Z_i y Z_{i+1}, medido en la dirección positiva de X_{i+1}.

b_i, coordenada entre los ejes X_i y X_{i+1}, constante cuando el par es de rotación.

θ_i, ángulo entre los ejes X_i e X_{i+1} medido en la dirección positiva de Z_i, constante cuando el par es prismático.

Sean

$$[Q_{i,i+1}]_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i \\ s\theta_i & c\theta_i c\alpha_i & c\theta_i s\alpha_i \\ 0 & s\alpha_i & c\alpha_i \end{bmatrix} \tag{1}$$

$$[a_{i,i+1}]_i = [a_i c\theta_i,\ a_i s\theta_i,\ b_i]^T \tag{2}$$

donde $c(\ ) \equiv \cos(\ )$ y $s(\ ) \equiv \mathrm{sen}(\ )$

$[Q_{i,i+1}]_i$ y $[a_{i,i+1}]_i$, abreviadas como Q_i y a_i, respectivamente, representan una matriz ortogonal referida a los ejes X_i, Y_i, Z_i, que gira éstos a una posición coincidente con los correspondientes X_{i+1}, Y_{i+1}, Z_{i+1}, y un vector que une los orígenes O_i y O_{i+1}, dirigido del primero al segundo y referido a los ejes del iº eslabón, respectivamente.

Las condiciones de cerradura en orientación y desplazamiento son:

$$Q_1 Q_2 \cdots Q_5 Q_6 = Q \tag{3}$$

$$[a_1]_1 + [a_2]_1 + \ldots + [a_6]_1 = r \tag{4}$$

que constituyen un sistema de 12 ecuaciones escalares en 6 incógnitas, siendo independiehtes sólo 6 de ellas. Las cantidades conocidas son Q y r, que representan el giro necesario para sobreponer los ejes X_1, Y_1, Z_1 con X_7, Y_7, Z_7 y el vector que une el origen del sistema inercial con un punto R del OT. Al obtener el vector axial [14] de Q se tiene:

$$\mathrm{vect}\ (Q) = e\ \mathrm{sen}\ \phi \tag{5}$$

donde e es el vector característico real de Q asociado con el valor característico +1 y ϕ, el ángulo de rotación. Al combinar (3) y (5) se tiene:

$$\mathrm{vect}\,(Q_1\, Q_2 \cdots Q_6) = e \operatorname{sen} \phi \tag{6}$$

Las ecuaciones (4) y (6) constituyen un sistema algebraico no lineal de sexto orden de la forma:

$$f(\theta) = \begin{bmatrix} f_r(\theta) \\ \cdot \\ f_t(\theta) \end{bmatrix} = 0 \tag{7}$$

donde $\theta = [\theta_1, \theta_2, \ldots, \theta_6]^T$ representa cinco valores diferentes asociados con pares de rotación y uno que puede estar asociado con un par prismático, o bien, con otro par de rotación; además:

$$f_r(\theta) = 2\,\mathrm{vect}([Q_{1,2}]_1\,[Q_{2,3}]_2 \cdots [Q_{6,7}]_6) - 2 \operatorname{sen} \phi = 0 \tag{8a}$$

$$f_t(\theta) = [a_{1,2}]_1 + [a_{2,3}]_1 + \cdots + [a_{6,7}]_1 - [r]_1 = 0 \tag{8b}$$

La solución de (7) se obtiene aplicando el método de Newton-Raphson, que converge cuadráticamente con valores próximos a la solución [15]. El esquema iterativo de Newton-Raphson es el siguiente:

$$\theta^{k+1} = \theta^k + \Delta\theta^k \tag{9a}$$

donde $\Delta\theta^k$ es la solución del sistema

$$J(\theta^k)\, \Delta\theta^k = - f(\theta^k) \tag{9b}$$

y $J(\theta^k)$ es la matriz jacobiana evaluada en $\theta = \theta^k$ del sistema (9b), que se calcula a partir de:

$$J(\theta) = \begin{bmatrix} \partial f_r(\theta)/\partial\theta \\ \cdot \\ \partial f_t(\theta)/\partial\theta \end{bmatrix} \tag{10}$$

Cuando θ_i está asociado a un par de rotación se tiene:

$$\frac{\partial f_r(\theta)}{\partial\theta_i} = (I \text{ tr } P - P)\, Q_1 Q_2 \ldots Q_{i-1}\, e_i \tag{11a}$$

siendo

$$P = [Q_{1,2}]_1\, [Q_{2,3}]_2 \ldots [Q_{6,7}]_6 \tag{11b}$$

e_i: vector unitario paralelo al eje z_i

I : matriz identidad

$$\frac{\partial f_t(\theta)}{\partial\theta_i} = \frac{\partial \dot{x}_1}{\partial\dot{\theta}_i} = \frac{\partial x_1}{\partial\theta_i} = Q_1 Q_2 \ldots Q_{i-1}(e_i \times x_i) \tag{11c}$$

x_i se calcula aplicando el algoritmo de Horner [15] para evaluación de polinomios, esto es, como:

$$\begin{aligned} x_6 &= [a_{6,7}]_6 \\ x_k &= [a_{k,k+1}]_k + [Q_{k,k+1}]_k\, x_{k+1} \qquad k = 5,4,3,2,1 \end{aligned} \tag{11d}$$

Cuando el par es prismático, b_i es una traslación (ver Fig 1) y se emplea:

$$\frac{\partial f_r(\theta)}{\partial\theta_i} = 0 \tag{12a}$$

$$\frac{\partial f_t(\theta)}{\partial \theta_i} = Q_1 Q_2 \ldots Q_{i-1}\, e_i \qquad (12b)$$

La velocidad se calcula a partir a las siguientes expresiones:

$$\frac{\partial x_1}{\partial \theta}\, \dot{\theta} = v \qquad (13a)$$

$$\mathrm{vect}(\dot{P}P^T) = \omega \qquad (13b)$$

siendo v la velocidad lineal del punto P del OT en el sistema 1 y ω, la velocidad angular vectorial del OT. Después de algunas manipulaciones, las ecuaciones (13) quedan como [11]:

$$J(\theta)\, \dot{\theta} = \dot{r} \qquad (14)$$

donde

$$\dot{r} = \begin{bmatrix} \omega' \\ v \end{bmatrix} \text{ y } \quad \omega' = (I\ \mathrm{tr}P - P)\, \omega \qquad (15)$$

válida sin modificaciones para pares de rotación o prismáticos.

La aceleración se obtiene a partir de:

$$\dot{\omega} = \frac{d}{dt}\, \mathrm{vect}\, (\dot{P}P^T) = \frac{\partial \omega}{\partial \dot{\theta}}\, \ddot{\theta} + \frac{1}{2}\, \dot{\theta}^T \frac{\partial^2 \dot{\omega}}{\partial \dot{\theta}}\, \dot{\theta} \qquad (16a)$$

$$\dot{v} = \frac{\partial x}{\partial \theta}\, \ddot{\theta} + \dot{\theta}^T\, \frac{\partial^2 x_1}{\partial \theta^2}\, \dot{\theta} \qquad (16b)$$

Se requieren ahora las siguientes expresiones cuando se tiene un par de rotación

$$\frac{\partial^2 \dot{\omega}}{\partial \theta_i \partial \theta_j} = [e_i]_1\ x[e_j]_1 \qquad i = 1, \ldots, 6 \quad j = i, \ldots, 6 \qquad (17a)$$

$$\frac{\partial^2 x_1}{\partial\theta_i \partial\theta_j} = [e_i]_1 \times \frac{\partial x_1}{\partial\theta_j} \tag{17b}$$

Cuando se tiene un par prismático en la iª posición se tiene

$$\frac{\partial^2 \dot{\omega}}{\partial\theta_i \partial\theta_j} = 0 \tag{18a}$$

$$\frac{\partial^2 x_1}{\partial\theta_i \partial\theta_j} = [e_i]_1 \times [e_j]_1 \qquad i<j \tag{18b}$$

Despues de algunas simplificaciones [11], se tiene:

$$J(\theta)\ \ddot{\theta} = \ddot{r} \tag{19}$$

donde

$$\ddot{r} = \begin{bmatrix} (I\ \mathrm{tr}\ P - P)\ [\dot{\omega} - \frac{1}{2}\dot{\theta}^T \frac{\partial^2 \dot{\omega}}{\partial\dot{\theta}^2}\dot{\theta}] \\ \dot{v} - \dot{\theta}^T \frac{\partial^2 x_1}{\partial\theta^2}\dot{\theta} \end{bmatrix} \tag{20}$$

El modelo cinemático aquí propuesto está formado por las ecs (8a y b), (14) y (19), que proporcionan los valores de las variables asociadas con los pares cinemáticos de la cadena, así como sus dos primeras derivadas. Las ecuaciones dichas se realizaron en un programa de computadora que es aplicable indistintamente a cadenas cinemáticas provistas de seis pares de rotación o bien de 5 pares de rotación y uno prismático.

Ejemplo

Se desea simular el mecanismo mostrado en la Fig 2, que es del tipo RSRC. El punto O_7 permanece fijo, cambiando de orientación a razón constante de 1 rad/s. En la Fig 3 se muestran los sistemas coordenados involucrados, así como la sustitución del par esférico por tres pares de rotación con los ejes concurrentes en el centro del esférico, en tanto que el par cilíndrico, por un par de rotación y otro prismático. Para este mecanismo particular se obtiene fácilmente una expresión que relaciona el giro ϕ con el desplazamiento s [16] mediante:

$$s(t) = a \operatorname{sen} \phi + \sqrt{b^2 - c^2 - a^2 \cos^2 \phi} \tag{21}$$

no siendo así para las demás variables

El error obtenido al comparar el valor calculado de s con el dado por la ec (21), se muestra en la Fig 4. Se emplearon los siguientes valores que aparecen en la Fig 2: $a = 1.0$, $b = 3.0$ y $c = 2.0$ en unidades de longitud.

Conclusiones

Los resultados se obtuvieron en una microcomputadora Apple IIe, siendo 580 el número total de operaciones requeridas para el cálculo de θ, $\dot{\theta}$ y $\ddot{\theta}$ por posición, en base a una iteración en el método de Newton-Raphson, cuando todos los pares son de rotación, reduciéndose este número a 556 al introducir un par prismático. El algoritmo presentado se puede aplicar por lo tanto al control de manipuladores como el brazo stanford [1] o al análisis de mecanismos de las más variadas topologías, como se mostró con el ejemplo.

'jom.

Bibliografía

1. R.P. Paul, Robot Manipulators: Mathematics, Programming and Control, MIT Press, Cambridge, Massachusetts, 1981. pp. 56-63.

2. R.P. Paul, B. Shimano, G.E. Mayer, "Kinematic control equations for simple manipulators", IEEE Trans on Systems, Man and Cybernetics, Vol. SMC-11 (6), junio 1981, pp 449-460.

3. J.Y.S. Luch, M.W. Walher, R.P. Paul, "Resolved-acceleration control of mechanical manipulators", IEEE Trans on Automatic Control, Vol. AC-25 (3) junio 1980, pp 468-494

4. J. Angeles y A.A. Rojas, "Programa para analizar digitalmente el desplazamiento del mecanismo espacial 7R general", Memoria del IV Congreso de la Academia Nacional de Ingeniería, A.C., Mérida Yuc. 1978 pp 88-91

5. J. Duffy y C. Crane, "A displacement analysis of the general spatial 7-link, 7R mechanism", Mechanism and Machine Theory, Vol. 15, 1980 pp. 153-159.

6. H. Albala, "Displacement analysis of the general n-bar, single loop, spatial linkage", Journal of Mechanical Design, Trans. ASME Vol. 104, April 1982, pp. 504-525.

7. R.I. Alizade y J. Duffy, "Mathematical models for analysis and synthesis of spatial mechanisms-IV", Mechanism and Machine Theory, Vol. 18, No. 5, 1983, pp 323-328

8. D.E. Whitney, "The mathematics of coordinated control of prosthetic arms and manipulators", Journal of Dynamic systems Measurement and Control, diciembre 1972, pp. 303-309.

9. M.W. Walker y D.E. Onin, "Efficient dynamic computer simulation of robotic mechanisms", Journal of Dynamic systems, Measurement and Control, septiembre 1982, Vol. 104, pp. 205-211.

10. J. Angeles y A.A. Rojas, "Subprogramas para el análisis de cadenas cinemáticas con 7 eslabones acoplados mediante pares de rotación", Memoria del IX Congreso de la Academia Nacional de Ingeniería, A.C., León, Gto. 1983 pp. 102-106.

11. J. Angeles y A.A. Rojas, "Software for the analysis of general seven link, revolute-coupled kinematic chains", Informe interno, DEPFI-UNAM. 1983.

12. "Industrial Robots of the world", Industrial Robot, marzo 1983, pp. 64-67

13. J. Denavit y R.S. Hartenberg, "A kinematic notation for lower pair mechanisms based on 4x4 matrices" , J. Appl. Mech. pp. 215-221, junio 1955.

14. D.C. Leigh, Non-linear Continuum Mechanics, McGraw-Hill, New York, 1968 pp. 42-43.

15. N.S. Bakhvalov, Numerical Methods, Mir, Moscu 1977 pp. 426-431 y 248.

16. J. Angeles, Spatial Kinematic Chains. Analysis, Synthesis and Optimization, Springer Verlag, Berlin 1982.

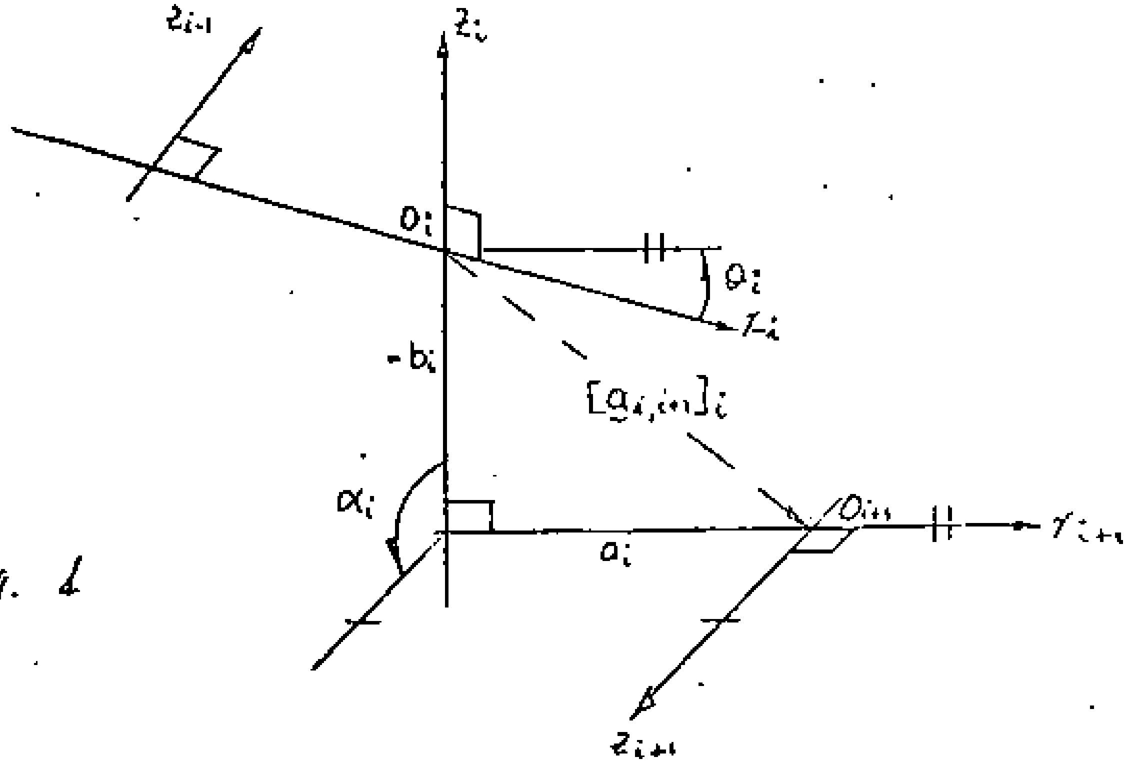


Fig. 1

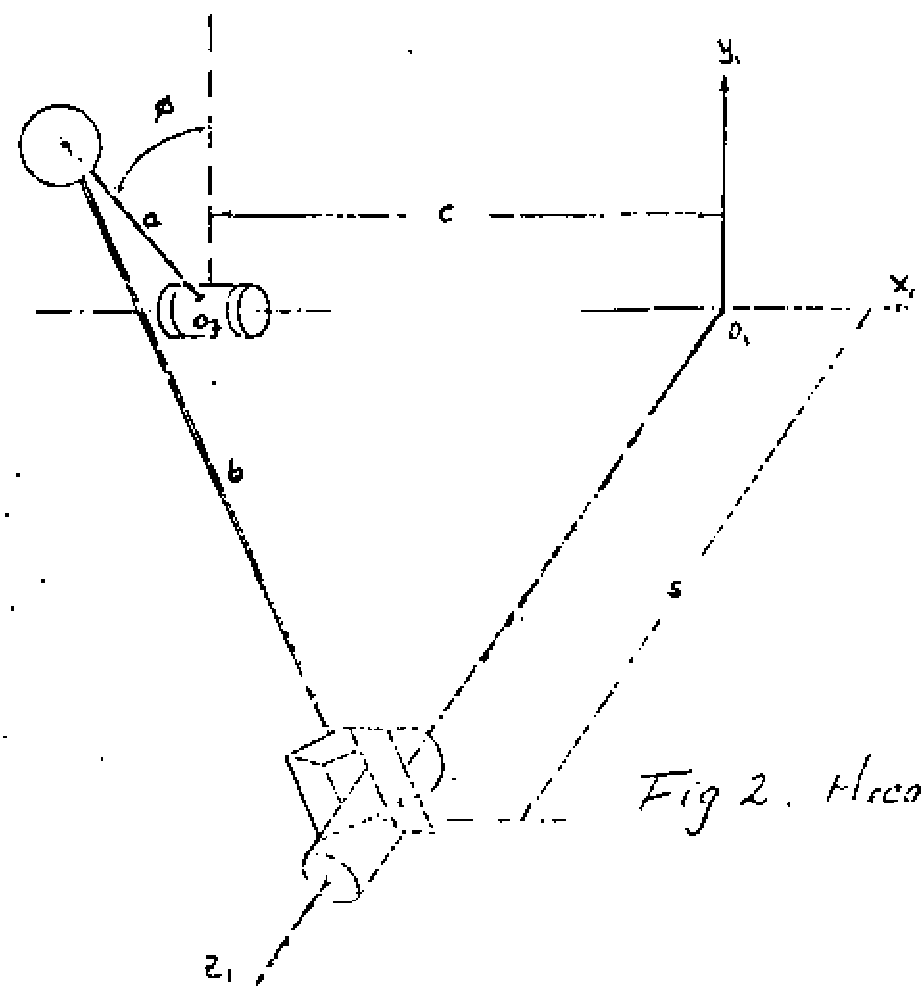


Fig 2. Mecanismo RSRC

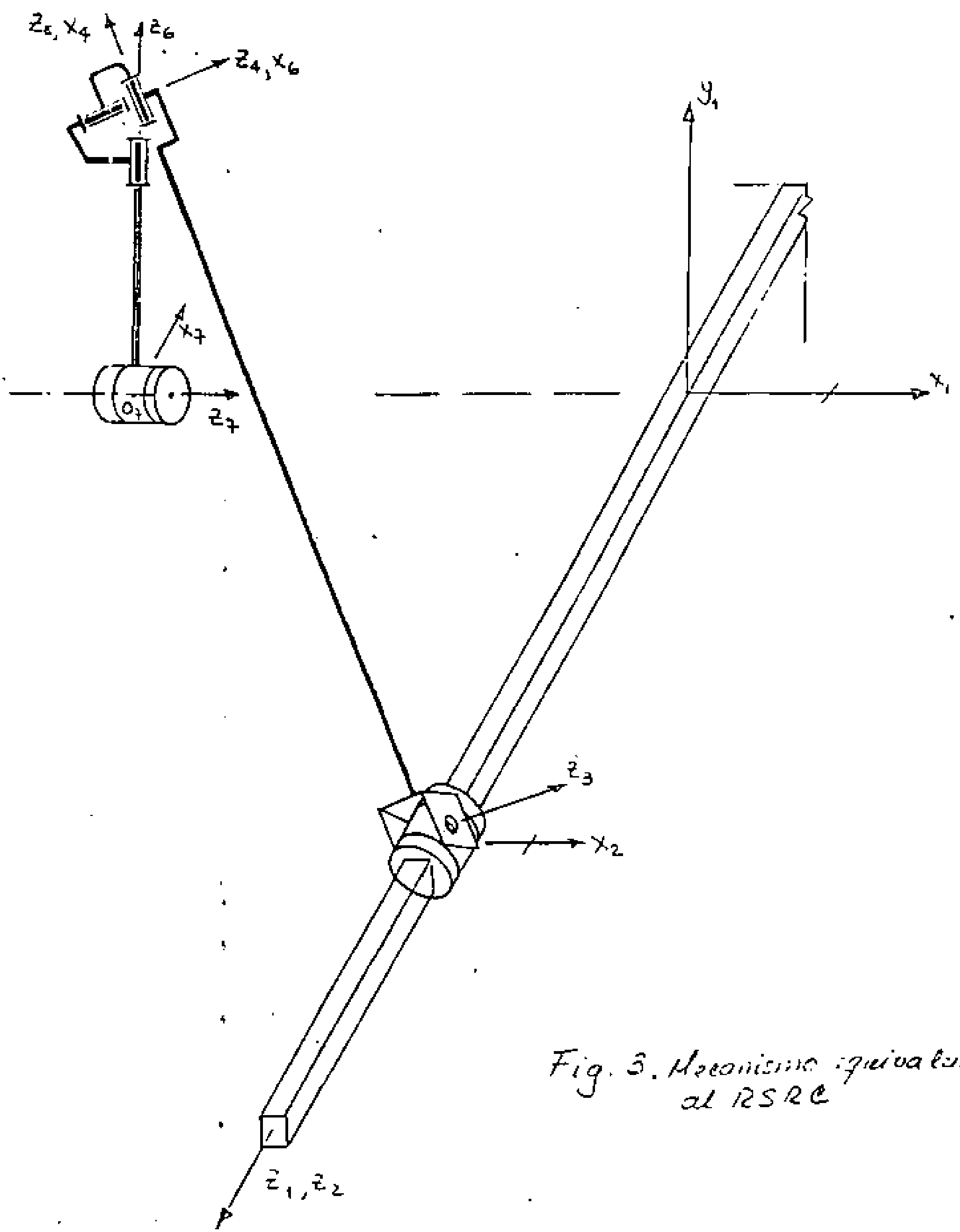


Fig. 3. Mecanismo equivalente al RSRC

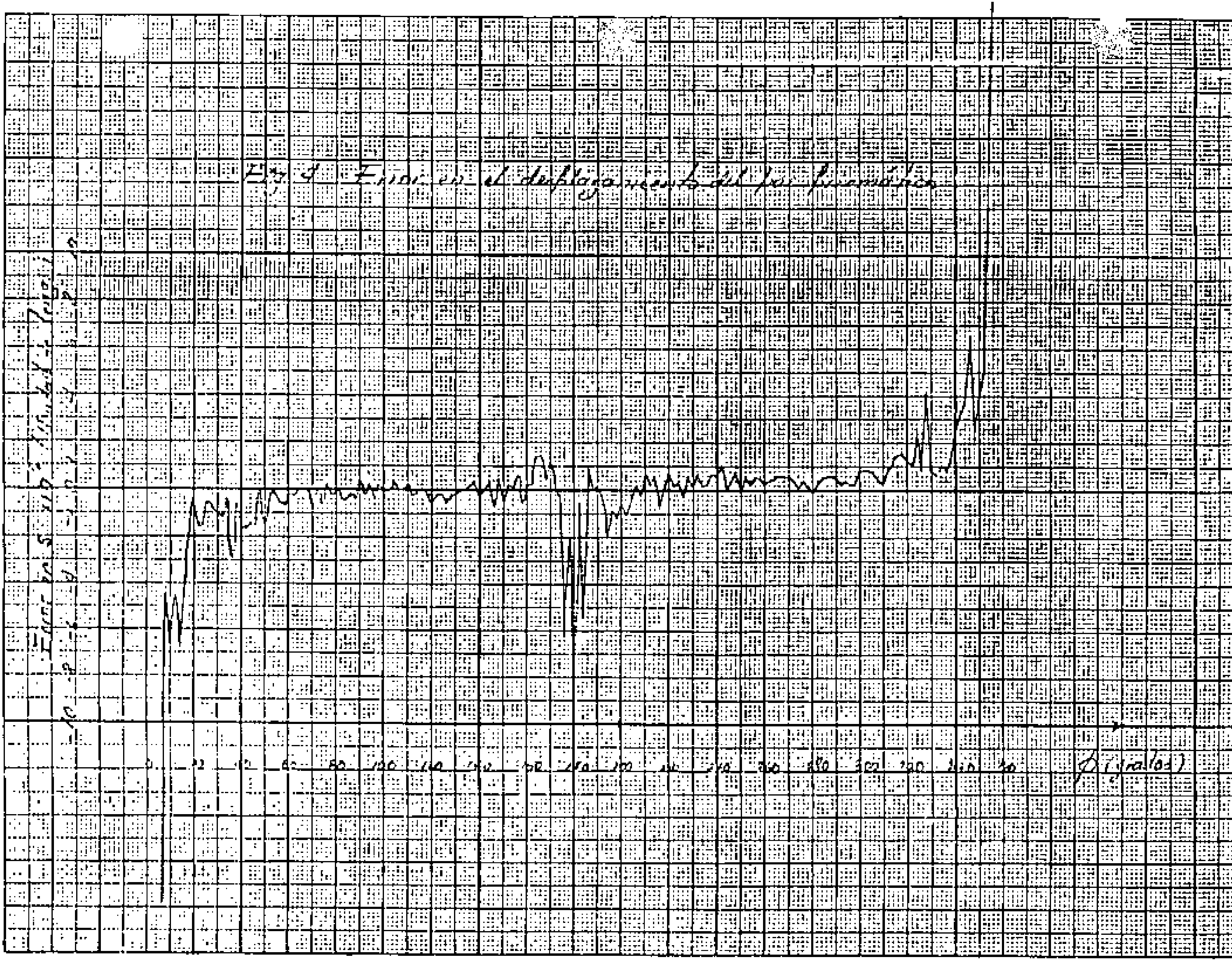


Fig. 4 Error en el desplazamiento del par cinemático
φ (grados)

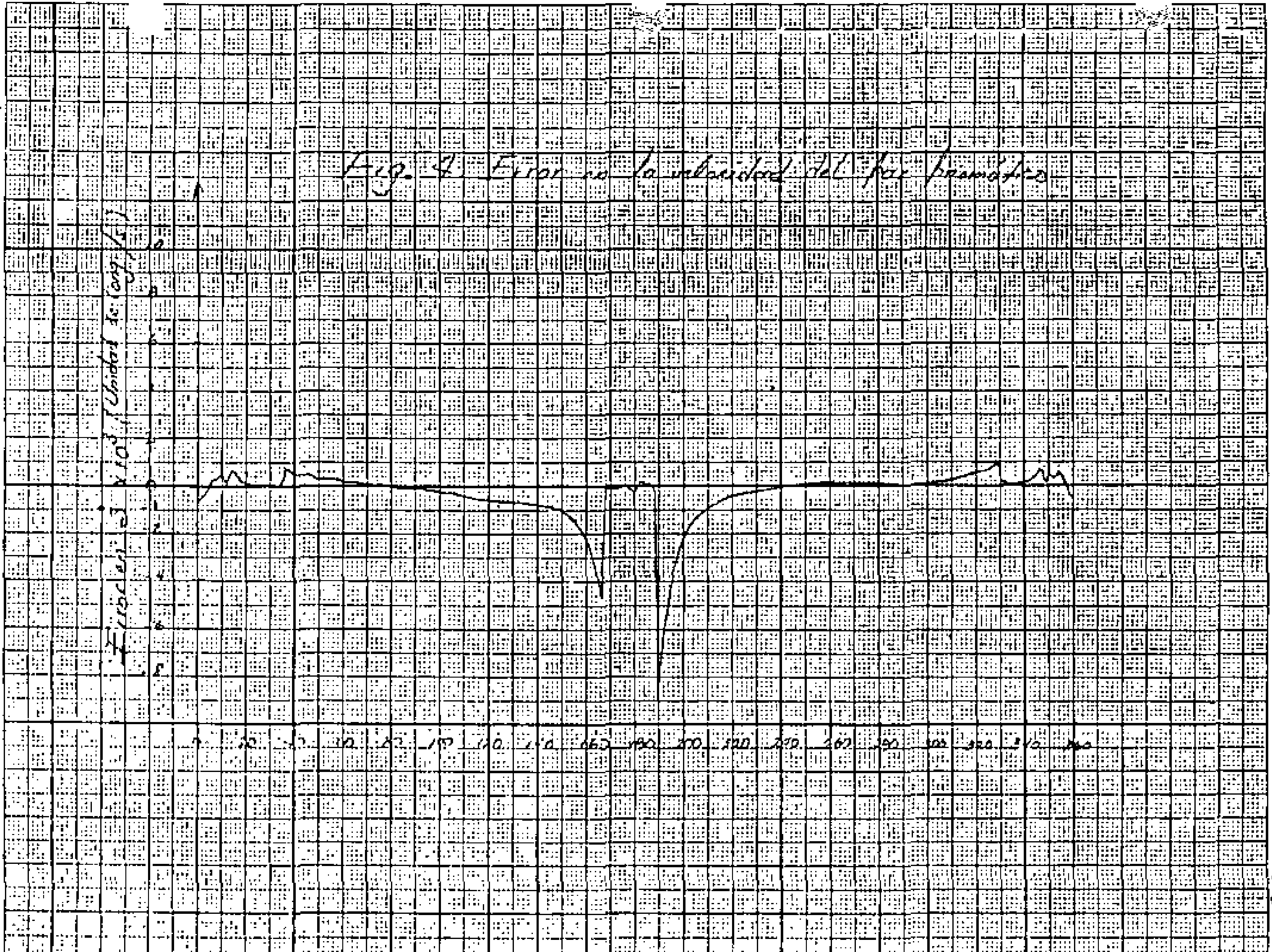


Fig. 7 Error en la velocidad del haz transmitido

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISENO CINEMATICO DE MAQUINARIA

ANALISIS CINEMATICO DE MANIPULADORES CON ARTICULACIONES REDUNDANTES HACIENDO USO DE MINIMIZACION CUADRATICA

JUNIO, 1984.

" ANALISIS CINEMATICO DE MANIPULADORES CON ARTICULACIONES REDUNDANTES HACIENDO USO DE MINIMIZACION CUADRATICA "

Resumen

Se desarrolla un programa de computadora para la solución de sistemas de ecuaciones no lineales subdeterminados, haciendo uso de minimización cuadrática. Como aplicaciones particulares se presenta el problema del análisis cinemático de manipuladores con articulaciones redundantes.

El programa que se presenta puede resolver, asimismo, un problema donde la función objetivo, a minimizar, no sea de norma cuadrática.

Introducción

Debido a que en el estudio de manipuladores se puede presentar el caso de articulaciones redundantes, es decir , que se tengan grados de libertad extra, obteniéndose un sistema de ecuaciones subdeterminado es necesario resolver el problema tratando de extremizar alguna función de los parametros del mecanismo, en este caso, se toma la norma cuadrática de sus ángulos. El programa desarrollado puede aplicarse a cualquier otro tipo de problemas de sistemas de ecuaciones no lineales subdeterminados, extremizando funciones cuadráticas o no cuadráticas.

Desarrollo

El problema consiste en hallar una solución particular de un sistema algebraico no lineal, de la forma:

$$\underline{f}(\underline{x}) = \underline{0} \tag{1}$$

donde $\underline{f}$, $\underline{x}$ y $\underline{0}$ son vectores de dimensión m, n y m, respectivamente, con $m < n$. Dado que el problema es subdeterminado, se dispone, en general, de múltiples soluciones, por lo que es necesario proponer una función objetivo

$z = z(\underset{\sim}{x})$ a extremizar, siendo z un número real. Por lo tanto, podemos plantear el problema como:

$$\min_{\underset{\sim}{x}} z = z(\underset{\sim}{x})$$

sujeta a

$$\underset{\sim}{f}(\underset{\sim}{x}) = \underset{\sim}{0}$$

Para establecer un método que resuelva este problema, se recurren a argumentos geométricos. Cada función $f_i(\underset{\sim}{x})$ representa una superficie en un espacio R^n, al que pertenece $\underset{\sim}{x}$. El conjunto de ecuaciones representado por (1), es decir:

$$f_1(x_1,x_2,\dots,x_n) = 0$$
$$f_2(x_1,x_2,\dots,x_n) = 0$$
$$\dots \qquad \dots$$
$$f_m(x_1,x_2,\dots,x_n) = 0$$

define un subconjunto de puntos de R^n que satisfacen simultaneamente las m ecuaciones no lineales. Así, pues, esas ecuaciones definen una curva Γ en R^n, de grado de libertad n-m, que representa la intersección de esas m superficies.

Supóngase que se conoce un punto P_o de Γ en R^n, con vector de posición $\underset{\sim}{x}^o$, que satisface el sistema (1). Se desea ahora determinar un nuevo punto P^1 de Γ para el cual la función objetivo tenga un valor menor que para $\underset{\sim}{x}^o$, esto es, se desea determinar un vector $\underset{\sim}{x}^1$ tal que:

$$\underset{\sim}{f}(\underset{\sim}{x}^1) = 0$$

y

$$z(\underset{\sim}{x}^1) < z(\underset{\sim}{x}^o)$$

si tal punto no existe, $\underset{\sim}{x}^o$ es un mínimo local del problema.

Debido a que Γ es una curva en R^n, no es posible, en general, definir un punto P^1 de Γ que esté muy alejado de P^o. Se buscará, entonces, un punto cercano a P^o, linealizando el problema y buscando un punto Q^1, de vector de posición $\underset{\sim}{x}^1$ que este contenido en la tangente de Γ en P^o, orientado en la dirección opuesta a ∇z. Como Q^1 puede estar tan alejado de P^1 como sea, es necesario imponer una condición adicional, es decir, que en Q^1, z alcance su valor mínimo a lo largo de la tangente. Para esto definimos:

$$\bar{x}_1 = x_1 + \Delta\bar{x}_1 \tag{2}$$

donde $$\Delta\bar{x}_1 = -\alpha\xi \tag{3}$$

siendo ξ un vector (no necesariamente unitario) contenido en la tangente de Γ, siendo perpendicular a los m vectores $\{\nabla f_i\}$, i = 1,2,...,m. Tal vector satisface, entonces, el siguiente sistema de ecuaciones:

$$\begin{bmatrix} (\nabla f_1)^T \\ (\nabla f_2)^T \\ \cdots \\ (\nabla f_m)^T \end{bmatrix} \xi = \underset{\sim}{0} \qquad J\xi = 0 \tag{4}$$

La ecuación (4) establece que ξ se encuentra en el espacio de J y define un plano perpendicular a la tangente a Γ, como se muestra en la fig. 1.

Si $J^T\lambda$ es el vector del plano más próximo a ∇z, entonces

$$\xi = \nabla z - J^T\lambda \tag{5}$$

así, λ se obtiene como la solución de mínimos cuadrados de

$$J^T\lambda = \nabla z \tag{6}$$

que es

$$\lambda = (JJ^T)^{-1}J\nabla z \tag{7}$$

por lo que, sustituyendo (7) en (5)

$$\xi = [1-J^T(JJ^T)^{-1}J]\nabla z \tag{8}$$

y de (3)

$$\Delta\bar{x}_1 = \alpha[J(JJ^T)^{-1}J - 1]\nabla z \tag{9}$$

donde α se escoge como el valor que minimiza $z(x^o + \Delta\bar{x}_1)$, entonces, α se determina como la solución al problema:

$$\min_{\alpha} z(x^o - \alpha\xi) \tag{10}$$

donde tanto x^o como ξ se conocen.

Para el caso mas general, en el que z sea una función arbitraria de x, α se puede determinar resolviendo un problema no lineal de minimización en una dirección. Una forma eficaz de resolver tal problema es mediante la subrutina FMIN [1].

Si en particular, la función objetivo es cuadrática, es decir, para algún $\bar{x}$ fijo se tiene:

$$z = \frac{1}{2}(\underset{\sim}{x} - \bar{\underset{\sim}{x}})^T W(\underset{\sim}{x}-\bar{\underset{\sim}{x}}) \tag{11}$$

siendo W positiva definida, se tiene que z tendra un mínimo a lo largo de la tangente en un punto donde se anule $dz/d\alpha$, es decir, el problema se reduce a la búsqueda del mínimo en una dirección; pero

$$\frac{dz}{d\alpha} = \left(\frac{\partial z}{\partial \underset{\sim}{x}^1}\right)^T \cdot \frac{\partial x^1}{\partial \alpha} \tag{12}$$

siendo $$\underset{\sim}{x}^1 = \underset{\sim}{x} - \bar{\underset{\sim}{x}} = \underset{\sim}{x}^o - \alpha\underset{\sim}{\xi} - \bar{\underset{\sim}{x}} = \underset{\sim}{x}^o - \bar{\underset{\sim}{x}} - \alpha\underset{\sim}{\xi} \tag{13}$$

por lo que $$\frac{\partial z}{\partial \underset{\sim}{x}^1} = W(\underset{\sim}{x} - \bar{\underset{\sim}{x}}) = W(\underset{\sim}{x}^o - \bar{\underset{\sim}{x}} - \alpha\underset{\sim}{\xi}) \tag{14}$$

y $$\frac{\partial x^1}{\partial \alpha} = \frac{\partial}{\partial \alpha}(\underset{\sim}{x}^o - \alpha\underset{\sim}{\xi}) = -\underset{\sim}{\xi} \tag{15}$$

por lo tanto $$\frac{dz}{d\alpha} = -\underset{\sim}{\xi}^T W(\underset{\sim}{x}^o - \bar{\underset{\sim}{x}} - \alpha\underset{\sim}{\xi}) = 0 \tag{16}$$

$$\Rightarrow \quad \alpha\underset{\sim}{\xi}^T W\underset{\sim}{\xi} = \underset{\sim}{\xi}^T W(\underset{\sim}{x}^o - \bar{\underset{\sim}{x}}) \tag{17}$$

pero $$W(\underset{\sim}{x}^o - \bar{\underset{\sim}{x}}) = \nabla z \tag{18}$$

por lo tanto $$\alpha = \frac{\underset{\sim}{\xi}^T W \nabla z}{\underset{\sim}{\xi}^T W \underset{\sim}{\xi}} \tag{19}$$

$$\underset{\sim}{\xi}^T W \nabla z = (\nabla z)^T [1 - J^T(JJ^T)^{-1}J]\ W \nabla z \tag{20}$$

$$\underset{\sim}{\xi}^T W \underset{\sim}{\xi} = (\nabla z)^T [1 - J^T(JJ^T)^{-1}J]\ W(\nabla z)^T [1 - J^T(JJ^T)^{-1}J] \tag{21}$$

Si además, W = 1

$$\underset{\sim}{\xi}^T W \underset{\sim}{\xi} = (\nabla z)^T [1 - J^T(JJ^T)^{-1}J]^2 \nabla z =$$
$$= (\nabla z)^T [1 - J^T(JJ^T)^{-1}J] \nabla z \tag{22}$$

por lo tanto $\alpha = 1$

Una vez determinada α , se tiene $\Delta \underset{\sim}{x}^1$ calculando $\underset{\sim}{\xi}$ de la ecuación (6) por medio de las subrutinas Hecomp y Holve [2]. Sin embargo se tendrá la configuración de la fig. 2, de donde se observa que, aunque $z(\bar{x}_1) < z(x^o)$, $\bar{P}_1$, cuyo vector de posición es $\bar{x}_1$, esta fuera de Γ, esto es,

$$f(\bar{x}_1) \neq 0 \tag{23}$$

o sea, $\bar{x}_1$ no satisface el sistema (1). Es necesario determinar ahora una corrección $\Delta\bar{\bar{x}}_1$ a $\bar{x}_1$ tal que

$$x_1 = \bar{x}_1 + \Delta\bar{\bar{x}}_1 \tag{24}$$

sí satisfaga a (1). Para resolver este problema existen varias alternativas que son:

Caso 1.- Como $\Delta\bar{x}_1$ está en el complemento del plano normal a Γ, si $\bar{P}_1$ no esta muy alejado de P_o, se puede pensar que, si se pasa un plano por $\bar{P}_1$ paralelo a la normal a Γ, ese plano cortará a Γ, como se muestra en la fig. 3. La simplificación estriba en que $\Delta\bar{\bar{x}}_1$ se habrá obtenido mediante la solución del sistema

$$f(\bar{x}_1 + \Delta\bar{\bar{x}}_1) = 0 \tag{25}$$

donde $$\bar{\bar{x}}_1 = J^T\mu \tag{26}$$

Como se conocen $\bar{x}_1$ y J, la incógnita es μ, de dimensión m. Así la ecuación (25) se reduce a

$$F(\mu) = 0 \tag{27}$$

que es un sistema algebraico no lineal de m ecuaciones en m incógnitas, que se puede resolver por el método de Newton - Raphson para sistemas determinados, esto se hace mediante la subrutina NRDAMP [3] que a su vez requiere de las subrutinas DECOMP y SOLVE para la solución de sistemas de ecuaciones determinados en base a la descomposicion LU de matrices [4].

Caso 2.- Otra alternativa es seleccionar P_1 como el punto de Γ más próximo a $\bar{P}_1$, esto es, como el punto de tangencia de una esfera centrada en $\bar{P}_1$ con Γ. En este caso, $\Delta\bar{\bar{x}}_1$ se puede obtener como la solución al problema (fig. 2):

$$\min_{\Delta \bar{\bar{x}}_1} 1/2 \; \Delta \bar{\bar{x}}_1^T \Delta \bar{\bar{x}}_1 \tag{28}$$

sujeta a $$\underline{f}(\bar{\bar{x}}_1 + \Delta \bar{\bar{x}}_1) = \underline{0} \tag{29}$$

Incorporando la ecuación (29) a la función objetivo, mediante multiplicadores de Lagrange . Así, se define:

$$\Psi(\Delta \bar{\bar{x}}_1) = \frac{1}{2} \Delta \bar{\bar{x}}_1^T \Delta \bar{\bar{x}}_1 + \underline{y}^T \underline{f}(\bar{\bar{x}}_1 + \Delta \bar{\bar{x}}_1) \tag{30}$$

que tiene un extremo cuando

$$\frac{d\,\Psi}{d\,\Delta \bar{\bar{x}}_1} = \Delta \bar{\bar{x}}_1 + \left(\frac{\partial f}{\partial \bar{\bar{x}}_1}\right)^T \underline{y} = 0 \tag{31}$$

donde $$\underline{x} = \bar{\bar{x}}_1 + \Delta \bar{\bar{x}}_1 \tag{32}$$

Así $$\frac{\partial f}{\partial \Delta \bar{\bar{x}}_1} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial \Delta \bar{x}_1} = J \cdot 1 = J \tag{33}$$

Sustituyendo (33) en (31) y despejando, se tiene

$$\Delta \bar{\bar{x}}_1 = J^T \underline{y} \tag{34}$$

por lo tanto: $$\underline{f}(\bar{\bar{x}}_1 + J^T \underline{y}) = \underline{f}(\underline{y}) = \underline{0} \tag{35}$$

que es un sistema no lineal de ecuaciones en m incógnitas como en el caso anterior, por lo que también puede resolverse mediante la subrutina NRDAMP.

Caso 3.- Por último, se puede usar

$$J\,\Delta x = -f \tag{36}$$

o bien $$\Delta x = -J^T(JJ^T)^{-1} f \tag{37}$$

actualizando J. En cualquier caso, se alcanzará un mínimo cuando se satisfagan los criterios de primerio y de segundo orden, obteniéndose a continuación. Adjuntando la restricción (1) a la función objetivo $z = z(x)$, obteniéndose

$$\Psi(\underline{x}) = z(\underline{x}) + \underline{\lambda}^T \underline{f}(\underline{x}) \tag{38}$$

se tiene un punto estacionario de Ψ cuando se anula su gradiente, escogiendo $\underline{\lambda}$ de manera que produzca una $\underline{x}$ que satisfaga (1). Así,

$$\frac{\partial \Psi}{\partial \underline{x}} = \nabla z + J^T \underline{\lambda} = \underline{0} \tag{39}$$

o bien, $$J^T \underline{\lambda} = -\nabla z \qquad (40)$$

que son las ecuaciones de normalidad, o sea la condición necesaria de primer orden para la existencia de un mínimo. La interpretación geométrica de la ecuación (40) es que, en un punto estacionario, ∇z se encuentra en el plano normal a Γ. La interpretación algebraica de esta ecuación es que, en un punto estacionario, ∇z está en el codominio de J^T, un subespacio de R^n, de dimensión $m < n$.

La solución de la ecuación (36) puede obtenerse mediante la subrutina SUBDEL que resuelve un sistema algebraico lineal subdeterminado, auxiliado por la subrutina HECOMP.

Descripción del programa

El programa desarrollado, SANLSU, puede resolver el problema según los casos 2 y 3 descritos anteriormente. En la fig. 4 se muestra el diagrama de flujo del programa. Los listados del programa y las subrutinas para cada ejemplo resuelto se muestran en el apéndice A.

Una vez obtenidos los resultados del problema, se almacenan en un archivo de datos que es leído por el programa GRAMAN que dibuja el mecanismo en pantalla con opción de graficar en papel empleando la subrutina SCRPRT. El listado del programa GRAMAN también se encuentra en el apéndice A.

La computadora empleada en la solución de estos problemas es una APPLE IIe de 64K de memoria, con lo que se demuestra que no es necesaria gran cantidad de memoria para resolver este tipo de problemas.

Ejemplos

Ejemplo 1: Considere el siguiente problema en el que $m=1$, $n=2$ y $\underline{f}$ y z están dadas, respectivamente por:

$$f(x_1,x_2) = x_1^2 + 4x_2^2 - 1 = 0$$
$$z(x_1,x_2) = 1/2(x_1^2 + x_2^2)$$

Determinar los puntos de coordenadas (x_1,x_2) que, satisfaciendo $f(x_1,x_2) = 0$, minimicen z.

Solucion:

La interpretación geométrica del problema se ilustra en la fig. 5, donde la elipse esta representada por la función F y Z representa la mitad del cuadrado de la distancia de un punto al origen. Como se puede observar de la figura, en los puntos A, B, C y D z alcanza valores estacionarios que representan la solución del problema.

La matriz jacobiana de f y el gradiente de z son, respectivamente:

$$J = \begin{bmatrix} 2x_1 & 8x_2 \end{bmatrix} \qquad \nabla z = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

En los puntos A y C, J y z toman los siguientes valores:

$$J = \begin{bmatrix} 2 & 0 \end{bmatrix} \qquad z = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

y en B y D:

$$J = \begin{bmatrix} 0 & 4 \end{bmatrix} \qquad z = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$$

De estos resultados se verifica la ecuación (6), es decir, la estacionaridad de los puntos A, B, C, y D. Los resultados del problema se muestran en el apéndice B.

Ejemplo 2: Se requiere posicionar un punto en un plano mediante una cadena cinemática abierta (manipulador articulado) de triple grado de libertad, con cuatro eslabones articulados, mostrada en la figura 6. La longitud de sus eslabones móviles a_2, a_3, a_4 son unitarias. Determinar los ángulos θ_1, θ_2 y θ_3 entre sus eslabones para que el punto P tenga las coordenadas del punto Q(0,1). Más aún, se requiere que el punto P genere una trayectoria prescrita.

Solución:

Debido a que en el problema existen infinidad de soluciones que posicionan el punto P en Q, se debe seleccionar la mejor de estas según algún criterio que permita escoger una función z a extremizar.

De la fig. 6 se puede ver que la ecuacion $\underset{\sim}{f}(\underset{\sim}{x}) = \underset{\sim}{0}$ es :

$$f_1(\theta_1,\theta_2,\theta_3) = a_1 c(\theta_1) + a_2 c(\theta_1+\theta_2) + a_3 c(\theta_1+\theta_2+\theta_3) - x = 0$$

$$f_2(\theta_1,\theta_2,\theta_3) = a_1 s(\theta_1) + a_2 s(\theta_1+\theta_2) + a_3 s(\theta_1+\theta_2+\theta_3) - y = 0$$

se puede elegir z como :

$$z = 1/2(\theta_1^2+\theta_2^2+\theta_3^2) \qquad \text{donde: } c()=\cos() \text{ y } s()=\text{sen}()$$

que pũede representar una función de costo, donde se desea minimizar las rotaciones de los eslabones. En base a lo anterior, podemos formar el jacobiano de f y el gradiante de z que son, entonces:

$$J = \begin{bmatrix} -s(\theta_1)-s(\theta_1+\theta_2)-s(\theta_1+\theta_2+\theta_3) & -s(\theta_1+\theta_2)-s(\theta_1+\theta_2+\theta_3) & -s(\theta_1+\theta_2+\theta_3) \\ c(\theta_1)+c(\theta_1+\theta_2)+c(\theta_1+\theta_2+\theta_3) & c(\theta_1+\theta_2)+c(\theta_1+\theta_2+\theta_3) & c(\theta_1+\theta_2+\theta_3) \end{bmatrix}$$

$$\nabla z = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

Los resultados numéricos, así como las gráficas de los manipuladores se mustran en el apéndice B. En primer lugar se resolvió el problema según el caso 2 antes descrito (fig. 7), las figuras 8 y 9 representan al manipulador siguiendo una trayectoria recta, habiéndose resuelto el problema según el caso 3, en la figura 9 sólo se muestran algunas de las configuraciónes que se siguierón para llegar al punto máximo.

Otro problema que se resolvió fue el de generar una trayectoria aleatoria. En la figura 10 se muestran los resultados según el caso 3 y en la figura 11 se muestra la configuración original y el primer punto de la trayectoria, ya que el caso 2 no convergió debido a la gran separación entre los puntos de la trayectoria.

Ejemplo 3: Obtener la posición óptima de un manipulador de 6 eslabones articulados, a partir de una posición dada.

Solución:

La configuración general del mecanismo se muestra en la fig. 12. Las ecuaciones para el análisis de desplazamiento de una cadena cinemática pueden obtenerse mediante las condiciones de cerradura de desplazamiento y de rotación. De acuerdo con el método y la notación de Denavit y Hartenberg [5] los n eslabones de una cadena cinemática se numeran ordenadamente, 1 a n y a i^o eslabón se fija el sistema coordenado X_i, Y_i, Z_i. Así $[Q_{i,i+1}]_i$ representa una matriz ortogonal, referida a los ejes X_i, Y_i, Z_i que gira éstos a una posición coincidente con los correspondientes X_{i+1}, Y_{i+1}, Z_{i+1}; por su parte $[a_{i,i+1}]_i$ es el vector que une los origenes O_i y O_{i+1} de los ejes anteriores dirigidos del primero al segundo y referido a ejes fijos al i^o eslabón. Así, las condiciones de cerradura son:

$$[Q_{1,2}]_1 \, [Q_{2,3}]_2 \; \ldots \; [Q_{5,6}]_1 = [\underset{\sim}{Q}]_1$$

para rotación y

$$[a_{1,2}]_1 + [a_{2,3}]_1 + \; \ldots \; + [a_{5,6}]_1 = \{\underset{\sim}{r}\}_1$$

para desplazamiento, donde $\{\underset{\sim}{r}\}_1$ es el vector de posición del punto P del organo terminal.

Debido a que en nuestro caso no importa la orientación del organo terminal, podemos considerar unicamente las condiciones de cerradura para desplazamiento y, haciendo coincidir el punto B con el A, tendremos un sistema de 3 ecuaciones con cinco incognitas, es decir:

$$\underset{\sim}{f}(\theta\;,\theta\;,\theta\;,\theta\;,\theta\;) = [a_{1,2}]_1 + \ldots + [a_{5,6}]_1 - \{\underset{\sim}{r}\}_1 = \underset{\sim}{0}$$

Para el cálculo de $\partial f/\partial \underset{\sim}{\theta}$ definamos:

$$\underset{\sim}{x}_1(\underset{\sim}{\theta}) = \underset{\sim}{a}_1 + \underset{\sim}{Q}_1\underset{\sim}{a}_2 + \underset{\sim}{Q}_1\underset{\sim}{Q}_2\underset{\sim}{a}_3 + \underset{\sim}{Q}_1\underset{\sim}{Q}_2\underset{\sim}{Q}_3\underset{\sim}{a}_4 + \underset{\sim}{Q}_1\underset{\sim}{Q}_2\underset{\sim}{Q}_3\underset{\sim}{Q}_4\underset{\sim}{a}_5$$

donde : $\underset{\sim}{a}_i = [a_{i,i+1}]_i$

$$\underset{\sim}{Q}_i = [Q_{i,i+1}]_i$$

que puede calcularse mediante el algoritmo de Horner para evaluación de polinomios [6] como:

$$\underset{\sim}{x}_5 = \underset{\sim}{a}_5$$

$$\underset{\sim}{x}_k = \underset{\sim}{a}_k + \underset{\sim}{Q}_k\,\underset{\sim}{x}_{k+1} \qquad k = 4,3,\ldots,1$$

Así: $\dfrac{\partial f}{\partial \underset{\sim}{\theta}_i} = \dfrac{\partial \underset{\sim}{x}_1}{\partial \underset{\sim}{\theta}_i}$

según la fig,13 y la notación de Denavit y Hartenberg se tiene

$$\underset{\sim}{a}_i = \begin{bmatrix} a_i c(\theta_i) \\ a_i s(\theta_i) \\ b_i \end{bmatrix}$$

$$\underset{\sim}{Q}_i = \begin{bmatrix} c(\theta_i) & -s(\theta_i)c(\alpha_i) & s(\theta_i)s(\alpha_i) \\ s(\theta_i) & c(\theta_i)c(\alpha_i) & -c(\theta_i)s(\alpha_i) \\ 0 & s(\alpha_i) & c(\alpha_i) \end{bmatrix}$$

donde a_i es la distancia entre los ejes Z_i y Z_{i+1}, b_i es la coordenada de la intersección de X_{i+1} con Z_i en el sistema X_i, Y_i, Z_i, α_i es el ángulo entre Z_i y Z_{i+1}, medido en la dirección positiva de X_{i+1}.

Para la solución de este problema se utilizaron, además, las subrutinas VECX y PRODQ [7].

Los resultados se muestran en el apéndice B, así como las configuraciones del manipulador en las figuras 14 y 15.

Conclusiones

Como se puede observar, el método más eficiente fué el implementado para el caso 3, ya que no es necesario que los puntos de la trayectoria esten muy cercanos, como se requiere para el caso 2, además, el tiempo de proceso para el caso 2 fue mayor que para el tercer caso.

Hay que hacer notar que este programa puede ser implementado en cualquier computador ya que no requiere de gran capacidad de memoria.

El programa puede modificarse fácilmente para resolver cualquier problema de este tipo, es decir, solución de sitemas no lineales de ecuaciones subdeterminados.

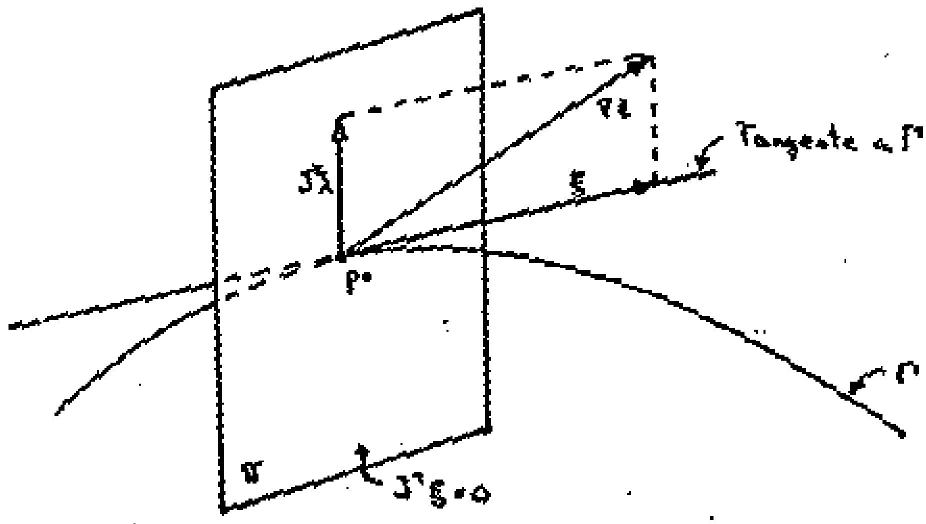


FIGURA 1

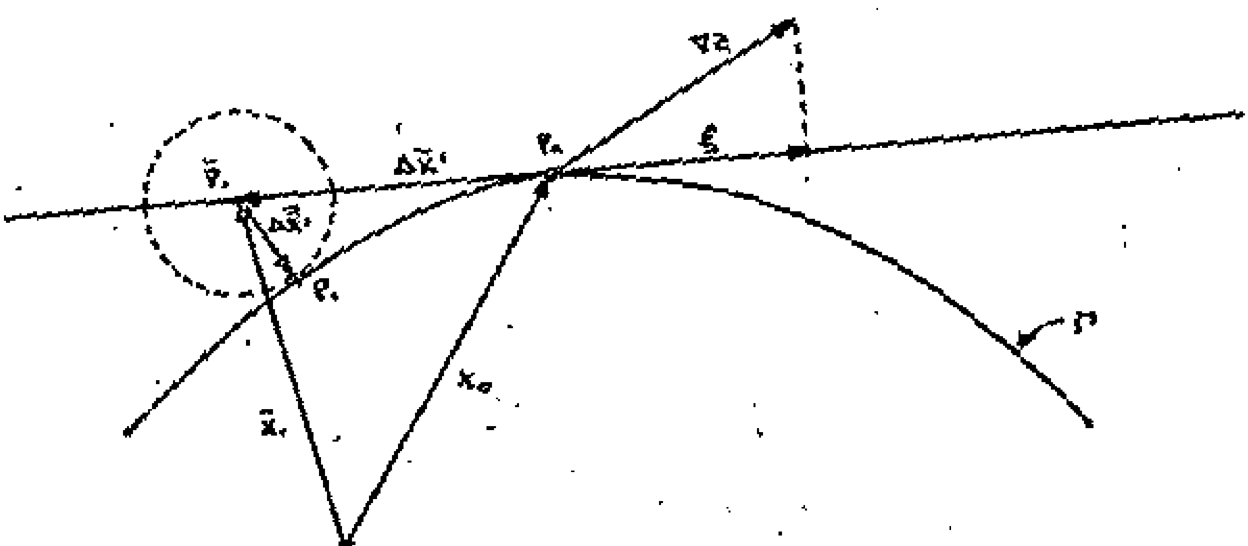


FIGURA 2

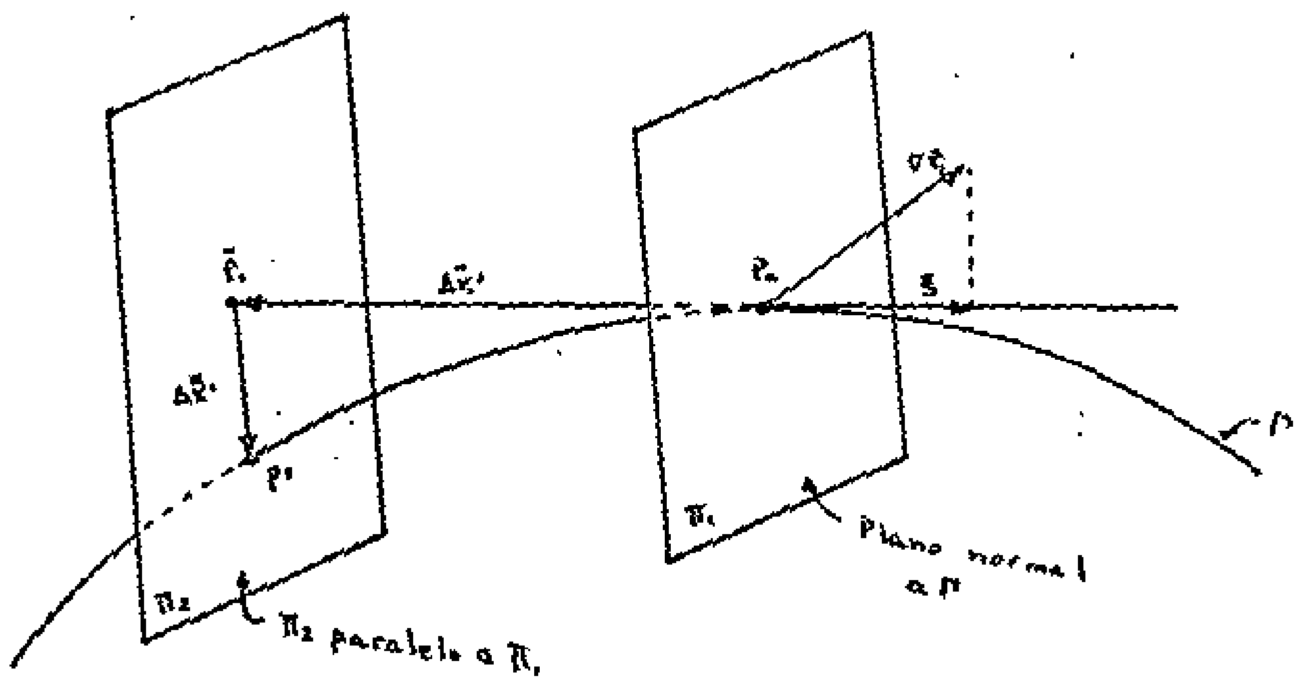
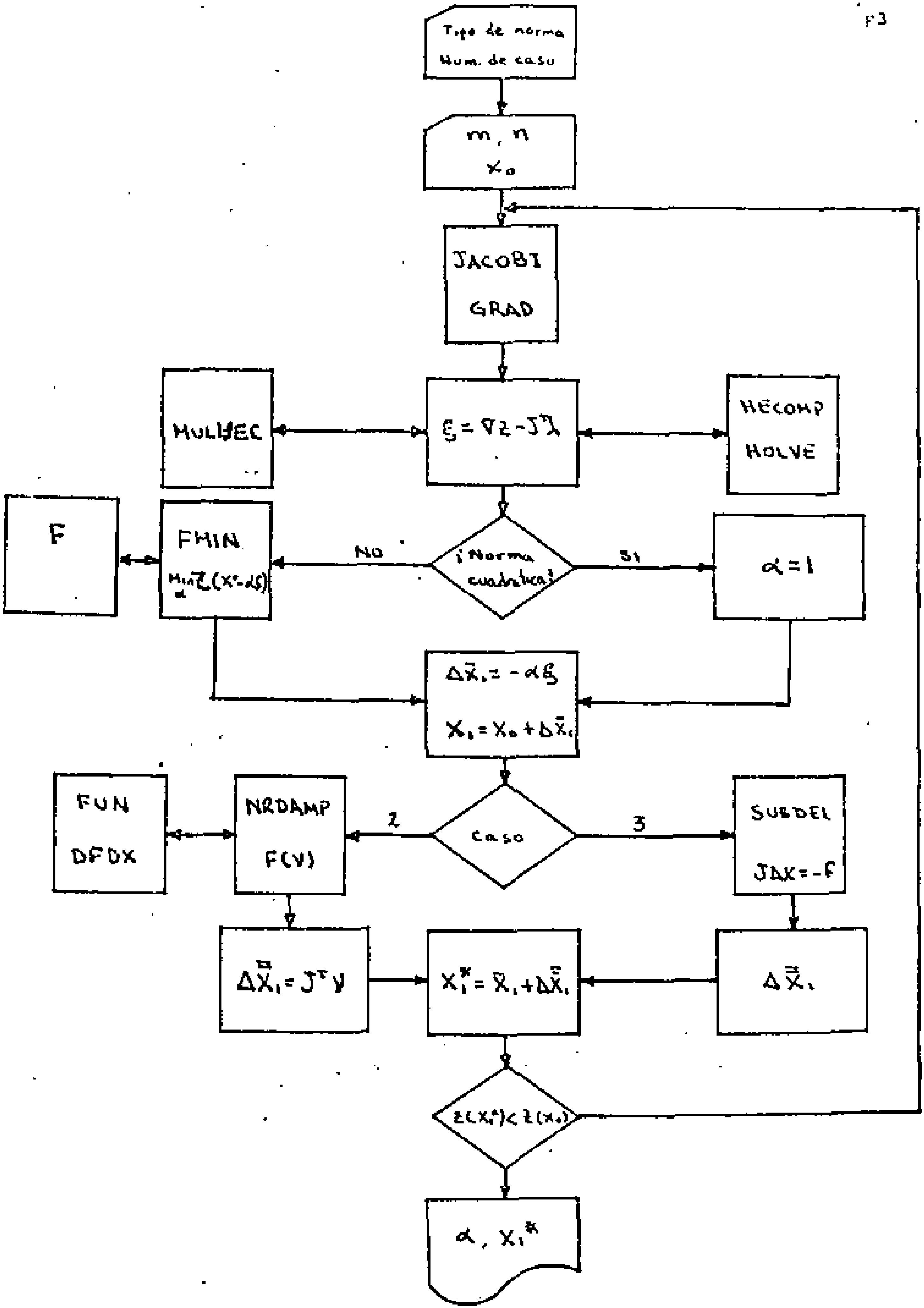


FIGURA 3


Tipo de norma
Num. de caso
m, n
x_0
JACOBI
GRAD
MULVEC
$\xi = \nabla z - J^T \lambda$
DECOMP
SOLVE
F
FMIN
$\min_{\alpha} z(x^0 - \alpha\xi)$
NO
¿Norma cuadrática?
SI
$\alpha = 1$
$\Delta \bar{x}_1 = -\alpha \xi$
$x_1 = x_0 + \Delta \bar{x}_1$
FUN
DFDX
NRDAMP
F(Y)
2
caso
3
SUBDEL
$J\Delta x = -F$
$\Delta \bar{x}_1 = J^T y$
$x_1^* = \bar{x}_1 + \Delta \bar{x}_1$
$\Delta \bar{x}_1$
$z(x_1^*) < z(x_0)$
$\alpha,\ x_1^*$

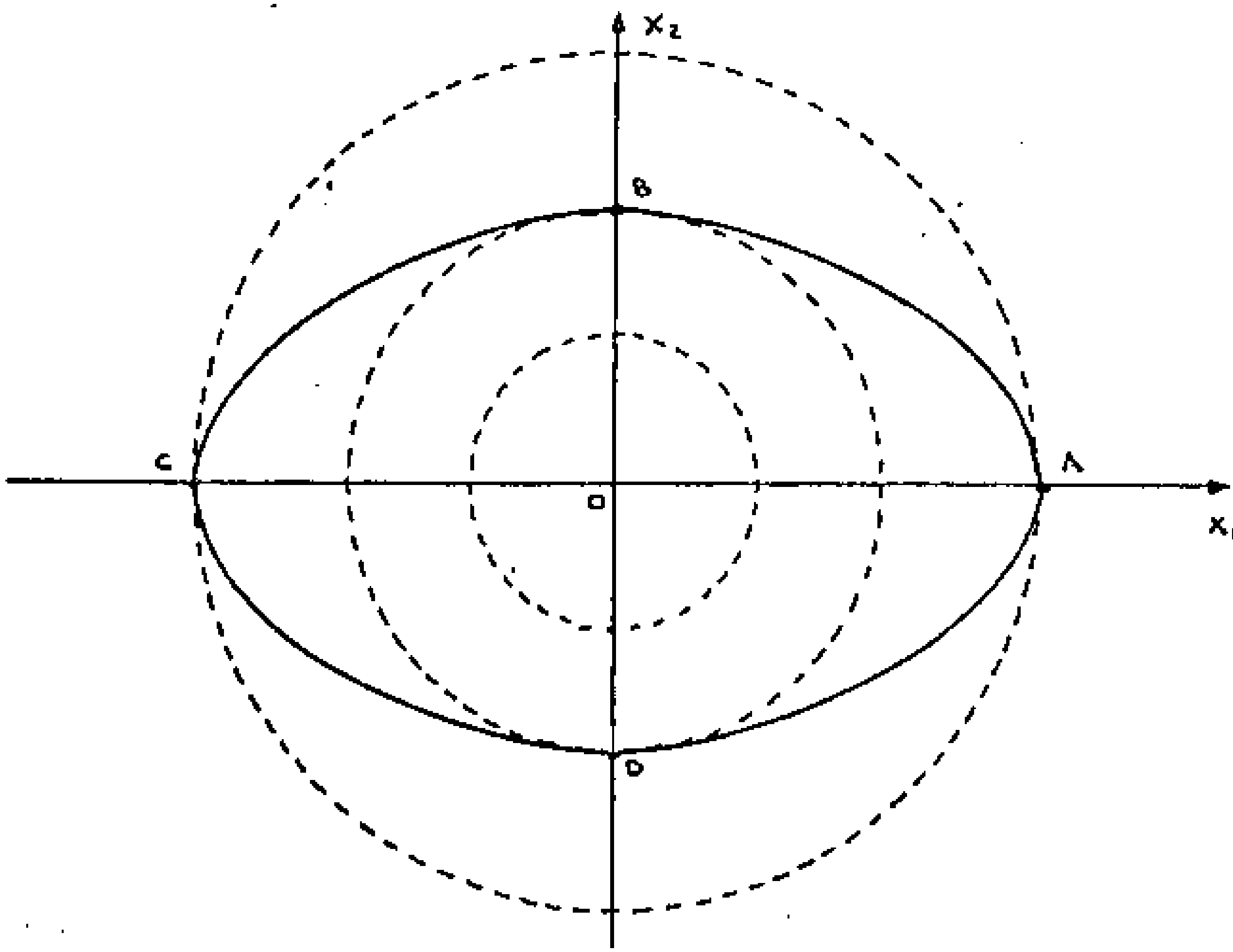


FIGURA 5

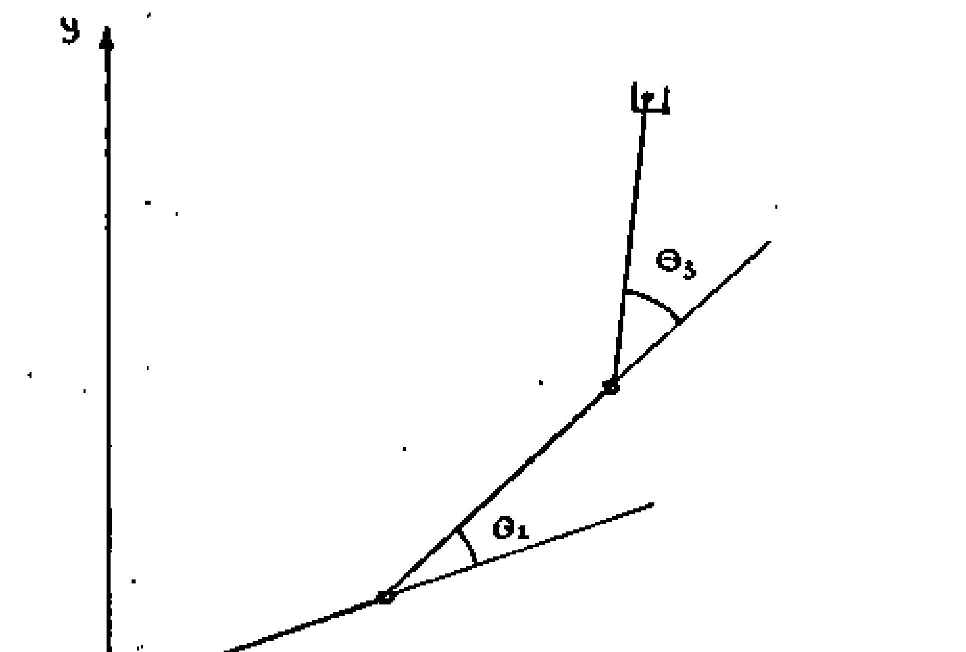


FIGURA 6

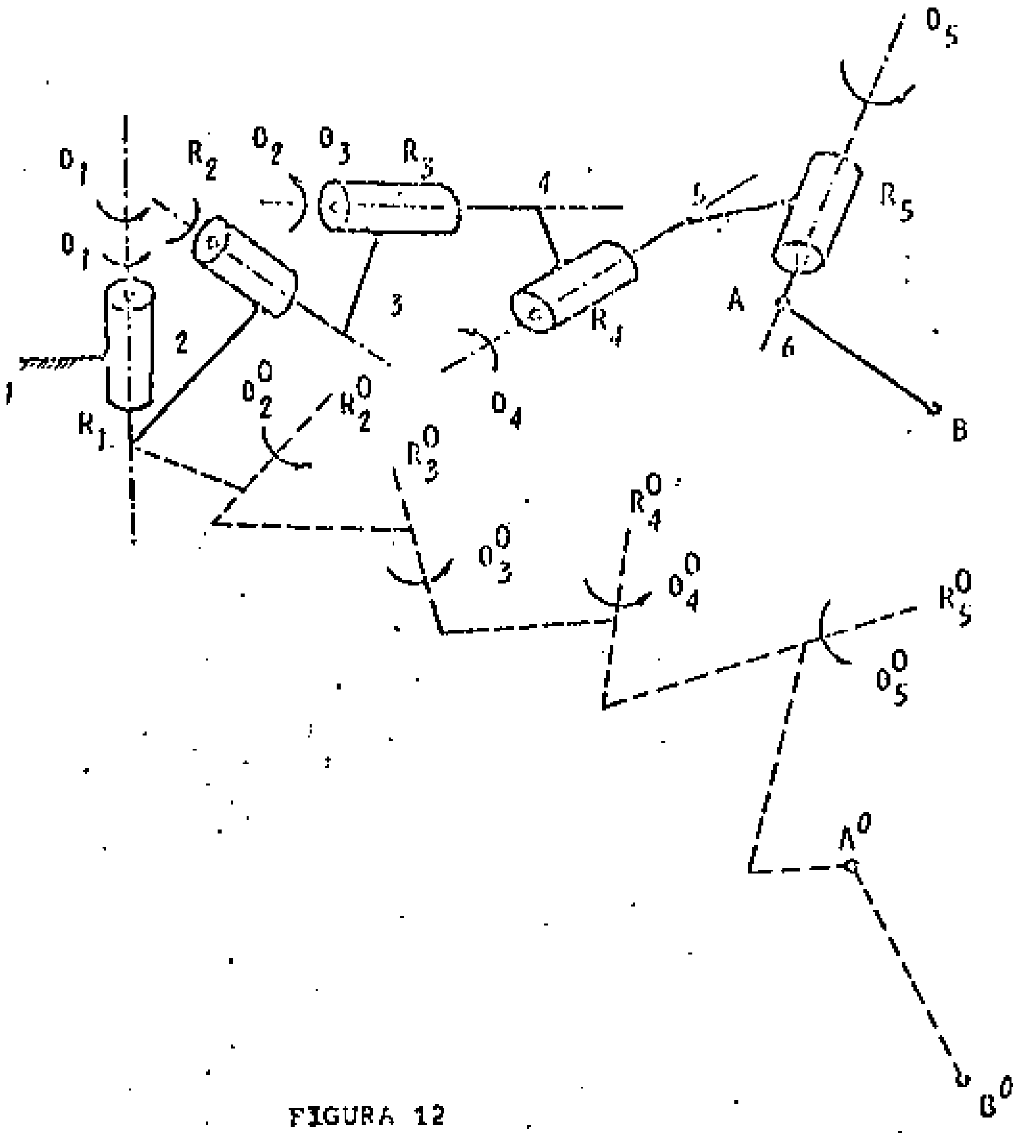


FIGURA 12

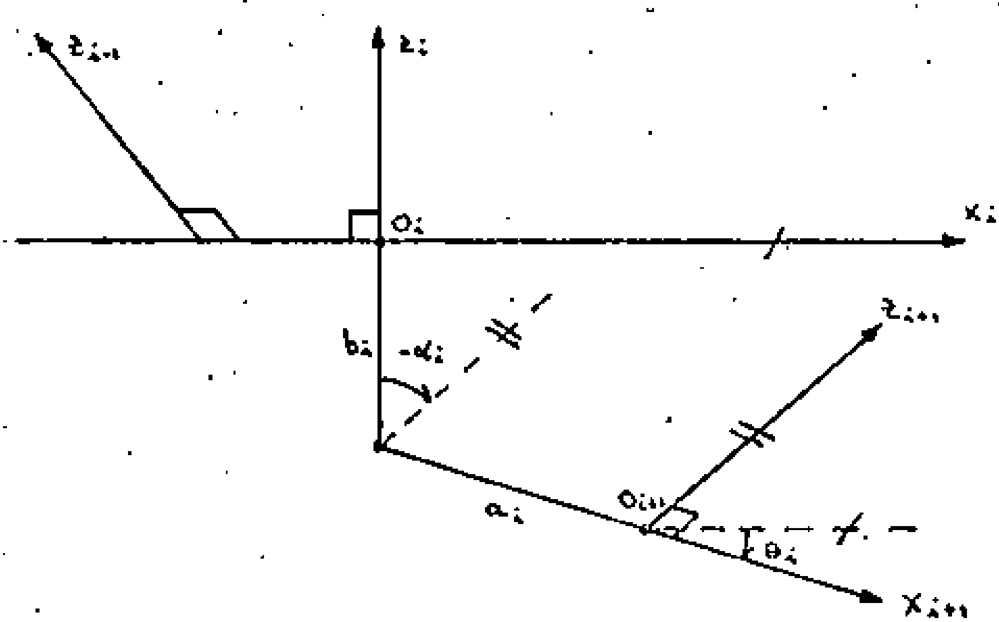


FIGURA 13

EJEMPLO 1

```
      PROGRAM SANLSU
C
C           PROGRAM SANLSU
C
C     ESTE PROGRAMA CALCULA LA SOLUCION DE UN SISTEMA ALGEBRAICO
C     NO LINEAL SUBDETERMINADO DE LA FORMA :
C                        F(X)=0
C     DONDE F, X Y O SON VECTORES DE DIMENSION M, N Y M RESPECTIVAMENTE
C     CON M<N.  DEBIDO A ESTO ES NECESARIO PROPONER UNA FUNCION OBJETIVO
C     Z=Z(X) A EXTREMIZAR, SIENDO Z UN NUMERO REAL.
C     EN RESUMEN, TENEMOS EL SIGUIENTE PROBLEMA DE PROGRAMACION MATEMA-
C     TICA:
C                        MIN Z=Z(X)
C                         X
C     SUJETA A:          F(X)=0
C
      DIMENSION GZ(2),DELTX(2),X1(2),DELTX1(2),XFIN(2),U(2)
      REAL MU(1),JAC(1,2),JACT(2,1),JTL(2)
      COMMON X0(2),M,N
      COMMON CSI(2)
      WRITE(*,'(A\)')' SI DESEA IMPRESORA,TECLEE : '
      READ(*,'(I1)') I
      IF(I.EQ.1) THEN
         OPEN(6,FILE='PRINTER:')
      ELSE
         OPEN(6,FILE='CONSOLE:')
      ENDIF
      WRITE(6,50)
      WRITE(*,'(A\)')' 1 SI EL PROBLEMA ES DE NORMA CUADRATICA '
      READ(*,'(I1)') I1
      IF(I1.EQ.1) ALFA=1.0
      WRITE(6,100)
      READ(*,200) M,N
      WRITE(6,200) M,N
      WRITE(6,300)
      READ(*,400) (X0(I),I=1,N)
      WRITE(6,400) (X0(I),I=1,N)
      WRITE(6,350)
      TOLX=0.0000001
      TOLF=0.000001
      DAMP=0.5
      MAX=200
      KMAX=100
      WRITE(6,600) TOLX,TOLF,DAMP,MAX,KMAX
      MU(1)=0.2
11    KONT=KONT+1
      CALL JACOBI(JAC)
      CALL GRAD(GZ)
      DO 5 I=1,N
         JTL(I)=GZ(I)
            DO 5 J=1,M
5     JACT(I,J)=JAC(J,I)
      CALL HECOMP(N,N,M,JACT,U)
      CALL HOLVE(N,N,M,JACT,U,JTL)
      DO 3 I=1,N
            DO 3 J=1,M
3     JACT(I,J)=JAC(J,I)
      CALL MULVEC(N,M,JACT,JTL,JTL)
      DO 4 I=1,N
4     CSI(I)=GZ(I)-JTL(I)
      ...  ALFA=FMIN(-1.5,1.5,1.0E-5)
```

```
                DELTX(I)=-J FANCOT(I)
6       X1(I)=XO(I)+DELTX(I)
        CALL NEGAUC(MI,X1,JACT,TOLX,TOLF,DAMF,ITER,MAX,KMAX)
        CALL MULVEC(N,M,JACT,MU,DELTX1)
        DO 9 I=1,N
9       XFIN(I)=X1(I)+DELTX1(I)
        DO 14 I=1,N
                IF(ABS(XFIN(I)-XO(I)).GT.1.0E-6) GO TO 12
14      CONTINUE
        GO TO 10
12      DO 13 I=1,N
13      XO(I)=XFIN(I)
        GO TO 11
10      WRITE(6,'(A)') ' LA SOLUCION FINAL DEL PROBLEMA ES:'
        WRITE(6,500) ALFA
        WRITE(6,501) (CSI(I),I=1,N)
        WRITE(6,505) (MU(I),I=1,M)
        WRITE(6,507) (XFIN(I),I=1,N)
        STOP
50      FORMAT(3X,' PROGRAMA PARA LA SOLUCION DE SISTEMAS DE ECUACIO
     &  'NES'/15X,' NO LINEALES SUBDETERMINADOS'/)
100     FORMAT(3X,' DAME LAS DIMENSIONES DE F(X) Y X (M Y N)')
150     FORMAT(I1)
200     FORMAT(2I2)
250     FORMAT(3X,' NO SE LLEGA A LA SOLUCION DESPUES DE 50 VECES')
300     FORMAT(3X,' DAME XO')
350     FORMAT(' TOLX, TOLF, DAMP, MAX Y KMAX')
400     FORMAT(2F10.5)
500     FORMAT(/3X,'ALFA'/3X,F10.4)
501     FORMAT(3X,'CSI'/3X,2F10.4)
505     FORMAT(3X,'MU'/3X,F10.4)
507     FORMAT(3X,'XF(I)'/3X,2F10.4)
600     FORMAT(2E15.5,F5.3,2I5)
700     FORMAT(2F15.5,F5.3,2I3)
        END
C
C
        SUBROUTINE JACOBI(JAC)
C
C       SUBRUTINA QUE CALCULA EL JACOBIANO DE LA FUNCION F(X) EMPLEADA
C       EN EL PROGRAMA SANLDU.
C
C       ENTRADAS:
C               M       = DIMENSION DE F(X)
C               N       = DIMENSION DE X
C               XO      = VALOR INICIAL
C       SALIDAS:
C               JAC     = JACOBIANO DE F(X)
C
        REAL JAC(1,2)
        COMMON XO(2),M,N
        DO 1 I=1,M
                DO 1 J=1,N
1       JAC(I,J)=0.0
        JAC(1,1)=2.0*XO(1)
        JAC(1,2)=3.0*XO(2)
        RETURN
        END
```

```
C
C
      SUBROUTINE GRAD(GZ)
C
C     SUBRUTINA QUE CALCULA EL GRADIENTE DE LA FUNCION Z(X) EMPLEADA
C     EN EL PROGRAMA SANLRU.
C
C     ENTRADAS:
C              N        = DIMENSION DE Z(X)
C              XO       = PUNTO INICIAL
C     SALIDAS:
C              GZ       = GRADIENTE DE Z(X)
C
      DIMENSION GZ(2)
      COMMON XO(2),M,N
      DO 1 I=1,N
1     GZ(I)=0.0
      DO 2 I=1,N
2     GZ(I)=XO(I)
      RETURN
      END
      SUBROUTINE FUN(X,F,JACT,P)
      COMMON XO(2),M,N
      DIMENSION Y(2),P(2),F(1),X(1)
      REAL JACT(2,1)
      DO 1 I=1,M
            DO 1 J=1,N
1     Y(J)=P(J)+JACT(J,I)*X(I)
      F(1)=Y(1)**2+4.0*Y(2)**2-1.0
      RETURN
      END
C
C
      SUBROUTINE DFDX(X,DF,JACT,P)
      COMMON XO(2),M,N
      DIMENSION X(1),P(2),DF(1,1)
      REAL JACT(2,1)
      DF(1,1)=2.0*(P(1)+JACT(1,1)*X(1))*JACT(1,1)+
     1 8.0*(P(2)+JACT(2,1)*X(1))*JACT(2,1)
      RETURN
      END
      REAL FUNCTION F(ALFA)
      REAL ALFA
      COMMON XO(2),M,N
      COMMON CSI(2)
      CALL FUNZ(ALFA,Z)
      F=Z
      RETURN
      END
C
C
      SUBROUTINE FUNZ(ALFA,Z)
C
C     CALCULA LA FUNCION Z(XO-ALFA*CSI)
C
      COMMON XO(2),M,N
      COMMON CSI(2)
      Z=0.0
      DO 1 I=1,N
1     Z=Z+(XO(I)-ALFA*CSI(I))**2
      Z=0.5*Z
      RETURN
      END
```

EJEMPLO 2


```
$USES [illegible] IN #5:[illegible].CODE
$USES [illegible] IN #5:[illegible].CODE
      PROGRAM SANLSD
C
C             PROGRAM SANLSD
C
C     ESTE PROGRAMA CALCULA LA SOLUCION DE UN SISTEMA ALGEBRAICO
C     NO LINEAL INDETERMINADO DE LA FORMA :
C                          F(X)=0
C     DONDE F, X Y 0 SON VECTORES DE DIMENSION M, N Y M RESPECTIVAMENTE
C     CON M<N.  DEBIDO A ESTO ES NECESARIO PROPONER UNA FUNCION OBJETIVO
C     Z=Z(X) A EXTREMIZAR, SIENDO Z UN NUMERO REAL.
C     EN RESUMEN, TENEMOS EL SIGUIENTE PROBLEMA DE PROGRAMACION MATEMA-
C     TICA:
C                          MIN  Z=Z(X)
C                           X
C     SUJETA A:            F(X)=0
C
      DIMENSION GZ(3),DELTX(3),X1(3),DELTX1(3),XMIN(3),D(3),AY(2)
      DIMENSION CERO(2),XYO(2),AUX(3),F(2)
      REAL MU(2),JAC(2,3),JACT(3,2),ATL(3)
      COMMON XO(3),M,N
      COMMON DSI(3)
      COMMON NORMA
      OPEN(3,FILE='#5:[illegible].DATA',STATUS='NEW')
      WRITE(*,'(A)')' SI DESEA IMPRESION, TECLEE 1'
      READ(*,'(I1)') I
      IF(I.EQ.1) THEN
        OPEN(6,FILE='PRINTER:')
      ELSE
        OPEN(6,FILE='CONSOLE:')
      ENDIF
      WRITE(6,50)
1     WRITE(6,650)
      READ(*,150) NCASO
      WRITE(6,150) NCASO
      IF(NCASO.GT.5.OR.NCASO.LE.1) GO TO 1
      WRITE(6,750)
      READ(*,150) NORMA
      WRITE(6,150) NORMA
      IF(NORMA.EQ.1) ALFA=1.0
      WRITE(6,100)
      READ(*,200) M,N
      WRITE(6,200) M,N
      TOLX=1.0E-5
      TOLF=0.0001
      PAMP=0.8
      MAX=200
      KMAX=100
      WRITE(6,350)
      WRITE(6,600) TOLX,TOLF,PAMP,MAX,KMAX
      WRITE(6,300)
      READ(*,400) (XO(I),I=1,N)
      WRITE(6,400) (XO(I),I=1,N)
      WRITE(3,510) (XO(I),I=1,N)
      MU(1)=0.10
      MU(2)=0.10
      CERO(1)=0.0
      CERO(2)=0.0
2     WRITE(6,450)
      READ(*,400) (AY(I),I=1,M)
      WRITE(6,400) (AY(I),I=1,M)
```

```
3     KONT=KONT+1
      IF(KONT.EQ.21) GO TO 17
      WRITE(*,'(A#)')
      CALL JACOBI(DY0)
      CALL GRAD(GZ)
      DO 4 I=1,N
            AUX(I)=GZ(I)
               DO 4 J=1,M
4     JACT(I,J)=JAC(J,I)
      CALL DECOMP(N,N,M,JACT,U)
      CALL SOLVE(N,N,M,JACT,U,AUX)
      DO 5 I=1,N
               DO 5 J=1,M
5     JACT(I,J)=JAC(J,I)
      CALL MULVEC(N,N,JACT,AUX,UTL)
      DO 6 I=1,N
6     CSI(I)=GZ(I)-UTL(I)
      IF(NORMA.EQ.2) ALFA=FMIN(-1.5,1.5,1.0E-6)
      DO 7 I=1,N
               DELTX(I)=-ALFA*CSI(I)
7     X1(I)=X0(I)+DELTX(I)
      IF(NEWO.EQ.S) GO TO 9
      CALL NEWAMP(NU,X1,XY,JACT,TOLX,TOLF,DAMP,ITER,MAX,KMAX)
      DO 8 I=1,N
8     MU(I)=AMOD(MU(I),6.283184)
      CALL MULVEC(N,M,JACT,MU,DELTX1)
      GO TO 10
9     CALL FUNF(F,X0,XY)
      DO 19 I=1,M
19    F(I)=-F(I)
      CALL SUBDEL(JAC,M,N,F,DELTX1,U)
10    DO 11 I=1,N
11    XFIN(I)=X1(I)+DELTX1(I)
      CALL FUNF(XY0,XFIN,CEF0)
      DO 12 I=1,M
            IF(ABS(XY(I)-XY0(I)).GT.1.0E-3) GO TO 13
12    CONTINUE
      GO TO 15
13    DO 14 I=1,N
14    X0(I)=XFIN(I)
      GO TO 3
15    DO 16 I=1,N
      XFIN(I)=AMOD(XFIN(I),6.283184)
16    X0(I)=XFIN(I)
      WRITE(6,200) KONT
      WRITE(6,250) (XY(I),I=1,M)
      WRITE(6,300) ALFA
      WRITE(6,505) (MU(I),I=1,N)
      WRITE(6,510) (XFIN(I),I=1,N)
      WRITE(3,510) (XFIN(I),I=1,N)
      WRITE(6,550)
      READ(*,'(I1)') IT
      WRITE(6,'(I1)') IT
      IF(IT.EQ.1) GO TO 2
      CLOSE(3)
      WRITE(6,'(A#)') ' *** FIN DEL PROBLEMA ***'
      STOP
17    WRITE(6,'(A)') ' NO SE LLEGA A LA SOLUCION DESPUES'
      WRITE(6,'(A)') '  DE  20 VECES'
      CLOSE(3)
      STOP
50    FORMAT(5X,' PROGRAMA PARA LA SOLUCION DE SISTEMAS DE ECUACIO'
     &'NES',/5X,'                 NO LINEALES SUBDETERMINADOS')
100   FORMAT(/X,' DAME LAS DIMENSIONES DE F(X) Y X (M Y N)')
150   FORMAT(I1)
```

A6

```
250   FORMAT(' PUNTO DEL ORGANO TERMINAL'/5X,3F10.3)
350   FORMAT(5X,' DAME LOS ANGULOS INICIALES DE LOS ESLABONES')
370   FORMAT(' TOLX, TOLF, DAMP, MAX Y KMAX')
400   FORMAT(5F10.5)
450   FORMAT(' DAME LAS COORDENADAS DEL PUNTO DEL ORGANO TERMINAL')
500   FORMAT(3X,'ALFA:'/5X,F10.2)
550   FORMAT(3X,'NU:'/5X,3F10.3///)
600   FORMAT(3X,'THETA(I)'/5X,3F10.3)
610   FORMAT(3F15.8)
650   FORMAT(//' TECLEE 1 SI DESEA DAR OTRO PUNTO DE LA TRAYECTORIA')
700   FORMAT(2E15.5,F5.3,2I5)
750   FORMAT(//' INDIQUE EL NUMERO DE CASO')
710   FORMAT(2F15.5,F5.3,2I5)
800   FORMAT(' ¿ES UN PROBLEMA DE NORMA CUADRATICA (1 SI; 2 NO)?')
900   FORMAT(///' SOLUCION DEL PROBLEMA DESPUES DE ',I3,' VECES')
      END
C
C
      SUBROUTINE JACOBI(JAC)
C
C     SUBRUTINA QUE CALCULA EL JACOBIANO DE LA FUNCION F(X) EMPLEADA
C     EN EL PROGRAMA GANLSQ.
C
C     ENTRADAS:
C             M        = DIMENSION DE F(X)
C             N        = DIMENSION DE X
C             XO       = VALOR INICIAL
C     SALIDAS:
C             JAC      = JACOBIANO DE F(X)
C
      REAL JAC(2,3)
      COMMON XO(3),M,N
      DO 1 I=1,M
            DO 1 J=1,N
1     JAC(I,J)=0.0
      JAC(1,3)=-SIN(XO(1)+XO(2)+XO(3))
      JAC(1,2)=JAC(1,3)-SIN(XO(1)+XO(2))
      JAC(1,1)=JAC(1,2)-SIN(XO(1))
      JAC(2,3)=COS(XO(1)+XO(2)+XO(3))
      JAC(2,2)=JAC(2,3)+COS(XO(1)+XO(2))
      JAC(2,1)=JAC(2,2)+COS(XO(1))
      RETURN
      END
C
C
      SUBROUTINE GRAD(GZ)
C
C     SUBRUTINA QUE CALCULA EL GRADIENTE DE LA FUNCION Z(X) EMPLEADA
C     EN EL PROGRAMA GANLSQ.
C
C     ENTRADAS:
C             N        = DIMENSION DE Z(X)
C             XO       = PUNTO INICIAL
C     SALIDAS:
C             GZ       = GRADIENTE DE Z(X)
C
      DIMENSION GZ(3)
      COMMON XO(3),M,N
      COMMON NORMA
      DO 1 I=1,N
1     GZ(I)=0.0
      DO 2 I=1,N
      IF(NORMA.EQ.2) GZ(I)=1.0
2     GZ(I)=XO(I)
```

```
C
C
      SUBROUTINE FUNC(F,XO,X)
      DIMENSION F(2),X(2),XO(3)
      Y1=XO(1)
      Y2=Y1+XO(2)
      Y3=Y2+XO(3)
      F(1)=COS(Y1)+COS(Y2)+COS(Y3)-X(1)
      F(2)=SIN(Y1)+SIN(Y2)+SIN(Y3)-X(2)
      RETURN
      END



      SUBROUTINE CHODCL(A,M,N,B,X,D)
      INTEGER M,N
      DIMENSION A(N,M),B(M),X(N),D(N),AT(3,2),CM(3,2)
C
      DO 10 I=1,N
            DO 5 J=1,M
                  AT(I,J)=A(J,I)
                  CM(I,J)=A(J,I)
5           CONTINUE
10    CONTINUE
      CALL MULMAT(M,A,M,AT,N)
      DO 20 I=1,M
            DO 15 J=1,N
                  A(I,J)=AT(J,I)
15          CONTINUE
20    CONTINUE
C
C     SUSTITUSION HACIA ATRAS
C
      X(1)=B(1)/A(1,1)
      DO 30 I=2,M
            CC=0.0
            DO 25 J=1,I-1
                  CC=A(I,J)*X(J)+CC
25          CONTINUE
            X(I)=B(I)-CC
            X(I)=X(I)/A(I,I)
30    CONTINUE
C
C     SUSTITUCION HACIA ADELANTE
C
      B(M)=X(M)/AT(M,M)
      DO 40 I=M-1,1,-1
            CC=0.0
            DO 35 J=I+1,M
                  CC=AT(I,J)*B(J)+CC
35          CONTINUE
            B(I)=X(I)-CC
            B(I)=B(I)/AT(I,I)
40    CONTINUE
      DO 50 I=1,N
            X(I)=0.0
            DO 45 J=1,M
                  X(I)=CM(I,J)*B(J)+X(I)
45          CONTINUE
50    CONTINUE
      RETURN
```

AD

```
      SUBROUTINE MULVEC(L1,L2,A,B,C)
C
C     ESTA SUBRUTINA CALCULA [illegible]
C
C     ENTRADAS:
C           L1      = NUMERO DE RENGLONES DE A Y C
C           L2      = NUMERO DE COLUMNAS DE A Y [illegible] DE B
C           A       = MATRIZ QUE [illegible]
C           B       = VECTOR QUE [illegible]
C     SALIDAS:
C           C       = PRODUCTO DE A CON B
C
      DIMENSION A(L1,L2),B(L2),C(L1)
      DO 1 I=1,L1
    1 C(I)=0.0
      DO 2 I=1,L1
            DO 2 K=1,L2
    2 C(I)=A(I,K)*B(K)+C(I)
      RETURN
      END
C
C
      SUBROUTINE DECOMP(MDIM,M,N,A,U)
      INTEGER MDIM,M,N
      REAL A(MDIM,N),U(MDIM)
      REAL ALPHA,BETA,GAMMA
C
C     REDUCCION DE HOUSEHOLDER DE UNA MATRIZ RECTANGULAR A SU FORMA
C     TRIANGULAR SUPERIOR.
C
C     MDIM    DIMENSION DE [illegible] DE A
C     M       NUMERO DE RENGLONES DE A
C     N       NUMERO DE COLUMNAS DE A
C     A       MATRIZ DE MXN CON M MAYOR QUE N
C
C     ENTRADAS:
C             MATRIZ A REDUCIR
C     SALIDAS:
C             MATRIZ REDUCIDA E INFORMACION DE LA REDUCCION
C             U = VECTOR DE DIMENSION M
C
C     ENTRADA: IGNORADA
C     SALIDA: INFORMACION DE LA REDUCCION
C
C     ENCUENTRA REFLEXIONES QUE ANULAN A(I,K),I=K+1,...,M
C
      DO 6 K=1,N
            ALPHA=0.0
            DO 1 I=K,M
                  U(I)=A(I,K)
                  ALPHA=ALPHA+U(I)*U(I)
    1       CONTINUE
            [illegible]
            [illegible]
            [illegible]
            [illegible]
            A(K,K)=[illegible]
            [illegible]
C
C     [illegible]
C
            [illegible]
            DO 5 J=[illegible]
                  [illegible]
```

```
                  GAMMA=GAMMA+U(I)*A(I,J)
2              CONTINUE
               GAMMA=GAMMA/BETA
               DO 3 I=K,M
                  A(I,J)=A(I,J)-GAMMA*U(I)
3              CONTINUE
4           CONTINUE
4     CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SOLVE(MDIM,M,N,A,U,B)
      INTEGER MDIM,M,N
      REAL A(MDIM,N),U(M),B(M)
      REAL BETA,GAMMA,T
C
C     SOLUCION DE MINIMOS CUADRADOS DE UN SISTEMA SOBREDETERMINADO
C     ENCUENTRA X TAL QUE MINIMIZA LA NORMA AX-B.
C
C     MDIM,M,N,A,U SON LOS RESULTADOS DE HECOMP
C     B             VECTOR DE DIMENSION M
C
C     ENTRADAS:
C           LADO DERECHO DE LA ECUACION
C     SALIDAS:
C           PRIMERAS N COMPONENTES IGUAL A LA SOLUCION
C           ULTIMAS M-N COMPONENTES IGUAL A LA TRANSF. RESIDUAL
C
C     DIVISION ENTRE CERO IMPLICA QUE NO EXISTE RANGO COMPLETO
C
      DO 3 K=1,N
         T=A(K,K)
         BETA=-U(K)*A(K,K)
         A(K,K)=U(K)
         GAMMA=0.0
         DO 1 I=K,M
            GAMMA=GAMMA+A(I,K)*B(I)
1     CONTINUE
         GAMMA=GAMMA/BETA
         DO 2 I=K,M
         B(I)=B(I)-GAMMA*A(I,K)
2        CONTINUE
         A(K,K)=T
3     CONTINUE
C
C     SUSTITUCION HACIA ATRAS
C
      DO 5 KB=1,N
         K=N+1-KB
         B(K)=B(K)/A(K,K)
         IF(K.EQ.1) GO TO 5
         KM1=K-1
         DO 4 I=1,KM1
            B(I)=B(I)-A(I,K)*B(K)
4        CONTINUE
5     CONTINUE
      RETURN
      END
C
C
      SUBROUTINE FUNC(X,F,JACT,P,XY)
      COMMON XS(3),K,M
      DIMENSION Y(3),P(3),F(2),X(2),XY(2)
```

```
      IJ=I+1
      CALL MULVEC(L,M,JACT,X,JTMU)
      DO 1 I=1,N
    1 Y(I+1)=P(I)+JTMU(I)+Y(I)
      F(1)=COS(Y(2))+COS(Y(3))+COS(Y(4))-XY(1)
      F(2)=SIN(Y(2))+SIN(Y(3))+SIN(Y(4))-XY(2)
      RETURN
      END
C
C
      SUBROUTINE DFDX(X,DF,JACT,P)
      COMMON XO(3),M,N
      DIMENSION X(2),P(3),DF(2,2)
      REAL JACT(3,2),JTMU(3)
      CALL MULVEC(N,M,JACT,X,JTMU)
      A1=P(1)+JTMU(1)
      A2=A1+P(2)+JTMU(2)
      A3=A2+P(3)+JTMU(3)
      A4=JACT(1,1)+JACT(2,1)
      A5=A4+JACT(3,1)
      A6=JACT(2,1)+JACT(2,2)
      A7=A6+JACT(3,2)
      DF(1,1)=-(JACT(1,1)*SIN(A1)+A4*SIN(A2)+A5*SIN(A3))
      DF(1,2)=0.0
      DF(2,1)=0.0
      DF(2,2)=JACT(2,1)*COS(A1)+A6*COS(A2)+A7*COS(A3)
      RETURN
      END
C
C
      SUBROUTINE NRDAMP (X,F,XY,JACT,TOLX,TOLF,DAMP,ITER,MAX,KMAX)
      REAL X(2),P(3),XY(2),JACT(3,2),F(2),DF(2,2),DELTA(2),WORK(2)
      INTEGER IP(2)
      COMMON XO(3),N,M
C
C     ESTA SUBRUTINA ENCUENTRA LAS RAICES DE UN SISTEMA ALGEBRAICO
C     NO-LINEAL DE ORDEN N, POR EL METODO DE NEWTON RAPHSON CON
C     AMORTIGUAMIENTO
C
      KONT=1
      ITER=0
      CALL FUN(X,F,JACT,P,XY)
      FNOR1=FNORM(F,N)
      IF(FNOR1.LT.TOLF) GO TO 4
    1 CALL DFDX(X,DF,JACT,P)
      CALL DECOMP(N,N,DF,COND,IP,WORK)
      K=0
C
C     SI LA MATRIZ JACOBIANA ES SINGULAR, LA SUBRUTINA REGRESA AL
C     PROGRAMA PRINCIPAL.  EN CASO CONTRARIO, SIGUE EL PROCESO.
C
      CONDP=COND+1
      IF(CONDP.EQ.COND) GO TO 14
      CALL SOLVE(N,N,DF,F,IP)
      DO 2 I=1,N
    2 DELTA(I)=F(I)
      DELNOR=FNORM(DELTA,N)
      IF(DELNOR.LT.TOLX) GO TO 4
      DO 3 I=1,N
    3 X(I)=X(I)-DELTA(I)
      GO TO 5
    4 FNOR2=FNOR1
      GO TO 6
    5 CALL FUN(X,F,JACT,P,XY)
      KONT=KONT+1
```

A11

```
C
C     PRUEBA DE LA NORMA DE LA FUNCION.  SI NO DECRECE, SE INTRODUCE
C     EL AMORTIGUAMIENTO.
C
      IF(FNOR2.LT.FNOR1) GO TO 10
      IF(K.EQ.KMAX) GO TO 16
      K=K+1
      DO 8 I=1,N
         IF(K.GE.2) GO TO 7
         DELTA(I)=(DAMP-1.)*DELTA(I)
         GO TO 8
7        DELTA(I)=DAMP*DELTA(I)
8     CONTINUE
      DELNOR=FNORM(DELTA,N)
      IF(DELNOR.LT.TOLX) GO TO 16
      DO 9 I=1,N
9     X(I)=X(I)+DELTA(I)
      GO TO 5
10    IF(ITER.GT.MAX) GO TO 16
      ITER=ITER+1
      FNOR1=FNOR2
      GO TO 1
11    RETURN
12    DO 13 I=1,N
13    WRITE(6,120) I,X(I)
      RETURN
14    WRITE(6,130) ITER,KONT
      GO TO 12
16    RETURN
C     WRITE(6,140) ITER,FNOR2,KONT,COND
C     GO TO 12
C
120   FORMAT(5X,'X(',I3,') = ',E20.6)
130   FORMAT(5X,'EN LA ITERACION NUM. ',I3,' LA MATRIZ JACOBIANA '
     -      'ES SINGULAR.'/5X,'LA FUNCION SE EVALUO ',I3,' VECES'/
     -      5X,'EL VALOR ACTUAL DE X ES: '//)
C140   FORMAT(10X,'EL PROCESO DIVERGE EN LA ITERACION NUM. ',I3/10X,
C    -      'LA NORMA DE LA FUNCION ES: ',E20.6/10X,'LA FUNCION FUE'
C    -      ' EVALUADA ',I3,' VECES'/10X,'LA CONDICION DE LA MATRIZ'
C    -      ' ES: ',E20.6/10X,'EL VALOR ACTUAL DE X ES: '//)
      END
C
C
      FUNCTION FNORM(F,N)
      REAL F(N)
      I=1
      DO 1 J=2,N
1     IF(ABS(F(I)).LE.ABS(F(J))) I=J
      FNORM=ABS(F(I))
      RETURN
      END
C
C
      SUBROUTINE DECOMP(NDIM,N,A,COND,IPVT,WORK)
      REAL A(NDIM,N),COND,WORK(N)
      INTEGER IPVT(N)
C
C     DESCOMPONE UNA MATRIZ REAL POR ELIMINACION GAUSSIANA Y ESTIMA
C     LA CONDICION DE LA MATRIZ.
C     EMPLEA LA SUBRUTINA SOLVE PARA CALCULAR LA SOLUCION DE SISTEMAS
C     LINEALES.
C
C     ENTRADAS:
C             NDIM    = DIMENSION DE RENGLONES DEL ARREGLO QUE CONTIENE A
C             N       = ORDEN DE LA MATRIZ
```

```
C     SALIDA
C        A     = CONTIENE UNA MATRIZ TRIANGULAR SUPERIOR U Y UNA
C                VERSION PERMUTADA DE UNA MATRIZ TRIANGULAR INFE-
C                RIOR I-L DE TAL MANERA QUE (MATRIZ PERMUTADA)*A      412
C                =L*U
C        COND  = CONDICION DE A. PARA EL SISTEMA LINEAL A*X=B,
C                LOS CAMBIOS DE A Y B PUEDEN OCASIONAR CAMBIOS
C                CONDICION ES IGUAL A COND+1, A ES SINGULAR PARA
C                LA PRECISION DE TRABAJO.  COND=1.0E+32 SI SE DE-
C                TECTA LA SINGULARIDAD.
C        IPVT  = VECTOR PIVOTE.
C                IPVT(K)=INDICE DEL RENGLON PIVOTE K
C                IPVT(N)=(-1)**(NUM. DE INTERCAMBIOS)
C        WORK  = ESPACIO DE TRABAJO.  EL VECTOR DE TRABAJO DEBE
C                SER DECLARADO E INCLUIDO EN EL LLAMADO.
C
C     EL DETERMINANTE DE A PUEDE SER OBTENIDO EN LA SALIDA D
C        DET(A)=IPVT(N)*A(1,1)*A(2,2)*...*A(N,N)
C
      IPVT(N)=1
      IF(N.EQ.1) GO TO 80
      NM1=N-1
C
C     CALCULA 1-NORMA DE A
C
      ANORM=0.0
      DO 10 J=1,N
         T=0.0
         DO 5 I=1,N
            T=T+ABS(A(I,J))
    5    CONTINUE
         IF(T.GT.ANORM) ANORM=T
   10 CONTINUE
C
C     ELIMINACION GAUSSIANA CON PIVOTEO PARCIAL
C
      DO 35 K=1,NM1
         KP1=K+1
C
C     ENCUENTRA EL PIVOTE
C
         M=K
         DO 15 I=KP1,N
            IF(ABS(A(I,K)).GT.ABS(A(M,K))) M=I
   15    CONTINUE
         IPVT(K)=M
C
C     SE SALTA SI EL PIVOTE ES CERO
C
         IF(M.NE.K) IPVT(N)=-IPVT(N)
         T=A(M,K)
         A(M,K)=A(K,K)
         A(K,K)=T
         IF(T.EQ.0.0) GO TO 35
         DO 20 I=KP1,N
            A(I,K)=-A(I,K)/T
   20    CONTINUE
C
C     INTERCAMBIO Y ELIMINACION POR COLUMNAS
C
         DO 30 J=KP1,N
            T=A(M,J)
            A(M,J)=A(K,J)
            A(K,J)=T
            IF(T.EQ.0.0) GO TO 30
```

```
                  A(I,J)=A(I,J)+T*A(I,K)
25             CONTINUE
30          CONTINUE
35    CONTINUE
C
C     COND=(1-NORMA DE A)*(UNA ESTIMACION DE 1-NORMA DE A-INVERSA)
C     ESTIMACION OBTENIDA POR UN PASO DE ITERACION INVERSA PARA EL
C     VECTOR SINGULAR PEQUENO.  ESTO INVOLUCRA SOLUCION EN DOS SIS-
C     TEMAS DE ECUACIONES, (A-TRANSPUESTA)*Y=E Y A*Z=Y DONDE E ES
C     EL VECTOR DE +1 O -1 ESCOGIDO PARA PROVOCAR CRECIMIENTO EN Y.
C     ESTIMACION=(1-NORMA DE Z)/(1-NORMA DE Y)
C
C     RESUELVE (A-TRANSPUESTA)*Y=E
C
      DO 50 K=1,N
         T=0.0
         IF(K.EQ.1) GO TO 45
         KM1=K-1
         DO 40 I=1,KM1
            T=T+A(I,K)*WORK(I)
40       CONTINUE
45       EK=1.0
         IF(T.LT.0.0) EK=-1.0
         IF(A(K,K).EQ.0.0) GO TO 90
         WORK(K)=-(EK+T)/A(K,K)
50    CONTINUE
      DO 60 KB=1,NM1
         K=N-KB
         T=0.0
         KP1=K+1
         DO 55 I=KP1,N
            T=T+A(I,K)*WORK(K)
55       CONTINUE
         WORK(K)=T
         M=IPVT(K)
         IF(M.EQ.K) GO TO 60
         T=WORK(M)
         WORK(M)=WORK(K)
         WORK(K)=T
60    CONTINUE
      YNORM=0.0
      DO 65 I=1,N
         YNORM=YNORM+ABS(WORK(I))
65    CONTINUE
C
C     RESUELVE A*Z=Y
C
      CALL SOLVE(NDIM,N,A,WORK,IPVT)
      ZNORM=0.0
      DO 70 I=1,N
         ZNORM=ZNORM+ABS(WORK(I))
70    CONTINUE
C
C     ESTIMA LA CONDICION
C
      COND=ANORM*ZNORM/YNORM
      IF(COND.LT.1.0) COND=1.0
      RETURN
C
C     UNO POR UNO
C
80    COND=1.0
      IF(A(1,1).NE.0.0) RETURN
C
C     SINGULARIDAD EXACTA
```

```
      [illegible]
      RETURN
      END
C
C
      SUBROUTINE SOLVE(NDIM,N,A,B,IPVT)
      INTEGER NDIM,N,IPVT(N)
      REAL A(NDIM,N),B(N)
C
C     SOLUCION DEL SISTEMA LINEAL A*X=B
C     NO SE USE SI SE DETECTA SINGULARIDAD EN DECOMP
C
C     ENTRADAS:
C             NDIM    = DIMENSION DE RENGLONES DECLARADA DEL ARREGLO A
C             N       = ORDEN DE LA MATRIZ
C             A       = MATRIZ TRIANGULARIZADA OBTENIDA EN DECOMP
C             B       = VECTOR DEL LADO DERECHO
C             IPVT    = VECTOR PIVOTE OBTENIDO DE DECOMP
C     SALIDAS:
C             B       = VECTOR SOLUCION, X
C
C     ELIMINACION HACIA ADELANTE
C
      IF(N.EQ.1) GO TO 50
      NM1=N-1
      DO 20 K=1,NM1
            KP1=K+1
            M=IPVT(K)
            T=B(M)
            B(M)=B(K)
            B(K)=T
            DO 10 I=KP1,N
                  B(I)=B(I)+A(I,K)*T
10          CONTINUE
20    CONTINUE
C
C     SUSTITUCION HACIA ATRAS
C
      DO 40 KB=1,NM1
            KM1=N-KB
            K=KM1+1
            B(K)=B(K)/A(K,K)
            T=-B(K)
            DO 30 I=1,KM1
                  B(I)=B(I)+A(I,K)*T
30          CONTINUE
40    CONTINUE
50    B(1)=B(1)/A(1,1)
      RETURN
      END




      REAL FUNCTION FMIN(AX,BX,TOL)
      REAL AX,BX,TOL
C
C     SE DETERMINA UNA APROXIMACION AL PUNTO DONDE F TIENDE A UN MINIMO
C     EN EL INTERVALO (AX,BX).
C     ENTRADAS:
C             AX      = LIMITE IZQUIERDO DEL INTERVALO INICIAL
C             BX      = LIMITE DERECHO DEL INTERVALO INICIAL
```

```
C                          TOL ES LA TOLERANCIA FINAL (TOL.GE.0.0)
C          F          = SUBPROGRAMA QUE EVALUA LA FUNCION F(X) PARA CUAL-
C                          QUIER X EN EL INTERVALO (AX,BX)
C      SALIDAS:                                                    A15
C          FMIN       = APROXIMACION A LA ABSCISA DONDE F TIENDE A UN
C                          MINIMO
C
C      EL METODO USADO ES UNA COMBINACION DEL METODO DE LAS SECCIONES DO-
C      RADAS Y LA INTERPOLACION DE SUCESIONES PARABOLICAS SIENDO LA CON-
C      VERGENCIA MAS RAPIDA QUE EN EL METODO DE FIBONACCI.  SI F TIENE
C      UNA SEGUNDA DERIVADA CONTINUA Y POSITIVA EN EL MINIMO (QUE NO ESTE
C      EN AX O BX) LA CONVERGENCIA ES SUPERLINEAL, Y USUALMENTE DEL ORDEN
C      DE 1.324....
C      LA FUNCION F NUNCA SE EVALUA EN DOS PUNTOS SEPARADOS POR UNA DIS-
C      TANCIA MENOR A EPS*ABS(FMIN)+(TOL/3.0), DONDE EPS ES APROXIMADA-
C      MENTE LA RAIZ CUADRADA DE LA PRECISION RELATIVA DE LA MAQUINA.  SI
C      F ES UNA FUNCION UNIMODAL Y LOS CALCULOS DE F SON SIEMPRE UNIMODA-
C      LES CUANDO ESTAN SEPARADOS CUANDO MENOS SQRT(EPS)*ABS(X)+(TOL/3.0), EN-
C      TONCES FMIN APROXIMA LA ABSCISA DEL MINIMO GLOBAL DE F EN EL IN-
C      TERVALO AX,BX CON UN ERROR MENOR QUE APROXIMAR A UN LOCAL, PERO
C      PROBABLEMENTE NO GLOBAL, MINIMO PARA LA MISMA PRECISION.
C      ESTE SUBPROGRAMA ES UNA VERSION LIGERAMENTE MODIFICADA DEL ALGOL
C      60 PROCEDURE LOCALMIN DADO EN RICHARD BRENT, ALGORITHMS FOR MINIMI-
C      ZATION WITHOUT DERIVATIVES, PRENTICE-HALL, INC. (1973).
C
      REAL A,B,C,D,E,EPS,XM,P,Q,R,TOL1,TOL2,U,V,W
      REAL FU,FV,FW,FX,X
C
C     C ES LA RAIZ INVERSA DE LA RAIZ DE LA RAZON DORADA
C
      C=0.5*(3.0-SQRT(5.0))
C
C     EPS ES APROXIMADAMENTE LA RAIZ CUADRADA DE LA PRECISION RELATIVA
C     DE LA MAQUINA
C
      EPS=1.0
   10 EPS=EPS/2.0
      TOL1=1.0+EPS
      IF(TOL1.GT.1.0) GO TO 10
      EPS=SQRT(EPS)
C
C     INICIALIZACION
C
      A=AX
      B=BX
      V=A+C*(B-A)
      W=V
      X=V
      E=0.0
      FX=F(X)
      FV=FX
      FW=FX
C
C     LA ITERACION PRINCIPAL EMPIEZA AQUI
C
   20 XM=0.5*(A+B)
      TOL1=EPS*ABS(X)+TOL/3.0
      TOL2=2.0*TOL1
C
C     CHECA EL CRITERIO DE PARO
C
      IF(ABS(X-XM).LE.(TOL2-0.5*(B-A))) GO TO 90
C
C     ES NECESARIA UNA SECCION DORADA?
C
```

```
C
C          AJUSTA LA PARABOLA
C
           R=(X-W)*(FX-FV)
           Q=(X-V)*(FX-FW)
           P=(X-V)*Q-(X-W)*R
           Q=2.0*(Q-R)
           IF(Q.GT.0.0) P=-P
           Q=ABS(Q)
           R=E
           E=D
C
C          ES ACEPTABLE LA PARABOLA?
C
30         IF(ABS(P).GE.ABS(0.5*Q*R)) GO TO 40
           IF(P.LE.Q*(A-X)) GO TO 40
           IF(P.GE.Q*(B-X)) GO TO 40
C
C          UN PASO DE INTERPOLACION PARABOLICA
C
           D=P/Q
           U=X+D
C
C          F NO DEBE SER EVALUADA CERCA DE AX O BX
C
           IF((U-A).LT.TOL2) D=SIGN(TOL1,XM-X)
           IF((B-U).LT.TOL2) D=SIGN(TOL1,XM-X)
           GO TO 50
C
C          UN PASO DE SECCION DORADA
C
40         IF(X.GE.XM) E=A-X
           IF(X.LT.XM) E=B-X
           D=C*E
C
C          F NO DEBE SER EVALUADA MUY CERCA DE X
C
50         IF(ABS(D).GE.TOL1) U=X+D
           IF(ABS(D).LT.TOL1) U=X+SIGN(TOL1,D)
           FU=F(U)
C
C          ALMACENA A,B,V,W Y X
C
           IF(FU.GT.FX) GO TO 60
           IF(U.GE.X) A=X
           IF(U.LT.X) B=X
           V=W
           FV=FW
           W=X
           FW=FX
           X=U
           FX=FU
           GO TO 20
60         IF(U.LT.X) A=U
           IF(U.GE.X) B=U
           IF(FU.LE.FW) GO TO 70
           IF(W.EQ.X) GO TO 70
           IF(FU.LE.FV) GO TO 80
           IF(V.EQ.X) GO TO 80
           IF(V.EQ.W) GO TO 80
           GO TO 20
70         V=W
           FV=FW
           W=U
           FW=FU
           GO TO 20
```

```
10      X=U
        FX=FU
        GO TO 20
C
C       TERMINA LA ITERACION PRINCIPAL
C
90      FMIN=X
        RETURN
        END
C
C
        SUBROUTINE FUNZ(ALFA,Z)
C
C       CALCULA LA FUNCION Z(X0-ALFA*CSI)
C
        COMMON X0(3),M,N
        COMMON CSI(3)
        COMMON NORMA
        Z=0.0
        IF(NORMA.EQ.2) GO TO 2
        DO 1 I=1,N
1       Z=Z+(X0(I)-ALFA*CSI(I))**2
        Z=0.5*Z
        RETURN
2       DO 3 I=1,N
3       Z=Z+ABS(X0(I)-ALFA*CSI(I))
        RETURN
        END
C
C
        REAL FUNCTION F(ALFA)
        REAL ALFA
        COMMON X0(3),M,N
        COMMON CSI(3)
        COMMON NORMA
        CALL FUNZ(ALFA,Z)
        F=Z
        RETURN
        END




1UCGO TURTLEGRAFICOS
AUCOS DE1T1
        PROGRAM GRAMON
C
C       PROGRAMA PARA GRAFICACION DE UN MANIPULADOR
C
        DIMENSION T(3)
        INTEGER N
        OPEN(3,FILE='RS:MANG.C.DATA')
        CALL INITTU
        N=1
0       IF(N.GT.5) N=1
        CALL PCOLOR(3)
        CALL MOVETO(140,50)
        CALL TURNTO(0)
        READ(3,100) (T(I),I=1,3)
        A1=50.0*COS(T(1))
        B1=50.0*SIN(T(1))
```

```
      A3=50.0*COS(T(3))
      B3=50.0*SIN(T(3))
      MT1=IFIX(57.2958*T(1))
      MT2=IFIX(57.2958*T(2))
      MT3=IFIX(57.2958*T(3))
      A1=SQRT(A1**2+B1**2)
      A2=SQRT(A2**2+B2**2)
      A3=SQRT(A3**2+B3**2)
      MA1=IFIX(A1)
      MA2=IFIX(A2)
      MA3=IFIX(A3)
      GO TO (1,2,3,4,5,6),N
1     CALL PENCOL(1)
      GO TO 7
2     CALL PENCOL(6)
      GO TO 7
3     CALL PENCOL(7)
      GO TO 7
4     CALL PENCOL(8)
      GO TO 7
5     CALL PENCOL(10)
      GO TO 7
6     CALL PENCOL(11)
7     CALL TURN(MT1)
      CALL MOVE(MA1)
      CALL TURN(MT2)
      CALL MOVE(MA2)
      CALL TURN(MT3)
      CALL MOVE(MA3)
      READ(X,'(I1)') N
      IF(N.GE.1) GO TO 8
      CLOSE(2)
      IF(N.LT.0) CALL CORERT
100   FORMAT(3F15.3)
      STOP
      END
```

Ejemplo 3

```
$USES USUSDPL IN #5:SUBDPL.CODE
$USES UFMIN IN #5:FMIN.CODE
      PROGRAM SANLSU
C
C           PROGRAM SANLSU
C
C     ESTE PROGRAMA CALCULA LA SOLUCION DE UN SISTEMA ALGEBRAICO
C     NO LINEAL SUBDETERMINADO DE LA FORMA :
C                       F(X)=0
C     DONDE F, X Y 0 SON VECTORES DE DIMENSION M, N Y M RESPECTIVAMENTE
C     CON M<N.  DEBIDO A ESTO ES NECESARIO PROPONER UNA FUNCION OBJETIVO
C     Z=Z(X) A EXTREMIZAR, SIENDO Z UN NUMERO REAL..
C     EN RESUMEN, TENEMOS EL SIGUIENTE PROBLEMA DE PROGRAMACION MATEMA-
C     TICA:
C                       MIN Z=Z(X)
C                        X
C     SUJETA A:         F(X)=0
C
      DIMENSION BZ(5),DELTX(5),X1(5),DELTX1(5),XFIN(5),U(5),XY(3)
      DIMENSION CERO(3),XYO(3),AUX(5),F(3),P(20),D(36),X(30)
      REAL JAC(3,5),JACT(5,3),JTL(5)
      COMMON XO(5),M,N
      COMMON CSI(5)
      COMMON NORMA
      OPEN(3,FILE='#5:MANPUL.DATA',STATUS='NEW')
      WRITE(*,'(A$)')' SI DESEA IMPRESORA TECLEE 1 '
      READ(*,'(I1)') I
      IF(I.EQ.1) THEN
         OPEN(6,FILE='PRINTER:')
      ELSE
         OPEN(6,FILE='CONSOLE:')
      ENDIF
      WRITE(6,50)
1     WRITE(6,350)
      WRITE(6,750)
      READ(*,150) NORMA
      WRITE(6,150) NORMA
      IF(NORMA.EQ.1) ALFA=1.0
      WRITE(6,100)
      READ(*,200) M,N
      WRITE(6,200) M,N
      WRITE(6,350)
      READ(*,700) (P(I),I=1,15)
      WRITE(6,700) (P(I),I=1,15)
      RAD=3.14159265358979/180.0
      WRITE(6,300)
      READ(*,400) (XO(I),I=1,N)
      WRITE(6,400) (XO(I),I=1,N)
      DO 21 I=1,N
            XO(I)=RAD*XO(I)
            P(I+10)=RAD*P(I+10)
            P(I+15)=SIN(P(I+10))
            P(I+10)=COS(P(I+10))
21    CONTINUE
      WRITE(3,515) ((P(I),I=1,20)
      CERO(1)=0.0
      CERO(2)=0.0
      CERO(3)=0.0
2     WRITE(6,450)
      READ(*,400) (XY(I),I=1,M)
      WRITE(6,400) (XY(I),I=1,M)
```

```
3       KONT=KONT+1
        IF(KONT.EQ.11) GO TO 17
        WRITE(*,'(A)') '.'
        CALL FUNF(F,XY,P,D,X,XY)
        CALL JACOBI(X,JAC)
        CALL GRAD(GZ)
        DO 4 I=1,M
              AUX(I)=GZ(I)
              DO 4 J=1,M
4       JACT(I,J)=JAC(J,I)
        CALL HECOMP(N,N,M,JACT,U)
        CALL HOLVE(N,N,M,JACT,U,AUX)
        DO 5 I=1,N
              DO 5 J=1,M
5       JACT(I,J)=JAC(J,I)
        CALL MULVEC(N,M,JACT,AUX,JTL)
        DO 6 I=1,N
6       CSI(I)=GZ(I)-JTL(I)
        IF(NORMA.EQ.2) ALFA=FMIN(-2.5,2.5,1.0E-6)
        DO 7 I=1,N
              DELTX(I)=-ALFA*CSI(I)
7       X1(I)=X0(I)+DELTX(I)
        DO 19 I=1,M
19      F(I)=-F(I)
        CALL SUBDEL(JAC,M,N,F,DELTX1,U)
        DO 11 I=1,N
11      XFIN(I)=X1(I)+DELTX1(I)
        CALL FUNF(XYC,XFIN,F,D,X,CERO)
        DO 12 I=1,M
              IF(ABS(XY(I)-XYO(I)).GT.1.0E-1) GO TO 13
12      CONTINUE
        GO TO 15
13      DO 14 I=1,N
14      X0(I)=XFIN(I)
        GO TO 3
15      DO 16 I=1,N
        XFIN(I)=AMOD(XFIN(I),6.283185307179586)
16      X0(I)=XFIN(I)
        WRITE(6,800) KONT
        WRITE(6,250) (XY(I),I=1,M)
        WRITE(6,500) ALFA
        WRITE(3,515) (XFIN(I),I=1,N)
        DO 22 I=1,N
22      XFIN(I)=XFIN(I)/RAD
        WRITE(6,510) (XFIN(I),I=1,N)
        WRITE(6,550)
        READ(*,'(I1)') IT
        WRITE(6,'(I1)') IT
        IF(IT.EQ.1) GO TO 2
        CLOSE(3)
        WRITE(6,'(A)') ' ***  FIN DEL PROBLEMA ***'
        STOP
17      WRITE(6,'(A)') ' NO SE LLEGA A LA SOLUCION DESPUES'
        WRITE(6,'(A)') ' DE  10  VECES  '
        CLOSE(3)
        STOP
50      FORMAT(3X,' PROGRAMA PARA LA SOLUCION DE SISTEMAS DE ECUACIO'
     &  'NES'/5X,'             NO LINEALES SUBDETERMINADOS')
100     FORMAT(3X,' DAME LAS DIMENSIONES DE F(X) Y X (M Y N)')
150     FORMAT(I1)
200     FORMAT(2I2)
250     FORMAT(/' PUNTO DEL ORGANO TERMINAL'/3X,5F15.5)
300     FORMAT(3X,' DAME LOS ANGULOS INICIALES DE LOS ESLABONES')
350     FORMAT(' TOLX, TOLF, DAMP, MAX Y KMAX')
400     FORMAT(5F10.5)
```

```
 500      FORMAT(3X,'ALFA='/5X,F10.2)
 510      FORMAT(3X,'THETA(I)'/5X,5F10.3)
 515      FORMAT(5F15.3)
 550      FORMAT(//' TECLEE 1 SI DESEA DAR OTRO PUNTO DE LA TRAYECTORIA')
 600      FORMAT(2E15.5,F5.3,2I5)
 650      FORMAT(//' CASO 3')
 700      FORMAT(2F15.5,F5.3,2I5)
 750      FORMAT(' ?ES UN PROBLEMA DE NORMA CUADRATICA (1 SI; 2 NO)?')
 800      FORMAT(///' SOLUCION DEL PROBLEMA DESPUES DE ',I3,' VECES')
 850      FORMAT(' DAME LOS PARAMETROS DEL MECANISMO (A, B, ALFA)')
 900      FORMAT(15F5.1)
          END
C
C
          SUBROUTINE JACOBI(X,DF)
C
C         SUBRUTINA QUE CALCULA EL JACOBIANO DE LA FUNCION F(X) EMPLEADA
C         EN EL PROGRAMA SANLSU.
C
C         ENTRADAS:
C                  M        = DIMENSION DE F(X)
C                  N        = DIMENSION DE X
C                  X0       = VALOR INICIAL
C         SALIDAS:
C                  DF       = JACOBIANO DE F(X)
C
          DIMENSION X(30),DF(3,5)
          COMMON X0(5),M,N
          L=12
          DO 1 K=1,N
                  L=L+3
                  DO 1 J=1,M
 1        DF(J,K)=X(L+J)
          RETURN
          END
C
C
          SUBROUTINE GRAD(GZ)
C
C         SUBRUTINA QUE CALCULA EL GRADIENTE DE LA FUNCION Z(X) EMPLEADA
C         EN EL PROGRAMA SANLSU.
C
C         ENTRADAS:
C                  N        = DIMENSION DE Z(X)
C                  X0       = PUNTO INICIAL
C         SALIDAS:
C                  GZ       = GRADIENTE DE Z(X)
C
          DIMENSION GZ(5)
          COMMON X0(5),M,N
          COMMON NORMA
          DO 1 I=1,N
 1        GZ(I)=0.0
          DO 2 I=1,N
          IF(NORMA.EQ.2) GZ(I)=1.0
 2        GZ(I)=X0(I)
          RETURN
          END
C
C
          SUBROUTINE FUNF(F,THETA,P,Q,X,XY)
          DIMENSION F(3),XY(3),THETA(5),X(30),Q(36),P(20),TE3(15)
          COMMON X0(5),M,N
          DO 1 I=1,N
                  TE3(I)=THETA(I)
```

```
      TCS(I+10)=SIN(TCS(I))
    1 CONTINUE
      CALL PRODQ(TCS,P,Q)
      CALL VECX(TCS,P,Q,X)
      DO 2 I=1,N
    2 F(I)=X(I)-XY(I)
      RETURN
      END
C
C
      SUBROUTINE PRODQ(TCS,P,Q)
      DIMENSION TCS(15),P(20),Q(36)
C
C     EN ESTA SUBRUTINA SE CALCULAN Y SE ALMACENAN LOS PRODUCTOS:
C     Q1 EN Q(1-9)
C     Q1*Q2 EN Q(10-18)
C     Q1*Q2*Q3 EN Q(19-27)
C     Q1*Q2*Q3*Q4 EN Q(28-36)
C
      Q(1)=TCS(6)
      Q(2)=TCS(11)
      Q(3)=0.0
      Q(4)=-TCS(11)*P(11)
      Q(5)=TCS(6)*P(11)
      Q(6)=P(16)
      Q(7)=TCS(11)*P(16)
      Q(8)=-TCS(6)*P(16)
      Q(9)=P(11)
      DO 1 I=2,4
            IP=9*(I-1)
            SC=TCS(I+10)*P(I+10)
            SS=TCS(I+5)*P(I+10)
            CC=TCS(I+10)*P(I+15)
            CS=TCS(I+5)*P(I+15)
            IA=IP-9
            DO 1 J=1,3
            Q(IP+J)=Q(IA+J)*TCS(I+5)+Q(IA+J+3)*TCS(I+10)
            Q(IP+J+3)=-Q(IA+J)*SC+Q(IA+J+3)*SS+Q(IA+J+6)*P(I+15)
            Q(IP+J+6)=Q(IA+J)*CC-Q(IA+J+3)*CS+Q(IA+J+6)*P(I+10)
    1 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE VECX(TCS,P,Q,X)
      DIMENSION TCS(15),P(20),Q(36),X(30)
C
C     ESTA SUBRUTINA CALCULA EL VECTOR Xi Y SU PRIMERA DERIVADA CON
C     RESPECTO A THETA.
C
      X(13)=P(5)*TCS(10)
      X(14)=P(5)*TCS(15)
      X(15)=P(10)
      X(28)=-Q(28)*X(14)+Q(31)*X(13)
      X(29)=-Q(29)*X(14)+Q(32)*X(13)
      X(30)=-Q(30)*X(14)+Q(33)*X(13)
      DO 1 I=1,4
            K=5-I
            KK=[illegible]
            KKP3=[illegible]
            PX1=P(K)+X(KKP3+1)
            PX2=P(K+10)*X(KKP3+2)-P(K+15)*X(KKP3+3)
            X(KK-2)=TCS(K+5)*PX1-TCS(K+10)*PX2
            X(KK-1)=TCS(K+10)*PX1+TCS(K+5)*PX2
            X(KK)=P(K+5)+P(K+15)*X(KKP3+2)+P(K+10)*X(KKP3+3)
```

```
      KD=29-7*I
      IF(I.EQ.4) GO TO 2
      X(KD+1)=-Q(KD)*X(KK-1)+Q(KD+3)*X(KK-2)
      X(KD+2)=-Q(KD+1)*X(KK-1)+Q(KD+4)*X(KK-2)
      X(KD+3)=-Q(KD+2)*X(KK-1)+Q(KD+5)*X(KK-2)
      GO TO 1
2     X(KD+1)=-X(KK-1)
      X(KD+2)=X(KK-2)
      X(KD+3)=0.0
1   CONTINUE
    RETURN
    END
```


Referencias

[1] Forsythe G.E., Malcom M.A. y Moler C.B. Computer Methods for Mathematical Computations, Prentice-Hall Inc. Englewood Cliffs N.J. 1977.

[2] Moler C.B., Matrix Eigenvalue and Least Square Computations, Computer Science Department, Stanford University, Stanford, California, 1973.

[3] Moler C.B., Matrix Computations With Fortran and Paging, Communications of the A.C.M., Volume 15, num. 4, April 1972.

[4] Moler C.B., Algorithm 423, Linear Equation Solver, Communications of the A.C.M., Volume 15, num. 4, April 1973.

[5] Angeles J., Spatial Kinematic Chains. Analysis, Synthesis and Optimization, Springer-Verlag, Berlín 1982.

[6] Ralston A., Introducción al Análisis Numérico, Limusa, México 1978.

[7] Angeles J., Rojas A., Subprogramas para el Análisis de Cadenas Cinemáticas con 7 Eslabones Acoplados Mediante Pares de Rotación.

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO CINEMATICO DE MAQUINARIA

GUIDANCE OF AXIALLY SYMMETRIC RIGD BODIES USING FIFTH-DEGREE OF FREEDOM REVOLUTE COUPLED MANIPULATORS

DR. JORGE ANGELES ALVAREZ

JUNIO, 1984.

Palacio de Minería Calle de Tacuba 5 primer piso Deleg. Cuauhtémoc 06000 México, D.F. Tel.: 521-40-20 Apdo. Postal M-2285

GUIDANCE OF AXIALLY-SYMMETRIC RIGID BODIES USING FIFTH-DEGREE-OF-FREEDOM--REVOLUTE-COUPLED MANIPULATORS.

Jorge Angeles[1]

Abstract

A kinematic model is constructed that allows the computation of all joint angles associated with fifth-degree-of-freedom revolute -coupled kinematic chains guiding an axially-symmetric rigid body through a set of prescribed configurations. Moreover, if velocity and acceleration specifications are introduced, the model also provides the first two derivatives of the joint angles. Finally, the applicability of the algorithm presented here, to the analysis of single-degree-of-freedom single-loop-6R closed chains is shown with an example.

1 Professor, University of Mexico. DEPFI-UNAM, C. Universitaria. Apdo. Postal 70-256. 04510 México, D.F. MEXICO.

JAA'jom.

CONDUITE DE CORPS REGIDES AXIALEMENT SYMMETRIQUES AU MOYEN DE MANIPULATEURS Ã CINQ COUPLES ROTOÏDES

Jorge Angeles

Résumé

L'auteur présente un modèle cinématique permettant de calculer tous les angles associés aux cinq couples rotoïdes des chaînes cinématiques à cinq degrés de liberté, destinées à guider des corps rigides axialement symétriques au moyen d'un ensemble de configurations préétablies.
En outre, si des specifications de vitesse et d'accélération y sont introduites, le modèle fournit aussi les deux premières dérivées des cinq angles. En fin, l'applicabilité de l'algorithme présenté ici à l'analyse des chaînes fermées à un degré de liberté comportant six couples rotoïdes est montré par moyen d'un exemple.

Nomenclature

x: n-dimensional vector over the real field

X: mxn matrix over the real field

X_i, Y_i, Z_i: Cartesian axes fixed to the ith link of the chain

$[x]_i$, x_i: 3x1 array containing the components of 3-dimensional vector x referred to X_i, Y_i, Z_i

$[X]_i$, X_i: 3x3 array containing the components of 3x3 matrix X referred to X_i, Y_i, Z_i.

tr (X): the scalar invariant of x matrix X, $X_{11} + \ldots X_{nn}$

R_i: the axis of the ith revolute pair of the chain, coincident with Z_i.

θ_i: the angle of rotation of link $i+1$ with respect to link i, its sign being defined by the positive direction of Z_i.

α_i: the angle between R_i and R_{i+1}, its sign being defined by the positive direction of axis X_{i+1}, which is directed in turn along the common perpendicular to R_i and R_{i+1}, from R_i to R_{i+1}.

a_i: The distance between axes R_i and R_{i+1}, hence always positive

b_i: the coordinate of the intersection of axes X_{i-1} and Z_i in frame X_i, Y_i, Z_i

$[Q_{i,i+1}]_i$, Q_i: the matrix rotating axes X_i, Y_i, Z_i into an orientation parallel pairwise to X_{i+1}, Y_{i+1}, Z_{i+1}, respectively, referred to X_i, Y_i, Z_i coordinates

$[a_{i,i+1}]_i$, a_i: the vector connecting the origins of X_i, Y_i, Z_i and X_{i+1}, Y_{i+1}, Z_{i+1}, directed from the former to the latter, in X_i, Y_i, Z_i coordinates.

r_A, r_B: the position vector of points A and B, respectively, in the specified configuration, measured from the origin of X_1, Y_1, Z_1. 4

v_A, v_B: the velocity of points A and B, respectively

a_A, a_B: the acceleration of points A and B, respectively

Introduction

Industrial robot manipulators in use are frequently required to perform tasks involving the guidance of rigid bodies with axial symmetry, e.g. turned workpieces, painting nozzles, arc welding pistols, etc. Most of the time, however, these tasks are realized with sixth-degree-of--freedom manipulators. Since the guidance of axially-symmetric rigid bodies does not involve their orientation about the axis of symmetry, it seems natural to perform these tasks with a fifth-degree-of freedom manipulator. If, on the other hand, a sixth-or greater-degree-of--freedom manipulator is to be used anyway, the redundant degree of freedom can be used to optimize some given performance index. Hence the interest to establish a kinematic model allowing the computation of the joint angles and their time derivatives, associated with multiple-degree-of-freedom manipulators in the presence of incomplete specification of the hand motion.

Following the approach introduced in [1], the set of desired equations is derived regarding the problem as one involving the guidance of a segment of a rigid line, by specifying the displacement, velocity and acceleration of any two points of that segment. Of course, these specified variables should meet the compatibility condition guaranteeing that the distance between these points is preserved throughout any physically possible motion.

The kinematics of rigid segments has been considered in [2] regarding only the velocity and acceleration analyses. A complete account of the kinematics of rigid segments, lines and points is given in [3]. 6
Furthermore, an analysis of 5R open kinematic chains to guide axially-symmetric rigid bodies appeared in [4]. The approach followed in this paper differs from these, being aimed at devising a real-time implementable algorithm oriented towards the computer-control of 5R robot manipulators, applied to the guidance of axially symmetric rigid bodies.

Displacement analysis

In this Section, reference is made to the fifth-degree-of-freedom kinematic chain depicted in Fig 1. Moreover, the notation and method of Denavit and Hartenberg [5, pp 343-355] is applied throughout. Thus, a set of Cartesian coordinate axes $\{X_i, Y_i, Z_i\}$ is attached to the i th link. According to Denavit and Hartenberg's notation, axis Z_i is placed along the axis of pair R_i, its positive direction denoting the direction in which angle θ_i is measured. Axis X_i is defined as the common perpendicular to the axes of R_{i-1} and R_i, directed from the former to the latter. Notice that there is absolute freedom to chose X_1 as any line in a plane perpendicular to Z_1. The orthogonal matrix rotating axes labelled i into an orientation paralel to those labelled $i+1$, referred to axes i, is represented as $[\, Q_{i,\, i+1}\,]_i$ or as Q_i, for compactness. Finally, the vector connecting the origins O_i and O_{i+1}, respectively, directed from the former to the latter, in i-coordinates, is represented as $[\, a_{i,\, i+1}\,]_i$, or as a_i, for compactness. According to the nomenclature, these are[2]

$$[\, Q_{i,\, i+1}\,]_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i \\ 0 & s\alpha_i & c\alpha_i \end{bmatrix} \tag{1}$$

$$[\, a_{i,\, i+1}\,]_i = [\, a_i c\theta_i,\ a_i s\theta_i,\ b_i\,]^T \tag{2}$$

Let line AB of Fig 1 be the axis of symmetry of the rigid body, not shown there, that is meant to be guided. Furthermore, let r_A and

---

[2] Throughout, $c(\ ) \equiv \cos(\ )$, $s(\ ) \equiv \sin(\ )$.

r_B be the *prescribed values* of the position vectors of A and B, respectively, letting x_1 and y_1 be the synthesized values of the same position vectors. Thus, r_A and r_B are known constants, whereas $x_1 = x_1(\theta)$, $y_1 = y_1(\theta)$, are functions of θ, where θ denotes the 5-dimensional vector whose ith component is θ_i. Vectors x_1 and y_1 can be computed recursively as follows:

$$x_4 = a_4 \tag{3a}$$

$$x_k = a_k + Q_k x_{k+1}, \quad k = 3, 2, 1 \tag{3b}$$

$$y_5 = a_5 \tag{4a}$$

$$y_k = a_k + Q_k y_{k+1}, \quad k = 4, 3, 2, 1 \tag{4b}$$

The displacement equations are, then,

$$x_1 = r_A, \quad y_1 = r_B \tag{5}$$

Now, the six-dimensional vector f is defined as

$$f \equiv \begin{bmatrix} f_1 \\ \hdashline f_2 \end{bmatrix}$$

f_1 and f_2 being the three-dimensional vectors defined in turn as

$$f_1 = x_1 - r_A, \quad f_2 = y_1 - r_B \tag{6}$$

Thus, the displacement equations lead to the following sixth-order nonlinear algebraic system in five unknowns:

$$f(\theta) = 0 \tag{7}$$

This system can be solved using Newton-Raphson's method [6, pp 248-253] as follows: Let θ^0 be an initial "guess" for the solution. Then generate the sequence $\theta^1, \ldots, \theta^k, \ldots$ which, if converges, will do so quadratically [6, pp 222-226]. This sequence is generated recursively from Newton-Raphson's iterative scheme:

$$\theta^{k+1} = \theta^k + \Delta\theta^k \tag{8}$$

where $\Delta\theta^k$ is computed from the Taylor expansion of $f(\theta)$ at $\theta=\theta^k$, which leads to

$$J(\theta^k)\,\Delta\theta^k = -f(\theta^k) \tag{9}$$

Equation (9) is a sixth-order linear algebraic system in five unknowns, its Jacobian matrix J being 6x5. Thus J cannot be inverted properly speaking. However, from the nature of the problem, out of the six scalar equations appearing in (9), only five are independent. In fact, no matter what the value of θ^k is, $f(\theta^k)$, and hence $-f(\theta^k)$, lies in the range of J, if the motion of points A and B is not to violate the rigidity condition. As a matter of example, consider the one-degree-of-freedom 6R linkage[3] depicted in Fig 2 for the particular values $\theta_1 = -\theta_3 = \theta_5 = 120°$, $\theta_2 = \theta_4 = 0°$. The motion of this linkage can be analysed as one of an open 5R kinematic chain, as shown in Example 1. For that configuration,

$$J = \begin{bmatrix} 0 & 0 & \sqrt{3} & 0 & 0 \\ -2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ \hdashline 0 & 0 & \sqrt{3} & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & -1/2 & 0 & -1/2 & 0 \end{bmatrix}$$

[3] A wire model of this is for sale commercially under the trade mark HEXIFLEX^{TR}.

Now assume that the link connecting O_6 with O_1 is removed, thus obtaining an open chain. Moreover, it is desired to compute the angle increments $\Delta\theta_i$ ($i = 1, \ldots, 5$) for the following value of Δf:

$$\Delta f = \begin{bmatrix} \Delta x_1 \\ \Delta y_1 \end{bmatrix}; \ \Delta x_1 = \begin{bmatrix} 0 \\ 0 \\ \xi \end{bmatrix}, \ \Delta y_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Δx_1 being specified parallel to the Z_1 axis, otherwise arbitrarily.

This motion clearly complies with the rigid-body condition, for $\Delta y_1 - \Delta x_1$, the displacement increment of B with respect to A, is perpendicular to AB, as it should be if the distance between A and B is to remain constant. Hence Δf lies in the range of J. Eq (9) thus yields, for the assumed values of θ_i,

$$\Delta\theta_1 = \Delta\theta_3 = \Delta\theta_5 = 0, \ \ \Delta\theta_2 = -\Delta\theta_4 = \xi$$

If, on the other hand, Δf is specified as

$$\Delta f = \begin{bmatrix} \Delta x_1 \\ \Delta y_1 \end{bmatrix}; \ \ \Delta x_1 = \begin{bmatrix} \xi \\ 0 \\ 0 \end{bmatrix}, \ \Delta y_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

which clearly violates the rigid-body condition, then eq (9) leads to an inconsistent equation system producing, from the first equation

$$\Delta\theta_3 = \frac{\sqrt{3}}{3}\xi$$

and, from the fourth equation,

$$\Delta\theta_3 = 0$$

which contradicts the former one . This is due to the fact that Δf does not lie in the range of J because it does not comply with the rigid-body condition.

Thus, except for singular configurations, i.e. those leading to a rank-defficient Jacobian matrix - one for which rank (J) <5-, $\Delta\theta^k$ can be solved for from eq (9). Since this equation contains, in general, 5 consistent linearly independent scalar equations, but one does not know which one is redundant, 'Gauss' algorithm or, equivalently, the LU decomposition [7, pp 27-33] cannot be applied to it directly. The solution proposed here is to regard eq (9) as a 6x5 overdetermined linear system and compute its "least-squere" solution. This will be, in fact, *its solution*, given the consistency of the involved system of equations. There are several ways of computing the least-square solution of eq (9), but the one preferred by the author is using Householder reflections [8], which produces implicitly the *pseudo-inverse* [9, pp 103-113] of J, J^I. Thus

$$\Delta\theta^k = - J^I (\theta^k) f (\theta^k) \qquad (10a)$$

with

$$J^I (\theta^k) = [J^T (\theta^k) J (\theta^k)]^{-1} J^T (\theta^k) \qquad (10b)$$

Pseudo-inverses have been already used in connection with the analysis of multiple-degree-of-freedom manipulators [10], though their use is not very popular.

Velocity analysis

In this section, a system of six linearly-dependent consistent equations is derived that allows the computation of $\dot{\theta}$, given the velocities of A and B, henceforth denoted by the 3-dimensional vectors v_A and v_B, respectively. These are, clearly

$$\dot{x}_1 = v_A, \ \dot{y}_1 = v_B \tag{11}$$

with

$$\dot{x}_1 = \frac{\partial x_1}{\partial \theta}\dot{\theta}, \ \ \dot{y}_1 = \frac{\partial y_1}{\partial \theta}\dot{\theta} \tag{12}$$

which thus lead to

$$J(\theta)\,\dot{\theta} = v \tag{13}$$

with the 6-dimensional vector v given by

$$v = \begin{bmatrix} v_A \\ \hline v_B \end{bmatrix} \tag{14}$$

Recalling eq (10), $\dot{\theta}$ can then be obtained as

$$\dot{\theta} = J^I v \tag{15}$$

J^I being given as in eq (10b). It is pointed out here that, since J^I was already computed in computing θ, it need not be recomputed. In fact, using subroutines HECOMP and HOLVE [11] to obtain the least-square solution (10), HECOMP is first applied to J in order to rendering it upper triangular. Next, HOLVE is applied to the transformed J and the right-hand side of eq (9) to produce solution (10). This means that HECOMP is applied only once, whereas HOLVE twice. Most of the operations involved in the aforementioned procedure 13 are performed in HECOMP; thus, $\dot{\theta}$ is obtained at virtually no additional cost.

Acceleration analysis

Next the computation of $\ddot{\theta}$ is outlined. This is clearly obtained by differentiating eqs (11) with respect to time:

$$\ddot{x}_J = a_A, \quad \ddot{y}_J = a_B \tag{16}$$

where a_A and a_B denote the accelerations of A and B, respectively. Furthermore, from eqs (12),

$$\ddot{x}_J = \frac{\partial x_J}{\partial \theta}\,\ddot{\theta} + \left(\frac{\partial^2 x_J}{\partial \theta^2}\,\dot{\theta}\right)\dot{\theta} \tag{17}$$

with a similar one for $\ddot{y}_J$. Substituting eq (17) and its $\ddot{y}_J$ counterpart in eqs (16), one obtains

$$J(\theta)\,\ddot{\theta} = a \tag{18}$$

with

$$a = \left[\begin{array}{c} a_A - \left(\dfrac{\partial^2 x_J}{\partial \theta^2}\,\dot{\theta}\right)\dot{\theta} \\ \hdashline a_B - \left(\dfrac{\partial^2 y_J}{\partial \theta^2}\,\dot{\theta}\right)\dot{\theta} \end{array}\right] \tag{19}$$

Hence, $\ddot{\theta}$ is obtained as

$$\ddot{\theta} = J^I(\theta)\,a \tag{20}$$

and, as said before in connection with $\dot{\theta}$, at virtually no additional cost.

Example 1. Analysis of a 6R single-degree-of-freedom closed kinematic chain. The parameters defining the architecture of the chain are given below. Regarding links 6 and 5 as the fixed and the input link, respectively, determine the histories $\theta_i(t)$, $\dot{\theta}_i(t)$, $\ddot{\theta}_i(t)$ $(i = 1, \ldots, 5)$, $\theta_6(0) = 0$, $\dot{\theta}_6 = 1$ rad/s and $\ddot{\theta}_6 = 0$. The parameters are:

$$a_i = 1, \quad b_i = 0, \quad \alpha_i = 90°, \quad i = 1, \ldots, 5, \quad \alpha_6 = -90°.$$

Fig 2 shows the kinematic chain described by the foregoing parameters, for the configuration

$$\theta_1 = -\theta_3 = \theta_5 = 120°, \quad \theta_2 = \theta_4 = \theta_6 = 0°$$

Solution:

Due to the symmetries of this linkage,

$$\theta_1 = -\theta_3 = \theta_5, \quad \theta_2 = -\theta_4 = \theta_6$$

Vectors x_k $(k = 4, 3, 2, 1)$ are readily computed as

$$x_4 = a_4 = [\, c_4, s_4, 0\,]^T = [\, c_2, -s_2, 0\,]^T$$

$$x_3 = a_3 + Q_3 x_4 = [\, c_1(1+c_2), -s_1(1+c_2), -s_2 \,]^T$$

$$x_2 = a_2 + Q_2 x_3 = \begin{bmatrix} c_2 [\,1+c_1(1+c_2)\,] - s_2^2 \\ s_2 [\,1+c_1(1+c_2)\,] + c_2 s_2 \\ -s_1(1+c_2) \end{bmatrix}$$

$$x_1 = a_1 + Q_1 x_2 = \begin{bmatrix} c_1\left\{1+c_2[\,1+c_1(1+c_2)]\ - s_2^2\right\} - s_1^2(1+c_2) \\ s_1\left\{1+c_2[\,1+c_1(1+c_2)]\ - s_2^2\right\} + c_1 s_1(1+c_2) \\ s_2[\,1+c_1(1+c_2)]\ + c_2 s_2 \end{bmatrix}$$

Vectors y_k $(k = 5, 4, 3, 2, 1)$ are similarly computed as

$$y_5 = a_5 = [c_5, s_5, 0\,]^T = [\, c_1, s_1, 0\,]^T$$

$$y_4 = a_4 + Q_4 y_5 = [c_2 (1+c_1), -s_2(1+c_1), s_1\,]^T$$

$$y_3 = a_3 + Q_3 y_4 = \begin{bmatrix} c_1 \ [1+c_2 (1+c_1)\,] - s_1^2 \\ -s_1 \ [1+c_2(1+c_1)\,] - c_1 s_1 \\ -s_2(1+c_1) \end{bmatrix}$$

$$y_2 = a_2 + Q_2 y_3 = \begin{bmatrix} c_1 c_2[\, c_1+1+c_2(1+c_1)] - s_2^2(1+c_1) \\ c_1 s_2[\, c_1+1+c_2(1+c_1)] + c_2 s_2(1+c_1) \\ - s_1[\, c_1+ 1+c_2\,(1+c_1)] \end{bmatrix}$$

$$y_1 = \begin{bmatrix} c_1\left\{1+c_1c_2[\, c_1+1+c_2(1+c_1)] - s_2^2(1+c_1)\right\} - s_1^2[\, c_1+1+c_2(1+c_1)] \\ s_1\left\{1+c_1c_2[\, c_1+1+c_2(1+c_1)] - s_2^2(1+c_1)\right\} + c_1s_1[\, c_1+1+c_2(1+c_1)] \\ s_2c_1[\, c_1+1+c_2(1+c_1)] + c_2s_2(1+c_1) \end{bmatrix}$$

The displacement equations are now obtained equating x_1 with r_A and y_1 with r_B, r_A and r_B being the position vectors of points A and B, respectively, in X_1, Y_1, Z_1 coordinates. Hence

$$c_1\left\{1+c_2[\, 1+c_1(1+c_2)] - s_2^2\right\} - s_1^2(1+c_2) = -(1+c_2) \qquad (i)$$

$$s_1\left\{1+c_2[\, 1+c_1(1+c_2)] - s_2^2\right\} + c_1s_1(1+c_2) = 0 \qquad (ii)$$

$$s_2\,[\, 1+c_1(1+c_2)\,] + c_2s_2 = s_2 \qquad (iii)$$

$$c_1\left\{1+c_1c_2\,[c_1+1+c_2(1+c_1)] - s_2^2(1+c_1)\right\} - s_1^2\,[c_1+1+c_2(1+c_1)] = -1 \qquad (iv)$$

$$s_1\left\{1+c_1c_2[c_1+1+c_2(1+c_1)] - s_2^2(1+c_1)\right\} + c_1s_1[\, c_1+1+c_2(1+c_1)] = 0 \qquad (v)$$

$$c_1s_2[\, c_1+1+c_2(1+c_1)] + c_2s_2(1+c_1) = 0 \qquad (vi)$$

From eq (iii), with $s_2 \neq 0$ and $c_1 \neq -1$.

$$(1+c_1)\,(1+c_2) = 1 \qquad (I/O)$$

which is the input-output equation, valid for $c_1 \neq -1$. For $c_1 = -1$, $s_2 = 0$.

Now, let $\{\ \}_1$ and $[\ \ ]_1$ be the corresponding brackets appearing in eqs (i-iii). $\{\ \}_2$ and $[\ \ ]_2$ are correspondingly defined for eqs (iv - vi). If (I/O) is substituted into these, one has

$$[\ \ ] = \frac{1+2c_1}{1+c_1}, \ [\ \ ] = 1 \qquad (vii)$$

$$\{\ \}_1 = -\frac{c_1}{1+c_1}, \qquad = -c_1 \qquad (viii)$$

Substitution of relations (*vii* & *viii*) into eqs (*i* - *vi*) renders both sides identical, thus verifying the validity of the input-output equation. From (I/O) it is clear that neither θ_1 nor θ_2 can attain the value π. In fact, a mobility analysis yields the following mobility ranges:

$$-120° \leq \theta_i \leq 120°, \ i = 1, \ldots, 6$$

The program implementing the algorithm presented in this paper was tested with this linkage, its input-output equation (I/O) being plotted in Fig 3a. The program output was compared with the foregoing closed-form solution and the displacement, velocity and acceleration errors were plotted as shown in Figs 3b, 3c and 3d, respectively.

As in the case of seven-link chains [1], the displacement, velocity and acceleration errors grow unbounded as the Jacobian matrix approaches a singularity which, in this case, manifests itself as a deficiency of its rank. This occurs at singular configurations of the linkage under study, i.e. at dead-point positions of its input link, $\theta_6 = \theta_2 = \pm 120°$.

Example 2. Synthesis of a spatial trajectory.

Given the 5R manipulator whose parameters appear in Table 1, determine the histories $\theta_i(t)$, $\dot{\theta}_i(t)$, for $0 \leq t \leq T$, $i = 1, \ldots, 5$, in order to guide line AB, as shown in Fig 4, through a given spatial trajectory. The location of point B, in fifth-link coordinates, is given by $[r]_5 = [a_5, 0, 0]^T$, whereas the trajectory is chosen as the upper branch of the intersection of the sphere (S) with the cylinder (C) given below:

$$(x - a)^2 + (y + b)^2 + z^2 = 15a^a \qquad (S)$$

$$x^2 + z^2 = a^2 \qquad (C)$$

upper branch: $y > b$ (B)

the said surfaces being given in fixed coordinates, i.e. in X_1, Y_1, Z_1 coordinates. It is required, moreover, that

$$x = -a \cos \beta$$

$$y = a \sin \beta$$

with β being chosen as a smooth-enough function of time, i.e. one for which at least $\dot{\beta}$ and $\ddot{\beta}$ be continuous functions of time, in the interval $[0, T]$. Additionally, the following is imposed on β:

$$\beta(0) = 0, \ \beta(T) = 2\pi, \ \dot{\beta}(0) = \dot{\beta}(T) = 0, \ \ddot{\beta}(0) = \ddot{\beta}(T) = 0$$

Such a β function can be readily synthesized using spline functions, as shown in [12, 13]. The following parameter values were assumed:

$$a = 300 \text{ mm}, \ b = 2\,220 \text{ mm}, \ T = 60 \text{ s}$$

The orientation of AB was specified as follows. Let unit vectors

e_t, e_n, e_b denote the tangent, normal and binormal vectors of the trajectory. Then AB is to be orientated so that:

$$r_B - r_A = a_5\, e_n$$

Table 1

(lengths in mm, angles in degrees)

$a_1 = 0$	$b_1 = 0$	$\alpha_1 = 90$
$a_2 = 0$	$b_2 = 479$	$\alpha_2 = 90$
$a_3 = 0$	$b_3 = 0$	$\alpha_3 = 90$
$a_4 = 35.3$	$b_4 = 1016$	$\alpha_4 = -90$
$a_5 = 146$	$b_5 = 0$	$\alpha_5 = -90$

These values were taken from. The sixth-degree-of-freedom manipulator described in [14], from which the sixth revolute pair was removed.

Solution:

The reference configuration was chosen to be the following:

$$\theta_1 = 90°,\ \theta_2 = 180°,\ \theta_3 = 90°,\ \theta_4 = \theta_5 = 0°$$

In order to guide line AB from its position in the reference configuration to the initial configuration along the prescribed trajectory, determined by $\beta=0$, continuation was used, as donde in [1]. To this end, point B was made to trace a straight path between its two positions, that in the reference one, B^0, and that in $\beta=0$, B^I. This path was divided into 10 segments in order to ensure that the initial "guess" for the Newton-Raphson procedure lie close enough to the solution sought, which guarantees and accelerates its convergence. In fact, four interations were needed, at most, in order to reach convergence, in this stage.

Along the prescribed trajectory, one whole kinematic analysis was performed every 0.5s, i.e. 120 points on the given trajectory where fully analyzed. At virtually all of these points, convergence was reached after 3 iterations, which was the largest number of iterations required, for a tolerance of 10^{-6} imposed on $\Delta\theta$. At a few points about $\beta=0$, convergence was reached after only two iterations.

Computed values of θ_i, $\dot{\theta}_i$ and $\ddot{\theta}_i$ $(i=1,\ldots,5)$ are plotted in Figs 5-9. From these figures it becomes apparent that the smoothness of function β, synthesized with the aid of a cubic periodic spline, is reflected in the smoothness of the θ_i functions obtained.

Conclusions

The method presented here, aimed at the complete analysis of five-link revolute-coupled kinematic chains, differs basically from other methods intended to solve the same problem. Contrary to the usual practice of analyzing open kinematic chains (manipulators) as closed ones by introducing a "fictitious" closing link, the approach followed here is oriented to the analysis of open chains, regarding closed ones (linkages), as particular cases of the former. It resorts to efficient numerical methods of solution of nonlinear overdetermined algebraic systems. The system obtained here is derived disregarding the orientation of the body meant to be guided, while intending to position two of its points, which is also an original approach. The results show quick convergence, which suggests the applicability of the algorithm, and the program implementing it, to the real-time kinematic control of robot manipulators of the topology assumed here, i.e. 5R, but of arbitrary architecture otherwise.

Acknowledgements

The research work reported here was mainly supported by the Graduate Division of the Faculty of Engineering - National Autonomous University of Mexico (UNAM). The programming assistance of Mr. Angel A. Rojas-Salgado, a graduate student of this Division under a fellowship of the UNAM Program for Faculty Support, is highly acknowledged.

References

1. Angeles J, "Analysis of general seven-link revolute-coupled kinematic chains", Internal Report, DEPFI-University of Mexico, 1983 (Publication pending)

2. Henderson J M and Meriam J L, "On the space rotation of a two-point link", Mechanism and Machine Theory, Vol 10, 1975, pp 347-354

3. Tsai L-W and Roth B, "Incompletely specified displacements: Geometry and spatial linkage synthesis", ASME paper No. 72-Mech-13, presented 2 at the Mechanisms Conference, San Francisco, Calif., Oct. 8-12, 1972

4. Sugimoto K and Duffy J, "Analysis of five degree of freedom robot arms", J Mechanisms, Transmissions, and Automation in Design, Trans ASME, Vol 105, 1983, pp 23-27

5. Hartenberg R S and Denavit J, Kinematic Synthesis of Linkages, Mc Graw-Hill Book Co, New York, 1964

6. Dahlquist G and Björck Å, Numerical Methods, Prentice-Hall Inc, Englewood Cliffs (NJ), 1974.

7. Forsythe G E and Moler C B, Computer Solution of Linear Algebraic Systems, Prentice-Hall Inc, Englewood Cliffs (NJ), 1967.

8. Businger P and Golub G H, "Linear Least square solutions by Housholder transformations" in Bauer F L et al (editors), Handbook for Automatic Computation. Vol II: Linear Algebra (edited by Wilkinson H G and Reinsch C), Springer-Verlag, Berlin, 1971, pp 111-118

9. Ben-Israel A and Greville T N E, Generalized Inverses: Theory and Applications, John Wiley & Sons, Inc, New York, 1974

10. Fournier A, Génération des mouvements en robotique. Applications des inverses généralisées et des pseudo-inverses. Thèse d'Etat (Doctor's Thesis), Montpellier (France), 1980

11. Moler C B, Matrix Eigenvalue and Least Square Computations, Computer Science Department, Stanford University, Stanford, Calif, 1973

12. Angeles J, "Synthesis of plane curves with prescribed local geometric properties using periodic splines", Computer-Aided Design, Vol 15, No 3, May 1983, pp 147-155

13. Angeles J, "Síntesis de curvas planas cerradas usando funciones 'spline' paramétricas y periódicas", Revista de la Academia Nacional de Ingeniería, A C, Mexico City, Vol II, No 1, 1983

14. Whitney D E, "The mathematics of coordinated control of prosthetic arms and manipulators", Journal of Dynamic systems, Measurement and control, Vol 94, December 1972, pp 303-309

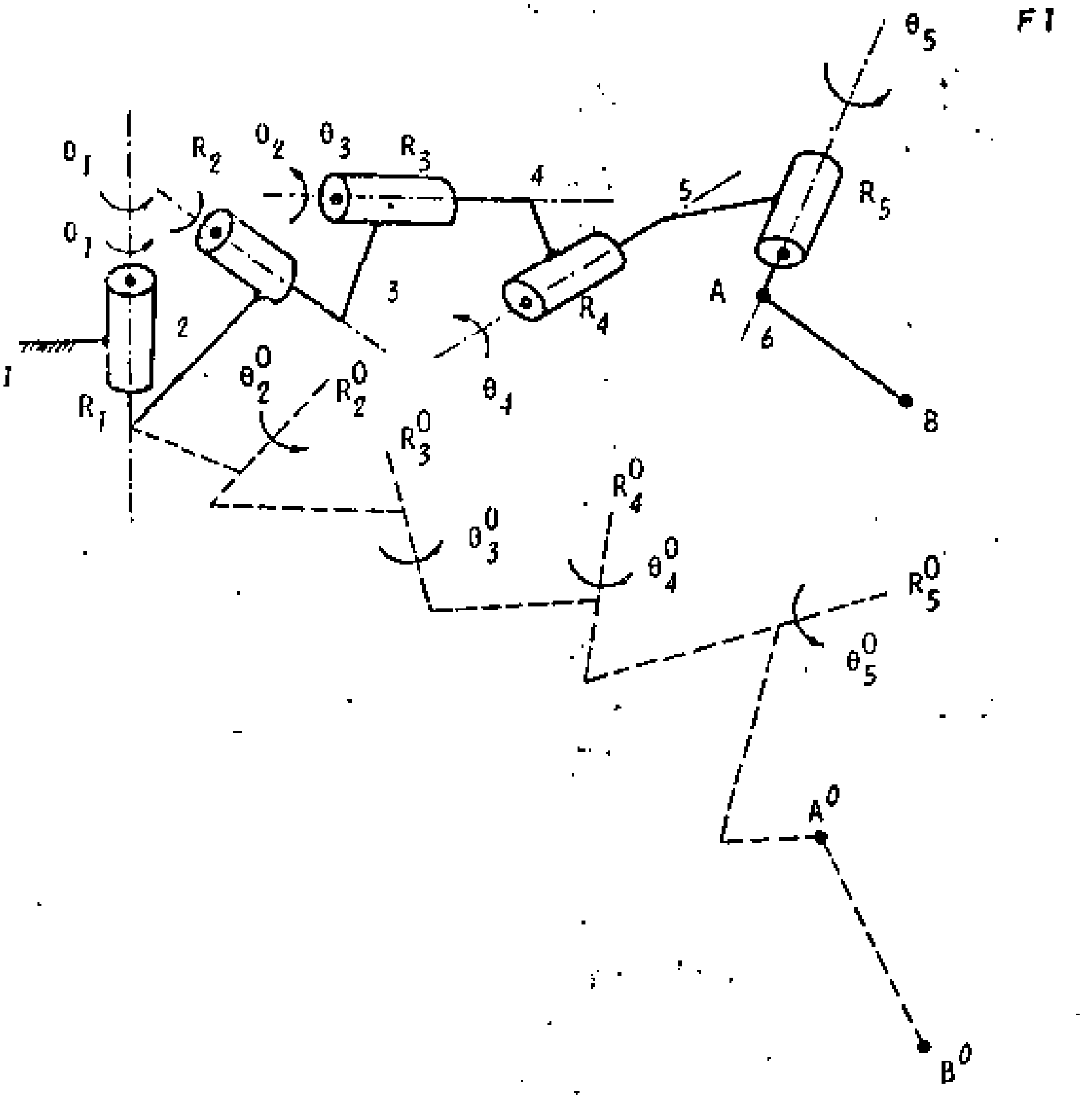


Fig 1 Architecture of a general fifth-degree-of-freedom revolute-coupled manipulator

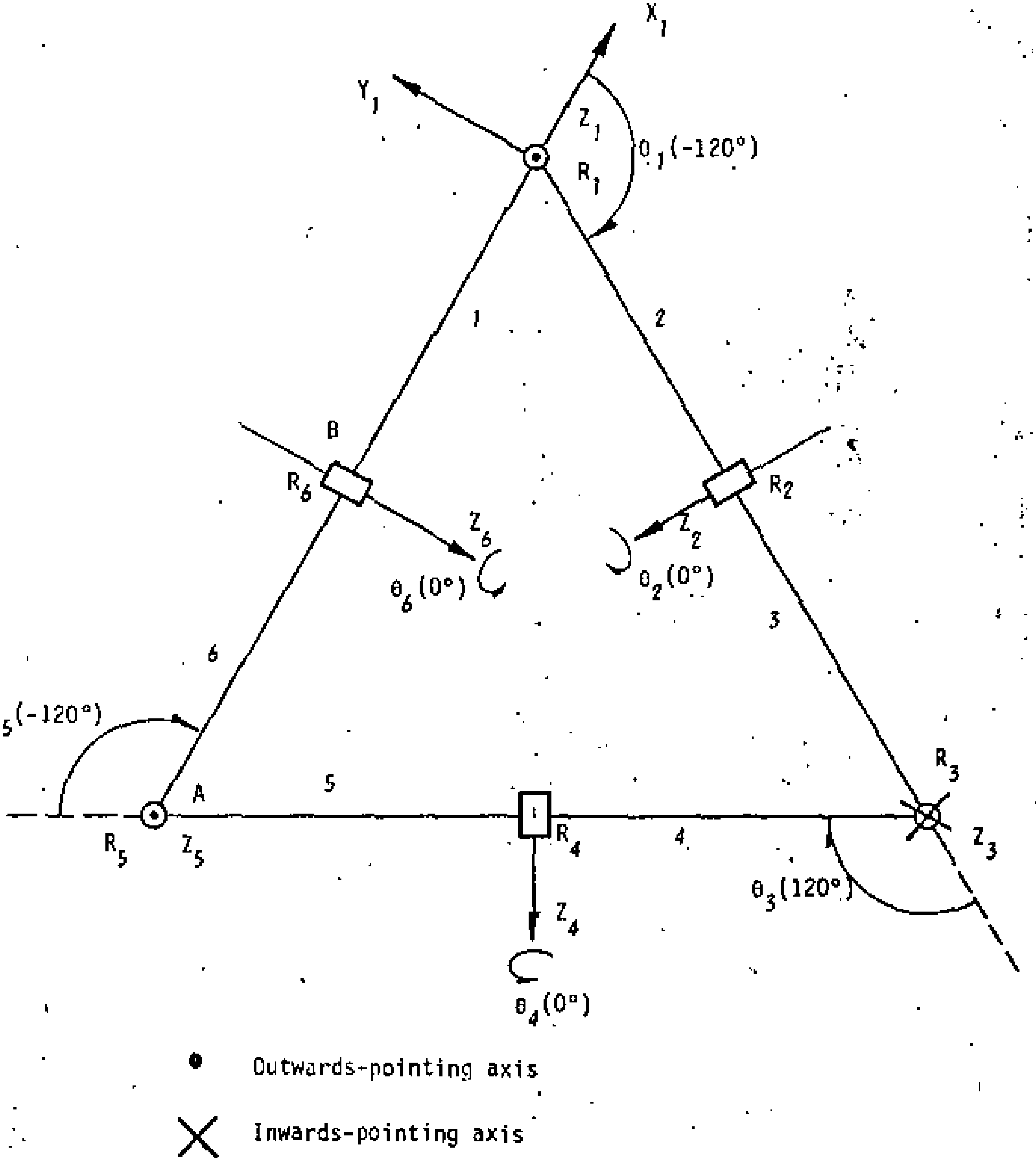


Fig 2 Layout of a 6R-single-degree-of-freedom linkage

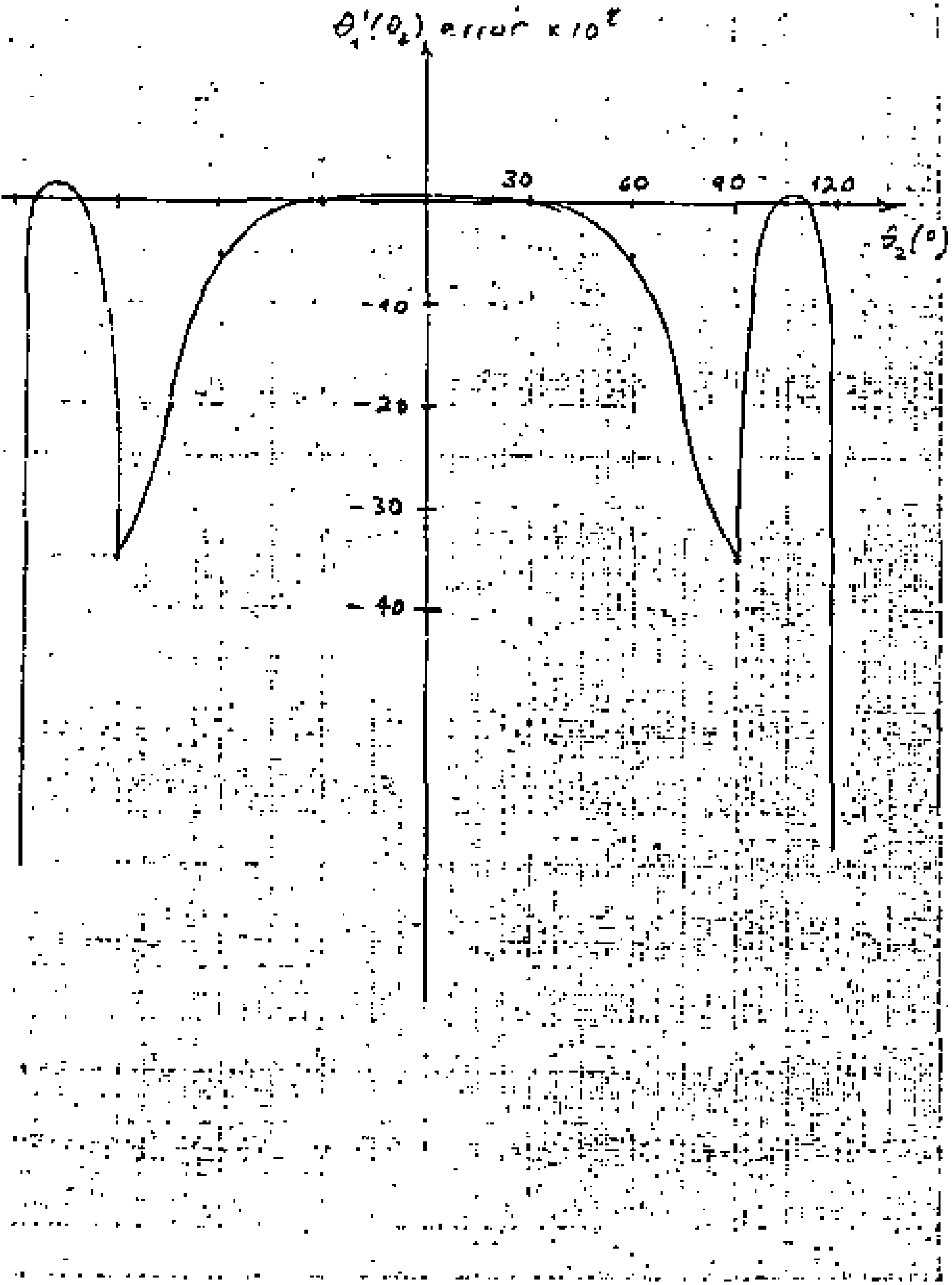


FIG 3b

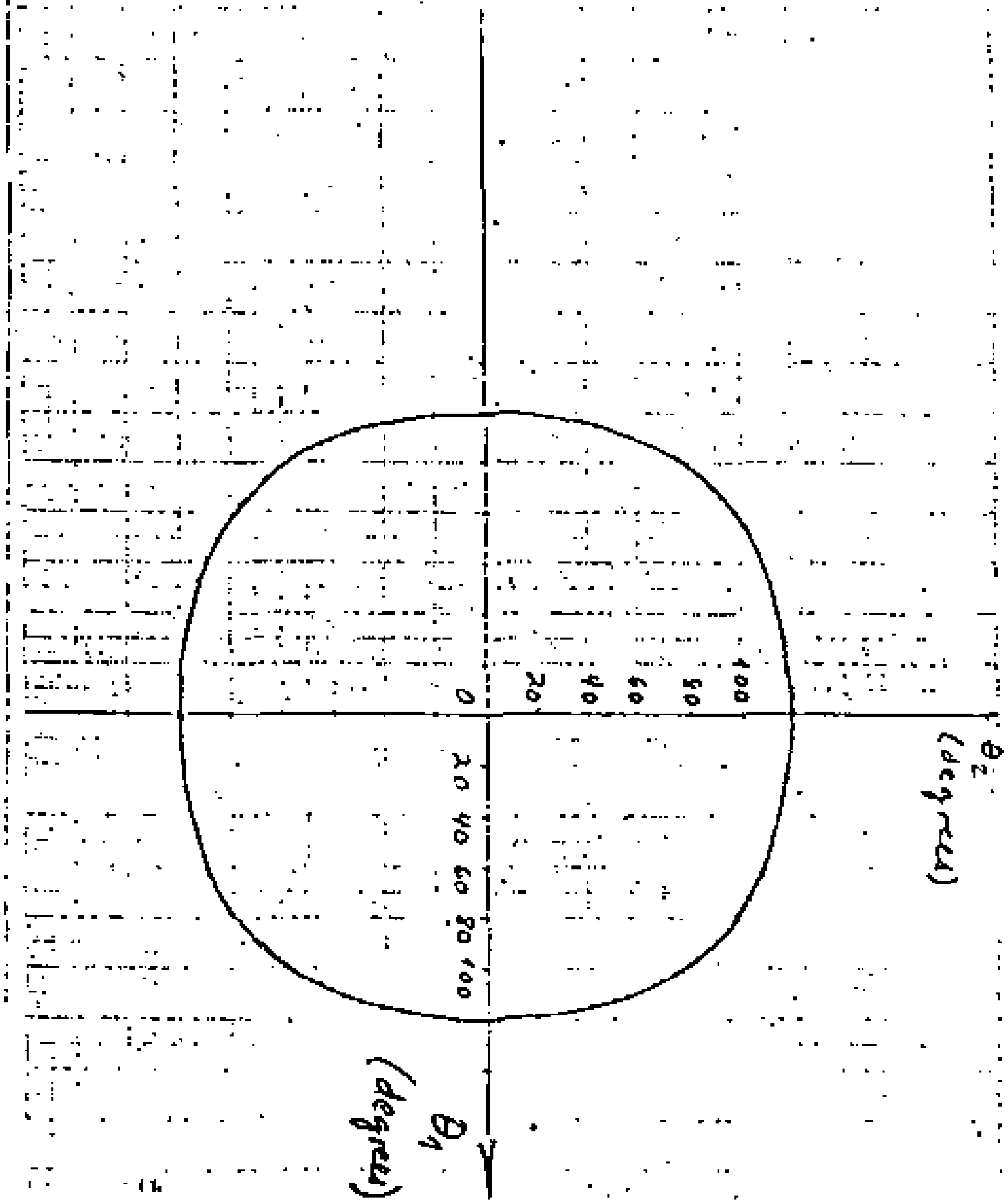


FIG 3a

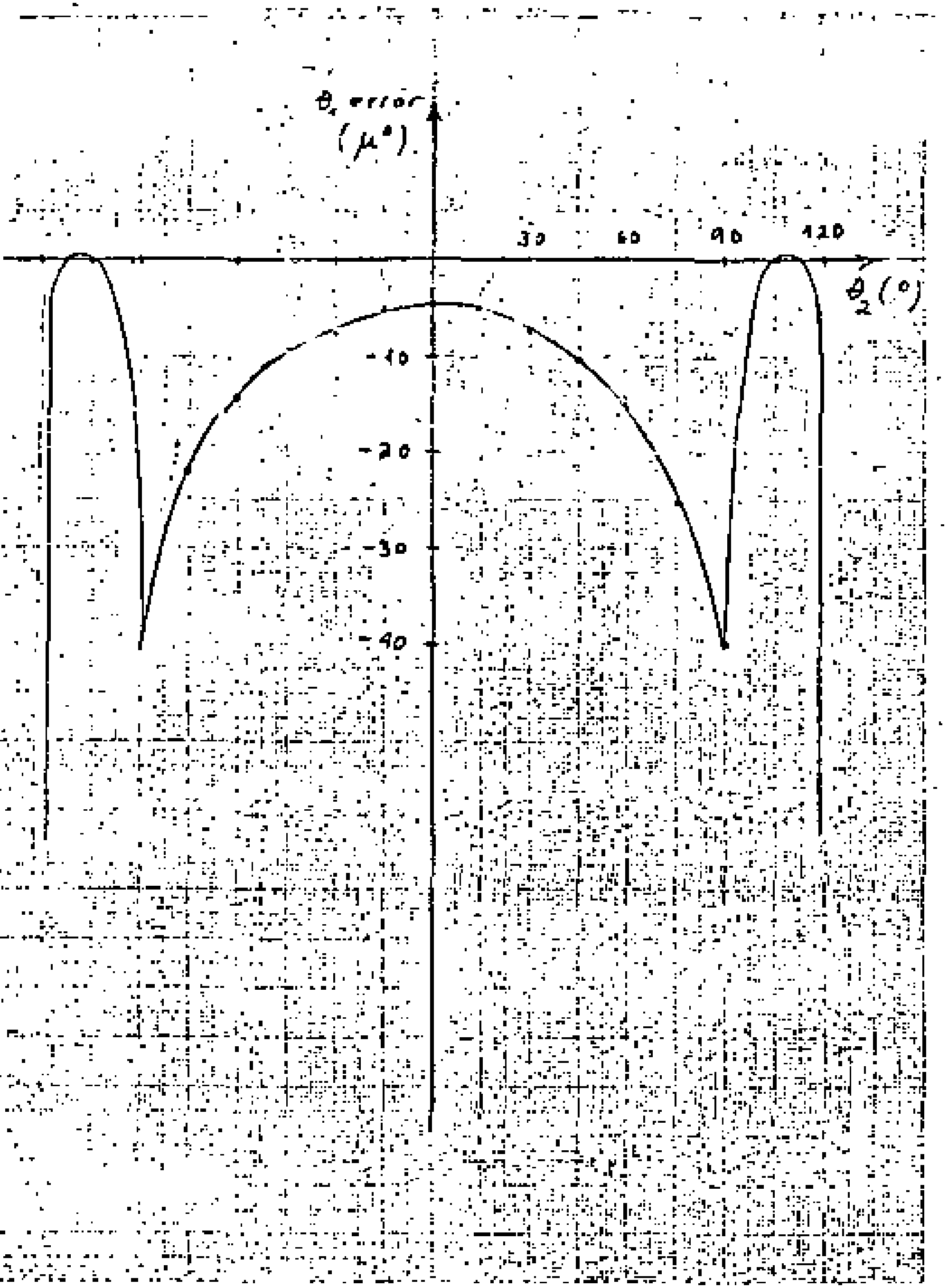


FIG 3c

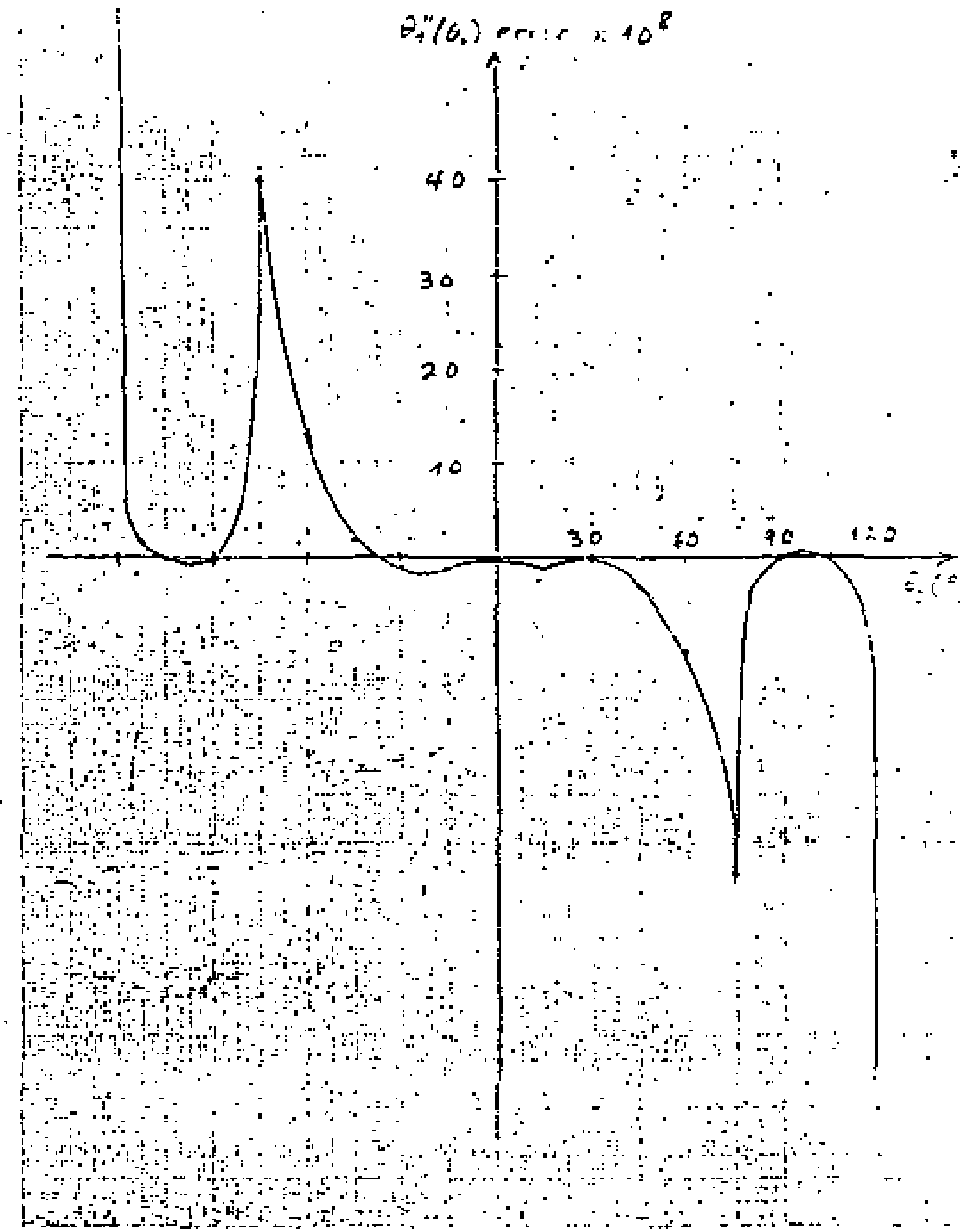


FIG 3d

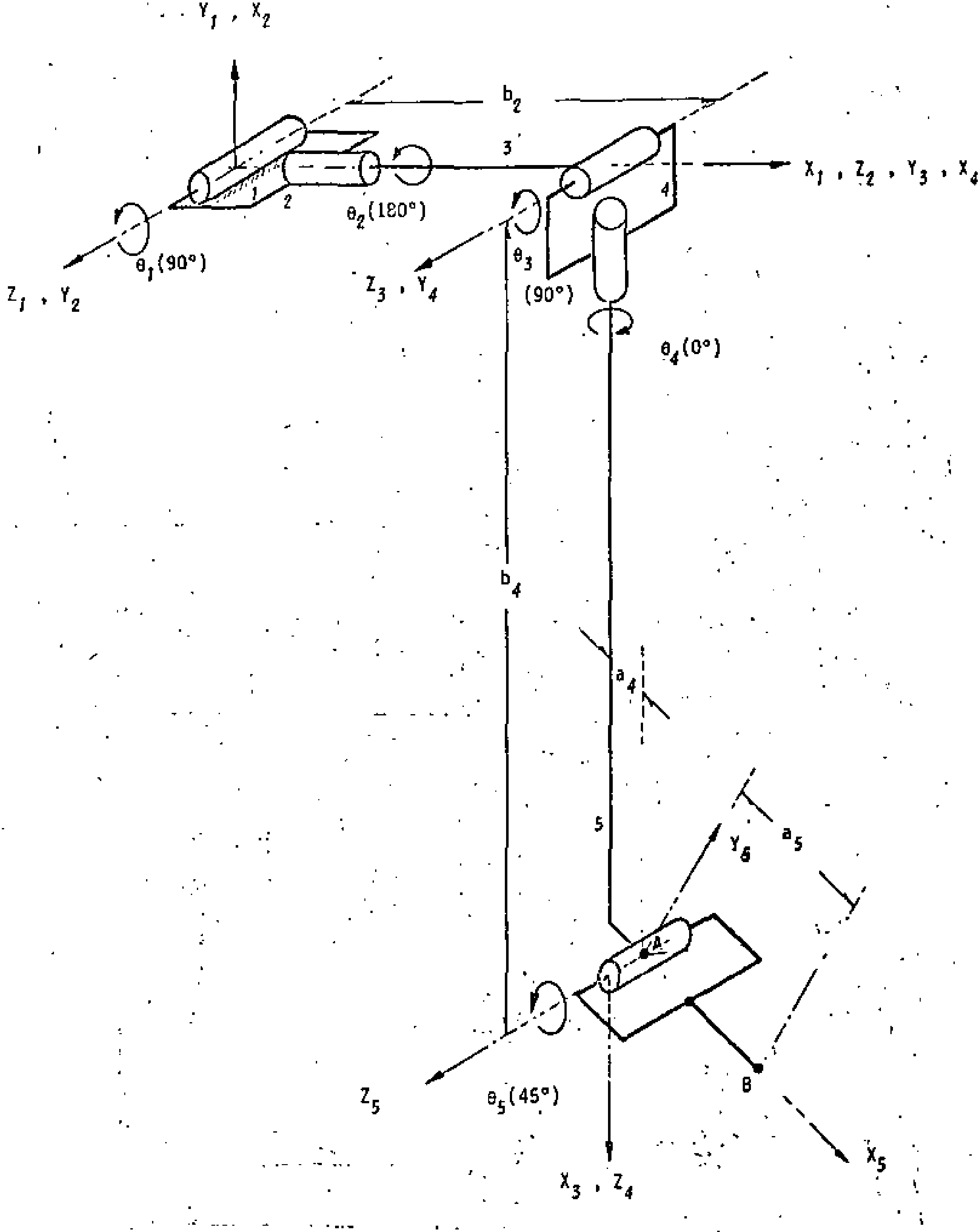


Fig 4 Fifth-degree-of-freedom manipulator.

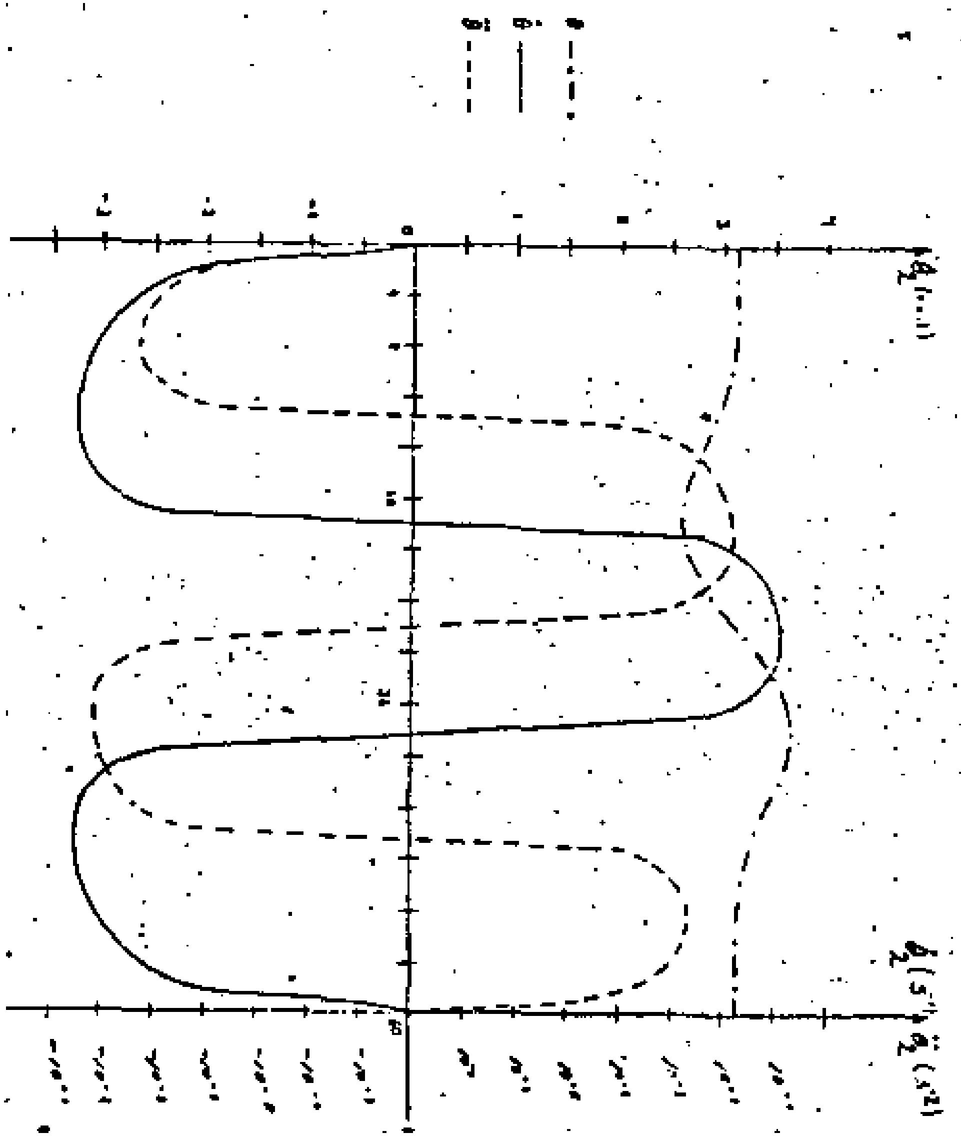


FIG 6

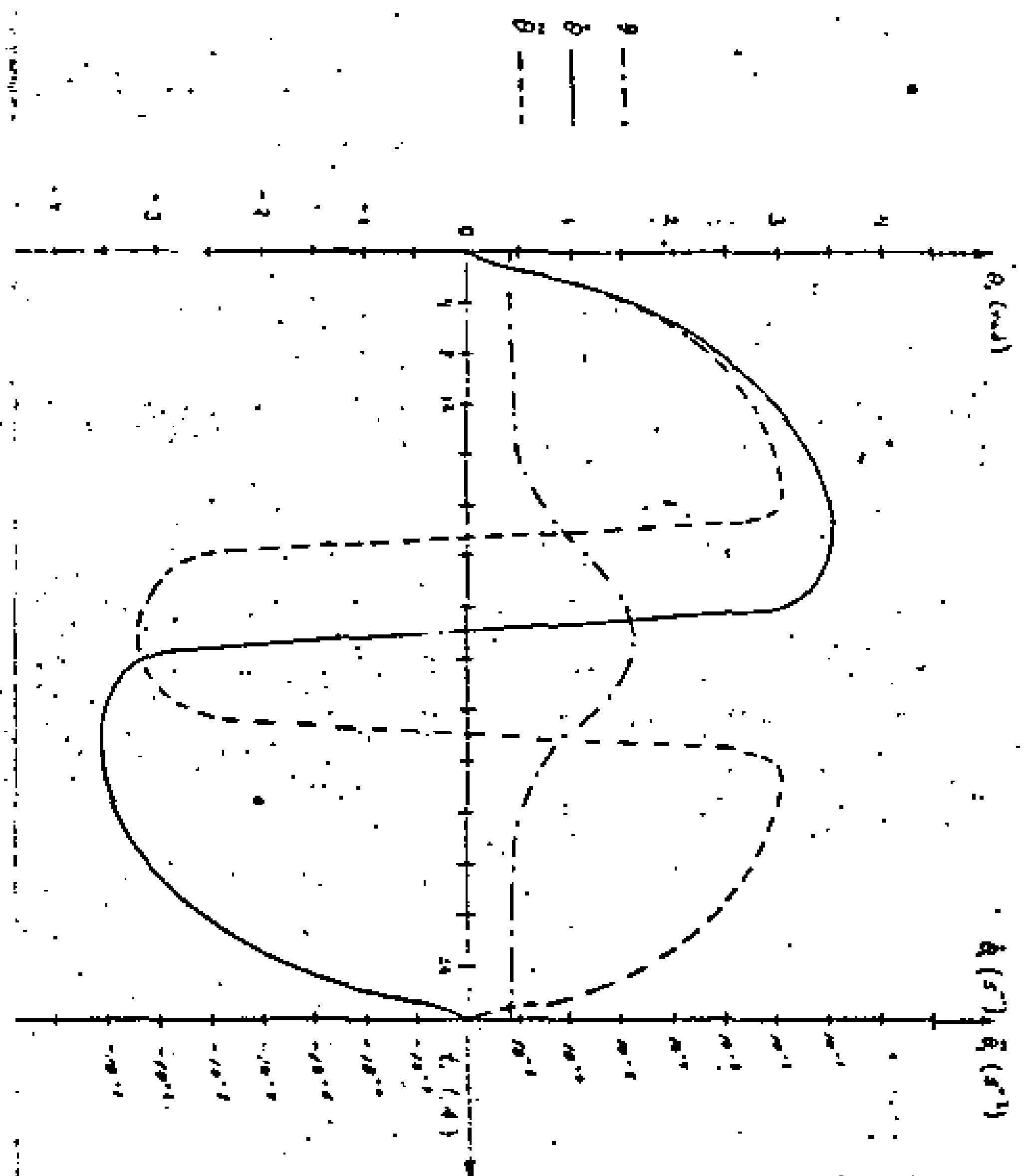


FIG 5

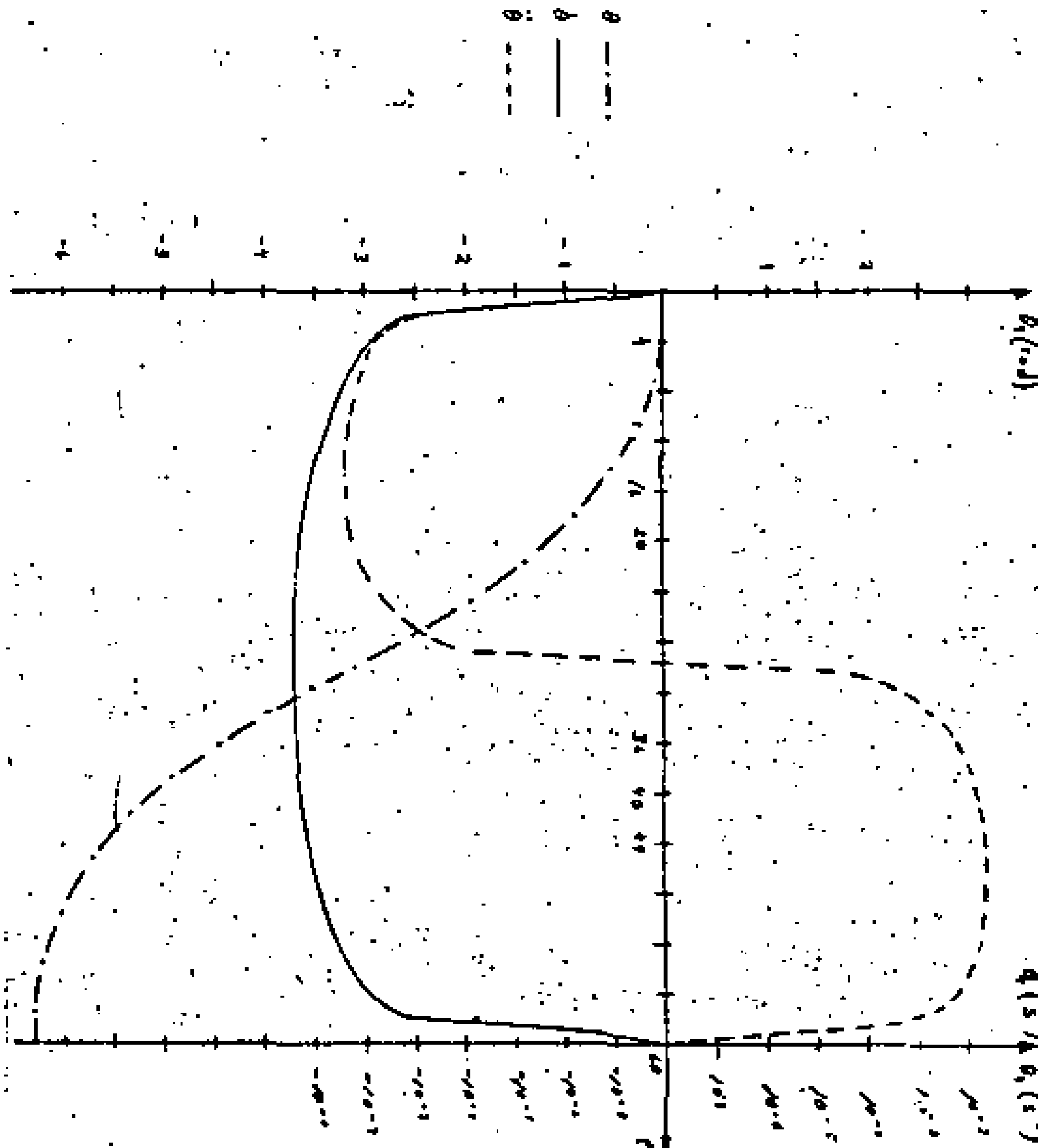


FIG 8

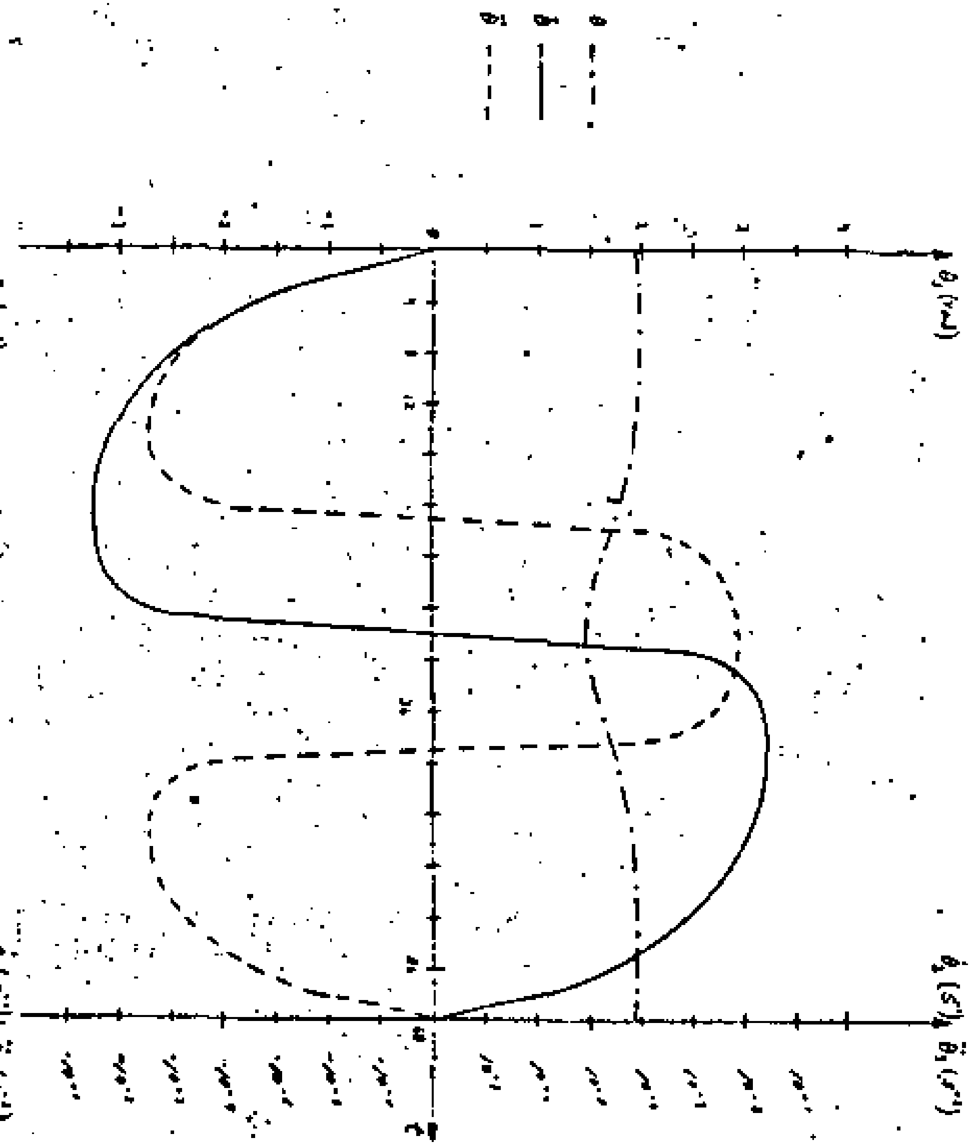


FIG 7

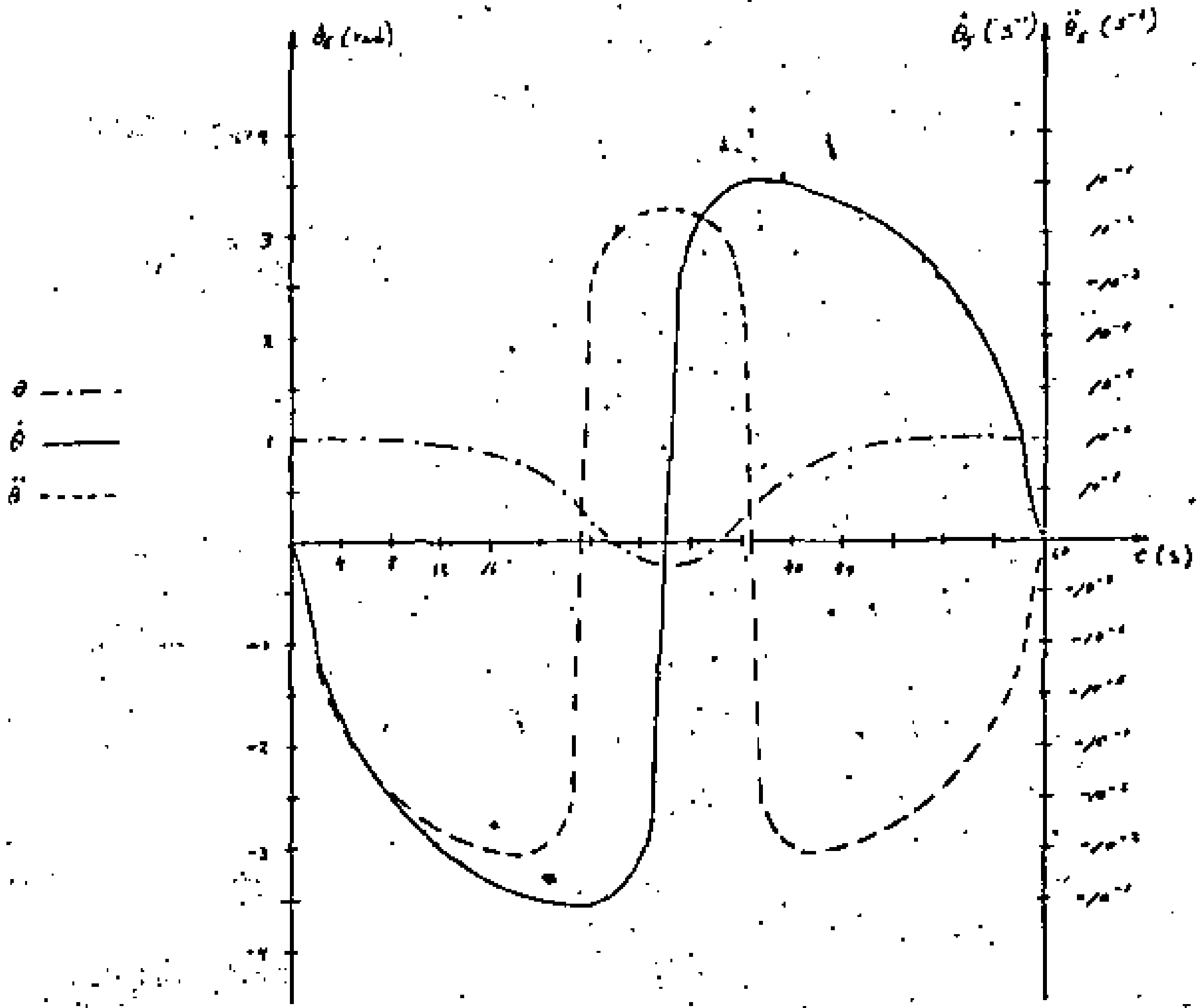


FIG 9

DISEÑO CINEMATICO DE MAQUINARIA

OPTIMAL SYNTHESIS OF TRANSLATIONS - FOLLOWER CAM MECHANISM
WITH PRESCRIBED FUNCTIONAL CONSTRAINTS

DR. JORGE ANGELES ALVAREZ
DR. CARLOS LOPEZ CAJUN

JUNIO, 1984.

OPTIMAL SYNTHESIS OF TRANSLATING-FOLLOWER CAM MECHANISMS WITH PRESCRIBED FUNCTIONAL CONSTRAINTS

Jorge ANGELES and Carlos LOPEZ-CAJUN
Facultad de Ingeniería, Universidad Nacional
Autónoma de México, Apdo. Postal 70-256
04510 México, D.F, MEXICO.

Abstract

A novel approach is presented, that allows the computer synthesis of minimum-size cam profiles, which are to produce a given follower-displacement programme while meeting prescribed functional constraints. These can be imposed either on the pressure angle or on the eccentricity of the contact point depending upon the type of follower that is being synthesised. The paper is limited to the synthesis of cam profiles for translating roller followers, but the same approach has been successfully applied to flat-face followers, as reported previously. The results obtained for the example included show the applicability of the procedure to the automatic design of this type of cam follower mechanisms.

Keywords; optimal synthesis, cam mechanisms, optimum design, automatic design

1. INTRODUCTION

The synthesis of cam mechanisms to produce a given displacement programme of the follower is divided into two stages [1]. The first one is meant to produce an easy-to-handle smooth-enough function of the angle of rotation of the cam disk, representing the displacement of the follower. The second one deals with the synthesis of the cam profile. The synthesis of the displacement programme has been based traditionally on a limited set of functions giving rise to parabolic, harmonic, cycloidal, trapezoidal and polynomial motions. Based on the last type, Thompoulus and Knowles [2] proposed a novel method of displacement-programme synthesis using linear programming, but it is not until very recently that a totally new approach has been introduced using spline functions [3,4].

As to the profile synthesis, the traditional approach, based on graphical methods, has been abandoned in favour of numerical methods, that have been called for given the extensive use of computers and NC-machine tools [5]. The literature on graphical methods of profile synthesis is rather broad, as can be seen in [6], that on computer-oriented methods becoming abounding, particularly since the 1970's [6-14]. The approach introduced in this paper, regarding the optimal synthesis of cam profiles, is that of mathematical programming, but, as seen in the discussion, it does not require the application of sophisticated and time-consuming optimisation methods.

2. THE FOLLOWER-DISPLACEMENT PROGRAMME

In this Section a method is described that allows to synthesise follower-displacement programmes, starting from an acceleration programme that is specified over a discrete set of values, $\{\psi_i\}_1^n$ of the angle ψ measuring the rotation of the disk cam involved. Let the displacement of the follower be given by $c+s(\psi)$, where c is a constant representing the minimum value of that displacement, $s(\psi)$ being a positive definite function whose minimum value is zero, its maximum value being h, the rise of the follower. Constant c is computed from considerations concerning the profile synthesis, as will be shown later, the objective of this Section being the synthesis of $s(\psi)$. This will be performed using a section of a *basic function* $\tau(\psi)$ which is synthesised using spline functions as shown in [4]. $\tau(\psi)$ is synthesised by prescribing a set of acceleration values*, $\{\tau''(\psi_i)\}_1^n \equiv \{\tau''_i\}_1^n$, so that $\tau''_1=\tau''_n$, in order to obtain a periodic function $\tau(\psi)$. To this end, periodic splines are used. Thus, the set $\{\tau_i\}_1^n \equiv \{\tau(\psi_i)\}_1^n$ is computed by solving a linear system of equations. For instance, τ''_i, $i=1,2,\ldots,n$ can be specified as having an harmonic distribution, vanishing at ψ_1 and ψ_n, the extremes of the interval of interest, which can be specified in turn as $[0,2\pi]$. Moreover, τ'' is also specified to vanish at inner points of this interval, in order to ensure that the displacement function will have a zero acceleration at its highest point, the one for which the displacement attains the value $c+h$. Furthermore, in order to meet the condition that the velocity be a continuous function of ψ, symmetries are introduced that guarantee the vanishing of the velocity at the two dwell intervals of the follower. Thus τ''_i is specified such that $\tau''_j=0$, the whole function, shown in Fig. 1, being assumed to be periodic and symmetric with respect

* If $c+s(\psi)$ represents the displacement of the follower then its velocity and acceleration are given by $s'(\psi)\dot{\psi}$ and $s'(\psi)\ddot{\psi}+s''(\psi)\dot{\psi}^2$, respectively, $\dot{\psi}$ and $\ddot{\psi}$ being the velocity and the acceleration of the cam disk.

to line $\psi=\psi_{2m-1}=\pi$, the midpoint of the interval $[0,2\pi]$. If the function is further assumed to be odd with respect to lines $\psi=\psi_m=\pi/2$ and $\psi=\psi_{3m-2}=3\pi/2$, then τ'' will also vanish at B and C. Moreover, this way, τ' will also vanish at A, B and C. The *basic function* to produce $\sigma(\psi)$ can be obtained, in turn, from the curve shown in Fig.1 by shifting the ψ axis to the line $\tau=-\tau_0$. A proper way to prescribe τ''_i is as follows:

$$\tau''_i=k\sin\psi_i,\quad i=1,\ldots,m \tag{1a}$$

with

$$\psi_i=\frac{\pi}{2(m-1)}(i-1) \tag{1b}$$

where constant k is chosen so as to normalize τ, i.e, such that $\tau_0=1/2$. Using a periodic spline, with m=11, the curve appearing in Fig.1 was obtained with k=0.64191.

The displacement programme for the follower of a cam mechanism is now assumed to have a lower dwell in the interval $[0,\psi_1]$, then a rise in the interval $[\psi_1,\psi_2]$, next a higher dwell in the interval $[\psi_2,\psi_3]$ and, finally, a return in the interval $[\psi_3,2\pi]$. Let

$$\Delta\psi_1=\psi_1,\ \Delta\psi_2=\psi_2-\psi_1,\ \Delta\psi_3=\psi_3-\psi_2,\ \Delta\psi_4=2\pi-\psi_3 \tag{2a}$$

The foregoing displacement programme can now be synthesised from the *basic function* τ vs ψ of Fig.1 by properly scaling separately its intervals $[0,\pi]$. To this end, redefine ψ as

$$\psi:=\frac{\Delta\psi_2}{\pi}\psi,\quad 0\le\psi\le\pi \tag{2b}$$

$$\psi:=\frac{\Delta\psi_4}{\pi}(\psi-\pi),\quad \pi\le\psi\le 2\pi \tag{2c}$$

Now function $\sigma(\psi)$ is defined as

$$\sigma(\psi)=0,\quad 0\le\psi\le\psi_1 \tag{3a}$$
$$\sigma(\psi)=h\tau(\psi),\quad \psi_1\le\psi\le\psi_2 \tag{3b}$$
$$\sigma(\psi)=h,\quad \psi_2\le\psi\le\psi_3 \tag{3c}$$
$$\sigma(\psi)=h\tau(\psi),\quad \psi_3\le\psi\le 2\pi \tag{3d}$$

A subprogram was written that allows the automatic synthesis of follower-displacement programmes by inputting the following data: ψ_1, ψ_2, ψ_3 and h. Its output is $\sigma(\psi)$, $\sigma'(\psi)$ and $\sigma''(\psi)$. The procedure introduced to compute the *basic function* $\tau(\psi)$, on the other hand, need not be changed if an acceleration distribution different from the one given in (1a&b) is chosen. In fact, only the set $\{\tau''_i\}_1^m$ has to be changed as data input to the spline subroutines synthesising function $\tau(\psi)$. Moreover, if the follower is of the translating type, $\sigma(\psi)$ and hence c and h will have units of length. They will be measured in radians if the follower is of the oscillating type. The synthesis of the cam profile for translating followers of the roller type is next discussed. That of the cam profile for translating followers of the flat-face type is reported elsewhere [15].

3. OPTIMAL SYNTHESIS OF CAM PROFILES FOR TRANSLATING ROLLER FOLLOWERS

In this Section it is assumed that the displacement programme is specified up to a constant c, i.e. function $\sigma(\psi)$ is known, whereas constant c is to be determined. Let the displacement of the follower be denoted by $s(\psi)$, i.e.

$$s(\psi)=c+\sigma(\psi) \tag{4}$$

and hence

$$s'(\psi)=\sigma'(\psi),\quad s''(\psi)=\sigma''(\psi) \tag{5}$$

Reference is made now to Fig.2, showing the layout of a cam with a translating knife-edge follower. The interest of synthesising such a cam profile lies in the fact that this is the *pitch curve* for a roller follower, the pressure angle for both types of followers being identical [16]. In Fig.2, line P is the path of the follower, lines T and N are the tangent and the normal to the cam profile, given by its polar equation $\rho=\rho(\theta)$, at the contact point Q. Moreover, lines OF and OC are fixed to the frame and to the cam disk, respectively, e being the eccentricity of the follower path. Angle α is the pressure angle. From the geometry of Fig. 2,

$$s(\psi)=\rho(\theta)\sin(\theta+\psi) \tag{6a}$$
$$e=\rho(\theta)\cos(\theta+\psi) \tag{6b}$$
$$\alpha=\theta+\psi+\phi-\pi \tag{6c}$$

where

$$\phi=\tan^{-1}\frac{\rho(\theta)}{\rho'(\theta)} \tag{6d}$$

In order to keep the mechanical advantage of the mechanism within acceptable limits, α is usually bounded properly. Since α can be either positive or negative, its absolute value is bounded as

$$|\alpha|\le\alpha_M \tag{7}$$

where α_M is normally chosen to lie "close" to 0°, for a value of 90° would render the mechanical advantage zero. A value that is widely accepted is 30°. Finding the value of ψ at which $|\alpha|$ attains its extrema is not so simple, for this function is not differentiable at the origin, where it attains its minimum. Hence a different even function of α that be smooth enough has to be extremised. A good candidate is $\cos\alpha$. Using relations (6a&d), this turns out to be

$$\cos\alpha=\frac{s(\psi)}{\{[s'(\psi)-e]^2+s^2(\psi)\}^{1/2}} \tag{8}$$

Now $\cos\alpha$ will be kept within bounds as

$$\cos\alpha\ge\cos\alpha_M\equiv c_M \tag{9}$$

i.e.

$$\min_{\psi}(\cos\alpha)=c_M \tag{10}$$

The extrema of (8) are now found by zeroing its derivative with respect to ψ. This is readily obtained as*

$$\frac{d\cos\alpha}{d\psi}=\frac{(s'-e)[(s'-e)s'-ss'']}{\{(s'-e)^2+s^2\}^{3/2}} \tag{11}$$

which vanishes under either of the next two conditions:

i) $s'=e$ (12a)

ii) $(s'-e)s'=ss''$ or $\frac{s'-e}{s}=\frac{s''}{s'}$ (12b)

Condition i) leads to the maximum +1, as can be readily verified from eq(8), whereas condition ii) leads to the minimum c_M. Let ψ_0 be the value of ψ, not as yet determined, producing the minimum. From eq(8), then

* Henceforth, a prime on a variable means its derivative with respect to ψ, i.e. $s'\equiv ds/d\psi$. Similarly, $s''\equiv d^2s/d\psi^2$.

$$\left|\frac{s_0'-e}{s_0}\right| = \tan\alpha_M \qquad (13)$$

with s_0, s_0' and s_0'' defined as the values attained by $s(\psi)$, $s'(\psi)$ and $s''(\psi)$ at $\psi=\psi_0$. Relation (13) was already obtained by Chicurel [17]. Since $s'(\psi)=\sigma'(\psi)$ and $s''(\psi)=\sigma''(\psi)$, eqs(12b) and (13) lead to

$$\left|\frac{\sigma''(\psi_0)}{\sigma'(\psi_0)}\right| = \tan\alpha_M \qquad (14a)$$

or

$$|\sigma''(\psi_0)| = \tan\alpha_M |\sigma'(\psi_0)| \qquad (14b)$$

Substitution of eqs(4&5), evaluated at $\psi=\psi_0$, into eq(13) and the assumed positive definiteness of $\sigma(\psi)$ and c lead to

$$|\sigma_0'-e| = \tan\alpha_M(c+\sigma_0) \qquad (15a)$$

On the other hand, during the rise phase the follower velocity is positive, as shown in Fig.3. From those plots it can also be seen that eq(14b) holds at two distinct values of σ'', namely σ_1'' and σ_2''. The positive value leads to $\sigma=\sigma_0^{I}$ whereas the negative one leads to $\sigma=\sigma_0^{II} > \sigma_0^{I}$. Substitution of these values into eq(15a), together with the condition

$$c > 0 \qquad (15b)$$

define the set of values along which condition (10) holds during the rise phase. This is plotted in Fig.4. In that figure, this set is formed by the two branches of lines 1 and 2 making an angle α_M with either direction of the c axis, the dashed section being excluded, for it violates (15b).

Since $\cos\alpha$ also attains stationary values at the lowest follower position (i.e. during the dwell phase D_1), the following must hold for any set of values (c,e):

$$e = c\tan\alpha_M \qquad (16)$$

Eq(16) is also plotted in Fig.4. From that figure, it is clear that points P and Q satisfy eqs(15a&b and 16) simultaneously. Taking the radius of the base circle, r, as a measure of the area, which is plausible because r is the lowest value of $\rho(\theta)$, the distance OP leads to the minimum value of r.

To this end it was assumed that ψ_0 was known. In fact, from the foregoing discussion, such value of ψ, satisfying eq(14b), corresponds to the one which zeroes the function

$$f(\psi)=\sigma''(\psi)+\tan\alpha_M\sigma'(\psi) \qquad (17)$$

This function is plotted in Fig.5.

Values of ψ_0 for different pressure angles were obtained and plotted in Fig.6. From that plot the designer can select the appropriate percentage for calculating ψ_0 for the problem at hand.

Once the value of ψ_0 has been determined one can proceed to determining the values of σ_0, σ_0' and σ_0'' and thence the values of c_0 and e_0 which in turn will determine the minimum radius of the base circle. From Fig.4,

$$c_0 = \frac{1}{2}\,\overline{OA}\,, \quad e_0 = \frac{1}{2}\,\overline{OB} \qquad (18a)$$

where

$$\overline{OA} = \frac{\sigma_0'}{\tan\alpha_M} - \sigma_0 \qquad (18b)$$

and

$$\overline{OB} = \sigma_0' - \sigma_0\tan\alpha_M \qquad (18c)$$

The cam profile is now readily obtained by combining eqs(6a&b) to yield

$$\rho = [s^2(\psi)+e^2]^{1/2} \qquad (19a)$$

which produces a value of ρ for each given value of ψ, i.e. it produces the function $\rho=\rho(\psi)$. In order to obtain the function $\theta=\theta(\psi)$, eqs(6a&b) are again combined to produce

$$\theta = \tan^{-1}\left[\frac{s(\psi)}{e}\right]-\psi \qquad (19b)$$

which thus produces a function $\theta=\theta(\psi)$.

Regarding now ψ as a parameter between functions $\rho(\psi)$ and $\theta(\psi)$, the function $\rho=\rho(\theta)$ is readily obtained. The foregoing procedure clearly produces a discrete set of n pairs $\{(\rho_i,\theta_i)\}_1^n$, which then have to be interpolated in order to produce a continuous function $\rho=\rho(\theta)$ that would allow both the plotting of the cam profile and the punching of the tape guiding a NC-machine tool used for its automatic production. Out of the different procedures to interpolate the obtained set, the one that is proposed here is based upon periodic parametric splines, which have proved [4,18,pp.27-28] to have the following advantages:

i) they require a low number (small value of n) of sample values $\{(\rho_i,\theta_i)\}_1^n$ to produce a given imposed accuracy
ii) their parameters are readily computed by solving a system of linear equations that is symmetric, tridiagonal and diagonally dominant
iii) as a consequence of ii), the arising system of equations is well conditioned.

A subprogram was written that, for given values of α_M, σ_0 and σ_0', produces the optimising values e_0 and c_0 given by eqs(18a) and for a given value of n, produces the set $\{(\rho_i,\theta_i)\}_1^n$, which is used next to interpolate the cam profile with periodic parametric splines. The subprogram produces also the interpolated values of e_t, e_n and r_c representing the unit tangent and normal vectors to the cam profile and its radius of curvature, respectively. Now, specifying a value d of the radius of the roller, the profile of the corresponding cam can be obtained from

$$r_r = r_k - d\,e_n \qquad (20)$$

where r_r and r_k are the position vectors of points on the cam profile for the roller and for the knife-edge followers, respectively. This way, knowing the cam profile for the knife-edge follower, as well as the radius d of the roller, eq(20) allows to obtain the cam profile for the roller follower. This would complete the optimal synthesis of the cam profile sought, if no additional constraints were to be imposed. There are, however, two more items that need be taken into account, namely i) the maximum allowable value of the radius of the roller, d, to avoid the phenomenon known as *undercutting*, and ii) the minimum allowable value of the radius of curvature of the cam profile for the roller follower, r_{cr}, which is necessary to specify in order to avoid too large values of the contact stress.

Let κ_k and κ_r be the curvature of the cam profile for the knife-edge follower and for the roller follower, respectively. Since the unit tangent vectors to both profiles are identical, one readily obtains

$$\kappa_r = \frac{\kappa_k}{1-a\kappa_k} \tag{21}$$

If the denominator of the right-hand side of eq(21) vanishes, the curvature of the roller-follower cam profile, κ_r, will become infinity. This means that, at values of ψ where that denominator vanishes, the said cam profile has a cusp, which defect is known as *undercutting*. Moreover, both the *pitch curve* and the roller-follower cam profile should have, at corresponding points, i.e. at the same value of ψ, curvatures with the same sign. This is thus attained if and only if,

$$1-a\kappa_k>0,\quad 0\le\psi\le 2\pi \tag{22a}$$

Relation (22a) holds, in turn, if

$$a<(r_{ck})_{min} \tag{22b}$$

or, equivalently, if a is specified as a given fraction f of $(r_{ck})_{min}$, i.e.

$$a=f(r_{ck})_{min},\quad 0<f<1 \tag{22c}$$

The minimum value of r_{ck} is now computed. To this end, r'_{ck} is zeroed and the values of r_{ck} at stationary points are computed. It can be readily shown that r_{ck} attains stationary values at the dwell phases. These, however, are not, in general, global extrema, for which reason the global minimum is sought both in the rise and in the return phases. Both r_{ck} and $r'_{ck}(\psi)$ are given by [19]

$$r_{ck} = \frac{N(\psi)}{D(\psi)} \tag{23}$$

and

$$r'_{ck} = \frac{1}{D(\psi)}[N'(\psi)-r_{ck}D'(\psi)] \tag{24}$$

with

$$N(\psi) = [s^2+(s'-e)^2]^{3/2} \tag{25a}$$
$$D(\psi) = s^2+(s'-e)(2s'-e)-ss'' \tag{25b}$$
$$N'(\psi)=3[ss'+s''(s'-e)][s^2+(s'-e)^2]^{1/2} \tag{25c}$$
$$D'(\psi)=s(2s'-s''')+3s''(s'-e) \tag{25d}$$

The zeros of r'_{ck} can be readily computed with the aid of Subroutine ZEROIN [20] or any other efficient subprogram intended to find the zeros of a nonlinear function of a real argument. The global minimum of r_{ck} is then substituted into eq(22c), thus producing the desired value of a, thereby completing the synthesis of the proposed mechanism. The software realising the described synthesis produces values of technical interest concerning the cam profile, as well. These are: area, location of its centroid, principal moments of inertia and orientation of the principal axes of inertia. All these values are produced in order to ease the static and dynamic analysis of the overall mechanism.

At this point it is worth mentioning the impossibility of lines 1 and 2, Fig.4, of passing through the origin, which in turn, would lead to $r=0$. In fact, this would imply

$$\sigma_0=\sigma'_0\tan\mu_H$$

However, from eq(14a) and taking into account the positive definiteness of σ'_0 during the rise phase, one has

$$\sigma_0=|\sigma''_0| \tag{27}$$

Without loss of generality, one can assume that ξ_0 is a supporting point, ξ_i, of the spline curve. From the relationships between the sets $\{\sigma_i\}_1^N$, $\{\sigma'_i\}_1^N$ and $\{\sigma''_i\}_1^N$ for cubic splines [18, pp. 27-29], eq(27) would lead to

$$\sigma''_4=\sigma''_{4+l} \tag{28}$$

which is impossible to happen, given the way the set $\{\sigma''_i\}_1^N$ was specified in eqs(1a&b).

In the case of a cam with a radial follower, $e=0$, and $\cos\mu$ attains stationary values either when $s'=0$ or else when $s'^2=ss''$. The first condition leads to a minimum value of the pressure angle, whereas the second one leads to the maximum. Recalling that $s=\sigma$, $s'=\sigma'$ and $s''=\sigma''$, one has

$$\sigma'^2=(\sigma+e)\sigma'' \text{ or } \frac{\sigma'}{\sigma+e}=\frac{\sigma''}{\sigma'} \tag{29}$$

Substitution of eq(14a) into eq(29) and the fact that $\sigma'>0$ during the rise phase lead to

$$\sigma'_0=\tan\mu_H(\sigma_0+e) \tag{30a}$$

where σ_0, σ'_0 and σ''_0 are defined as before. From eq(30a) one obtains the radius of the base circle as

$$r_0=\frac{\sigma'_0-\sigma_0\tan\mu_H}{\tan\mu_H} \tag{30b}$$

4. EXAMPLE

Obtain the cam profile of the translating roller-follower cam mechanism producing the displacement programme given in Table 1. The maximum pressure angle is limited to 30°, and $f=0.50$. The subprogram produced the cam profile appearing in Fig.7, with a value of a of 16.30mm.

Phase	Angle of rotation (ψ)	Displacement (mm)
D_1 (dwell)	36°	0
R^1 (rise)	72°	+50.0
D_2 (dwell)	144°	0
R^2 (return)	108°	-50.0

Table 1 Follower-displacement programme

The geometric properties of the cam profile produced by the subprogram are:

Area=19530.36mm² Centroid=(-10.89,-19.53), E_1 and E_2 are the principal axes of inertia at the centroid, the corresponding moments of inertia being I_1=40192170.0mm⁴ and I_2=33070254.0mm⁴.

5. CONCLUSIONS

An automatic procedure, implemented with the aid of several computer subprograms, was developed. This procedure allows the digital and graphical production of the minimum-size cam profile that generates a given displacement programme for a roller follower while observing bounds on the pressure angle. Radial and offset followers were considered herein. The software presented here yields, additionally, the radius of the roller as a fraction of the minimum value of the radius of curvature of the pitch curve, in order to avoid undercutting. Moreover, it produces relevant geometric properties of the profile, such as its area, its centroid location, the orientation of its principal axes and values of its principal moments of inertia. The software is user oriented and requires no deep knowledge of the algorithm described herein. It is a part of a wider program system intended for the synthesis of cam profiles of various types of followers, out of which the one corresponding to flat-face followers was presented previously.

6. ACKNOWLEDGEMENTS

This work was realized at the CAD Laboratory of the Graduate Division of the Faculty of Engineering, UNAM. The support of this Faculty is gratefully acknowledged.

7. REFERENCES

[1] Rothbart, H.A: Cams. John Wiley, N.Y., 1956.

[2] Thompoulos, N.T. and Knowles, T.W: Use of linear programming for cam design. Int. J. Mach. Des. Tool & Res. Vol.15, pp.257-265(1975).

[3] García de Jalón, J. and Nó Sánchez, M: Application of b-spline functions to the motion specification of cams. ASME paper 80-DET-28 (1980).

[4] Angeles, J: Synthesis of plane curves with prescribed local geometric properties using periodic splines. Computer-Aided Design. Vol. 15, pp.147-155(1983).

[5] Grant, B. and Soni, A.H: A survey of cam manufacture methods. Trans. ASME J. Mech. Des. Vol. 101, pp.455-464(1979).

[6] Chen, F.Y: A survey of the state of the art of cam system dynamics. Mechanism and Machine Theory. Vol.12, pp.201-224(1977).

[7] Mischke, C.: Optimal offset on translating follower plate cams. Trans. ASME J. Engng. Ind. Vol. 92, pp.172-176(1970).

[8] Buchsbaum, F. and Freudenstein, F: On a class of cam-type angular-motion compensators. Trans. ASME J. Engng. Ind. Vol.95, pp.487-496 (1973).

[9] Chen, F.Y: Kinematic synthesis of cam profiles for prescribed acceleration by a finite integration method. Trans. ASME J. Engng. Ind. Vol. 95, pp.519-524(1973).

[10] Chen, F.Y: Analysis and design of cam-driven mechanisms with non-linearities. Trans. ASME J. Engng. Ind. Vol.95, pp.685-694(1973).

[11] Fagel, P.A. and Reiss, J.J: The building block method of cam design. ASME paper 76-DET-36(1976).

[12] Weber, Jr. T: The precision of manufacturing of cams. ASME paper 76-DET-37(1976).

[13] Angeles, J. and Arteaga, O: Optimal synthesis of cam mechanisms via the methods of Newton-Raphson and Runge-Kutta. Proceedings of the Second IFToMM International Symposium on Linkages and Computer-Aided Design Methods. Bucharest, Vol.III-1, pp.1-12(1977).

[14] Gantier, M.A. and Uicker, Jr. J.J: Design charts for disk cams with reciprocating roller followers. Trans. ASME J. Mech. Des. Vol.101, pp. 465-470(1979).

[15] Angeles, J. and López-Cajún, C: Diseño automatizado de mecanismos de leva de disco con seguidor traslacional de cara plana. Memoria del IX Congreso de la AMIAC, México, pp.91-93(1983).

[16] Mabie, H.H. and Ocvirk, F.W: Mechanisms and Dynamics of Machinery. John Wiley, N.Y. 1975.

[17] Chicurel, R: Cam size minimization by offsetting. ASME paper 62-[illegible]-77(1962).

[18] Späth, H: Spline-Algorithmen zur Konstruktion glatter Kurven und Flächen. R. Oldenburg, Munich, 1976.

[19] Terauchi, Y. and El-Shakery, S.A: A computer aided method for optimum design of plate cam size avoiding undercutting and separation phenomena-I. Mechanism and Machine Theory. Vol.18, pp.157-163(1983).

[20] Forsythe, G.E., Malcolm M.A. and Moler, C.B: Computer Methods for Mathematical Computations. Prentice-Hall, N.J., 1977.

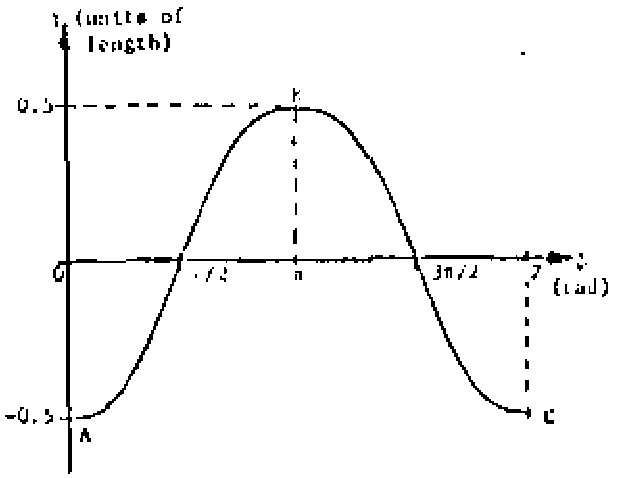


Fig.1 Periodic function generating the follower-displacement programme

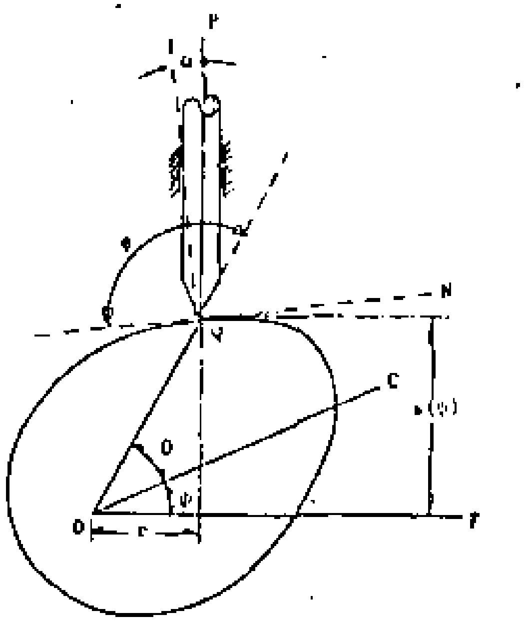


Fig.2 Geometry of a cam mechanism with a translating knife-edge follower

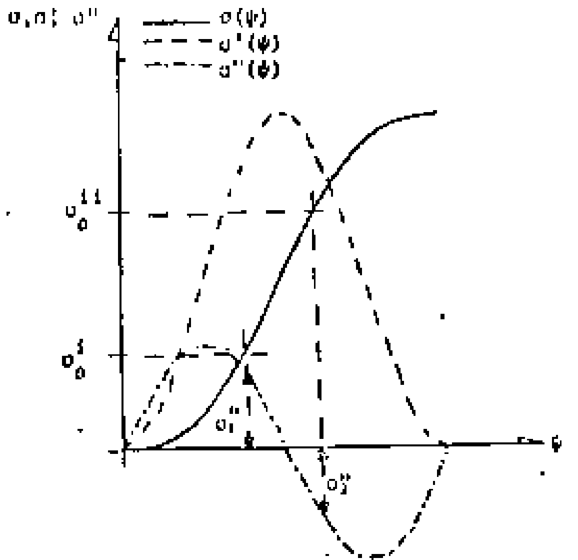


Fig.3 Displacement, velocity and acceleration programmes

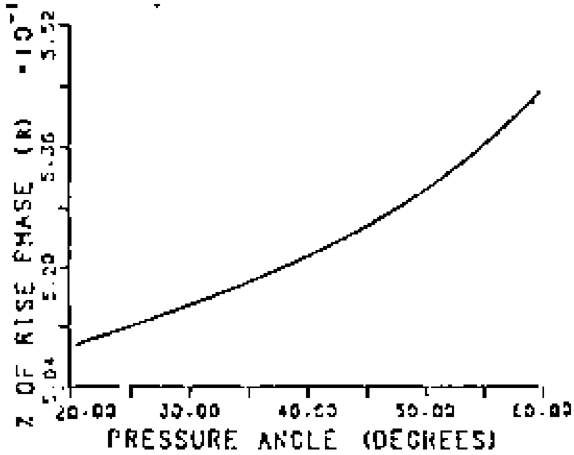


Fig.6 % of the rise phase vs. pressure angle

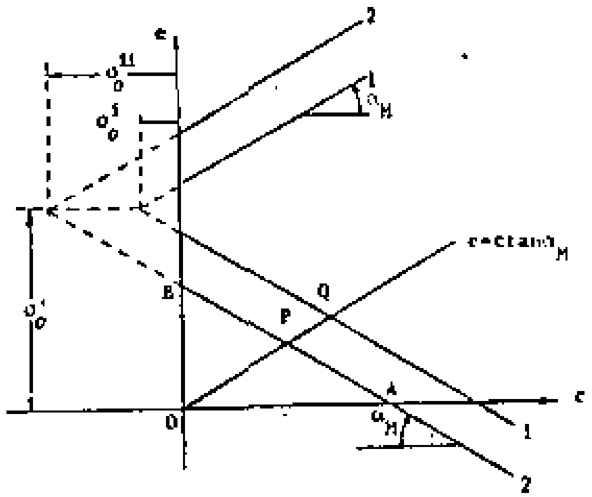


Fig.4 Plots of eqs(15&16)

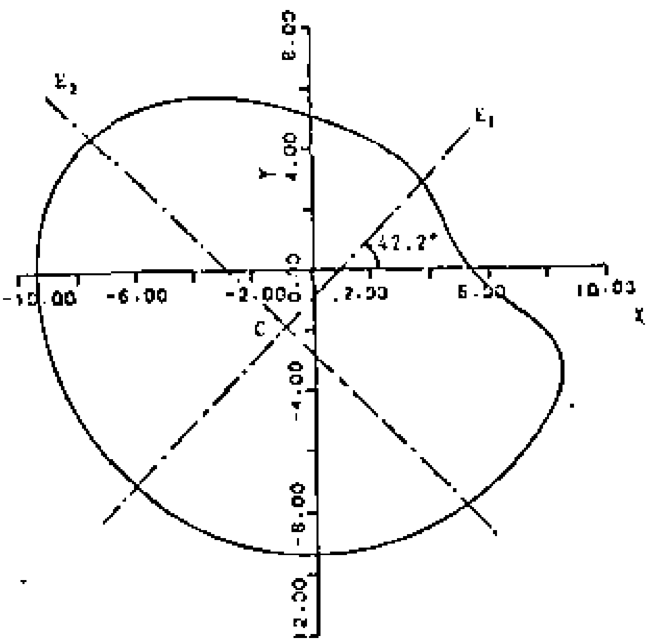


Fig.7 Cam profile

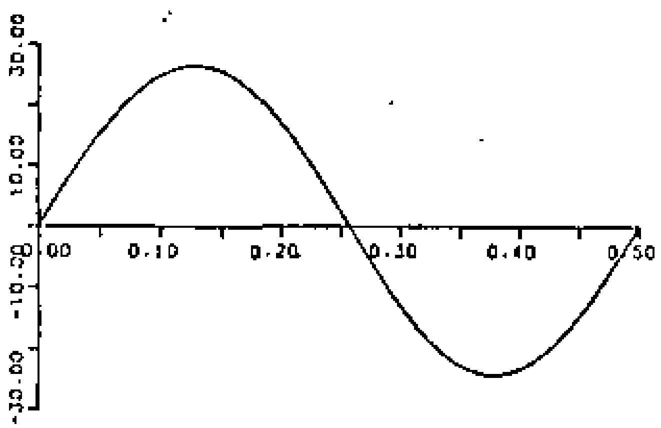


Fig.5 Plot of eq(17)

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO CINEMATICO DE MAQUINARIA

OPTIMAL SYNTHESIS OF OSCILLATIONS ROLLER-FOLLOWER CAM MECHANISMS WITH PRESCRIBED FUNCTIONAL CONSTRAINTS

DR. JORGE ANGELES ALVAREZ

DR. CARLOS LOPEZ CAJUN

JUNIO, 1984.

OPTIMAL SYNTHESIS OF OSCILLATING ROLLER-FOLLOWER CAM MECHANISMS WITH PRESCRIBED FUNCTIONAL CONSTRAINTS

Jorge Angeles and Carlos López-Cajún
Facultad de Ingeniería, Universidad Nacional
Autónoma de México
Apdo. Postal 70-256
04510 México, D.F.
MEXICO.

ABSTRACT

The synthesis of the cam profile producing a given displacement program of its oscillating roller follower, while enclosing a minimum area and having a prescribed maximum pressure angle, is presented. The displacement program of the follower is synthesized, in turn, using cubic periodic splines. While the mathematical programming approach is followed throughout, it is shown that the problem can be solved using a simple root-finding routine applied to a nonlinear equation in one unknown. The procedure is illustrated with a fully-solved example showing its applicability to the automatic design of cam-follower mechanisms.

INTRODUCTION

While the literature on cam mechanisms has widespread since the 1970's, incorporating computer-oriented methods of analysis, synthesis and manufacture [1-17], cam mechanisms with oscillating followers have received very little attention, except for [2] and some other references not fully devoted to this type of mechanisms. This paper follows two previous ones [18,19] concerning the optimal synthesis of cam mechanisms with translating flat-face and roller followers. The approach introduced in those papers is now applied to the synthesis of disk cams with oscillating roller followers producing the desired angular motion of the follower, while enclosing a minimum area and having a prescribed maximum pressure angle. Departing from the usual practice of prescribing the follower motion via harmonic, cycloidal or polynomial functions, all of which contain a limited number of free parameters whose values are chosen so as to match the different motion phases (lower dwell, rise, upper dwell and return), the authors use cubic periodic splines. To this end, the concept of function and curve synthesis introduced in [20,21] is resorted to. This way, a computer program interacting with the user enables the latter to obtain a smooth follower motion with the desired phases. A complete turn of the cam plate is thus divided into the four phases having angular lengths $\Delta\psi_i$, for $i=1,2,3,4$. The user can supply these lengths either in degrees or as percentages of the total turn. In order to offer the user a visual verification of the follower displacement program, the software graphical capabilities provide a display either on a CRT or on a plotter. Once the follower-displacement program is synthesized, the method presented here proceeds to determine the cam disk position at which the pressure angle attains its maximum absolute value. This is done by finding the roots of a nonlinear equation in one single unknown, namely the variable defining the cam disk position. This value then provides the geometric parameters of the cam mechanism. All in all, a visual verification of intermediate results is possible, but the software is so designed as to enable a totally automatic mode of operation.

SYNTHESIS OF THE FOLLOWER-DISPLACEMENT PROGRAM

The angular displacements of the cam plate and the follower are denoted by ψ and ϕ, respectively. Moreover, ϕ is assumed to be the sum of a constant c, as yet to be determined, and a positive definite function $\sigma(\psi)$, whose minimum value is 0, its maximum value being A, the amplitude of the follower oscillation, i.e.

$$\phi(\psi) = c + \sigma(\psi) \qquad (1)$$

Function $\sigma(\psi)$ is synthesized using section PQ of the cubic periodic spline $\tau(\psi)$ shown in Fig 1. This is

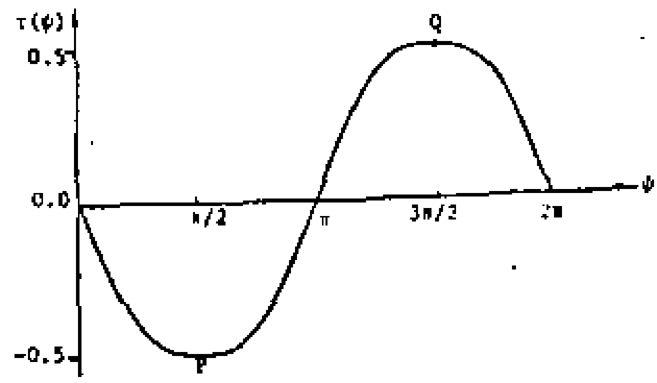


Fig. 1 Periodic spline curve

synthesized, in turn, by specifying a harmonic distribution of $\tau''(\psi)$ at a set of equally spaced n points in the interval $[0,2\pi]$. (The relationship between $\tau''(\psi_i)=\tau_i''$ and $\tau(\psi_i)=\tau_i$, for $i=1,2,\ldots,n$ is linear [22]. Hence, the unknown ordinates of the supporting points of the spline can be obtained as the solution to a linear system of equations, as shown in [20&21]. Moreover, in order to obtain a vanishing slope of the resulting spline, at both P and Q, $\tau(\psi)$ is prescribed to be odd with respect to $\psi=\pi$ and even with respect to $\psi=\pi/2$, $3\pi/2$. The introduction of the said symmetries results in a smaller number of independent ordinates τ_i, in fact, only $(n+3)/4$. The resulting system of equations is associated to a $m\times m$ matrix that is symmetric, positive definite and tridiagonal, all properties of which render it well conditioned and very simple to handle in obtaining the unknown ordinates. A proper scaling of the section PQ of Fig 1, plus a rigid body translation permit the obtention of the rise phase, R_1, of function $\sigma(\psi)$, of Fig 2. Finally, the return phase, R_2, of $\sigma(\psi)$ is synthesized by first reflecting section PQ of $\tau(\psi)$ with respect to $\psi=\pi$, then scaling it and shifting it exactly as in synthesizing R_1. Constant c of function $\phi(\psi)$ is next determined so as to produce a prescribed maximum pressure angle α_M, while making the area of the cam disk a minimum. This procedure is outlined in next Section.

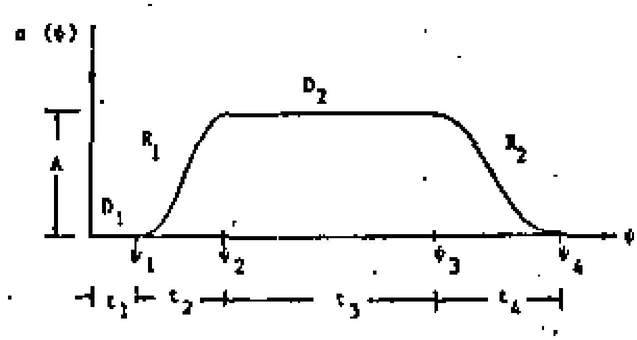


Fig.2 Follower displacement program

SYNTHESIS OF THE CAM PROFILE

The layout of the mechanism that is being synthesized is shown in Fig 3. From that figure, the following relations are readily derived:

$$\rho^2=a^2+b^2-2ab\cos\phi \tag{2}$$

$$\theta=\sin^{-1}\left[\frac{b}{\rho}\sin\phi\right]-\psi \tag{3}$$

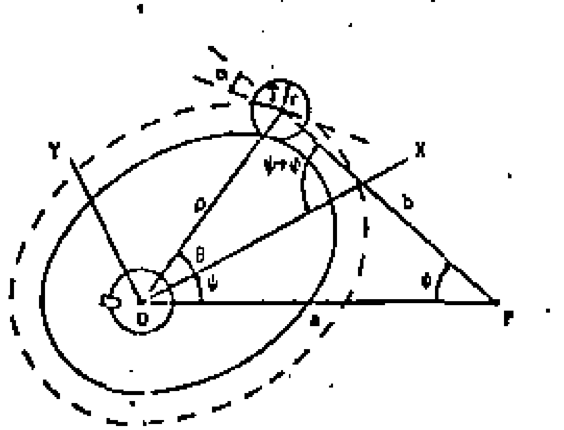


Fig 3 Cam mechanism with oscillating roller-follower

In order to obtain a good torque transmission, the pressure angle α, shown in Fig 3, is bounded as

$$|\alpha| < \alpha_M \tag{4}$$

However, determining the exact position at which $|\alpha|$ attains its maximum value is not straightforward, given the lack of smoothness of the absolute-value function at the origin. Hence, the extrema of an even function of α that be smooth enough are sought. As already done in [18&19], the values ψ_0 of ψ, at which cosα attains its minima are now found. From Fig 3, cosα is obtained from the inner product of the vector connecting P with T and the unit tangent to the pitch curve shown dotted in that figure. This is, in turn, the trajectory traced by the center of the roller on the cam disk. Thus,

$$\cos\alpha=-\frac{\sin\phi}{[1+\beta^2(1+\phi')^2-2\beta(1+\phi')\cos\phi]^{1/2}} \tag{5}$$

with

$$\beta=\frac{b}{a} \tag{6}$$

At this moment it is pointed out that the radical of the right-hand side of eq(5) is positive definite. In fact it equals the squared length of a triangle having sides 1 and $\beta(1+\phi')$, both making an angle ϕ.

Zeroing of $d\cos\alpha/d\phi$ leads to

$$C\beta^2+D\beta+E=0 \tag{7a}$$

where

$$C=\phi'(1+\phi')^2\cos\phi-\phi''(1+\phi')\sin\phi \tag{7b}$$

$$D=\phi'(1+\phi')(2\cos^2\phi+\sin^2\phi)-\phi''\sin\phi\cos\phi \tag{7c}$$

$$E=\phi'\cos\phi \tag{7d}$$

Now, $|\alpha|$ attains its maxima at values of ϕ where cosα attains its minima. Let

$$\min_{\psi}(\cos\alpha)=m \tag{8}$$

ψ_0 being the particular value of ψ at which cosα=m.

Fig. 4 shows a mechanism configuration at which the follower is at its lower dwell. Hence,

$$\cos c=\beta \tag{9}$$

$$\sin c=\frac{\rho_0}{a}=(1-\beta^2)^{1/2} \tag{10}$$

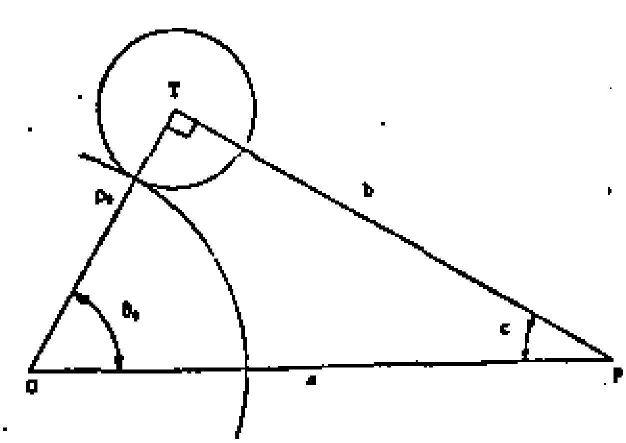


Fig 4 Follower at its lower dwell

Substitution of eqs(1,8,9&10) into eq(5) leads to

$$F\beta^4 + 2G\beta^2 + H^2 = 0 \tag{11a}$$

where

$$F = K_1^2 + 4s_o^2K_2^2 \tag{11b}$$

$$G = K_1H - 2s_o^2K_2^2 \tag{11c}$$

$$H = c_o^2 - m^2 \tag{11d}$$

$$K_1 = s_o^2 - c_o^2 - m^2[(1+\sigma_o')^2 - 2(1+\sigma_o')c_o] \tag{11e}$$

$$K_2 = (1+\sigma_o')m^2 - c_o \tag{11f}$$

with

$\sigma_o = \sigma(\psi_o)$, $\sigma_o' = \sigma_o'(\psi_o)$, $c_o = \cos\sigma_o$, $s_o = \sin\sigma_o$

From eq(11a) one obtains β as a function of ψ_o, which then is substituted into eq(7a) together with eqs(1,8,9&10), thus producing a nonlinear equation, $f(\psi_o)=0$, in one single unknown, ψ_o.

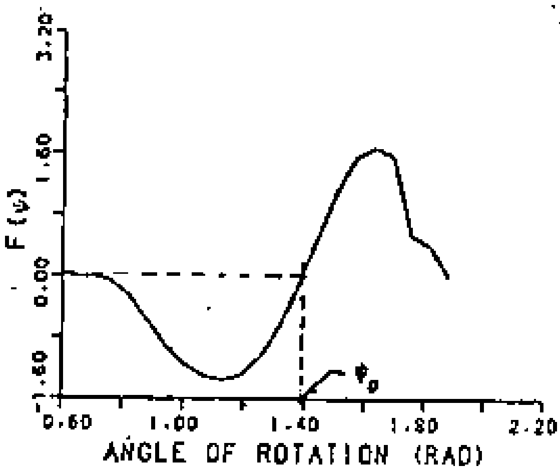


Fig. 5 Plot of function $f(\psi)$

Fig 5 shows a plot of $f(\psi)$ for the conditions given in the Example. The value of ψ at which $f(\psi)$ vanishes, ψ_o, can be obtained either visually from that plot or iteratively, using a suitable algorithm. As previously done [18&19], subroutine ZEROIN [23] is used within the program written for the synthesis of this type of follower. Once ψ_o is determined, σ_o and hence σ_o' and σ_o'' can be readily determined from function $\sigma(\psi)$, of Fig 2. With σ_o and σ_o' already determined, parameter β is obtained from eq(11a). Constant c can thus be computed from eq(9). Function $\phi(\psi)$ is therefore totally determined. Substituting these values into eqs(2&3), for a discrete set of values of ψ, $\{\psi_i\}_1^n$, a set of pairs $\{\rho_i/a, \theta_i\}_1^n$ can be readily computed. These determine n points of the pitch curve, which are now used as the supporting points of a cubic periodic parametric spline used to trace the normalized said curve, i.e. for $a=1$. The actual size of this curve is then obtained once parameters a and r, shown in Fig 3, are defined.The radius of the roller for a unit length a, r/a, can be computed from the pitch curve for a unit length a, by prescribing r/a to be a given fraction q of the minimum radius of curvature of the said pitch curve, in order to avoid undercutting. The said fraction can be determined, in turn, from contact stress considerations, as pointed out in [2]. The radius of curvature of the pitch curve, r_p, and the pressure angle, α, are given by [2]

$$r_p = \frac{a\sin^2\phi}{(1-\phi')\cos^2\alpha\left(\frac{\sin\phi}{\cos\phi} + \frac{\phi'}{1-\phi'}\sin(\alpha+\phi)\frac{\phi''}{(1-\phi')^2}\cos(\alpha+\phi)\right)} \tag{12}$$

$$\alpha = \tan^{-1}\left[\cot\phi - \frac{\beta(1-\phi')}{\sin\phi}\right] \tag{13}$$

The software realizing the method presented here produces $r_p(\psi)$ both numerically and graphically. Moreover, it yields the extrema of this function with the aid of subroutine DMIN [23]. Fig 6 shows a plot of r_p/a for the conditions given for the Example. Let r_m be the minimum value of r_p/a, p_p, p_c and n denoting the position vectors of corresponding points on the pitch curve and on the cam profile, as well as the unit normal vector of both curves at corresponding points, respectively. The cam profile is thus synthesized from the relation [19]:

$$p_c = p_p - rn \tag{14}$$

The cam profile obtained from eq(14) corresponds to a unit value of parameter a. This can be chosen, in turn, from considerations of space availability together with maximum allowable value of the contact stress. Once this parameter is determined, the actual cam profile is determined by scaling the foregoing normalized parameters, which is done by a simple multiplication.

EXAMPLE

Synthesize a cam follower mechanism for an oscillating roller follower, which will produce the follower displacement program appearing in Table 1. The pressure angle is to attain a maximum absolute value of 30°, the amplitude of the follower oscillations being prescribed as 45°

Phase	Angle of rotation (ψ)	Displacement
D_1 (dwell)	36°	0°
R^1 (rise)	72°	+45°
D_2 (dwell)	144°	0°
R^2 (return)	108°	-45°

Table 1 Prescribed angular-displacement program

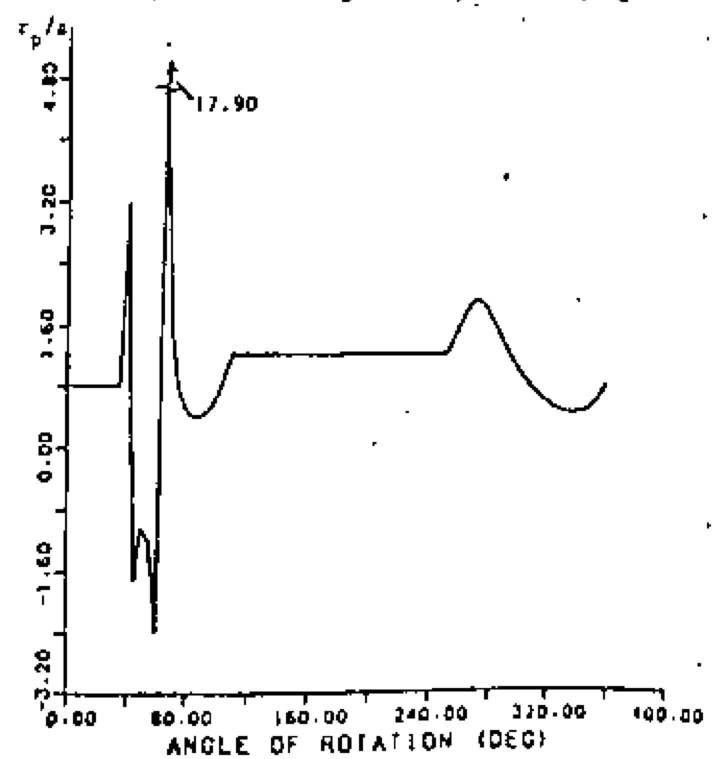


Fig. 6 Plot of r_p/a

The synthesis was executed with the software developed for the implementation of the method presented here. The values obtained, for a unit value of a, were

$$\psi_o = 82.61°$$

$$c = 55.64°$$

$$\beta = 0.57$$

$$r_m = 0.38$$

Prescribing q=0.75 the radius of the roller was thus set as

$$r/a = 0.285$$

Function $f(\theta)$, and the radius of curvature of the pitch curve are plotted in Figs 5 and 6, respectively. The synthesized cam profile is shown in Fig 7. Finally, the software produced the following geometric parameters:

Area of the cam disk = 2.0961
Centroid coordinates: x = -0.1877, y = -0.0805
Principal moments of
inertia at the centroid: I_1 = 0.4544, I_2 = 0.3596

The principal axes of inertia, E_1, E_2, corresponding to I_1 and I_2, are shown in Fig 7.

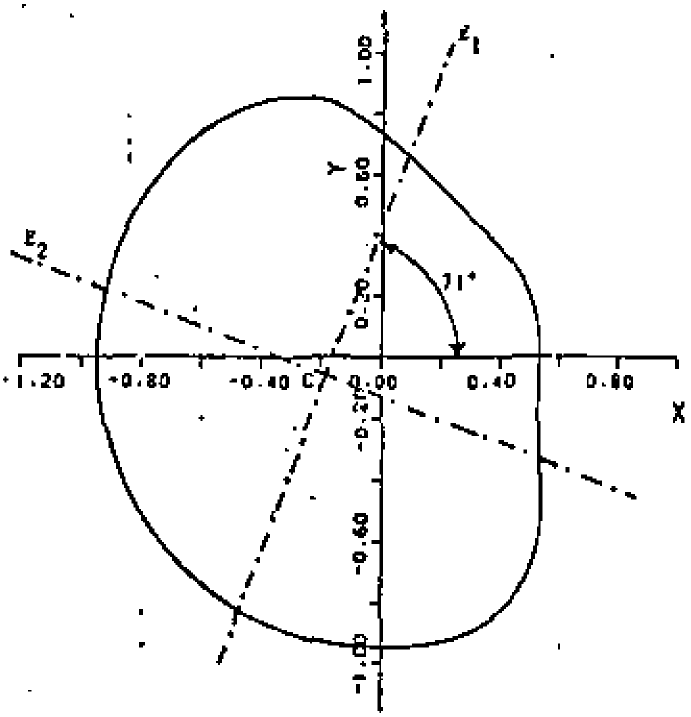


Fig. 7 Cam profile

CONCLUSIONS

The method presented here implicitly produces a minimum-size cam disk for an oscillating roller-follower, moving according to a prescribed angular-displacement program. In fact, by imposing the condition that the maximum absolute value of the pressure angle attains a given value α_M, the procedure produces the minimum-size cam disk. The use of cubic periodic splines, for both the synthesis of the angular-displacement program of the follower and the synthesis of the pitch curve, and hence for that of the cam profile, allows a relatively simple computation of the geometrical parameters of the entire cam mechanism. The software implementing the foregoing method gives the designer the freedom to choose the radius of the roller by allowing him/her to specify it as a fraction, q, freely chosen. Moreover, the designer can determine freely the overall size of the mechanism by properly choosing parameter a. He/she can do this considering space availability and maximum contact stress. Since the paper is concerned with the pure geometric synthesis of the mechanism, such considerations were left aside here. The software can be integrated, however, to a more general CAD program [4] enabling the designer to couple it to a FEM package allowing him/her to consider such effects as contact stress, stress concentrations and failure criteria. Finally, the software implementing this method produces geometric parameters such as area, centroid location and principal moments of inertia, that are necessary for a static and dynamic analysis of the mechanism.

ACKNOWLEDGEMENTS

The research work reported here was performed at the CAD Laboratory of the Graduate Division of the Faculty of Engineering-National Autonomous University of Mexico. The authors gratefully acknowledge the full support of this Division (DEPFI-UNAM).

REFERENCES

1. Mischke, C., "Optimal Offset on Translating Follower Plate Cams", Journal of Engineering for Industry, Vol. 92, No. 3 Feb. 1970, pp.172-176.

2. Sermon, C. F., and Liniecki, A., "Search for Optimum Solution of a Single Disk Cam Mechanism with an Oscillating Roller Follower", ASME Paper No. 72-MECH-61, 1972.

3. Buchsbaum, F., and Freudenstein, F., "On a Class of Cam-Type Angular Motion Compensators", Journal of Engineering for Industry, Vol.95, No.2, May 1973, pp.4[illegible].

4. Tesar, D., and Matthew, G. K., The Dynamic Synthesis, Analysis, and Design of Modeled Cam Systems, Lexington Books, Lexington, Massachusets, 197[illegible]

5 Chakraborty, J., and Dhande, S. G., Kin[illegible] and Geometry of Planar and Spatial Cam Mechanisms, [illegible] Wiley, New York, 1977.

6. Hiroshi, T., Kinematics Design of Cam Follower Systems, Dissertation, Columbia University, 1976.

7. Chen, F. Y., "A Survey of the State of the Art of Cam Systems Dynamics", Mechanism and Machine Theory, Vol. 12, 1977, pp.201-224.

8. Angeles, J., and Arteaga, O., "Optimal Synthesis of Cam Mechanisms via the Methods of Newton-Raphson and Runge-Kutta", Second IFToMM International Symposium on Linkages and Computer-Aided Design Methods, Bucharest, Romania, Vol. III, 1977, pp.[illegible]-12.

9. Grant, B., and Soni, A.H., "A Survey of Cam Manufacture Methods", Journal of Mechanical Design, Vol.101, Jul. 1979, pp.455-564.

10. Nó Sánchez, M., and García de Jalón, J., "Application of B-Spline Functions to the Motion Specification of Cams", ASME Paper No. 80-DET-28, 1980.

11. Berzak, N., "Optimization of Cam-Follower Systems With Kinematic and Dynamic Constraints", Journal of Mechanical Design, Vol. 104, Jan. 1982, pp.29-33.

12. Chen, F. Y., Mechanics and Design of Cam Mechanisms, Pergamon Press, New York, 1982.

13. Di Benedetto, A., and Vinciguerra, A., "Kinematic Analysis of Plate Cam Profiles not Analytically Defined", Journal of Mechanical Design, Vol. 104, Jan. 1982, pp. 34-38.

14. Rao, S. S., and Gavane, S. S., "Analysis and Synthesis of Mechanical Error in Cam-Follower Systems", Journal of Mechanical Design, Vol. 104, Jan. 1982, pp. 52-62.

15. Ghosh, A., and Yadav, R.P., "Synthesis of Cam-Follower Systems with Rolling Contact", Mechanism and

Machine Theory, Vol. 18, No. 1, 1983, pp.49-56.

16. Terauchi, Y., and El-Shakery, S. A., "A Computer - Aided Method for Optimum Design of Plate Cam Size Avoiding Undercutting and Separation Phenomena-I", Mechanism and Machine Theory, Vol. 18, No. 2, 1983, pp.157-163.

17. Terauchi, Y., and El-Shakery, S. A., "A Computer - Aided Method for Optimum Design of Plate Cam Size Avoiding Undercutting and Separation Phenomena-II", Mechanism and Machine Theory, Vol. 18, No. 2, 1983, pp.164-170.

18. Angeles, J., and López-Cajún, C., "Diseño Automatizado de Mecanismos de Leva de Disco con Seguidor Traslacional de Cara Plana", Memoria del IX Congreso de la ANIAC, León, Gto., México, Sep. 1983, pp.91-93.

19. Angeles, J., and López-Cajún, C., "Optimal Synthesis of Translating-Follower Cam Mechanisms With Prescribed Functional Constraints", International Symposium on Design and Synthesis, Tokyo, Japan, Jul. 1984.

20. Angeles, J., "Synthesis of Plane Curves With Prescribed Local Geometric Properties Using Periodic Splines", Computer-Aided Design, Vol. 15, May 1983, pp. 147-155.

21. Angeles, J., "Síntesis de Curvas Planas Cerradas Usando Funciones "Spline" Paramétricas y Periódicas", Revista de la ANIAC, México, Vol. 2, No. 1, Mar. 1983, pp.53-81.

22. Späth, H., Spline-Algorithmen zur Konstruktion glatter Kurven und Flächen, 2nd Ed, R. Oldenburg, Munich, FRG, 1978, pp.27-28.

23. Forsythe, G., Malcolm, M. A., and Moler, C. B., Computer Methods for Mathematical Computations, Prentice Hall, New Jersey, 1977.

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO CINEMATICO DE MAQUINARIA

DISEÑO AUTOMATIZADO DE MECANISMOS DE LEVA DE DISCO CON SEGUIDOR TRASLACIONAL DE CARA PLANA

DR. JORGE ANGELES ALVAREZ

DR. CARLOS S. LOPEZ CAJUN

JUNIO, 1984.

DISEÑO AUTOMATIZADO DE MECANISMOS DE LEVA DE DISCO CON SEGUIDOR TRASLACIONAL DE CARA PLANA

Jorge Angeles Alvarez
Carlos S. López Cajún
División de Estudios de Posgrado de
la Facultad de Ingeniería, UNAM
Apdo. Postal 70-256
04510 México, D.F.

Resumen

Se presenta la síntesis automática de mecanismos de leva con seguidor traslacional de cara plana. La curva de desplazamiento del seguidor se sintetiza mediante curvas spline periódicas, las cuales satisfacen condiciones prescritas de aceleración. En base a lo anterior se determina el radio óptimo del círculo base para un descentramiento máximo dado. Finalmente, el perfil de la leva se obtiene mediante curvas spline paramétricas periódicas, que interpolan los puntos generados por las ecuaciones de síntesis.

Abstract

The computer-aided synthesis of disk cam mechanisms with translational flat face follower is presented. The follower displacement program is synthesised by periodic spline curves, which satisfy prescribed acceleration conditions. Based on the foregoing, the optimum radius of the base circle is determined for a given maximum contact point eccentricity. Finally, the cam profile is obtained by fitting periodic parametric spline curves through the points generated by the synthesis equations.

Introducción

Con el advenimiento de las máquinas herramienta de control numérico, la manufactura de levas es más confiable y exacta [1]. Por otra parte, durante las últimas décadas se han realizado diversos estudios sobre el análisis y la síntesis de mecanismos de leva [2], muchos de los cuales involucran el uso de la computadora. En particular, sobre diseño óptimo de levas, pueden citarse los trabajos de Chicurel [3], Mischke [4] y Angeles y Arteaga [5], entre otros. En este trabajo se presenta un caso particular de síntesis de levas; pero, a diferencia de la práctica tradicional, que se basa en el uso de un número limitado de funciones que contienen a su vez un número limitado de parámetros independientes, aquí se muestra el uso de curvas spline periódicas sintetizadas en forma tal que satisfacen condiciones de aceleración prescritas y que garantizan continuidad en la velocidad del seguidor [6,7]. La introducción de funciones spline permite contar con un número arbitrario de parámetros independientes (los coeficientes de la spline), que permiten satisfacer un número igualmente arbitrario de condiciones sobre el desplazamiento del seguidor. Las curvas spline han sido utilizadas por Mó Sánchez y García de Jalón [6] para obtener los programas del seguidor. Sin embargo, en este trabajo, dichas curvas se encuentran almacenadas en un banco de datos y, mediante escalamiento (y/o reflexión), pueden generarse los programas de desplazamiento de seguidor requeridos, de los cuales se pueden obtener los puntos de velocidad máxima y por lo tanto, el radio óptimo del círculo base para un descentramiento máximo permitido. Por último, conociendo el radio del círculo base, el perfil de la leva se obtiene al interpolar los puntos generados por las ecuaciones de síntesis, mediante curvas spline paramétricas periódicas. Las funciones spline periódicas usadas aquí para representar el desplazamiento del seguidor son de la forma

$$s(\psi)=a_k(\psi-\psi_k)^3+b_k(\psi-\psi_k)^2+c_k(\psi-\psi_k)+d_k$$

para $\psi_k \le \psi \le \psi_{k+1}$

donde a_k, b_k, c_k $(k=1,\ldots,n)$ se obtienen de las condiciones de periodicidad $s(0)=s(2\pi)$, $s'(0)=s'(2\pi)$, $s''(0)=s''(2\pi)$ y de continuidad en $\psi_1, \psi_2, \ldots, \psi_n$ tanto en $s(\psi)$ como en $s'(\psi)$ y en $s''(\psi)$, para satisfacer valores prescritos de $s''(\psi)$ en $\psi_1, \psi_2, \ldots, \psi_n$. (Los detalles pueden verse en [7]).

Definición del Problema

Dados los intervalos t_1, t_2, t_3 y t_4, así como la elevación del seguidor, h, que definen el programa de desplazamiento de éste, ver Fig 1, obtener el perfil de la leva de área mínima, que genere ese programa y que tenga un descentramiento máximo dado del punto de contacto, PA, de la Fig 2.

Síntesis del programa de desplazamiento del seguidor

Se requiere sintetizar la curva $s(\psi)$ vs. ψ de la Fig. 1, conectando las fases de reposo R_1 y R_2, de forma tal que E y B sean tangentes a R_1 y R_2 en ψ_1, ψ_2, ψ_3, y ψ_4. Esto es para garantizar que la velocidad del seguidor sea continua en los puntos de conexión. Además, dado que la aceleración del seguidor es una función lineal de $s''(\psi)$ y con el fin de garantizar la continuidad de dicha aceleración en ψ_1, ψ_2, ψ_3 y ψ_4, $s''(\psi)$ debe anularse en estos puntos. Más aún, se desea que la aceleración cambie suavemente entre ψ_1 y ψ_2 y entre ψ_3 y ψ_4. Lo anterior se puede lograr [7] mediante el escalamiento adecuado del tramo PQ de la curva spline periódica de la Fig. 3, la cual se encuentra almacenada en un banco de datos. La síntesis de la curva B, definida entre ψ_3 y ψ_4, se logra mediante la reflexión y escalamiento del mismo tramo PQ.

Síntesis del perfil de la leva

Considerando la Fig. 2, sean M y L líneas fijas al marco del mecanismo y a la leva, respectivamente. El ángulo ψ mide la rotación de la leva con respecto al marco, en tanto que $s(\psi)$ puede considerarse como la suma de una constante c, el radio del círculo base, más una función positiva definida $a(\phi)$, cuyo valor mínimo es cero, siendo su valor máximo igual a h.

De la geometría de la Fig. 2, se tiene:

$$s(\phi)=\rho(\phi)\operatorname{sen}(\theta+\phi) \qquad (1)$$

Igualando las velocidades en la dirección vertical del punto A, se tiene:

$$v_{AL} = v_{AS} \qquad (2)$$

donde v_{AL} es la componente vertical de la velocidad de A como punto de la leva y v_{AS} es la velocidad de A como punto del seguidor; por lo tanto:

$$\rho(\phi)\dot{\phi}\cos(\theta+\phi)=\dot{s} \qquad (3a)$$

o

$$\rho(\phi)\cos(\theta+\phi)=s'(\phi) \qquad (3b)$$

De (1) y (3b) se tiene:

$$\rho^2(\phi) = s^2(\phi) + s'^2(\phi) \qquad (4)$$

$$\theta(\phi) = \tan^{-1}\left[\frac{s}{s'}\right]-\phi \qquad (5)$$

Combinando (4) y (5) se obtiene el perfil de la leva dado por $\rho = \rho(\theta)$.

Determinación del radio óptimo del círculo base

De la geometría de la Fig. 2, se tiene:

$$\overline{BA} = \rho\cos(\theta+\phi) \qquad (6)$$

Sustituyendo (6) en (3a) y despejando $\overline{BA}$:

$$\overline{BA} = \frac{\dot{s}}{\dot{\phi}} = \frac{ds}{d\phi} \qquad (7)$$

Puesto que:

$$\overline{BA} \le \overline{BA}_{máx} \qquad (8)$$

donde

$$\overline{BA}_{máx} = \epsilon_{máx}c$$

siendo $\epsilon_{máx}$ el máximo descentramiento (por unidad de longitud de radio del círculo base) permitido, se tiene:

$$\frac{\overline{BA}}{c} \le \epsilon_{máx} \qquad (9)$$

y, de (7),

$$s'(\phi) \le \epsilon_{máx}c \qquad (10)$$

de donde puede obtenerse el radio óptimo del círculo base

$$c_{opt} = \frac{\text{Máx}|s'(\phi)|}{\epsilon_{máx}} \qquad (11)$$

Las velocidades máximas corresponden a los puntos de inflexión de las curvas E y B, esto es, a puntos donde la aceleración se anula y que por construcción de las curvas corresponden a los valores $\phi = (\phi_1+\phi_2)/2$ y $\phi = (\phi_3+\phi_4)/2$

Descripción del algoritmo

1. Lee los intervalos t_1, t_2, t_3, t_4, la elevación h y el máximo descentramiento permitido $\epsilon_{máx}$
2. Genera las curvas de elevación y descenso del seguidor mediante escalamiento adecuado. Grafica esta curva
3. Obtiene c_{opt}
4. Obtiene $\theta(\phi)$ y $\rho(\phi)$
5. Genera $\rho(\theta)$. Lo grafica mediante curvas spline paramétricas periódicas

Ejemplo

Obténgase el perfil de la leva que produzca el movimiento del seguidor mostrado en la Tabla 1 para un descentramiento máximo de 50%, y una elevación h=5 unidades de longitud. El perfil obtenido se muestra en la Fig. 4.

Conclusiones

Se mostró un procedimiento para obtener, en forma automática, tanto el programa de desplazamiento del seguidor como el perfil de la leva de área mínima con un descentramiento máximo prescrito del punto de contacto. El algoritmo utilizado se realizó en un programa de computadora, al cual tiene acceso el usuario en forma conversacional y proporciona los resultados tanto en forma numérica como gráfica. En el último caso, se pueden obtener éstos en pantalla o en copia dura mediante el uso de un graficador. Se utilizaron para este fin las instalaciones del Laboratorio de Cálculo Automatizado para el Diseño de la División de Estudios de Posgrado de la Facultad de Ingeniería, UNAM.

Referencias

1. Grant B. y Soni A.H., "A survey of cam manufacture methods", Journal of Mechanical Design, July 1979, Vol. 101, pp. 455-464.
2. Chen F.Y., "A survey of the state of art of cam system dynamics", Mechanism and Machine Theory, 1977, Vol. 12, pp 201-224.
3. Chicurel R., "Cam size minimization by offsetting", ASME paper No. 62-WA-22, 1962
4. Mischke C., "Optimal offset on translating follower plate cams", Journal of Engineering for Industry, February 1970, Vol. 92.
5. Angeles J. y Arteaga O., "Optimal synthesis of cam mechanisms via the methods of Newton-Raphson and Runge-Kutta", Second IFToMM International Symposium on Linkages and Computer Aided Design Methods, Bucharest, Romania, June 16-21, 1977, Vol. III, pp 1-12.
6. Ro Sánchez M. y García de Jalón J., "Application of B-spline functions to the motion specification of cams", ASME paper No. 80-DET-28, 1980.

7. Angeles J., "Synthesis of plane curves with prescribed local geometric properties using periodic splines", Computer-Aided Design, May 1983.

8. Angeles J., Análisis y Síntesis Cinemáticos de Sistemas Mecánicos, Editorial Limusa, C. de México, 1978.

TABLA 1

Fase	Intervalo	Tipo de movimiento
1	$t_1 = 36°$	reposo (R_1)
2	$t_2 = 108°$	elevación $(E\ )$
3	$t_3 = 144°$	reposo (R_2)
4	$t_4 = 72°$	descenso $(B\ )$

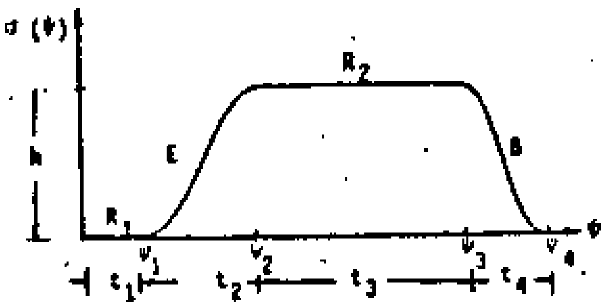


Fig. 1 Programa de desplazamiento del seguidor

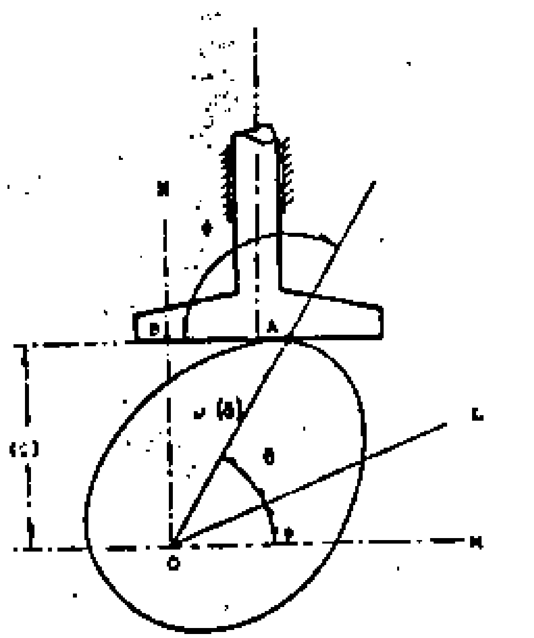


Fig. 2 Leva de disco con seguidor de cara plana

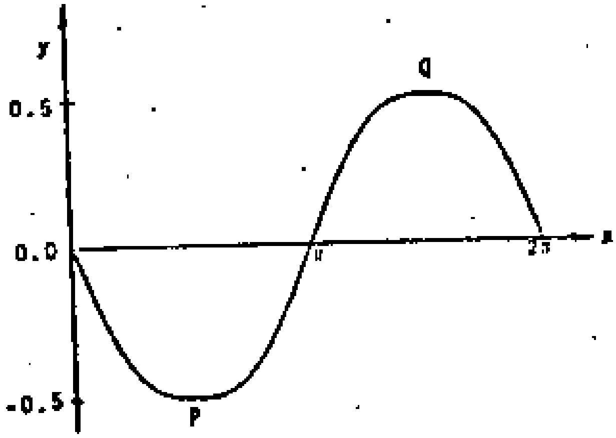


Fig. 3 Curva spline periódica

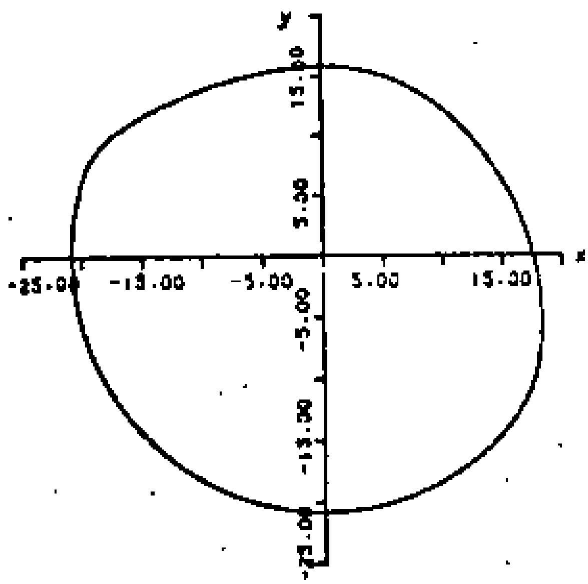


Fig. 4 Perfil de la leva

Nota final: El perfil de la leva se obtuvo con 37 puntos de apoyo distribuidos de la siguiente manera: 9 puntos de apoyo en cada período de reposo y 10 puntos de apoyo en los períodos de ascenso o descenso.